

# Towards Service Robots for the Human Environment: the Robutler

Ulrich Hillenbrand, Christian Ott, Bernhard Brunner, Christoph Borst, and Gerd Hirzinger

Institute of Robotics and Mechatronics

German Aerospace Center (DLR), 82234 Wessling, Germany

*Ulrich.Hillenbrand@dlr.de*

**Abstract**—We present an experimental service robot, a mobile hand-arm system with anthropomorphic features as well as some capability for autonomous behavior. The system integrates a mobile platform, an arm optimized for light weight, and an articulated dexterous four-finger hand with basic navigation skills, real-time scene analysis by stereo vision, compliant motion control, and an intuitive man-machine interface. It thus comprises many of the key features required to realize the vision of a robotic servant that acts and interacts in the human living environment. The task we consider here is manipulation of bottles and glasses for preparing and serving drinks. The purpose of this article is to provide a comprehensive view of the whole system, revealing the synergies between its components that lead to real-world performance within the addressed problem domain.

Published in Proceedings *Mechatronics & Robotics – MechRob 2004*, pp. 1497-1502.

## I. INTRODUCTION

In the aging societies of the industrialized world, there will be an increasing demand for intelligent machines performing services in interaction or collaboration with humans. These machines will share with us our living environment and will have to perform everyday tasks in human-like manners.

As a consequence, in the field of service robotics, systems are needed that can handle everyday objects in unconstraint environments, in much the same way as they are handled by humans. Interaction with changing and, hence, only partially known environments requires intelligent sensor-data processing and advanced control architectures. Addressing this grand challenge for robotics, we present a mobile hand-arm system with anthropomorphic features as well as some capability for autonomous behavior. The system integrates, on the hardware side, a mobile platform, an arm optimized for light weight, and an articulated dexterous four-finger hand with, on the software side, basic navigation skills, real-time scene analysis by stereo vision, compliant motion control, and an intuitive man-machine interface; see Figure 1. We have called this system the *Robutler*.

In the present scenario, the world of the *Robutler* consists of various bottles and glasses that are freely arranged on a table. Its task is to prepare and offer drinks. Initially, the *Robutler* locates the table and navigates towards it, using a laser range scanner mounted on the mobile platform. Then an interpretation of the table-top scene is computed from the images taken by three on-board cameras. Object identities and poses are estimated by matching empirically learned models



Fig. 1. The *Robutler*: a mobile hand-arm system with stereo cameras and laser scanner. The arm shown is the second generation of DLR's Light-Weight Robots.

against 3D-point data that have been obtained from stereo processing. Different types of bottles and glasses are located and distinguished under varying conditions of lighting. Once the scene is understood, the *Robutler* autonomously executes a sequence of actions. For instance, it may unscrew the cap from a bottle, grasp the bottle, and pour a drink into a glass.

A hierarchical command interface enables the user to instruct the *Robutler* at various levels of abstraction. In particular, the interface provides graphical representations of the objects as detected in the scene, displayed in a virtual-reality environment. It thus allows the user to issue high-level commands about actions to take w.r.t. the objects in an intuitive manner. Alternatively, the *Robutler* can be instructed via a speech interface.

We consider the present experimental system a significant step towards designing service robots usable in human environments.



Fig. 2. The third generation of the DLR Light-Weight Arm.

## II. THE ARM

Service robots need to operate in the human environment or even cooperate with humans. Therefore, some basic requirements have to be met by the manipulators, making them quite distinct from industrial hardware. Service robotic applications require light-weight arms for safety reasons and human-friendly interaction, as well as to support mobility. Interaction with not fully known environments demands compliant arms and fingers, controlled by information extracted from many different sensors. To integrate multisensory components in arms and hands, sophisticated mechatronic concepts and a flexible control architecture are needed.

Over the last years, our focus on service robotics was driven by strong considerations, how to push robotic technologies for real-world applications. The design philosophy of DLR's light-weight robots [1] has been to realize a type of manipulator similar to the kinematic redundancy and dynamics of the human arm, i.e., with 7 degrees of freedom (DOFs), a load-to-weight ratio of better than 1:2, and a high dynamic performance; see Figure 2. All joints have position and torque sensors. In particular, measuring joint torques supports more accurate impedance control.

## III. THE HAND

For humans, hands and arms are universal actuators to interact with their environment. Grasping, carrying, and manipulating a wide variety of objects, using tools, catching and throwing things are basic abilities that we need them for. As robots start to serve and collaborate with man, it is natural to equip robots with artificial hands.

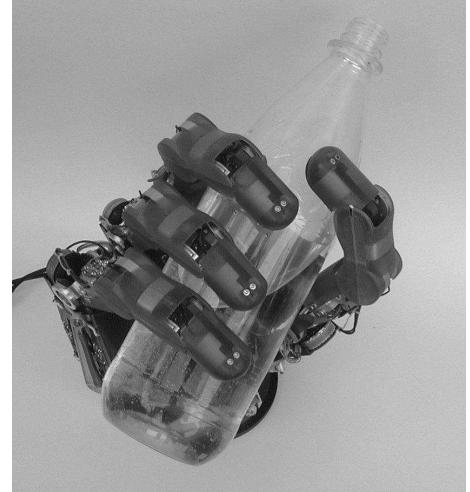


Fig. 3. DLR Hand II - power grasp.

The objects we want to manipulate, bottles and glasses, are challenging: glass can break and drinks can be spilled. A sensitive and compliant manipulator is therefore required.

The DLR hand features 4 fingers with anthropomorphic kinematics: 4 joints per finger with a collective 3 DOFs (distal joints coupled). An additional DOF resides in the palm for adaptation to power or precision grasps; see Figure 3. Joint control is accomplished with position, speed, and torque sensors for each finger DOF. High-level control can be driven by signals from a 6-DOF force-torque sensor in each finger tip.

In the case of precision grasps, usual finger design only allows for point contacts with the object at the finger tips. The fingers on the DLR hand, however, can bend backwards to create a much more robust line contact at the distal finger links (pinch grasp); see Figure 8.

## IV. THE VISUAL SYSTEM

The task of vision for a service robot is to deliver an interpretation of the scene that contains both geometric and semantic information. In other words, the *Robutler* has to know *where* objects are placed and *what* can be done with them. Clearly, the semantics of an object is not generally conceivable from vision alone. This kind of knowledge can presently be only attached to objects in a database. It is thus natural for a service robot to work with an a-priori known set of objects.

In future applications of service robots, however, it will also be necessary to manipulate objects which are a-priori not known. The basic action to take then is to grasp the object. We hence also consider the visual analysis of unknown objects. In section V, we will turn to the problem of grasp planning.

The objects we are dealing with, bottles and glasses, are challenging not only for manipulation, but also from the visual point of view. The reason is that visual data of transparent objects are generally more ambiguous than of opaque, colored or textured objects.

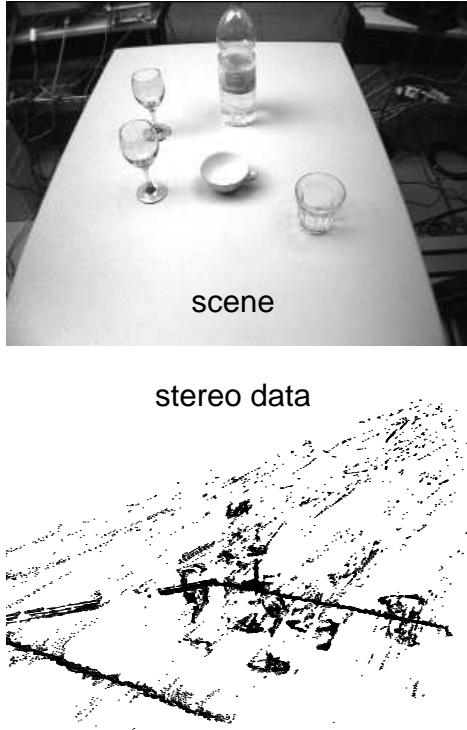


Fig. 4. Example scene and corresponding stereo data. The scene features a bottle, two wine glasses, a water glass, and a cup on a table, as seen in the top image taken by one of the cameras. The bottom image shows the 3D-data points obtained from stereo processing, viewed from the close left table corner. The objects and the table are mainly represented by contour points. Note the large amount of background data that arise as artifacts from stereo processing under such difficult conditions (transparent objects). This data set comprises 21,043 points.

### A. Stereo Processing

We use a three-camera system (Digiclops, Point Grey Research Inc.) to perform stereo processing with a horizontal and a vertical stereo pair (baseline 10 cm). Each image has a resolution of  $640 \times 480$  pixels. The algorithm employed is a straightforward local correspondence search by minimizing the sum of absolute differences over square patches of the LoG-filtered (Laplacian-of-Gaussian filter) images [2]. We have set a high threshold for valid regions of the LoG-filtered images, such that only regions of high scene contrast are taken into account. As a result, mainly surface creases, sharp bends, and depth discontinuities contribute to further processing. The output from stereo processing thus is a sparse representation of the scene by rather few 3D-data points (around 20,000), outlining the objects on the table; see Figure 4.

### B. Scene Interpretation

The sparse scene representation that we obtain from stereo processing has two major advantages for the processing steps to follow: it keeps the computational load low and it allows for a reliable scene interpretation by simple methods adapted from fuzzy-set and probability theory; see [3] for details. It turns out that objects standing on a weakly textured plane,

like our table, are visually well described by a few 3D-data points restricted to high-contrast regions of the scene.

Let  $\mu_i(\omega, d) \in [0, 1]$  be the probability for a data point at position  $d \in \mathbb{R}^3$  in data space to arise from object  $i$  with pose parameters  $\omega$ . These functions can be empirically learned by sampling histograms of the stereo data produced by the objects of interest. In fuzzy-set theory,  $\mu_i(\omega, d)$  would be called the ‘membership function for data points of object  $i$  with pose  $\omega$ ’. Traditional fuzzy logic, however, is independent of any probabilistic interpretation of membership values.

The quality of match of  $n$  object models  $\{\mu_1(\omega, d), \mu_2(\omega, d), \dots, \mu_n(\omega, d)\}$  to  $N$  data points  $\{d_1, d_2, \dots, d_N\}$ , or, in fuzzy-set slang, the degree of membership of the data to the objects, is given by

$$\begin{aligned} M(\omega_1, \omega_2, \dots, \omega_n, d_1, d_2, \dots, d_N) \\ := \sum_{i=1}^n M_i(\omega_i, d_1, d_2, \dots, d_N) \\ := \sum_{i=1}^n \sum_{j=1}^N \mu_i(\omega_i, d_j), \end{aligned} \quad (1)$$

where  $\omega_1, \omega_2, \dots, \omega_n$  are the objects’ pose parameters. A scene interpretation is obtained by optimizing the function  $M(\omega_1, \omega_2, \dots, \omega_n, d_1, d_2, \dots, d_N)$  with respect to the pose parameters of the  $n$  objects. The pose parameters are constrained to take consistent values, that is, objects cannot intersect each other. Moreover, an object  $i$  is included in a scene interpretation, only if its contribution  $M_i(\omega_i, d_1, d_2, \dots, d_N)$  to the best match exceeds some object-specific threshold.

For the case of our table-top scenes, the pose parameters  $\omega_i$  of an object can take values in intervals  $[a_1, a_2] \times [b_1, b_2] \times [0, 2\pi)$  for translation on the table and rotation around a table-orthogonal axis. Rotationally symmetric objects have poses only in  $[a_1, a_2] \times [b_1, b_2]$ . Search and optimization across all translations  $(x, y) \in [a_1, a_2] \times [b_1, b_2]$  is efficiently implemented by a fuzzy variant of the generalized Hough transform [4], [5], [6] with a single template  $\mu_i(x, y, d)$ . We have quantized the space of translations at intervals of 2 mm in both dimensions. The achieved translational accuracy of estimated object positions is usually within 2 quanta, i.e., 4 mm, up to distances of 1 m from the cameras.

For different rotational views  $\phi \in [0, 2\pi)$  of object  $i$ , various templates  $\mu_i(x, y, \phi, d)$  for discrete viewing angles  $\phi \in \{\phi_1, \phi_2, \dots\}$  have to be employed. The orientation  $\Phi$  of an object  $i$  at table-top position  $(x, y)$  can then be estimated by weighted averaging over the template orientations, that is,

$$\begin{aligned} \Phi = \arg \left[ e^{i\phi_{k-1}} \sum_{j=1}^N \mu_i(x, y, \phi_{k-1}, d_j) \right. \\ + e^{i\phi_k} \sum_{j=1}^N \mu_i(x, y, \phi_k, d_j) \\ \left. + e^{i\phi_{k+1}} \sum_{j=1}^N \mu_i(x, y, \phi_{k+1}, d_j) \right]. \end{aligned} \quad (2)$$

Here  $\mu_i(x, y, \phi_k, d)$  is assumed to be the best-matching template from the set  $\{\mu_i(x, y, \phi_1, d), \mu_i(x, y, \phi_2, d), \dots\}$ , and  $\phi_{k-1}$  and  $\phi_{k+1}$  are the two cyclic neighbors of orientation  $\phi_k$ . Provided the best-matching template is indeed the one with closest orientation, the orientation-estimate error will always be less than  $\Delta\phi/2$  for an angular increment  $\Delta\phi = |\phi_1 - \phi_2| = |\phi_2 - \phi_3| = \dots$  between templates.

An example of a visual scene interpretation is given in Figure 5. The scene features a bottle, two wine glasses, one partially occluding the other, a water glass, and a cup, as seen in Figure 4. The cup is included as an outlier that is not sought, i.e., the system has no model of cups. As shown in the figure, the bottle, wine glasses, and water glass are correctly located and identified, and the system is not confused by the cup. The cup may be marked as an unknown obstacle. Processing time, all the way through from image acquisition to scene interpretation, is for this scene around 5.7 seconds on a Pentium 4 CPU at 2.4 GHz under Linux.

We have found that the described method for scene interpretation can cope with challenging objects such as glasses under partial occlusion and over large variations of lighting (daylight, fluorescent light, spotlight).

### C. Surface Analysis

For an unknown object, the best we can do is to extract a description of its surface that is sufficiently informative for a grasp to be inferred. We currently follow two different strategies: first, by sampling surface normals from the stereo-data points; second, by fitting generalized cylinders to the stereo data; see Figure 6. When using surface normals as the description, at least two different views of an object are required to obtain information about possible grasps. On the other hand, fitting generalized cylinders implies strong symmetry assumptions about the objects. In both cases, a clustering algorithm is employed to obtain robust estimates of the parameters in the presence of outliers.

Note that, because the models applied here are very generic and abstract, the resulting surface description cannot be as accurate as for known objects. A grasp planner has to be able to cope with various sorts of imperfections of surface description; a compliant hand controller has to compensate for errors in estimated surface shape once the hand is in touch with the object.

## V. GRASP PLANNING

For a known object from a database, a set of grasps may be stored that is adequate for its semantics and use. For instance, a glass of water should be handled so as to prevent spilling. In particular, no real-time grasp planning is required. However, a highly desirable functionality for a service robot with a dexterous hand is autonomous grasping of unknown objects.

Any approach to grasp planning needs kinematic and geometric information about the hand and the objects to be grasped. In other words, it has to be known how the hand can move and how hand and objects are shaped.

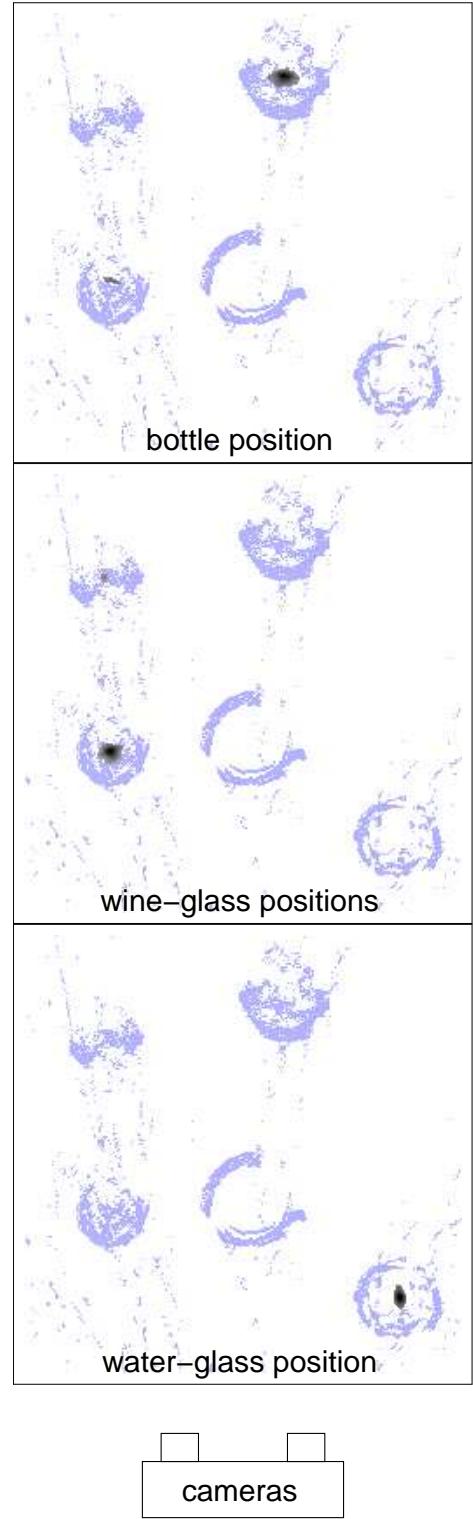


Fig. 5. The three panels show the distribution of evidence across the table (top view) for positions of bottles, wine glasses, and water glasses for the scene shown in Figure 4. A darker shade means higher evidence, white means no evidence. Superimposed are the 3D-data points. The system did not look for cups. The evidence  $M_i(x, y, d_1, d_2, \dots, d_N)$  for object  $i \in \{\text{bottle, wine glass, water glass}\}$  is calculated according to equation (1) for all discrete positions  $(x, y)$  on the table. The spurious evidence for a bottle near the position of one of the wine glasses (see top panel) is suppressed by the larger evidence for a wine glass at that position (see middle panel). In fact, all sought objects are correctly located and identified, while the cup is ignored.



Fig. 6. Surface analysis of an unknown object, a bottle of cleaning liquid. Shown are one of the camera images, a set of normal vectors computed from the corresponding stereo-data points, and generalized cylinders fitted to the stereo data.

The kinematic and geometric model of an artificial hand is usually available from CAD. A geometric description of unknown objects has to be established autonomously and in real time by 3D sensing like stereo vision (cf. section IV), but also structured light, laser scanners, shape from shading etc. may be used. For grasp planning, we need a polyhedral model that may have gaps and unconnected faces, and may also be imperfect in various other ways.

#### A. The Grasp-Planning Problem

Grasps for dexterous hands are reasonably divided in two main categories [7]: precision grasps for high manipulability where only the fingertips or distal finger links are in contact with the object, and power or enveloping grasps when high forces must be resisted or exerted and where the whole hand can be in contact with the object; see Figure 7.

The planning for these two types of grasps is very different. For precision grasps, one has to search for a few fingertip contacts (3 to 5) that allow for a stable grasp. For most of these contact sets, one can compute more than one valid hand configuration, as we show in [8]. Power grasps, on the other hand, are mainly determined by the geometry of the hand and the object. So one first tries to find a suitable hand configuration to “wrap” the fingers around the object and then calculates the resulting contacts.



Fig. 7. Power (left) and precision (right) grasp configuration of the DLR Four-Finger Hand.



Fig. 8. Planned pinch grasp.

#### B. The Grasp Planner for the DLR Hand

As stated above, it is possible to start precision-grasp planning from finding contacts and then calculate a valid hand configuration to realize a grasp. Moreover, we have observed that about 20% of all grasps computed from four randomly chosen contact points on a set of geometrical and real world objects (cube, sphere, cylinder, coffee mug, martini glass, etc.) result in force closure, which is the most common quality criterion for precision grasps [9], [10]. Therefore, we have decided to implement a random-sampling-based grasp planner. The algorithm can be summarized as follows, for details see [11].

- 1) Choose four contact points randomly on the object.
- 2) Calculate a kinematically valid hand configuration using an optimization approach [8]; see Figure 8 for a pinch-grasp example. Return to step 1) if there is none.
- 3) Perform a collision test between hand and object to see if geometric constraints are satisfied. Return to step 1) in case of a collision.
- 4) Compute a quality measure for the grasp.
- 5) Store the grasp in a list sorted by grasp quality and return to step 1), until there are enough grasps in the list or a given time limit is exceeded.

The most interesting step is the quality computation, step 4). We use a measure introduced by Ferrari and Canny [12] which physically means: The quality of a grasp is as good as the

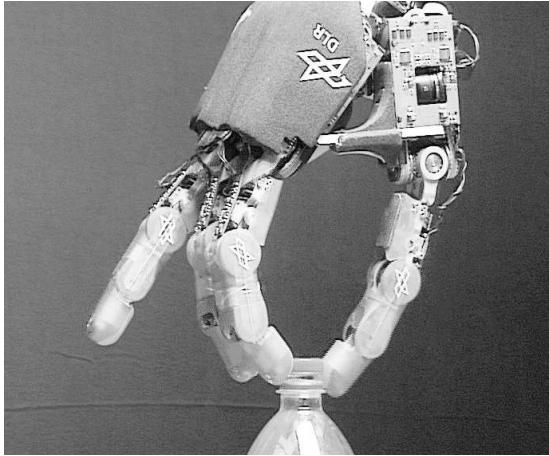


Fig. 9. The DLR Four-Finger Hand, unscrewing the cap from a bottle.

minimal wrench that breaks the grasp if all fingers can press with unit forces. This measure is physically more justified than the widely-used variant where the sum of all finger forces is limited to unit force. We have developed an incremental convex-hull construction algorithm to calculate the Ferrari-Canny measure. With this technique, we can compute and evaluate about 100 valid and force-closure grasp candidates in 20 to 60 seconds, depending on the complexity of the object, on a Pentium III CPU at 700 MHz under Linux. We then choose the best of these candidates as the final grasp.

## VI. THE TASK

We imitate a constrained but realistic domestic scenario. Various bottles and glasses are freely arranged on a table. It is the task of the *Robutler* to prepare and offer drinks. The user may order, via a graphical or a speech interface, a particular drink (water, wine, etc.). The *Robutler* will navigate towards the table and choose the appropriate bottle, unscrew its cap (if the bottle has a cap), grasp it, and pour into the appropriate glass; see Figures 9 and 10. It will then pick up the glass and offer it to the user. Alternatively, the user may command the *Robutler* to pour from a specified bottle into a specified glass and so on.

It is important to note that unscrewing the cap from a bottle is a true fine-manipulation task that requires both visual precision and compliant motion control of the three fingers involved; cf. Figure 9. Compliance makes the fingers adaptable to residual errors incurred from visual scene interpretation.

The command interface between the user and the robot is reduced to *what* has to be done. All the information on *how* the task is accomplished is hidden in the autonomous execution layer.

## VII. CONCLUSION

We have presented a mobile hand-arm system with anthropomorphic features, combined with visual perception, dexterous grasping skills, and an intuitive man-machine interface. The *Robutler* thus integrates on the levels of mechatronic

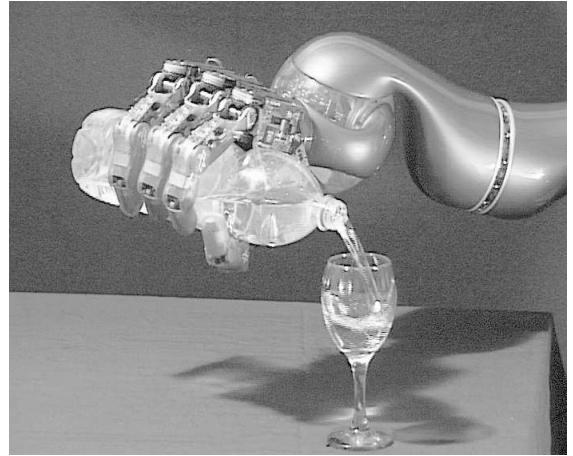


Fig. 10. The DLR Light-Weight Arm with Four-Finger Hand, pouring water into a glass.

design, control strategies, as well as machine intelligence some of the key components required for the next generation of service robots, that is, robots that will function in the human environment. We have validated robust system performance for a restricted but realistic task domain, preparing and offering drinks. In the future, we expect to expand the skills of the *Robutler* through recognition and manipulation of more general shape classes.

## REFERENCES

- [1] G. Hirzinger, A. Albu-Schäffer, M. Hähnle, I. Schäfer, and N. Sporer, "On a new generation of torque controlled light-weight robots," in *Proceedings of the IEEE Int. Conference on Robotics and Automation*, Seoul, Korea, May 2001, pp. 1087 – 1093.
- [2] M. Okutomi and T. Kanade, "A multiple-baseline stereo," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, pp. 353–363, 1993.
- [3] U. Hillenbrand, "On the relation between probabilistic inference and fuzzy sets in visual scene analysis," *Pattern Recognition Letters*, 2004, to be published.
- [4] D. H. Ballard, "Generalizing the Hough transform to detect arbitrary shapes," *Pattern Recognition*, vol. 13, pp. 111–122, 1981.
- [5] A. Samal and J. Edwards, "Generalized hough transform for natural shapes," *Pattern Recognition Letters*, vol. 18, pp. 473–480, 1997.
- [6] N. Bonnet, "An unsupervised generalized hough transform for natural shapes," *Pattern Recognition*, vol. 35, pp. 1193–1196, 2002.
- [7] M. R. Cutkosky and R. D. Howe, "Human grasp choice and robotic grasp analysis," in *Dextrous Robot Hands*, S. T. Venkataraman and T. Iberall, Eds. Springer Verlag, 1990, ch. 1.
- [8] C. Borst, M. Fischer, and G. Hirzinger, "Calculating hand configurations for precision and pinch grasps," in *Proc. IEEE/RSJ Conf. on Intelligent Robots and Systems*, EPFL, Lausanne, Switzerland, October 2002, pp. 1553 – 1559.
- [9] B. Mishra and N. Silver, "Some Discussion of Static Gripping and Its Stability," *Transactions on Systems, Man and Cybernetics*, vol. 19, no. 4, pp. 783 – 796, 1989.
- [10] A. Bicchi and V. Kumar, "Robotic grasping and contact: A review," in *Proceedings of the IEEE Int. Conference on Robotics and Automation*, San Francisco, California, April 2000, pp. 348 – 353.
- [11] C. Borst, M. Fischer, and G. Hirzinger, "A Fast and Robust Grasp Planner for Arbitrary 3D Objects," in *Proceedings of the IEEE Int. Conference on Robotics and Automation*, Detroit, Michigan, May 1999, pp. 1890–1896.
- [12] C. Ferrari and J. Canny, "Planning Optimal Grasps," in *Proceedings of the IEEE Int. Conference on Robotics and Automation*, Nice, France, May 1992, pp. 2290–2295.