Visual Determination of 3D Grasping Points on Unknown Objects with a Binocular Camera System

Alexa Hauck, Johanna Rüttinger, Michael Sorg, Georg Färber

Lab. for Process Control and Real-Time Systems
Technische Universität München
Munich, Germany

Email: {a.hauck,m.sorg}@ei.tum.de

Abstract

In the field of hand-eye coordination, most state-of-the-art systems still require the user to select the grasping points manually. We present a system which autonomously determines 3D grasping points on unknown objects from a pair of greyscale images. The object to be grasped is segmented automatically when put into the scene. Grasping points are searched on the object silhouette; their stability is evaluated by a heuristic algorithm, primarily based on the skeleton of the region.

The 3D grasping pose is estimated by triangulation using a simplified geometrical model of the camera system; the corresponding points in the second image are determined via dynamic programming. The whole system has been implemented and validated on the experimental hand-eye system MinERVA.

1 Introduction

Despite the increasing popularity of the research field of visual control of robot manipulators, a system that is able to grasp arbitrary objects autonomously still seems to be a distant goal. This may be because up to now efforts have mainly been directed at the question how to use visual information in a motion control loop. The resulting methods for *visual servoing*[1] have brought forward a number of impressive systems (for a collection of articles see [10, 14, 23]). However, many of these systems do not approach the problem of visually guided *grasping*, but restrict themselves to the problem of how to *position* the end-effector under visual guidance, leaving open the question where to position it.

Most of the systems which actually perform grasping (e.g. Wunsch et al. [24], Tonko et al. [22], Allen et al. [1]) retrieve suitable grasping points from a geometric object model after estimating the pose of the object to grasp. This approach presents two main difficulties for the usage in an autonomous system: First, it requires a precise geometric calibration of at least some parts of the hand-eye system, depending on the hand-eye configuration. To become as calibration-insensitive as possible, a visual servoing method should be employed in which the (visual) position of the grasping points and that of the grasping device, e.g. the tips of a two-finger gripper, are measured in exactly the same way. Secondly, it requires geometric models of all objects to be grasped.

Hollinghurst [13] presented a system which fulfills at least some of the criteria above: The position of grasp and gripper are both determined via affine stereo; to extract the former, however, it is assumed that the object possesses parallel planar surfaces, which can be seen as geometric model knowledge.

Without knowledge about the object and with only one view of the object, there is only one place to look for grasping points: its silhouette, or *apparent contour*. There already exist methods to determine grasps, heuristic [16, 20] and analytical ones [21, 7] (see Sec. 2.1), but they all operate on images from a single camera, and therefore need additional context knowledge to be applicable to 3D grasping.

In order to overcome this problem, we developed a system which determines grasps on the apparent contours in a pair of images from a stereo camera system. Sec. 2 describes the underlying heuristic algorithm and the corresponding image processing, including a method for the automatic detection of the object to grasp. The reconstruction of 3D grasps is addressed

[1] For an extensive survey see [6], for a tutorial [15].

in Sec. 3; the main components are a matching algorithm based on *dynamic programming* followed by triangulation using a simplified geometric model of the camera system. The system is validated in Sec. 4 with experiments on the hand-eye system MinERVA.

2 Determining 2D grasps

In this section, we first review existing approaches and develop our own algorithm to determine grasps on an apparent contour (Sec. 2.1), then move on to describe the developed image processing modules for the detection of the object (Sec. 2.2) and the extraction of grasping points (Sec. 2.3). For image processing, the image analysis system HALCON [8, 18] was employed.

2.1 Finding grasping points

How to stably grasp an object is a research field of its own; more information can be found e.g. in [4]. Grasping unknown objects based on visual information only limits the field of applicable methods.

Kamon et al [16] present a heuristic algorithm to determine candidate pairs of contour points from a single image of an overhead camera and to evaluate the stability of the resulting grasp. The lack of 3D information is compensated by a try-and-error scheme: Successful grasps are learned by executing them with a real robot and measuring the resulting stability visually. As quite a number of the generated candidate grasps are not successful, this approach is not suitable for on-line experiments.

Another heuristic method using similar but more restrictive criteria for the evaluation of grasp stability is described by Sanz et al [20]; it has been implemented in an eye-in-hand visual servoing system. The 3D problem is not addressed.

Taylor et al [21] present an analytic algorithm to determine antipodal grasps on the apparent contour. By using this algorithm in an active vision system, the relative depth of the grasping points can be estimated to check if the grasp is antipodal in 3D as well. For the 2D case, this approach was extended by Davidson & Blake [7] to determine immobilising grasps ("caging").

The development of "yet another" algorithm in the presence of the described, successful methods was prompted by the observation that they unnecessarily restrict the set of possible solutions: The analytical algorithms by looking for grasps that are stable even when grasping with point-sized fingers in the absence of friction, and the heuristic ones by not using 3D information. For example, Sanz et al. require that the grasping points lie in the vicinity of the axis of maximum (2D) inertia and as close as possible to the (area) centroid to minimize the effect of gravity and the need for rotational friction. In the case of a object standing on a table, this restriction is unnecessary.

Therefore, we start by classifying objects as *lying* or *standing*, using the triangulation method described in Sec. 3.3. Quasi-spherical objects make up a class of their own, as they could be termed lying and standing at the same time. As in [5] and [20], the main criterion of our algorithm is based on is the *symmetry* of the object silhouette. In contrast to [20], symmetry is evaluated using the *skeleton* or *medial axis* (see Sec. 2.3). Grasps are evaluated using the following criteria in addition to symmetry:

1. the distance between the two points
2. the angle between the line connecting the two points and the horizontal plane
3. the distance of this line to the area centroid

The stability of a grasp is estimated as a weighted sum of these measures. The weights are specific for each object class: as already mentioned, criterion (3) for example is meaningless for standing objects.

The algorithm searches for grasping points until the stability estimate reaches a certain threshold, thus it finds a probably successful grasp but not necessarily the optimal one. The reason behind this is that without further knowledge about the object (e.g. density distribution or material) one cannot guarantee that a grasp is optimal, anyway.

2.2 Object detection

The main problem when working on the apparent contour of an object is that a very precise *segmentation* is required to prevent looking for grasping points on a shadow. Robust segmentation is a problem in itself, therefore researchers often resort to putting dark objects on white tables. We are no real exeption to this rule. However, as one of our scenarios sees the robot in front of a table on which the objects to grasp are placed, we developed a module capable of detecting any change in the scene and thereby segmenting the object to grasp.

First, to reduce run-time computation, the scene is initalized by defining a *region of attention* (see Fig. 1). This region is periodically checked for any changes in two consecutive images. When placing an object into the scene, the hand will first enter the region of attention, triggering a kind of alarm; only after it has

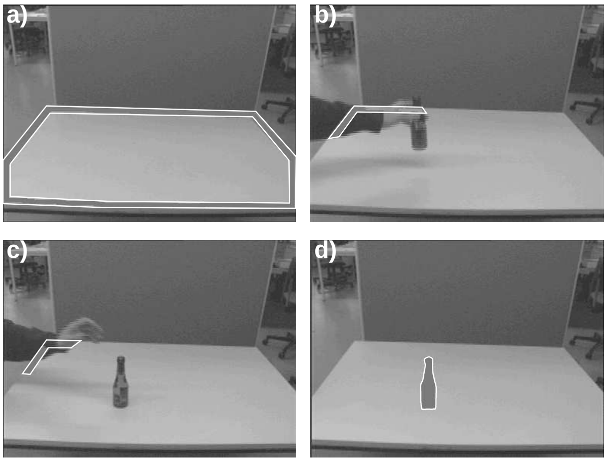

Figure 1: Scene in front of the robot: (a) empty table with region of attention, (b) before placing the object, (c) after placing the object, (d) segmented object.

left the region again, the inner region is checked for changes. This is performed by a so-called *dynamic threshold* operator provided by HALCON: This operator segments an image using a local threshold. Small changes in the scene will be melt into the static background, whereas larger changes will be signalled to the user program. A detected object is then segmented more precisely using combinations of morphological operations.

2.3 Feature extraction

After segmenting the region corresponding to the object, *features*, in the form of grasping points, are to be extracted. As mentioned in Sec. 2.1, the principal criterion is the *local symmetry* of the region as the object is to be grasped using a two-fingered gripper. A morphological feature well suited for the description of local symmetry is the *skeleton* or *medial axis* (see e.g. [9] and Fig. 2a). Each point on the skeleton corresponds to the center of a maximal-sized disk contained within the region. The main difference of our approach to the one of Blake [5] is that we do not search for symmetrical or antisymmetrical pairs of contour tangents, but directly work on the skeleton, which can be extracted efficiently using a HALCON operator. First, it is partitioned into line segments. The longer such a segment is, the higher is the probability of a stable grasp, so that's where the algorithm starts looking for grasping points. The contour is intersected with a line perpendicular to the skeleton segment iteratively until the computed stability measure of the grasp meets a given threshold (see Fig. 2).

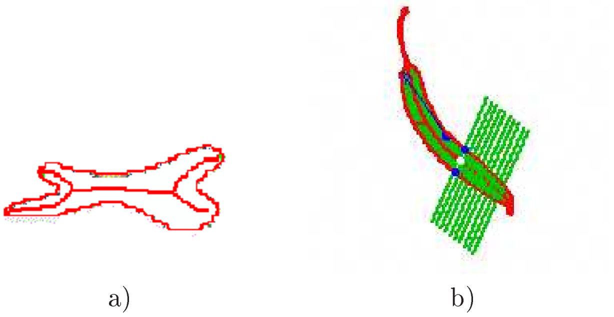

Figure 2: (a) skeleton, (b) finding grasping points.

3 From 2D to 3D

To reconstruct a 3D grasp from a stereo image pair, first the corresponding grasping points in the two images have to be found (Sec. 3.1). Based on a simplified geometric camera model (Sec. 3.2), a triangulation method can then be applied (Sec. 3.3).

3.1 Matching

The correspondence problem falls into two parts: The easier one is finding points corresponding in 2D, so to say *apparent correspondences*. This can be achieved by matching the two apparent contours and establishing point-to-point correspondences. For this, we employ an algorithm which was originally developed for object recognition [2]. Here, silhouettes in form of centroidal profiles are compared using *dynamic programming* [3], which yields a distance measure describing the similarity of the two shapes, and the point-to-point correspondences.

The second, much more difficult part is to assure that the points are projections of one and the same 3D point. Without further knowledge about the object, this can be achieved via epipolar geometry (see e.g. [19]). The problem with this approach, again, is that it is too restrictive: assuming a gripper with "real", i.e. not point-sized fingers, many grasps are feasible even if the 2D points do not precisely correspond in 3D, as e.g. on a rotationally symmetric object.

However, the similarity measure yielded by the matching algorithm can be used to check if the two cameras get similar views of the object (see Fig. 3 for the opposite case). As the camera baseline is small in comparison with the average object distance, the remaining errors can be tolerated.

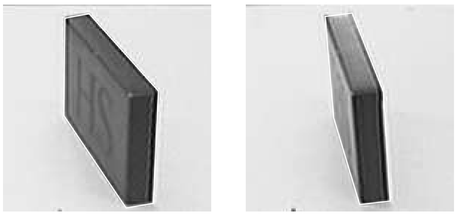

Figure 3: Left and right camera view of a polyhedral object.

3.2 Geometric model of the camera system

The main disadvantage of directly using an estimated 3D position to control robot motion is that the geometric models of the hand-eye system have to be very precise to enable a successful grasping. We therefore will integrate the method described in this paper into a position-based visual servoing system (see Sec. 4.2). This allows to use a simplified model of the camera system.

The cameras are mounted on a standard pan-tilt head; one can therefore assume that the x- and z-axes of the cameras are co-planar, i.e. the y-component of a grasping point is identical. Vergence is allowed. Fig. 4 shows the resulting, planar model.

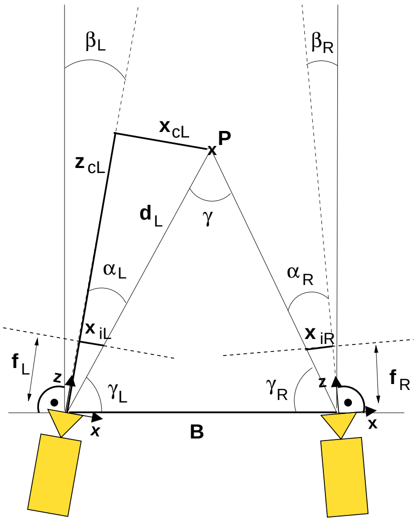

Figure 4: Simplified model of the stereo camera system.

Internal and external parameters have been identified using a multi-image calibration method based on a 2D calibration object [17] which is provided by HALCON.

3.3 Triangulation algorithm

Using this camera model, the 3D position of a point P relative to the left camera (x_{cL},y_{cL},z_{cL}), given its 2D pixel coordinates ($X_{\{L,R\}}$,$Y_{\{L,R\}}$) in both images, can be calculated via the equations for perspective projection, based on the pinhole camera model:

$$x_{cL} = \frac{x_{iL} \cdot z_{cL}}{f_L} \qquad y_{cL} = \frac{y_{iL} \cdot z_{cL}}{f_L} \tag{1}$$

$$x_{iL} = (X_L - C_x) \cdot S_x \quad y_{iL} = (Y_L - C_y) \cdot S_y \tag{2}$$

with (C_x, C_y) being the principal point and $S_{\{x,y\}}$ the scaling factors. As we are using wide-angle cameras, it is useful to compensate for the radial distortion:

$$x^*_{iL} = \frac{x_{iL}}{1 + \kappa \cdot r_{iL}} \qquad y^*_{iR} = \frac{y_{iR}}{1 + \kappa \cdot r_{iR}} \tag{3}$$

with $r_{i\{L,R\}} = x^2_{i\{L,R\}} + y^2_{i\{L,R\}}$ and κ being a coefficient describing radial distortion, which is identified during calibration.

z_{cL} is calculated using the trigonometric relations of Fig. 4:

$$z_{cL} = d_L \cdot \cos \alpha_L \tag{4}$$

$$\alpha_L = \arctan \frac{x_{iL}}{f_L} \qquad \alpha_R = \arctan \frac{-x_{iR}}{f_R} \tag{5}$$

with $f_{\{L,R\}}$ being the respective focal lengths of the cameras. Using the tangential formula one can derive:

$$\tan \gamma = \frac{B \cdot \sin \gamma_L}{d_L - B \cdot \cos \gamma_R} \tag{6}$$

with the baseline B, $\gamma_{\{L,R\}} = 90° - \beta_{\{L,R\}} - \alpha_{\{L,R\}}$, $\gamma = 180° - \gamma_L - \gamma_R$, and solve it for d_L:

$$d_L = \frac{B \cdot \cos(\alpha_L + \beta_L)}{\tan(\alpha_L + \beta_L + \alpha_R + \beta_R)} + B \cdot \sin(\alpha_L + \beta_L) \tag{7}$$

4 Experimental results

The algorithms were implemented and tested on our experimental hand-eye system MinERVA (Manipulating Experimental Robot with Visually guided Actions), which consists of a 6 DOF manipulator arm (*amtec*) and a stereo camera system on a pan-tilt head (*RWI*) mounted in an anthropomorphic fashion (see Fig. 5). A variety of everyday objects were placed on tables of different height in front of the robot.

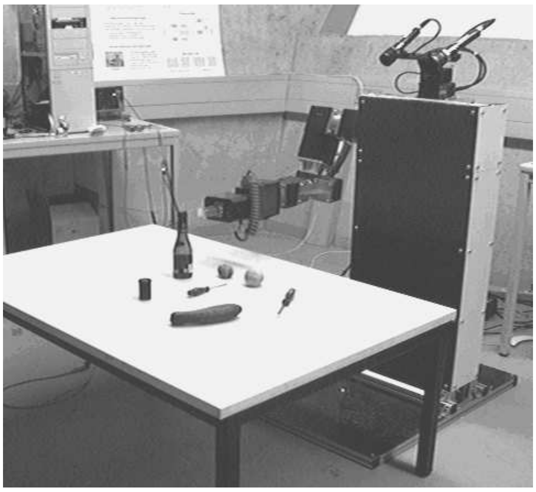

Figure 5: The experimental setup.

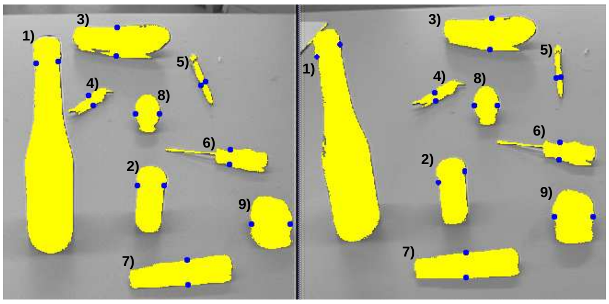

Figure 6: Resulting grasping points in the stereo image pair for: (1) bottle, (2) film box, (3) zucchini, (4) pepper, (5) pen, (6) screwdriver, (7) white board marker, (8) onion, (9) walnut.

4.1 Feature extraction

In almost all cases the object classification was successful. Failures resulted from errors in the rough distance estimation. However, this does not mean necessarily that no grasp will be computed, but that the starting conditions are less than ideal.

Fig. 6 shows the extracted grasping points in the two images.

The extracted grasps were often very similar to what a human would apply. That is to say, symmetries are found and different strategies are used for the three different object classes.

In the case of standing objects (no. 1 & 2), the grasps are always above the centroid of the segmented region to avoid toppling the object. Humans probably would grasp such objects from aside and not frontally, implicitly assuming (rotational) symmetry. In contrast, our method extracts *visible* grasps.

Lying objects (no. 3 - 7) are grasped from above, with variable orientation of the hand. Grasps in the vicinity of the area centroid are preferred, as they are more stable, at least in the case of objects with an almost uniform distribution of mass, which has to be assumed in the absence of further knowledge.

Spherical objects (no. 8 & 9) can be grasped from above or frontally. In the actual implementation, they are grasped frontally since in this case the fingers of the gripper are visible most of the time. This will facilitate the integration into the visual servoing system.

The obvious differences in the point-to-point correspondences are due to the fact that the contours were sub-sampled (factor 4) to speed up matching. The resulting errors are at the limit of what can be tolerated (see the following sections).

4.2 Reconstruction

The triangulation algorithm was tested by placing a known object at a known distance relative to the left camera, selecting corresponding points manually, and estimating their 3D position. In the relevant working space ($50cm - 90cm$ from the head), the resulting error was well below $0.5cm$ in all dimensions. No special pains have been taken regarding the calibration and the manual selection. The average error of the latter was 2 Pixel, which is similar to the error occurring during the extraction of the grasping points or the matching process.

In the case of a precise calibration of the head-arm relation, this error can be tolerated as the grasping area of the fingers is $1cm^2$ (see Fig. 7). Additional errors in the calibration of the hand-eye system will be compensated by using this method in a position-based visual servoing loop. The principal idea is that by using the same method (here: the triangulation algorithm) to estimate the position of target and gripper, calibration errors cancel out. For more information on position-based visual servoing see [15], for a detailed description of our motion control scheme [12].

4.3 Discussion

To evaluate the results, we will focus on criteria commonly used by researchers in computer vision: scope (“For what kind of objects in what kinds of scenes grasps can be found?”), robustness (“How much noise and occlusion can be tolerated?”), efficiency (“How much computing power/time is required?”), and correctness (“Will a detected grasp be successful?”).

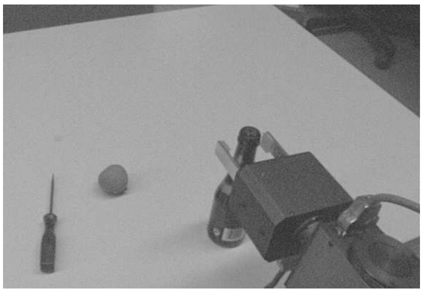

Figure 7: Camera view of the gripper at grasp position.

As already mentioned, the *scope* had been focussed on objects for which geometrical models cannot easily be constructed, i.e. non-polyhedral objects. In the other case, grasps can be modelled along with the object and then detected in the image by matching image features with model features as described for the task of object recognition in [11]. As shown in this section, indeed a great variety of everyday objects can be successfully processed. Polyhedral objects usually will be rejected during 2D matching. The main constraint on the kinds of scenes is that the object to grasp has to differ clearly from the background, in order to be segmented correctly. We will mitigate this problem by using active contours as proposed in [21] in the future. This will also enhance the *robustness* of the system, which is influenced by the same problems as the segmentation.

Concerning *efficiency*, it is worth noting that the fear expressed in [20] that 3D vision would be too costly with respect to processing time is unfounded: On a Pentium 166, the yet unoptimized software determines a 3D grasp in less than $0.5s$ for an object of average size including segmentation. As the visual servoing part has been designed specifically to work with asynchronous, definitely non-frame-rate feedback [12], this level of efficiency will already suffice. The main bottle-neck is the 2D matching which is highly dependent on the number of contour points (order $\mathcal{O}(mn)$). We plan to approach this problem by using a multi-scale algorithm.

The *correctness* can be evaluated qualitatively ("Does the determined grasp appear to be graspable to a human observer?") and quantitatively ("Would the robot successfully grasp the object when moving its gripper to the 3D grasp position?"). Qualitatively, it can be stated that with the used parameters the determined grasps always appeared to be correct; sometimes, however, no grasp is found. The quantitative correctness is harder to determine, as the module is part of a bigger system. The maximum error of the 3D position relative to the head ($0.5cm$ in all directions) alone could be tolerated; this is not true in case of additional errors in the head-to-arm calibration or the model of the manipulator itself. Actually we have integrated this method with the motion control module described in [12] and thereby realized the visual servoing system. The results showed that for objects placed in the reachable area of the robot the method was precise enough for grasping different objects (e.g. the neck of the bottle, see Fig. 5, 6).

5 Conclusion

We presented a method that will determine 3D grasps on unknown, non-polyhedral objects from a stereo pair of greyscale images. The method is reliable and fast, without using any image processing hardware. Integrated into a position-based visual servoing system, this method will bridge a gap on the way towards an autonomous hand-eye system by specifying the target position in the absence of a model of the object to grasp.

The method itself will be further improved by using active contours for a more robust segmentation, and by speeding up the matching process.

References

[1] P. K. Allen, A. Timcenko, B. Yoshimi, and P. Michelman. Automated Tracking and Grasping of a Moving Object with a Robotic Hand-Eye System. *IEEE Trans. on Robotics and Automation*, 9(2):152–165, Apr. 1993.

[2] T. Bandlow, A. Hauck, T. Einsele, and G. Färber. Recognising Objects by their Silhouette. In *IMACS Conf. on Comp. Eng. in Systems Appl. (CESA'98)*, pages 744–749, Apr. 1998.

[3] R. E. Bellman and S. E. Dreyfus. *Applied Dynamic Programming.* Princeton University Press, 1962.

[4] A. Bicchi, J. Burdick, and T. Yoshikawa, editors. *Workshop on Grasping, Fixturing, and Manipulation: Towards a Common Language.* In association with ICRA'98, May 1998.

[5] A. Blake. A Symmetry Theory of Planar Grasp. *Int. J. Robotics Research*, 14(5):425–444, Oct. 1995.

[6] P. I. Corke. Visual Control of Robot Manipulators – A Review. In K. Hashimoto, editor, *Visual Servoing*, pages 1–31. World Scientific Publishing Company, 1993.

[7] C. Davidson and A. Blake. Error-Tolerant Visual Planning of Planar Grasps. In *Proc. 6th Int. Conf. on Computer Vision*, Jan. 1998.

[8] W. Eckstein and C. Steger. Architecture for Computer Vision Application Development within the HORUS System. *Journal of Electronic Imaging*, 6(2):244–261, Apr. 1997.

[9] R. M. Haralick and L. G. Shapiro. *Computer and Robot Vision*, volume 1. Addison Wesley, 1992.

[10] K. Hashimoto, editor. *Visual Servoing*. World Scientific Publishing Company, 1993.

[11] A. Hauck, S. Lanser, and C. Zierl. Hierarchical Recognition of Articulated Objects from Single Perspective Views. In *Proc. Computer Vision and Pattern Recognition (CVPR'97)*, pages 870–883. IEEE Computer Society Press, 1997.

[12] A. Hauck, M. Sorg, T. Schenk, and G. Färber. What can be Learned from Human Reach-To-Grasp Movements for the Design of Robotic Hand-Eye Systems? In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA'99)*, pages 2521–2526, May 1999.

[13] N. J. Hollinghurst. *Uncalibrated Stereo and Hand-Eye Coordination*. PhD thesis, Department of Engineering, University of Cambridge, Jan. 1997.

[14] R. Horaud and F. Chaumette, editors. *Workshop on New Trends in Image-Based Visual Servoing*. In association with IROS'97, Sept. 1997.

[15] S. Hutchinson, G. D. Hager, and P. I. Corke. A Tutorial on Visual Servo Control. *IEEE Trans. on Robotics and Automation*, 12(5):651–670, Oct. 1996.

[16] I. Kamon, T. Flash, and S. Edelman. Learning to Grasp Using Visual Information. In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA'96)*, pages 2470–2476, 1996.

[17] S. Lanser and C. Zierl. Robuste Kalibrierung von CCD-Sensoren für autonome, mobile Systeme. In R. Dillmann, U. Rembold, and T. Lüth, editors, *Autonome Mobile Systeme*, Informatik aktuell, pages 172–181. Springer-Verlag, 1995.

[18] MVTec Software GmbH. *HALCON – The Software Solution for Machine Vision Applications*. http://www.mvtec.com/halcon/.

[19] S. Rahmann. Motion from curves for uncalibrated cameras. Master's thesis, TU München (extern Univ. Cambridge, GB), Oct. 1997.

[20] P. Sanz, A. del Pobil, J. Inesta, and G. Recatalà. Vision-Guided Grasping of Unknown Objects for Service Robots. In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA'98)*, pages 3018–3025, May 1998.

[21] M. Taylor, A. Blake, and A. Cox. Visually guided grasping in 3d. In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA'94)*, pages 761–766, 1994.

[22] M. Tonko, J. Schurmann, K. Schäfer, and H.-H. Nagel. Visually Servoed Gripping of a Used Car Battery. In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS'97)*, pages 49–54, Sept. 1997.

[23] M. Vincze and G. D. Hager, editors. *Workshop on Robust Vision for Vision-Based Control of Motion*. In association with ICRA'98, May 1998.

[24] P. Wunsch and G. Hirzinger. Real-Time Visual Tracking of 3D Objects with Dynamic Handling of Occlusion. In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA'97)*, pages 2868–2873, Apr. 1997.