FUZZY COLLISION AVOIDANCE USING STEREO-VISION

Andreas Bischoff [*] **and Michael Gerke** [*]

[*] *Department of Electrical Engineering, University of Hagen, D-58084 Hagen, Germany*

Abstract: A laboratory implementation of a reactive collision avoidance for industrial robots is presented, which holds real-time conditions for typical controllers. The approach is based on video supervision of the robot provided by stereo-vision. A fuzzy strategy is used to calculate collision-free deviation paths in real-time. The PUMA 562 robot is chosen for experimental verification of real-time feasibility.

Keywords: Realtime collision avoidance, Fuzzy-reactive strategy, stationary industrial robots, stereo-vision.

1. INTRODUCTION

In advanced robotics, there is still a strong need for a fast reactive collision avoidance computable in real-time. This is in order to prevent the robot from hazardous situations during motion.
Most of these situations occur due to unforeseen events (e.g. collision objects entering the workspace accidently) or incorrect remote operation (e.g. scenery misinterpretation during teleoperation).
A 3D sensor supervision of the robot and its environment could manage critical situations. A handling of collision conflicts implies a collision prediction and a strategy to solve the problem. Advances solutions not only cause a blockade to prevent damage, but recalculate actual paths locally and at the same time intend to reach preplanned goal positions on deviation paths.

Collision prediction from 3D sensor input becomes more and more feasible in real-time. However, finding a deviation path is still much more complicated because of the following reasons:

- Collision avoidance has to be calculated within the cycle time of the robot controller (typically < 30 milliseconds).
- Realtime collision avoidance uses local strategies and therefore cannot provide global solutions.
- A collision avoidance strategy has to consider the kinematics and dynamics of the real robot to provide feasible deviation paths. In the case of an industrial robot (e.g. with 6 revolute joints) this problem becomes hard to solve.
- Robot controller interfaces are used to enforce corrective action, whenever necessary. This remote access is a widespread feature in modern controllers.
- Any conventional rulebase for collision avoidance usually becomes very complex, if standard kinematics are regarded. It is not easy to maintain these rulebases, or to adapt them to different kinematics.

To avoid most of these difficulties, we chose 3D sensor fusion based on a 2-camera-system to get environmental information and we request internal data from robot sensorics to derive its position and orientation in 3D space. This information is merged together for collision detection. The consequent collision avoidance strategy is calculated based on a fuzzification of the robots workspace and consequent fuzzy inference to handle collision conflicts.

2. SYSTEM OVERVIEW

To implement our strategies for collision detection and avoidance, we use a PUMA 562 robot and observe its environment with a 3D vision system. To cause unpredictable events, a randomly moving object enters the PUMA workspace.
The PUMA robot is still under control of its original MARK III controller unit, which is enhanced with the Unimation second Quad-Serial board. This standard hardware extension enables real-time path modification via the ALTER interface. Here, a VME computer system acts as host. Three different CPUs and special image processing hardware are necessary to handle real-time communication with the robot and -in parallel- carry out image processing tasks based on the 2-camera sensor input.
A simplified data flow map is given in figure 1.

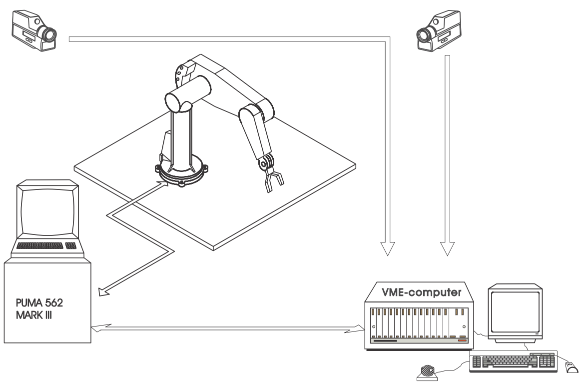

Fig. 1. Data exchange between subsystems

Data acquisition from external and internal sensorics provides knowledge about the robots location in its actual environment. Collision prediction and fuzzy collision avoidance leads to modified path segments, which solve any conflict locally.

3. SENSOR-BASED COLLISION PREDICTION

A 2-camera stereo vision system produces an enormous stream of video data. In order to keep the collision prediction problem feasible, our main objectives during image processing are twofold:

- Condense the video data towards the most relevant information about object configurations with respect to the robot.
- Generate 3D information about the robots' environment from two different 2D sensor inputs.

To provide these objectives an image processing procedure according to figure 2 has been set up.

3.1 *2D Image Processing*

The first steps of 2D image processing are carried out on video real-time hardware. This hardware exists twice, once for each of the 2D video channels. After grabbing the video frame an edge extraction is applied by means of filtering and thinning (erosion and dilation). The result is given in figure 3.

Based on the extracted image, an edge segmentation is carried out. At this point, a table of vectorized lines is available which is updated every 40 ms.

Vector	Start	End
1	22, 11	104, 32
2	205,303	97, 74
3	447, 28	399,211
4	89,503	193,359
⋮	⋮	⋮

Table 1. Unsorted vector table

From this table of unsorted vectors describing edges of the original image (after FIR filtering, computation of local gradients with look-up-tables, extraction of maxima of gradients, binarization and thinning) we derive closed contours by vector segments probably belonging together.

Contour	Vector	Start	End
1	1	22, 11	104, 32
	2	104, 32	88, 95
	3	88, 95	89,503
	4	89,503	193,359
	5	193,359	22, 11
2	1	205,303	97, 74
⋮	⋮	⋮	⋮

Table 2. Sorted list of closed contours

For each of these closed 2D contours we try to find an enclosing simple geometrical shape. This enclosing shape is reduced to some describing attributes (e.g. upper left corner, lower right corner).

3.2 *3D Image Processing*

As we intend to treat the collision prediction problem in 3D space, the most important step in image processing still lies ahead, namely the task of merging 2D positions (attributes of closed contours) towards 3D positions (attributes of closed bodies) according to stereo vision techniques. Some approaches are given in (N.Ayache, 1990), (R.L.DeValois and K.K.DeValois, 1988) and (R.Horaud and T.Skordas, 1988).
Stereo vision is subdivided here into several tasks:

- Camera modeling and lense correction
- Stereo calibration
- Solving the problem of correspondences.

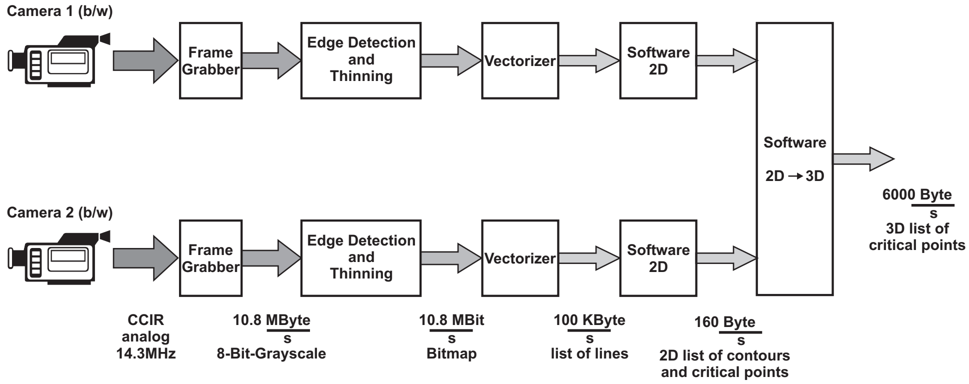

Fig. 2. Image processing procedure

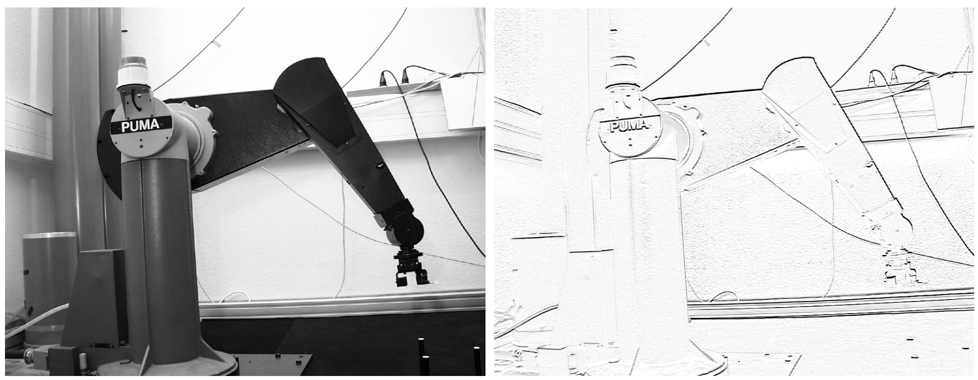

Fig. 3. Edge detection in realtime (original and extracted image)

3.2.1. *Camera modeling and calibration* Classical approaches (e.g. (R.Tsai, 1986)) for camera modeling and calibration make use of a set of internal and external camera parameters, which have to be identified. In our case, the physical identification of the camera is not a primary goal, but these parameters are necessary to find a relation between the 3D world and the specific 2D video image. A simple and robust method to find this relation without analyzing internal and external parameters is introduced by (R.C.Bolles, 1981). We applied lense correction algorithms to suppress their nonlinear radial distortions.
For calibration and correction purposes the Visual Information Processing (VIP) software of the Robotics and Vision Research group ((P.Kowesi and Huynh, 1998)) of the University of Western Australia turned out to be very helpful.

3.2.2. *Problem of correspondences* The problem of finding correspondences in 2D stereo images in order to derive the associated 3D positions in space is very challenging.
In our case, hardware image processing no longer provides greyscale images, but already delivers vectorized lists of edges and consequently attributes of closed contours. The advantage of these 2D image processing procedures is a drastical reduction of information towards identification of critical extremities in our 2D image. However, classical approaches for attribute-based stereo vision are no longer applicable.
In order to find critical objects in 3D space, we had to apply an own strategy:

- During 2D image processing we derive closed contours by vector segments probably belonging together.
- We calculate a simple enclosing geometrical shape for each closed contour. This enclosing shape is reduced to some describing attributes.
- The final problem is to associate these two attributes of each closed 2D contour to corresponding attributes derived from the second image processing system. As a result, attributes of an enclosing 3D body are found for matching closed contours. From these 3D attributes, a collection of critical positions in 3D space is calculated.

Figure 4 outlines the problem of merging two corresponding contours with critical positions.

As a result of hardware and software image processing, each real object detected by the stereo vision system is assigned to a 'virtual object' enclosing it. This 'virtual object' is of simple geometric shape, usually a cube. Any cube can be described by the two (min_{xyz}, max_{xyz}) positions of its opposite corners.
Assuming more than one object to be present in

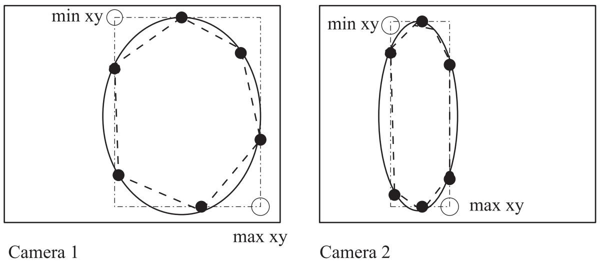

Fig. 4. Merging positions in stereo vision

the robots workspace, a set of 3D positions has to be compared to the actual robots pose (which can be described by critical extremities such as 'shoulder', 'elbow' and 'gripper'). We select the most critical 3D object position per video frame with respect to the PUMA robot to initiate a real-time fuzzy collision avoidance.

4. FUZZY COLLISION AVOIDANCE

The origins of a reactive real-time collision avoidance are in (H.Hoyer, 1984), (V.J.Lumelski and E.Cheung, 1993), and others.
A fuzzy strategy for real-time collision avoidance has been introduced and discussed intensively in (M.Gerke and H.Hoyer, 1995) and (M.Gerke and M.Tarokh, 1998). Its concept is outlined here briefly to give a basic understanding how it works.

The real-time computation of collision-free trajectories for an industrial robot in the presence of unknown moving objects hinges on drastic reductions of the problem.
The reduction of video information has already been discussed in the previous section, whereas the permanent real-time reduction of actual robot kinematics should be summarized here. For more details refer to (H.Hoyer and U.Borgolte, 1994*b*).
The idea behind a kinematics reduction is to identify robot extremities for the purpose of describing the robot shroud. It turns out that only few of such extremities are sufficient for representation of the robot.
For example, the end-effector, elbow and shoulder can be considered as extremities of a PUMA with six revolute joints. Knowledge about these extremities and their actual position in 3D space can be readily obtained from robot geometry and kinematics. Furthermore, it is shown in (H.Hoyer and U.Borgolte, 1994*a*) that these extremities can be used to compute the coordinates of a single virtual critical point for the robot in real-time.
Using this concept (again and again for small time-slices) the robot is essentially reduced to a single point in 3D space. This position as well as any object positions in workspace have to be compared for the purpose of collision checking and avoidance. Of course, the drastical reduction of a robot and any objects requires fast up-dating of scenery information in real-time.

It turns out to be convenient to describe the critical point of the robot and all information about critical objects in cylindrical (r, φ, z) coordinates. The origin of this cylindrical frame is located on the base frame of the robot. Beneath the (r, φ, z)-description of all critical 3D positions under consideration, for collision avoidance purpose it is helpful to include information about changes of critical positions $(\dot{r}, \dot{\varphi}, \dot{z})$ to treat different situations (object approaching, object departing) appropriately.
In order to achieve robust and intelligent collision avoidance, positions and velocities of critical locations are fuzzified[1]. A brief overview about how to subdivide the robots workspace into fuzzified sections is given in figure 5. It describes these fuzzy sections in the **base** plane; additional vertical intersection of the robots workspace is provided by fuzzified layers from floor level to upper level ('READY' pose of the PUMA).

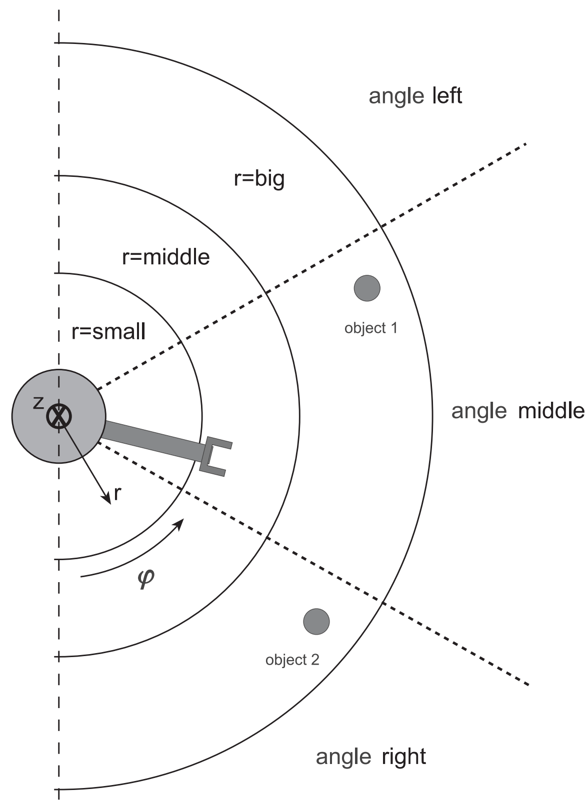

Fig. 5. Fuzzy sections subdiving the base plane

With respect to a collision avoidance strategy it has been found, that a retraction of the radial component of a robots most critical point is efficient in solving collision conflicts, whereas

[1] Fuzzy sets are positive big (PB), positive small (PS), medium (M), negative small (NS) and negative big (NB) for radial components. Equivalent expressions are used for angular and vertical position and velocities.

changing its angular component is important if radial retraction is either not sufficient or does not solve the conflict. Modifying the vertical component is useful in some cases when it is combined with radial retraction. Usually, the detection of a potential collision leeds to substantial changes in radial components accompanied by less drastical changes in vertical and (sometimes) angular components.
Based on these reasonings, a set of fuzzy inference matrices has been derived, one of which is given in figure 6.

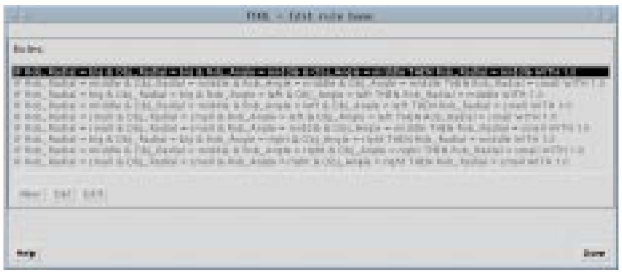

Fig. 6. Inference rulebase for collision avoidance

Let rule 1 of this particular rulebase serve as an illustrative example. It says:
IF Robot_Radial = big & Object_Radial = big & Robot_Angle = middle & Object_Angle = middle THEN Robot_Radial = middle WITH 1.0

This means, whenever the robot and an object (more precisely: their extremities) meet in the section **radial big, angular middle** (ref. to figure 5), an attempt is started to retract the robot towards a section **radial middle**.
Although there are strange object configurations possible, where the fuzzy rules would not provide a satisfactory solution to the collision avoidance problem (but cause a blockade instead), they are very straight forward for common conflicts.

To conclude the description of our fuzzy[2] strategy:
No additional weighting factors are used for individual rules. Defuzzification is based on the 'mean of maximum'-method applied to the actual output fuzzy set. This method is computable in real-time. Crisp output values in (r, φ, z) proposed by fuzzy inference are to be transformed back into joint positions. This problem is non-trivial, however in general a transformation can be calculated.

5. EXPERIMENTS

This section gives a brief impression how real-time collision avoidance performs during our experiments. Here are some details and comments about our experimental constraints:

- Image processing (hardware and software) is feasible in video real-time (25 frames per second) as long as the scenery remains simple. Complex sceneries slow down computation and reduce the rate of scenery interpretations. If so, a collision prediction becomes insufficient.
- To facilitate the detection of moving objects in our scenery, we reduced the amount of information through selection of a robust background and respraying of the PUMA arm. This increases contrasts and decreases critical reflexions.
- The PUMA is operated with its original MARK III controller and path modification is implemented via its (slow) serial ALTER interface.

During our experiments the PUMA carries out a simplified pick-and-place operation between two floor marks (tubes) in workspace. The preplanned motion is the straight path between these marks. Figures 7 to 10 present the collision avoidance of the PUMA when an object is entering that region.

Fig. 7. Realtime collision avoidance, part 1

Fig. 8. Realtime collision avoidance, part 2

[2] Fuzzy reasoning has been implemented using the FOOL software tool. Details can be found in (Uni, n.d.).

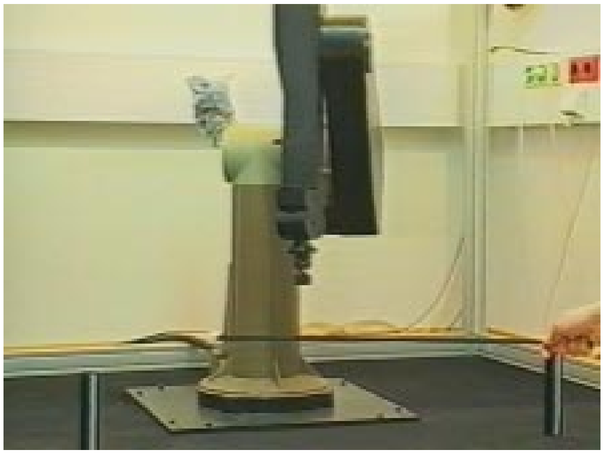

Fig. 9. Realtime collision avoidance, part 3

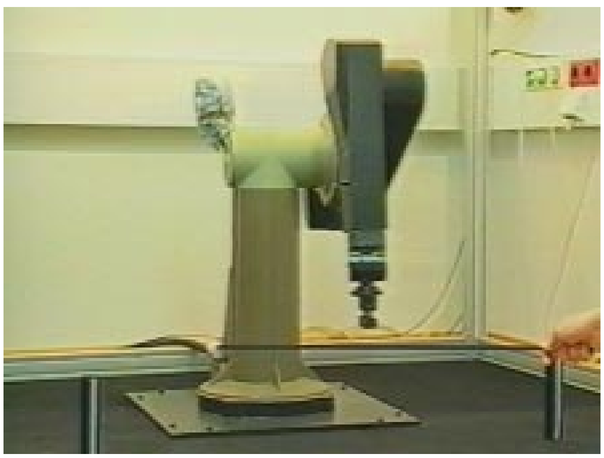

Fig. 10. Realtime collision avoidance, part 4

6. CONCLUSIONS AND OUTLOOK

It was our intention to prove, that collision avoidance can be calculated completely on a host system without extending the robot's original hardware and operation system, if some sort of controller interface for path modification is available. Of course any self-developed 'open controller' or -at least- a more advanced controller than the PUMA MARK III would have provided a faster and smoother performance during collision avoidance.

One of the major drawbacks of the current approach is the bottleneck of a serial communication between robot controller and host computer. Maximum speed allowed for this serial communication is 38400 (!) Baud. Via the ALTER communication link, information about the robots actual location is provided in one direction and path modification is enforced in opposite direction. The communication protocol is quiet simple, however strict real-time communication conditions have to be met. Any delay or failure to respond to communication requests leads to an error stop caused by the PUMA controller.

In order to evaluate the full functionality of our concept of real-time collision avoidance an alternative robot controller is considered for future experiments.

Our results can be transferred to mobile robots because of the simplified collision avoidance problem given in a planar environment (3 degrees of motion only). Of course this assumes a rearrangement of the stereo vision system (e.g. as on-board sensorics). An adaption and implementation of this fuzzy collision avoidance for a simple mobile robot under stereo-supervision has started.

7. REFERENCES

H.Hoyer (1984). On-line collision avoidance for industrial robots. In: *Preprints SYROCO 1984*.

H.Hoyer, M.Gerke and U.Borgolte (1994*a*). On-line collision avoidance for industrial robots with six degrees of freedom. In: *Proc.IEEE Int.Conf. on Robotics and Automation, ICRA 1994*. pp. 1258–1265.

H.Hoyer, M.Gerke and U.Borgolte (1994*b*). Real-time collision avoidance for industrial robots. In: *Preprints SYROCO 1994*. pp. 777–782.

M.Gerke and H.Hoyer (1995). Fuzzy collision avoidance for industrial robots. In: *Proc. 1995 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'95*.

M.Gerke and M.Tarokh (1998). Fuzzy logic real-time collision avoidance of multi-robots. In: *Proceedings: World Automation Congress WAC'98, 7th ISORA*.

N.Ayache (1990). *Artificial Vision for mobile Robots*. MIT Press. Boston, MA.

P.Kowesi, C.Fischer and Du Q. Huynh (1998). *VIP Version 4.0 User's Manual*. Robotics and Vision Research group, Department of C.S., University of Western Australia. http://www.cs.uwa.edu.au/robvis/VIP.html.

R.C.Bolles, J.H.Kremer, R.A.Cain (1981). A simple sensor to gather three-dimensional data. Technical report. SRI Tech Note 249, Menlo Park.

R.Horaud and T.Skordas (1988). Structural matching for stereo vision. In: *Proc. 9.th Int.Conf.on Pattern Recognition,ICPR'88*.

R.L.DeValois and K.K.DeValois (1988). *Spatial Vision*. Oxforn Psychology Series No. 14. Oxford University Press. Oxford.

R.Tsai (1986). An efficient and accurate camera calibration technique for 3 d-machine vision. In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. pp. 364–374.

Uni (n.d.). *FOOL, the Fuzzy Organizer Oldenburg*. http://condor.informatik.uni-oldenburg.de/FOOL.link/what.html.

V.J.Lumelski and E.Cheung (1993). Real-time collision avoidance in teleoperated whole-sensitive robot arm manipulators. *IEEE Trans. on Systems, Man and Cybernetics* **23**, 439–445.