Project no.:	**IST-FP6- IP-027657**
Project full title:	**Perception, Action & Cognition through Learning of Object-Action Complexes**
Project Acronym:	**PACO-PLUS**

Deliverable no.: D1.2

Title of the deliverable: Integration Report I

Contractual Date of Delivery to the CEC:	31. July 2006
Actual Date of Delivery to the CEC:	**31. July 2006**
Organisation name of lead contractor for this deliverable	University of Karlsruhe (UniKarl)
Author(s): Tamim Asfour, Pedram Azad, Aleš Ude, Norbert Krüger and Rüdiger Dillmann	
Participant(s): UniKarl, JSI, AAU	
Work package contributing to the deliverable:	WP1, WP2, WP8
Nature:	R
Version:	1.0
Total number of pages:	15
Start date of project:	1st Feb. 2006 **Duration:** 48 month

Project co-funded by the European Commission within the Sixth Framework Programme (2002-2006)		
Dissemination Level		
PU	Public	
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:

The technical report presents the design and implementation of PACO-PLUS robot platforms (humanoid head for active vision and audio perception and a haptic sensor concept for a five-finger anthropomorphic hand) as well as the specification of the hardware and software interfaces that allows for efficient exchange of hardware and software modules in order to minimize the overhead in the overall system development and for the integration of several components from different partners in the PACO-PLUS robot platforms. Furthermore, a summary of the communication between the groups in the first six month is given.

Keyword list: Active foveated vision system, tactile hand sensors, software and hardware interfaces, meetings

Table of Contents

0. INTRODUCTION 3

1. DESIGN OF A HUMANOID HEAD FOR FOVEATED ACTIVE VISION 3

1.1 MOTOR SYSTEM 4
1.2 SENSOR SYSTEM 4
1.3 COMPUTATIONAL SYSTEM 5

2. TACTILE SENSOR SYSTEM FOR A FIVE-FINGERED ROBOT HAND 6

3. SOFTWARE COMPONENTS 7

3.1 INTEGRATING VISION TOOLKIT (IVT) 7
3.2 DISTRIBUTED COMPUTER ARCHITECTURE (CLUSTER) 8
3.3 MODULAR CONTROLLER ARCHITECTURE (MCA) 8
3.4 ROBOT INTERFACE 9
3.5 COGNITIVE VISION SOFTWARE (COVIS) 10

4. OTHER HARDWARE ENVIRONMENTS 11

5. COMMUNICATION BETWEEN THE GROUPS 11

5.1 SUBGROUP “EXPLORATION IN UNCONSTRAINED ENVIRONMENTS” 11
5.2 SUBGROUP “AUGMENTING HUMAN ACTION” 12
5.3 SUBGROUP “LEARNING AND PLANNING” 13

REFERENCES 15

0. Introduction

This document presents the hardware and software components that are being developed for the PACO-PLUS project and gives an overview about the communication that took place between the partners.

In Section 1, the motor, sensor and computational system of a new humanoid head for foveated vision is described. Section 2 presents an initial specification of a tactile sensor system for a five-fingered robot hand. Section 3 summarises the developed software components and their interfaces. In Section 4 a hardware environment of reduced complexity situated at AAU is described that is used particularly in the context of WP 8.1. Section 5 gives an overview of the established PACO-PLUS subgroups and the communication between the partners.

1. Design of a Humanoid Head for Foveated Active Vision

The head can provide rich perceptual input necessary to realize various visuo-motor behaviours, e.g. smooth pursuit and saccadic movements towards salient regions, and also more complex sensory-motor tasks such as hand-eye coordination, gesture identification, human motion perception and linking of visual representations to the motor representations. The major design criteria were as follows:

- The robot head should be of realistic human size and shape while modelling the major degrees of freedom (DoFs) found in the human neck/eye system, incorporating the redundancy between the neck and eye DoFs.
- The robot head should feature humanlike characteristics in motion and response, that is, the velocity of eye movements and the range of motion will be similar to the velocity and range of human eyes.
- The robot head must enable saccadic motions, which are very fast eye movements allowing the robot to rapidly change the gaze direction, and smooth pursuit over a wide range of velocities.
- The optics should mimic the structure of the human eye, which has a higher resolution in the fovea.
- The vision system should mimic the human visual system while remaining easy to construct, easy to maintain and easy to control.

With this set of requirements, a first version of the head have been developed as part of a humanoid robot that will allow for the integration of motor control and perception. This is essential to enable explorative head, hand, and body movements for learning of OACs.

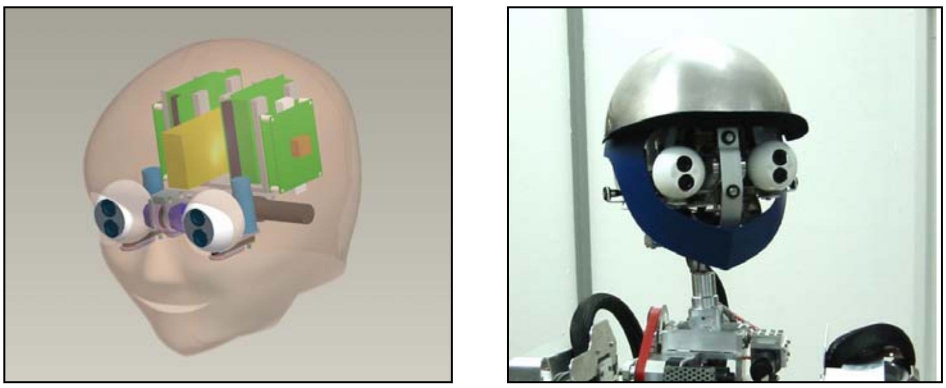

Fig. 1: CAD model of the new head (left) and the first realisation of the head (right). Two cameras per eye and six microphones. The eyes have a common tilt and can pan independently. The visual system is mounted on a 4 DOF neck mechanism.

1.1 Motor system

The head has seven DoFs and is equipped with two eyes. Each eye can independently rotate about a vertical axis (pan DoF), and the two eyes share a horizontal axis (tilt DoF). To approximate These two DoFs allow for human-like eye movements[1]. The visual system is mounted on a neck mechanism [1] with four DoFs organised as pitch-roll-yaw-pitch. The motors are driven by a 3-axis motor controller (see Section 1.3)

The joints are controlled by DC motors. For the pan joints we chose the brushless Faulhaber DC motor 1524-024 SR with backlash-free gear, IE2-512 encoder, 18/5 gear with 76:1 gear ratio, torque 2,5 mNm, and weight 70g. For the tilt joint we chose the Harmonic Drive motor PMA-5A-50 with backlash-free gear, 50:1 gear ratio, and torque 0,47 Nm. The calculation of the actuators characteristics was based on the desired specifications and the moment of inertia, as well as the different weight of components, which were given by the CAD software.

1.2 Sensor System

To start learning object-action complexes we must, firstly, identify regions that potentially contain objects of interest and secondly analyze these regions to build higher-level representations. While the first task is closely related to visual search and can benefit from a wide field of view, a narrower field of view resulting in higher-resolution images of objects is better suited for the second task. While the current technology does not allow us to exactly mimic the features of the human visual system and because camera systems that provide both peripheral and foveal vision from a single camera are still experimental, we decided for an alternative which allows to use commercially available camera systems that are less expensive and more reliable.

Foveated vision was realized using two cameras per eye, one with wide-angle lens for peripheral vision and one with narrow-angle lens for foveal vision. We use the Point Grey Research Dragonfly IEEE-1394 camera in the extended version (www.ptgrey.com). The extended version allows the CCD to be up to 6 inches away from the camera interface board. This arrangement helps with accessing hard to reach places, and with placing the lens into a small volume. Since the cameras are very light and are extended from the interface board by a flexible extension cable, they can be moved with small and low-torque servos.

The cameras can capture colour images at a frame rate of up to 30 Hz. They implement the DCAM standard, and transmit a raw 8 bit Bayer Pattern with a resolution of 640x480, which is then converted on the PC to a 24 bit RGB image. The cameras have a FireWire interface, which is capable of delivering data rates of up to 400 Mbps. The benefit of transmitting the Bayer Pattern is that only a third of the bandwidth is needed for transmitting the colour image without loosing any information. Thus, it is possible to run one camera pair at a frame rate of 30 Hz and the other at a frame rate of 15 Hz, all being synchronized over the same FireWire bus, without any additional hardware or software effort. Running the foveal cameras, which have a smaller focal length and thus a narrower view angle, at a lower frame rate is not a drawback because these cameras are not crucial for time critical applications such as tracking, but are utilized for detailed scene analysis, which does not need to be performed at full frame rate in most cases anyway.

The camera is delivered as a development kit with three micro lenses with the focal lengths 4, 6, and 8mm. In addition, one can use micro lenses with other focal lengths as well. We have chosen a 3 mm micro lens for the peripheral cameras and a 10 mm micro lens for the narrow angle cameras.

Furthermore, the head is equipped with six microphones (SONY ECMC115.CE7): two in the ears, two in the front and two in the rear of the head. These microphones will be used in the later phase of the project to achieve a richer multi-sensory representation of objects and environment and to support the integration of speech components in order to provide an additional information for interaction and natural communication.

---

[1] Human eyes have can rotate slightly about the direction of gaze. We decided to omit this DoF in the eyes because the pan and tilt DoFs are sufficient to cover the visual space

1.3 Computational System

The head (visual and motor system) are controlled by

- 3 Universal Controller Module (UCoM) units for low-level motor control and sensory data acquisition: 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 for calculations and data processing whereas the FPGA offers the flexibility and the hardware acceleration for special functionalities.
- One off-the-shelf PC/104 with Pentium 4, 2 GHz processor and 2 GB of RAM running under Debian Linux, kernel 2.6.8 with the Real Time Application Interface RTAI/LXRT-Linux. The PC is equipped with a dual FireWire card and a CAN bus card. The communication between the UCoMs and the PC 104 system take place via CAN bus

The basic control software is implemented in the *Modular Controller Architecture* framework MCA2 (www.mca2.org). MCA2 is described in Section 3.

Technical Data	
Kinematics	• 4 DoFs in the neck (lower pitch, roll, yaw, and upper pitch) • 3 DoFs for the eyes: common tilt and independent pan
Vision System	• Each eye is realized through 2 Point Grey Research Dragonfly color camera in the extended version with a resolution of 640x480 at 30 Hz.
Auditory System	• 6 microphones (SONY ECMC115.CE7): two in the ears, tow in the front and two in the rear of the head.
Universal Controller Module (UCoM)	• 3 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.
1 Embedded PC/104 system	• 2 GHz PC/104 system with dual FireWire card and CAN card. • Communication between the UCoMs and the PC 104 system via CAN bus.
Operation System	• The embedded PC104 system is running under Linux, kernel 2.6.8 with Real Time Application Interface RTAI/LXRT-Linux (debian distribution)
Head 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 groups. • 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

Tab. 1: .Motor, sensor and computational system of the head

2. Tactile Sensor System for a Five-Fingered Robot Hand

The haptic sensor system of the five-fingered robot hand is based on two types of sensors: positional sensors for determining and controlling the finger configuration and force sensors for measuring and controlling contact forces between the hand and a grasped object.

Every active joint of the robot hand is equipped with an angle encoder as position sensor. It is possible to read the joint position continuously via the RS232-Interface of the hand controller. The hand controller supports native position control for every joint. The locations of the encoders and therefore the eight active joints are indicated with circles in Fig. 2. The angle encoder is a miniature magnetic absolute encoder IC system with a resolution of 0.09° and uses standard industry components.

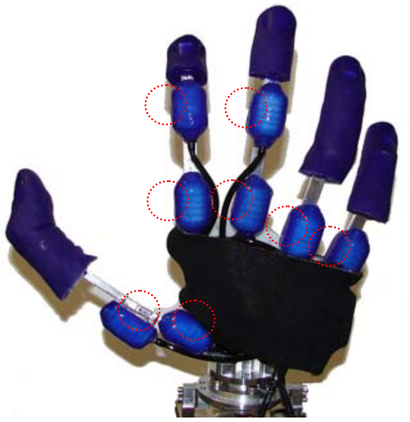

Fig. 2: The five-fingered hand [2]

For the tactile sensor system several sensor types have been investigated for their suitability. A strong requirement for the tactile sensors is to determine the contact normal force vector (CNFV) that allows for dextrous manipulation and reactive grasping with several common control algorithms. Sensors for force measurement may be divided in scalar and matrix type sensors. Using matrix force sensors the CNFV may be approximated from the measurement data when assuming a contact area larger than the sensor's grid resolution. Manufacturers of commercial matrix type force sensors are [3,4,5].

These sensors are usually manufactured as flexible sheets with uniformly distributed adjacent sensor cells. The cells usually have the shape of squares with the length of an edge ranging down to a few millimetres, defining the resolution.

There is no off-the-shelf solution available for matrix force sensors in the application area of robot hands. The sensors always need to be customized in terms of geometry and resolution, which results in considerable costs for this type of sensors. All manufacturers examined require the customer to use a special sensor signal processing HW unit for processing sensory data.

The sensor technology used for force sensors relies either on a variant of the FSR (Force Sensing Resistor) principle [3,5,6] or on a capacitive effect [4]. Although the term FSR itself is copyrighted by [6] the other manufacturers exploit material properties in a similar way, so we summarized them in this group.

With FSRs the resistance of the sensing element decreases with increasing pressure applied to it. FSRs in general require a break force to switch it from zero conductivity to a finite resistance value defining the beginning of the dynamic range. From here the characteristic usually follows an inverse power law.

With capacitive sensors the capacitance varies with pressure. For determining the amount of pressure these types of sensors are usually applied in oscillator circuits of which the output frequency is a direct measure of force.

An alternative, low-priced solution for measuring the CNFV is the disposal of an FSR Micro Joystick as offered by [6]. These Micro Joysticks were originally intended as input devices for notebook computers and are therefore available in large quantities and at a moderate price. The sensor area is made up of four FSR sectors upon which the actuator stick is pressed. The CNFV corresponds to the angle under which the actuator is pressed to the sensor area and the magnitude of the applied pressure. It can be calculated from the activation relation of the four measured force signals. In contrast to matrix sensors only one CNFV can be calculated with one sensor, therefore the sensors should be mounted on the primary contact points of the robot hand, i.e. the finger tips. Figure 3 shows a Micro Joystick attached to a robot finger with the standard cap as delivered by the manufacturer. This cap can be replaced easily by a silicon cap better satisfying the anthropomorphic design of the fingers.

The standard Micro Joystick FSRs by [6] have a break force value of about 0.7 N at an activation resistance of 100 KΩ and have a dynamic range of up to 7 N. The four FSRs may be easily interfaced to A/D-Converters for measurement.

In the first step the fingers of the robot hand will be equipped with Micro Joysticks. This appears to be sufficient to detect contact events and enable dynamic grasping. We believe that even edge tracking for shape identification could be performed with this type of sensors although we note that these sensors do not give a sensor image as comprehensive as when using matrix sensors. In addition, single FSRs will be attached to the palm delivering simple contact force information. This is adequate for the palm as no CNFV measurement is required here.

In a concurrent approach matrix sensors from [5] will be customized for the robot hand. Beside the CNFV measurements the utilization of matrix sensors will also allow the introduction of tactile image processing methods providing advanced haptic exploration capabilities. Until these sensors are customized and available for the robot hand the tasks of reactive, force-controlled grasping can be performed using Micro Joysticks at the fingertips and distributed FSRs in the palm.

Fig. 3: Microjoystick attached to the finger tip

3. Software Components

Integration of numerous software components on a humanoid robot system is a key task for a successful project. The successful definition of the necessary interfaces at the beginning of a project is crucial for allowing the different partners to start developing modules, while at the same time making sure that these can be integrated at a later point with minimum effort. In the following, we present five software frameworks and libraries, which build the base for the implementation and interconnection of software components for perception and action:

1. Integrating Vision Toolkit (IVT)
2. Distributed Computer Architecture (Cluster)
3. Modular Controller Architecture (MCA)
4. Robot Interface (our framework for integration on the robot)
5. Cognitive Vision Software (CoViS)

3.1 Integrating Vision Toolkit (IVT)

The Integrating Vision Toolkit (http://ivt.sourceforge.net) is a computer vision library we have developed, allowing us to start the development of vision components within minimum time. We have experienced that even though in theory many problems in computer vision have been solved for many years, in practice there is no library implementing the corresponding algorithms in a complete toolkit. Thus, usually each partner starts to develop their own library, offering the functionality they need. In addition to the enormous additional implementation effort, this approach would lead to crucial problems when wanting to put the modules from the partners together into an integrated system. Among the problems solved and functions implemented by our library are:

- Integration of various cameras and other image sources via a clean interface.
- Providing a 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 Graphical User Interfaces – for which we chose QT.
- Full support for the operating systems Windows, Linux, and Mac OS.

Using IVT for the implementation of vision components allows us to develop vision components that are **platform-** and **camera-independent**. This is crucial for a successful integration on the demonstration platform.

3.2 Distributed Computer Architecture (Cluster)

An artificial cognitive system must be capable of running processes of various complexity with different requirements regarding the delays and latencies. While on-board processing is best suited for tasks that require tight integration between perception and motor control and small delays, such an architecture becomes too limiting when emulating higher-level, computationally expensive cognitive processes. We developed a distributed computer architecture to facilitate real-time implementation of such processes. Our current computer cluster consists of six servers with one or two dual core processors and two workstations, which ensure better connectivity with the robotic hardware. Each computer is equipped with a display to enable visualization of intermediate stages of processing, which is essential for the analysis of cognitive processes. The cluster is connected via a series of 1 Gbit Ethernet networks, which can be separated to improve the data throughput of the system. We utilized the UDP protocol to stream the data such as uncompressed images from one computer to another in real-time. Facilities that allow the synchronization of a number of processes that run on the cluster in parallel are also provided.

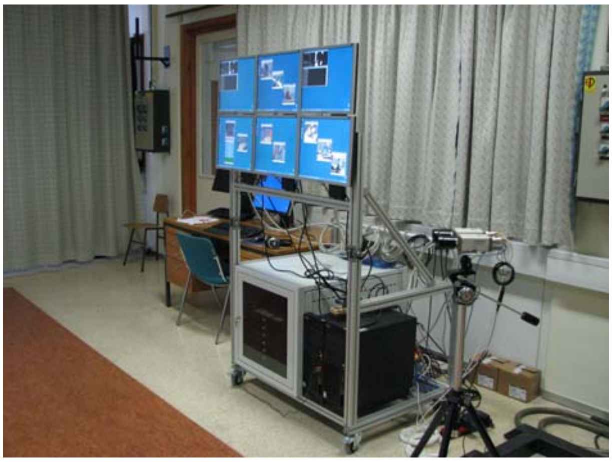

Fig. 4: Computer Cluster at JSI: six servers, two workstations, 1 Gbit Ethernet networks

3.3 Modular Controller Architecture (MCA)

The basic control software is implemented in the Modular Controller Architecture framework MCA2 (http://www.mca2.org). MCA2 is a modular, network transparent and real-time capable C/C++ framework for controlling robots and other kinds of hardware. In MCA2, all methods are implemented by C++ modules with standardized interfaces and thus allow an easy integration of software components.

The control parts can be extended under Linux, RTAI/LXRT-Linux, Windows, or Mac OS, and communicate beyond operating system borders. A graphical debugging tool (MCAbrowser), which can be connected via TCP/IP to the MCA processes, can visualize the connection structure of the control modules and their current values at run-time. The tool MCAGUI can create Graphical User Interfaces at runtime with various input and output entities. Both interfaces provide full access to the control parameters at runtime.

3.4 Robot Interface

Although MCA offers a very clean and transparent mechanism for developing and integrating relatively low-level software components related to control, kinematics, and dynamics, it is not suitable for the convenient implementation of higher-level modules. Thus, we have built a so-called “Robot Interface” on top of MCA, building an abstraction level for the access to the robot’s sensors and actuators. By using the robot interface, the user is not aware in which way the information is communicated to and from the relevant component of the robot. Scenarios can be implemented by different partners by simply inheriting and implementing the scenario interface. In this way the incoming messages, which result from sensory data processing, e.g. recognized speech commands or new positions from a sound localization module, are passed by calling the corresponding callback method of the interface. On the action side, from within a scenario, commands can be passed to the robot by simple method calls, such as:

- MoveHead: This method has different modes including forward kinematics, inverse kinematics with absolute positions, and two different tracking modes.
- MoveArm: This method has different modes including forward kinematics and different types of inverse kinematics.
- MovePlatform: This method has different modes including velocity-based control, point to point navigation, and navigation on graphs.

On the perception side, a scenario has access to the following:

- All current joint angle values for the head and the arms.
- Current position of the robot platform.
- Other sensor values, such as sensors integrated in the hands.
- Higher-level perceptive modules (this is explained in the following).

The higher-level perceptive modules are vision components accomplishing a specific task for the visual analysis of the environment. In general, two classes can be distinguished:

- Modules recognizing and localizing known entities.
- Modules exploring unknown scenes.

From the integrative point of view, it is important that all modules share the same interface in order to guarantee that vision components from the partners can be built in and replaced with minimum effort. Especially the step from analyzing scenes containing known entities, which have been learnt with modern and powerful object recognition and localization algorithms, to the autonomous exploration of unknown scenes can be prepared very elegantly. Our perception modules produce output as 3D rotations and translation in terms of 3D object models or 3D primitives (given in the Open Inventor format), such as planes, spheres, or cylinders. These build at the same time the interface to the robot’s grasp planning and execution system. By sharing the same data structures and output formats, we will be able to replace our currently used object recognition and localization systems by an explorative vision system being developed on the base of visual primitives, which play a central role in our project. By following this strategy, we can already now start to develop and test the high-level architecture for OACs, without being dependent on the completion of the modules for autonomous scene exploration.

3.5 Cognitive Vision Software (CoViS)

The Cognitive Vision Software is a C++ Vision library that has been developed since 1998 and which has been further worked on at AAU and BCCN. It provides an Early Cognitive Vision system (see, e.g., [7,8]) in which different visual modalities such as edges, corners, color, contrast transition, stereo and local motion become combined into a condensed and integrated representation. It has been largely influenced by the ideas developed in the EU project ECOVISION [9] in which modality integration has been investigated in the context of biological systems and this technical system has been realized. A central component of CoViS are so called multi-modal visual primitives [8]. These are local symbolic entities which work as first hypotheses in a disambiguation process that utilizes contextual information in terms of statistical and deterministic regularities to generate complete and reliable visual scene representations.

In the PACO+ project, the CoViS software will be linked and combined with the software components IVT and MCA. The robot interface is planned to be used at AAU/SDU to link the industrial robot RX60 (see section 4) to the vision components.

Summary	
Integrating Vision Toolkit	• Object-oriented vision library • Clean camera interface with numerous modules for all kinds of common cameras and image sources • Generic method for camera calibration • Allows to develop platform- and camera-independent vision applications • Many mathematical routines and image processing functions, such as segmentation routines, filters, 3D reconstruction, etc. • Is used as the base of medium- to high-level perceptive components
Modular Controller Architecture	• Framework for the implementation of controller architectures, beyond process and platform borders • Unified interface specialized for controller modules • Convenient debugging tool (MCAbrowser) • Convenient creation of Graphical User Interfaces at runtime (MCAGUI) • Is used for low-level control and forward / inverse kinematics
Robot Interface	• Framework for full access to the robot in a convenient manner • Gathering all sensory information and making it available • Providing functions for the control of the robot’s actors • Giving access to the visual perceptive components • Is used for the implementation of the high-level architecture
Cognitive Vision Software	• Processing of varies visual modalities such as optic flow, stereo, edges and corners, color, etc. • Early Cognitive Vision Scene Representation • Visually guided robot actions

4. Other hardware environments

Besides the anthropomorphic hardware environment described in section 1 to section 3, in the context of demo 1 we make use of an environment of reduced complexity at AAU/SDU consisting (see figure 4) of the following three sub-components:

1. The industrial robot RX60 from Staubli [10].
2. The two finger grasper PT-AP79 from Schunk [11].
3. The stereo head Bumblebee from Point Grey Research [12].

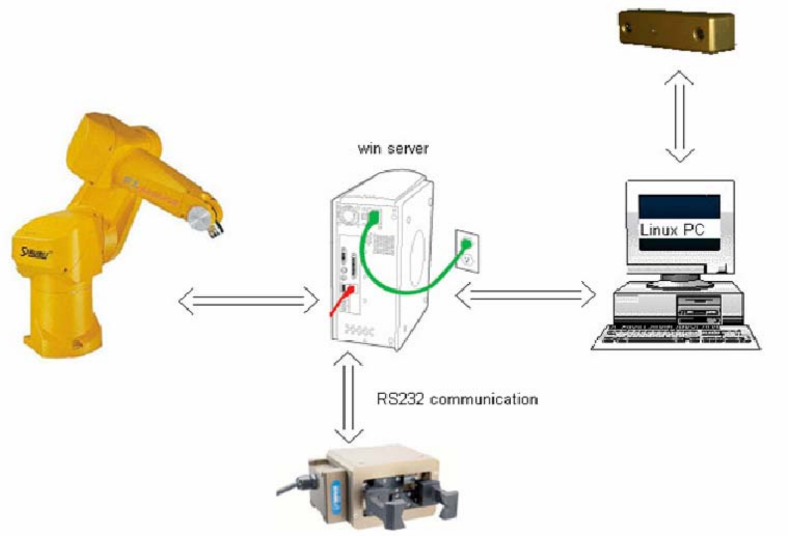

Fig. 4: The AAU/SDU hardware environment

The use of a different environment is motivated by two reasons. First, the price of the Karlsruhe robot environment was beyond the reserved AAU/SDU budget. An establishment of a complex hardware platform such as existing in Karlsruhe would also not be reasonable since the group at AAU has been focussed mainly on vision. Furthermore, due to its complexity, the Karlsruhe robot environment requires a large effort in terms of maintenance that could not be provided by AAU. Second and more importantly, the simplicity of the AAU environment allows us to address some issues more directly than in the sophisticated Karlsruhe environment. For example, problems in terms of precision of movements can be largely neglected with an industrial robot as the RX60. In [13], it is indicated that in the AAU hardware environment, we can perform grasps using 3D information computed by the vision software without making use of any servoing mechanisms. Also problems of inverse kinematics are reduced due to the fact that the RX60 robot has only six degrees of freedom. Furthermore, in the static stereo system calibration is much easier to be addressed than on the active head described in section 1.

Sub-components developed at AAU within the environment of reduced complexity will become integrated into the main Karlsruhe environment during the project.

5. Communication between the groups

Three sub-groups held a number of general meetings and phone conferences, which are listed in this report. In addition, there were numerous one-to-one discussions between the subgroup members as well as bilateral visits of project members on the sites of the partners.

5.1 Subgroup “Exploration in unconstrained Environments”

The “**Manipulation in unconstrained Environments**” sub-group within the PACO+ project was established to develop a framework for explorative behaviour (Demo 1) in which world-knowledge in terms of things and OACs become extracted and applied. This is supposed to happen to a large degree autonomously and purposefully. Triggered by initial reflex-like action mechanisms that gradually enlarge knowledge about objects and the action applicable on them. By that the basis for more complex planning algorithms is built. In contrast to Demo 2 (Augmented human action) at the first stage, we do not regard humans interfering. However, in the long run coaching and imitation mechanisms may take the role of reflexes to guide explorative behaviour to relevant parts of the action state-space and by this demo 1 and demo 2 will grow together.

An important sub-problem is the integration of a planning level making use of symbolic representations that need to be linked to lower levels of OAC representations. The guiding principle is that a (in the beginning naive) world model becomes gradually extended when expectation about the world become disappointed (surprise) and the system realises the necessity of extending the current world model to meet the requirements of successfully predicting the world it is acting in.

The main members of the subgroup are University of Aalborg / University of Southern Denmark (AAU, Norbert Krüger), Kungliga Tekniska Högskolan (KTH, Jan-Olof Eklundh and Danica Kragić), Universität Karlsruhe (UniKarl, Rüdiger Dillmann and Tamim Asfour), University of Leiden (UL, Bernhard Hommel), University of Liege (Justus Piater), University of Edinburgh (EDIN, Mark Steedman) and University of Göttingen (BCCN, Florentin Wörgötter).

The subgroup held a number of general meetings which are listed below. In addition, there were numerous one-to-one discussions between the subgroup members as well as bilateral visits of project members on the sites of the partners(e.g., a visit of Norbert Krueger in Stockholm, a one month stay of Daniel Aarno from KTH at AAU, a visit of Justus Piater in Leiden and Karlsruhe, and a visit of Norbert Krueger in Liege) .

The general meetings took place as follows:

- **October 6th,7th 2005**:,pre-meeting in London Stansted, attended by Jan-Olof Eklunsh (KTH), Volker Krueger (AAU), Tamim Asfour (UniKarl) , Pedram Azad (UniKarl), Aleš Ude (JSI), Norbert Krueger (AAU).

  Definition of two sub-groups responsible for demo 1 and demo 2. Demonstration of the five finger hand by UniKarl.

- **November 7th,8th 2005,** meeting in Goettingen attended by Tamim Asfour (UniKarl), Pedram Azad (UniKarl), Dirk Kraft (UniKarl), Florentin Woergoetter (BCCN), Nicolas Pugeault (EDI), Justus Piater (Liege), Renaud Detry (Liege), Norbert Krueger (AAU).

  Discussion and Specification of OAC concept. Definition of Subtasks. Specification of Demo 1 environment.

- **Februar 1st-3rd 2006,** kick-off meeting in Karlsruhe attended by all participants

  Discussion of system architecture and the integration of higher level planning components in demo 1.

- **March 28rd,29rd 2006,** meeting in Copenhagen attended by Alex Bierbaum (UniKarl), Pedram Azad (UniKarl), Bernhard Hommel (UL), Mark Steedman (EDIN), Florentin Woergoetter (BCCN), Gwendid van der Voort van der Kleij (UL), Nico Pugeault (EDIN), Dirk Kraft (UniKarl), Norbert Krueger (AAU), Daniel Aarno (KTH), Danica Kragic (KTH).

  System specification. Attention, Prediction and Surprise in a bootstrapping system. High level planning in exploration.

- **June 1st 2006,** meeting Edinburgh attended by Mark Steedman (EDIN), Florentin Woergoetter (BCCN), Norbert Krueger (AAU), Natalia Shilo (BCCN), Ron Petrick (EDIN), Christopher Geib (EDIN, via Skype).

  Further specification of high level reasoning and planning in demo 1.

5.2 Subgroup “Augmenting Human Action”

The “Augmenting Human Action” subgroup within the PACO+ project was established to provide suitable sensory and motor representations of actions, both for recognition and for synthesis, which should finally result in the implementation on the PACO+ hardware. At the initial stage, we are mainly interested in the acquisition of human actions using motion capture systems. We study both upper body and hand motions.

Later on, the acquired knowledge will provide a basis for robot learning by imitation scenario as well as outline the requirements for communicating more advanced actions / scenarios to the robot. The goal is to provide concepts and implementations that can be applied for the creation of advanced object-action complexes. The main members of the subgroup are University of Aalborg (AAU, Volker Krüger), Institute Jožef Stefan (JSI, Aleš Ude), Kungliga Tekniska Högskolan (KTH, Jan-Olof Eklundh and Danica Kragić), and Universität Karlsruhe (UniKarl, Rüdiger Dillmann, Tamim Asfour and Pedram Azad).

The subgroup held a number of general meetings and phone conferences, which took place as follows:

- **December 15th, 2005**, initial meeting at KTH, Stockholm, Sweden, attended by Tamim Asfour (UniKarl), Jan-Olof Eklundh (KTH), Danica Kragić (KTH), Volker Krüger (AAU), and Aleš Ude (JSI).

  Discussion about the overall goals of the "Augmenting Human Action" subgroup: topics to be considered and hardware to be used.

- **February 1st – 3rd, 2006**, general meeting at the University of Karlsruhe, Germany, attended by all project members.

  Establishment of various subgroups within the project, discussion about the role of the "Augmenting Human Action" subgroup in the PACO+ project.

- **March 31st, 2006**, Skype conference call, participants Tamim Asfour (UniKarl), Jan-Olof Eklundh (KTH), Danica Kragić (KTH), Volker Krüger (AAU), and Aleš Ude (JSI).

  In-depth discussion about every partner's role within the subgroup. Decision that each partner should prepare a short text describing his role within the subgroup and propose possible demonstrations on the PACO+ hardware after the first year of the project.

- **April 7th, 2006**, Skype conference call, participants Tamim Asfour (UniKarl), Pedram Azad (UniKarl), Jan-Olof Eklundh (KTH), Volker Krüger (AAU), and Aleš Ude (JSI).

  All four groups participating in the subgroup (AAU, JSI, KTH, and UniKarl) prepared texts about their possible contributions. Discussion based on the submitted texts.

- **April 25th, 2006**, Skype conference call, participants Tamim Asfour (UniKarl), Pedram Azad (UniKarl), Jan-Olof Eklundh (KTH), Danica Kragić (KTH), Volker Krüger (AAU), and Aleš Ude (JSI).

  A more detailed specification about what each partner can contribute for the demonstration after the first year of the project.

- **July 19th, 2006**, Skype conference call, participants Tamim Asfour (UniKarl), Danica Kragić (KTH), Volker Krüger (AAU), and Aleš Ude (JSI).

  Review of the achieved results, preparation for the general meeting in Stockholm at the end of the month.

- **July 27th, 2006,** Skype conference call, participants Tamim Asfour (UniKarl), Danica Kragić (KTH), Volker Krüger (AAU), and Aleš Ude (JSI).

  Preparation for the meeting in Stockholm at July 31

5.3 Subgroup "Learning and Planning"

The "Learning and Planning" subgroup within the PACO+ project was established to maintain a discussion about how to integrate the different learning techniques mastered by each of the partners. Namely, correlation based learning and reinforcement learning. Also, to initiate a cooperation for the implementation of categorization for learning, and to investigate how these tools may lead to 'discovery' of new OACs from simple ones. In addition, the subgroup also is working on an inverse kinematics module for the humanoid arm. These topics are related to WPs 6, 7, and 8.

The main members of the group are Carme Torras and Juan Andrade-Cetto (CSIC), Florentin Wörgötter (Göttingen), and Tamim Asfour (UniKarl).

The subgroup held a pair of general meetings, several phone conferences, and email exchanges. The general meetings took place as follows:

- **May 1st – 5th**, visit from Göttingen group to IRI-CSIC, Barcelona, attended by Florentin Wörgötter, Bettina Hoffman, and Christof Klodziejski (Göttingen), Juan Andrade-Cetto, Carme Torras, Alejandro Agostini, Pablo Jimenez, Enric Celaya, and Vicente Ruiz (CSIC).

  Discussion about Reinforcement Learning and Correlation Learning and their context within the project. Discussion on planning under uncertainty with deformable objects. Suggestion of a deformable object manipulation task for demonstration of decision making and planning under uncertainty.

- **June 19th – 21st**, meeting at Göttingen, attended by Carme Torras and Alejandro Agostini (CSIC), Christof Klodziejski, Florentin Wörgötter (Göttingen).

  Discussion on categorization and QL, and about the implementation of a categorization algorithm on a simple mobile robot application. Also, more discussion on the deformable object manipulation task.

In addition, the subgroup has initiated the following two student exchange collaborations:

- Mr. Stefan Ulbrich from UniKarl spent a semester (March – August 2006) at CSIC developing and implementing a Rational Bezier network-based inverse kinematics module for the humanoid arm, and tested it on simulations for the humanoid robot ARMAR-III, and with a working implementation for a Stäubli arm.
- Mr. Alejandro Agostini from CSIC is currently (June – November 2006) on a six month stay at Göttingen setting up categorization algorithms, and evaluating the combination of Correlation Learning and Reinforcement Learning using Categorization.

References

[1] A. Albers, S. Brudniok, and W. Burger. Design and development process of a humanoid robot upper body through experimentation, IEEE/RAS International Conference on Humanoid Robots, USA, 2004.

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

[3] Tekscan, http://www.tekscan.com

[4] Pressure Profile Systems Inc., http://www.pressureprofile.com

[5] Weiss Robotics, http://www.weiss-robotics.de/

[6] Interlink Electronics, http://www.interlinkelec.com/

[7] F. Wörgötter, N. Krüger, N. Pugeault, D. Calow, M. Lappe, K. Pauwels, M. Van Hulle, S. Tan, A. Johnston. Early Cognitive Vision: Using Gestalt-Laws for Task-Dependent, Active Image-Processing. Natural Computing 3(3):293-321. 2004.

[8] N. Krüger and M. Lappe and F. Wörgötter. Biologically Motivated Multi-modal Processing of Visual Primitives. Interdisciplinary Journal of Artificial Intelligence the Simulation of Behavious, AISB Journal, 1(5): 417-427, 2004.

[9] ECOVISION: Artificial Vision Systems based on early-cognitive cortical processing (EU project, STREP) 2001-2003.

[10] Stäubli Tec-Systems GmbH (Robotics), http://www.staubli.com

[11] Schunk GmbH & Co. KG, http://www.de.schunk.com

[12] Point Grey Research, http://www.ptgrey.com/

[13] Dirk Kraft. Bildbasiertes Greifen mit einer Fünf-Finger Hand. Master thesis at the University of Karlsruhe (2006).