The Thirteenth International Symposium on Artificial Life and Robotics 2008(AROB 13th '08),
B-Con Plaza, Beppu, Oita, Japan, January 31-February 2, 2008

Obstacle recognition system using monocular camera for an autonomous robot

Toshifumi Uewaki and Eiji Hayashi

Department of Mechanical Information Science and Technology
Faculty of Computer Science and Systems Engineering , Kyushu Institute of Technology
680-4, Kawazu, Iizuka-City, Fukuoka Prefecture, Japan

Abstract: We are developing an autonomous robot that recognizes its surrounding environment based on visual information obtained by a monocular camera. In order for the robot to move autonomously, it is necessary that it recognize obstacles and avoid them without collision. Therefore, we developed system that allows the robot to recognize obstacles through the use of image processing. In this paper, we explain in detail the image processing algorithm used in the obstacle recognition system, and we experimentally evaluate the system and discuss its problems based on the results.

Keywords: monocular camera, image-processing, obstacle recognition

I. INTRODUCTION

Industrial robots are extensively used in factories. Lately, however, robots intended for use by individuals, such as pet robots and cleaning robots, have been attracting attention. It is expected that robots that can perform work in human living environments such as the home or office will be developed in the near future. The capacity for autonomous movement is essential for such robots; they must be able to recognize the details of their surrounding environment. In light of this, we are developing an autonomous robot that recognizes its surrounding environment based on visual information obtained by a monocular camera. In this study, we developed system that allows the robot to recognize obstacles through the use of image processing. The robot's response varies according to the kind of obstacle detected, with the system recognizing a static obstacle and a moving obstacle by means of different image-processing systems.

II. System for Robot

The appearance of a robot developed in our laboratory is shown in Fig. 1 left. This robot has a drive mechanism utilizing two front wheels and one back wheel. The front wheels are attached to a motor that operates the wheels on either side independently, while the back wheel is a castor wheel. DC servo motors are used for the robot's drive mechanism, and position control and speed control are achieved by a control system governing the drive mechanism. In addition, the CCD camera used for environment recognition is installed on the head of the robot. All the devices are controlled with a PC inside the robot, and the robot's power is supplied by lead batteries.

Next, we explain the course determination program of the robot. The movement route to the destination is searched for with a "finite space map", and the robot moves along the route which is decided. A "finite space map" is a two-dimensional data map with information about the permanent environment of the robot (the size and position of arranged objects such as furniture, the size of the room, etc) written into it. The robot recognizes the environment by using the feedback values returned from the CCD camera and the motor encoders during movement. And the robot decides its course from the recognition results. We show the processing system configuration for course determination in Fig.1 right.

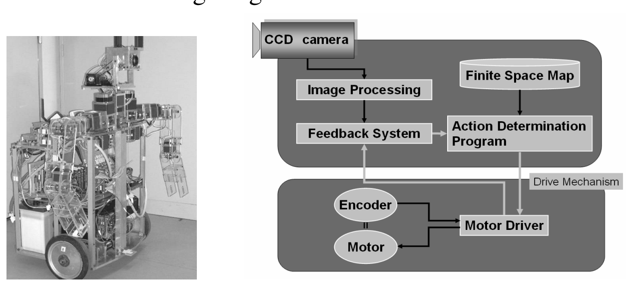

Fig.1 System for robot

III. Obstacle recognition system

1. Outline

This system assumes what is needed for the robot to recognize an obstacle during direct advance and avoid it.

Because what's necessary for this navigation is information to move a robot safely, the system does not recognize the detailed shape of the obstacle. The system only seeks to recognize the relevant information of the obstacle (position, width, height, and depth, etc). In addition, the position and the size of the detected obstacle are recorded in the finite space map.

2. Static obstacle recognition

In the recognition of static obstacles, the system extracts pixel groups whose shapes are much different from the main background pixels and interprets them as obstacles. The position and the width of the obstacle are estimated by extraction. In addition, the height and depth of the obstacle are estimated by using "motion stereo".

2.1 Method of extracting obstacle

First of all, the system converts 24-bit RGB image data into HSV data. HSV data include the image elements of hue, saturation, and value. The processing of the image data can be simplified by using HSV. The system samples a group of image pixels in a rectangular region at the bottom center of an image. The system uses the group of image data inside this region as its sample image data an then uses the deflection calculated by the sample data. The system extracts the floor region in terms of the difference of all pixels in the image. Fig. 2 shows an extracted obstacle.

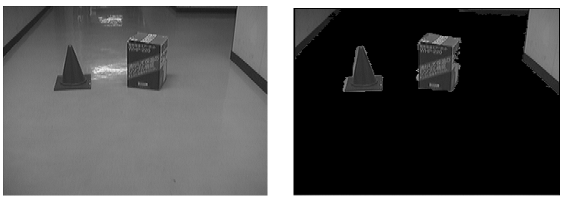

Fig.2 Extracted obstacle

Labeling is used to make sets of image pixels extracted from an image. This is processed from the extraction result of the floor region. The group of image pixels that leads to the detection of an obstacle is distinguished by this process. Fig.3 shows an example of labeling.

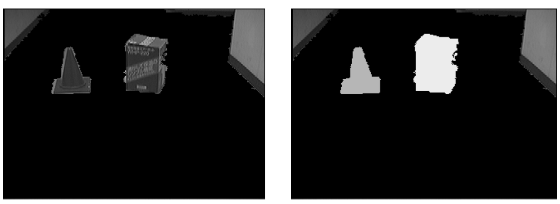

Fig.3 Example of labeling

2.2 Calculation of obstacle region

Next, the system analyzes the obstacle that has been determined to be an obstacle in terms of distance, width, height, and depth. We define the distance to the obstacle based on the lower side of a group of image pixels, and the system presumes the width of the obstacle based on the width of the group of image pixels. Next, we define (X, Y, Z) as the coordinate system of the three-dimensional space with its origin at the optical center and (x, y) as the image coordinate with its origin at the image center. We can describe the relations of both by a perspective projection model as follows.

$$x = f\frac{X}{Z} \qquad y = f\frac{Y}{Z} \qquad (1)$$

From this expression and geometric relations shown in Fig.4, we can express X and Z as follows.

$$X = \frac{xh}{f\sin\theta + y\cos\theta} \qquad Z = \frac{fh}{f\sin\theta + y\cos\theta} \qquad (2)$$

Because the camera angle θ, camera height h and focal length f are known, when the width of a group of image pixels is substituted for x of the expression (1), the width of the obstacle is found. Because Z_1 and Z_2 of Fig.4 are found from expression (2), d is calculated by geometric relations with them.

$$d = Z_1\cos\theta + \frac{Z_2}{\cos\theta} \qquad (3)$$

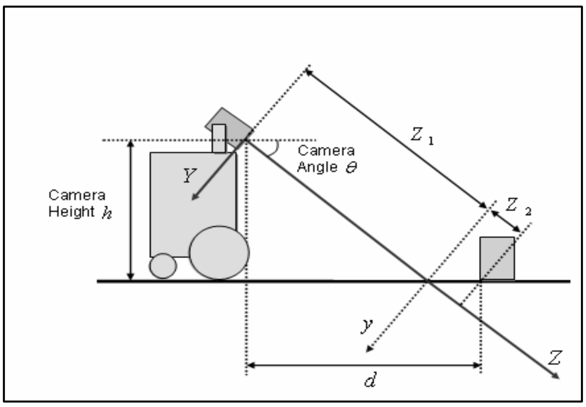

Fig.4 Geometric relation

We define the height and depth in terms of the uppermost part of a group of image pixels. The robot runs along the floor, and calculates the height and the depth by a motion stereo basis on data from the motor encoders and the position of the change in the image pixel group between the two images.

3. Moving obstacle recognition

In order for the robot to evade collision with the moving obstacle and to move safely, it is necessary that it learn motion information such as direction or speed in addition to position information. Therefore, the system is designed to detect moving obstacles with optical flow. First, a candidate moving obstacle region is roughly estimated using optical flow information. This region is then evaluated using its color information. The result recognized by this system is recorded on a map of the activity space given to the robot beforehand, as in the case of static obstacle recognition.

3.1 The candidate flow extraction of the moving obstacle

First, the system converts the RGB color image into grayscale image and smooths it. From the image, the system detects feature points (the corner and major contrasts in shade between adjacent pixels such as occur at edges). With KLT-Tracker, the system tracks a feature point between two pieces of images photographed at different times. Fig.5 shows the result of tracking feature points of images.

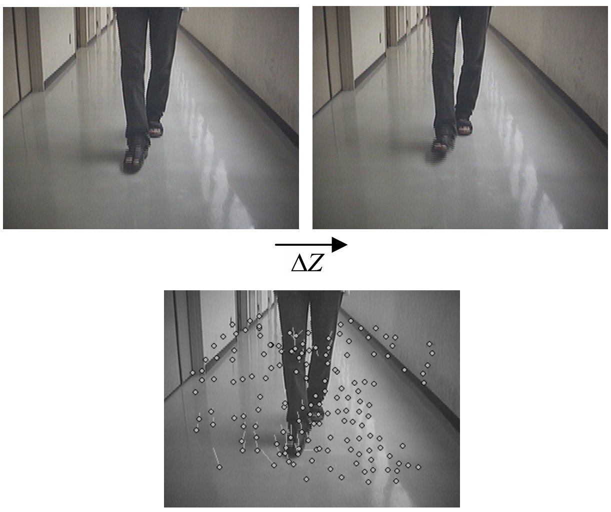

Fig.5 Tracking feature points of image

Next, from the tracking of feature points, the system makes a histogram of the direction and the size of the flow. As a characteristic of this histogram, almost any image will tend to have high frequency to a particular direction and size as in Fig. 6. Therefore, the system supposes frequency in direction and size to be the flow that occurs by the movement of the robot in time. We compare all the flows in the tracking image of feature points with this flow. If any of the results exceed the threshold, the system assumes the flow that of a candidate moving obstacle.

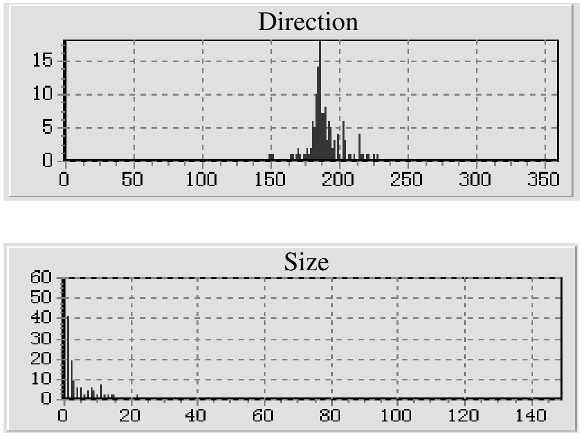

Fig.6 Histogram of direction and size

3.2 Moving obstacle extraction

The system estimates a moving obstacle region from a detection position, and the direction and size of the featured quantity of the candidate moving obstacle flow. In addition, a flow in the floor aspect may have a bad influence on region splitting when it is detected as a moving obstacle candidate flow. Therefore, using floor extraction information by static obstacle recognition, the system can remove it. We show the result in fig.7.

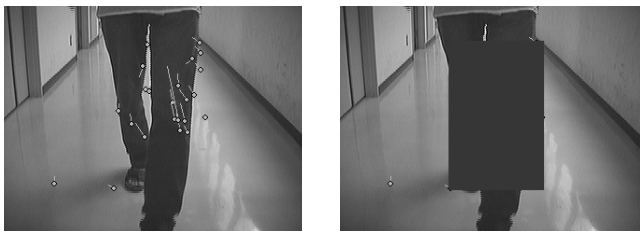

Fig.7 Extraction result

3.2 Obstacle position calculation

The position and width of the obstacle are calculated using an obstacle region extraction result. The calculati on method is the same as the distance and width calculat ion of the static obstacle recognition.

4. Evaluation of the obstacle extraction

We show the results of a test regarding the accuracy of the estimation of a static obstacle. When the system is actually used, processing will be repeated while the robot is moving. However, two images acquired to accurately measure the distance when the robot was in a fixed position were used to verify the accuracy of the data describing the obstacle (width, height, and depth).

Table.1 shows an example of the verified obstacle data. Based on these results, it was judged that the system collects obstacle data at a sufficient level of accuracy to evade an obstacle.

Table.1 Calculation result of object data (box)

	Actual measurement	Calculation result
Distance to obstacle	100cm	98cm
Obstacle position(horizontal direction)	Right 1cm	Right 3cm
Height	31cm	34cm
width	21cm	30cm
depth	21cm	17cm

5. Recording on the map of the obstacle

The robot evades an obstacle by using the results of obstacle estimation. In the case of a static obstacle, when the height of an obstacle is detected (if the height of the obstacle is 0, the system determines that the robot need not evade the obstacle), the obstacle's data (width and depth) is written into the finite space map. In the case of a moving obstacle, information on the depth of the obstacle is not sufficient to reflect it to a map. Because an obstacle moves, it is difficult to get the depth of the obstacle with the motion stereo which we used for static obstacle detection. Therefore, in the case of a moving obstacle, a temporary depth of the obstacle is set equal to its width. Fig. 8 shows an example of a recording on the map of a static obstacle.

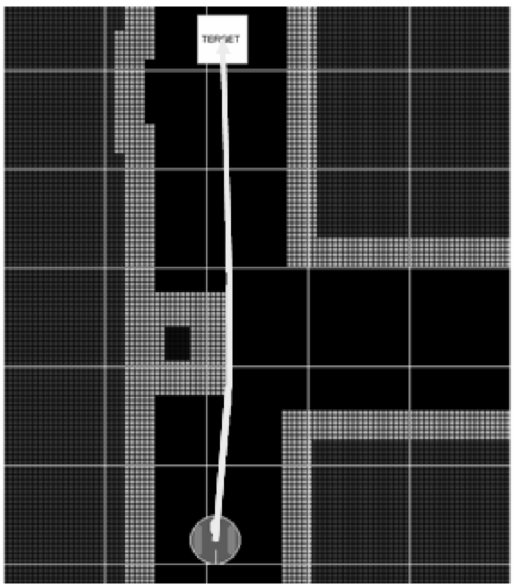

Fig.8 Recording on the map of a static obstacle

IV. Conclusion

By this report, we suggested system that allows the robot to recognize obstacle through the use of image processing. We confirmed that the robot could recognize information of the obstacle space where the static obstacle exists only or the moving obstacle exists only. A future problem is for the robot to determine a movement plan.

V. References

[1] Takuma Umeno, Eiji Hayashi, Navigation system fo r an autonomous robot using an ocellus camera in indoo r environment, proceedings of the 11th international sy mposium on artificial life and robotics, Vol1.11th, pp.22 9, Beppu, Oita, Japan, 2006

[2] Hiroshi Mizoguchi, Eiji Hayashi, Development of an obstacle recognition system for autonomous robots in indoor environment, proceedings of the 12th international symposium on artificial life and robotics, Vol1.12th, pp.348, Beppu, Oita, Japan, 2007