Outdoor Autonomous Navigation Using SURF Features

M. Tabuse[1], T. Kitaoka[2] and D. Nakai[3]

[1]Graduate School of Life and Environmental Science, Kyoto Prefectural University
1-5 Hangi-Cho Shimogamo, Sakyo-ku, Kyoto 606-8522, JAPAN
tabuse@kpu.ac.jp
[2]Department of Informatics and Environmental Science, Kyoto Prefectural University
1-5 Hangi-Cho Shimogamo, Sakyo-ku, Kyoto 606-8522, JAPAN
[3]Kyoto Prefectural Subaru High School
120 Mukaijima-Nishi-joke, Fushimi-ku, Kyoto 612-8156, JAPAN

Abstract: In this paper, we propose SURF feature based approach for outdoor autonomous navigation. In this approach, we capture environmental images by using omni-directional camera and extract features of these images by using SURF. We treat these features as landmarks and estimate robot self-location and direction of motion. SURF features are invariant under scale change and rotation and robust under image noise, change in light condition and change viewpoint. Therefore, SURF features are appropriate ones for robot self-location estimation and navigation.

Keywords: Mobile robot navigation, SURF feature, Robot self-location, Omni-directional camera

I. INTRODUCTION

Recently, autonomous mobile robots are actively researched. In order to work in the same environment as people, the mobile robot must posses the ability to move everywhere. The most basic function for the mobile robot is to be able to move to the destination autonomously. Therefore, the robot must estimate self-location and direction to the destination.

Many methods of autonomous navigation are proposed. Odometry is used in many mobile robots. However, it has the cumulative error in the movement. It is unsuitable to measures the movement of long distance. Therefore, odometry is used with other sensors, e.g., GPS, visual sensor, laser range finder, gyroscope, etc. GPS is a system that measures the position on the earth by using the space satellite. However, near a high building the position is not always measured accurately by GPS. Another navigation method is view based approach. In view based approach, many images are memorized and self-location estimation is performed by template matching. In general, image data is very large amount of information and calculating cost of template matching is large.

In this paper, we propose SURF feature [1] based approach. In this approach, we capture environmental images by using omni-directional camera and extract features of these images by using SURF. We treat these features as landmarks and estimate robot self-location and direction of motion. SURF features are invariant under scale change and rotation and robust under image noise, change in light condition and change view point. Furthermore, SURF calculation is several times faster than SIFT [2][3]. Therefore, SURF features are appropriate ones for robot self-location estimation and navigation, because a mobile robot moves around and environmental images are captured from different positions and directions.

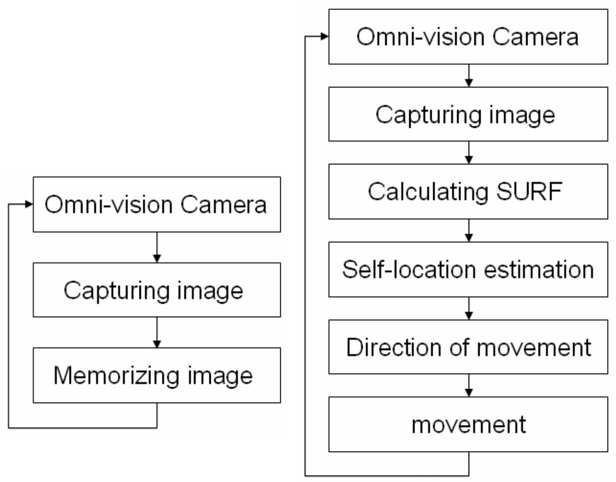
II. SYSTEM OVERVIEW

In our research, mobile robot navigation method consists of two mode; teaching mode and navigation mode. Fig.1 shows the flowchart of our method. In the teaching mode, the operator navigates the robot to a destination. The robot memorizes a sequence of environment using an omni-directional camera. In the navigation mode, the robot captures omni-directional environmental image, calculates SURF features in this image. Then the robot compares these features with SURF features of memorized images and estimates self-location and direction of movement.

III. Self-Location Estimation

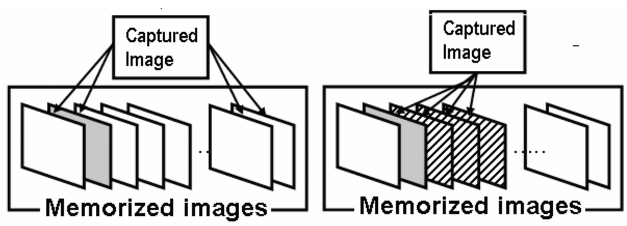
Firstly, we extract feature points in captured image and in memorized images using SURF. Then we find feature points in each memorized images matching to feature points in capture image. We estimate that a memorized image which has maximum number of matching feature points is neighbor of capture image.

Diagram of self-location estimation is shown in Fig.2. At a starting of navigation, robot doesn't know where it is. Hence, we find matching feature points in all memorized images and estimate self-location. After estimating the self-location, we find matching feature points in the image of self-location estimation and next 3 images.



(a) (b)

Fig.1. Flowchart of our method. (a) teaching mode. (b) navigation mode.



(a) (b)

Fig.2. Diagram of self-location estimation. (a) At first time, full searching. (b) After estimating the self-location, searching the image of self-location estimation and next 3 images.

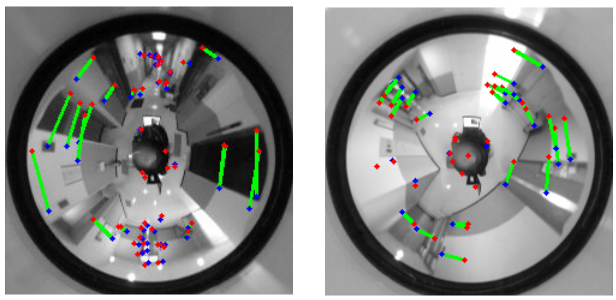
IV. Movement of Mobile Robot

In order to determine movement of mobile robot, we calculate angles of feature points in the neighbor memorized image used in self-location estimation to matching feature points in the next memorized image, and angles of feature points in capture image to matching feature point in the neighbor memorized image. Fig.3 shows feature points in an image and the matching feature points in the next image denoted by circles and movement of feature points denoted by lines. We divide omni-directional image in 4 regions, shown in Fig.4. In each region, we calculate average angle, (front angle θ_F, back angle θ_B, left angle θ_L, right angle θ_R), respectively. In the case of going forward shown in Fig.3 (a), θ_L and θ_R are large and similar angles. But, θ_F and θ_B are small. On the other hand, in the case of rotation shown in Fig.3 (b), θ_F, θ_B, θ_L and θ_R are similar. Therefore, we define movement angle Θ_M and rotation angle Θ_R as follows:

$$\Theta_M = \frac{\theta_R - \theta_L}{2} , \qquad (1)$$

$$\Theta_R = \frac{\theta_F + \theta_B}{2} . \qquad (2)$$

We determine movement of mobile robot using the movement angle Θ_M and the rotation angle Θ_R of the neighbor memorized image and the next memorized image and correct the direction of mobile robot using the rotation angle of the capture image and the neighbor image. For more detail, see the next section.



(a) (b)

Fig. 3. Movement of feature points. (a) going forward. (b) turning left.

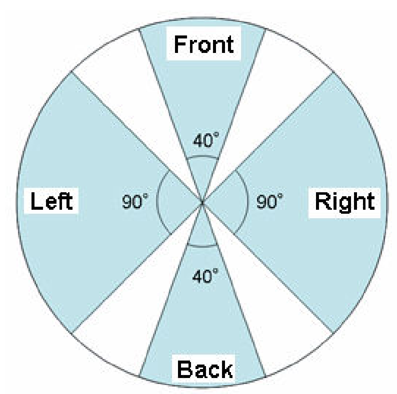


Fig. 4. 4 regions of omni-directional image.

V. Control System

In our experiments, we use an electric wheelchair as a mobile robot. The wheelchair is controlled by joystick controller. The joystick controller has two axes of movement and outputs 2 channel voltages (V_0, V_1). When we move the joystick forward (back), the joystick controller outputs $V_0 = 0$ [V] and $V_1 = 1.85$ [V] ($V_0 = 3.7$ [V] and $V_1 = 1.85$ [V]) and the wheelchair goes forward (back) at maximum speed, respectively. When we move the joystick left (right), the joystick controller outputs $V_0 = 1.85$ [V] and $V_1 = 0$ [V] ($V_0 = 1.85$ [V] and $V_1 = 3.7$ [V]) and the wheelchair turns left (right) at maximum speed, respectively.

Controlling the wheelchair by PC, we setup AD/DA converter in PC. We modify the wire lines connecting from the joystick controller to wheelchair main controller, so that 2 channel outputs of the joystick controller and of PC are switched with a switch (Fig. 5).

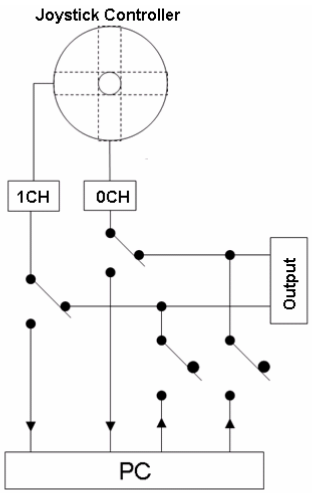


Fig. 5. Control system.

Fig. 6 shows 2 channel outputs while the wheelchair is moved by the joystick controller. In the area 1, two channel outputs (0CH, 1CH) are ($V_0 = 1.85$ [V] , $V_1 = 1.85$ [V]) and wheelchair is stopped state. In the area 2, two channel outputs are ($V_0 = 0$ [V] , $V_1 = 1.85$ [V]) and wheelchair goes forward at maximum speed. In the area 3, two channel outputs are ($V_0 = 0.25$ [V] , $V_1 = 3$ [V]) and wheelchair goes forward while turning right. In the area 4, two channel outputs are ($V_0 = 1.85$ [V], $V_1 = 0$ [V]) and wheelchair turns left.

Therefore, we define 2 channel outputs (V_0, V_1) as follows:

$$V_0 = f(1.85 - \alpha\Theta_M) \ , \qquad (3)$$

$$V_1 = f(1.85 + \beta(\Theta_R + \Delta\Theta_R)) \ , \qquad (4)$$

where Θ_M, Θ_R and $\Delta\Theta_R$ denote movement angle and rotation angle of feature points in the neighbor memorized image and the next memorized image memorized image, and rotation angles of feature points of capture image and the neighbor memorized image, respectively. $\alpha = 0.78$, $\beta = 0.1552$, determined by experiment, and

$$f(x) = \begin{cases} 0.0 & (x < 0.0) \\ x & (0.0 \le x \le 3.7) \\ 3.7 & (x > 3.7) \end{cases}$$

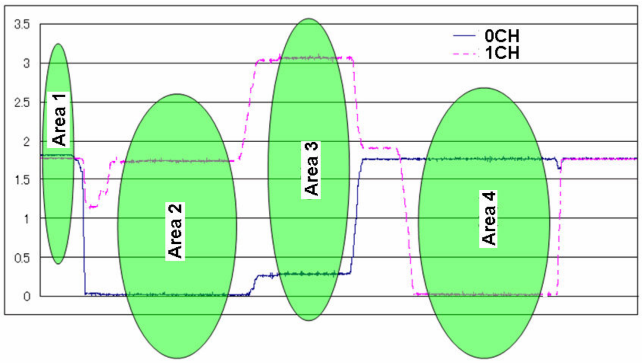


Fig. 6. 2 channel outputs.

VI. EXPERIMENT

1. Mobile Robot

A mobile robot is an electric wheelchair (SUZUKI MC2000). Omni-directional camera is setup at 150cm in height (Fig.7).

Hardware and software of our mobile robot are as follows:

Electric wheelchair : SUZUKI MC2000
Omni-directional camera :
Camera : SONY DCR-HC 88
Omni-directional mirror : Vstone VS-C42N-TR
Note PC : Dell Latitude E6400
(CPU : Core2Duo 2.66GHz, Memory : 2GB
OS : Windows XP)
AD/DA converter : CONTEC AD12-8(PM)
C++ Compiler : Microsoft Visual C++ 6.0
Computer Vision Library : OpenCV 1.1pre

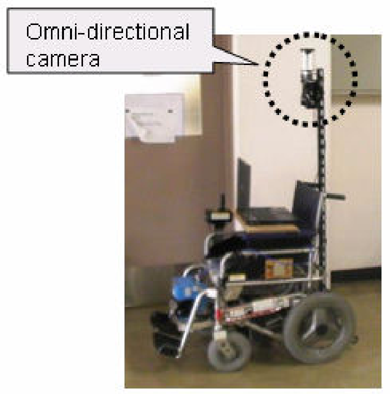


Fig. 7. Mobile robot

2. Teaching mode

In the teaching mode we control the wheelchair using the joystick controller and teach a navigation course. The course is about 150 m outdoor course on the campus of Kyoto Prefectural University, where the wheelchair goes straight from the entrance of the library to the corner, turns to the right and goes straight to the next corner, turns on the left and goes straight to the destination (Fig.8). We memorize 282 omni-directional images in this course.

Fig.8. Navigation course.

3. Navigation mode

In the navigation mode, the mobile robot is moved near the starting point and navigates to the destination autonomously. Fig. 9 shows several captured images in the navigation mode. Trajectories in the teaching and the navigation mode are shown in Fig. 10. The mobile robot navigates along the teaching course adjusting direction of motion. Maximum error of navigation trajectories is 3.3 m. Maximum speed is 4.5 km/h and average speed is 2.9 km/h.

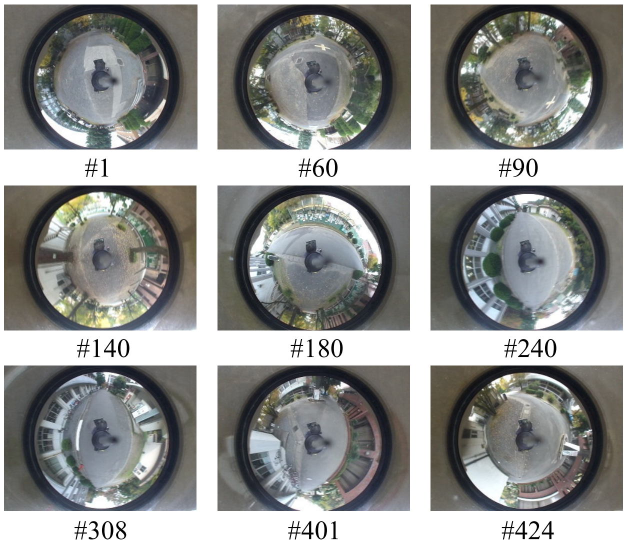


Fig. 9. Capture images in the navigation mode.

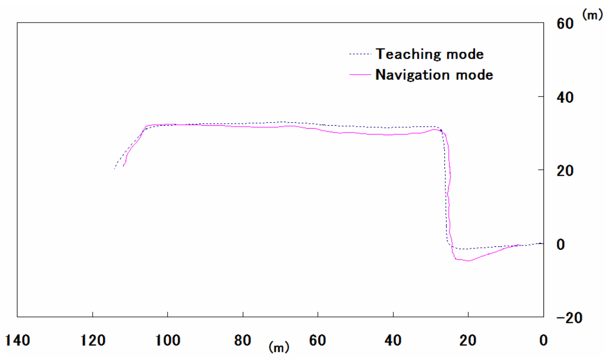


Fig.10. Trajectories in teaching mode and navigation mode.

VII. CONCLUSION

In this paper we present mobile robot navigation using SURF features. The processing time of SURF is several times shorter than that of SIFT. Therefore, the navigation speed of the mobile robot becomes near the person walking speed.

REFERENCES

[1] H. Bay, A. Ess, T. Tuytelaars, L. van Gool, Speeded-up Robust Features (SURF), Computer Vision and Image Understanding (CVIU), Vol. 110, No.3, 2008, pp. 346-359.
[2] D. G. Lowe, Object recognition from local scale-invariant features, Proc. of IEEE International Conference on Computer Vision (ICCV), 1999, pp.1150-1157.
[3] D. G. Lowe, Distinctive image features from scale-invariant keypoints, Proc. of International Journal of Co mputer Vision (IJCV), Vol. 60, No. 2, 2004, pp. 91-110.