In: IEEE/RSJ 2007 International Conference on Intelligent Robots and Systems, Oct. 29 – Nov. 2., 2007, San Diego, CA, USA

Fast Vision-Based Minimum Distance Determination Between Known and Unkown Objects

Stefan KUHN and Dominik HENRICH

Lehrstuhl für Angewandte Informatik III (Robotik und Eingebettete Systeme)
Universität Bayreuth, D-95445 Bayreuth, Germany
E-Mail: {stefan.kuhn, dominik.henrich}@uni-bayreuth.de
http://ai3.inf.uni-bayreuth.de/

***Abstract* – We present a method for quickly determining the minimum distance between multiple known and multiple unkown objects within a camera image. Known objects are objects with known geometry, position, orientation, and configuration. Unkown objects are objects which have to be detected by a vision sensor but with unkown geometry, position, orientation and configuration. The known objects are modeled and expanded in 3D and then projected into a camera image. The camera image is classified into object areas including known and unknown objects and into non-object areas. The distance is conservatively estimated by searching for the largest expansion radius where the projected model does not intersect the object areas classified as unknown in the camera image. The method requires only minimal computation times and can be used for surveillance and safety applications.**

Index Terms* – *vision, camera, distance determination, safety, surveillance

I. INTRODUCTION

Distance determination is useful in several domains. In modern vehicles it is used in distance control systems for parking and cruise control features. It is used for 3D scanning of objects and mobile robotics, to scan the environment in a plane and build maps. Recently new areas of application have been investigated. Distance determination is used for surveillance of exhibits in museums in order to avoid placing barriers around it. Human/robot cooperation tasks in industrial settings require distance determination between robot and human and thus to allow for removal of safety fences. A concept for human/robot cooperation using distance-controlled velocity is presented in [6].

In this domain, various sensors such as depth sensors, proximity sensors, cameras and tactile sensors have been used. Examples for depth sensors are stereo cameras and laser scanners. In the following, systems using different kinds of techniques and sensors in this domain are described.

SafetyEye [8] is a system that places virtual safety fences around a robot. Three color cameras are arranged in a small triangle and used to detect any violation of those virtual fences. Two types of fences are possible, placing warning and error areas around a robot. Stereo vision algorithms are used to detect the human, but unfortunately the frame rate of this system is not specified. It has a resolution of 7 cm at a distance of 5 m. Three "distances" can be distinguished, outside the warning area, inside the warning area, inside the error area. The virtual fences are static.

In [9] a laser range finder acquires 1½D distance information within a plane just above the floor of the work cell. Any dynamic obstacle detected is assumed to be a standing human and is approximated by a vertical cylinder. The smallest distance between this cylinder and the robot limits the maximum robot velocity. However, other obstacles or humans in a stooping posture may not be correctly approximated.

The system described in [5] uses capacitance sensors mounted on the robot. If the human is detected by those sensors, the robot's velocity is reduced. Thus only a binary determination of the distance is performed. Furthermore, it is unclear how distance can be determined when the robot carries a workpiece.

In [3] and [4] a vision-based human/robot cooperation system is presented, which calculates possible future intersections of the robot path with the dynamic environment in order to re-plan the path of the robot. Thus only the binary information about distance – *collision* and *no collision* – is used there.

In this paper, a method for fast vision-based determination of minimum distances between multiple known and multiple unknown objects is described. Known objects are objects with known geometry, position, orientation and configuration for example a robot. Unknown objects are objects which have to be detected by a classification of a camera image but with unknown geometry, position, orientation and configuration, such as a human. Distance calculation

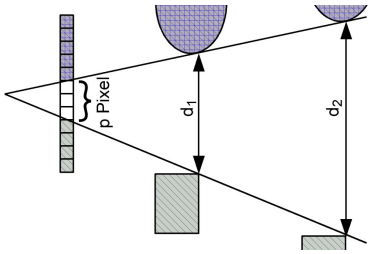

Fig. 1: Illustration of distance ambiguity in any image based distance determination. Different distances d_1 and d_2 may be projected to the same number of pixels p.

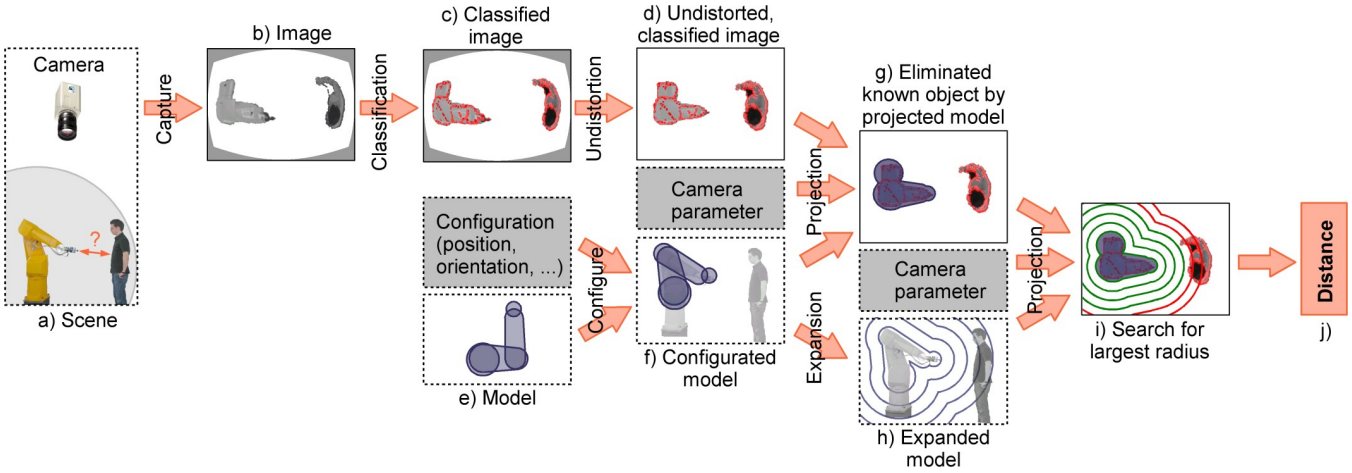

Fig. 2: Overview of the minimum distance determination process. (Explanation in the text.)

between known objects exclusively is not considered. This problem can be solved by standard methods like the modified GJK algorithm [1]. Also, distance determination between unkown objects exclusively is not considered.

Determining distances between known and unknown objects in a camera image by counting pixels is not suitable since the result is ambiguous, as shown in Fig. 1. Our approach uses 3D expansions of all known objects to search for the largest expansion of the projected known objects that does not intersect with objects classified as unkown. This results in conservatively estimated minimum distances.

The outline of the paper is as follows: In Section II, an overview of our minimum distance determination method is given. Section III describes the main aspects and steps of this method in more detail. In Section IV, extensions of our method are described. In Section V, experimental results in the domain of robotics are discussed.

II. Method Overview

The requirements to determine the minimum distance between multiple known and multiple unkown objects within a camera image are the following: A camera with known internal and external parameters like position, orientation, focal length, resolution, aperture angle and undistortion parameters is needed. Additionally, models of the known objects are needed, including position, orientation, and configuration.

With those information, the distance can be determined by the process illustrated in Fig. 2. At first an image (b) of the current scene (a) is captured. The pixels of this captured image are classified into object and non-object pixels (c). The classified image is undistorted into a pinhole camera model image (d). The models of the known objects are configured by the configuration parameters, i.e. position and orientation (e). The configured models (f) are projected by the camera parameters into the undistorted, classified camera image in order to eliminate pixels caused by the known objects (g). During a binary search, the models are expanded in 3D (h) and projected into the image several times with different expansion radii (i). The binary search determines the largest radius where the projected expanded models do not intersect classified object pixels. This largest radius represents the minimum distance between the known and the unkown objects (j).

III. Basic Method

In this section, the main aspects of the process are discussed. A simple classification method of captured images is described in Subsection A, since it is not the main focus of the paper. After classification of an image, undistortion of this classificated image into a pinhole camera model is necessary; as this method is widely known, it is only shortly described in Subsection B. The model of known objects is introduced in Subsection C and the expansion of those models is described in Subsection D. The fast projection of those expandable models is detailed in Subsection E. The search for the largest free expansion radius in the image is explained in Subsection F.

A. *Classification of the camera image*

In this section we describe a simple difference image approach to classify areas of the camera image into object and non-object areas similar to the method described in [2]. Methods from the domain of machine learning seem to be more powerful and could be used instead.

In an initial step, a reference image is created. This reference image is edge-filtered by a Laplacian filter and describes the empty scene. In determination mode, an image of the current scene is captured and edge filtered by a Laplacian filter, too. The difference between the filtered reference and filtered current image is used to calculate an object probability. A threshold can be manually or automatically determined to decide whether an object is actually located at this position. The edge filter is useful to eliminate diffuse shadows in the difference image. Other filters are conceivable.

B. *Undistortion of the camera image*

Camera images are typically distorted by the camera's lens so that the resulting image does not match a pinhole camera model. In most cases, this is due to barrel or pincushion distortion. In order to undistort such an image, the pixels of the undistorted image must be shifted according to a function dependent on the distance to the image center. The equation to calculate the undistorted pixel distance to the center of the image is

$$r_u = a + br_d + cr_d^2 + dr_d^3 + er_d^4 ,$$

TABLE I
MODEL OF THE ROBOT EXAMPLE SHOWN IN FIG. 3a)

Sphere	Radius	Position	Link	Hull
s_1	5.0	$\boldsymbol{m}_1$	1	1
s_2	4.3	$\boldsymbol{m}_2$	2	2
s_3	3.3	$\boldsymbol{m}_3$	2	2
s_4	3.1	$\boldsymbol{m}_4$	4	3
s_5	2.8	$\boldsymbol{m}_5$	4	3
s_6	2.3	$\boldsymbol{m}_6$	6	4

where r_u is the undistorted distance and r_d is the distorted distance to the center of the image. The coefficients a to e have to be determined, which can be done with standard camera calibration algorithms.

Other methods such as linear interpolation between several undistorted radii in order to undistort all positions of the distorted camera image are conceivable.

C. Model of known objects

To describe the geometry of the known objects, a sphere model is used, that allows for approximations of the object geometry and fast calculations (Fig. 3). The same model can be applied to more accurate triangulations of the object geometry, as described in Section IV.A. Here, the known objects may consist of several links connected via joints, with each link described by a local coordinate system. The geometry and configuration of these objects are known; that is, information exists about the position, orientation, and angles between the local coordinate systems. The entire object geometry is enclosed by spheres, each described by a position $\boldsymbol{m}_{sphere}$, a radius r_{sphere}, and a reference to a local coordinate system and a hull. All spheres that belong to one and the same hull are used to generate a 3D convex hull. Thus, the number of spheres necessary to describe the object geometry is reduced.

As an example, in Fig. 3a a robot is modeled in such a manner. Table I contains details about this model. As we can see, all spheres belong to a local coordinate system. Sphere s_2 and sphere s_3 are connected by the same hull association. This model can be expanded in 3D, as explained in the next subsection.

D. Expansion of the model

In order to expand a model in 3D, the radius of each sphere is expanded by the expansion radius r_{exp}; this is illustrated for our example model in Fig. 3b. The model is uniformly expanded and can be quickly projected into an image, to search for the largest free expansion radius. (By changing the order of expansion and projection, only distance measurements of pixels in image space would be possible and thus the problem of ambiguity as shown in Fig. 1 would arise.)

E. Fast projection of the model

To project the expandable 3D model, each expanded sphere is reduced to the silhouette circle describing the projection contour of the expanded sphere in the camera image (Fig. 3c). The circle lies in a plane containing the center of the sphere and orthogonal to the vector $\boldsymbol{m}_{sphere}$ from the focal point of the camera to the sphere center. The radius r_{circle} of the silhouette circle can be calculated by

$$r_{\text{circle}} = \frac{r_{\text{sphere}} + r_{\text{exp}}}{\cos\left(\arcsin\left(\frac{r_{\text{sphere}} + r_{\text{exp}}}{\|\boldsymbol{m}_{\text{sphere}}\|}\right)\right)}$$

In order to keep the calculations simple and fast, this circle is approximated by an equilateral polygon (Fig. 4), avoiding time-consuming calculations in succeeding steps. The approximation error of this equilateral polygon can be controlled by a maximum 3D deviation parameter d_{max} that simply defines the radius r_{max_circle} of a larger circle:

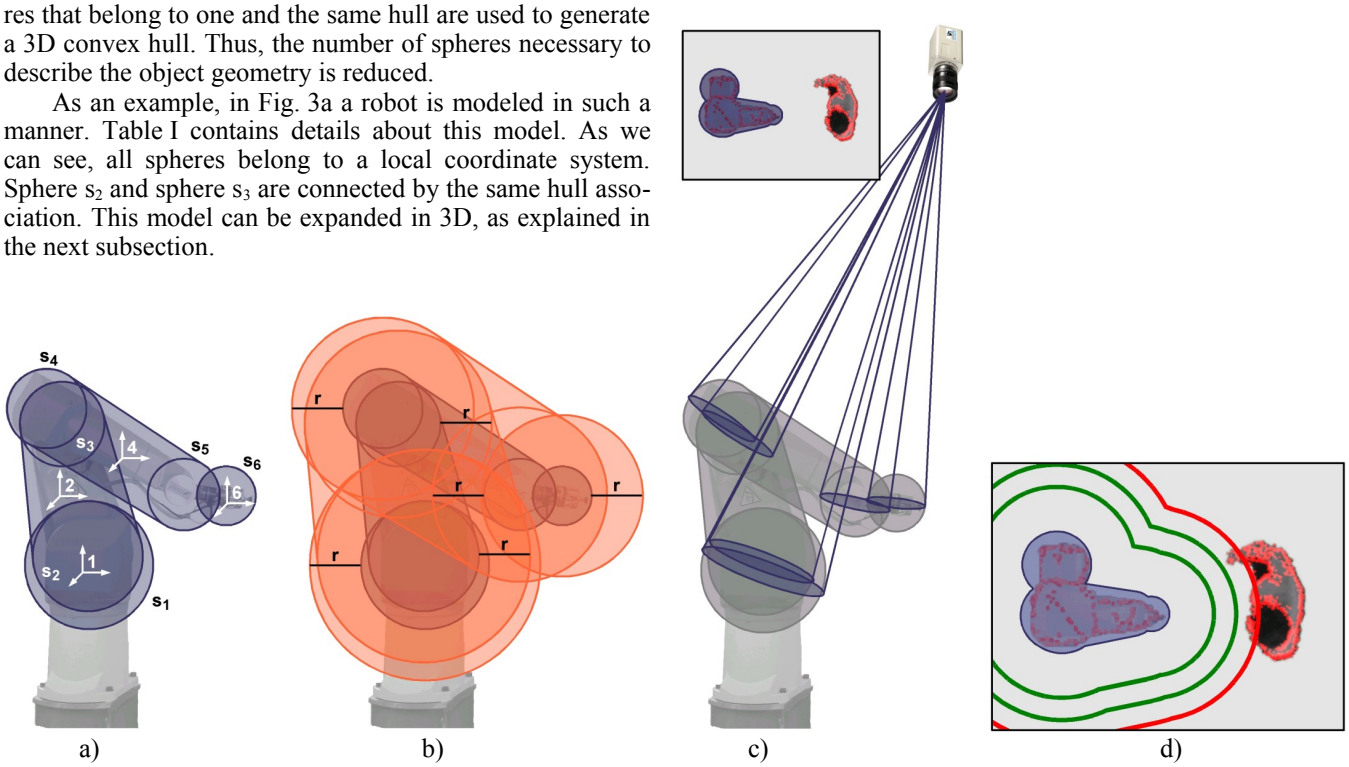

Fig. 3: Examples illustrating main aspects of the minimum distance determination process. Fig. a) shows the sphere model s_1, s_2, s_3, s_4, s_5, s_6 used to describe the geometry of a known object. b) Illustrates the expansion of the sphere model by radius r. c) Illustrates the projection of silhouette circles into the undistorted classified camera image. d) Illustrates the binary search for the largest expansion radius where no area of "unkown" objects is intersected.

$$r_{\text{max\_circle}} = \frac{r_{\text{sphere}} + r_{\text{exp}} + d_{max}}{\cos\left(\arcsin\left(\frac{r_{\text{sphere}} + r_{\text{exp}} + d_{max}}{\|\boldsymbol{m}_{\text{sphere}}\|}\right)\right)}$$

The equilateral polygon is placed between those two circles. The number n of necessary polygon vertices is calculated by

$$n = \left\lceil \frac{\pi}{\arccos\left(\frac{r_{\text{circle}}}{r_{\text{max\_circle}}}\right)} \right\rceil .$$

The vertices $\boldsymbol{p}_i^{view}$ of the equilateral polygon are computed relative to the view vector in the z direction of the camera with the equation

$$\boldsymbol{p}_i^{view} = \begin{pmatrix} r_{\text{max\_circle}} \cdot \cos(2\pi i/n) \\ r_{\text{max\_circle}} \cdot \sin(2\pi i/n) \\ \|\boldsymbol{m}_{\text{sphere}}\| \end{pmatrix}, \quad 1 \le i \le n ,$$

and the polygon is then rotated by $\boldsymbol{R}_y$ and $\boldsymbol{R}_z$ into the plane containing the center of the sphere and lying orthogonal to the vector $\boldsymbol{m}_{sphere}$

$$\boldsymbol{p}_i^{rot} = \boldsymbol{R}_z(\omega)\,\boldsymbol{R}_y(\varphi)\,\boldsymbol{p}_i^{view}, \quad 1 \le i \le n ,$$

where $\boldsymbol{R}_y(\varphi)$ and $\boldsymbol{R}_z(\omega)$ are the standard rotation matrices rotating a vector around the y- and z-axis of the coordinate system, respectively. The angles φ and ω are given by

$$\varphi = \arccos\left(\frac{m_{\text{sphere\_z}}}{\|\boldsymbol{m}_{\text{sphere}}\|}\right),$$

$$\omega = \begin{cases} \arccos\left(\dfrac{m_{\text{sphere\_x}}}{\sqrt{m_{\text{sphere\_x}}^2 + m_{\text{sphere\_y}}^2}}\right), & m_{\text{sphere\_y}} \ge 0 \\ 2\pi - \arccos\left(\dfrac{m_{\text{sphere\_x}}}{\sqrt{m_{\text{sphere\_x}}^2 + m_{\text{sphere\_y}}^2}}\right), & m_{\text{sphere\_y}} < 0 \end{cases}$$

The spheres may lie on or even behind the focal plane of the camera. In order to deal with those situations, a standard 3D convex hull algorithm of all polygons connected by a hull association is performed. This 3D convex hull is clipped at the focal plane of the camera and the resulting points are projected onto the camera image. Then a standard 2D convex hull algorithm is performed on the projected points. All resulting 2D hulls filled represent the projected model. If it can be guaranteed that all spheres of the model lie in front of the focal plane, the 3D convex hull step can be omitted.

Since the camera image is bounded, clipping at the image boundaries reduces the areas to be filled. Often several 2D convex hulls to be filled are stacked and thus those areas would be painted over many times. In order to avoid this, it is expedient to store the contours of the hulls and to perform the filling process ex post.

Finally, the special case where the focal point of the camera lies within a sphere $\|\boldsymbol{m}_{sphere}\| \le r_{sphere}$ has to be considered. In this case, the entire camera image is covered by the projected model.

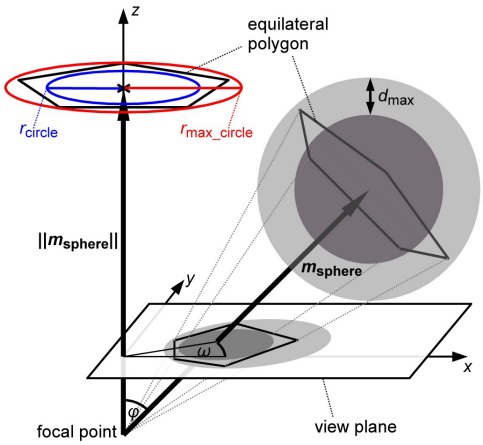

Fig. 4: Illustration of the projection.

F. *Search for the minimum distance*

In order to quickly find the largest free expansion radius of the models, a binary search is performed in the undistorted classified camera image, as illustrated in the example (Fig. 3d). If the projected expanded model intersects classified object areas, the radius is reduced, otherwise it is enlarged. The search is terminated either when the radius has been matched exactly, or when the desired number of iterations has been reached.

IV. Method Extensions

The basic method described can be extended in several ways. It is not limited to approximative sphere models. Instead, more accurate triangular models can also be handled (Subsection A). Furthermore, a slight modification places regions on the known object where no distance determination is performed so that the object can be contacted at that particular region (Subsection B). The single-camera method can also be extended to a multi-camera method to increase distance accuracy (Subsection C).

A. *Triangulated model*

Now we show that a model consisting of triangles can be represented by the spherical model as described in Section III.C. The three vertices of each triangle can be represented by three spheres with radius 0 and one common hull association. Thus, the triangular model can be completely transformed into the spherical model. Using such transformed models often results in many spheres, which adversely affects the computation time for the projection.

B. *Contact areas*

The model described is used both to eliminate pixels generated by the known objects and to determine the distance by expansion.

In order to create contactable regions on the known objects, for example to guide our robot manually, two different models must be used. The expansion model remains the same. The model for elimination of known object pixels is enlarged at the region that should be contactable. This results in a region where object pixels classified unkown are ignored and thus no distance determination is performed.

TABLE II
MODEL PARAMETERS FOR THE MODEL USED IN BOTH EXPERIMENTS

Sphere	Radius [mm]	Position (x; y; z) [mm]	Link	Hull
s_1	430	(0; 0; 0)	1	1
s_2	376	(-625; 0; 257)	2	2
s_3	305	(0; 0; 257)	2	2
s_4	310	(0; 0; 0)	3	3
s_5	214	(0; 0; 370)	6	3

C. *Multi camera system*

In [7] a method is presented that uses minimum distances determined from various camera views of a scene and calculates a more accurate minimum distance based on the assumption that only a specified number of unknown objects can be hidden in one camera. This approach is tested with one known and multiple unkown objects. It does not require solution of the correspondence problem.

V. EXPERIMENTAL RESULTS

In the following we state computation times for our implementation (Subsection A). (Subsection B) contains a real world example. Both subsections reside in the robotic domain. In both cases it is assumed that the spheres lie in front of the focal plane of the camera, so that the 3D convex hull step can be omitted, as described in Section III.E. A model with the parameters listed in Table II is used. The links are defined by the 6-axis industrial robot Staeubli RX130 with Denavit-Hartenberg parameters. The positioning of the camera differs between the two experiments. The images used for the minimum distance determinations have a resolution of 320x240 pixels. Our system is implemented in C++, compiled with the gcc version 4.1.0 and optimized with the -O3 flag, running on SUSE linux 10.1 (i586). A standard CPU (AMD Sempron(tm) 3000+, about 2 GHz, 512 KB Cache) and 512 MB RAM are used.

A. *Computation times*

To investigate the computation times of our implementation, the camera is positioned at $x = 0$ mm, $y = 0$ mm, $z = 6000$ mm and $\psi = -2.1$, $\varphi = -1.54$, $\theta = 0.9$ in the global robot coordinate system, has a focal length of 200 mm and the resolution of 320x240 pixels mentioned above. The maximum 3D deviation parameter d_{max} for the approximation of the model is set to 1 mm. The robot configuration is

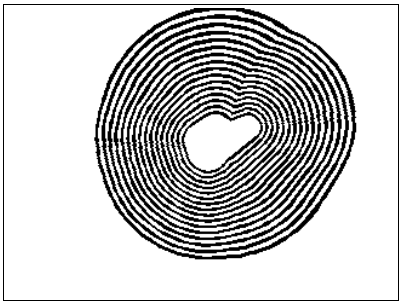

Fig. 5: Illustration of several robot expansions with radii 0 mm to 2048 mm drawn onto an image with a resolution of 320x240 pixel. The parameter set used to draw these expansions is also used for the computation time measurements in Fig. 6.

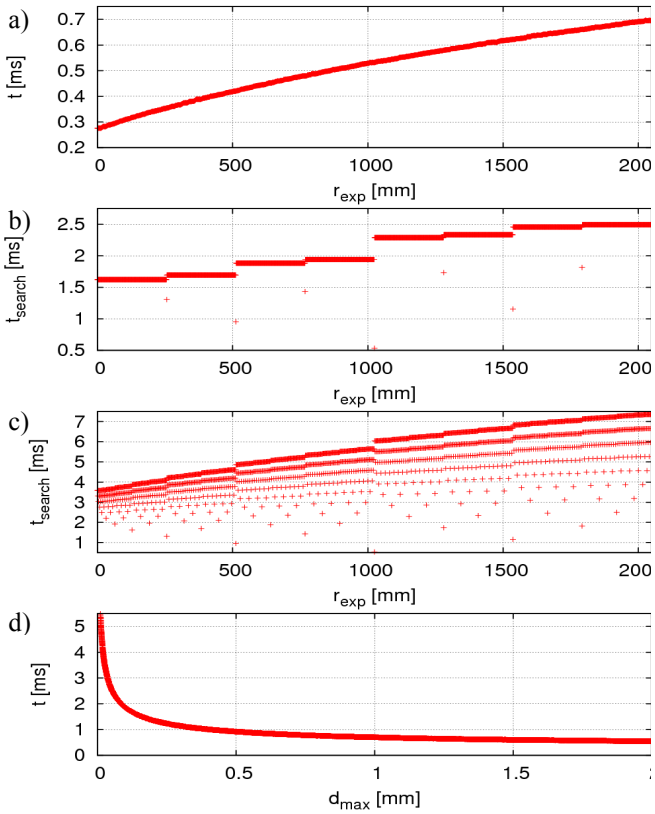

Fig. 6: Computation times for a given parameter set. a) shows the time needed to draw one single expanded version of the robot dependent on the expansion radius. b) and c) show the time needed to perform a binary search for a specific radius with four and eleven iterations respectively. The data of b) and c) are directly dependent on the data of diagram a). When a specific radius is matched by the binary search, the search is terminated. d) shows the computation times for the projection of the model with an expansion radius of 2000 mm dependent on the maximum 3D deviation parameter d_{max}.

$\boldsymbol{q}$ = (-39.21; -63.35; 202.24; 0.0; -27.39; 0). The first diagram in Fig. 6 shows the computation time required to draw one single expanded robot into the camera image dependent on the expansion radius. As we can see, the larger the projected model to be filled, the more computation time is needed to draw it. The second diagram shows the computation time required to search for a specific radius with four iterations and the third diagram shows the computation time to for eleven iterations. In both of the binary search determinations, the minimum search radius is set to 0 mm and the maximum search radius is set to 2048 mm (Fig. 5). Diagram d) shows the computation times for projecting the model expanded by $r_{exp} = 2000$ mm dependent on the maximum 3D deviation parameter d_{max}. This parameter affects the number of polygon vertices for each hull and thus the computation time.

In order to eliminate determination errors caused by the influence of the operating system, we performed multiple measurements and used the minimum value. The computation times are clearly fast enough for online use of this method in a real world example with a frame rate of 30 Hz.

B. *Real world example*

For this example we used a Firewire camera with a resolution of 640x480 Bayer pattern. The images are edge-filtered for each color channel, resulting in an edge-filtered RGB

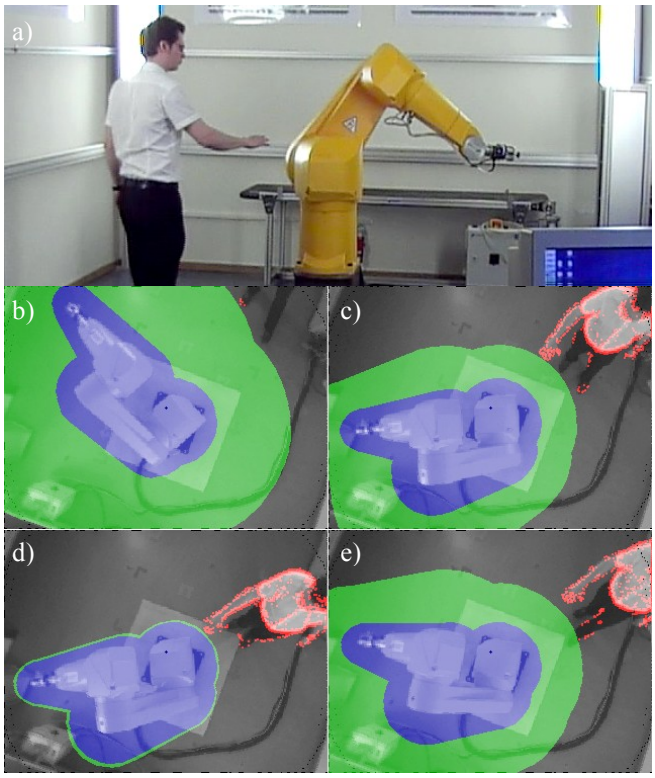

Fig. 7: Pictures of the realized surveillance system. a) shows the robot cell which is surveilled by one camera positioned above the robot. b) to e) are frames from the surveillance camera out of a system run. The blue small area eliminates the classified object pixel caused by the robot. The larger green area represents the largest non intersecting expanded and projected model. The red pixels at the human indicate the detected unknown objects. (The video can be viewed at http://ai3.inf.uni-bayreuth.de/projects/simero)

image with a resolution of 320x240. The camera is positioned at $x = 159$ mm, $y = 72$ mm, $z = 2062$ mm and $\psi = -2.094$, $\varphi = -1.538$, $\theta = 0.882$ in the global robot coordinate system and has a focal length of 200 mm. The classification process is based on a difference image approach, as described in Section III.A. The number of iterations in the binary search was set to six, so that 64 distance values can be distinguished. The minimum search distance was set to 0 mm and the maximum search distance was set to 640 mm. The maximum 3D deviation parameter d_{max} for the approximation of the model was set to 1 mm. The running system has a performance of about 29 - 30 Hz. The Unix time command result for 2167 frames was: *real: 1m13.559s; user: 0m54.315s; sys: 0m5.700s*.

Fig. 7 shows four images illustrating robot masking, distance determination and classified object pixels. The diagram in Fig. 8 shows the associated minimum distance determinations over a period of about 45 seconds. The four frames of Fig. 7b-e are marked in the diagram.. The determined distance scales the robot velocity between 0% and 100%.

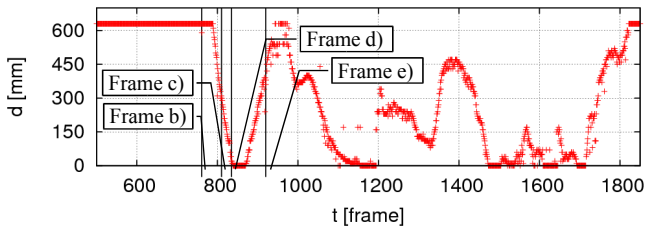

Fig. 8: Determined distances over a period of about 45 seconds. The positions of the four frames shown in Fig. 7 are marked in the diagram.

VI. Conclusions

A method for fast vision-based determination of minimum distances between multiple known and multiple unknown objects has been presented. Multiple known objects are handled either by simultaneous expandsion in the classified undistorted camera image and determination of the global minimum distance, or by successive minimum distance determination, resulting in a minimum distance per known object. Multiple unkown objects are handled implicitly, since the minimum distance determination is designed to expand the model until the nearest unkown object pixel is covered, independent of which unkown object this pixel belongs to. The distance is conservatively estimated by expanding spherical models of one or more known objects in 3D and performing a binary search for the largest expansion radius where the projected model does not intersect classified object pixels. Accuracy can be increased by using triangular models or a multi-camera system. Contact areas can be set up. The experimental results show that the calculations are very fast. This approach can be used in surveillance and safety contexts, for example in the domain of human/robot cooperation for exhibitis.

Acknowledgements

This work was supported by the German Research Foundation (DFG) under the project name „Sicherheitsstrategien für die Mensch/Roboter-Koexistenz und -Kooperation“ (SIMERO).

References

[1] Cameron S.: “Enhancing GJK: Computing Minimum and Penetration Distances between Convex Polyhedra“, In: International Conference on Robotics and Automation, April 1997.

[2] Ebert D.: “Bildbasierte Erzeugung kollisionsfreier Transferbewegungen fuer Industrieroboter“. Dissertation am Fachbereich Informatik der Universitaet Kaiserslautern, 2003.

[3] Ebert D., Henrich D.: “Safe Human-Robot-Cooperation: Image-based Collision Detection for Industrial Robots“, In: IEEE International Conference on Intelligent Robots and Systems, Lausanne, Sept 30th – Oct 5th, 2002.

[4] 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, Aug. 20-22, 2001

[5] Heiligensetzer P.: “Sichere Mensch-Roboter Kooperation durch Fusion haptischer und kapazitiver Sensorik“, Dissertation, University of Karlsruhe, 2003.

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

[7] Kuhn S., Gecks T., Henrich D.: “Velocity control for safe robot guidance based on fused vision and force/torque data“. In: IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems, pp. 485-492, Heidelberg, Germany, Sep 3-6, 2006.

[8] SafetyEye: http://www.safetyeye.de

[9] Som F.: “Sichere Steuerungstechnik fuer den OTS-Einsatz von Robotern“, In: 4. Workshop fuer OTS-Systeme in der Robotik, Stuttgart, Germany, Nov 2, 2005.