Safe Human-Robot-Cooperation: Image-based collision detection for Industrial Robots

Dirk M. Ebert and Dominik D. Henrich

Embedded Systems and Robotics Lab. (RESY)
Faculty of Informatics, University of Kaiserslautern, Germany
E-Mail: [debert | henrich]@informatik.uni-kl.de, http://resy.informatik.uni-kl.de/

Abstract

This paper analyzes the problem of sensor-based collision detection for an industrial robotic manipulator. A method to perform collision tests based on images taken from several stationary cameras in the work cell is presented. The collision test works entirely based on the images, and does not construct a representation of the Cartesian space. It is shown how to perform a collision test for all possible robot configurations using only a single set of images taken simultaneously.

1. Introduction

At present, the workspaces of robots are clearly separated from human workspaces. This is a result of the safety requirements prescribed by guidelines such as [ISO10218]. For future applications, it is necessary for humans and robots to cooperate safely in the same workspace. As the actual state of the environment is unknown, sensors are required. A sensor-controlled transfer motion from the current configuration into a target configuration is required as a basic skill to allow the robot to work safely with a human in the same workspace.

To achieve this goal, information on the actual state of the environment is required. Multiple cameras are convenient sensors as they are widely available and cost effective.

The problem is to extract a representation of the environment in the configuration space of the robot from the camera images. One approach would be to create a 3D model of the environment using methods like back-projection and then to use a robot model to determine the free and occupied configurations. However, free configurations in the neighborhood of the robot are false classified as occupied as it will be shown in section 2.

In this paper, we present an approach to directly infer from the scene images the state of the entire configuration space. A collision test is used as the standard interface to connect to the path planner. The path planner requests the state of a configuration, after which the collision test returns either *free* if the robot can move safely into that configuration or *occupied* if a collision would occur. However, only the queried configuration is tested. It is not guaranteed that there is a possible path from the current configuration to the queried one. The presented collision test can determine the state of all possible robot configurations and not just the neighboring configurations of the current one, while using only a single set of images taken simultaneously.

The problem of finding a free path for robots has been discussed widely and many solutions are available. For example, in [Gupta98] and [Noborio99], different approaches for path planning in the configuration space are presented.

In the past, many approaches have been discussed for sensor-based collision avoidance. However, most of them work with sensors that provide only local information. For example, in [Novak92] and [Feddema94] capacitance sensors were used as sensor skin. In [Lumelsky93], algorithms for whole-arm collision avoidance for robots with sensor skins were presented. In [Yu99], a wrist-mounted laser scanner was used. With only local sensor information available, only configurations close to the current robot configuration can be examined.

[Noborio01] presented an approach for image-based path planning in configuration space; however, the use of a wrist-mounted sensor is assumed and the approach required that the image of the scene in the target configuration to be known.

A general method for inferring the physical extent of objects from multiple images is the back-projection. It is widely used in Computer Graphics, for example in [Eckert00] and [Eisert00]. However, the focus here was on the precise reconstruction of the object in 3D space, including the texture information. This is not necessary for our problem, as the object can be coarsely reconstructed and there is no need to process the object texture. Also, only single objects were reconstructed, while in our problem multiple obstacles exist in the scene.

In [Noborio92], back-projection was used for robotics and multiple objects were considered. However, color cameras were required and only objects with sufficiently different colors can be separated.

An approach for performing image-based collision tests is the back-projection into configuration space, as presented in [Ebert01]. However, this approach reconstructs the whole configuration space for each collision test. Since common path planners do not test all configurations, this approach wastes computation time. Moreover, the memory requirements can become very large.

The rest of the paper is organized as follows: In Section 2, we present the investigated problem. In Section 3 we present our approach, while Section 4 details the experimental results.

2. Problem Analysis

We assume that the work-cell of the robot is observed by several stationary cameras and that the robot is completely captured in all camera images in all configurations. It is further assumed that the camera images are transformed into binary *scene images* that represent the difference between the current state and the empty state of the work-cell. If the robot cannot be removed from the cell, a method to create the artificial reference state can be used, as presented in [Ebert01]. It is also assumed that a geometric robot model exists, which can generate images of the robot in all required configurations. These images are called *robot images*. An image that shows only the obstacles in the scene without the robot is called *obstacle image*. It cannot be acquired directly and has to be computed using scene and robot images.

The basic idea is to construct an obstacle image for each camera view and to intersect these images with the robot images containing the robot in the test configuration. The idea behind this is an implicit reconstruction using the multiple camera images. If an explicit reconstruction is used, the approach would be as follows: First, the scene images are used to construct a 3D space representation. Then, the robot model would be used to generate a 3D representation of the robot.

The difficulty is that the reconstruction using the back-projection can be affected by both obstacle enlargement and phantom obstacles. This is exemplified in Figure 1. The real obstacles are represented by solid rectangles. If this scene is observed by two cameras, the hatched areas are the enlarged obstacles reconstructed from the camera images and the light crosshatched area is a phantom obstacle. If the third camera (hatched) is used in addition, the phantom obstacle disappears and the obstacle enlargement is represented by the dark crosshatched areas. [Niem97] presented analyses of such obstacle enlargement errors while reconstructing a single obstacle.

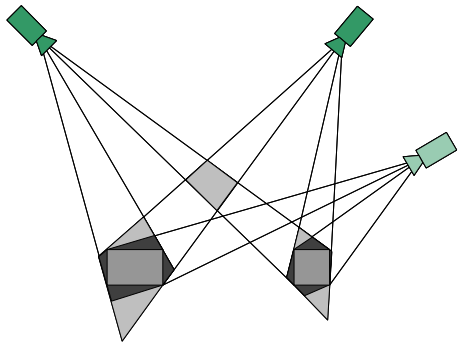

Figure 1: *Phantom obstacles and obstacle enlargement. [Ebert01]*

Thus, if just the real robot is removed from the space, the space around the current position would still be blocked due to the enlargement. This would lead to the failing of the path-planner. Therefore, the model has to take projection errors into account.

The quality of the reconstructed 3D space depends only on the number and position of the cameras. The 3D space contains no more information than all the camera images together with the camera location (if known). Therefore, there is no advantage in performing an explicit reconstruction of the 3D space. Furthermore, an approach that works directly with the images and camera information can achieve much better performance at the same quality. In the following, we present our approach for image-based collision detection.

3. Image-Based Collision Detection

It is assumed that the main program, for example a path planner, provides a configuration to be tested as parameter and expects a Boolean result indicating whether the configuration is free or occupied.

The general data flow is presented in Figure 2. The corresponding pixels of the current scene image and robot image are used to create the obstacle images using the mapping function presented in Table 1. The resulting obstacle images are combined with the test robot images to obtain the *intersection images* using the mapping shown in Table 2. The unknown pixels of these images are resolved, if possible, resulting in the collision images using the mapping shown in Table 3. The value of the pixels in all collision images is used to determine the Boolean result of the collision test.

All components are presented in detail in the following.

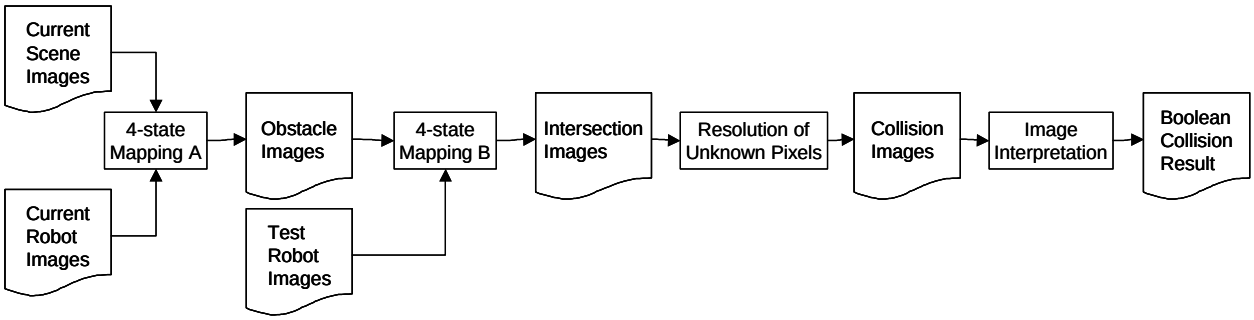

Figure 2: *Data flow within the collision test*

3.1. Obstacle Image Creation

The scene images provided by the sensors are binary difference images between the current and a reference image. Thus, in the image there is no difference between the robot and a (moving) obstacle. However, for the collision detection it is necessary to know which pixel belongs to the robot and which one belongs to the obstacle.

Table 1: *Mapping function to construct the obstacle images*

Scene Image	Robot Image	Obstacle Image
B(ackground)	B(ackground)	B(ackground)
B(ackground)	R(obot)	E(rror)
F(oreground)	B(ackground)	O(bstacle)
F(oreground)	R(obot)	U(nknown)

The binary pixels located at the same position in scene and robot image are used to determine one of the four values of the pixel in the obstacle image using the mapping presented in Table 1. If scene image pixel and robot image pixel are set to background, the corresponding pixel in the obstacle image is also background. If the scene image pixel is foreground and the robot image pixel is background, the object seen in this pixel is an obstacle. If the scene image pixel is foreground and the robot covers this pixel, the obstacle image pixel is set to unknown, as there might be an obstacle in front of or behind the robot. If the scene image pixel is background but the robot should cover this pixel, the obstacle image pixel is set to indicate an error because we see the background in a place where the robot should be. This error could occur for the following reasons: The robot image is incorrect because the robot model was wrong or the robot was not at the estimated position, or because there were problems with the sensor data. For example, an object in front of the robot with the same characteristics as the background might lead to this error.

The described process is illustrated for one camera in the following example in Figure 3. The left part represents the current scene image, while the middle part is the robot image generated by the robot model.

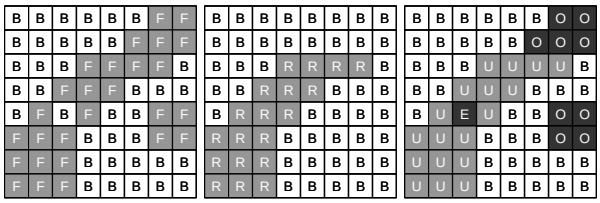

Figure 3: *Example of a scene image (left,) a robot image (middle) and resulting obstacle image (right)*

Note that the scene image contains a background pixel where the robot should be seen. Using the presented mapping, the obstacle image as shown the right part is obtained. If the sensor data already allows for distinction between the robot and obstacles, the above process is not necessary and the obstacle image can be directly retrieved from the sensor data without the help of the robot model. However, even in that case it cannot be assumed that no obstacles are located in places where unknown pixels are seen. Although the sensor data ensures there is no obstacle in front of the robot, there might still be one behind the robot.

Table 2: *Mapping function to construct the intersection images*

Obstacle Image	Test Robot Image	Intersection Image
B(ackground)	B(ackground)	B(ackground)
B(ackground)	R(obot)	B(ackground)
U(nknown)	B(ackground)	B(ackground)
U(nknown)	R(obot)	U(nknown)
O(bstacle)	B(ackground)	B(ackground)
O(bstacle)	R(obot)	O(bstacle)
E(rror)	B(ackground)	B(ackground)
E(rror)	R(obot)	E(rror)

3.2. Intersection Image Creation

In this step, it is determined which pixels in each camera could possibly reflect collisions. The test robot images show the robot in the configuration that is to be

tested. The obstacle image is masked using the test robot image.

Each corresponding pixel of obstacle image and test robot image is mapped using the mapping function presented in Table 2. The resulting intersection image pixel has the value of the obstacle image pixel if the test robot image pixel is robot, otherwise it is set to background.

For illustrations, the example in Figure 3 is continued in Figure 4.

B	B	B	B	B	B	B	B
B	B	B	B	B	B	B	B
B	B	B	B	B	B	B	B
B	B	R	R	R	B	B	B
B	R	R	R	R	R	B	B
R	R	R	B	B	R	R	B
R	R	R	B	B	B	B	B
R	R	R	B	B	B	B	B

B	B	B	B	B	B	B	B
B	B	B	B	B	B	B	B
B	B	B	B	B	B	B	B
B	B	U	U	U	B	B	B
B	U	E	U	B	B	B	B
U	U	U	B	B	B	O	B
U	U	U	B	B	B	B	B
U	U	U	B	B	B	B	B

Figure 4: *Test robot image (left) and resulting intersection image (right)*

3.3. Resolution of Unknown and Error Pixels

If there are unknown or error pixels in an intersection image, they have to be mapped to background or obstacle. This can be done using different techniques for both calibrated and uncalibrated camera systems; such methods are presented in the following.

If the camera system is not calibrated, there is no way to exactly match images from one camera with images from another camera. In this case, the unknown and error pixels cannot be resolved and are set to background. To ensure that there is no collision, the threshold (see Section 3.4) has to be adjusted.

If the camera system is calibrated, the error and unknown pixels of one camera can be resolved using images from other cameras. To do this, the corresponding epipolar lines in all other cameras are analyzed for each unknown or error pixel in the intersection image of one camera. The value of the pixel undergoing examination is set according to the mapping function in Table 3.

A pixel in the collision image is set to background only if none of the epipolar lines contains an obstacle pixel or if there is one epipolar line containing only background pixels. For all other cases, the pixel is set to obstacle. The idea is similar to the approach presented in [Matusik00], whereby the epipolar lines in different views were used to create an image of an object as seen from a certain view without the need to construct a 3D representation of the object.

Table 3: *Resolution of Unknown and Error Pixels*

Condition	New pixel value
One epipolar line containing only background pixels	Background
All epipolar lines containing only background or unknown pixels	Background
All other cases	Obstacle

The obstacle image in Figure 5 shows the same scene as Figure 3 and Figure 4, but from a different viewpoint. The hatched line is the epipolar line of the error pixel. If this is the only other view, the error pixel is set to background, as there are only background and unknown pixels on the epipolar line.

B	B	B	B	B	B	B	B
B	B	B	B	B	B	B	B
B	B	U	U	U	B	B	B
B	B	U	U	U	B	B	B
B	B	U	U	U	U	B	B
B	B	B	B	U	U	B	B
B	O	B	B	B	U	B	O
O	O	B	B	B	B	B	O

Figure 5: *Obstacle image with epipolar line*

3.4. Interpretation of Collision Images

The number of collision images containing obstacle pixels is determined. If that number exceeds a threshold, the tested configuration is considered occupied, otherwise, it is considered free. The threshold should reflect the number of views from which the obstacles might be occluded by the robot.

In our experiments, we assumed that an obstacle can be occluded in the view of only one camera. Therefore, the threshold was set to the number of cameras minus one.

4. Experimental Results

To measure the quality of the image-based collision test, the complete configuration space was tested; the following experiments were performed in simulations and with a real robot.

4.1. Simulation Experiments

Different approaches were compared in a simulated environment. Both the robot and the obstacles were represented by a number of spheres. Each link of the robot was approximated by five overlapping, equidistant spheres. The cameras were simulated by planes. The spheres were projected onto the planes using a parallel projection. The scenario is shown Figure 6. For a given scenario, the free space was computed using the examined collision test.

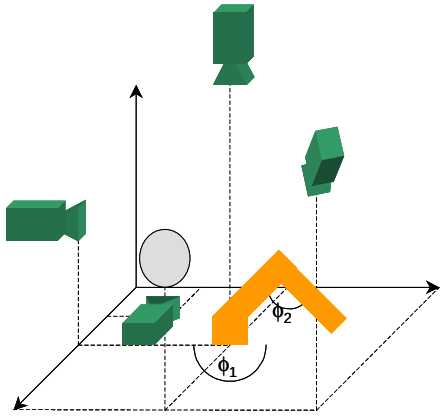

Figure 6: *Setup for simulation experiments. Four cameras survey the space from above, left, front and top right. The obstacle is the gray sphere located in the upper left quadrant. The robot can move the two joints f_1 and f_2 freely between 0° and 360°. Internal collisions of the robot are not considered.*

For comparison, we used an ideal collision test performed in true 3D space without using cameras. The test used the spheres to compute distances and was not affected by discretization errors resulting from camera pixels. The ideal collision test was used as a reference to compare with other collision tests.

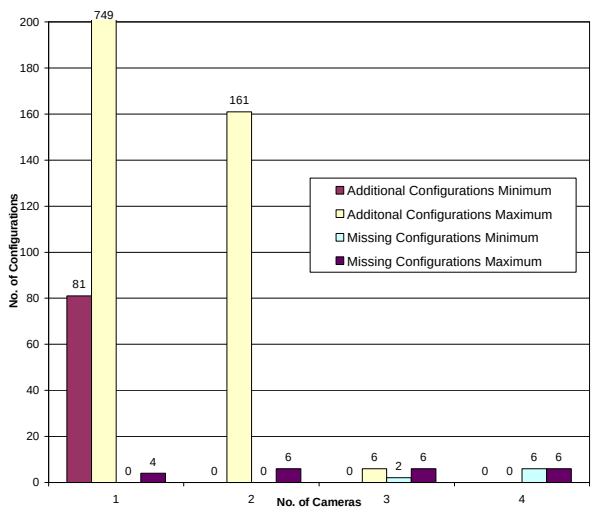

Figure 7: *Number of additional and missing configurations*

An *additional configuration* is a configuration that is reported free by the ideal collision test, but reported occupied by the collision test under examination. A *missing configuration* is a configuration that is reported occupied by the ideal collision test, but reported free by the collision test under examination In Figure 7, we see the number of additionally colliding and missing configurations reported by the collision tests in relation to the number of available cameras. For each number of cameras, all possible combinations of the four cameras are considered. The threshold was set to the number of cameras minus one.

Missing configurations can result from discretization errors during the synthetic generation of the difference images or from occlusion of the obstacle by the robot in more than one camera. Missing configurations in this experiment result mainly from discretization errors as the number of missing configurations does not depend on the number of used cameras.

4.2. Robot Experiments

As sensors, we used the grayscale CCD camera DMK 73/C connected to a DFG/BW1 Frame grabber. In our system, two frame grabbers were installed in one AMD-Athlon 1.2GHz PC, with two cameras attached to each frame grabber. The Stäubli RX130 robot manipulator was controlled by an Adept CS7 robot controller.

The images from all four cameras were acquired simultaneously. The discretization of the images was 64×64 pixels. The robot was moved successively into all configurations. In each configuration, all cameras took an image of the robot. These images were stored in a database and were used as the robot model. The experiments were only performed using the collision test for uncalibrated cameras. A weave-propagation algorithm as presented in [Latombe96] was used as path-planner. The configuration space was discretized according to the *MaxMove-Method* presented in [Henrich98].

The images from all cameras and the resulting scene images are shown in Figure 9. The black obstacle visible is blocking the path of the robot. The resulting configuration space is seen in Figure 8. The gray line shows a planned path in a 2D configuration space. The robot could move only in the axis 1 and 3, while the other axis were fixed.

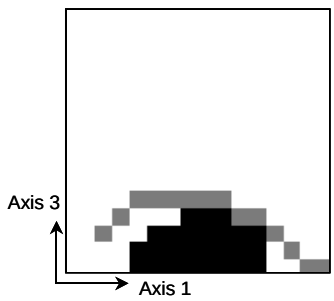

Figure 8: *Calculated configuration space in robot experiment*

Although the scene images are cluttered with many error pixels, the object reconstruction is good enough to allow for successful and safe path-planning.

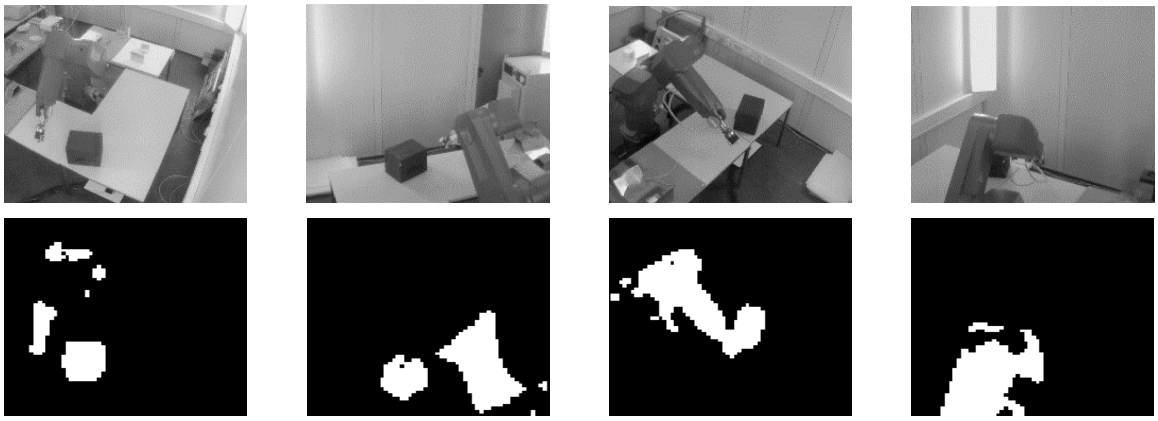

Figure 9: *Experiments with a set of camera images (top row) and corresponding scene images (bottom row)*

5. Conclusions

We presented a method to perform collision tests based directly on difference images. One problem that arose is that the robot itself appears in the sensor data, where it could possibly occlude obstacles. To remove this effect, the projection errors have to be taken into account. The presented collision test does this implicitly without the need of an explicit representation of the objects in the workspace. Generally, the collision tests succeeds in deciding whether a requested configuration is free or occupied, however there exist some configurations that are actually occupied but reported as free by the collision tests due to discretization errors during the synthetic generation of the difference images.

Future improvements should address the problem of resolving the unknown pixels in the intersection image. The approach using the epipolar lines is going to be evaluated in simulation and robot experiment. Also, the performance of the collision tests have to be evaluated in the presence of multiple objects in the work space. The number of missing configurations should be minimized by an optimum positioning of the cameras. To achieve this, further research in the semi-automatic optimum camera positioning is necessary.

References

[Ebert01] Ebert D., Henrich D.: "Safe Human-Robot-Cooperation: Problem Analysis, System Concept and Fast Sensor Fusion". In: IEEE Conference on Multisensor Fusion and Integration for Intelligent Systems, Baden-Baden, Germany, August 20 - 22, 2001.

[Eckert00] Gerald E.: “Automatic Shape Reconstruction of Rigid 3-D Objects from Multiple Calibrated Images”, In: Eusipco 2000 Proceedings, Tampere, Finland , 2000.

[Feddema94] Feddema J.T., Novak J.L.: “Whole Arm Obstacle Avoidance for Teleoperated Robots”. In: IEEE Robotics and Automation Proceedings, pp.3303 – 3309, 1994.

[Gupta98] Gupta K., Angel del Pobil (ed.): “Practical Motion Planning in Robotics; Current Approaches and Future Directions”. John Wiley & Sons Ltd, Chichester, England, 1998.

[Henrich98] Henrich D., Wurll Ch., Wörn H.: “On-line path planning with optimal C-space discretisation”. In: IEEE/RSJ Int. Conference on Intelligent Robots and Systems (IROS’98), Victoria, Canada, Oct. 12-16, 1998.

[ISO10218] ISO 10218, EN 775: ”Manipulating industrial robots – Safety”. (ISO 10218 modified), 1992.

[Latombe96] Latombe J-C.: ”Robot motion planning”. 4th print.; Boston : Kluwer Acad. Publ., 1996.

[Lumelsky93] Lumelsky V., Cheung E.: “Real-Time Collision Avoidance in Teleoperated Whole-Sensitive Robot Arm Manipulators”. In: IEEE Transactions on Systems, Man and Cybernetics, Vol.23 No.1, pp.194-203,1993.

[Matusik00] Matusik W., Buehler C., Raskar R., McMillan L. and Gortler S.: "Image-Based Visual Hulls.". In Proceedings of SIGGRAPH 2000.

[Niem97] Niem W.: "Error Analysis for Silhouette-Based 3D Shape Estimation from Multiple Views". In: Proc. on Int. Workshop on Synthetic - Natural Hybrid Coding and Three Dimensional Imaging, Rhodos, September 1997.

[Norobio99] Noborio H., Maeda Y., Urakawa K.: ”Three or More Dimensional Sensor-Based Path-Planning Algorithm HD-I“. In: IEEE International Conference on Intelligent Robots and Systems, pp.1699-1706, 1999.

[Noborio01] Noborio H., Nishino Y.: “Image-based Path-Planning Algorithm on the Joint Space”. In: IEEE International Conference on Robotics and Automation, pp. 1180-1187,Seoul, 2001.

[Novak92] Novak J.L., Feddema J.T.: ”A Capacitance-Based Proximity Sensor for Whole Arm Obstacle Avoidance”. In: IEEE Proceedings of the Intl. Conf. on Robotics and Automation, pp. 1307-1314, 1992.

[Yu99] Yu Y., Gupta K.: „Sensor-Based Roadmaps for Motion-Planning for Articulated Robots in Unknown Environment: Some Experiments with an Eye-in-hand System“. In: IEEE International Conference on Intelligent Robots and Systems, pp.1707-1714, 1999.