Vision and touch for grasping

Rolf P. Würtz[1]

Institute for Neurocomputing, Ruhr-University Bochum, Germany,
http://www.neuroinformatik.ruhr-uni-bochum.de/PEOPLE/rolf/
E-mail: Rolf.Wuertz@neuroinformatik.ruhr-uni-bochum.de

Abstract. This paper introduces our one-armed stationary humanoid robot GripSee together with research projects carried out on this platform. The major goal is to have it analyze a table scene and manipulate the objects found. Gesture-guided pick-and-place This has already been implemented for simple cases without clutter. New objects can be learned under user assistance, and first work on the imitation of grip trajectories has been completed.

Object and gesture recognition are correspondence-based and use elastic graph matching. The extension to bunch graph matching has been very fruitful for face and gesture recognition, and a similar memory organization for aspects of objects is a subject of current research.

In order to overcome visual inaccuracies during grasping we have built our own type of dynamic tactile sensor. So far they are used for dynamics that try to optimize the symmetry of the contact distribution across the gripper. With the help of those dynamics the arm can be guided on an arbitrary trajectory with negligible force.

1 Introduction

In order to study models for perception and manipulation of objects we have set up the robot platform *GripSee*. The long-term goal of this research project is to enable the robot to analyze a table scene and to interact with the objects encountered in a reasonable way. This includes the recognition of known objects and their precise position and pose as well as the acquisition of knowledge about unknown objects. A further requirement is an intuitive mode of interaction to enable a human operator to give orders to the robot without using extra hardware and requiring only minimal training. As the most natural way of transferring information about object locations is pointing, we have chosen to implement a hand gesture interface for user interaction

This article reviews some of the steps we have taken towards that goal and first achievements. The whole project has only been possible through the cooperation of a large group, and I am indebted to Mark Becker, Efthimia Kefalea, Hartmut Loos, Eric Maël, Christoph von der Malsburg, Abdelkader Mechrouki, Mike Pagel, Gabi Peters, Peer Schmidt, Jochen Triesch, Jan Vorbrüggen, Jan Wieghardt, Laurenz Wiskott, and Stefan Zadel for their contributions.

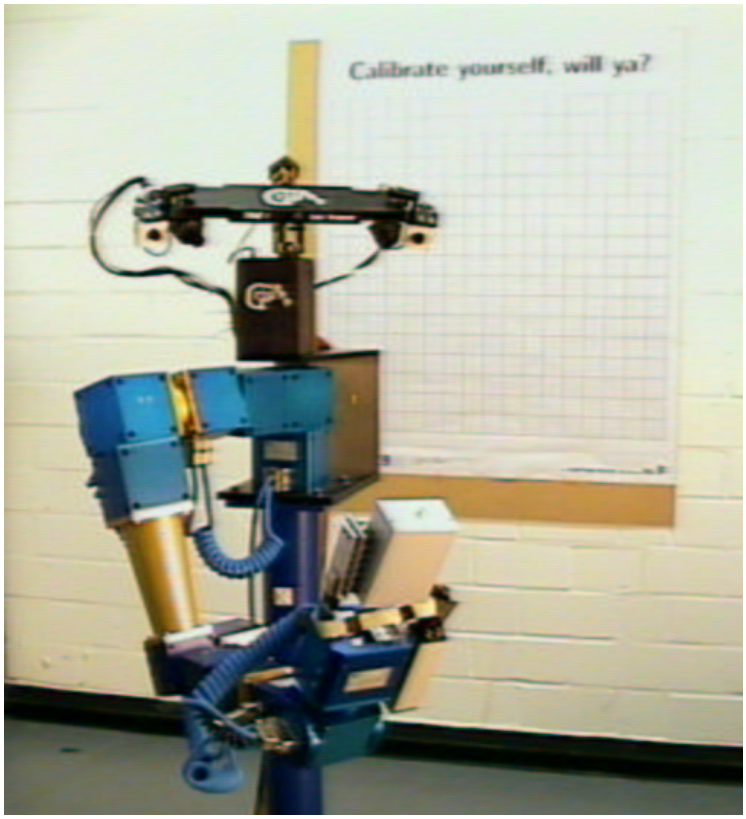

Fig. 1. GripSee's 3 DoF stereo camera head and 7 DoF manipulator.

2 Overall system

The robot hardware is shown in figure 1 and consists of the following components:

- A modular robot arm with seven degrees of freedom (DoF), kinematics similar to a human arm, and a parallel jaw gripper;
- a dual stereo camera head with three DoF (pan, tilt, and vergence) and a stereo basis of 30 cm for two camera pairs with different fields of view (horizontally 56° with color and 90° monochrome, respectively);
- a computer network composed of two Pentium PCs under QNX and a Sun UltraSPARC II workstation under Solaris.

Image acquisition is done by two color frame grabber boards controlled by one of the PCs, which also controls the camera head and performs real-time image processing for, e.g., hand tracking. The second PC controls the robot arm. Since image data has to be transferred between the processors, they are networked with FastEthernet to achieve sufficient throughput and low latencies.

Our software is based on the C++-library *FLAVOR* ("Flexible Library for Active Vision and Object Recognition") developed at our institute. FLAVOR comprises functionality for the administration of arbitrary images and other data types, libraries for image processing, object representation, graph matching, image segmentation, robotics, and interprocess communication [16].

3 Correspondence-based recognition

Recognition from visual data is required at two points in the desired functionality, namely the recognition of hand gestures and the recognition of objects.

In contrast to methods employing invariants or signatures our recognition methods rely on establishing a *correspondence map* between a stored aspect and the actual camera image. A big advantage of this type of recognition is that precise position information is recovered in addition to the object identity.

This position information in image coordinates can be used to measure 3-D position required for grasping by a combination of fixation and triangulation. For this to work the cameras must be calibrated relative to the position of the end effector. We have developed our own neural network-based method to do this, which is described in [10, 11].

3.1 Features

The difficulty of the correspondence problem depends on the choice of features. If grey values of pixels are taken as local features, there is a lot of ambiguity, i.e., many points from very different locations share the same pixel value without being correspondent. A possible remedy to that consists in combining local patches of pixels, which of course reduces this ambiguity. If this is done too extensively, i.e., if local features are influenced by a large area, the ambiguities disappear if identical images are used, but the features become more and more sensitive to distortions and changes in background.

As a compromise vectors of features at various scales (and orientations) can be used. One possibility is a wavelet transform based on complex-valued Gabor functions [6, 30, 31], with wavelets parameterized by their (two-dimensional) center frequency. At each image point, all wavelet components can be arranged into one feature vector, which is also referred to as a "jet". These features have turned out to be very useful for face and gesture recognition.

Depending on the application other feature vectors are more appropriate. The combination of different features has been systematically studied in [20]. For the object recognition described here, a combination of Mallat's multiscale edges [8] and object color has yielded good results. For the gesture recognition Gabor wavelets have been used together with color relative to a standard skin color. Generally, feature vectors seem to be better candidates for establishing correspondence maps than scalar features.

3.2 Elastic graph matching

Solving the correspondence problem from feature similarity alone is not possible, because the same feature may occur in an image more than once. Therefore, the relative position of features has to be used as well. One standard method for this is elastic graph matching: the stored views are represented by sparse graphs, which are vertex labeled with the feature vectors and edge labeled with the distance vector of the connected vertices. Matching is done by first optimizing the similarity of an undistorted copy of the graph in the input image and then optimizing the individual locations of the vertices. This results in a distorted graph whose vertices are at corresponding locations to the ones in the model graph. This procedure is described in full detail in [6, 28, 29].

3.3 Attention control

Conceptually, the solution of the correspondence problem includes segmentation of the scene – when a correct correspondence map is available, the background is eliminated as well. Practically, the matching algorithms become time consuming and error prone when the area covered by the object proper becomes smaller relative to the image size. Therefore, our recognition system usually employs some attention control to select areas that are more likely than others to contain an object. It is important that this be fast enough to allow a reaction by the system. Therefore, only simple cues such as color or motion are used.

A second step to simplify the recognition tasks is *fixation*, i.e. centering both cameras on an object point. This also has the advantage to minimize perspective distortions.

3.4 Bunch graphs

The major drawback of most correspondence-based recognition systems is that the computationally expensive procedure of creating a correspondence map must be done for *each* of the stored models. In the case of face recognition, this has been overcome by the concept of *bunch graphs* [28, 29]. The idea is that the database of models is arranged in such a way that corresponding graph nodes are already located at corresponding object points, e.g., a certain node lies on the left eye of all models. For large databases, this reduces the recognition time by orders of magnitudes. The resulting face recognition system has performed very well in the FERET test, an evaluation by an independent agency. It was one of two systems that underwent that test without requiring additional hand-crafted information such as the position of the eyes in the images to be analyzed. The performance on the difficult cases, where the images of persons were taken at widely different times was clearly better than the competitors' [15]. As a matter of fact, only one competitor underwent the test without requiring further information regarding the position of the eyes. This again illustrates how difficult the correspondence problem really is.

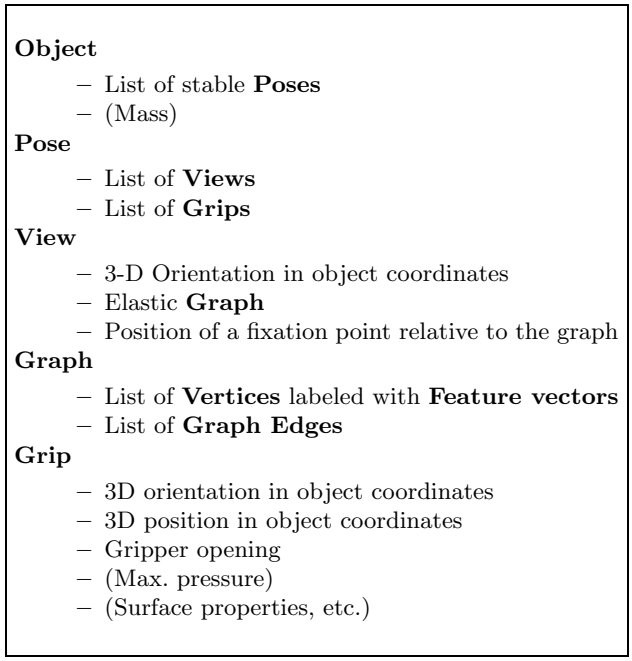

Fig. 2. Schema of our object representation. Items in parentheses are important object parameters which currently cannot be extracted from sensor data.

The bunch graph method exploits the fact that all faces are structurally very similar and correspondences inside the database make sense at least for a subset of salient points. For general object recognition this is much more difficult. Consequently, there is currently no suitable data format for storing many aspects of many objects. The organization of the database of all aspects of *one* object is subject of current research in our group [26, 14, 27, 13]. Another successful application lies in the recognition of hand gestures, where is has been used to make the recognition background invariant [24, 21, 23].

3.5 Object representation

An object representation suited for visually guided grasping has to integrate 2D visual features and 3D grip information about a known object, in order to apply a known grip when the situation requires it. Autonomous learning of the representation is highly desirable, therefore complicated constructs like CAD models are not considered. Rather, we adopt the view that visual recognition and application of a grip is mainly a recollection of what has been seen or done before, with the necessary slight modifications to adapt to the situation at hand.

The visual representation is view-based, i.e., for each different orientation of the object a set of visual features is stored, which are extracted from the left

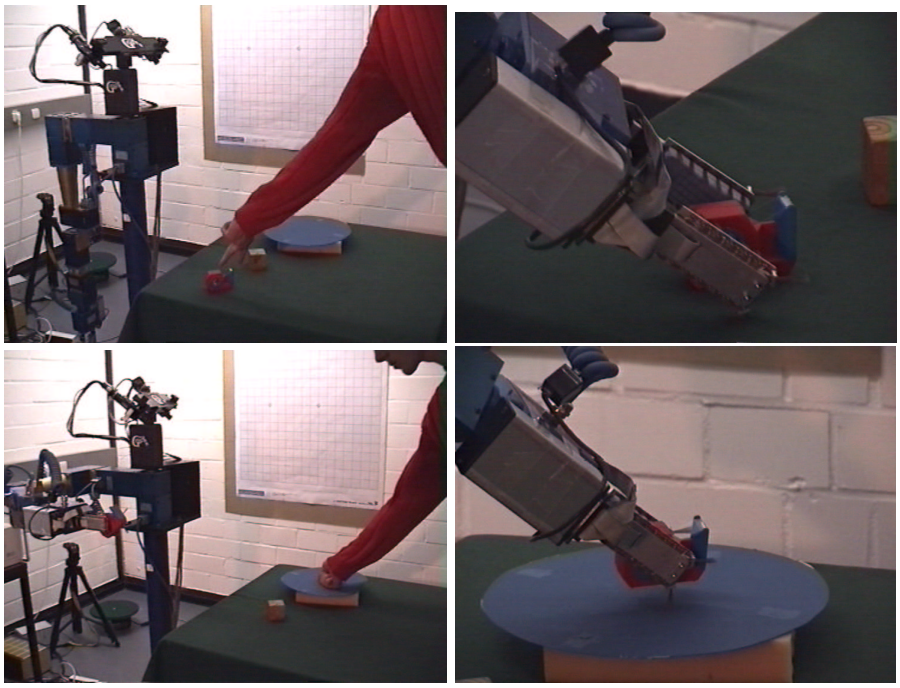

Fig. 3. Pick- and place behavior. The user points at one of the objects with a hand gesture (top left). The hand gesture is recognized and the closest object is picked up at the angle indicated by the gesture (top right). The robot waits for a new gesture (bottom left) and places the object at the indicated position (bottom right).

and right stereo images and grouped into a *model graph*, which preserves the topological relationships between the features. These model graphs are stored in a library for different objects with different orientations and are used to recognize known objects with a graph matching process, which is invariant under translation and scale. The grips are then associated with the views. A schema of our object representation can be found in figure 2.

4 Pick and place behavior

The visual recognition techniques outlined above can be applied to everyday objects as well as to hand gestures. Like all correspondence based methods they have the advantage to yield detailed position information in addition to object identity. This makes them well suited as subsystems for a humanoid robot that can analyze and manipulate a table scene under human control. Thus, we have successfully implemented a behavior module, which allows a user to point to one of the objects on the table and have the robot pick it up and place it at a desired position. So far it is assumed that the robot knows all the objects from their visual appearance and how to grasp them.

The system starts in a state, in which the robot stands in front of a table with various known objects and waits for a human hand to transmit instructions. The interaction with the human is initiated by moving a hand in the field of view of the robot. This movement is tracked using a fusion of skin color and motion cues. In order to point to an object the hand must stop, which is detected by the robot using the sudden vanishing of the motion cue during tracking.

Now a rough position of the hand is known, which is fixated by the camera head. After fixation, the type of hand gesture posture is determined by graph matching. This matching process also yields a refined estimate of the position of the hand's center, which is fixated. The actual identity of the hand gesture (fist, flat, etc.) can be assigned arbitrary additional information to be conveyed to the robot. We currently use it to code for the grasping angle relative to the table (steep grip vs. low grip, etc.).

It is expected that the hand now points at an object, thus the fixation point is lowered by a fixed amount of about 10 cm. Now, a simple localization algorithm detects the object, which is then fixated. Now the object recognition module determines the type of object, its exact position and size in both images and its orientation on the table.

Like in the posture recognition the cameras fixate on a central point of the recognized object (which is a feature known for each each view of each object). This yields the most precise object position that can be expected. Obviously, this precision is crucial for reliable grasping.

Now a grip suitable for the object and according to what was demanded by the hand gesture is selected from the set of grips attached to the object and transformed into world coordinates. A trajectory is planned, arm and gripper are moved towards the object, and finally the object is grasped. Another trajectory transfers it to a default position conveniently located in front of the cameras for possible further inspection.

Then the field of view is again surveyed for the operator's hand to reappear. In this state, it should be pointing with a single defined gesture to the three-dimensional spot where the object is to be put down. The 3D position is again measured by the already described combination of fixation movements and gesture recognition. Then the arm moves the the object to the desired location and the gripper releases it.

5 User-assisted object learning

The construction of the object database is a time consuming process, which includes taking images of the object from all possible orientations and coding useful grips from different angles. A good part of our work is dedicated to automate this construction.

Learning of a new object (or more precisely, one stable pose of a new object) is initiated by putting the object onto the table and having the robot create the various views by moving the object around. This procedure has two serious difficulties. First, a good grip must already be known for the robot to manipulate

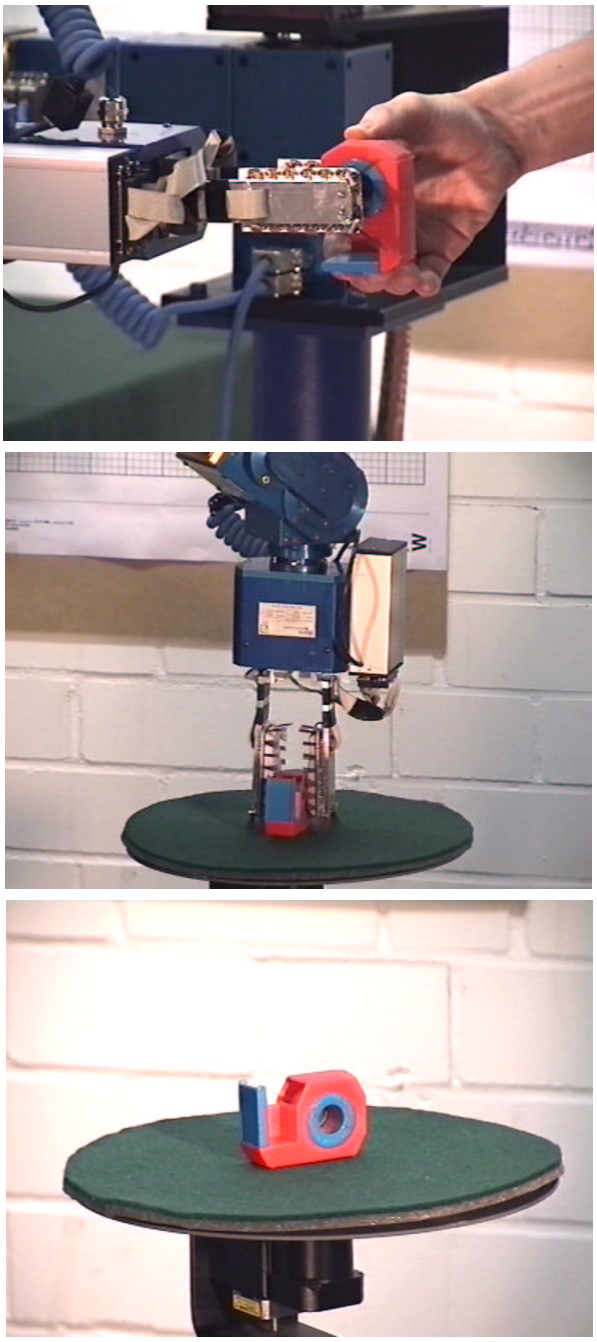

Fig. 4. Learning the representation of a new object. The object is placed into the gripper such that a stable grip is known (top). The robot puts the object onto a turntable (middle) and images of the various views are recorded without interference from the gripper (bottom).

the object in a predictable manner. Second, the actual orientation of the object should be known with good precision, because the error is likely to accumulate over the various views. The problem of learning new grips from scratch can only be solved by relying on tactile information. We have currently constructed and implemented tactile sensors on our gripper (see section 6), but the extraction of detailed object information is subject of future research.

In this situation, we have decided to solve both problems by what we call *user-assisted learning*. The general idea is that the acquisition of knowledge is as autonomous as possible, but a human operator still makes decisions about what is important and thus guides the process. Concretely, in the current case our learning procedure is as follows. The operator presents the novel object by putting it into the gripper (which has a defined position and orientation at that moment), in a position and orientation that are ideal for grasping. They thus define both a default grip and the object coordinate system. The robot closes the gripper, puts the object onto the center of a turntable, *fixates* on the grasping center, and takes a stereo image pair of the first object view. Then, the turntable is rotated by a specified increment and a second view is taken. This is repeated until a full circle of object views is acquired. After acquisition, all images are rotated around their center to compensate for the rotation associated with a combination of tilt and vergence.

After the images of the views are taken, they are converted into a collection of *labeled graphs*. In this learning step, it is assumed that the background is uniform (which is the case for the surface of the turntable) in order to avoid the necessity of complicated segmentation methods and to assure as clean graphs as possible in the representation (see [4] for details).

Two alternatives to using the turntable may be considered, namely having the robot place the object onto the table in all necessary orientations or holding it in the gripper while storing the visual information. Both have not been pursued so far in order to minimize the mechanical strain on the hardware. The second possibility, poses the additional problem of segmenting and subtracting the gripper itself from the image of the object. The quality of the object recognition depends rather critically on the quality of the model graphs. Our system therefore contains the possibility to modify the node positions by hand, but under good illumination the automatically created graphs are usually good enough.

6 Tactile sensors

In the course of the project it has become clear that visual information in a humanoid setup is not accurate enough for precise grasping. It should be enhanced with tactile information for fine tuning during the application of a grip. As none of the currently available sensors met our requirements for a robust system that was very sensitive for dynamic touch, a new type of sensor has been developed [18, 19]. In combination it complements two static elements (for x-

and y- direction, respectively) with 16 dynamic sensor elements on each jaw of the gripper.

As a first approach to behavior guided by tactile sensing we have implemented a method for guiding the robot arm manually on a desired trajectory. The signal from each sensor is binarized with a suitable threshold and triggers a predefined reflex movement, which is represented by a translational and a rotational movement vector relative to the gripper. The translation vector moves the contact position towards a target position, e.g., the center between both gripper jaws. This results in a radial vector field for the translation, where each direction is given by a vector pointing from the target position towards the sensor position. The rotation vector turns the gripper such as to maximize the number of sensor contacts, i.e., it attempts a parallel orientation of the gripper jaw to the hand's surface. The opening width of the gripper is controlled to keep loose contact with the hand. In case of contact with both gripper jaws it receives a small opening signal, and in case of one-sided contact a small closing signal.

This control can be used to guide the robot arm on a desired trajectory by putting a hand between the jaws. If adjusted suitably, the movement components of each sensor element add up to a motion which minimizes the asymmetry of the contact around the hand and thus follows the hand's movement through the configuration space. Given a simple static object with parallel grasping surfaces, the same strategy will eventually converge to a situation with a symmetric contact distribution around the desired grasping position on each gripper jaw. For grasping, this can be followed by pressure-controlled closing of the gripper. Turning this method into a good grasping strategy with minimal requirements on object shape and developing a systematic way to learn optimal vector fields are subjects of current research. Full details about the dynamics and the overall robot control can be found in [7].

7 Related work

A comprehensive study of the state of the art in robotic grasping is clearly beyond the scope of this paper. To put the work into perspective, I classify the methods used in terms of the possibilities provided in [3]. The control structure is hierarchical in the sense that the vision system estimates a 6D grasping position, and then a grasping trajectory is calculated. Possible grips are associated with object identities and part of the objects' description. The 3D position and the object's orientation on the table are derived from the recognition procedure, the remaining DoF of the grip orientation is derived from the user interaction. Different from the servoing systems in, e.g. [2], our system is currently "endpoint-open-loop", as no attempt is made to follow the end effector visually. A complete "endpoint-closed-loop" system like the one described in [5] would be desirable, but we will have to develop a suitable visual model of the end effector first that could accommodate the quite complicated occlusions arising at the critical moment of actual grasping. Tracking the manipulator with algorithms like in [9] is not enough in such situations. In order to circumvent these problems, we are

currently planning to correct the control errors resulting from visual inaccuracies using the tactile sensors.

Our system is designed to resemble the situation in human perception as closely as possible, which distinguishes it from the work done on industrial manipulators. The latter are usually non-redundant, and the cameras can be positioned according to the needs of the task at hand. Robot systems similar to our approach include [25] and [12]. The attention mechanisms and active depth estimate bear resemblance to the system described in [1].

8 Outlook

In the following I give an outlook on projects that are currently underway in order to enhance the robot's capabilities.

8.1 Imitation learning

In addition to learning the visual appearance of objects the learning of grips is important. As it is hard to define what constitutes a good grip an attractive method is to have the robot imitate trajectories performed by a human. We have measured the trajectories of a pair of thumb and index finger from a Gabor transform with specially adapted parameters [22] with encouraging results. Another project tries to estimate the arm position in whole-body gestures. First results have been published in [17].

8.2 Further problems

Among the problems to be tackled next is the interpretation of the output of the tactile sensors and the classification of situations such as a sliding object, a perfect grip, and an unstable grip from the time series displayed by the tactile sensors during grasping. A further task will be to learn a representation of 3D space in a way that is suitable for manipulation. Concretely, the robot body (and possibly other parts) must be represented so that collisions can be reliably avoided. The relative calibration of vision, touch and proprioception must be improved, and it is highly desirable to have it updated during normal operation. The optimization of grips from sensor data has already been mentioned. On the visual side the organization of the object database to make rapid recognition possible is the most pressing issue. Finally, the integration of the sensor information to a coherent percept of the environment and a good organization of the overall behavior must be pursued.

Acknowledgments. Financial support from the projects NEUROS (01 IN 504 E 9) and LOKI (01 IN 504 E 9) by the German Minister for Education and Research and from the RTN network MUHCI by the EU is gratefully acknowledged.

References

1. Sven J. Dickinson, Henrik I. Christensen, John K. Tsotsos, and Göran Olofsson. Active object recognition integrating attention and viewpoint control. *Computer Vision and Image Understanding*, 67:239–260, 1997.
2. G.D. Hager. A modular system for robust positioning using feedback from stereo vision. *IEEE Trans. Robotics and Automation*, 13(4):582–595, 1997.
3. Seth Hutchinson, Gregory D. Hager, and Peter I. Corke. A tutorial on visual servo control. *IEEE Transactions on Robotics and Automation*, 12(5):651–670, 1996.
4. Efthimia Kefalea. *Flexible Object Recognition for a Grasping Robot*. PhD thesis, Computer Science, Univ. of Bonn, Germany, March 1999.
5. Danica Kragić and Henrik I. Christensen. Cue integration for visual servoing. *IEEE Transactions on Robotics and Automation*, 17(1):18–27, 2001.
6. Martin Lades, Jan C. Vorbrüggen, Joachim Buhmann, Jörg Lange, Christoph von der Malsburg, Rolf P. Würtz, and Wolfgang Konen. Distortion invariant object recognition in the dynamic link architecture. *IEEE Transactions on Computers*, 42(3):300–311, 1993.
7. Eric Maël. *Adaptive and Flexible Robotics for Visual and Tactile Grasping*. PhD thesis, Physics Dept., Univ. of Bochum, Germany, December 2000.
8. S. Mallat and S. Zhong. Characterization of signals from multiscale edges. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14:710–732, 1992.
9. Kevin Nickels and Seth Hutchinson. Model-based tracking of complex articulated objects. *IEEE Transactions on Robotics and Automation*, 1997.
10. Mike Pagel, Eric Maël, and Christoph von der Malsburg. Self calibration of the fixation movement of a stereo camera head. *Autonomous Robots*, 5:355–367, 1998.
11. Mike Pagel, Eric Maël, and Christoph von der Malsburg. Self calibration of the fixation movement of a stereo camera head. *Machine Learning*, 31:169–186, 1998.
12. Josef Pauli. Learning to recognize and grasp objects. *Autonomous Robots*, 5(3/4):407–420, 1998.
13. Gabriele Peters. *Representation of 3D-Objects by 2D-Views*. PhD thesis, Technical Faculty, University of Bielefeld, Germany, 2001. In preparation.
14. Gabriele Peters and Christoph von der Malsburg. View Reconstruction by Linear Combinations of Sample Views. In submitted to: *International Conference on Knowledge Based Computer Systems (KBCS'2000), Mumbai, December 17 - 19, 2000*, 2000.
15. P. Jonathan Philips, Hyeonjoon Moon, Syed A. Rizvi, and Patrick J. Rauss. The FERET evaluation methodology for face-recognition algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10):1090–1104, 2000.
16. Michael Rinne, Michael Pötzsch, Christian Eckes, and Christoph von der Malsburg. Designing Objects for Computer Vision: The Backbone of the Library FLAVOR. Internal Report IRINI 99-08, Institut für Neuroinformatik, Ruhr-Universität Bochum, D-44780 Bochum, Germany, December 1999.
17. Achim Schäfer. Visuelle Erkennung von menschlichen Armbewegungen mit Unterstützung eines dynamischen Modells. Master's thesis, Physics Dept., Univ. of Bochum, Germany, February 2000.
18. Peer Schmidt. Aufbau taktiler Sensoren für eine Roboterhand. Master's thesis, Physics Dept., Univ. Bochum, Germany, December 1998.
19. Peer Schmidt, Eric Maël, and Rolf P. Würtz. A novel sensor for dynamic tactile information. *Robotics and Autonomous Systems*, 2000. In revision.

20. J. Triesch and C. Eckes. Object recognition with multiple feature types. In L. Niklasson, M. Bodén, and T. Ziemke, editors, *Proceedings of the ICANN'98, International Conference on Artificial Neural Networks*, pages 233–238. Springer, 1998.
21. J. Triesch and C. von der Malsburg. A gesture interface for robotics. In *FG'98, the Third International Conference on Automatic Face and Gesture Recognition*, pages 546–551. IEEE Computer Society Press, 1998.
22. J. Triesch, J. Wieghardt, C. v.d. Malsburg, and E. Maël. Towards imitation learning of grasping movements by an autonomous robot. In A. Braffort, R. Gherbi, S. Gibet, Richardson, and D. J., Teil, editors, *Gesture-Based Communication in Human-Computer Interaction, International Gesture Workshop, GW'99, Gif-sur-Yvette, France, March 17-19, 1999 Proceedings*, volume 17 of *Lecture Notes in Computer Science*. Springer-Verlag, 1999. ISBN 3-540-66935-3.
23. Jochen Triesch. *Vision-Based Robotic Gesture Recognition*. PhD thesis, Physics Dept., Univ. of Bochum, Germany, 1999.
24. Jochen Triesch and Christoph von der Malsburg. Robust classification of hand postures against complex backgrounds. In *Proceedings of the Second International Conference on Automatic Face and Gesture Recognition*, pages 170–175. IEEE Computer Society Press, 1996.
25. J.K. Tsotsos, G. Verghese, S. Dickinson, M. Jenkin, A. Jepson, E. Milios, F. Nuflot, S. Stevenson, M. Black, D. Metaxas, S. Culhane, Y. Ye, and R. Mann. PLAYBOT: A visually-guided robot to assist physically disabled children in play. *Image and Vision Computing*, 16:275–292, 1998.
26. J. Wieghardt and C. von der Malsburg. Pose-independent object representation by 2-d views. In *IEEE International Workshop on Biologically Motivated Computer Vision, May 15-17, Seoul*, 2000.
27. Jan Wieghardt. *Learning the Topology of Views: From Images to Objects*. PhD thesis, Physics Dept., Univ. of Bochum, Germany, July 2001.
28. Laurenz Wiskott. *Labeled Graphs and Dynamic Link Matching for Face Recognition and Scene Analysis*. Reihe Physik. Verlag Harri Deutsch, Thun, Frankfurt am Main, 1996.
29. Laurenz Wiskott, Jean-Marc Fellous, Norbert Krüger, and Christoph von der Malsburg. Face recognition by elastic bunch graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):775–779, 1997.
30. Rolf P. Würtz. *Multilayer Dynamic Link Networks for Establishing Image Point Correspondences and Visual Object Recognition*. Verlag Harri Deutsch, Thun, Frankfurt am Main, 1995.
31. Rolf P. Würtz. Object recognition robust under translations, deformations and changes in background. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):769–775, 1997.