Proc. 2003 IEEE/RSJ Int. Conf. on
Intelligent Robots and Systems,
pp. 2974-2979, Las Vegas, NV, 2003.

A View-Based Outdoor Navigation Using Object Recognition Robust to Changes of Weather and Seasons

Hiroaki Katsura[†], Jun Miura[†], Michael Hild[‡], and Yoshiaki Shirai[†]
[†]Department of Computer-Controlled Mechanical Systems, Osaka University
[‡]Department of Engineering Informatics, Osaka Electro-Communication University
{katsura,jun,shirai}@cv.mech.eng.osaka-u.ac.jp, hild@hilab.osakac.ac.jp

Abstract

This paper describes a view-based outdoor navigation method. In the method, a user first guides a robot along a route. During this guided movement, the robot learns a sequence of images and a rough geometry of the route. The robot then moves autonomously along the route with localizing itself based on the comparison between the learned images and input images. Since appearances of objects in images may vary much according to changes of seasons and weather in outdoor scenes, a simple image comparison does not work. We, therefore, propose a comparison method in which the robot first recognizes objects in images using object models which allow for appearance variations, and then compares recognition results of learned and input images. We also developed a method which automatically selects key images used for the comparison from an image sequence. Successful autonomous navigation experiments in our campus under various conditions show the feasibility of the method.

1 Introduction

Vision-based outdoor navigation of mobile robots is one of the active research areas in robotics. Many previous works use local visual features such as road boundaries for controlling robot motion [3, 1]. If such features are not necessarily available, methods for global localization are required for navigation.

Recently, many works have started using GPS for localization (e.g., [11]). GPS can usually provide reasonably accurate position information, but is not fully reliable because it cannot work well near a tall building due to multipath problems or occlusion of satellites.

Outdoor environments are generally much more complex and wider than indoor environments, so it is usually difficult for a user to give a robot a map of the environment in advance. A promising approach is thus the following two-phase one: a user first guides a robot along a route for learning the environment and then the robot moves along the route autonomously using the learned knowledge.

Several two-phase methods have been proposed, each of which has a different map representation and a map acquisition method. Kidono et al. [5] used a set of landmarks obtained by stereo as a map. Maeyama et al. [7] recorded odometry data and landmark positions around a guided path. These methods use nearby objects as landmarks and assume such objects are static; they may not be applicable to the case where objects such as cars change their positions in the learning and the navigation phase.

Views of objects are useful cues for global localization. Matsumoto et al. [8] developed a navigation method which is based on an image sequence obtained in the learning phase. Li [6] proposed a similar method for a panoramic image obtained by a movement along a route. Takeuchi et al. [10] developed a localization method based on a similarity between color distributions in an input image and a learned image. These methods do not consider color changes of objects according to changes of weather and/or seasons and, thus, the localization may not be robust enough.

Although appearances of objects may vary depending on weather or seasons, the location of relatively large objects such as buildings and trees do not change. We, therefore, propose a global localization method based on a comparison of such large objects in input and learned images. The most important issue is thus to develop an object recognition method robust to changes of weather and seasons.

We take a knowledge-based approach to outdoor scene recognition (e.g., [9, 2]). A feature of ours is to use multiple object color models which consider their variations according to changes of weather and seasons. We also use a flexible recognition and matching method which allows for multiple hypotheses on the type of a region in recognition; such a hypothesis ambiguity is resolved by adopting a flexible matching.

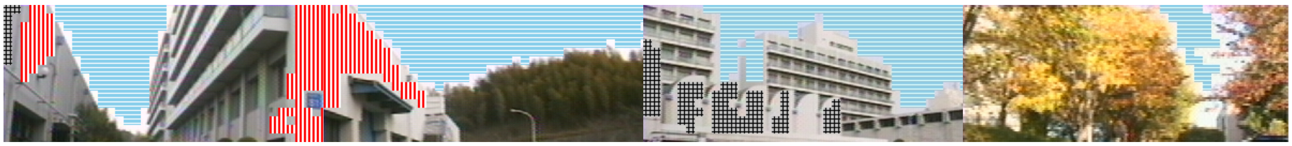
Our robot primarily uses vision for global localization but additionally uses odometry and GPS (where available) to obtain a rough estimate of robot position. It also uses a laser range finder to locally avoid obstacles.

2 Object Recognition

We are currently interested in navigation in our campus, where buildings, trees, cars, bicycles, and other small objects exist. Since cars and bicycles are different time to

(a) (b) (c) (d)

Fig. 1: Input color images.



(a) (b) (c) (d)

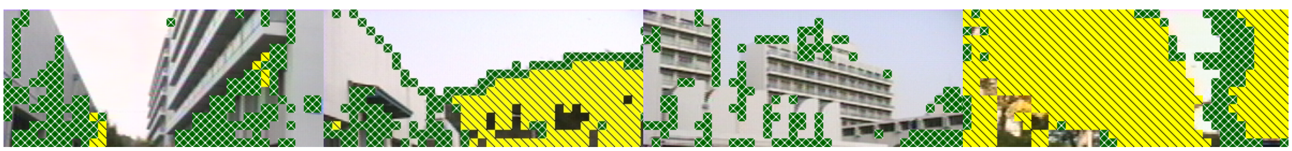
Uniform region. Sky-or-uniform region. Sky region.

Fig. 2: Recognition results of uniform and sky regions.

(a) (b) (c) (d)

Building region.

Fig. 3: Recognition results of building regions.



(a) (b) (c) (d)

Tree region. Tree-possible region.

Fig. 4: Recognition results of tree regions.

time, we use buildings, trees, and the sky which exist in the upper half of input images as landmarks, and develop methods for recognizing them. Specifically, we consider the following four object regions:

- *Uniform regions*: large uniformly-colored regions corresponding to sidewalls of buildings.
- *Sky regions*: regions corresponding to the sky.
- *Building regions*: regions with long segments corresponding to windows and boundaries of buildings.
- *Tree regions*: regions corresponding to trees.

In recognizing an image (resolution is 304 (width) $\times$ 128 (height)), we divide an image into a set of small windows, examine colors and edges of each window, and classify the window into one of the above objects based on the colors and the edges.

2.1 Recognition of Uniform and Sky Regions

The sky and sidewalls of buildings usually constitute large uniform regions in images, so we first extract such regions and then classify them.

We divide an image into windows of 8×8 pixels in size, and extract edges whose magnitudes by Sobel operator are larger than a threshold (currently, 10) in each window. If the number of edges is less than a threshold (currently, 10) and if the variances of R, G, and B values are all less than a threshold (currently, 300), the window is extracted as a uniformly-colored one. Such windows are then merged into large uniform regions.

From the knowledge of position, color, and shape of the sky in the image, we define the following condition that a region is the sky:

- It touches the upper boundary of an image.
- Its average intensity is larger than 120 (in 8 bits).
- It is composed of 10 or more windows.
- The width of its upper part is larger than that of its lower part.

If the color of the sky gradually changes, or if there are blue sky parts and clouds parts in the sky, the sky is divided into smaller uniform regions and thus may not be recognized as a single sky region at first. So we examine regions adjacent to sky regions which have already been recognized, and if such an adjacent region satisfies the following conditions:

- Its average intensity is larger than 120.
- The number of edges on the boundary between the region and adjacent sky regions is small enough (currently, 0.075%).
- The boundary length with respect to the area of the region is longer than a threshold (currently, 0.15).

then this region is classified as a sky region. This examination is repeated until no more sky regions are found.

After extracting sky regions, the remaining uniform regions which do not touch the upper boundary of an image are classified as *uniform regions*. For the rest of uniform regions (which touch the upper boundary), if a region has a straight-line boundary at the top of the region, the region is classified as a uniform region; otherwise, we consider that we cannot determine the class of the region and call it a *sky-or-uniform region*.

Fig. 2 shows the recognition results of sky and uniform regions in the images shown in Fig. 1. In Fig. 2(b), a sidewall of a building on the left is classified as a uniform-or-sky region because the boundary between the building and the sky is not strong enough. In Fig. 2(c), a sidewall in the center is merged with the sky region. The rest parts are all correctly recognized.

2.2 Recognition of Building Regions

Boundaries of buildings and windows have long straight edge segments. We use such edge segments as cues for detecting buildings. For detecting long edges, we divide an image into windows of 16×16 pixels in size, and extract edges whose magnitudes are larger than a threshold (currently, 30) in each window. If the number of edges is larger than a threshold (currently, 10), we apply a Hough transform to detect long straight edge segments. We then merge connected windows having such edge segments and if a merged set of windows is large enough (composed of 10 or more windows), the windows are recognized as building regions. Fig. 3 shows the recognized building regions in the input images shown in Fig. 1.

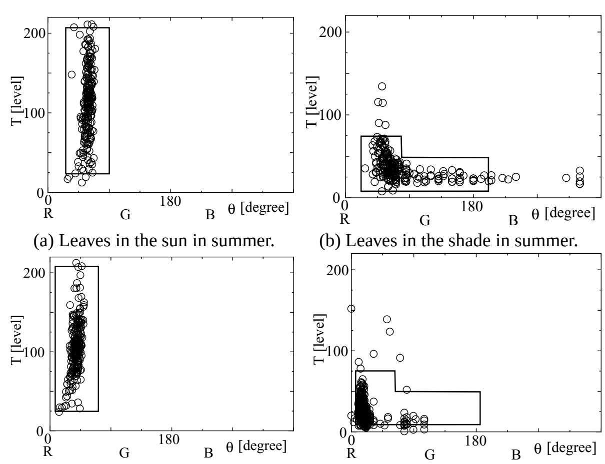
2.3 Recognition of Tree Regions

Tree regions have a large number of edges of branches and leaves. We divide an input image into a set of windows of 8×8 pixels in size, and if the number of edges of a window is equal to or larger than a threshold (currently, 10) and the number of pixels whose colors match with that of trees exceeds a threshold (80% of the window size), the window is classified as a tree region.

Colors of trees vary according to changes of weather and seasons. We manually extracted tree regions in various scenes, and examined the relationship between intensity (T) and hue (θ) in these regions. Fig. 5 shows the relationships in four cases. In the case of leaves in the sun in summer (see Fig. 5(a)), the hue value does not change much even if the intensity changes. In the case of leaves in the shade in summer (see Fig. 5(b)), the hue shifts to blue due to the ambient light from the blue sky, especially in low-intensity regions. In the case of colored leaves in fall (see Fig. 5(c)(d)), the hue slightly shifts to red. We also examined cloudy cases and found that the intensity becomes smaller but the hue does not change much. From these examinations, we model the leaf colors by the four polygonal regions in Fig. 5. If the number of pixels in a window whose colors match with at least one of models (a) and (c) in Fig. 5 exceeds a threshold (currently, 80% of the window size), the window is classified as a tree region. If the number of pixels in a window whose colors match with at least one of models (b) and (d) in Fig. 5 exceeds the same threshold, the window is classified as a *tree-possible* region, since it may be a tree region but cannot be definitely determined to be so.

In the case of trees with leaves being fallen, background objects may be visible through branches, and their colors may not match with the tree color models. Concerning edges of branches and leaves, in general, their magnitudes are weak and their directions are widely distributed. Fig. 6 shows example distributions of accumulated edge magnitudes for a tree region without leaves, that with leaves, and a building region. The resolution of direction is 20 degrees; the accumulated edge magnitude in each direction is normalized by the number of edges in a window. The figure clearly indicates that tree regions has a much lower maximum value than the building region. From this observation, we calculate this distribution for a region which has an enough number of edges but has a different color from the tree color models, and if the maximum value is less than a threshold (currently, 12), the region is classified as a *tree-possible region*.

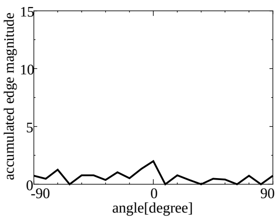
Fig. 4 shows the recognition result of tree regions in the images shown in Fig. 1. Note that both green and colored leaves are correctly recognized.



(a) Leaves in the sun in summer. (b) Leaves in the shade in summer.

(c) Colored leaves in the sun in fall. (d) Colored leaves in the shade in fall.

Fig. 5: Relationship between intensity (T) and hue (θ).



(a) Trees without leaves. (b) Trees with leaves. (c) Buildings.

Fig. 6: Distributions of accumulated edge magnitudes.

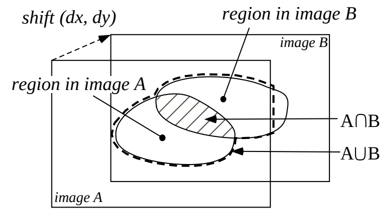


Fig. 7: Similarity between regions is given by $A \cap B$ / $A \cup B$.

3 Evaluation of Matches between Images

A match between two images is evaluated by the similarity of recognition results of every object in both images. The object recognition method outputs four object regions in an image. If two images are taken at roughly the same robot position and orientation, the positions of these object regions in the images should be similar. Since we do not need a very accurate localization, we calculate similarity values for a certain range of relative displacements between two images (obtained by shifting one of the images) and use the highest similarity value as the evaluation of the match between the images.

For each object, the corresponding regions are obtained in both images. The similarity for the object is given by the ratio of the area of the intersection of the regions to that of the union of them (see Fig. 7). We require that the similarity of each object is higher than a threshold for a successful match. The similarity value of a match is then given by the average of those similarity values. When the areas of an object is small in *both* images, however, we do not take the object into consideration because the recognition result of the object is not reliable. Finally, if the similarity of a match is larger than a certain threshold, the match is considered to be successful.

Regions which have not been recognized definitely to be an object (i.e., uniform-or-sky and tree-possible regions) are specially treated in matching. A uniform-or-sky region is treated as a sky region only when the corresponding region in the other image is a sky region; otherwise it is treated as a uniform region. Concerning a tree-possible region, since it is sometimes detected erroneously in a dark region, it is treated as a tree region only when the corresponding region is a tree region; otherwise, we ignore the region as a false detection.

Parameters used in matching are determined as follows. We first set the allowable amounts of the horizontal shift (dx) and the vertical one (dx) to $[-40, 40]$ and $[-24, 24]$ in pixels, respectively, by considering the change of the robot orientation during autonomous movements. To reduce the computation time, actual shift values are selected every 8 pixels in both direction.

The thresholds on the area and the similarity of an object depend on the object. An object which is usually recognized stably can be reliably used in matching, even if its area is small. Our recognition method recognizes objects in the following descending order of recognition stability: the sky, trees, uniform regions, and buildings. So we use 6% and 18% as the thresholds on the area for the first three and the last one, respectively. Concerning the threshold on the similarity, 0.6, 0.3, and 0.2 are used for sky regions, tree regions, and uniform or building regions, respectively. We use 0.4 as the threshold for judging whether two images are matched.

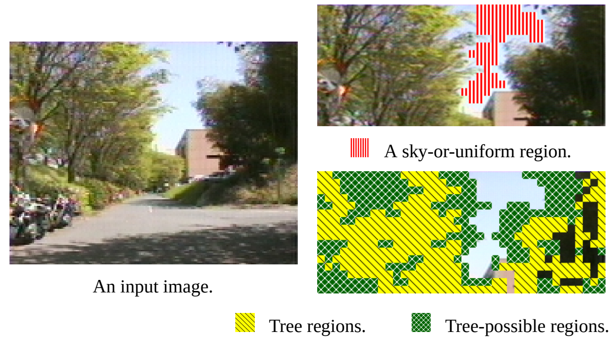


Fig. 8: An input image and its recognition result.

Table 1: Matching result between images in Fig. 1-(d) and Fig. 8.

s_{max}	dx	dy	s_U	s_S	s_B	s_T
0.590	-32	16	—	0.625	—	0.556

Table 1 shows the matching result for the images shown in Fig. 1(d) and Fig. 8. Fig. 8 also shows the recognition results. The table indicates the similarity values for the match (s_{max}), sky regions (s_S), and tree regions (s_T), and the best shift (dx, dy). The similarity value for building regions is not obtained because the regions are small in both images. That for uniform regions is not used either because the sky-or-uniform region in Fig. 8(a) is considered to be the sky in matching with the sky region in Fig. 2(d).

4 Navigation Strategy

We consider that a route consists of corners and straight paths connecting them, and prepare different navigation strategies for these two types. When moving along a straight path, the robot repeatedly performs the image matching for localizing itself and for determining a target direction; it moves towards the direction by a visual feedback using two cameras pointing both forward and backward [4]. For turning at a corner, the robot first detects it using odometry and GPS (if available), and then determines a point to start turning based on the image matching and range finder data.

To realize these strategies, the robot records the following data during a guided movement. For a straight path, it records a sequence of images and odometry data of the positions where these images are taken. From the image sequence, a set of images, called *key images*, are selected which are used for the image matching (see Sec. 5). For a corner, the robot records the image just after turning and the location of the corner obtained by odometry and GPS.

To avoid collision with obstacles such as parking cars, the robot continuously measures the distance to nearby objects using a laser range finder, and takes a collision-avoidance movement if necessary.

5 Selection of Key Images

Key images are selected as follows. At a corner, a key image is the one taken when the robot finished turning, and is easily selected by detecting a turning motion using odometry. For a straight path, since the robot usually performs a zigzag motion when guided by a user, it is important to select images which were taken when the robot certainly directed towards a next corner.

In a zigzag motion of the robot, the change of the robot orientation is considered to be roughly approximated by a trigonometrical function. If the function is almost symmetric about the path the robot is following, time points when the robot directs right forward can be detected by extracting images where the absolute value of image motion (optic flow) is locally maximum. Fig. 9 shows the change of the averaged horizontal image motion during the movement between points (a) and (b) in Fig. 10. $\times$ marks in the figure indicate the frames which were manually selected as key images; all selected key images are almost at local maxima. At the same time, it is not good to use all such images because two images taken at a small position interval are usually very similar to each other. So we repeatedly select key images among such images so that each key image does not successfully match with previously-selected ones [8].

6 Experimental Results

We performed several experiments for testing the recognition, matching, and navigation methods using the path shown in Fig. 10.

6.1 Recognition and Matching

We first prepared 43 key images from an image sequence taken on the path from Start to point (d) (see Fig. 10) on Apr. 19, 2002 (sunny) and another sequence on the path from point (d) to Goal on Sep. 21, 2002 (sunny). We then obtained three sequences of input images with $1\,[sec]$ interval by guiding the robot on the path.

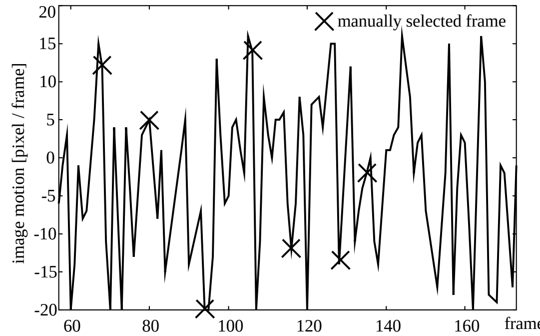


Fig. 9: Change of image motion.

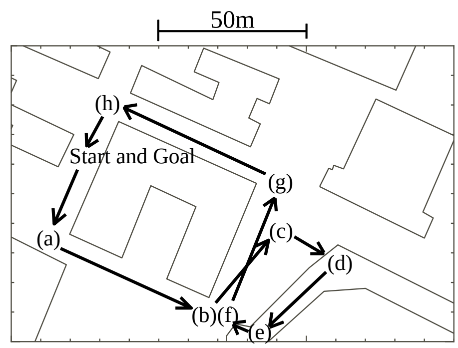


Fig. 10: The path for experiments.

We manually assigned two consecutive key images to each input image as correct matches. The details of input images and the matching results are summarized in Table 2. About 95% of the input images were successfully matched with the corresponding key images. About 57% of the successful matches gained the maximum similarity for the correct key images. There are two cases where correct matches failed to gain the maximum similarity. One is the case where images were taken at positions where the change of view according to the robot motion is small; the other is the case where the maximum similarities were obtained for the key images of completely different positions. Since the matching considers the history of movement in the autonomous navigation phase, such failure of gaining the maximum similarity did not cause a navigation failure.

Table 2: Image details and matching results

Date of image input		3/10/02	10/2/02	1/11/03
Weather		cloudy	sunny	cloudy
Start Position		(a)	(b)	Start
Goal Position		(e)	(e)	Goal
Status of leaves		partially fallen	partially colored	partially fallen
Number of images		124	125	472
success	Max. similarity	60	93	243
	Otherwise	57	28	210
failure		7	4	19

(c) (e)

Fig. 11: Key images.

(c) (e)

Fig. 12: Input images.

The time for recognition and matching is about $1\,[sec]$ in average using a PC with Pentium II (400MHz). Our recognition and matching method uses many parameters (or thresholds) as explained above. Although the parameters were tuned by experiments, they seem to work well under a large variety of conditions.

6.2 Autonomous Navigation

We describe the result of a navigation experiment on Jan. 13, 2003 (cloudy), which used the manually-selected key images mentioned above. The robot successfully moved along the whole path of about $350\,[m]$ in Fig. 10 in about 11 minutes. On the path between points (g) and (h), since GPS data were not available, the robot detected the approach to corner (h) by odometry. Figs. 11 and 12 show the key images and the successfully-matched input ones at corners (c) and (e), respectively. Fig. 13 shows snapshots of the experiment.

We conducted similar experiments 20 times, including the case of using a part of the path, from Sep. 2002 to Jan. 2003. During the experiments, the robot successfully recognized 47 out of 48 corners; one failure case is that tree colors changed too much due to a backlight. The robot failed in matching 27 out of 269 times; these matching failures did not cause a navigation failure, however, since the robot succeeded in the subsequent matchings.

6.3 Two-Phase Approach

We then tested the feasibility of the two-phase approach. That is, we first guided the robot along the path shown in Fig. 10 and the robot followed the path based on automatically-extracted key images. We performed the same experient twice in Mar. 2003 and succeeded in both trials.

7 Conclusions and Discussion

This paper has proposed an outdoor navigation method which localizes a robot by comparison of recognition re-

Fig. 13: Snapshots of autonomous navigation.

sults of input and learned key images. Using multiple color models makes the localization robust to changes of weather and seasons. We have also developed a method for automatically extracting key images from an image sequence based on the motion analysis of the sequence. We successfully applied the method to the navigation of our robot in our campus under various conditions.

The recognition of the corner is, at present, critical to the success of navigation. We are now developing a method for detecting and recovering from a failure of corner recognition. We are also planning to perform experiments at various other places in our campus to test the robustness of the system in more detail.

References

[1] 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.

[2] S. Hirata, Y. Shirai, and M. Asada. Scene Interpretation Using 3-D Information Extracted from Monocular Color Images. In *Proc. IROS-92*, pp. 1603–1610, 1992.

[3] H. Ishiguro, K. Nishikawa, and H. Mori. Mobile Robot Navigation by Visual Sign Patterns Existing in Outdoor Environments. In *Proc. IROS-92*, pp. 636–641, 1992.

[4] H. Katsura, M. Hild, J. Miura, and Y. Shirai. View-Based Outdoor Navigation Using Object Recognition Robust to Changes of Seasons and Weather. In *Proceedings of the 8th Robotics Symposia*, 2003. (in Japanese).

[5] K. Kidono, J. Miura, and Y. Shirai. Autonomous Visual Navigation of a Mobile Robot Using a Human-Guided Experience. *Robotics and Autonomous Systems*, Vol. 40, No. 2-3, pp. 121–130, 2002.

[6] S. Li. Localization along Routes Based upon Iconic and Global Positioning System Information in Large-Scale Outdoor Environments. *Advanced Robotics*, Vol. 15, No. 7, pp. 749–762, 2001.

[7] S. Maeyama, A. Ohya, and S. Yuta. Autonomous Mobile Robot System for Long Distance Outdoor Navigation in University Campus. *J. of Robotics and Mechatronics*, Vol. 9, No. 5, pp. 348–353, 1997.

[8] Y. Matsumoto, M. Inaba, and H. Inoue. Visual Navigation using View-Sequenced Route Representation. In *Proc. ICRA-96*, pp. 83–88, 1996.

[9] T.M. Strat and M.A. Fischler. Context-Based Vision: Recognizing Objects Using Information from Both 2-D and 3-D Imagery. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 13, No. 10, pp. 1050–1065, 1991.

[10] Y. Takeuchi and M. Hebert. Evaluation of Image-Based Landmark Recognition Techniques. Technical Report CMU-CS-95-116, The Robotics Institute, Carnegie Mellon University, July 1998.

[11] R. Thrapp, C. Westbrook, and D. Subramanian. Robust Localization Algorithms for an Autonomous Campus Tour Guide. In *Proc. ICRA-2001*, pp. 2065–2071, 2001.