Project no.:	**027657**
Project full title:	**Perception, Action & Cognition through learning of Object-Action Complexes**
Project Acronym:	**PACO-PLUS**
Deliverable no.:	**D1.1.1**
Title of the deliverable:	**A Humanoid hardware system equipped with basic sensory-motor primitives**

Contractual Date of Delivery to the CEC:	31. January 2008	
Actual Date of Delivery to the CEC:	31. January 2008	
Organisation name of lead contractor for this deliverable:	UniKarl	
Author(s): Tamim Asfour, Alex Bierbaum, Pedram Azad, Kai Welke and Rüdiger Dillmann		
Participant(s): UniKarl		
Work package contributing to the deliverable:	WP1.1	
Nature:	R/D	
Version:	Draft	
Total number of pages:	10	
Start date of project:	1$^{\text{st}}$ Feb. 2006	**Duration:** 48 month

Project co-funded by the European Commission within the Sixth Framework Programme (2002–2006)

Dissemination Level

PU	Public	X
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Abstract:

In this report, we present the ongoing work on the humanoid robot ARMAR-III toward the implementation of the PACO-PLUS cognitive control architecture on a humanoid platform. The software architecture of the ARMAR-III interface has been adapted to provide mechanisms, which allow integrating modules of the different partners in a flexible manner. Furthermore, the perceptual and manipulative abilities of ARMAR-III has been extended in order to perform complex tasks on the humanoid platform and enable acquiring of OACs on a humanoid platform. Furthermore, we describe the further development of the humanoid active head and the extensions of the its sensor system as well as the latest development of the tactile sensor for the hand and the improvements of the hand control system.

Keyword list: Humanoid robot ARMAR-III. Humanoid head with foveated active vision. Tactile sensor for a five-fingered hand.

Table of Contents

1. Executive Summary

An ongoing task throughout the project is the integration of the PACO-PLUS cognitive control architecture on the humanoid platform ARMAR-III at UniKarl. Within this task, the different modules of the proposed architecture are adapted to match the requirements, which follow from the low-level control mechanisms, the kinematic structure and the perceptual abilities of ARMAR-III. The software architecture of the ARMAR-III interface has been adapted to provide mechanisms, which allow integrating modules of the different partners in a flexible manner. Furthermore, the perceptual and manipulative abilities of ARMAR-III are continuously being extended in order to perform complex tasks on the humanoid platform [A]. Various skills has been developed to allow vision-based object localization, manipulation and graping of objects, collision detection and force-based control.

The developed humanoid head for foveated active vision has been improved in terms of robustness, user-friendliness and maintenance. The sensor system of the head consisting of two cameras per eye, one with a wide-angle lens for peripheral vision and one with a narrow-angle lens for foveal vision and six microphones has been extended by a gyroscope as a measurement unit for stabilization and control of the cameras. Two copies of the head have been built at UniKarl and delivered to JSI and KTH in September 2007. The heads are exact copies of the head used at UniKarl on the humanoid robot ARMAR-III [1]. The fact of using the same hardware with the same controller will lead to a further minimization of the overhead in the integration of several vision components in the PACO-PLUS robot platforms.

The first version of the artificial skin segments necessary for haptic exploration has been improved. In this context, a miniaturized, modular tactile sensor has been realized. Apart from the sensor electronics, a pad-shaped sensor actuation layer used to cover the sensor elements has been developed. Additionally, a revised version of the hardware necessary for data acquisition has been realized and successfully tested. The new skin segments have been evaluated on the five finger hand and the jaw gripper at UniKarl and SDU.

2. Work on the Humanoid Robot ARMAR-III

An ongoing task throughout the project is the integration of the PACO-PLUS cognitive control architecture as on the humanoid platform ARMAR-III at UniKarl. Within this task, the different modules of the proposed architecture are adapted to match the requirements, which follow from the low-level control mechanisms, the kinematic structure and the perceptual abilities of ARMAR-III. The software architecture of the ARMAR-III interface is adapted to provide mechanisms, which allow integrating modules of the different partners in a flexible manner. Furthermore the perceptual and manipulative abilities of ARMAR-III are continuously extended in order to perform complex tasks on the humanoid platform. Thees abilities has been extended by various skills, which allow vision-based object localization, manipulation and graping of objects, collision detection and force-based control.

The software and control architecture of the humanoid robot ARMAR-III consists of three layers (Figure 1). On the lowest level, digital signal processors (DSP) perform low-level sensorimotor control realized as cascaded velocity-position control loops. On the same level, hardware such as microphones, loudspeakers, and cameras are available. All these elements are connected to the PCs in the mid-level, either directly or via CAN bus. The software in the mid-level is realized using the Modular Controller Architecture (MCA2, www.mca2.org). The PCs in the mid-level are responsible for higher-level control (forward and inverse kinematics), the holonomic platform, or speech processing.

The first two levels can be regarded as stable i.e. the implemented modules remain unchanged. The programming of the robot, in particular for the partners, takes place on the highest level only. Here, the so-called

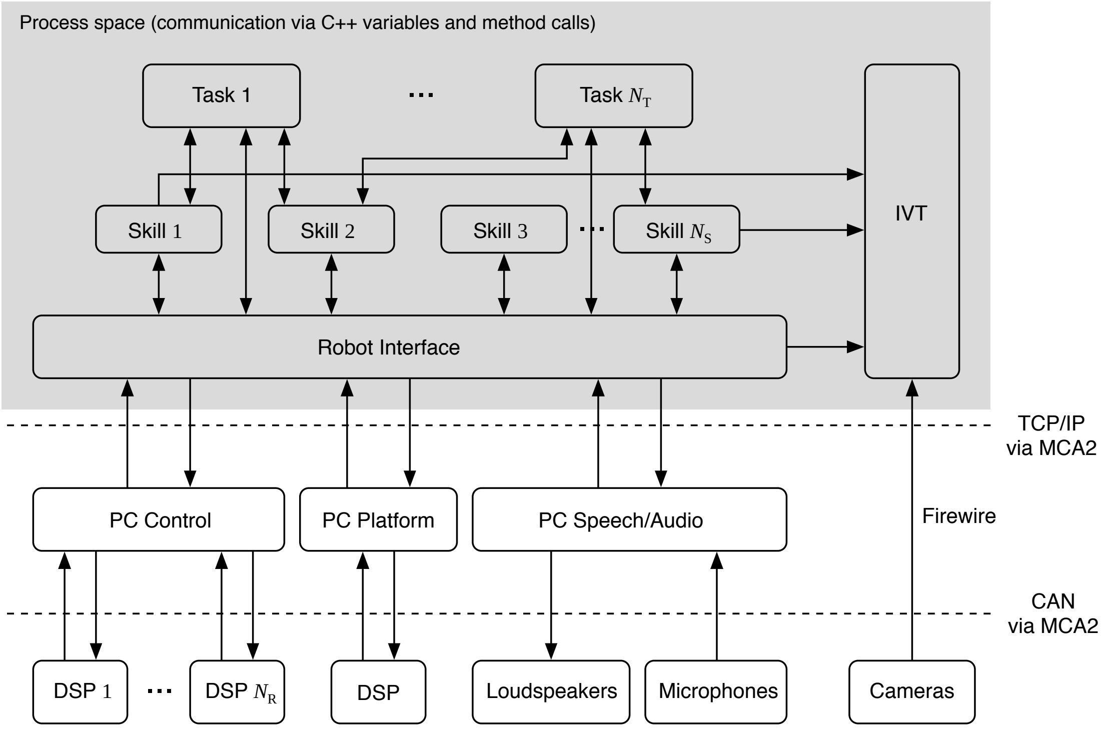


Figure 1: Software and control architecture of ARMAR-III.

robot interface (see also Deliverable D1.2, month 6) allows convenient access to the robots sensors and actuators via C++ variables and method calls. As an example, the robots head can be moved by different variants of the method MoveHead, e.g. using an inverse kinematics algorithm or setting all joint angles directly. At the same time, the sensor values for the current joint positions and velocities are available via variables. Access to the arms, the hip, and the platform are offered in the same manner.

To allow effective and efficient programming of the robot, in addition to direct access to the robots sensors and actors, two abstraction levels are defined: tasks and skills. While so far tasks have been implemented manually by hard coded combination of several skills for a specific purpose, the next step is to automatically generate tasks at run-time based on the output of the planning module. In contrast, skills are implemented capabilities of the robot that can be regarded as atomic. Currently, on ARMAR-III the following skills are available:

- `SearchForObjectSkill`: Searching for known objects using the active head. This skill scans the space in front of the robot by moving the head and performing a full scene analysis at each target position. The skill can be parameterized to either search for a specific object or include all object representations currently available in the database.

- `VisualGraspSkill`: Grasping of objects. This skill makes use of a visual servoing approach including one arm and the hip to grasp an object that has been previously recognized and localized. It controls the position closed-loop while continuously updating the position of the robots hand and the target object. The hand is tracked by using a marker.

- `PlaceSkill`: Placing objects on even surfaces. With this skill, it is possible to place a previously grasped object on an even surface, such as a table. Force information acquired from a 6D force sensor in the wrist is evaluated to determine the contact forces with the surface while placing the object.

- `HandOverSkill`: Handing over objects to or from the robot. With this skill, objects can be either handed over to the robot or received from the robot. In both cases, information acquired from a 6D force sensor is evaluated to determine when to close respectively open the hand.

- `OpenDoorSkill`: Opening various doors. This skill can be regarded as a more complex respectively higher-level skill compared to the previously introduced ones, since it makes use of other skills. It first uses a module that can recognize and localize handles of doors, which are then grasped making use of the VisualGraspSkill. Once the handle has been grasped, this skill opens the door using a force-control approach.
- `CloseDoorSkill`: Closing various doors. Using this skill, open doors can be closed, given an initial target position to touch the door of interest. Again, a force-control approach is used to perform the action.

All skills have in common a continuously called run method, in which they perform their action and sensing. The return value of this method signalized its state, i.e. whether the skill is still *operating*, finished with *success*, or finished with *failure*. After success or failure, a skill switches to the state waiting. Each skill can be parameterized and has its own specific configuration data structure for this purpose. In the same way, the result of a skill is communicated.

3. Further Development of the Humanoid Active Head

The developed humanoid head for foveated active vision has been improved in terms of robustness, user-friendliness and maintenance. The sensor system of the head consisting of two cameras per eye, one with a wide-angle lens for peripheral vision and one with a narrow-angle lens for foveal vision and six microphones has been extended by a gyroscope as a measurement unit for stabilization and control of the cameras.

Two copies of the head have been built at UniKarl and delivered to JSI and KTH in September 2007. The heads are exact copies of the head used at UniKarl on the humanoid robot ARMAR-III [1]. The fact of using the same hardware with the same controller will lead to a further minimization of the overhead in the integration of several vision components in the PACO-PLUS robot platforms. An overview on the motor, sensor and computational system of the head is given in Table 1.

3.1 Auditory System

The humanoid robot head has been equipped with a six channel microphone system for 3D localization of acoustic events. As sensors off-the-shelf miniature condensor microphones were selected. A sensor pair was placed at the ear locations in the frontal plane of the head. Another pair was placed on the median plane of the head at the vertical level of the nose, a sensor on the face side and a sensor at the back of the head. The third sensor pair was also mounted on the median plane but at the level of the forehead.

For each microphone a pre-amplifier with phantom power supply is required. These units are commercially not available in the required dimensions. Therefore, a miniature six channel condenser microphone pre-amplifier with integrated phantom power supply was developed as a single *printed circuit board* (PCB) with dimensions of only 70×40 *mm*. The amplified microphone signal is conducted to a multichannel sound card on the PC side. The acoustic sensor system proved high sensitivity for detecting acoustic events while providing a good signal to noise ratio. In preliminary experiments we successfully performed localization of acoustic events.

Table 1: Overview on the motor, sensor and computational systems of the active vision head.

Kinematics	3 DOF in the eyes arranged as common eyes tilt and independent eyes pan.
	4 DOF in the neck arranged as lower pitch, roll, yaw and upper pitch.
Actuator	DC motors and Harmonic Drives.
Vision system	Each eye is realized through two Point Grey Dragonfly color cameras in the extended version with a resolution of 640x480 at 30 Hz. (See www.ptgrey.com)
Auditory system	Six microphones (SONY ECMC115.CE7): two in the ears, tow in the front and two in the rear of the head.
Inertial system	Xsens MTIx gyroscope-based orientation sensor, which provides drift-free 3D orientation as well as 3D acceleration. (See www.xsens.com) Six microphones (SONY ECMC115.CE7): two in the ears, tow in the front and two in the rear of the head.
Universal Controller Module (UCoM)	Three UCoM units for motor control: The UCoM is a DSP-FPGA-based device which communicates with the embedded PCs via CAN-Bus. By using a combination of a DSP and a FPGA, a high flexibility is achieved. The DSP is dedicated to calculations and data processing whereas the FPGA offers the flexibility and hardware acceleration for special functionalities.
Control PC	Embedded PC with dual FireWire card and CAN card. Communication between the UCoMs and the PC 104 system via CAN bus.
Operation System	The embedded system is running under Linux, kernel 2.6.8 with Real Time Application Interface RTAI/LXRT-Linux (debian distribution)
Control Software	The basic control software is implemented in the Modular Controller Architecture framework MCA (www.mca2.org). The control parts can be executed under Linux, RTAI/LXRT-Linux, Windows or Mac OS and communicate beyond operating system borders.
	Graphical debugging tool (mcabrowser), which can be connected via TCP/IP to the MCA processes to visualize the connection structure of the control parts.
	Graphical User Interface (mcagui) with various input and output entities.
	Both tools (mcabrowser and mcagui) provide access to the interfaces and control parameters at runtime
Integrating Vision Toolkit [2]	Computer vision library which allows to start the development of vision components within minimum time and provides support for the operating systems Windows, Linux, and Mac OS. The library contains a considerable amount of functions and features like the integration of various cameras, generic and convenient application for calibrating single cameras and stereo camera systems, distortion correction and rectification, various filters and segmentation methods, efficient mathematical routines, especially for 3-D computations, stereo reconstruction, particle filter framework, platform-independent multithreading, convenient visualization of images and the integration of a library the development of QT Graphical User Interfaces.

3.2 Inertial System

Though the drives of the head kinematics are equipped with incremental encoders we decided to add a gyroscope-based orientation and heading reference sensor (see Figure 2).

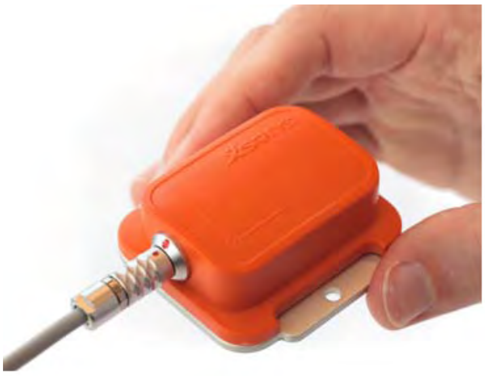


Figure 2: *MTIx* attitude and heading reference system.

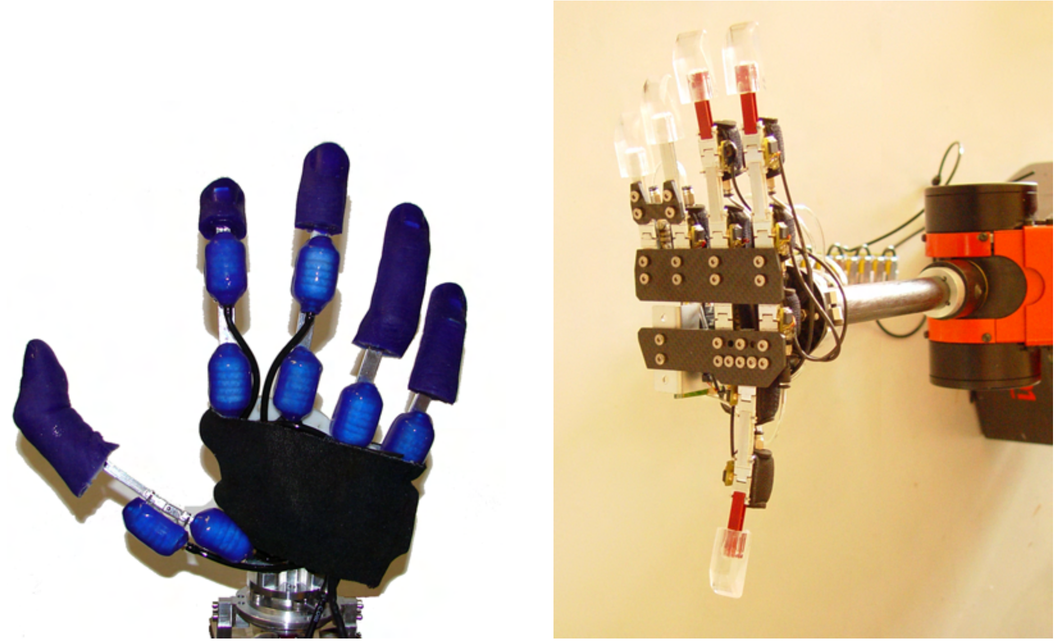
The sensor is an integrated attitude and heading reference system manufactured by the XSense company. It provides drift-free 6D orientation and acceleration measurement data and interfaces to a host PC (head control PC) via USB. The sensor will serve as a robot-equivalent sense of balance. It is especially useful for calibration and referencing of the head attitude and the detection of the body posture. In conjunction with the kinematic model and incremental encoder readings, partly redundant information about heading and orientation of the head is generated, which may further be used for diagnostics purposes. This is superior to the exclusive deployment of encoder readings as the kinematic model exposes uncertainty due to mechanical tolerances. Currently, the support of the attitude reference system in the head positioning control software is being implemented.

3.3 Kinematic Calibration of the Active Head-Eye System

The implementation of the necessary head modules, which allow using active vision methods is on ongoing task. In order to allow calibrated 3D perception in an open-loop application, an off-line procedure for the kinematic calibration of the active camera system has been developed. During the off-line procedure a calibration object is positioned in the front of the vision system. The joints of the head are moved independently in order to determine sequences of extrinsic camera calibrations for each joint. Based on these extrinsic calibrations, a target function is formulated, which is then minimized using a nonlinear least squares optimizer. The implemented method is based on work conducted in [4]. Modifications to the proposed method have been made to avoid systematic errors, to provide advanced calibration accuracy as well as to increase the flexibility considering the kinematic structure of the ARMAR-III head.

4. Further Development of the Humanoid Robot Hand

The humanoid robot ARMAR-III is equipped with two pneumatically actuated robot hands as introduced in [3]. The robot hand is equipped with joint position sensors, which measure the joint angles of all eight actuated finger joints (See Figure 4). The hand is controlled by a micro-controller subsystem via a serial RS232 interface. The original controller firmware supports basic position control of the finger joints and continuous position data reporting to a higher level control system.



(a) Original anthropomorphic robot hand. (b) Redesigned hand with parallel kinematics.

Figure 3: The original anthropomorphic and and the new hand with parallel kinematics.

4.1 Hand Kinematics

First investigations of several technical issues and initial grasping experiments revealed that the complex anthropomorphic shape of the fingertips is not suitable for stable grasping of objects. Also a precise flexion of the thumb could not be achieved with the predefined trajectories of thumb bending. The control of the thumb flexion is required as it presents a distinctive feature between different grasp classes. As a consequence, the hand kinematic was redesigned to meet these requirements. The new design is shown in Figure 3. Through the new kinematics structure, the thumb moves around the palm joint, which allows to bring the hand in configurations where the thumb and the remaining finger are parallel to each other. Our experiments show that the new robot hand is able to perform stable power, precision and hook grasps.

4.2 Hand Valve Array

Beside the kinematics also the pneumatic actuation mechanism of the robot hand has been improved. Originally, the eight joint actuators of the hand were actuated by a common inlet valve, a common outlet valve and an individual actuator control valve. The control firmware made use of a multiplexing strategy, which sequentially connected each actuator control valve to either the inlet our outlet valve. This concept kept the number of control valves and microcontroller I/O-ports low but results in an intolerable control dead time for the individual joint as only one joint at a time could be actuated. Due to the low transients in pneumatic technology, the control loop cycle time was limited to 600 *ms* per joint which is not satisfying for force and position control in grasping. As a consequence each joint actuator was equipped with an individual inlet and outlet valve, so that the control loop time could be decreased to 50 *ms* per joint. This also required the deployment of a different controller module that could operate the increased number of I/Os.

4.3 Hand Control and Pressure Sensing

Basic position control mode on the basis of a three step controller was delivered with the original firmware of the hand. Our experiments revealed that this did not exhibit the required robustness towards disturbance, i.e. load changes and non-linearity effects due to friction. Therefore, the new implementation of the low level control scheme has been realized. As the pneumatic robot hand has a compliant mechanical design, a stiff position control behavior is not favored. A combined force-position controller is clearly better suited to the demands in prehensile grasping and haptic exploration. A controller of this type may be implemented, e.g. as impedance control or cascaded control scheme. In any case force feedback information is necessary. For the pneumatic robot hand this could be achieved easily by measuring the pressure of the joint actuators. Therefore, miniature pressure sensors for the actuators were integrated and the hand firmware was enhanced to deal with the new functionality. By doing this, we were able to measure both joint force and position simultaneously. Currently, the realization of the hybrid position/force control scheme for the individual finger joints is under development. The additional pressure sensors also bring advantages for haptic sensing and system diagnostics. They will be used as additional proprioceptive sensors to measure external forces induced by touch. Furthermore, they may serve as diagnostic indicators in case of a leakage in the pneumatic system.

4.4 Tactile Sensors

The development and integration of the tactile sensors has made considerable progress during the last year (see also D.4.1.2). In cooperation with SDU, we have developed versatile encapsulated miniature sensor modules, which can be manufactured easily (see Figure 4). Two FSR-based cursor navigation sensors are mounted to a carrier board, each containing four force sensitive fields. The sensitive area is covered with a sensor actuation layer made of rubber plastic. The layer serves as a force concentrator, which leads to a significantly increasing of the sensitivity of the sensor system.

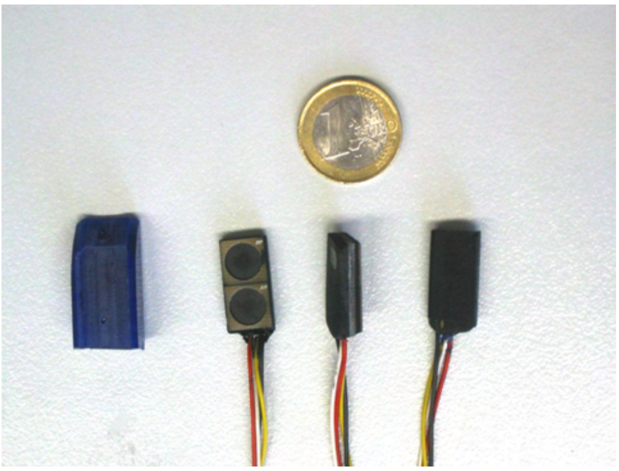


Figure 4: Versatile miniature tactile sensors (right). Fingertip of the five-fingered hand (left).

The sensor modules possess a serial I^{2}C-bus interface [5], which offers easy interconnectivity to higher level control systems while maintaining a low number of wiring cables. The modules require only two wires for power supply and two wires for communication. As communication is based on a bus system, the modules can be easily connected in row. Three of the modules have been integrated in the fingertips of the robot hand.

Attached Papers

[A] T. Asfour, P. Azad, N. Vahrenkamp, K. Regenstein, A. Bierbaum, K. Welke, J. Schröder, and R. Dillmann. Toward humanoid manipulation in human-centred environments. *Robotics and Autonomous Systems*, 56(11):54–65, 2008.

References

[1] T. Asfour, K. Regenstein, P. Azad, J. Schröder, N. Vahrenkamp, and R. Dillmann. ARMAR-III: An Integrated Humanoid Platform for Sensory-Motor Control. In *humanoids*, 2006.

[2] P. Azad. Integrating Vision Toolkit. `http://ivt.sourceforge.net`.

[3] A. Kargov, T. Asfour, C. Pylatiuk, R. Oberle, H. Klosek, S. Schulz, K. Regenstein, G. Bretthauer, and R. Dillmann. Development of an anthropomorphic hand for a mobile assistive robot. In *Rehabilitation Robotics, 2005. ICORR 2005. 9th International Conference on*, pages 182–186, 28 June-1 July 2005.

[4] M. Li. Kinematic calibration of an active head-eye system. In *IEEE Transactions on Robotics and Automation*, volume 14, pages 153–158, 1998.

[5] NXP (former Philips Semiconductors). *I^2C-bus specification and user manual*, Rev. 03, 19th June 2007. http://www.nxp.com.

ARTICLE IN PRESS

Available online at www.sciencedirect.com

ScienceDirect

Robotics and Autonomous Systems ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Robotics and Autonomous Systems

www.elsevier.com/locate/robot

Toward humanoid manipulation in human-centred environments

T. Asfour*, P. Azad, N. Vahrenkamp, K. Regenstein, A. Bierbaum, K. Welke, J. Schröder, R. Dillmann

University of Karlsruhe, Institute for Computer Science and Engineering (CSE/IAIM), P.O. Box 6980, D-76128 Karlsruhe, Germany

Abstract

In order for humanoid robots to enter human-centred environments, it is indispensable to equip them with manipulative, perceptive and communicative skills necessary for real-time interaction with the environment and humans. The goal of our work is to provide reliable and highly integrated humanoid platforms which on the one hand allow the implementation and tests of various research activities and on the other hand the realization of service tasks in a household scenario. In this paper, we present a new humanoid robot currently being developed for applications in human-centred environments. In addition, we present an integrated grasping and manipulation system consisting of a motion planner for the generation of collision-free paths and a vision system for the recognition and localization of a subset of household objects as well as a grasp analysis component which provides the most feasible grasp configurations for each object.

Keywords: Mechatronics; Control architecture; Motion planning; Object recognition and localization; Grasp planning

1. Introduction

Our current research interest is the development of humanoid robots which safely coexist with humans, interactively communicate with humans and usefully manipulate objects in built-for-human environments. In particular, we address the integration of motor, perception and cognition components such as multimodal human–humanoid interaction and human–humanoid cooperation in order to be able to demonstrate robot manipulation and grasping tasks in a kitchen environment as a prototypical human-centred one [11]. Recently, considerable research work has been focused on the development of humanoid biped robots [9,1,14,27,24,4]. However, in order for humanoid robots to enter human-centred environments, it is indispensable to equip them with manipulative, perceptive and communicative skills necessary for real-time interaction with the environment and humans. The goal of our work is to provide reliable and highly integrated humanoid platforms which on the one hand allow the implementation and tests of various research and on the other hand the realization of manipulation and grasping tasks in a household scenario.

The paper is organized as follows. In Section 2, we describe the different components of the humanoid robot, its kinematics and sensor systems. Section 3 describes the control architecture including its hardware and software modules. The motion planning algorithms for generating of collision-free paths are described in Section 4. In Section 5, the developed and implemented vision algorithms for object recognition and localization are described. The grasp analysis system which provides the most feasible grasp configurations for each object is presented in Section 6.

2. The humanoid robot ARMAR-III

In designing our robot, we desire a humanoid that closely mimics the sensory and sensory-motor capabilities of the human. The robot should be able to deal with a household environment and the wide variety of objects and activities encountered in it. Therefore, the humanoid robot ARMAR-III (see Fig. 1) has been designed under a comprehensive view so that a wide range of tasks (and not only a particular task) can be performed. The upper body of the robot has been designed to be modular and lightweight while retaining similar size and

* Corresponding author.
E-mail address: asfour@ira.uka.de (T. Asfour).
URL: http://wwwiaim.ira.uka.de (T. Asfour).

0921-8890/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.robot.2007.09.013

Please cite this article in press as: T. Asfour, et al., Toward humanoid manipulation in human-centred environments, Robotics and Autonomous Systems (2007), doi:10.1016/j.robot.2007.09.013

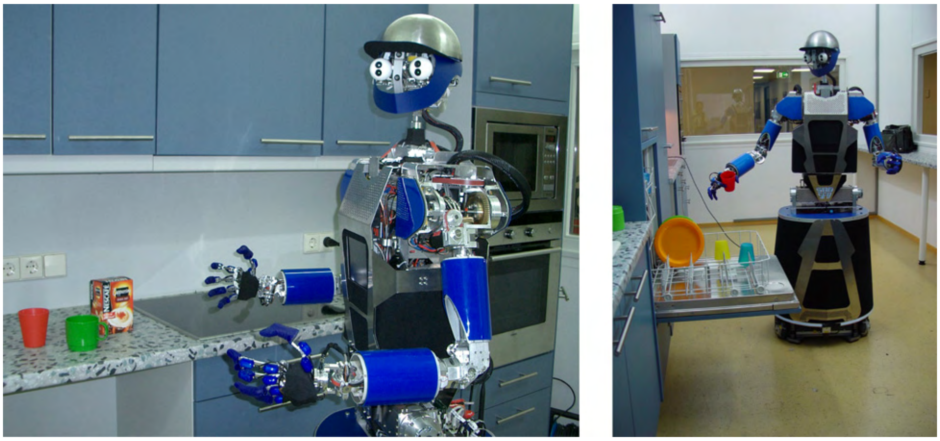


Fig. 1. The humanoid robot ARMAR-III with an active head with foveated vision, two arms and two five-fingered hands and a holonomic mobile platform.

Table 1
Specification of ARMAR-III

Weight		135 kg (incl. 60 kg battery)	
Height		175 cm	
Speed		1 m/s	
DOF	Eyes	3	Common tilt and independent pan
	Neck	4	Lower Pitch, Roll, Yaw, upper Pitch
	Arms	2 × 7	3 DOF in each shoulder, 2 DOF in each elbow, and 2 in each wrist
	Hands	2 × 8	Five-fingered hands with 2 DOF in each Thumb, 2 DOF in each Index and Middle, and 1 DOF in each Ring and Pinkie.
	Toros	3	Pitch, Roll, Yaw
	Platform	3	3 wheels arranged in angles of 120°
Actuator		DC motors + Harmonic Drives in the arms, neck, eyes, torso and platform. Fluidic actuators in the hand.	
Sensors	Eyes	2 Point Grey (www.ptgrey.com) Dragonfly cameras in each eye, six microphones and a 6D inertial sensor (http://www.xsens.com).	
	Arms	Motor encoders, axis sensors in each joint, torque sensors in the first five joints and 6D force–torque sensor (http://www.ati-ia.com) in the wrist.	
	Platform	Motor encoders and 3 Laser-range finders (http://www.hokuyo-aut.jp).	
Power supply		Switchable 24 V Battery and 220 V external power supply.	
Operating system		Linux with the Real-Time Application Interface RTAI/LXRT-Linux.	
Computers and communication		Industrial PCs and PC/104 systems connected via Gigabit Ethernet and 10 DSP/FPGA control units (UCoM) which communicate with the control PC via CAN bus.	
User interface		Graphical user interface (GUI) connected to the robot via wireless LAN and natural speech communication.	

proportion as an average person. For the locomotion, we use a mobile platform which allows for holonomic movability in the application area. From the kinematics control point of view, the robot consists of seven subsystems: head, left arm, right arm, left hand, right hand, torso and a mobile platform. The specification of the robot is given in Table 1. In the following the subsystems of the robot are briefly described. For detailed information the reader is referred to [4].

Head: The head has seven DOF and is equipped with two eyes. The eyes have a common tilt and can pan independently. Each eye is equipped with two digital colour cameras, one with a wide-angle lens for peripheral vision and one with a narrow-angle lens for foveal vision to allow simple visuo-motor behaviours such as tracking and saccadic motions towards salient regions, as well as more complex visual tasks such as hand-eye coordination. The visual system is mounted on a four DOF neck mechanism [2] (lower pitch, roll, yaw, upper pitch). For the acoustic localization, the head is equipped with a microphone array consisting of six microphones (two in the ears, two in the front and two in back of the head). Furthermore, an inertial sensor is installed in the head for stabilization control of the camera images.

Upper body: The upper body of the robot provides 33 DOF: 14 DOF for the arms and three DOF for the torso. The arms are designed in an anthropomorphic way: three DOF in the shoulder, two DOF in the elbow and two DOF in the wrist. Each arm is equipped with a five-fingered hand with eight DOF (see [29]). In order to achieve a high degree of mobility and to allow simple and direct cooperation with humans, the structure (size, shape and kinematics) of the arms should has been designed to be similar to that of the human arm. The goal of performing manipulation tasks in human-centred environments generates a number of requirements for the sensor system, especially for that of the manipulation system. Each joint of

Please cite this article in press as: T. Asfour, et al., Toward humanoid manipulation in human-centred environments, Robotics and Autonomous Systems (2007), doi:10.1016/j.robot.2007.09.013

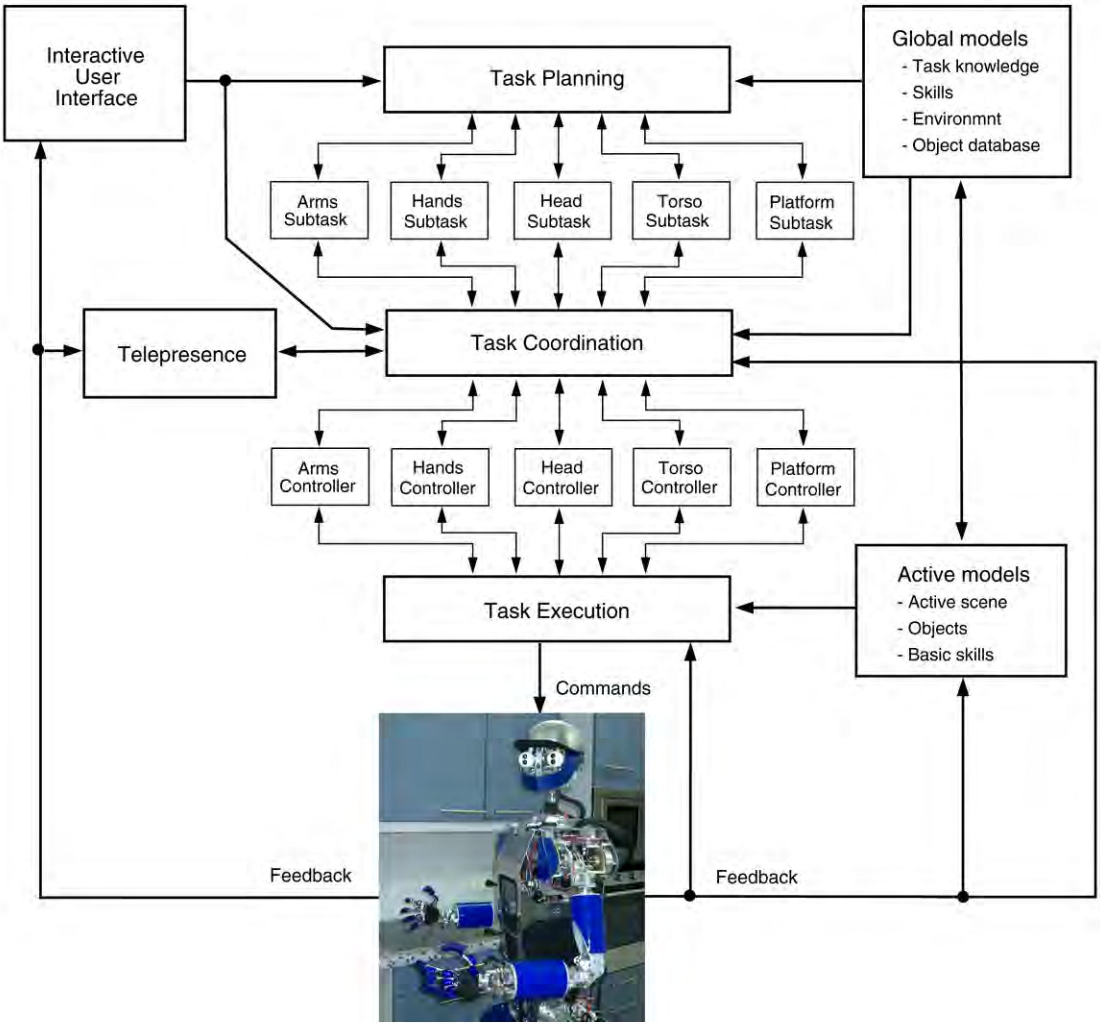


Fig. 2. Hierarchical control architecture for coordinated task execution in humanoid robots: planning, coordination and execution level.

the arms is equipped with motor encoder, axis sensor and joint torque sensor to allow position, velocity and torque control. In the wrists 6D force/torque sensors are used for hybrid position and force control. Four planar skin pads [15] are mounted to the front and back of each shoulder, thus also serving as a protective cover for the shoulder joints. Similarly, cylindrical skin pads are mounted to the upper and lower arms respectively.

Mobile platform: There are several requirements for the locomotion system of a humanoid robot: Mobility which is necessary to extend the workspace of the robot and stability which is most essential to insure humans safety. Therefore, the locomotion of the robot is realized using wheel-based holonomic platform, which allows for a high flexibility in our kitchen application area. The holonomic locomotion is obtained by using wheels with passive rolls at the circumference. Such wheels are known as Mecanum wheels or Omniwheels. In addition, a spring-damper combination is used to reduce vibrations.

The sensor system of the platform consists of a combination of three Laser-range-finders (Laser-scanner) and optical encoders to localize the platform. The scanners are placed at the bottom of the base plate 120° to each other. A scan range of 240° per sensor allows complete observation of the environment. The maximum scan distance is 4 m. A low scan plane of 60 mm was chosen due to safety reasons to detect small objects and foot tips. Optical encoders deliver a feedback about the actual wheel speeds to the speed control, and serve as a second input, together with the scanner data, to a Kalman–Filter which estimates the position of the platform. The platform hosts the power supply and the main part of the robot computer system.

3. Robot control architecture

The control architecture is structured into the three following levels: a task planning level, a synchronization and coordination level and a sensor-motor level (see Fig. 2). A given task is decomposed into several subtasks. These represent sequences of actions the subsystems of the robot must carry out to accomplish the task goal. The coordinated execution of a task requires the scheduling of the subtasks and their synchronization with logical conditions, external and internal events [3]. Fig. 2 shows the block diagram of the control architecture with three levels, global and active models and a multimodal user interface.

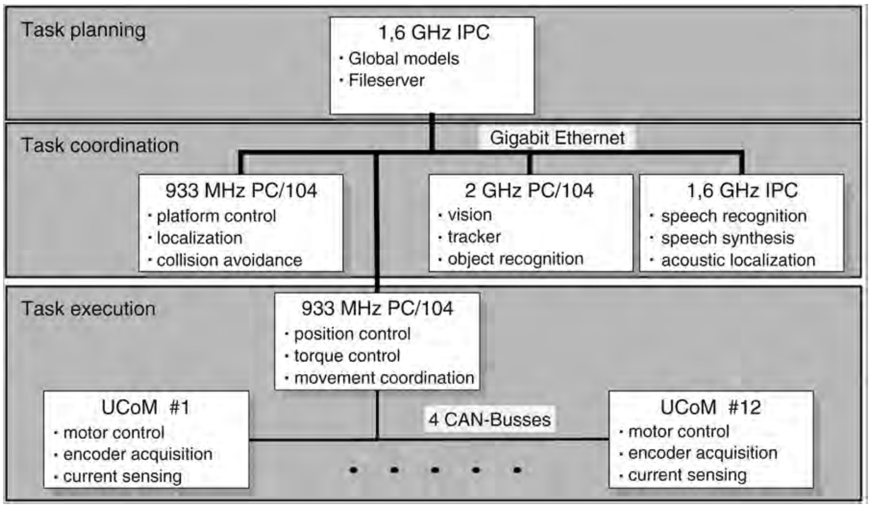


Fig. 3. The computer architecture: The used hardware is based on industrial standards and the developed Universal Controller Module (UCoM).

- The task planning level specifies the subtasks for the multiple subsystems of the robot. This level represents the highest level with functions of task representation and is responsible for the scheduling of tasks and management of resources and skills. It generates the subtasks for the different subsystems of the robot autonomously or interactively by a human operator. The generated subtasks for the lower level contain the whole information necessary for the task execution, e.g. parameters of objects to be manipulated in the task or the 3D information about the environment. According to the task description, the subsystem's controllers are selected here and activated to achieve the given task goal.
- The task coordination level activates sequential/parallel actions for the execution level in order to achieve the given task goal. The subtasks are provided by the task planning level. As it is the case on the planning level the execution of the subtasks in an appropriate schedule can be modified/reorganized by a teleoperator or user via an interactive user interface.
- The task execution level is characterized by control theory to execute specified sensory-motor control commands. This level uses task specific local models of the environment and objects. In the following we refer to those models as *active models*:
- The active models (*short-term memory*) play a central role in this architecture. They are first initialized by the global models (*long-term memory*) and can be updated mainly by the perception system. The novel idea of the active models, as they are suggested here, is the ability for the independent actualization and reorganization. An active model consists of the internal knowledge representation, interfaces, inputs and outputs for information extraction and optionally active parts for actualization/reorganization (update strategies, correlation with other active models or global models, learning procedure, logical reasoning, etc.).
- The user interface provides in addition to graphical user interfaces (GUIs) the possibility for interaction using natural language. Telepresence techniques allow the operator to supervise and teleoperate the robot and thus to solve exceptions which can arise from various reasons.

Internal system events and execution errors are detected from local sensor data. These events/errors are used as feedback to the task coordination level in order to take appropriate measures. For example, a new alternative execution plan can be generated to react to internal events of the robot subsystems or to environmental stimuli.

Computer architecture: The control architecture described in Section 3 are realized using embedded Industrial PCs, PC/104 systems and DSP/FPGA modules, so called UCoM (Universal Controller Module), which are responsible for the sensory-motor control. The PCs are connected via switched Gigabit Ethernet whereas the communication between the UCOMs and the control PC is realized using four CAN buses to fulfil real-time requirements of the task execution level. The connection to a user interface PC is established by wireless LAN. An overview over the structure of the computer architecture is given in Fig. 3. The requirements of the task planning and task coordination levels could be fulfilled with embedded Industrial PCs and PC/104 systems. The requirements for the execution level could not be met with off-the-shelf products. Therefore, new control units (UCoM) consisting of a combination of a DSP and an FPGA on one board have been developed. For more details about the control boards can be found in [4].

Software environment: The computers are running under Linux with the Real-Time Application Interface RTAI/LXRT-Linux. For the implementation of the control architecture we have used the framework MCA.[1] It provides a standardized module framework with unified interfaces. The modules can be easily connected into groups to form more complex functionality. These modules and groups can be executed under Linux, RTAI/LXRT-Linux, Windows or Mac OS and communicate beyond operating system borders. Moreover, graphical debugging tools can be connected via TCP/IP to the

[1] www.mca2.org.

MCA processes, which visualize the connection structure of the modules and groups. These tools provide access to the interfaces at runtime and a graphical user interface with various input and output entities.

4. Collision-free motion planning

A motion planner which can be used in a real-time environment needs to accomplish several requirements. The planner should be fast and the planned trajectories should be adapted to a changing environment. Previous work that address the problem of dynamic environments like [26] and [8] suffers from several significant shortcomings and drawbacks. With these approaches it is possible to realize a planner that is able to react on dynamic obstacles, but they are not practical for highly redundant robot systems like humanoid robots. To deal with the complexity of motion planning problems we rely on a multiresolutional planning system that is able to task-dependently combine different planning algorithms with varying detail levels of the robot. It is clear that a path planning algorithm for a mobile platform can use a low resolution for the hand models, e.g. by turning off the kinematic chain and regarding the complete hand as one joint with a bounding box. On the other hand, in the case of dexterous manipulation and grasping tasks a higher resolution model of the hand is necessary. In order to robustly execute the planned trajectories, the visibility of the target objects is considered in the planning phase. Therefore, the expected perception of a target object is calculated by simulating the camera output and thus biasing the RRT-based search toward regions where the robot will have good visibility [19].

Guaranteeing collision-free paths: Since the high number of degrees of freedom, our motion planning approaches use sampling-based algorithms according to the Rapidly-Exploring-Trees (RRTs) from LaValle and Kuffner [17,16]. In all sampling-based approaches, the sampling resolution of the configuration space (C-space) can be specified with a resolution parameter. The choice of the resolution parameter affects the quality of the result as well as the runtime of the algorithm. If the resolution is too high, the runtime will be unnecessarily long. On the other hand, with a low resolution, the planner will run fast but might not consider some obstacles. Another problem, that arises from sampling the C-space, is to guarantee the collision-free status of a path between two configurations. Regardless which sampling resolution is chosen, there is no guarantee that the path between two neighbouring samples is collision-free [26,32].

To overcome this problem Quinlan has introduced in [26] an approach, which can be used to guarantee a collision-free path between two C-space samples. Quinlan calculates bubbles of free space around a configuration and therefore can guarantee a collision-free path segment by overlapping these bubbles along the segment. To retrieve the radius of the free bubbles, the Quinlan method needs the minimum obstacle distance of the robot in workspace. These calculations are time-consuming and slow down the planning process, since a lot of distance calculations are needed for path validation.

Using enlarged robot models: The long runtime of the free bubble approach arises from the high number of workspace distance calculations. With the enlarged model approach we apply a method to guarantee a collision-free status of a path without any distance computations. This results in a faster path validation and thus in a speedup of the planning algorithm. The enlarged models are constructed by slightly scaling up the convex 3D models of the robot so that the minimum distance between the surfaces of the original and the enlarged model reaches a lower bounding $d_{\text{freespace}}$. Fig. 4 shows the original collision model of the right arm and the transparent enlarged models ($d_{\text{freespace}} = 20$ mm).

Planning with enlarged robot models: When using the enlarged models for collision checking and the collision checker reports a collision-free situation, a lower bound for the obstacle distance of the original collision models can be calculated. We can avoid the time-consuming distance calculations by setting the obstacle distance to this lower bound. Using the lower bound for the distance results in smaller free bubble radii and thus in more sampling calculations along a path segment. However, this overhead is compensated by avoiding time-consuming distance calculations.

Lazy collision checking: In [28], a lazy collision checking approach was presented, in which the collision checks for C-space samples (milestones) and path-segments are decoupled. We adapt the idea of lazy collision checking to speed up the planning process and introduce a two-step planning scheme [32]. In the first step the normal sampling-based RRT algorithm searches a solution path in the C-space. This path is known to be collision-free at the path points, but the path segments between these points could result in a collision. In the second validation step we use the enlarged model approach to check the collision status of the path segments of the solution path. If a path segment between two configurations c_i and c_{i+1} fails during the collision test, we try to create a local detour by starting a subplanner which searches a way around the C-space obstacle (see Fig. 4). Thus we do not guarantee the complete RRT to be collision-free on creation, instead we try to give a collision-free guarantee of the sampling-based solution afterward and reduce the costly guarantee checks to the path segments.

Results: In Fig. 5 the planning setup is shown where the planner has to find a trajectory for the right arm of ARMAR III. The task of moving the arm with seven DOF from the left to the right cupboard leads to situations where the robot has low workspace clearance to operate. Therefore 6% of the solution paths, generated by the purely sampling-based planner, result in collisions in workspace. A planner using free bubbles to guarantee the collision free execution of the solution increases the average planning time from 3 to over 8 s. By using the lazy collision check approach it is possible to find a guaranteed collision-free solution in 2.5 s which is sufficient for real world applications.

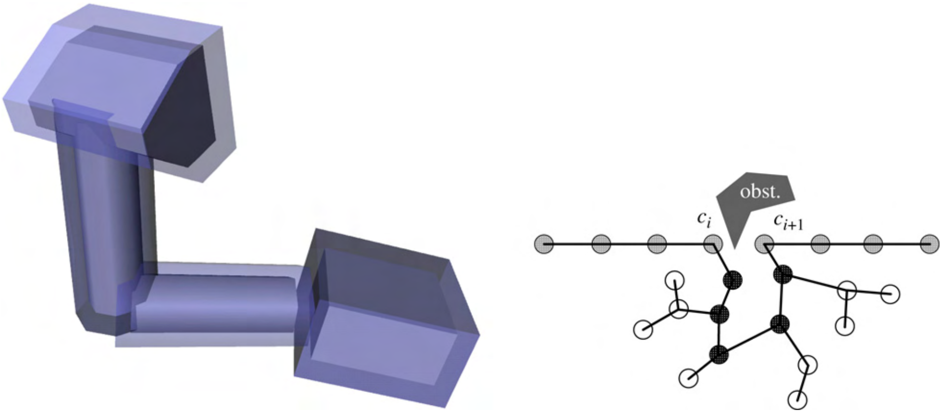


Fig. 4. Motion planning using enlarged robot models. Simplified 3D model and enlargement of the right arm (left) and the validated collision path (right).

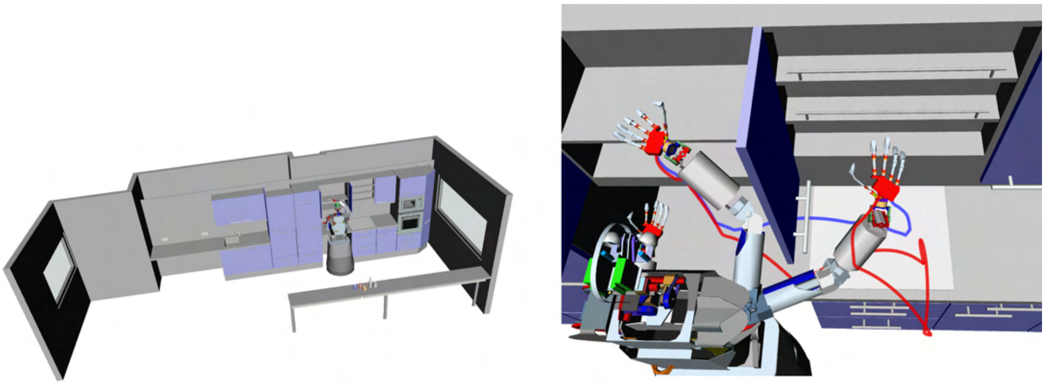


Fig. 5. The planning environment with the robot (left). Start and goal configuration of the arm with the solution paths (right).

5. Object recognition and localization

To allow the robot to perform the intended tasks in a household environment, it is crucial for the robot to perceive his environment visually. In particular, it must be able to recognize the objects of interest and localize them with a high enough accuracy for grasping. For the objects in the kitchen environment, which we use for testing the robot's skills, we have developed two object recognition and localization systems for two classes of objects: objects that can be segmented globally, and objects exhibiting a sufficient amount of texture, allowing the application of methods using local texture features.

Among the first class of objects are coloured plastic dishes, which we chose to simplify the problem of segmentation, in order to concentrate on complicated tasks such as the filling and emptying of the dishwasher. Among the second class of objects are textured objects such as tetrapacks, boxes with any kind of food, or bottles, as can be found in any kitchen.

5.1. Recognition and localization based on shape

In the following, we give an outline of our approach for shape-based object recognition and localization, in which appearance-based methods, model-based methods and stereo vision are combined. A 3D model of the object is used for generating multiple views. A detailed description is given in [5].

Segmentation: For the proposed shape-based approach, the objects have to be segmented. In the presented examples, this is done by performing colour segmentation in HSV colour space for coloured dishes. In order to use stereo vision, segmentation is performed for the left and the right image. The properties of the resulting blobs are represented by the bounding box, the centroid of the region and the number of pixels being part of the region. Using this information together with the epipolar geometry, the correspondence problem can be solved efficiently and effectively.

Region processing pipeline: Before a segmented region can be used as input for appearance-based calculations it has to be transformed into a normalized representation. For application of Principle Component Analysis (PCA), the region has to be normalized in size. This is done by resizing the region to a squared window of 64 × 64 pixels with bilinear interpolation while keeping the aspect ratio of the region. In the second step, the gradient image is calculated for the normalized window, which leads to a more robust matching procedure, as shown in [5]. Finally, in order to achieve invariance to constant multiplicative illumination changes, the signal energy of each gradient image I is normalized (see [23,5]) to achieve invariance to variations in the embodiment of the edges.

6D localization: Ideally, for appearance-based 6D localization with respect to a rigid object model, for each object training views would have to be acquired in the complete six dimensional space i.e. varying orientation and position. However, in practice it is not possible to solve the problem in this six dimensional space directly within adequate time.

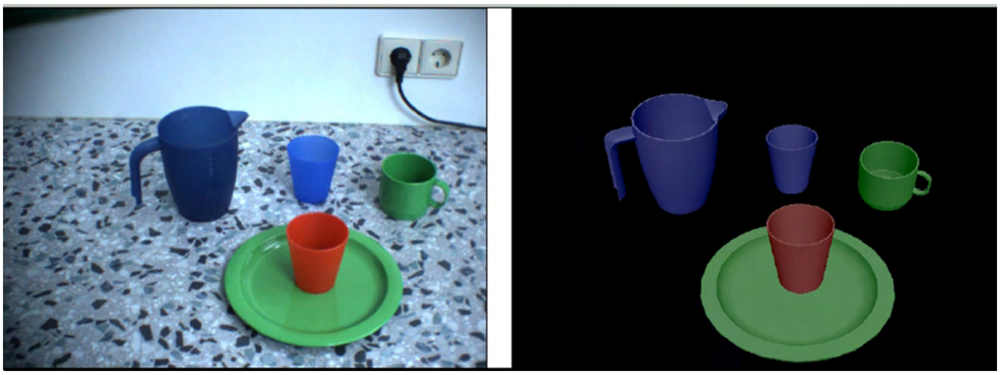


Fig. 6. Typical result of a scene analysis. Input image of the left camera (left) and 3D visualization of the recognition and localization result (right).

Therefore, we solve the problem by calculating the position and the orientation independently in first place. A first estimate of the position is calculated by triangulating the centroids of the colour blobs. A first estimate of the orientation is retrieved from the database for the matched view. Since the position influences the view and the view influences the position of the centroids, corrective calculations are performed afterwards. Details are given in [5].

Convenient acquisition and real-time recognition: A suitable hardware setup for the acquisition of the view set for an object would consist of an accurate robot manipulator and a stereo camera system. However, the hardware effort is quite high, and the calibration of the kinematic chain between the head and the manipulator has to be known for the generation of accurate data. Therefore, we have used a 3D model of the object to generate the views. By using an appearance-based approach for a model-based object representation in the core of the system, it is possible to recognize and localize the objects in a given scene in real time—which is by far impossible with a purely model-based method, as explained in [5]. To achieve real-time performance, we use PCA to reduce dimensionality from $64 \times 64 = 4096$ to 100. 3D models of rather simple shapes can be generated manually. For more complicated objects we use the interactive object modelling centre presented in [7]. In Fig. 6, typical result of a scene analysis with the input images and the 3D visualization of the recognition and localization are shown.

5.2. *Recognition and localization based on texture*

In the following, we present our system for the recognition and localization of textured objects, which builds on top of the approach proposed in [18]. Details, in particular of our 6D localization approach using stereo vision, are given in [6].

Feature calculation: Various texture-based 2D point features have been proposed in the past. One has to distinguish between the calculation of feature points and the calculation of the feature descriptor. A feature point itself is determined by the 2D coordinates (u, v). Since different views of the same image patch around a feature point vary, the image patches can not be correlated directly. The task of the feature descriptor is to achieve a sufficient degree of invariance with respect to the potentially differing views. In general, such descriptors are computed on the base of a *local* planar assumption.

We have tested three different features respectively descriptors: Shi-Tomasi features and representing a patch by a view set [22,33], the Maximally Stable Extremal Regions (MSER) in combination with the Local Affine Frames (LAF) as presented in [25], and the SIFT features [18]. Our experiences with these features are described in [6].

The best results could be achieved with the SIFT features. The SIFT descriptor is fully rotation invariant and invariant to skew and depth to some degree. The feature information used in the following is the position (u, v), the rotation angle φ and a feature vector $\{\mathbf{x}_j\}$ consisting of 128 floating point values. These feature vectors are matched using a cross correlation. As the SIFT features are gradient based, sharp input images with high contrast lead to more features of high quality.

Object recognition: Given a set of n features $\{u_i, v_i, \varphi_i, \{\mathbf{x}_j\}_i\}$ with $i \in \{1, \ldots, n\}$ and $j \in \{1, \ldots, 128\}$ that have been calculated for an input image, the first task is to recognize which objects are present in the scene. Simply counting the features does not lead to a robust system since the number of wrong matches increases with the number of objects. Therefore, it is necessary to incorporate the feature positions with respect to each other into the recognition process. The state-of-the-art technique for this purpose is the general Hough transform. We use a two dimensional Hough space with the parameters u, v; the rotative information φ is used within the voting formula, as described in [6]. After the voting procedure, instances of an object in the scene are represented by maxima in the Hough space.

2D localization: After having found an instance of an object, the feature correspondences for this object are filtered by considering only those ones that have voted for this instance. For these correspondences (see Fig. 7), first, an affine transformation is calculated with a least-squares method in an iterative procedure. After the final iteration, a full homography is calculated with the remaining correspondences to achieve maximum accuracy. Using the homography instead of the affine transformation throughout the whole iterative procedure does not lead to a robust system, since the additional two degrees of freedom make the least squares optimization too sensitive to outliers.

Please cite this article in press as: T. Asfour, et al., Toward humanoid manipulation in human-centred environments, Robotics and Autonomous Systems (2007), doi:10.1016/j.robot.2007.09.013

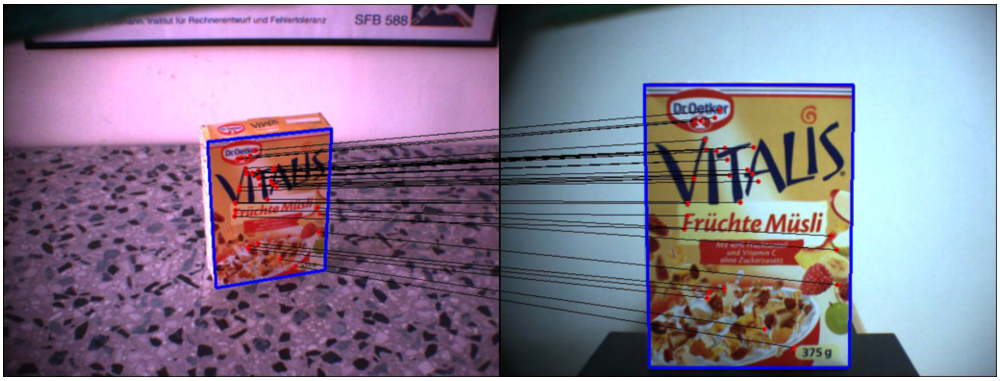


Fig. 7. Correspondences between current view of the scene and training image. Only the valid features after the filtering process are shown. The blue box illustrates the result of 2D localization. Input image (left) and training image (right). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

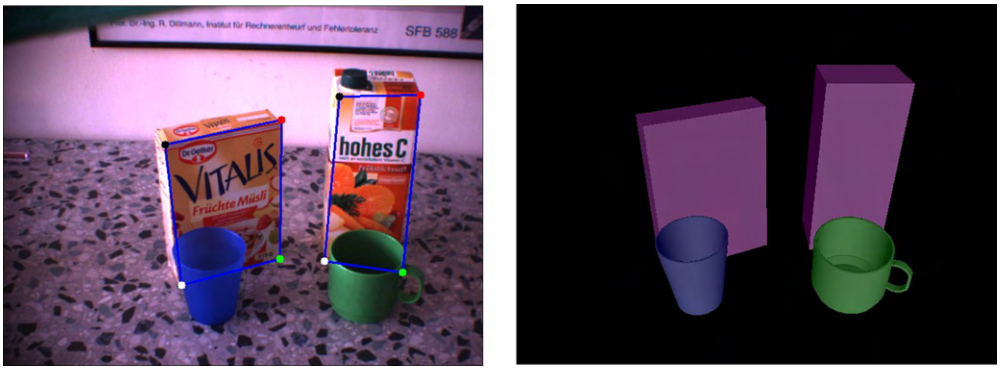


Fig. 8. Recognition and localization result for an exemplary scene. Input image (left) and 3D visualization of the result (right).

6D localization: The state-of-the-art technique for 6D localization is to calculate the pose based on the correspondences between 3D model coordinates and image coordinates from one camera image. This is usually done by using the POSIT algorithm [10] or similar methods. The drawback is that the correctness of the calculated pose depends on the accuracy of the 2D correspondences only. In particular, the depth information is very sensitive to small errors in the 2D coordinates of the correspondences. The smaller the area is that the matched features span in relation to the total area of the object, the greater this error becomes. The inaccurate calculated homography for the right object in Fig. 8 (left) illustrates this circumstance. However, for a successful grasp, accurate depth information is crucial. Therefore, our strategy is to make explicit use of the calibrated stereo system in order to calculate depth information with maximum accuracy. Our approach for cuboids consists of the following steps:

- Determine highly textured points within the calculated 2D contour of the object in the left camera image by calculating Shi-Tomasi features [30] (which produces more suitable features than SIFT for correlation in a standard stereo setup, since scale invariance is not verified and not necessary).
- Determine correspondences with subpixel-accuracy in the right camera image for the calculated points by using Zero Mean Cross Correlation (ZNCC) in combination with the epipolar geometry.
- Calculate a 3D point for each correspondence.
- Fit a 3D plane into the calculated 3D point cloud.
- Calculate the intersections of the four 3D lines through the 2D corners in the left camera image with the 3D plane.

The result of this algorithm are the 3D coordinates of the four corners of the object's front surface, given in the world coordinate system. Occlusions are handled by performing the fitting of the 3D plane with a RANSAC algorithm [12]. To offer the same interface as for the subsystem presented in Section 5.1, the 6D pose must be determined on the base of the calculated 3D corner points. For this purpose, a simple but yet accurate 3D model of a cuboid for the object is generated manually. The pose of this model with respect to the static pose stored in the file is determined by calculating the optimal transformation between the calculated 3D corner points and the corresponding 3D corner points from the 3D model. This is done by using the method proposed in [13].

6. Programming of grasping and manipulation tasks

The central idea of our approach for the programming and execution of manipulation tasks is the existence of a database with 3D models of all the objects encountered in the robot workspace and a 3D model of the robot hand. This allows for an extensive offline analysis of the different possibilities to grasp an object, instead of focusing on fast online approaches. From this central fact we have developed an integrated grasp planning system, which incorporates a vision system for the localization

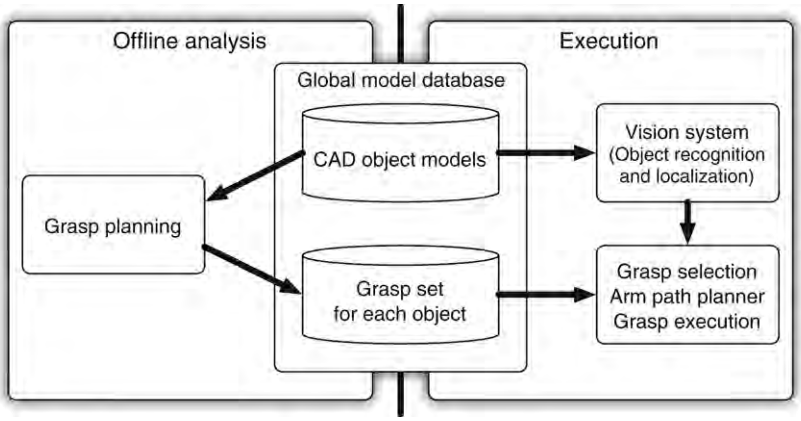


Fig. 9. Functional description of the integrated grasp planning system.

and recognition of objects (Section 5), a path planner for the generation of collision-free trajectories (Section 4) and an offline grasp analyser that provides the most feasible grasp configurations for each object. The results provided by these modules are stored and used by the control system of the robot for the execution of a grasp of a particular object. The functional description of the grasp planning system is depicted in Fig. 9. We emphasize that our approach describes a first step toward a complete humanoid grasping and dexterous manipulation system. The integrated grasp planning system, which has been presented in [21], will be explained briefly in the following. The system consists of the following parts:

- The *global model database*. 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 analyser* that uses a model of the object to be grasped together with a model of the hand to compute a set of stable grasps in a simulation environment. The results of this analysis are stored in the grasp database and can 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 Section 5.
- Once an object has been localized in the workspace of the robot, a grasp type for this 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. The path planner generates collision-free trajectories to reach the specified grasp position and orientation.

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 suffer from several disadvantages when considering the grasp execution in human-centred environments. These problems arise from the inaccuracy and uncertainty about the information of the object. Since we are using models of the objects, this uncertainty comes mainly from the location of the object. Usually, the contact-based grasp description requires the system to be able to reach precisely the contact points and exert precise forces. 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. 10):

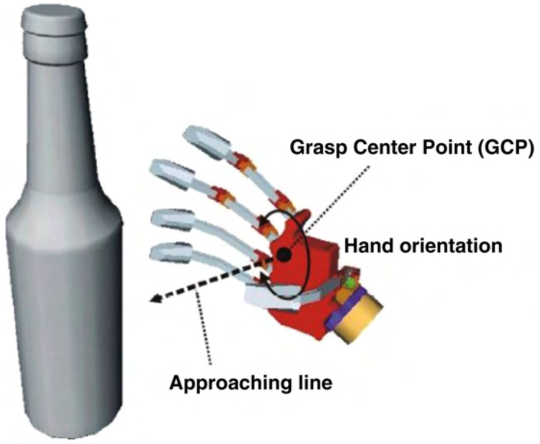


Fig. 10. Schematics with the grasp descriptors.

- *Grasp type:* A qualitative description of the grasp to be performed. The grasp type 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.

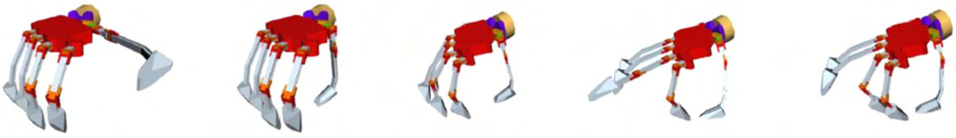


Fig. 11. Hand preshapes for the five representative grasp types.

It is important to note that all directions are given with respect to an object-centred coordinate system. The real approach directions result from matching this relative description with the localized object pose in the workspace of the robot. An important aspect when considering an anthropomorphic hand is how to relate the hand with respect to the grasp starting point (GSP) and the approaching direction. For this purpose we define the grasp centre point (GCP) of the hand. 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 (see Fig. 10). The GCP is aligned with the GSP of the grasp. Then the hand is orientated and preshaped according to the grasp descriptors and finally moves along the approaching line.

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. In addition, this representation is more robust to inaccuracies since it only describes starting conditions and not final conditions like a description based in contacts points.

We perform an extensive offline grasp analysis for each object by testing a wide variety of hand preshapes and approach directions. The analysis is carried out in 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 the online execution on the robot. As grasping simulation environment we use GraspIt! [20], which has 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. Due to the mechanical limitations of the robot hand, we have made a selection of the most representative grasps that can be executed by the robot hand. Fig. 11 shows the grasp patterns we have considered in our analysis. These are three power grasps (hook, cylindrical and spherical) and two precision grasps (pinch and tripod). A detailed description of the grasp analysis in given in [21].

7. Conclusion

We have presented a new humanoid robot consisting of an active head for foveated vision, two arms with five-fingered hands, a torso and a holonomic platform. The robot represents a highly integrated system suitable not only for research on manipulation, sensory-motor coordination and human-robot interaction, but also for real applications in human-centred environments. We presented an integrated system for the programming and execution of grasping and manipulation tasks in humanoid robots. The system incorporates a vision system for the recognition and localization of objects, a path planner for the generation of collision-free trajectories and an offline grasp analyser that provides the most feasible grasp configurations for each object.

In the two German exhibitions CeBIT 2006 and Automatica 2006, we could present the currently available skills of ARMAR-III. In addition to the robot's abilities to perceive its environment visually, we also showed how we can communicate with the robot via natural speech. Among the motor-skills we presented were the active tracking of objects with the head, combining neck and eye movements according to [31], basic arm reaching movements, early hand grasping tasks and force-based control of the platform movements. All skills were presented in an integrated demonstration.

Acknowledgements

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) and funded by the European Commission.

References

[1] K. Akachi, K. Kaneko, N. Kanehira, S. Ota, G. Miyamori, M. Hirata, S. Kajita, F. Kanehiro, Development of humanoid robot HRP-3, in: IEEE/RAS International Conference on Humanoid Robots, 2005.

[2] A. Albers, S. Brudniok, W. Burger, Design and development process of a humanoid robot upper body through experimentation, in: IEEE/RAS International Conference on Humanoid Robots, 2004, pp. 77–92.

[3] T. Asfour, D. Ly, K. Regenstein, R. Dillmann, Coordinated task execution for humanoid robots, in: Experimental Robotics IX, STAR, Springer Tracts in Advanced Robotics, Springer Verlag, 2005.

[4] T. Asfour, K. Regenstein, P. Azad, J. Schröder, N. Vahrenkamp, R. Dillmann, ARMAR-III: An integrated humanoid platform for sensory-motor control, in: IEEE/RAS International Conference on Humanoid Robots, 2006.

[5] P. Azad, T. Asfour, R. Dillmann, Combining appearance-based and model-based methods for real-time object recognition and 6D-localization, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, Beijing, China, 2006.

[6] P. Azad, T. Asfour, R. Dillmann, Stereo-based 6D object localization for grasping with humanoid robot systems, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, San Diego, USA, 2007.

[7] R. Becher, P. Steinhaus, R. Zöllner, R. Dillmann, Design and implementation of an interactive object modelling system, in: Robotik/ISR, München, Germany, May 2006.

[8] O. Brock, L. Kavraki, Decomposition-based motion planning: A framework for real-time motion planning in high-dimensional configuration spaces, in: icra, 2001.

[9] G. Cheng, S. Hyon, J. Morimoto, A. Ude, S. Jacobsen, CB: A humanoid research platform for exploring neuroscience, in: IEEE/RAS International Conference on Humanoid Robots, 2006.

[10] D. DeMenthon, L. Davis, D. Oberkampf, Iterative pose estimation using coplanar points, in: International Conference on Computer Vision and Pattern Recognition, CVPR, 1993, pp. 626–627.

Please cite this article in press as: T. Asfour, et al., Toward humanoid manipulation in human-centred environments, Robotics and Autonomous Systems (2007), doi:10.1016/j.robot.2007.09.013

[11] R. Dillmann, Teaching and learning of robot tasks via observation of human performance, Robotics and Autonomous Systems 47 (2–3) (2004) 109–116.

[12] M.A. Fischler, R.C. Bolles, Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography, Communications of the ACM 24 (1981) 381–395.

[13] B.K.P. Horn, Closed-form solution of absolute orientation using unit quaternions, Journal of the Optical Society of America 4 (4) (1987) 629–642.

[14] J.L.I.W. Park, J.Y. Kim, J. Oh, Mechanical design of humanoid robot platform KHR-3 (KAIST humanoid robot-3: HUBO, in: IEEE/RAS International Conference on Humanoid Robots, 2005.

[15] O. Kerpa, K. Weiss, H. Wrn, Development of a flexible tactile sensor for a humanid robot, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, Las Vegas, Nevada, 2003, pp. 1–6.

[16] J. Kuffner, S. LaValle, RRT-connect: An efficient approach to single-query path planning, in: IEEE International Conference on Robotics and Automation, 2000.

[17] S. LaValle, Rapidly-exploring random trees: A new tool for path planning. Technical report, Computer Science Dept., Iowa State University, October 1998.

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

[19] P. Michel, C. Scheurer, J. Kuffner, N. Vahrenkamp, R. Dillmann, Planning for robust execution of humanoid motions using future perceptive capability, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, 2007 (in press).

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

[21] A. Morales, T. Asfour, P. Azad, S. Knoop, R. Dillmann, Integrated grasp planning and visual object localization for a humanoid robot with five-fingered hands, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, Beijing, China, 2006.

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

[23] S. Nayar, S. Nene, H. Murase, Real-time 100 object recognition system, in: IEEE International Conference on Robotics and Automation, vol. 3, Minneapolis, USA, 1996, pp. 2321–2325.

[24] K. Nishiwaki, T. Sugihara, S. Kagami, F. Kanehiro, M. Inaba, H. Inoue, Design and development of research platform for perception-action integration in humanoid robots: H6, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, 2000, pp. 1559–1564.

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

[26] S. Quinlan, Real-time modification of collision-free paths. Ph.D. Thesis, Stanford University, 1994.

[27] S. Sakagami, T. Watanabe, C. Aoyama, S. Matsunage, N. Higaki, K. Fujimura, The intelligent ASIMO: System overview and integration, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, 2002, pp. 2478–2483.

[28] G. Sanchez, J. Latombe, A single-query bi-directional probabilistic roadmap planner with lazy collision checking, in: International Symposium on Robotics Research, Lorne, Victoria, Australia, 2001.

[29] S. Schulz, C. Pylatiuk, A. Kargov, R. Oberle, G. Bretthauer, Progress in the development of anthropomorphic fluidic hands for a humanoid robot, in: IEEE/RAS International Conference on Humanoid Robots, Los Angeles, 2004.

[30] J. Shi, C. Tomasi, Good features to track, in: International Conference on Computer Vision and Pattern Recognition, CVPR, Seattle, USA, 1994, pp. 593–600.

[31] A. Ude, C. Gaskett, G. Cheng, Support vector machines and gabor kernels for object recognition on a humanoid with active foveated vision, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, 2004, pp. 668–673.

[32] N. Vahrenkamp, T. Asfour, R. Dillmann, Efficient motion planning for humanoid robots using lazy collision checking and enlarged robot models. in: IEEE/RSJ International Conference on Intelligent Robots and Systems, October 2007 (in press).

[33] K. Welke, P. Azad, R. Dillmann, Fast and robust feature-based recognition of multiple objects, in: IEEE/RAS International Conference on Humanoid Robots, Genoa, Italy, 2006.

Tamim Asfour is post-doctoral researcher at the Institute for Computer Science and Engineering, University of Karlsruhe (TH), Germany. His research interests include humanoid robotics, motion planning, humanoid manipulation, generating of human-like humanoid motion, mechatronics and system integration in humanoid robots.

Pedram Azad is Ph.D. student at the Institute for Computer Science and Engineering, University of Karlsruhe (TH), Germany. His research interests include humanoid robotics, computer vision and human motion capture.

Nikolaus Vahrenkamp is Ph.D. candidate at the Institute for Computer Science and Engineering, University of Karlsruhe (TH), Germany. His research interests include humanoid robotics, motion planning, realtime collision avoidance, generating of human-like humanoid motion.

Kristian Regenstein is a researcher at the group Interactive Diagnosis — and Servicesystems (IDS), Research Center for Information Technologies (FZI), Germany. He focuses on humanoid robotics, computer architecture and the developement of system components for humanoid robots.

Alexander Bierbaum is Ph.D. student at the Institute for Computer Science and Engineering, University of Karlsruhe (TH), Germany. His research focuses on tactile sensing, dexterous manipulation and haptic exploration for humanoid robot hands and embedded control systems.

Kai Welke is Ph.D. student at the Institute for Computer Science and Engineering, University of Karlsruhe (TH), Germany. His interests include computer vision, active vision for learning of multimodel object representations.

Please cite this article in press as: T. Asfour, et al., Toward humanoid manipulation in human-centred environments, Robotics and Autonomous Systems (2007), doi:10.1016/j.robot.2007.09.013

Joachim Schroeder is a research assistant at the Institute for Computer Science and Engineering, University of Karlsruhe (TH), Germany. His research focus is on mobile robotics, path-planning and parking algorithms, decision making for autonomous road vehicles, driver-assistance-systems.

Rüdiger Dillmann is Professor at the Computer Science Faculty, University of Karlsruhe (TH), Germany. His research interestes include humanoid robotics, machine learning, Programming by Demonstration and human-centered technologies.

Please cite this article in press as: T. Asfour, et al., Toward humanoid manipulation in human-centred environments, Robotics and Autonomous Systems (2007), doi:10.1016/j.robot.2007.09.013