Proceedings of the 2006 IEEE/RSJ
International Conference on Intelligent Robots and Systems
October 9 - 15, 2006, Beijing, China

Integrated Grasp Planning and Visual Object Localization For a Humanoid Robot with Five-Fingered Hands

Antonio Morales
Department of Computer Science and Engineering
University Jaume I
Campus Riu Sec, E-12071 Castellón, Spain
morales@uji.es

Tamim Asfour, Pedram Azad,
Steffen Knoop and Rüdiger Dillmann
Institute of Computer Science and Engineering
University of Karlsruhe
Haid-und-Neu-Straße 7, 76131 Karlsruhe, Germany
{asfour,azad,knoop,dillmann}@ira.uka.de

***Abstract*— In this paper we present a framework for grasp planning with a humanoid robot arm and a five-fingered hand. The aim is to provide the humanoid robot with the ability of grasping objects that appear in a kitchen environment. Our approach is based on the use of an object model database that contains the description of all the objects that can appear in the robot workspace. This database is completed with two modules that make use of this object representation: An exhaustive offline grasp analysis system and a real-time stereo vision system. The offline grasp analysis system determines the best grasp for the objects by employing a simulation system, together with CAD models of the objects and the five-fingered hand. The results of this analysis are added to the object database using a description suited to the requirements of the grasp execution modules. A stereo camera system is used for a real-time object localization using a combination of appearance-based and model-based methods. The different components are integrated in a controller architecture to achieve manipulation task goals for the humanoid robot.**

I. Introduction

The attention of the robotics community has been drawn more and more to humanoid robots in the last years. Their design, building and applications addresses many interesting research challenges: biped walking, human-robot interaction, autonomy, interaction with unstructured and unknown environments, and many others. Among them, the development of manipulation skills is of utmost importance and one of the most complex.

One of the main challenges that humanoid developers have to face when considering manipulation issues is the design of robot hands and arms. In the case of hands for humanoids their design is guided by the need of a great versatility, which means a large number of fingers and degrees of freedom, the reduced size and the human-like appearance. A constant issue has been to design human-size light arm/hand systems either focusing on a pure mechanical approach [1] or taking some anthropomorphic and biological inspiration [2]. A recent work, *Domo*, has focused on the design of compliant and reliable humanoid arms able to run for days in a secure way for humans [3]. The limited size of robot hands complicates the dispositions of the joint actuators. The solution usually comes from the use of novel actuation systems, pneumatic or fluidic [4] or cable driven [5].

In order to deal with manipulation tasks in human-centered environments, an intensive use of sensor information, particularly visual and tactile, within closed control loops is indispensable. Visual information has been used mainly to identify and apprehend the pose and shapes of objects [6], [7]. Especially relevant for dexterous manipulation tactile information has been used to reach stable grasps through finger gating or for controlling whole body grasping [8]. Several control architectures were proposed for manipulation tasks. Their main goals are to coordinate a set of behaviors implied by manipulation [9], to introduce learning in the sensor motor coordination [10], and to get inspiration from biology findings [2].

The work presented in this paper is part of long term German Humanoid Project, which goal is to develop a robot aimed to assist humans in tasks of everyday life [11]. To reach these goals, many complex abilities and characteristics are included: Humanoid shape, multimodality and the ability to cooperate with humans and learn. In the aspect of manipulation it includes the ability to learn from demonstration and to use high level cognitive models of objects and tasks.

In this paper we present an integrated approach for grasp planning. The central idea of this system is the existence of a database with the models of all the objects present in the robot workspace. From this central fact we develop two necessary modules: a visual system able to locate and recognize the objects (Sec. III), and an offline grasp analyzer that provides the most feasible grasps configuration for each object (Sec. IV). The results provided by these modules are stored and used by the control system of the humanoid to decide and execute the grasp of a particular object. We emphasize that this paper describes a first step towards a complete humanoid grasping system. At this stage the use of object and hand models allows the fast development and test of multiple interactive manipulation skills. In the long-term it is desirable, and is our purpose to develop grasping and manipulation strategies able to deal with unmodelled and unknown objects.

1-4244-0259-X/06/$20.00 ©2006 IEEE

II. System overview

Since the robot has to work in an environment mostly designed for humans, the approach of the whole project has been to build a anthropomorphic arm/hand system that allows to imitate the way humans perform these activities. ARMAR, our humanoid robot, has 23 mechanical degrees-of-freedom (DOF). From the kinematics control point of view, the robot consists of five subsystems: Head, left arm, right arm, torso and a mobile platform [12]. The head has 2 DOFs arranged as

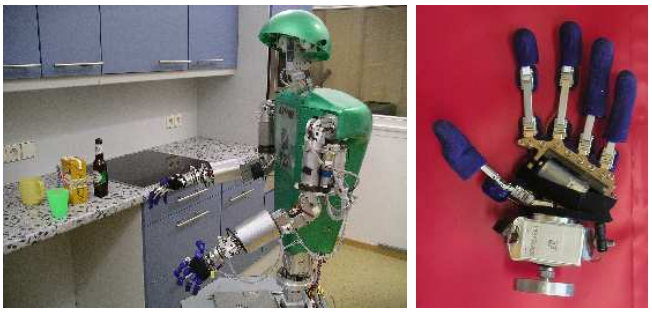


Fig. 1. The humanoid robot ARMAR with the 5-finger hands

pan and tilt and is equipped with a stereo camera system and a stereo microphone system. Each of the arms has 7 DOFs and is equipped with 6 DOFs force torque sensors on the wrist. Each hand has five fingers and 11 DOFs (3 for the thumb and 2 for the other four fingers) driven by fluidic actuators [4].

A functional description of the grasp planning system described in this paper is depicted in figure 2. It consists of the next parts:

- The **global model database**. It is the core of our approach. It contains not only the CAD models of all the objects, but also stores a set of feasible grasps for each object. Moreover, this database is the interface between the different modules of the system.
- The offline **grasp analyzer** that uses the model of the objects and of the hand to compute on a simulation environment a set of stable grasps (see Sec. III). The results produced by this analysis are stored in the grasps database to be used by the other modules.
- A **online visual procedure to identify objects in stereo images** by matching the features of a pair of images with the 3D prebuilt models of such objects. After recognizing the target object it determines its location and pose. This information is necessary to reach the object. This module is described in detail in section IV.
- Once an object has been localized in the work-scene, a grasp for that object is then selected from the set of precomputed stable grasps. This is instanced to a particular arm/hand configuration that takes into account the particular pose and reachability conditions of the object. This results in an approaching position and orientation. A path planner reaches that specified grasp location and orientation. Finally, the grasp is executed. These modules are not described in this paper since they are still under development.

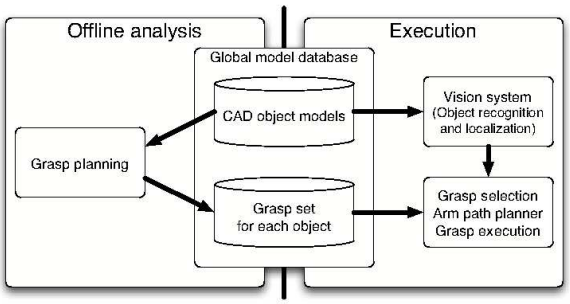


Fig. 2. Overview of the system.

III. Offline grasp analysis

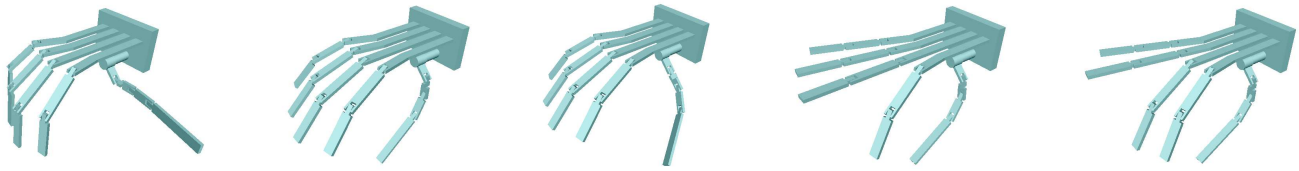
In most of the works devoted to grasp synthesis, grasps are described as sets of contact points on the object surface where forces/torques are exerted. However, this representation of grasps presents several disadvantages when considering their execution in human-centered environments. These problems arise from the inaccuracy and uncertainty about the information of the object. Since we have models of the shapes of the objects this uncertainty comes mainly from the location of the object and inaccuracy in the positions of the mobile humanoid. Usually, the contact-based grasp description requires the system be able to reach precisely the contact points and exert precise forces.

It is possible to include inaccuracy in the force/torque models, but this paper faces this problem from a different approach. In our approach grasps are described in a qualitative and knowledge-based fashion. Given an object, a grasp of that object will be described by the following features (see Fig. 4):

- **Grasp type:** A qualitative description of the grasp to be performed (see Fig. 3). The type of the grasp has practical consequences since it determines the grasp execution control, i.e.: the hand preshape posture, the control strategy of the hand, which fingers are used in the grasp, the way the hand approaches the objects and how the contact information of the tactile sensors is interpreted.
- **Grasp starting point (GSP):** For approaching the object, the hand is positioned at a distant point near it.
- **Approaching direction:** Once the hand is positioned in the GSP it approaches the object following this direction. The **approaching line** is defined by the GSP and the approaching direction.
- **Hand orientation:** the hand can rotate around the approaching direction. The rotation angle is a relevant parameter to define grasp configuration.

It is important to note that all directions are given with respect to an object centered coordinate system. The real approach directions result from matching of this relative description with the localized object pose in the workspace of the robot.

A main advantage of this grasp representation is its practical application. A grasp can be easily executed from the informa-



(a) Hook (b) Cylindrical (c) Spherical (d) Pinch (e) Tripod

Fig. 3. Hand preshapes for the five types of grasp.

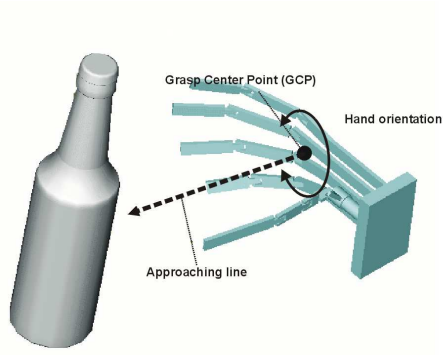


Fig. 4. Schematics with the grasp descriptors

tion contained in its description, and is better suited for the use with execution modules like arm path planning. Moreover this representation is more robust to inaccuracies since it only describes starting conditions and not final conditions like a description based in contacts points.

It is important to notice too, that this approach involves the existence of an execution module able to reach a stable grasp from the given initial conditions. This module will require the uses of sensor information, tactile or visual, to complete the grip. This module is out of the scope of this paper.

A. Grasp types

Cutkosky describes 16 different grasping hand postures for human hands [13]. The taxonomy described by Cutkosky is very complete and present many grasps that could be hardly executed by our anthropomorphic five-fingered hand attending to its mechanical limitations. Hence, we have made a selection of the most representative grasps that can be executed with our hand. These are three power grasps: hook, cylindrical and spherical and prismatic; and two precision grasps: pinch and tripod (see Fig 3). For this selection we have considered that the only finger with abduction/adduction mobility is the thumb, being thus the only one able to change its opposition with respect to the other fingers.

- **Hook grasp:** In this grasp the hand opposes the gravity. All fingers, but the thumb, form a hook that would enclose a cylindrical shaped object. The palm might exert force opposing the fingers. The thumb does not participate in any case.
- **Cylindrical grasp:** All fingers close around a cylindrical object. The thumb opposes completely the other four fingers.
- **Spherical grasp:** All fingers close around a ball-shaped object. The thumb is disposed in a way that it maximizes the area covered by the fingers.
- **Pinch grasp:** The grasp is characterized by the opposition of the thumb and index finger tips. The rest of the fingers do not participate. This is appropriate to grasp thin objects.
- **Tripod grasp:** In this case the grasp is conformed by the opposition of the thumb fingertip against the index and middle finger tips. This grasp is useful to grasp small objects.

Precision grasps only imply contacts on the finger tips, while power grasps use contacts on the whole hand surface, finger tips, phalanxes, and palm. This difference is relevant for the design of the execution controller. Roughly, for the execution of a power grasp the hand approaches the object until it makes contact, and then closes the fingers. However, in the case of precision grasps, the fingers have to close at a certain distance so that only the finger tips make contact with the object.

An important aspect when considering an anthropomorphic hand is how to relate the hand with respect the grasp starting point (GSP) and the approaching direction. For this we define for the hand the grasp center point (GCP). It is a virtual point that has to be defined for every hand and that is used as reference for the execution of a given grasp. Figure 4 depicts the parameterization of a grasp. The GCP is aligned with the GSP of the grasp. Then the hand is oriented and preshaped according to the descriptors of the grasp. Finally, it moves along the approaching line.

B. Methodology of the analysis

An important characteristic in our system is that there exists a 3D CAD model for every objects that appears in the workspace. This allows for extensive offline analysis of the different possibilities to grasp an object, instead of focusing on fast online approaches. To accomplish this we have also built a computer 3D model of the hand.

We perform an extensive analysis for each object that consists of testing a wide variety of hand preshapes and approach directions. This analysis is carried out on a simulation environment where every tested grasp is evaluated according to a quality criterion. The resulting best grasps for each object are stored in order to be used during online execution of the robot.

We use GraspIt! [14] as grasping simulation environment It has some very convenient properties for our purposes such as the inclusion of contact models and collision detection algorithms, and the ability to import, use and define object and robot models.

Our approach to compute stable grasps on 3D objects is inspired by a previous work by Miller et al. using GraspIt! [15]. The offline analysis follows four steps to find the grasps for a given object:

1) The shape of the object model is approximated by a set of basic shape primitives (boxes, cylinders, spheres and cones). There are many ways to obtain these primitive approach. GraspIt! doesn't provide any procedure to produce them. We assume that the primitive description of the objects is part of the model of an object.
2) A set of candidate grasps is generated automatically for every primitive shape of the object description. A grasp candidate consists of a hand type, a grasp starting point, an approach direction and a hand orientation. For every primitive there exists a set of predefined grasp types and approaching directions [15].
3) Each grasp candidate is tested within the simulation environment. The hand is placed in the grasp starting point and oriented according to the approaching direction and hand orientation. Then, the hand is preshaped depending on the grasp type.
   The approach phase is different for power and precision grasps. For power grasps, the hand moves opened along the approach direction until it touches the object. Then, it closes and the quality of the grasp is evaluated. If the quality is under certain threshold then the hand opens, backs a step amount and closes again. This sequence is repeated until a maximum stability measurement is reached.
   However, in the case of precision grasps, a different test is designed: 1) the hand is preshaped at the grasp starting point , 2) it closes and the grasp is evaluated if there exist a contact with object 3) it opens again and moves a step forward, 4) steps 2 and 3 are repeated until it reaches a maximum stability or a maximum number of steps is reached. Following this procedure we ensure that the first contacts with the object are made with the fingertips.
   The final position of the hand and the quality obtained is stored.
4) Finally, all final grasps that are over the minimum threshold are sorted and stored.

Part of this procedure is available in the source code of GraspIt! [14]. However it is designed exclusively for the Barrett Hand [16]. We have redesigned it to adapt it to our hand model.

As a metric for evaluating the quality of a grasp we use the magnitude of the largest worst-case disturbance wrench that can be resisted by a grasp of unit-strength. This metric is described in detail by Ferrari and Canny [17].

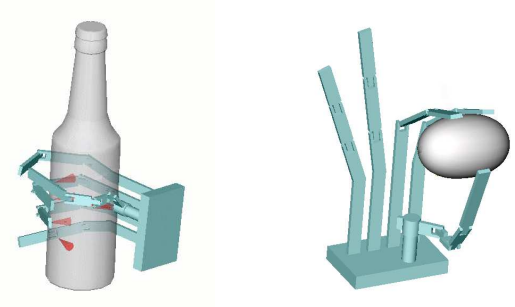


Fig. 5. Two examples of grasps produces by the grasp planning

Finally two examples of the grasps obtained for a beer bottle and an egg are shown in Fig. 5.

C. Grasp database

All stable grasps computed for every object are stored in a database in order to be used by execution modules. Every grasp stored includes the grasp type, the grasp starting point, hand orientation, approaching direction and the quality measure obtained from the simulation. This value is used by the other modules to select the best grasp for a given object.

IV. Object recognition and localization

In general, any component of a vision system in a humanoid robot for application in a realistic scenario has to fulfill a minimum number of requirements. In the particular context of the grasping system presented in this paper, the main requirements are these.

1) The component has to deal with a potentially moving robot and robot head: The difficulty caused by this is that the problem of segmenting objects can not be solved by simple background subtraction. The robot has to be able to recognize and localize objects in an arbitrary scene when approaching the scene in an arbitrary way.
2) Recognition of objects has to be invariant to 3D rotation and translation: It must not matter in which rotation and translation the objects are placed in the scene.
3) Objects have to be localized in 6D (location + orientation) with respect to a 3D rigid model in the world coordinate system: It is not sufficient to fit the object model to the image, but it is crucial that the calculated 3D pose is sufficiently accurate in the world coordinate system. In particular, the assumption that depth can be recovered from scaling with sufficient accuracy in practice is questionable.
4) Computations have to be performed in real-time: For realistic application, the analysis of a scene and accurate localization of the objects of interest in this scene should take place at frame rate in the optimal case, and should not take more than one second.

A. The Limits of State-Of-The-Art Model-Based Systems

Most model-based object tracking algorithms are based on relatively simple CAD wire models of objects, as the example

llustrated in Figure 6. Using such models, the starting and end points of lines can be projected very efficiently into the image plane, allowing real-time tracking of objects with relatively low computational effort. However, the limits of such systems are clearly the shapes they can deal with. Most real-word objects, such as cups, plates and bottles, can not be represented in this manner. The crux becomes clear when taking a look at an object with a complex shape, as it is the case for the can illustrated in Figure 7.

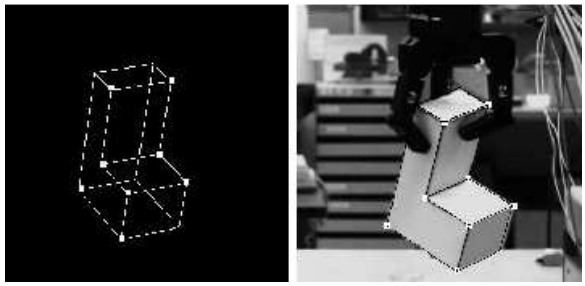


Fig. 6. Illustration of an object modeled by a wire model from [6]

The only practical way to represent such an object as a 3D model is to approximate its shape by a relatively high number of polygons. To calculate the projection of such a model into the image plane practically the same computations a rendering engine would do, have to performed. But not only the significantly higher computational cost makes common model-based approaches not feasible, also from a conceptual point of view the algorithms can not be extended for complex shapes: Objects which can be represented by straight lines and even planes have the property that each edge of the object is represented by a straight line in the model, which are then used for matching. As soon as an object also has curved surfaces this is not the case anymore: the edges of the polygons do not correspond to potentially visible edges. In [18], we show that a purely model-based approach for arbitrary 3D object models would take more than five minutes for the analysis of one potential region, having a database of three objects.

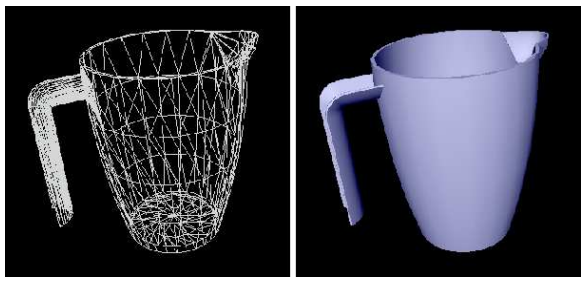


Fig. 7. Illustration of a 3D model of a can

B. Our Approach

Our approach combines the benefits of model-based and *global* appearance-bases methods [19] for object recognition and localization. Recently, *local* appearance-based methods using texture features have become very popular [20]–[23]. However, these methods are only applicable for sufficiently textured objects, which is often not the case for the objects of interest for our intended application [18].

In [18], we present a system which can build object representations for appearance-based recognition and localization automatically, given a 3D model of the object. An initial estimate for the position of the object is determined through stereo vision, while an initial estimate for the orientation is determined by retrieving the rotation the recognized view was produced with. Then, a number of correction calculations are performed for accurate localization, which is explained in detail in [18]. An outline of the overall algorithm is given by the following steps:

1) Perform color segmentation in both images.
2) Determine color blobs with a connected components algorithm.
3) Match the blobs in the left and right image on the base of their properties and the epipolar geometry.
4) For each matched blob:
5) Calculate initial estimate for the position by stereo triangulation.
6) Determine the best matching view by calculating the Nearest Neighbor in the PCA eigenspace.
7) Determine initial estimate for the orientation by retrieving the stored rotation for the recognized match.
8) Apply pose correction formulate as presented in [18].
9) Verify validity by comparing the size of the blob to the expected size, determined on the base of the calculated pose and the object model.

Fig. 8. Illustration of the color segmentation result for the colors red and green

As we show in [18], our system is very robust and is able to recognize and localize the objects in our test environment accurately and reliably in real-time. Recognition and localization for one potential region takes approximately 5 ms on a 3 GHz CPU, with a database of five objects: a cup, a cup with a handle, a measuring cup, a plate, and a small bowl. An exemplary segmentation result is shown in Figure 8; the result of a full scene analysis is visualized in Figure 9.

V. Discussion and Conclusion

At this point it is important to mention the work of Kragic et al. [6] due to the similarity in some aspects to our work. They present a visual tracking system also able to recognize objects. Once an object is recognized the model and pose of it is sent to GraspIt!. A human operator uses GraspIt! visualization and

Fig. 9. Recognition and localization result for an exemplary scene. Left: left input image. Right: 3D visualization of the result.

analysis tools to determine a stable grasp with the Barrett Hand. Later the grasp is executed.

On the visual part the main difference is that we are able to deal with arbitrarily complex shaped objects, while Kargic et al. are limited to planar-faced objects. Another main difference to our approach is that we compute grasps automatically and offline, without a human operator. The addition of these features, five-fingered hands, automatic grasping synthesis, and realistic shaped objects in a realistic environment (but with simplified texture/colours) makes our approach more complete and autonomous.

To conclude, in this paper we have presented an integrated approach that includes an offline grasp planning system with an visual object identification system. The integration of these two modules relies on the use of an appropriate object and grasp representation database that is also described.

However, the work presented here is only a part of a larger manipulation system. Some modules are still required, in order to execute any of the grasps computed. First, in any situation several grasp candidates are possible, but only one can be executed. A module that selects one taking into account the task and the execution conditions is necessary. Once a grasp is selected, an arm motion planner is necessary to move the hand to the pregrasping location according to the grasp description and the object pose. And finally, a module that executes the grasp using tactile and visual feedback has to be developed too.

ACKNOWLEDGMENT

The work described in this paper was partially conducted within the German Humanoid Research project SFB588 funded by the German Research Foundation (DFG: Deutsche Forschungsgemeinschaft) and the EU Cognitive Systems project PACO-PLUS (FP6-2004-IST-4-027657) funded by the European Commission. Support for the first author is provided in part by the spanish government under project DPI2001-3801 and DPI2004-01920; by the Generalitat Valenciana under projects inf01-27, GV01-244, CTIDIA/2002/195, GV05/137; and by the Fundació Caixa-Castelló under project P1-1B2005-28, P1- 1A2003-10.

REFERENCES

[1] A. Konno, K. Nagashima, R. Furukawa, K. Nishiwaki, T. Oda, M. Inaba, and H. Inoue, "Development of a humanoid robot *saika*," in *IEEE/RSJ International. Conference on Intelligent Robots and Systems*, Grenoble, France, Sept. 1997, pp. 805–810.

[2] C. Laschi, P. Dario, M. Carrozza, E. Guglielmelli, G. Teti, D. Taddeucci, F. Leoni, B. Massa, M. Zecca, and R. Lazzarini, "Grasping and manipulation in humanoid robotics," in *IEEE International Conference on Humanoid Robots (Humanoids 2003)*, Karlsruhe, Germany, Oct. 2003.

[3] A. Edsinger-Gonzalez and J. Webber, "Domo: a force sensing humanoid robot for manipulation research," in *IEEE International Conference on Humanoid Robots*, Santa Monica, California, Nov. 2004, in CD.

[4] S. Schulz, C. Pylatiuk, A. Kargov, R. Oberle, and G. Bretthauer, "Progress in the development of anthropomorphic fluidic hands for a humanoid robot," in *IEEE-RAS/RSJ Inernational Conference on Humanoid Robots (Humanoids 2004)*, Santa Monica, California, Nov. 2004.

[5] W. Bluethmann, R. Ambrose, M.Diftler, S. Askew, E. Huber, M. Goza, F. Rehnmark, C. Lovchik, and D. Magruder, "Robonaut: A robot designed to work with humans in space," *Autonomous Robots*, vol. 14, no. 2–3, pp. 179 – 197, Mar. 2003.

[6] D. Kragic, A. Miller, and P. Allen, "Real-time tracking meets online grasp planning," in *IEEE International Conference on Robotics and Automation*, Seoul, Republic of Korea, May 2001, pp. 2460–2465.

[7] G. Taylor and L. Kleeman, "Integration of robust visual perception and control for a domestic humanoid robot," in *IEEE International Conference on Robotics and Automation*, Sendai, Japan, Sept. 2004, pp. 1010–1015.

[8] R. Platt Jr., A. Fagg, and R. Grupen, "Extending fingertip grasping to whole body grasping," in *IEEE International Conference on Robotics and Automation*, Taipei, Taiwan, Sept. 2003, pp. 2677–2682.

[9] R. Platt Jr., O. Brock, A. Fagg, D. Karupiah, M. Rosenstein, J. Coelho, M. Huber, J. Piater, D. Wheeler, and R. Grupen, "A framework for humanoid control and intelligence," in *IEEE International Conference on Humanoid Robots (Humanoids 2003)*, Karlsruhe, Germany, Oct. 2003.

[10] M. Cambron and R. Peters, "Learning sensory motor coordination for grasping by a humanoid robot," in *IEEE International Conference on Systems, Man, and Cybernetics*, vol. 1, Nashville, Tennessee, Oct. 2000, pp. 6–13.

[11] R. Becher, P. Steinhaus, and R. Dillmann, "ARMAR II - a learning and cooperative multimodal humanoid robot system," *"'International Journal of Humanoid Robotics"'*, vol. 1, no. 1, pp. 143–155, 2004.

[12] T. Asfour, K. Berns, and R. Dillmann, "The humanoid robot ARMAR: Design and control." in *The 1st IEEE-RAS International Conference on Humanoid Robots (HUMANOIDS 2000)*, MIT, Boston, USA, 7-8 September, 2000.

[13] M. Cutkosky, "On grasp choice, grasp models and the design of hands for manufacturing tasks," *IEEE Transactions on Robotics and Automation*, vol. 5, no. 3, pp. 269–279, 1989.

[14] A. Miller and P. Allen, "Graspit!: A versatile simulator for robotic grasping," *IEEE Robotics & Automation Magazine*, vol. 11, no. 4, pp. 110–122, Dec. 2004.

[15] A. Miller, S. Knoop, H. Christensen, and P. Allen, "Automatic grasp planning using shape primitives," in *IEEE International Conference on Robotics and Automation*, Taipe, Taiwan, Septmeber 2003.

[16] Barrett Technology Inc., http://www.barrett.com/.

[17] C. Ferrari and J. Canny, "Planning optimal grasps," in *IEEE International Conference on Robotics and Automation*, Nice, France, May 1992, pp. 2290–2295.

[18] P. Azad, T. Asfour, and R. Dillmann, "Combining appearance-based and model-based methods for real-time object recognition and 6d-localization," in *International Conference on Intelligent Robots and Systems (IROS)*, Beijing, China, 2006.

[19] S. Nayar, S. Nene, and H. Murase, "Real-time 100 object recognition system," in *International Conference on Robotics and Automation (ICRA)*, vol. 3, Minneapolis, USA, 1996, pp. 2321–2325.

[20] D. G. Lowe, "Object recognition from local scale-invariant features," in *International Conference on Computer Vision (ICCV)*, Corfu, Greece, 1999, pp. 1150–1517.

[21] V. Lepetit, L. Vacchetti, D. Thalmann, and P. Fua, "Fully automated and stable registration for augmented reality applications," in *International Symposium on Mixed and Augmented Reality (ISMAR)*, Tokyo, Japan, 2003, pp. 93–102.

[22] E. Murphy-Chutorian and J. Triesch, "Shared features for scalable appearance-based object recognition," in *IEEE Workshop on Applications of Computer Vision*, Breckenridge, USA, 2005.

[23] S. Obdrzalek and J. Matas, "Object recognition using local affine frames on distinguished regions," in *British Machine Vision Conference (BMVC)*, vol. 1, Cardiff, UK, 2002, pp. 113–122.