In: 37th International Symposium on Robotics (ISR 2006) / 4th German Conference on Robotics (Robotik 2006) München, Germany May 15 to 17, 2006

Multi-Camera Collision Detection allowing for Object Occlusions

Thorsten Gecks
University of Bayreuth
Chair for Applied Computer Science III
(Robotics and Embedded Systems)
Germany

Dominik Henrich
University of Bayreuth
Chair for Applied Computer Science III
(Robotics and Embedded Systems)
Germany

Speaker: Prof. Dr. Dominik Henrich, University of Bayreuth, Chair for Applied Computer Science III (Robotics and Embedded Systems), D-95440 Bayreuth, Germany, dominik.henrich@uni-bayreuth.de

Topic: Session B1: Safety for assistive robots

Keywords: Collision Detection, Path Planning, Difference Image, Multi-Camera Fusion

Abstract:

A multi-camera-based collision detection system is presented. We describe the computation of global collision information for the entire surveilled workspace based on local collision information extracted from camera images. If there are known occlusions (e.g., by the robot), the system is able to recover object collision information by fusing multiple camera images. The algorithm presented is part of the safety system of a prototype realizing human-robot-cooperation.

Introduction

Humans need reliable and safe interaction with robots in order to increase the level of acceptance and trust of the robotic partner at work or even at home. Our research aims to provide secure interaction between humans and robots to prevent severe injuries to humans and (costly) collisions with environment obstacles.

To safeguard robot operation, a secure system needs to detect dynamic obstacles such as humans and determine appropriate measures to avoid collisions with the obstacles detected (i.e., path planning), all while still being capable of contacting objects to carry out useful capabilities, such as pick-and-place operations or guided motions [2]. To achieve this goal, the acquisition of the current state of the environment is necessary. Cameras are convenient sensors for this task, as they are widely available and cost-efficient with respect to various criteria such as resolution and update rate. A comparison with other sensor hardware can be found in [5].

State of the Art

In the past, many approaches for sensor-based collision avoidance have been discussed. For example, in [12] and [7] capacitance sensors are used as an artificial skin. In [9], algorithms for whole-arm collision avoidance for robots with artificial skins are presented. In [16], a wrist-mounted laser scanner is used. Reference [11] presents an approach for image-based path planning in configuration space. However, the use of a wrist-mounted camera is assumed and the approach requires image information about the environment in the target configuration. The approaches mentioned only provide sensor information about obstacles in close proximity to the current robot configuration, thus only local path planning is possible. Furthermore, these sensors fail to provide the robot with safe object transportation capabilities. For gripped objects exceeding a certain size, it is nearly impossible to measure distances from the object to the environment, because it is impractical to place short-range distance sensors on the object. On the other hand, if sensors are placed on the robot's body, the transported object could obstruct the line of sight from the sensor to the environment. This could result in a collision between the transported object and the environment, as not all distances from the object to the environment are known. Sensors capable of acquiring obstacle information for the entire work space can overcome this limitation and provide the basis for secure path planning for the robot arm, including transported work pieces.

There are several techniques to acquire information about the entire environment. The use of multiple cameras, with each one surveilling the entire work space, is a technique widely used in computer graphics to determine the physical extent of objects by fusing multiple images of a scene with the back-projection method, as mentioned for example in [6]. However, the focus in [6] is the precise reconstruction of an object in 3D space, including texture information, which is not necessary for this problem. For our purposes the shape of the object can be coarsely reconstructed and object textures can be neglected, as only the physical extent of obstacles is of concern for safety issues. Furthermore, this technique only reconstructs single objects but we must deal with multiple obstacles within the robot's work space.

Reference [10] and [1] present a system for obstacle detection with stationary CCD cameras. These cameras are used to establish multiple passive light barriers, a concept related to back-projection. Evenly distributed points in the robot workspace are mapped to a pixel in each camera. If the features of the pixel differ from the given background values beyond a certain threshold within each camera, the system assumes the corresponding point in space to be occupied by an obstacle. If such a point exists within the robot's path, the current movement is suspended. Path planning based on obstacle information is not implemented.

The MEPHISTO system [15] allows for human coexistence with mobile robots. It combines laser scanners mounted on the robot with a global monitoring system consisting of color cameras surveying the floor on which robots and humans move. The images obtained by the cameras are compared to a reference image. The difference image is mapped on the floor in the form of a polygonal region, allowing for fast collision detection. The system provides a path planning service for the mobile robots under surveillance in a 3-dimensional configuration space (2D-position and rotation around the axis perpendicular to the ground plane).

Similar to [10] and the back projection method, we use a difference image method that combines several cameras to detect obstacles within the workspace. Based on the difference images, the system plans alternative paths around the detected obstacles. This concept is a promising technique for the realization of safe robot workspaces.

System Overview

The workspace of the robot is monitored by several stationary cameras, each of which monitors the entire workspace shared by humans and the robot (see Figure 1). To detect obstacles within the workspace, the image processing subsystem utilizes a difference image method [4]. It detects humans and other obstacles by comparing current images with reference images of the workspace [3, 8] (Figure 2 a, b, c, d, e), yielding a *current difference image*. Pixels within the difference image are labeled *Foreground* or *Background*. Generally, reference images are composed of static environmental objects only; they do not contain the robot and human operators and are generated during a preceding setup step.

The collision test requires a difference image containing only obstacles, because the robot itself is not considered to be an obstacle. Thus, we have to eliminate any foreground pixels caused by the robot from the current set of foreground pixels. To do so, a *current robot image* is generated (Figure 2 f, g, h, i) using calibrated cameras and a model of the robot in the current configuration. Afterwards, all pixels within the workspace difference image (Figure 2 e) are set to background, where the generated robot image contains a foreground pixel, resulting in an image containing only obstacles (Figure 2 j). Collisions with future robot configurations are detected by intersecting the set of foreground pixels from a generated robot image the future configuration (Figure 2 k, g, l) with the obstacles reconstructed from the workspace difference image (Figure 2 m,j,n). If the intersection is not empty, the camera signals a collision.

Figure 1: Work cell surveilled by four cameras (indicated by icons). The transparent cube indicates the surveilled space.

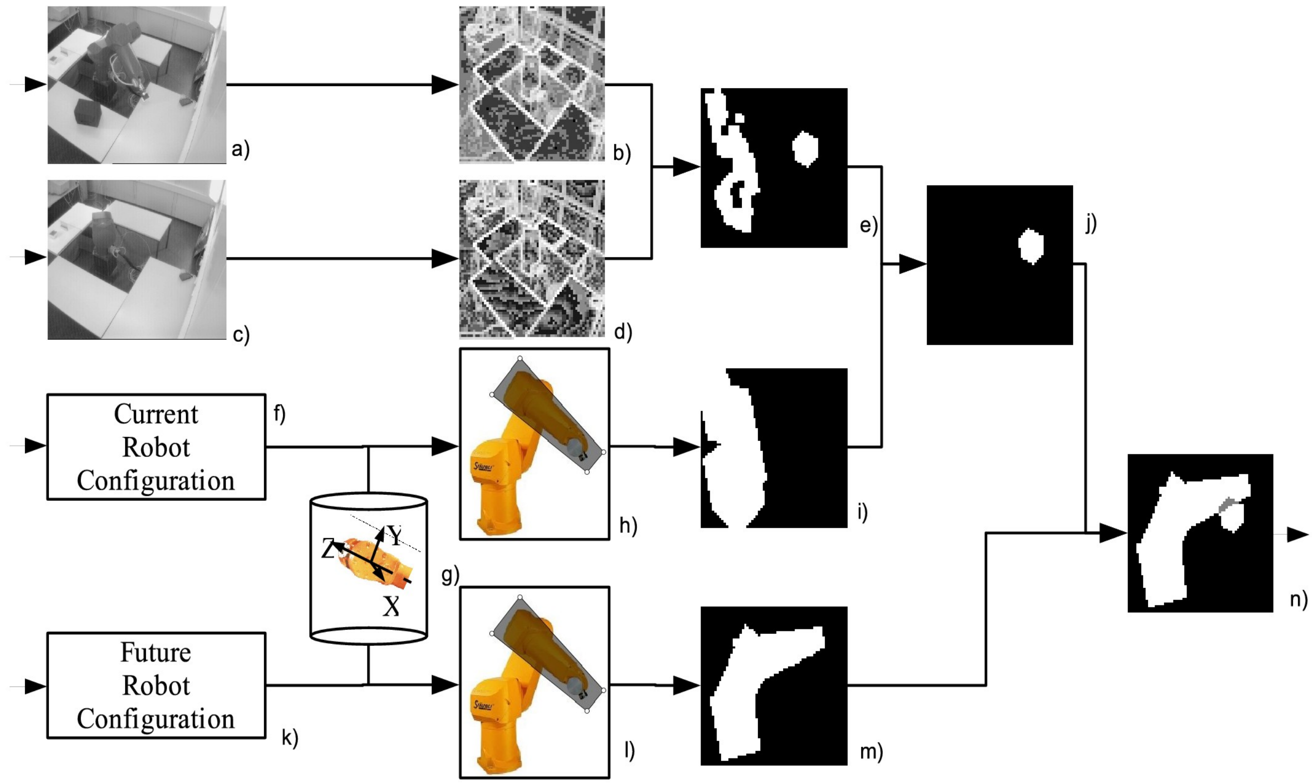

Figure 2: Flow chart of the collision detection system with simplified computation steps for a single camera.

This single-camera collision test has a major drawback; if obstacles in front of or behind the robot obscure a specific camera's view, this can cause collision detection failure as obstacles are not detected. To overcome this problem, we can use multiple cameras looking at the scene from different perspectives. Objects that are occluded can be reconstructed by using an unobstructed view of these objects from other cameras and an epipolar line method (for a detailed overview of multiple-view geometry, see [13]). In the following, we describe an algorithm that extends and modifies the computation step depicted in (Figure 2 e, i, j).

First, the current robot image is generated for each camera (Figure 2 i, Figure 3 d, e) and each foreground pixel of this image is copied into the current difference image of each camera and the copied pixel is labeled *robot pixel*, producing the images shown in Figure 3 a and f.

In the following, we describe the algorithm for Camera 0, but the algorithm is the same for Camera 1. First, a new *scene image* (Figure 3 c) is generated, containing at first only background pixels and object pixels copied from the current difference image. For each foreground pixel of the current robot image in Camera 0 (Figure 3 d), the corresponding epipolar line in Camera 1 is checked for intersection with *object pixels* (any pixel that is not a background or robot pixel) in the current difference image of Camera 1 (Figure 3 g). If there is any intersection, the corresponding pixel in Camera 0 might belong to an object and the respective pixel in the Camera 0 scene image (Figure 3 c) is labelled *pseudo-obstacle pixel*.

The area covered by epipolar lines in Camera 1 originating from all robot pixels in Camera 0 is illustrated in Figure 3 g by the light grey polygon and vice-versa in Figure 3 b for epipolar lines. We additionally marked the epipolar lines covering the object pixels within Figure 3 g (the dark grey polygon). These lines correspond to the subset of robot pixels shown by the light orange area in the scene image (Figure 3 c) representing the scene image as seen from Camera 0. When applied to Camera 1, the algorithm yields the scene image shown in Figure 3 h. It contains only an *obstacle*, as the epipolar lines in Camera 0 (Figure 3 b) had no intersection with an object.

The reconstruction is conservative in the manner in which it reconstructs at least any occluded object, but non-existent objects can be "reconstructed" too, as shown in the following.

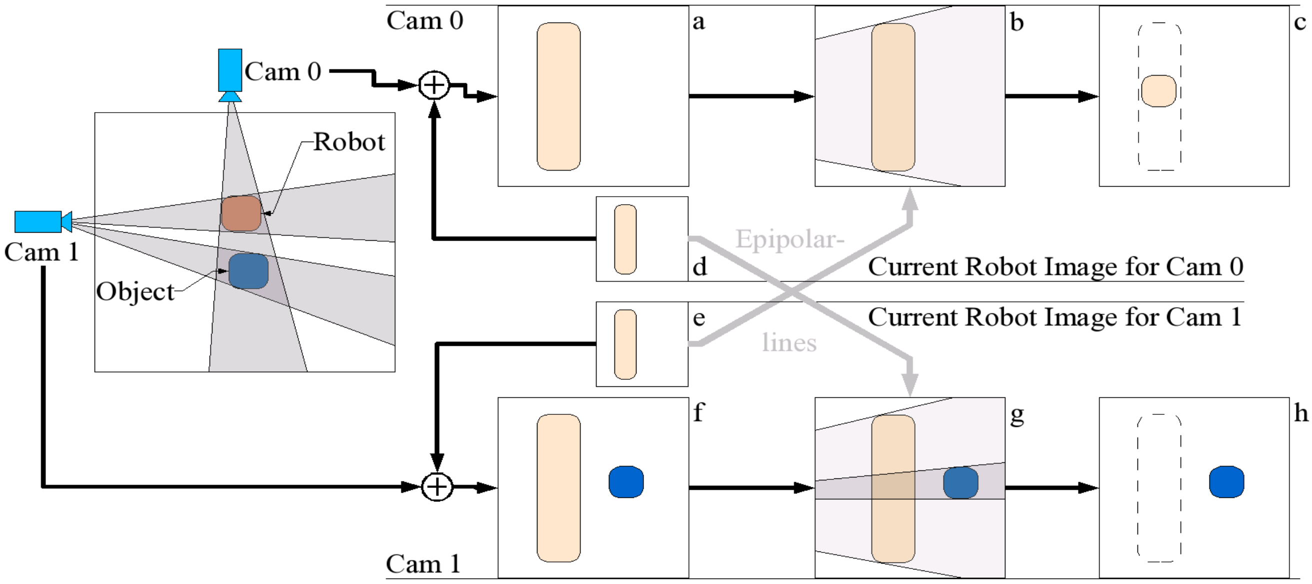

Figure 3: Object reconstruction in a case involving occlusion and multiple cameras. In images a-h, the object is represented by the small blue square and the robot by the tall orange rectangle. The picture on the left shows a top view of the scene, on the right is a sequence of three images from Camera 1 (a,b,c) and Camera 0 (f,g,h). Images a and b show the scene from each camera's perspective with the current robot images (d,e) added. Images c and h show the resulting scene images with pixels pseudo-obstacle label in orange and obstacle label in blue.

By reconstructing objects using epipolar lines, the future robot positions can be tested for collision with obstacles and pseudo-obstacles. The collision test can be formulated as follows: A robot position is considered collision-free if a single camera is free from collision with obstacles or pseudo-obstacles. Intersections with obstacles and pseudo-obstacles are treated equally. Despite this intuitive approach, special cases can arise where the reconstruction of objects leads to robot immobility, if combined with the simple collision test described previously. Figure 4 shows an example of a robot-object configuration producing robot immobility, since the robot is in touch with a pseudo-obstacle.

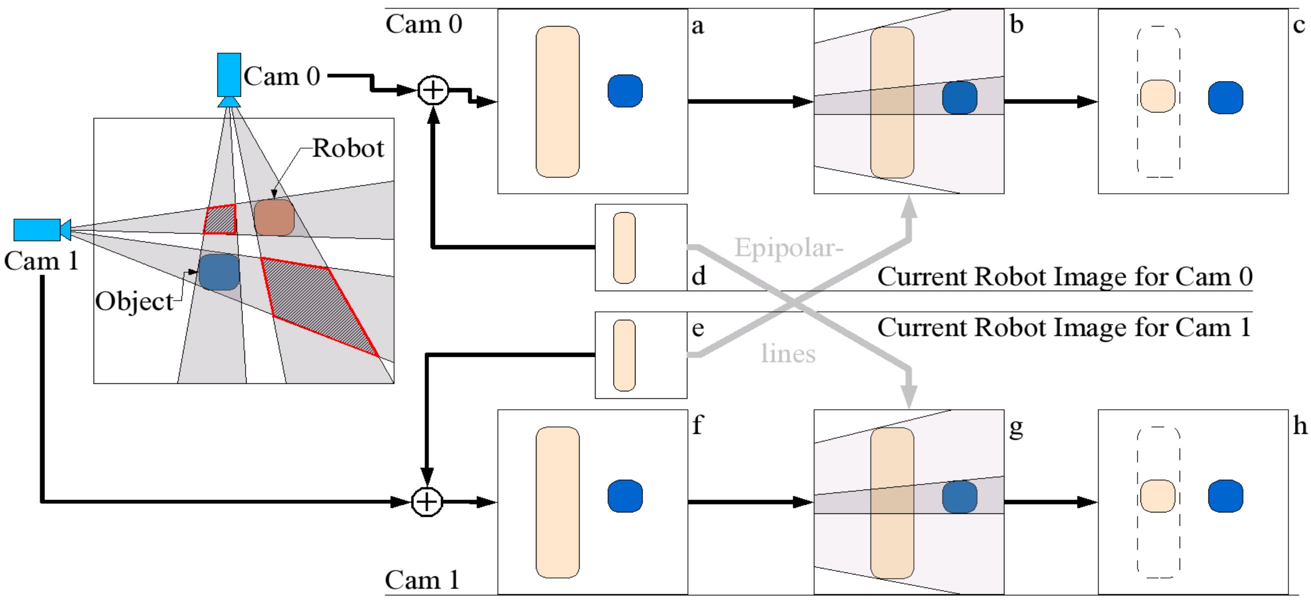

Figure 4: Object reconstruction in a case involving occlusion and multiple cameras.

In the situation of Figure 4, pseudo-objects are reconstructed for both cameras, therefore any robot positions similar to the current position always produce intersections with pseudo objects. Thus, the robot remains immobile, because any reasonable path planning starts with the robot's current position, which in this case is not collision-free. Intuitively, we could state that the robot is collision-free within its very own volume, so we obviously need an adapted rule for a collision test. To address this issue, we created a table containing all possible cases of intersection with (pseudo) objects in both cameras.

Robot Pixels Intersection in Cam 0 with		Robot Pixels Intersection in Cam 1 with		Case
Object Pixels	**Pseudo Object Pixels**	**Object Pixels**	**Pseudo Object Pixels**	
false	false	false	false	a
false	false	false	true	a
false	false	true	false	a
false	false	true	true	a
false	true	false	false	a
false	true	false	true	b
false	true	true	false	c
false	true	true	true	c
true	false	false	false	a
true	false	false	true	c
true	false	true	false	d
true	false	true	true	d
true	true	false	false	a
true	true	false	true	c
true	true	true	false	d
true	true	true	true	d

Table 1: Cases of intersection between generated robot pixels and object- and pseudo-object pixels in a two-camera system.

All of the cases mentioned in Table 1 are illustrated by sample robot positions in Figure 5. Before we describe each case in detail in the following list, we must define a threshold variable θ determining the maximum number of cameras in which an object may be occluded by the robot (see [4] for details). This definition also holds for partial occlusion of the object. For the scenario detailed in Figure 4, we will set this threshold to $\theta = 1$.

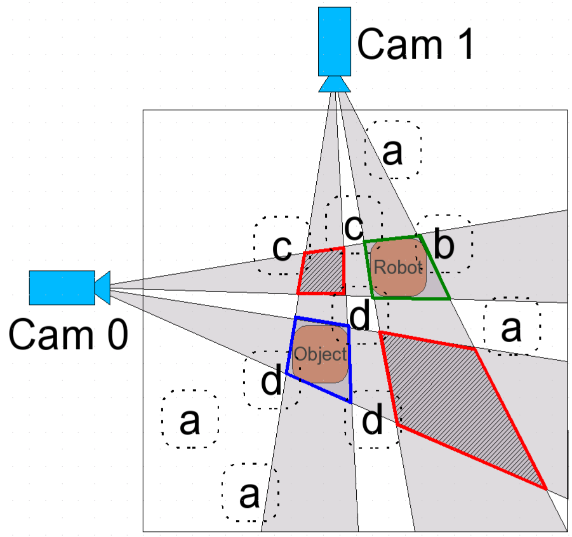

Figure 5: Sample robot positions in top view for the collision cases from Table 1. The robot is of a simplified type, moving only in the ground plane (i.e., a mobile robot). The sample positions are depicted with dotted-line squares containing the respective case identifier. In addition to the robot and object areas, the hatched areas with the red border could contain objects as well, while leading to the same camera image. Thus, collisions with these areas must be treated like collisions with a real object to ensure safety.

- Case *a:* At least one camera is completely collision free (no intersections). In this case there can be no collision with an obstacle at all, because if any volume intersection exists between a (pseudo) object and the robot, each camera contains an intersection of the respective projected volumes (for a proof see [5]).
- Case *b*: All cameras show collisions with pseudo-obstacles only. As long as θ is not equal to the number of cameras, we can state that the position is collision-free, because if it were to result in a collision, at least one camera would show a collision with an obstacle.
- Case *c*: Here one camera shows a collision with an obstacle and none of the cameras is completely free of collisions. Because θ is 1 and there are two cameras, a single collision with an obstacle is enough to determine that this robot position causes a collision (although it may not not actually collide, because the pseudo-obstacles do not really exist).
- Case *d*: All cameras show a collision with at least one obstacle, indicating a definite collision with the obstacle volume. It is possible that even these positions do not really collide, because the back projection of object silhouettes is a conservative approximation.

In summary, the collision test can be stated with the following equations. Starting with the intersection of a future robot image with obstacles and pseudo-obstacles in each camera, an overall collision statement for the given robot position is calculated.

$$Coll_{RO}(R_i, S_i) := (R_i \cap O_i \neq \emptyset)$$
$$Coll_{RP}(R_i, S_i) := (R_i \cap P_i \neq \emptyset)$$
$$Coll \;:=\; \left(|\{ i | Coll_{RO}(T_i, S_i) \}| \geq (C - \theta) \right) \quad \wedge \quad \left(|\{ j | Coll_{RO}(\boldsymbol{T}_j, \boldsymbol{S}_j) \vee Coll_{RP}(\boldsymbol{T}_j, \boldsymbol{S}_j) \}| = C \right)$$

In these equations, i is the camera identifier, S_i is the scene image containing sets of foreground pixels labeled obstacle (O_i) or pseudo-obstacle (P_i). $Coll_{RO}$ / $Coll_{RP}$ is true if any intersection with the set of generated robot pixels (R_i) exists. C is the overall number of cameras and θ is the occlusion threshold mentioned above. Thus, a collision is detected if more than $C-\theta$ cameras detect an obstacle collision and no camera is completely free of either obstacle or pseudo-obstacle collisions.

Conclusion

We presented a supervision system that uses stationary cameras to enable safe human and robot coexistence in a shared workspace. The basic obstacle detection uses a simple, efficient difference image method. Possibly occluded objects are reconstructed to ensure safety even in the case of occlusion through the robot. The system can be adapted to different kinds of robots by varying the number of cameras and modifying the occlusion threshold θ accordingly. Compared to [4], the collision test retains the information about reconstructed objects (pixels labeled pseudo-obstacle) as long as possible in order to distinguish between different cases of robot pixel intersections, ultimately providing improved robot mobility. The collision test can be used as a basis for an collision-free path planner.

References

[1] Ameling W. (Ed.): „Flexible Handhabungsgeräte im Maschinenbau“; Ergebnisse aus dem Sonderforschungsbereich 208, VCH Publishing, 1996

[2] Henrich, D., Kuhn, S.: "Modeling Intuitive Behavior for Safe Human/Robot Coexistence and Cooperation", In: 2006 IEEE International Conference on Robotics and Automation, May 15-19, 2006, Orlando, Florida USA

[3] 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, pp. 239-244, Baden-Baden, Germany, August 20 - 22, 2001

[4] Ebert D., Henrich D.: „Safe Human-Robot-Cooperation: Image-based collision detection for Industrial Robots" In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1826-1831, Lausanne, October 2 - 4, 2002

[5] Ebert D.: “Bildbasierte Erzeugung kollisionsfreier Transferbewegungen für Industrieroboter“ PhD Thesis, Informatics Faculty, University of Kaiserslautern, Germany, 2003

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

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

[8] Heinzen F.: „SIMERO – Robuste und Schnelle Erzeugung von Silhouetten aus Grauwertbildern“, Diploma Thesis, Informatics Faculty, University of Kaiserslautern, Germany, 2003

[9] 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.

[10] Meisel A.; Föhr R.; Ameling W.:“3D-Kollisionsschutzsensor auf der Basis von CCD-Kameras“, In SENSOR 91, pp. 157-170, Nürnberg, May 1991

[11] 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.

[12] 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.

[13] Pollefeys, Marc: "Tutorial on 3D Modeling from images", ECCV 2000, Dublin, Ireland

[14] Radke R. J., Andra S., Al-Kofahi O., Roysam B.: „Image Change Detection Algorithms: A Systematic Survey“, IEEE Transactions on Image Processing, 2004

[15] Steinhaus P., Ehrenmann M., Dillmann R.: MEPHISTO: A Modular and Existensible Path Planning System Using Observation. ICVS 1999 361-375

[16] 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.