AN ANTHROPOMORPHIC GRASPING APPROACH FOR AN ASSISTANT HUMANOID ROBOT

A. Morales
Department of Computer Engineering and Science
Universitat Jaume I
Castellón, Spain

P. Azad, T. Asfour, D. Kraft, S. Knoop and R. Dillmann
Institute of Computer Science and Engineering, Universität Karlsruhe (TH)
Germany

A. Kargov, Ch. Pylatiuk, S. Schulz
Institute for Applied Computer Sciences, Forschungszentrum Karlsruhe GmbH
Germany

Speaker: Antonio Morales, Email: morales@icc.uji.es

Topic: Anthropomorphic robot hands
Keywords: Humanoids manipulation, grasp planning, robot hands

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 off-line grasp analysis system and 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.

1 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. This paper describes an approach in the field of humanoid manipulation.

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 design is guided by the need for great versatility, which means a large number of fingers and degrees of freedom, reduced size and 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, fluidic [4] or cable driven [5].

The work presented in this paper is part of the longterm German Humanoid Project, which aims at developing a humanoid robot that can assists humans in everyday service tasks[6]. It should be able to adapt to an unstructured environment, have the necessary skills to help humans, and be able to communicate with them in a friendly and intelligent manner. This project represents a large effort that implies developments in many fields: Mobile robotics, human-machine interaction, machine learning, vision, cognitive reasoning, robotic manipulation, etc.

Manipulation skills, as a part of this system, have to meet these goals too. The robot has to help humans in manipulation tasks, and, particularly, be able to manipulate objects in this environment. This implies not only to have the appropriate hardware tools (i.e.: hand, arm and sensory system) but also the necessary computational skills.

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. The humanoid robot ARMAR [7] 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.

Figure 1: The humanoid robot ARMAR in the kitchen

The head has 2 DOFs arranged as 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. The robot is also endowed with two five-finger anthropomorphic hands, whose requirements and design are described in this paper.

We present an integrated approach for humanoid manipulation and grasping. This approach considers two main parts. First we focus on the design of a suitable anthropomorphic hand for humanoid manipulation paying special attention to the kind of tasks to be carried out by it. To achieve this goal we first carry out an analysis of the requirements of the robotic hand (sec. 2). A hand is designed following the conclusions of this analysis (Sec. 3).

On the second part we describe the computational part of our approach. Two main main modules are described in this paper: An offline grasp planner (Sec. 5) that determines the best grasps for every object in the work space, and a vision system able to recognize and locate objects (sec 6). An overview of the whole system is presented in section 4.

2 Analysis of the requirements of an anthropomorphic robot hand

For a useful design of the hand it is necessary to understand the tasks to be performed by the robot, and the objects and the environment where these will be executed. In this project the designated environment is human homes. In particular, and in the current stage, the main focus lies on a kitchen environment. A real kitchen has been build as a testbed for the humanoid. Fig. 1 shows the robot in this environment.

2.1 Environment and task

Within that experimental environment the humanoid has to deal with all kind of objects and to perform some basic manipulation tasks. Despite this apparent simplicity, we have complex requirements. First of all objects of all shapes and sizes appear in this environment:

- **Bottles, glasses, cups:** Cylindrical objects of medium/large size. In the case of cups and glasses their inner sides are reachable. In the case of cups they have a handle, too.
- **Plates, trays:** Basically planar large objects.
- **Pens, cutlery:** Small elongated objects.
- **Door handles, drawers:** Cylindrical objects with a small radius but larger than in the previous category. Note that in the kitchen environment there are all kinds of doors: Closets, dishwasher, oven, microwave, etc.
- **Arbitrary small objects.**
- **Switches, buttons, rotary switches:** They are not movable objects, but their size is similar to the objects in the previous category.

As important as the objects to be manipulated are, the tasks or manipulative actions themselves. That is, for what this objects are going to be grasped:

- **Transportation:** The objects are grasped to be transported. This implies a firm grasp, lifting the object, and releasing it somewhere else.
- **Filling/pouring:** Applied onto bottles, glasses and cups. This requires a firm grasp that allows orienting the object as needed.
- **Opening/closing:** This implies a firm grasp of the handle of the door.
- **Pushing/pulling:** A more general action than the previous, since it can be exerted on many objects.
- **Switching/pushing buttons.**
- **Giving/receiving from humans:** Receiving implies a receptive posture that reacts when the human approaches the object to the robot. The posture and the reaction can be different depending on the type of object.
- **Shaking hands:** In this case, it receives a human hand.

Notice that most of these tasks do not require very precise or dexterous manipulation, but a large flexibility in the skills of the hand.

2.2 Grasp taxonomy

To make an anthropomorphic hand appear natural the movements and grasp types have to resemble those of the human hand. In this sense many taxonomies of the human hand grasping postures have been studied and been translated to robotic hands. In particular [8] presents 17 different categories of the human grasp patterns. Two things have to be considered about these taxonomies. First, they are derived from a study of the human hand, which is much more flexible and capable than any robot hand, so a grasp taxonomy for a robot hand can be only a simplified subset of the one compiled for humans. Second, studies of human behavior while grasping real objects show that patterns really used by humans present differences with those in taxonomies [9].

As a conclusion, any taxonomy proposed is only a reference of the kind of configurations a robot hand should be able to reach. Here we describe the most often-used grasp patterns, that should be considered for developing a fully capable hand:

- **Power:** The contacts against the objects are made through large surfaces of the hand, including the finger phalanxes and the palm. Large forces are applied with these grasps.
  - **Spherical:** Used to grasp spherical objects.
  - **Cylindrical:** Used grasp long objects that can not be be enclosed by the hand.
  - **Lateral:** The thumb exerts a force against the lateral side of the index finger.
- **Precision:** The contacts are made by the fingertips.
  - **Prismatic (Pinch grasp):** Used to grasp elongated (with a small diameter) and very small objects. It can be made from two fingers (thumb+index), up to five.
  - **Circular (Tripod):** Used to grasp circular or round objects, it can be made with three fingers (tripod), four or five.
- **Non-Prehensile:**
  - **Hook:** The hand forms a hook on the object, and the force is exerted against an external force (gravity).
  - **Button/press (pointing)**
  - **Pushing (flat hand)**

It is very relevant to our work and our goals to relate the categories that appear in this taxonomy with the objects, and the tasks described above. Table 1 summarizes the range of manipulations activities in which the hand is going to be involved and relates them to the grasp patterns necessary to execute them.

Table 1: Objects, tasks and grasp patterns

Object	Task	Grasp pattern
Bottles, glasses and cups	Transporting	Power, cylindrical (from the side or from above)
	Pouring/filling	Power, cylindrical (from the side)
Cups (using handle)	Pouring/filling	Power,lateral/Precision, Pinch
Plates/trays	Transporting	Power,Lateral/Precision, Pinch
	Receiving from human	Non-prehensile, planar hand
Pens/cutlery	Transporting	Precision, Prismatic
Door handles	Closing	Power, cylindrical
	Opening	Power, cylindrical/Non-prehensile, hook
Small objects	Transporting	Precision, Circular (Tripod)
Switches, buttons	Pushing	Non-prehensile, Pointing
Rotary switches, bottle taps	Rotating	Power, Lateral / Precision, Tripod

2.3 Requirements of the robot hand

In the previous subsections we have studied the task, the objects, and the kind of grasps which are necessary to execute the different tasks. Here we analyze how these necessities can be fulfilled by a robotic hand, or more precisely, which features are necessary in the hand to be able to carry out the desired tasks. We will focus not only on the mechanical characteristics of the hand, but also on the sensorial capabilities.

2.3.1 Degrees of freedom/joints and geometry

For a good execution of power grasps (cylindrical and spherical) the fingers envelop the object. So they need to have some flexibility to extend along the object surface. By providing the fingers with at least two joints we ensure the adaptability of the fingers to the object shape. This is particularly necessary for the thumb, the index and the middle finger since they are involved in almost all the power grasps. Although it is recommended for all the fingers to have at least two joints.

The thumb is a special case since its function is to oppose the other four fingers. This allows to form the pincer that is the base for all power and precision grasps. Apart from the flexibility guided by having two joints the thumb should be able to be oriented against the other fingers, particularly index and middle. This can be achieved by adding an adduction/abduction degree of freedom in the base of the finger.

As a prove of dexterity and the possibilities of a hand the thumb fingertip should be able to reach the index and middle fingertips, at least, without reaching their joint limits. These requirements would facilitate most of the precision grasps, the tripods for an instance.

Most of the power grasps are realized by pressing the objects against the hand palm. In this sense, it would be of great interest to have a robot hand with a palm suitable for executing such actions. This can be achieved by building the palm as a large planar surface, and designing the mechanical parts of the other fingers in a way that they do not occlude the surface of the palm. A related desirable property is the possibility to conform a planar hand. This allows the pushing posture in which the whole hand makes contact against a planar surface.

2.3.2 Exerted forces

The tasks can include grasping of heavy objects. This is the case e.g. for full bottles, glasses and cups. In these cases it is necessary to ensure that the hand applies enough force to have firm grasps of these objects. That is, it is able to avoid that the objects slip when they are manipulated, and external forces are applied to them. These forces are due to gravity or the acceleration generated by the arm when the object is moved. The expected maximum object weight is about 1,5 Kg.

On the other hand, light objects are also to be considered. In that case smaller forces are necessary. So it is desirable to have the possibility of a wide range of forces exerted by the hand.

2.3.3 Sensors

In an unstructured environment like ours, no manipulation action can be successfully executed without sensorial information. The location of the objects is undetermined. The only way to know if there has been a contact is by the use of some kind of sensorial information.

In the case of prosthetic hands, the human vision is a powerful tool in addition to its cognitive reasoning about objects and shapes. But in the case of robots imitating these capabilities it is too complicated and currently not feasible.

First of all, it is necessary to know the configuration of the hand. This is only possible by the use of position sensors that measure at any moment the angular value of a joint or DOF. This is absolutely necessary during grasping operations since allows geometric reasoning about the contacts and the shape of an object. The lack of these will dramatically restrict the kind of control techniques that can be applied.

In manipulation, probably the most necessary sensors are contact sensors. This is the only secure way to verify a contact between the hand and an object. These contact sensors have to be placed not only on the fingertips but also on the inner surface of the phalanx and in several places of the palm. It is interesting to note that for lateral grasps it would be convenient to have contact sensors on the side of the index finger.

2.4 Final remarks

In conclusion the main requirements and improvements that a robot hand should present are these:

- All the finger joints should have position sensors that allow to know the exact angle in which they are flexed during a grasping action.
- A hand palm suitable for being used for power grasp. That is, it is composed of a large planar surface, and the mechanical parts of the fingers do not interfere with the palm.
- The mobility of the thumb should allow to oppose the thumb with two other fingertips, especially the index and the middle.
- The control of the actuators should allow to reach any angle within their range, and this should be programmable from a controller computer.

It is important to notice that these are the desirable requirements from the point of view of their use in manipulation tasks. Next section describes the hand finally designed and built trying to fulfill these requirements.

3 Design of an anthropomorphic five-fingered hand

Service robots that are designed to assist old and handicapped people in a domestic or nursing environment must be able to grasp and manipulate a manifold of objects. Consequently an artificial robot hand is needed that mimics the size, morphology and functionality of a human hand as closely as possible. Contrary to other hand designs where many DC motors are used for actuation of each independent DOF a new approach was chosen. Eleven small flexible fluidic actuators [10, 11] are integrated into the fingers‘ joints (A in Fig. 2). These actuators consist of reinforced elastic chambers that expand during inflation. Like all pneumatic actuators they have a very good power to weight ratio and good dynamics plus a low weight. All fingers can be flexed in less than half a second.

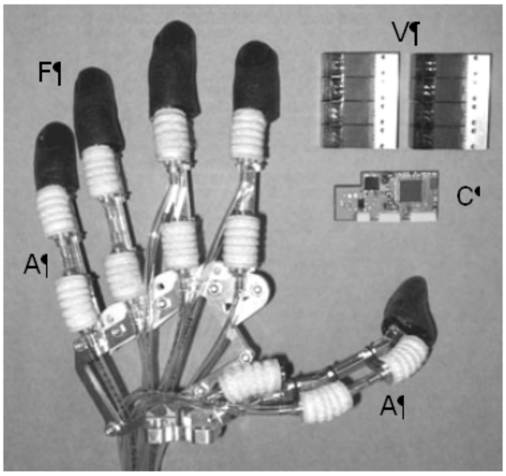

Figure 2: The artificial hand consists of 11 fluidic actuators (A), an electronic board with local controller (C), eight miniature valves (V) and a skeletal framework with elastic finger pads (F).

Using the new actuation technology the power source (compressed air) can be located away from the hand resulting in a very low mass of the robot hand of 243g. This reduces the impact of the hand in an accidental collision with a human drastically. However, a load to weight ratio of 40:1 can be achieved. The pressurised air is transmitted by a pneumatic line to miniature valves at the metacarpus of the hand (V in Fig. 2). So the number of independent DOF depends on the number of valves and robot hands with four to twelve valves have been built. The flexion movement of the digits is generated by expansion during inflation of the actuators, whereas the extension is performed by vacuum supported by a spring element. The thumb can perform an opposition movement to enable palmar prehension, being the most important grasping pattern. Additionally different precision and power prehension patterns can be performed due to multi articulation, such as cylindrical grasp, lateral grasp, tripod pulp grasp, hook grasp, and spherical grasp. Compliant fingers that can give way and adapt stiffness both actively and passively ensure that different objects can be grasped adaptively. As well the movements appear smoother. The framework of the hand is made of aluminium with high tensile strength and can be compared to the skeleton of a natural hand. All fingers are constructed identical and follow a modular design concept, which allows for a variety of hand sizes and hand geometries. Modularity also simplifies the maintenance and reduces production costs. Additionally, the number of actuators can be varied by replacing active joints with passive ones and vice versa.

Control of the finger movements is performed by a 3-level hierarchial control system and is depicted in Fig. 3. The lowest level is responsible for local grip control using signals from position and force sensors of the digits. An electronic board (C in Fig. 2) was developed, that consists of a programmable microcontroller PIC16F877 (Microchip Technology Inc., USA), drivers for the valves, analog-digital converter and a serial RS232 interface driver for communication with higher levels of control. Alternatively data transfer can be done via Bluetooth technology. The higher levels of control are responsible for the selection of the appropriate prehension pattern and hand-arm coordination. Finger joint position is controlled by eight 2/2-way custom made miniature valves using pulse width modulated signals. Finger joint angles are determined contactless by using magnets and miniature size programmable magnetic rotary encoders with a resolution of 0.35 degrees or 1024 positions per revolution.

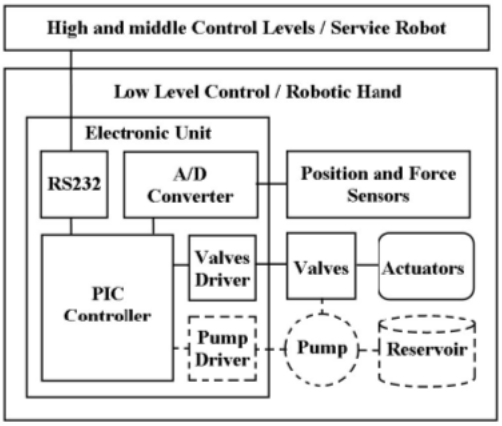

Figure 3: Block diagram of the control system.

4 System overview

An important challenge that has to be faced by a humanoid robot is caused by the operation in environments designed for humans and, more importantly, inhabited by humans. These environments are highly unstructured and unpredictable. In order to deal with these circumstances within the field of manipulation, the main approaches rely on the intensive use of particularly visual and tactile sensor information with a closed control loop. Visual information has been used mainly to identify and apprehend the pose and shapes of objects [12, 13]. Especially relevant for dexterous manipulation, tactile information has been used to reach stable grasps through finger gating or for controlling whole body grasping [14]. Several control architectures were proposed for manipulation tasks. Their main goals are to coordinate a set of behaviors implied by manipulation [15], to introduce learning in the sensor motor coordination [16], and to get inspiration from biology findings [2].

The central idea of our approach 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. 5), and an offline grasp analyzer that provides the most feasible grasps configuration for each object (Sec. 6). The results provided by these modules are stored and used by the control system of the humanoid to decide and execute the grasping 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 manipu-

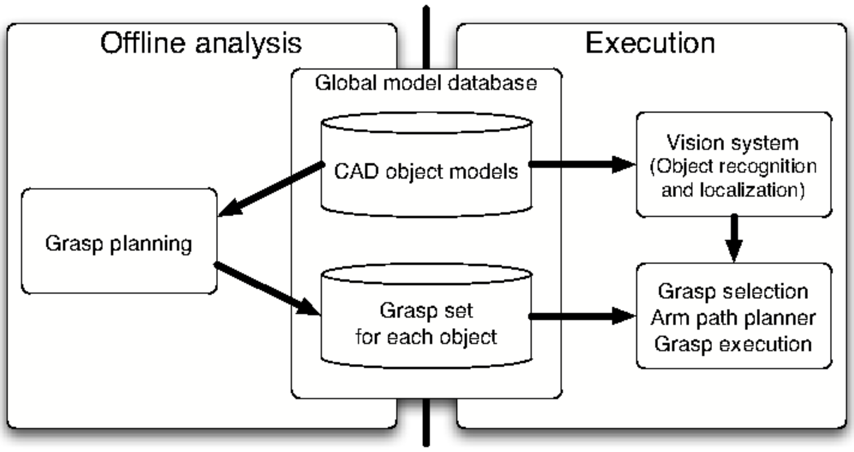

Figure 4: Overview of the system.

lation 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.

A functional description of the grasp planning system described in this paper is depicted in figure 4. It consists of the following 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 in a simulation environment a set of stable grasps (see Sec. 5). The results produced by this analysis are stored in the grasp database to be used by the other modules.
- An **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 6.
- 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.

5 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 that the system is able to reach the contact points precisely 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. 5):

- **Grasp type:** A qualitative description of the grasp to be performed. This is one of the grasps described in the taxonomy on subsection 2.2 (see also Fig. 6). The type of the grasp has practical consequences since it determines the control policy to follow for executing the grasp. This includes the grasp execution control, the hand preshape posture, the

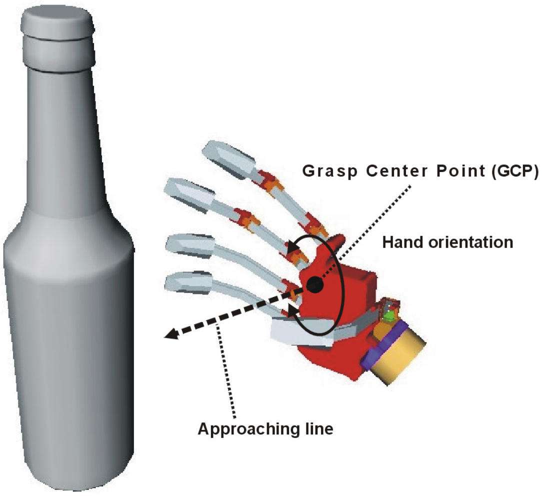

Figure 5: Schematics with the grasp descriptors

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 the 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 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 information 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.

5.1 Grasp types

In section 2.2 we described the basic grasp types that, inspired from human physiology, can be adapted to robot hands. From this taxonomy we do not consider the non-prehensile pointing and pushing hand patterns since they are not true grasps. With respect to the others, the power lateral grasps is discarded due to its complexity and because the index finger has not been tested for lateral efforts. The remaining five grasps are going to be studied in simulation: Three power grasps (spherical, cylindrical and hook), and two precision ones (pinch and tripod).

However the five-fingered anthropomorphic hand described in section 3 has eight independent joints and is actuated via only four air-pressure valves. This means that actually only four degrees of freedom are available, and that some of the joints are coupled. This limitations restrain the hand postures that can be produced. In particular the tripod grasp can not be produced, or more precisely, it can not be distinguished from a cylindrical hand preshape.

Despite these considerations, in this paper we have decided to use a model in which all eight joints can be actuated independently. There are two reasons for this: first, the hands developed in the future will likely allow the independent actuation

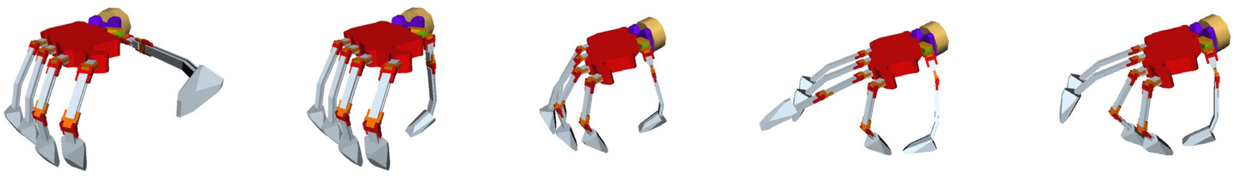
(a) Hook (b) Cylindrical (c) Spherical (d) Pinch (e) Tripod

Figure 6: Hand preshapes for the five grasp types.

of all joints; an second, the grasp analysis methodology described in the next section is equally well suited for both joint configurations. We consider here five grasp types: Precision pinch and tripod, power hook, cylindrical and spherical. They are depicted in Figure 6.

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 5 depicts the parameterization of a grasp. The GCP of the hand is aligned with the GSP of the grasp. Then the hand is oriented and preshaped according to the description of the grasp. Finally, it moves along the approaching line.

5.2 Methodology of the analysis

An important characteristic in our system is that there exist 3D CAD models for the objects that appear 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 built a 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![17] as grasping simulation environment, that has been developed by Andrew T. Miller and others in the Department of Computer Science of the Columbia University. 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. for grasp planning with the Barrett hand[18] using GraspIt![19]. 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. It would be interesting to introduce methods to produce this representation automatically from the the CAD model of an object. In our case we have produced them manually trying to find the best primitive approach to the object shape, and minimizing the number of primitives.
2. A set of candidate grasps is generated automatically for every primitive shape of the object description. A grasp candidate consists of a hand preshape, 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 [19].
3. Each grasp candidate is tested within the simulation environment. The hand is placed at 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.

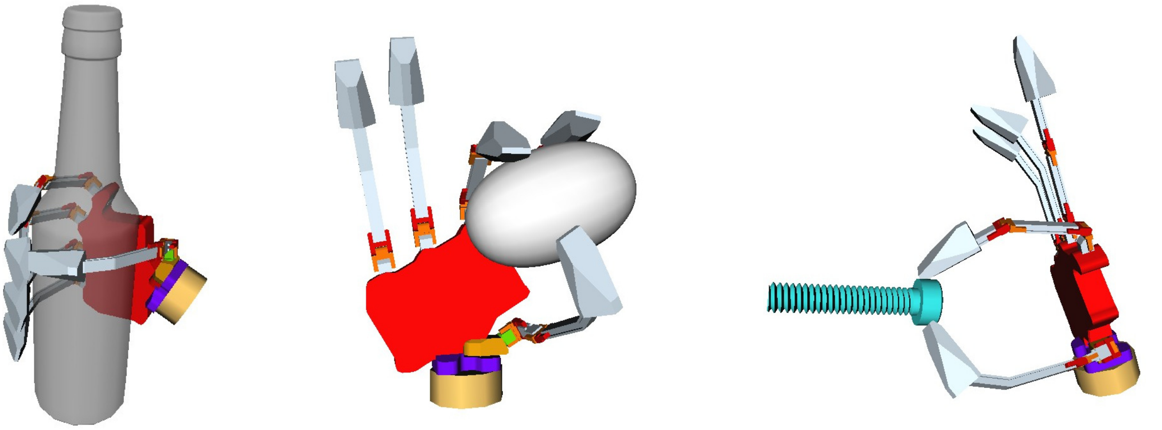

Figure 7: Three examples of grasps produces by the grasp planning

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 the 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 are stored.

4. Finally, all resulting grasps that are over a minimum threshold quality are sorted and stored.

Part of this procedure is available in the source code of GraspIt![17]. However it is designed exclusively for the Barrett Hand. 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[20]. Three examples of the grasps obtained for a beer bottle, an egg and a screw are shown in Fig. 7.

5.3 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.

6 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 the following:

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.

6.1 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 8. 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 9.

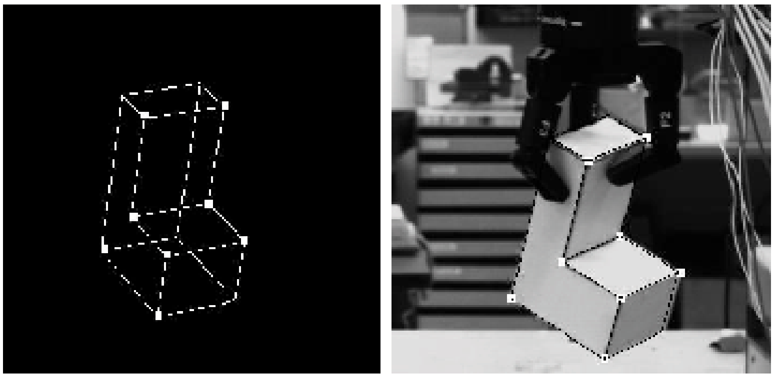

Figure 8: Illustration of an object modeled by a wire model from [12]

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 [21], 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.

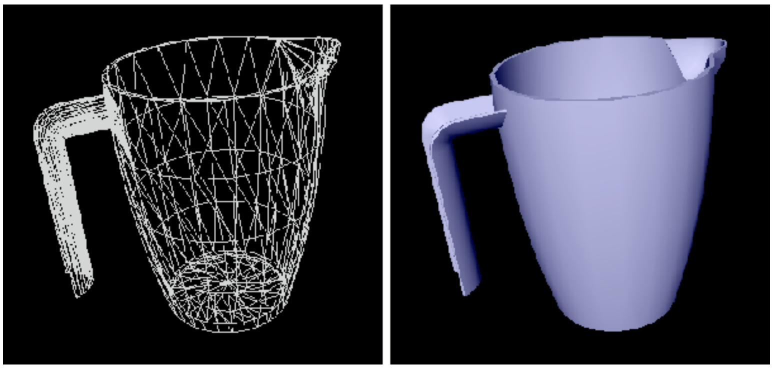

Figure 9: Illustration of a 3D model of a can

6.2 Our Approach

Our approach combines the benefits of model-based and *global* appearance-bases methods [22] for object recognition and localization. Recently, *local* appearance-based methods using texture features have become very popular [23, 24, 25, 26]. 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 [21].

In [21], 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 [21]. An outline of the overall algorithm is given by the following steps:

- Perform color segmentation in both images.

- Determine color blobs with a connected components algorithm.
- Match the blobs in the left and right image on the base of their properties and the epipolar geometry.
- For each matched blob:
- Calculate initial estimate for the position by stereo triangulation.
- Determine the best matching view by calculating the Nearest Neighbor in the PCA eigenspace.
- Determine initial estimate for the orientation by retrieving the stored rotation for the recognized match.
- Apply pose correction formulate as presented in [21].
- 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.

Figure 10: Illustration of the color segmentation result for the colors red and green

As we show in [21], 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 10; the result of a full scene analysis is visualized in Figure 11.

Figure 11: Recognition and localization result for an exemplary scene. Left: Left input image. Right: 3D visualization of the result.

7 Discussion and Conclusion

At this point it is important to mention the work of Kragic et al. [12] due to the similarity in some aspects to our work. They present a visual tracking system 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 analysis tools to determine a stable grasp for 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 Kragic 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/colors) 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 a visual object recognition 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

This work has been performed in the framework of the German humanoid robotics program (SFB 588) funded by the German Research Foundation (*DFG: Deutsche Forschungsgemeinschaft).* 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,* pages 805–810, Grenoble, France, September 1997.

[2] C. Laschi, P. Dario, M.C. 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, October 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, November 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, November 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,* 14(2–3):179 – 197, March 2003.

[6] R. Becher, P. Steinhaus, and R. Dillmann. ARMAR II - a learning and cooperative multimodal humanoid robot system. *"'International Journal of Humanoid Robotics"',* 1(1):143–155, 2004.

[7] 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.

[8] M.R. Cutkosky. On grasp choice, grasp models and the design of hands for manufacturing tasks. *IEEE Transactions on Robotics and Automation*, 5(3):269–279, 1989.

[9] K. Bernardin, K. Ogawara, K. Ikeuchi, and R. Dillmann. A sensor fusion approach for recognizing continuous human grasping sequences using hidden markov models. *IEEE Trassactions on Robotics,* 21(1):47–57, February 2005.

[10] S. Schulz, Ch. Pylatiuk, and G. Bretthauer. A new ultralight anthropomorphic hand. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, Seoul, Korea, 2001.

[11] S. Schulz, Ch. Pylatiuk, A. Kargov, R. Oberle, and G. Bretthauer. Progress in the development of anthropomorphic fluidic hands for a humanoid robot. In *The International Conference on Humanoid Robots*, Los Angeles, Nov 2004.

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

[13] 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*, pages 1010–1015, Sendai, Japan, September 2004.

[14] R. Platt Jr., A.H. Fagg, and R.A. Grupen. Extending fingertip grasping to whole body grasping. In *IEEE International Conference on Robotics and Automation,* pages 2677–2682, Taipei, Taiwan, September 2003.

[15] R. Platt Jr., O. Brock, A.H 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, October 2003.

[16] M.E. Cambron and R.A. Peters. Learning sensory motor coordination for grasping by a humanoid robot. In *IEEE*

International Conference on Systems, Man, and Cybernetics, volume 1, pages 6–13, Nashville, Tennessee, October 2000.

[17] A.T Miller and P.K Allen. Graspit!: A versatile simulator for robotic grasping. *IEEE Robotics & Automation Magazine*, 11(4):110–122, December 2004.

[18] Barrett Technology Inc. `http://www.barrett.com/`.

[19] A.T. Miller, S. Knoop, H.I. Christensen, and P.K Allen. Automatic grasp planning using shape primitives. In *IEEE International Conference on Robotics and Automation*, Taipe, Taiwan, Septmeber 2003.

[20] C. Ferrari and J. Canny. Planning optimal grasps. In *IEEE International Conference on Robotics and Automation*, pages 2290–2295, Nice, France, May 1992.

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

[22] S. Nayar, S. Nene, and H. Murase. Real-time 100 object recognition system. In *International Conference on Robotics and Automation (ICRA)*, volume 3, pages 2321–2325, Minneapolis, USA, 1996.

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

[24] 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)*, pages 93–102, Tokyo, Japan, 2003.

[25] 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.

[26] S. Obdrzalek and J. Matas. Object recognition using local affine frames on distinguished regions. In *British Machine Vision Conference (BMVC)*, volume 1, pages 113–122, Cardiff, UK, 2002.