Obstacle Avoidance of Autonomous Mobile Robot using Stereo Vision Sensor

Masako Kumano Akihisa Ohya
Shin'ichi Yuta
Intelligent Robot Laboratory
University of Tsukuba, Ibaraki, 305-8573 Japan
E-mail: {masako, ohya, yuta}@roboken.esys.tsukuba.ac.jp

Abstract

The goal of this research is to develop a real time obstacle avoidance system for autonomous mobile robots using a stereo vision sensor.

At first, an obstacle detection method is proposed. It is based on stereo measurement without any search of the corresponding points to match them. This method is fast enough for a mobile robot which has poor capabilities to carry out respectable image processing. However there happened a problem that some ghost objects are detected. We'll describe the solution and an experimental result which shows the effectiveness of the improved method.

Then, a strategy of obstacle avoidance and an implementation of the proposed method to the mobile robot are described.

1 Introduction

When mobile robots move in a real environment, recognition of surrounding objects is a big subject. Visual information is widely used for navigation and obstacle detection of mobile robot[1-8].

In this research, we consider to use stereo vision sensor. But stereo image processing method usually demands huge amount of time. The most important point for real-time sensing is how to deal with this information effectively.

The purpose of this research is to realize an autonomous obstacle avoidance in an usual environment. To realize that, both a reliable sensor and a suitable path generate system are needed.

In this paper, at first we cover the obstacle detection method which processing is fast enough for the robot to find 3D objects at quick rate. We also talk about detection failure with this method and how to improve it with an experimental result. Next, the system of whole motion control contains route running and obstacle avoidance by planing suitable paths. Finally conclusion and future works are described.

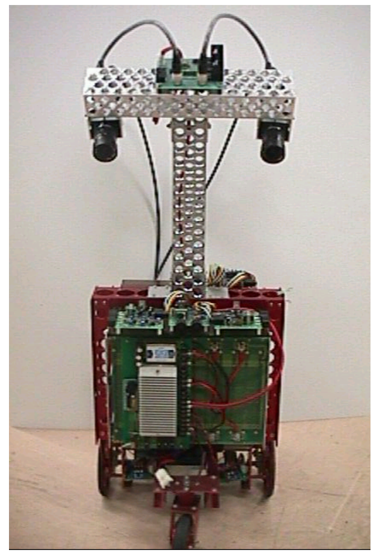

Figure 1: Mobile robot equipped with stereo vision sensor

2 Stereo Vision Sensor

The set of stereo vision sensors we use in this research is composed of two monochrome CCD cameras equipped with about 90 degrees wide-angle lenses, which are fixed on the left and right side with the same height at the top of the robot(See Figure 1). Two images are captured synchronously on an image processing board.

It always requires large mount of processing time to search corresponding points between right and left images for recognizing 3D objects. Some methods have been already proposed in order to obtain data of obstacle existence at fast rate[9][10]. Their principles are as follows: First, all objects which are taken in each image are supposed to be completely drawn on the floor. Second, one image is estimated from the other by the matrix calculated with relative position. Third, compare each point on the real image with the corresponding one on the estimated image. If there is

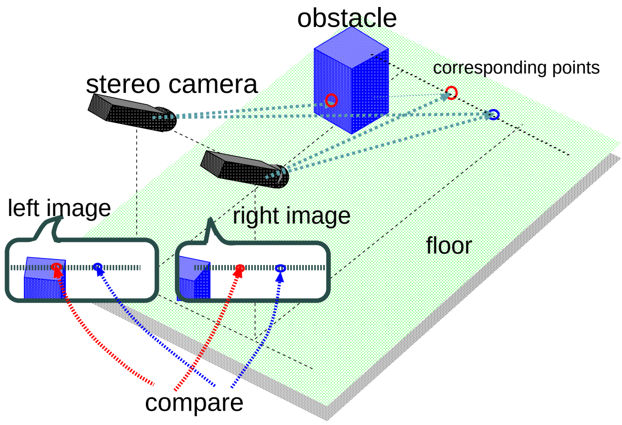

Figure 2: The principle of obstacle detection in this research. It makes real time obstacle detection possible without any search of the corresponding points on each images.If both right and left of the brightnesses of corresponding points are almost equal, there is not any obstacles there. The difference of brightnesses means there is something around the point.

a certain difference of brightness, around there any 3D objects are detected. These methods succeeded to shorten processing time.

But most of them as usual need considerable time. Then we suggested an appropriate method for obstacle detection as a real time processing of the visual information(See Figure 2). In this method we also take advantage of the idea of the previous method, but only for a small area in the images which is necessary to consider the future motion of the robot. Instead of calculating the estimate image, we need to prepare the table of corresponding points in advance. Using it and just comparing each brightness value at the same point, we can understand if there is something at the supposed position or not.

The detection needs only 35.2[msec], this number is an average of repeated 1000 times detections according to the following conditions: Using a table which includes 150 corresponding points location data; they are actually assumed to be on the floor, 40[cm], 65[cm] and 100[cm] in front of the robot in the line of 150[cm] length. Figure 4 shows those points' location. 35.2[msec] includes 33[msec] of capture time, so it shows how fast this process is executed. An object of 5[cm] width $\times$ 5[cm] height can be detected.

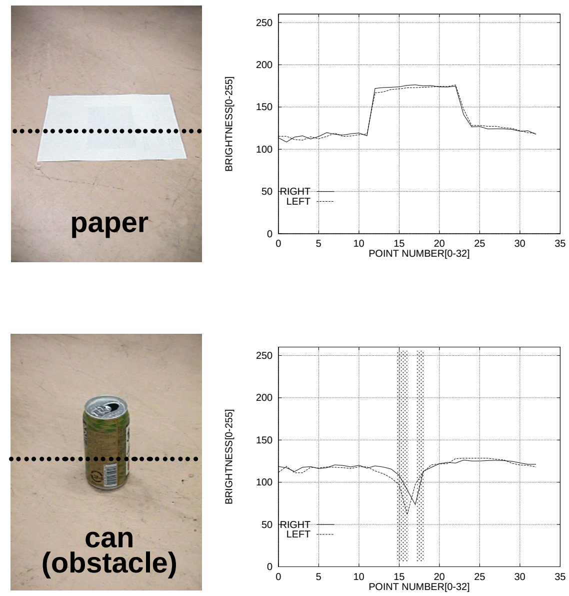

Figure 3: The sample results of a flat(paper) and 3D(can: obstacle) objects. Horizontal axis means corresponding points from left to right, vertical axis shows the brightness of each point, solid line is for right image and dotted one is for left. There are differences between right and left value at the point around both end of the can.

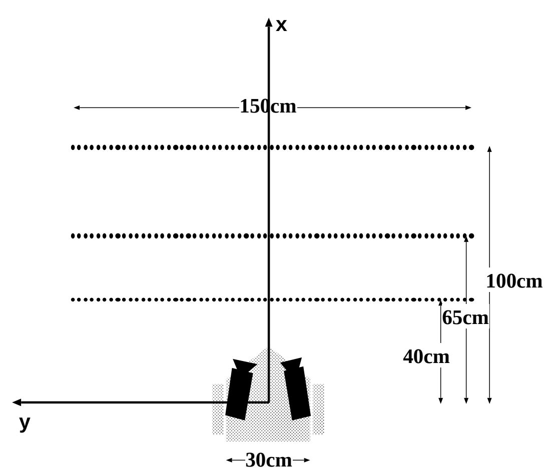

Figure 4: The relative location of corresponding points for stereo camera from the robot.

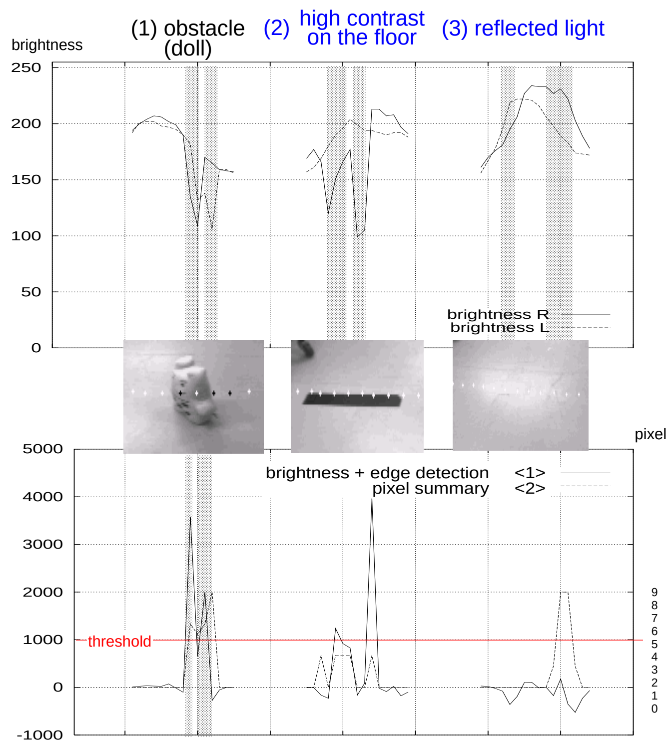

Figure 5: Experimental result dealing with obstacle(1), black tape on the floor(2) and reflected light(3), before (upper part) and after (lower part) improvement. Horizontal axis shows the location of the corresponding points on the floor. Upper graph's vertical axis displays brightnesses of those points and lower graph's one expresses the experimental result after improvement. Areas where obstacles are detected are marked by colored bars. In the upper graph, there are detection errors, in the lower one they disappeared thanks to improvement.

3 Improvement of Detection Method

However there are two detection errors that occur using this method. That means some ghost objects are occasionally detected.

One is caused by reflected lights (see the right picture of Figure 5), and the other is by high contrast on the floor (the middle one).

3.1 Reflected ceiling lights on the floor

First error is caused by reflected ceiling lights on the floor. The cause is the position where lights are reflected in the image depends on the location of the right and left cameras, it leads to a certain difference of brightness at the same corresponding points. It can seen at Figure 5 right hand upper graph(3), that there is nothing on the floor but sensor has made wrong judgment.(Colored bars mean that obstacles are existing around them.) However, there is a big difference between a reflected light and a real object, that is how distinct their edges are. Reflection of light is blurred, whereas a real object has a clear contrast from its surroundings. So it is possible to distinguish real objects from reflections.

Lower graph of Figure 5 has two types of lines. Result of this improvement for reflected lights is showed at solid line$\langle 1 \rangle$. It presents the value of each point that is affected by both its brightness and edge intensity – calculated by difference of the point's brightness and next point's one. The sharper the edge of objects around the point, the larger the value of the point becomes. Certain threshold has been set and some experiments have been made. The detection errors related to reflected lights have been canceled. Figure 5's right hand graphs show that this method works successfully.

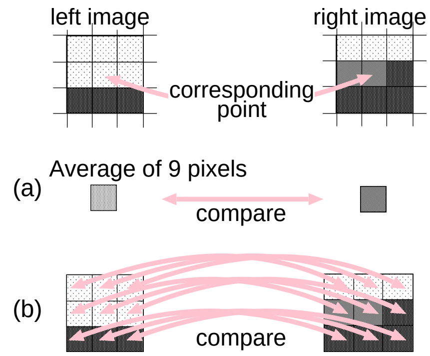

Figure 6: Improved method for high contrast on the floor

3.2 High contrast on the floor

Second error happens around the border on the floor between different colors which intensities are remarkably changed(See Figure 5 upper (2)). The reason we guess is the corresponding points on each right and left image which are previously calibrated using original process are not corresponding strictly. But exactness of calibration has a limit, and it is necessary to improve the object detection method with allowable margins of exactness.

Then we replace the detection algorithm by a more tolerant one. Previous method compares the averages of nine points' brightnesses shown in Figure 6(a). New one compares the brightness of the points shown in Figure 6(b). If a certain number of corresponding points have different brightness, we deduce the existence of an obstacle around these points. So dotted line$\langle 2 \rangle$ in Figure 5 lower graph, corresponds to integers from 0 to 9. Here threshold is set to 4. After this improvement, errors are significantly decreased.

When both $\langle 1 \rangle$ and $\langle 2 \rangle$ are greater than certain thresholds, real obstacle exists around the point. The judgment if there is an obstacle or not is the result of the logical and between the two comparisons presented above. Thus experimental results show that both implementations were successful. This can be seen on Figure 5 where the colored bars disappeared in the case of (2) and (3) after the new processing whereas they remained present in the case of a real obstacle.

4 Realization of Obstacle Avoidance Behavior

First we decided some preconditions for autonomous obstacle avoidance by mobile robot. Robot already knows the path to follow which is given by human, uses the information of path and obstacle existence to plan a suitable new path to the goal by itself.

Therefore the route description affects every decision, and in the beginning we need to define its details. What kind of motions are suitable for our robot equipped with stereo vision sensor on the top of it? Mention sensor information, its sensing area depends on the motion of robot itself and the area has limit, that means to realize totally safe movement requires to plan the paths only in well-known environment. Conversely some motions contain running on unknown area, therefore spin turn should be banned. For these reasons, each running path is described as a straight line or an arc. Whole route description which binds each paths smoothly becomes a target route to follow.

Next discussion is about how to deal with those paths conveniently. We adopted a state transition list, regarding a straight line or an arc as a statement. One state contains its *state number*, which *motion* (line or arc), *start and end positions* (x, y, θ), the *radius* of the circle (if state of arc) and the *number of next state*(Shown at Table 1). Using the above list and the preservation of current state, the robot can control its objective behavior.

While the robot executes general motion the list is handled in order. If any obstacles are found, suitable avoidance and recovery are demanded. So we consider that the list dynamically changes according to the detected point and can be performed suitably in order to accomplish both avoidance and returning. Concrete means are the following: 1) Plan new route which avoids the point where an obstacle has been detected. 2) After state preservation part recognizes that a new state was added at the end of the list,

Table 1: Original state transition list

	type of route	origin point x	origin point y	origin point θ	radius	termination x	termination y	termination θ	next state
0	straight	x_{b_0}	y_{b_0}	θ_{b_0}		x_{e_0}	y_{e_0}	θ_{e_0}	1
1	arc	x_{b_1}	y_{b_1}	θ_{b_1}	r_1	x_{e_1}	y_{e_1}	θ_{e_1}	2
2	arc	x_{b_2}	y_{b_2}	θ_{b_2}	r_2	x_{e_2}	y_{e_2}	θ_{e_2}	3
3	straight	x_{b_3}	y_{b_3}	θ_{b_3}		x_{e_3}	y_{e_3}	θ_{e_3}	4

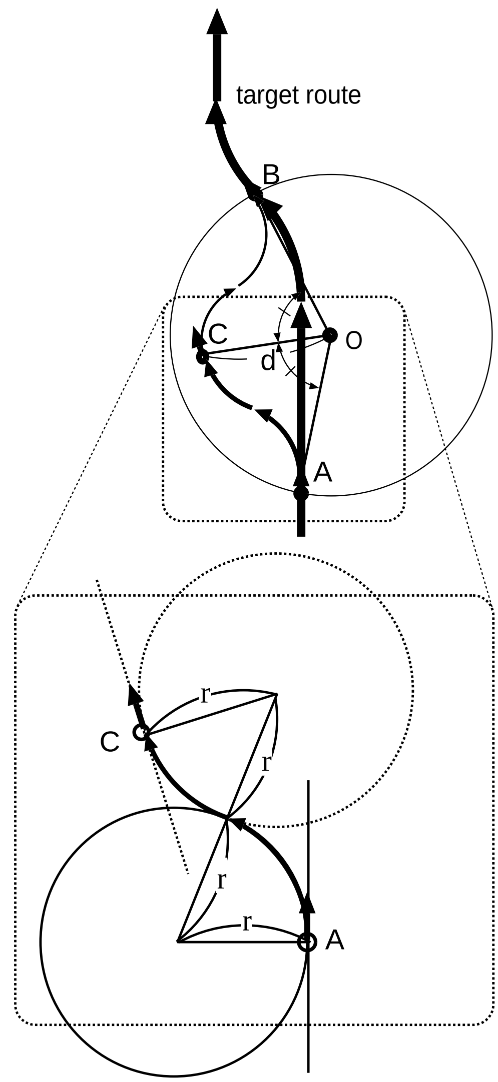

Figure 7: The method to construct avoidance paths. The position where an obstacle is detected (O), the current position of the robot (A) and the original route list (Table 1) are used. [B has to be calculated as the point where a robot recover to the original route, C is decided as a relay point between A and B. The paths are determined by those three points using some arc lines.]

Table 2: The list after generation of avoidance paths

	type of route	origin point x	origin point y	origin point θ	radius	termination x	termination y	termination θ	next state
0	straight	x_{b_0}	y_{b_0}	θ_{b_0}		x_{e_0}	y_{e_0}	θ_{e_0}	1
1	arc	x_{b_1}	y_{b_1}	θ_{b_1}	r_1	x_{e_1}	y_{e_1}	θ_{e_1}	22
2	arc	x_{b_2}	y_{b_2}	θ_{b_2}	r_2	x_{e_2}	y_{e_2}	θ_{e_2}	3
3	straight	x_{b_3}	y_{b_3}	θ_{b_3}		x_{e_3}	y_{e_3}	θ_{e_3}	4
:		:		:		:			
22	arc	$x_{b_{22}}$	$y_{b_{22}}$	$\theta_{b_{22}}$	r_{22}	$x_{e_{22}}$	$y_{e_{22}}$	$\theta_{e_{22}}$	23
23	arc	$x_{b_{23}}$	$y_{b_{23}}$	$\theta_{b_{23}}$	r_{23}	$x_{e_{23}}$	$y_{e_{23}}$	$y\theta_{e_{23}}$	3

it moves the pointer of current state to top of new state. 3) The last state's next state will be changed as the state which contains the recovering point. Using the list like this, just updating it enables the management motion of the robot as well as we expected.

5 Planing Avoidance Paths

When the obstacle is found, the path should be bended by replacing the original path with a new path to avoid it. The new path can be constructed using the detected obstacle position and current robot position(See Figure 7).

Next the points where robot should pass through to avoid an obstacle and recover to original route are demanded to be decided. The recovery point is defined as follows: Calculate two points where the line – according to the given list –, and the circle – its center is detected obstacle position and its radius is the length between detected obstacle position and current robot position – cross each other. The point which does not correspond to the current robot position is defined as the recovery position to the original route. Also the position to pass is the position where both half of the angle made by above mentioned three points and far certain distance from the detected position. Considering the size of robot, the distance is beforehand decided.

The same conditions mentioned in last section are also adopted to new route as avoidance ones, so every path should be described by a straight line or an arc. Here every path can be generated as two arcs when the location and orientation of two end points are given. For smooth motion of the robot the radius of each arc would like to be as large as possible. That means above two arcs designed the same timing will become equal radius.

Following those rules, a unique avoidance path can be constructed from the information of the detected obstacle position, current robot position and the original route list.

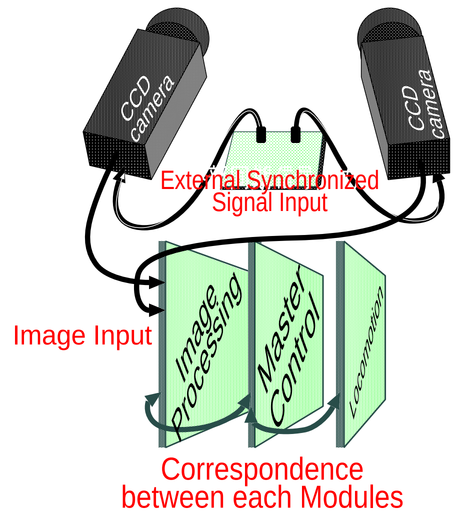

Figure 8: Stereo vision sensor system

6 Implementation

In this research we use the mobile robot "YAMABICO"[11] with a stereo vision sensor (Figure 1). Master control, locomotion control and image processing boards put on the rack of robot have each own CPU (Transputer T805 /20MHz) seen at Figure 8. We can control its motion easily thanks to original softwares that have already been developed in our laboratory. A high level command, for example, "Track the straight line which includes position(x,y) and its orientation is theta on the predefined coordinates system",can control the robot exactly.

The system we implemented is composed as follows: The above mentioned state transition list, the current state management process, the route runner process which control the robot's motion according to the list, the process generating new paths using the detection information and current robot position. Those work each other to aim at complete expected motions including both route running and obstacle avoidance.

7 Conclusion

In this paper we described high speed obstacle detection method and the way to decrease its detection errors. Then we presented the method of obstacle avoidance and its implementation. It already confirmed that robot autonomously runs according to given list, it proves that some of necessary functions were successfully implemented. The rest of implementation is the path generation

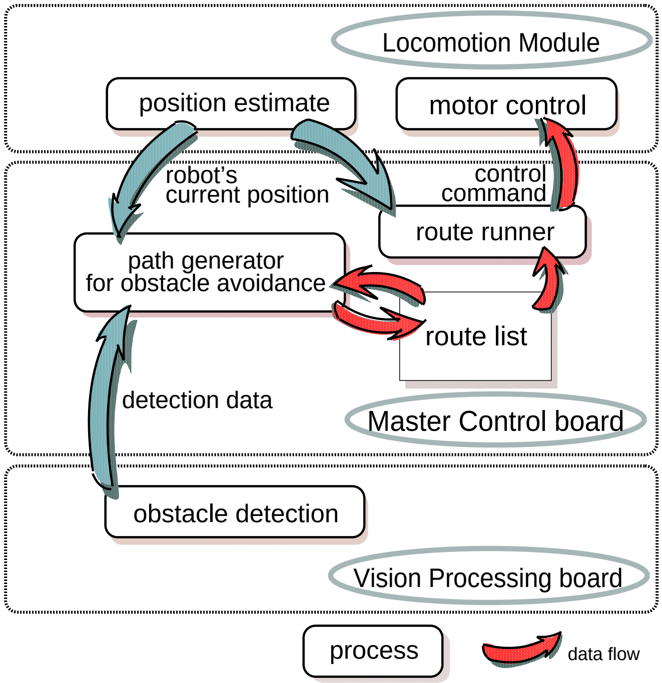

Figure 9: The sketch of motion control processes

processes.

Future task is to complete whole implementation and conduct some experiences in various environment with several obstacles to evaluate this system. Our final goal is real-time autonomous execution of given movement avoiding obstacles naturally.

Acknowledgement

The authors would like to thank Mr. Launay Fabien for the careful proofreading for preparing this manuscript. We also thank very much Mr. Tomoaki Yoshida of the University of Tsukuba for his useful suggestions and support of this research.

References

[1] T. Camus, D. Coombs, M. Herman and T. Hong, ‘Real-time single-workstation obstacle avoidance using only wide-field flow divergence,” in *Proc. 13th Int. Conf. Pattern Recognition*, Aug. 1996.

[2] A. Kosaka and A. Kak, ‘Fast vision-guided mobile robot navigation using model-based reasoning and prediction of uncertainties,” *CVGIP–Image Understanding*, vol. 56, no. 3, pp. 271-329, Nov. 1992.

[3] K. Sugihara, ‘Some location problems for robot navigation using a single camera,” *Computer Vision, Graphics and Image Processing*, vol. 42, no. 1, pp. 112-129, Apr. 1988.

[4] T. Tsubouchi and S. Yuta, ‘Map assisted vision system of mobile robots for reckoning in a building environment,” in *Proc. IEEE Int. Conf. Robotics Automat.*, Mar./Apr. 1987, pp. 1978-1984.

[5] Y. Matsumoto, M. Inaba and H. Inoue, ‘Visual navigation using view-sequenced route representation,” in *Proc. IEEE Int. Conf. Robotics Automat.*, Apr. 1996, pp. 83-88.

[6] T. Ohno, A. Ohya and S. Yuta, ‘Autonomous navigation for mobile robots referring pre-recorded image sequence,” in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots Syst.*, Nov. 1996, pp .672-679.

[7] A. J. Muñoz and J. González, ‘Localizing mobile robots with a single camera in structured environments,” in *Proc World Automation Congress –Robotic and Manufacturing Systems*, May. 1996, pp.539-544.

[8] D. J. Kriegman, E. Triendl and T. O. Binford, ‘Stereo vision and navigation in buildings for mobile robots,” *IEEE Trans. Robotics Automat.*, vol. 5, no. 6, pp. 792-803, Dec. 1989.

[9] Guo-Wei Zhao, Shin'ichi Yuta: "Obstacle Detection by Vision System for an Autonomous Vehicle", Intelligent Vehicles Symposium, pp.31-36, 1993

[10] K. Onoguchi, N. Takeda and M. Watanabe: "Planar Projection Stereopsis Method for Road Extraction", International Conference on Intelligent Robots and Systems, vol.1 pp.249-256, 1995

[11] S. Yuta, S. Suzuki and S. Iida, ‘Implementation of a small size experimental self-contained autonomous robot –sensors, vehicle control, and description of sensor based on behavior–,” R. Chatila et al. Eds, Experimental Robotics II, Springer-Verlag, 1993, pp. 344-358.