A HIERARCHICAL 3D CIRCLE DETECTION ALGORITHM APPLIED IN A GRASPING SCENARIO

Emre Başeski, Dirk Kraft and Norbert Krüger
The Maersk Mc-Kinney Moller Institute, University of Southern Denmark
Campusvej 55 DK-5230 Odense M, Odense, Denmark

Keywords: 3D circle detection, Grasping, Stereo vision, Hierarchical representation.

Abstract: In this work, we address the problem of 3D circle detection in a hierarchical representation which contains 2D and 3D information in the form of multi-modal primitives and their perceptual organizations in terms of contours. Semantic reasoning on higher levels leads to hypotheses that then become verified on lower levels by feedback mechanisms. The effects of uncertainties in visually extracted 3D information can be minimized by detecting a shape in 2D and calculating its dimensions and location in 3D. Therefore, we use the fact that the perspective projection of a circle on the image plane is an ellipse and we create 3D circle hypotheses from 2D ellipses and the planes that they lie on. Afterwards, these hypotheses are verified in 2D, where the orientation and location information is more reliable than in 3D. For evaluation purposes, the algorithm is applied in a robotics application for grasping cylindrical objects.

1 INTRODUCTION

Circles are important structures in machine vision since they are a common feature for natural and human-made objects and they provide more information than points and lines about the pose of an object. In 3D vision, there are various ways of obtaining edge-like 3D entities (sparse stereo) from a stereo camera setup. Once the sparse stereo data is grouped with respect to a perceptual organization scheme, certain structures can be extracted from individual or combinations of these perceptual groups. Both, in dense and sparse stereo the correspondence finding phase in 3D reconstruction reduces the reliability of the information. Therefore, while detecting a certain structure like a 3D circle by using this kind of information, one needs to take into account the noise and uncertainty of the information.

The algorithms that are used to detect 3D circles can be grouped into three categories. The first category consists of *voting algorithms* like the Hough transform (Duda et al., 2000). Due to the size of the parameter space, voting algorithms require much more memory and computational power than other algorithms.

The second category contains *analytical algorithms* which use the geometric properties of circles (e.g., (Xavier et al., 2005)). For laser-range data, this kind of algorithms run fast and are robust because of the high-reliability of input data. Stereo vision on the other hand, introduces too many outliers and uncertainties that make the geometrical properties unstable.

The last category involves *fitting algorithms*. They are traditionally based on minimizing a cost function which depends on a distance function that measures errors between given points and the fitted circle (Jiang and Cheng, 2005; Chernov and Lesort, 2005; Shakarji, 1998). The fitting process can be done either in 3D or in 2D. If it is done in 2D, the optimal plane for the given points is calculated and the points are projected onto that plane. If the fitting is done in 3D, the minimization starts with an initial estimate and tries to converge to the optimal circle. However, to guarantee convergence, a good initialization is required. This can be done by starting with multiple initializations, which decreases the computational efficiency drastically. One can reduce the parameter space as in (Jiang and Cheng, 2005) but the noisy nature of stereo vision data decreases the probability of convergence. Therefore, although fitting in 2D is a decoupled solution (plane fitting and curve fitting are handled separately), it is more advantageous in terms of efficiency and reliability for noisy data.

In this article, an algorithm which is based on fitting in 2D is presented. Note that, the common practice for such approaches is using only 3D information

Baseski E., Kraft D. and Kruger N. (2009).
A HIERARCHICAL 3D CIRCLE DETECTION ALGORITHM APPLIED IN A GRASPING SCENARIO.
In *Proceedings of the Fourth International Conference on Computer Vision Theory and Applications*, pages 496-502
DOI: 10.5220/0001796004960502

and its projection onto 2D. The main specifity of our approach is, instead of using 3D information only, a hierarchical representation is used which represents visual information at different levels of semantic (e.g., 2D versus 3D) as well as different spatial complexity (local versus global). By that we obtain information with different levels reliability. Furthermore, there is a verification process, which is also performed using different levels in the representation hierarchy.

In this work, the hierarchical representation presented in (Krüger et al., 2004) is used. An example is presented in Figure 1 which shows what kind of information exists on different levels of the representation. At the lowest level of the hierarchy, there is the image with its pixel values (Figure 1(a)). At the second level, there exists the filtering results (Figure 1(b)) which give rise to the multi-modal 2D primitives at the third level (Figure 1(c)). At the third level, not only the 2D primitives but also 2D contours (Figure 1(d)) are available that are created using the perceptual organization scheme in (Pugeault et al., 2006). The last level contains 3D primitives and 3D contours (Figure 1(e-f)) created from 2D information of the input images.

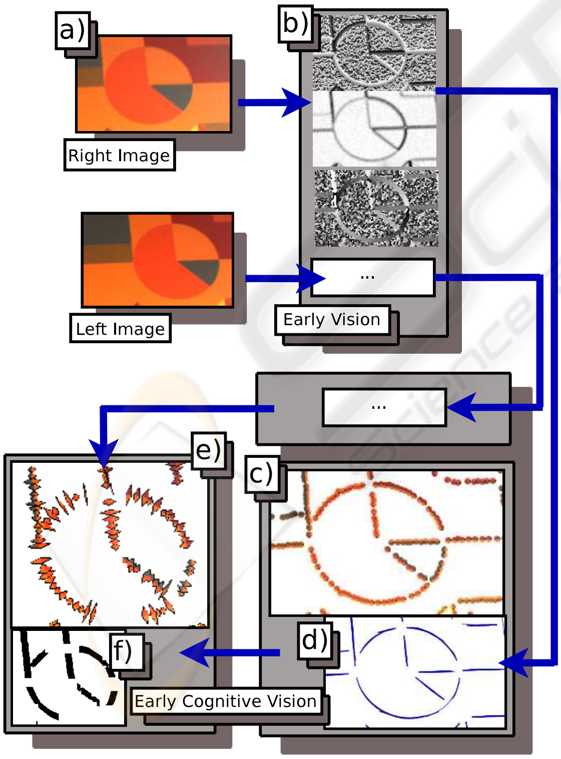


Figure 1: Different type of information that is available in the representation hierarchy **(a)** Original image **(b)** Filtering results **(c)** 2D primitives **(d)** 2D contours **(e)** 3D primitives **(f)** 3D contours.

Since the reliability and the amount of data decreases as the level of the representation hierarchy increases ((Pugeault et al., 2008)), lower levels should be used to verify the operations done in higher levels. For example, localization of a shape in 3D can be checked in 2D, once the perspective projection of the shape is known. Note that, there are more primitives and their orientation and location information is more reliable in 2D.

The key idea of our approach is to use different aspects of visual information according to their locality/globality, their semantic richness as well as their reliability in an efficient way. For example, it is known that 2D information is more reliable than 3D (since the stereo correspondence problem introduces additional errors) but 3D information is required to find 3D position, 3D orientation, and the radius of a circle. We make use of this trade-off, so that semantic reasoning on a higher level (e.g., 3D information leads to 3D hypotheses) becomes verified on a lower but more reliable level (e.g., 2D information) by feedback mechanisms. Another aspect is the locality of the data being used at the different steps of processing. By using semi-global features (i.e., 2D and 3D contours) for the computation of hypotheses we decrease computational time significantly. Since these hypotheses are verified using local features, the effect of additional errors inherent in contours are minimized. In this way, we make optimal use of the different levels of the hierarchical representation.

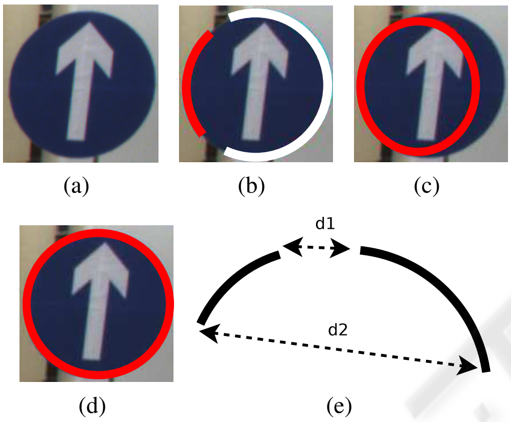
The rest of the article is organized as follows: In Section 2, the circle detection algorithm is introduced and some evaluation results in different scenarios with high variation in terms of circle sizes, 3D positions and orientation as well as number of circles and other factors such as occlusion are discussed. The experiments done on different objects in a grasping scenario where 3D dimension and location play an important role are presented in Section 3. We conclude with an evaluation of the algorithm based on these experiments.

2 CIRCLE DETECTION

The algorithm can be summarized in four steps as (1) ellipse hypotheses creation (Section 2.1), (2) verification of these hypotheses (Section 2.2), (3) creating circles by transferring the verified hypotheses to 3D (Section 2.3) and (4) verifying the created circles in 2D (Section 2.4).

2.1 Computing Ellipse Hypotheses

Because of the correspondence problem in the 3D reconstruction process, the information in 2D can not be transferred to 3D completely. Therefore, contours in 2D contain more primitives than corresponding 3D contours and a 2D contour can contain projections of more than one 3D contour. These facts are the motivation to use 2D contours to search for 2D ellipses in the image. Another important fact is that, a single 2D contour may not be big enough to compute the ellipse that we are searching for. In Figure 2(c) and (d), the ellipses fitted to contours in Figure 2(b) are shown. Since the red contour is not big enough, the ellipse fitted to that contour is not the desired one.



(a) (b) (c)

(d) (e)

Figure 2: **(a)** Original image **(b)** Two contours on the circle (One is red and the other is white) **(c)** Fitted ellipse to the red contour in (b) **(d)** Fitted ellipse to the white contour in (b) **(e)** Two curves can be merged if min(d1,d2) is small enough.

Having too small data sets for fitting is a common problem originating from perceptual organization. To overcome this difficulty, a merging mechanism has been proposed in (Ji and Haralick, 1999) which is based on proximity. Two curve segments are merged if the distance between their closest end points is smaller than a certain value (Figure 2(e)). The first step of the algorithm starts with merging the 2D contours by using the proximity criterion. This merging operation creates a new set of 2D contours which contain the old 2D contours and their combinations.

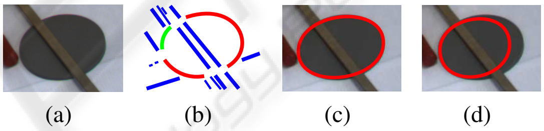
Let C_i be the set of all 3D contours whose projections on the image plane are contained in the 2D contour c_i. Then, for the 3D contour C_j, $P \cdot C_j \in c_i$ iff $C_j \in C_i$ (P is the projection matrix). Note that when two 2D contours are combined, the result is represented as c_k^+ and the set of 3D contours whose projections on the image plane are contained by the combination is represented as C_k^+.

The ellipse hypotheses e_k that the 3D circles are based on are created from the combined contours where c_k^+ is the 2D combined contour to which e_k is fitted. The ellipse fitting is done using the algorithm in (Pilu et al., 1996) which is an ellipse specific least-squares fitting method. The fitted ellipses are represented using the general ellipse equation given in (1).

$$ax^2 + 2bxy + cy^2 + 2dx + 2fy + g = 0 \qquad (1)$$

2.2 Verification of Ellipse Hypotheses

Since we use the merged contours, the fitting procedure creates a lot of false ellipses as well as true ones. Therefore, not all the fitted ellipses are really in the scene. A true ellipse is shown in Figure 3(c) which is fitted to the combination of the two red contours in Figure 3(b) and a false ellipse is shown in Figure 3(d) which is fitted to the combination of the bottom red and the green contour in Figure 3(b).



(a) (b) (c) (d)

Figure 3: **(a)** Input image **(b)** 2D contours **(c)** A true ellipse **(d)** A false ellipse.

The elimination of false ellipses is done by finding the *significance* (Lowe, 1987) of the ellipses. The percentage of covered length of e_i is calculated from all 2D primitives (represented by π_j) that satisfy the following equations:

$$\|\pi_j - e_i\| \leq \alpha_1 \qquad (2)$$

$$|\arctan(\frac{d}{dx} e_i|_{(\bar{x}_j,\bar{y}_j)}) - \theta_j| \leq \alpha_2 \qquad (3)$$

where α_1 and α_2 are thresholds, (2) is the distance between π_j and e_i, (3) is the difference between the slope of e_i at $(\bar{x}_j, \bar{y}_j)$ and the orientation of π_j (represented by θ_j) and $(\bar{x}_j, \bar{y}_j)$ is the coordinate of the closest point on e_i to π_j. If π_j satisfies (2) and (3), its patch size (the diameter of the patch covered by the primitive) is added to the total covered length of e_i. If the percentage of total covered length of e_i with respect to its perimeter is higher than a threshold, namely α_3, the ellipse is qualified as a true ellipse. The true ellipses for some scenes are shown in Figure 4 where $\alpha_1 = 1\,pixel$, $\alpha_2 = 10^\circ$ and $\alpha_3 = 60\%$.

2.3 Computing 3D Circle Hypotheses

Due to the fact that the perspective projection of a circle on the image plane is an ellipse, it is possible to

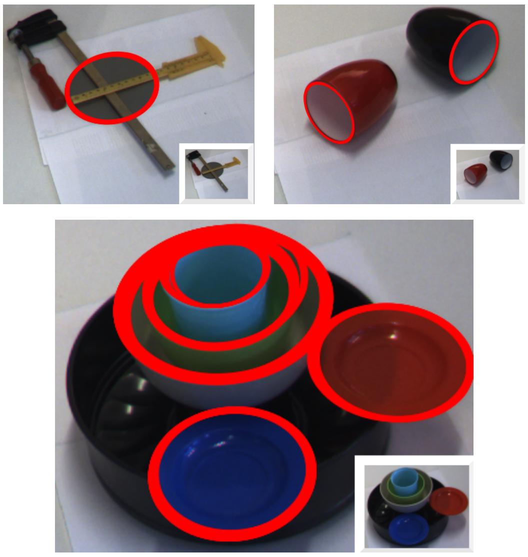


Figure 4: Some true ellipse examples.

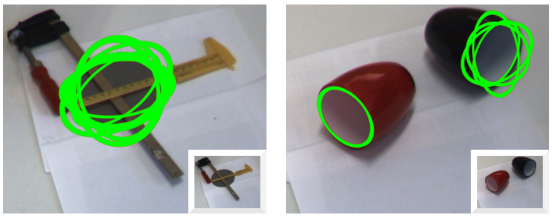
reconstruct the 3D circle, once the plane that the circle lies on is known. Therefore, at this point, to create 3D circles, the only further information we need is the plane p_i on which the circle that will be created from ellipse e_i lies. After calculating p_i, camera geometry can be used to find all the parameters of the 3D circle whose perspective projection is e_i. Since we know the 2D contour c_i^+ which gave rise to e_i, it is possible to use the 3D contours C_i^+ whose projections are contained by c_i^+ to fit p_i. This operation gives the normal vector of the 3D circle as it is parallel to the normal vector of p_i. What is missing for the 3D circle is the center and the radius in 3D.

To find the center and the radius of the circle, discrete points on e_i are multiplied with the pseudo-inverse of the projection matrix (P^+) to create rays, passing through the camera center and the discrete points of the ellipse. The intersections of these rays and the fitted plane p_i gives 3D points which are supposed to belong to the 3D circle. The center of mass of these 3D points gives the center of the 3D circle and this center is used to calculate the radius as the average distance of the 3D points to the center. Note that, the 3D circles calculated in the this step can be represented in parametric form as:

$$R\cos(t)\vec{u} + R\sin(t)(\vec{n} \times \vec{u}) + \vec{c} \tag{4}$$

where $\vec{u}$ is a unit vector from the center of the circle to any point on the circumference; R is the radius; $\vec{n}$ is a unit vector perpendicular to the plane and $\vec{c}$ is the center of the circle.

Some results are presented in Figure 5(a-b). Note that more than one combined contour can represent the same ellipse and they produce correct circles as well as false ones because of the 3D reconstruction uncertainties. The false circles are eliminated in the next step.



(a) (b)

Figure 5: **(a-b)** Projection of 3D circles on the image plane before verification.

2.4 Final Selection of Circle Hypotheses

As the last step, our aim is to find which 3D circle is the best for ellipses that have been represented by more than one combined contour. Let $\mathcal{E}_i$ be the set of ellipses that are similar. It is impossible for them to have the same curve parameters so we can measure the similarity between two ellipses as a cost function depending on the distance between their centers, the difference of their perimeters and orientations. The main idea of the last step is to calculate the *significance* of ellipses which are projections of circles created from the ellipses in set $\mathcal{E}_i$. We do the evaluation in 2D since the amount and the reliability of data in this dimension is higher than 3D. To find the ellipse which is the perspective projection of a 3D circle, we can pick 5 points of the circle on the image plane and use the implicit equation of the conic through 5 points as in (5).

$$\begin{vmatrix} x^2 & xy & y^2 & x & y & 1 \\ x_1^2 & x_1y_1 & y_1^2 & x_1 & y_1 & 1 \\ & & \cdots & & & \\ x_5^2 & x_5y_5 & y_5^2 & x_5 & y_5 & 1 \end{vmatrix} = 0 \tag{5}$$

The 5 points can be created from (4) for $t \in \{0, 80...320\}$. Equation 5 gives the generic equation of an ellipse as in (1). Therefore, we find the *significance* of these projected ellipses by using all 2D primitives π_j that satisfy Equations (2) and (3). For each set $\mathcal{E}_i$, only the one circle with the highest significance is kept. Some results are presented in Figure 6 and 7.

2.5 Problems

Although the algorithm is stable on tilted, partially occluded and cluttered circles, perceptual organization can create problems in case of good continuation between circular and non-circular parts. Figure 8(b)

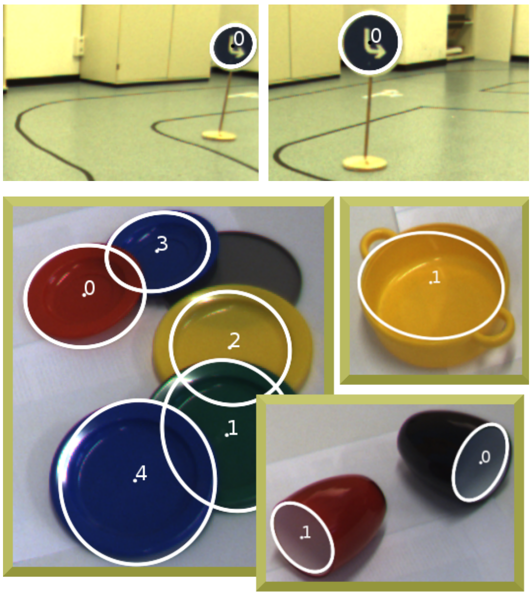


Figure 6: 3D circle detection results on different scenarios. (White ellipses are the projections of 3D circles onto the image plane).

illustrates a case, where the red 2D contour combines a circular and a non-circular part. In such cases, the remaining circular part (e.g., green contour in Figure 8(b)) may create a valid ellipse hypothesis but transferring this hypothesis to 3D is heavily dependent on the plane that is fitted to the 3D points and usually this situation leads to incorrect 3D circles as shown in Figure 8(c).

3 APPLICATION IN A GRASPING SCENARIO

The algorithm described in the previous section is applied in a robot grasping application. In this section we describe the setup and use of this application to evaluate the circle detection.

3.1 System Description

The robotic system used consist of a six degree of freedom industrial robot (Stäubli RX-60B), a two finger parallel gripper (Schunk PG 70) and a Point Grey BumbleBee2 stereo camera (see Figure 9(a)). The camera is calibrated relative to the robot coordinate system. Therefore the output of the above algorithm can be directly used for the computation of the grasping position.

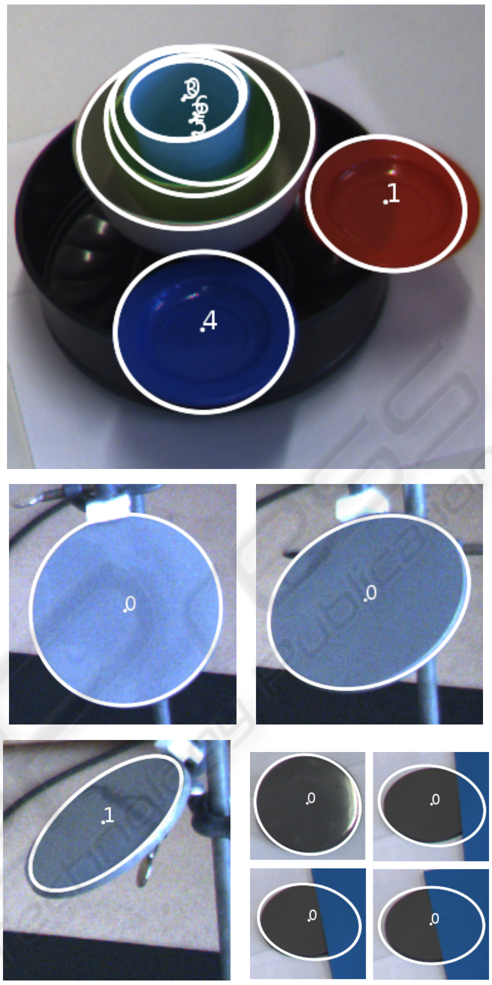


Figure 7: 3D circle detection results for multiple objects, different orientation and occlusion. (White ellipses are the projections of 3D circles onto the image plane).

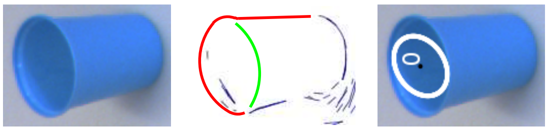
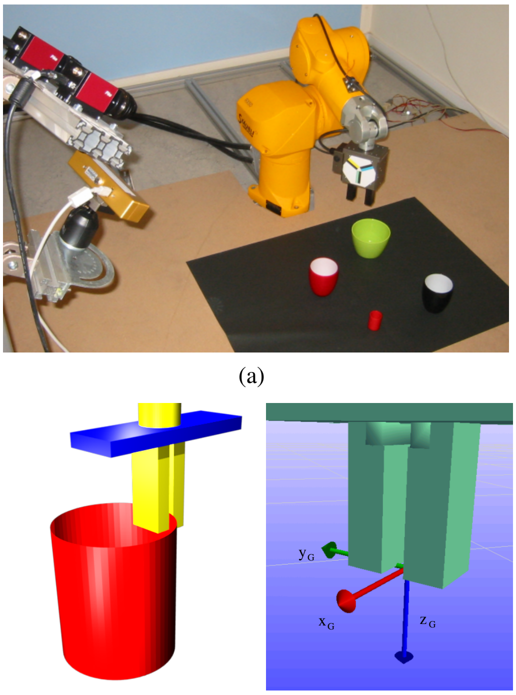


Figure 8: **(a)** Original image **(b)** 2D contours corresponding to (a) **(c)** Detected 3D circle.

3.2 Grasp Definition

For this work we selected one of the grasps defined in the grasping application to evaluate the quality of the circle detection. The cylindrical object is grasped on its brim (see Figure 9(b)). The position of the grasp is expressed similar to the parametric form in (4). From this observation directly follows that there is actually not one possible grasp, but a one dimensional manifold of grasps (varying the grasp position around the



(a)

(b)

(c)

Figure 9: (a) Robot system consisting of six degree of freedom industrial robot, two finger gripper and two stereo camera systems (The lower camera systems was used for this work). (b) Grasp at the brim of the cylindrical object. (c) Gripper coordinate system.

circumference of the circle). Additionally the grasping depth h can be chosen according to the requirements of the scene. The position p of the grasper can therefore be defined as:

$$\vec{p} \quad = \quad R\cos(t)\vec{u} + R\sin(t)(\vec{n}\times\vec{u}) + \vec{c} - \vec{n}h \; . \quad (6)$$

Figure 9(c) shows the position and orientation of the grasper coordinate system defined at the end of the fingers. The grasper needs to be aligned in the following way: $\vec{z_G} = -\vec{n}$ and $\vec{y_G} = \cos(t)\vec{u} + \sin(t)(\vec{n}\times\vec{u})$. While the gripper opening can be defined as $d = \min(2R, d_{max})$.

3.3 Evaluation

Figure 10 shows a number of scenarios where the gripper is moved to the grasping position computed based on the circle information ($h = 2\,cm$, t was used in a standard configuration except when this would have lead to a collision). For the set of experiments shown, the number of true positives (a circle that exists in the scene is detected) is 35, the number of false negatives (a circle that exists in the scene is not detected) is 1 and the number of false positives (a circle is detected that is not present in the scene) is 13. As a conclusion, 97.2% of the circles present in the scene have been detected and out of all detected circles (true positives and false positives), 72.9% of them correspond to the circles present in the scene. Note that, the false positives occur for relatively big circles where the numerical stability decreases. On the other hand, using the saliency measure (which is high for true positives) of the found circles, the true positives have higher chance to be choosen for grasping. Also, the different setups show that our system is able to cope with different levels of complexity.

4 CONCLUSIONS

We have discussed a 3D circle detection algorithm which makes use of different aspects of 2D and 3D information for hypothesis generation and verification. To be able to cope with the uncertainties of sparse stereo data, 3D circles are localized in 3D by considering 2D hypotheses and verified in 2D, where the information is more reliable. The potential of the approach has been shown on a grasping application for different scenarios. As a future work, the problem of combining circular and non-circular parts will be handled by splitting 2D contours with respect to junctions and 3D structure of the contour.

ACKNOWLEDGEMENTS

The work described in this paper was conducted within the EU Cognitive Systems project PACO-PLUS (IST-FP6-IP-027657) funded by the European Commission.

REFERENCES

Chernov, N. and Lesort, C. (2005). Least Squares Fitting of Circles. *J. Math. Imaging Vis.*, 23(3):239–252.

Duda, R. O., Hart, P. E., and Stork, D. G. (2000). *Pattern Classification*. Wiley-Interscience Publication.

Ji, Q. and Haralick, R. M. (1999). A Statistically Efficient Method for Ellipse Detection. In *ICIP (2)*, pages 730–734.

Jiang, X. and Cheng, D.-C. (2005). Fitting of 3D Circles and Ellipses Using a Parameter Decomposition Approach. In *3DIM '05: Proceedings of the Fifth International Conference on 3-D Digital Imaging and Modeling*, pages 103–109. IEEE Computer Society.

Krüger, N., Lappe, M., and Wörgötter, F. (2004). Biologically Motivated Multi-modal Processing of Visual Primitives. *The Interdisciplinary Journal of Ar-*

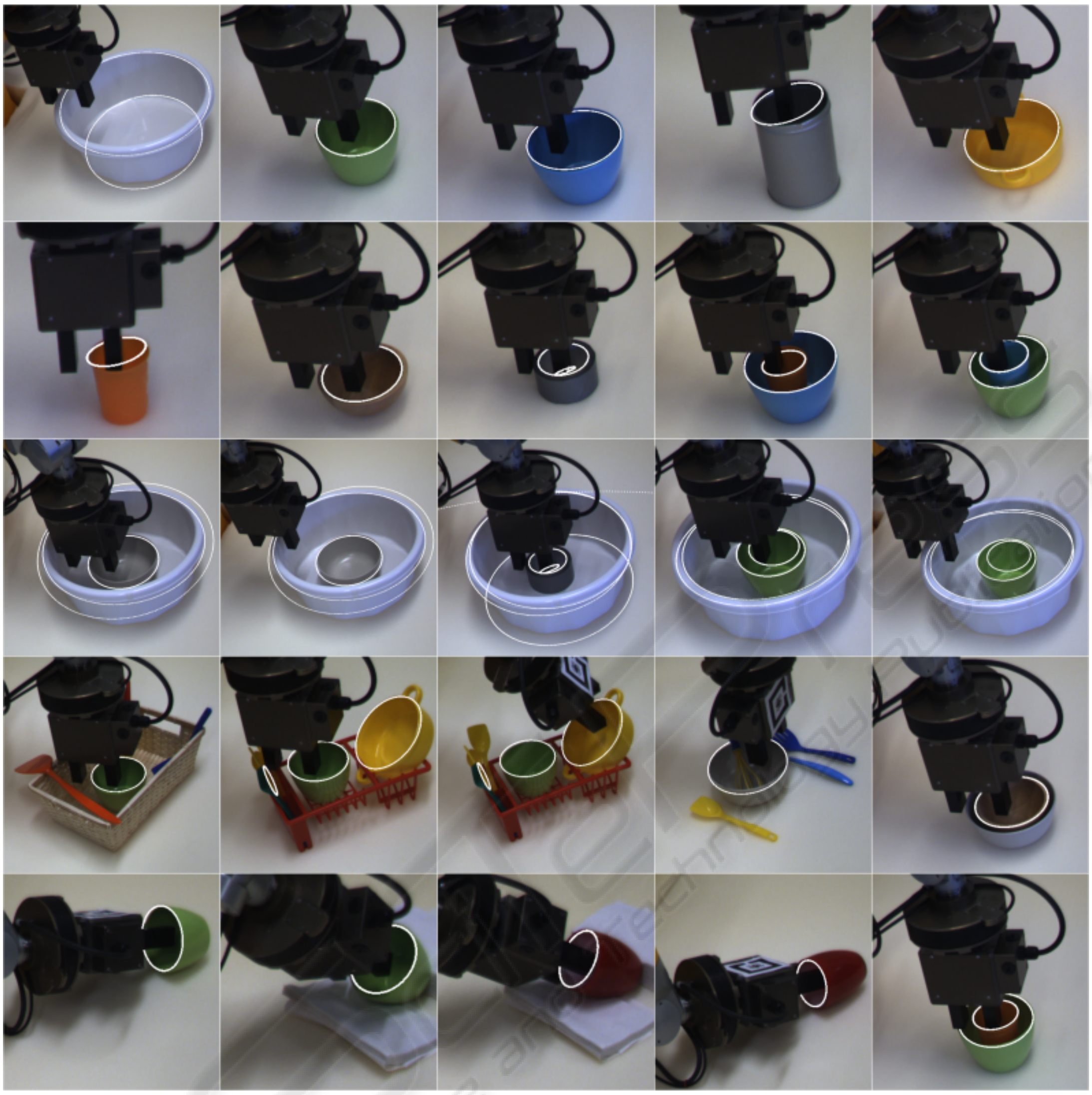


Figure 10: Detected circles and applied grasps. The circles were drawn into the images and the occluded parts were corrected afterward to improve the readers scene understanding. The scenes are of different complexity, starting out with single objects, going to objects included in each other, multiple (and more complex) objects and finally tilted single objects.

tificial Intelligence and the Simulation of Behaviour, 1(5):417–428.

Lowe, D. G. (1987). Three-Dimensional Object Recognition from Single Two-Dimensional Images. *Artificial Intelligence*, 31(3):355–395.

Pilu, M., Fitzgibbon, A., and Fisher, R. (1996). Ellipse-Specific Direct Least-Square Fitting. In *In Proc. IEEE ICIP*.

Pugeault, N., Kalkan, S., Başeski, E., Wörgötter, F., and Krüger, N. (2008). Reconstruction Uncertainty and 3D Relations. In *Proceedings of Int. Conf. on Computer Vision Theory and Applications (VISAPP'08)*.

Pugeault, N., Wörgötter, F., and Krüger, N. (2006). Multi-modal Scene Reconstruction Using Perceptual Grouping Constraints. In *Proceedings of the IEEE Workshop on Perceptual Organization in Computer Vision (in conjunction with CVPR'06)*.

Shakarji, C. (1998). Least-Squares Fitting Algorithms of the NIST Algorithm Testing System. *Res. Nat. Inst. Stand. Techn.*, 103:633–641.

Xavier, J., Pacheco, M., Castro, D., Ruano, A., and Nunes, U. (2005). Fast Line, Arc/Circle and Leg Detection from Laser Scan Data in a Player Driver. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 3930–3935.