PDA Interface for Humanoid Robots

Sylvain Calinon and Aude Billard

ASL3-I2S-STI
Swiss Institute of Technology in Lausanne - EPFL
CH-1015 Lausanne, Switzerland
{sylvain.calinon,aude.billard}@epfl.ch
http://asl.epfl.ch/

Abstract. To fulfill a need for natural, user-friendly means of interacting and reprogramming toy and humanoid robots, a growing trend of robotics research investigates the integration of methods for gesture recognition and natural speech processing. Unfortunately, efficient methods for speech and vision processing remain computationally expensive and, thus, cannot be easily exploited on cost- and size-limited platforms. Personal Digital Assistants (PDAs) are ideal low-cost platforms to provide simple speech and vision-based communication for a robot.
This paper investigates the use of Personal Digital Assistant (PDA) interfaces to provide multi-modal means of interacting with humanoid robots. We present PDA applications in which the robot can track and imitate the user's arm and head motions, and can learn a simple vocabulary to label objects and actions by associating the user's verbal utterance with the user's gestures. The PDA applications are tested on two humanoid platforms: a mini doll-shaped robot, *Robota*, used as an educational toy with children, and *DB*, a full body 30 degrees of freedom humanoid robot.

1 Introduction

End-user communication with robots is usually provided either by PC-based user interfaces, using common programming techniques, or by simple button-based remote controls. While these methods are very suitable for highly constrained environments, where reprogramming of the robot need not be continuous, these are undesirable for applications requiring the robot to work with laymen in their daily environment.

With the recent introduction on the market of affordable humanoids and toy robots (for example [12]), children as well as adults have started to spend a significant part of their leisure time engaging with these creatures. Toy robots have to fulfill a very difficult task, that to entertain, and, in some cases, that to educate [3, 7]. To this end, they are provided with behaviors that, in some ways, emulate the behaviors of the natural creatures they mimic [1, 5].

To fulfill a need for natural, user-friendly means of interacting and reprogramming toy and humanoid robots, a growing trend of robotics research investigates the integration of methods for gesture recognition and natural speech processing, as part of algorithms for robot programming by demonstration, e.g. [10, 9, 15].

Fig. 1. *Robota*, and the connection of a small camera to the *iPAQ PocketPC*, using an *Expansion Pack* to read *Compact-Flash* memory cards.

Providing robots with capabilities for speech and vision, such that they mimic human everyday communication, is an open research issue. Efficient methods for such processing remain computationally expensive and, thus, cannot be easily exploited on cost- and size-limited platforms. The rapid development of multimedia applications for Personal Digital Assistants (PDAs) make these handheld devices an ideal low-cost platforms to provide simple speech and vision-based communication for a robot. PDAs are light and can, therefore, easily fit a small robot, without overburden the robot's total weight. PDAs are easy to handle: they can be carried in one hand or in a pocket. Therefore, they can easily be used as a multi-modal remote-control for directing full-size robots.

There is a growing interest in developing PDA applications to remotely control mobile robots, e.g. [11, 17, 13, 16]. The present work follows closely such a trend and investigates the use of PDA interfaces to provide easy means of directing and teaching humanoid robots. Specifically, we develop PDA applications in which the robot can track and imitate the user's arms and head motions, and can learn a simple vocabulary to label objects and actions by associating the user's verbal utterance with the user's gestures. The applications are tested on two humanoid platforms: a mini doll-shaped robot, *Robota*, used as an educational toy with children, and *DB*, a full body 30 degrees of freedom humanoid robot.

This paper is divided as follows: Section 2 describes the architecture of the PDA language acquisition game, and its implementation in the *Robota* toy humanoid robot. Section 3 presents the implementation of the PDA language ac-

quisition game in the full size *DB* humanoid robot. Results for both experiments are given in Section 4. Finally, Section 5 discusses the results in the view of the current and further developments.

2 PDA Application for the *Robota* Toy Robot

2.1 PDA Tools

Fig. 2. Screenshot of the language acquisition application running on the *iPAQ*. The user has a visual feedback of his movements captured by the camera, with colored areas where the tracking take place.

The PDA used in our application is the *iPAQ 3850 PocketPC*. It is provided with a StrongARM 32-bit RISC Processor working at 206MHz, with 64Mb of RAM. It communicates with the robot via a serial interface. A *FlyCam-CF camera* is connected to the *iPAQ* via a *CompactFlash* Memory Card slot, through the *PocketPC Expansion Pack* (see Figure 1). The camera faces the user, taking snapshots of 160x120 pixels at a 15 images/sec frame rate. The *PocketPC* with the camera is mounted on the front of the robot.

CONVERSAY and *ELAN* software development kits (SDKs) provide speech recognition and speech synthesis of spoken English. Vision and speech processing are performed by the *PocketPC*.

The operating system (OS) and development tools used for our applications are *Microsoft PocketPC 2002*, and *embedded Visual C++* (freely available on *Microsoft* website). The SDKs used for speech recognition, speech synthesis and camera data acquisition are available only for the *PocketPC 2002* OS. Transition

of our application to free OS such as Linux will be considered when more open source codes will be available for speech processing and camera data acquisition for PDAs.

2.2 Robota

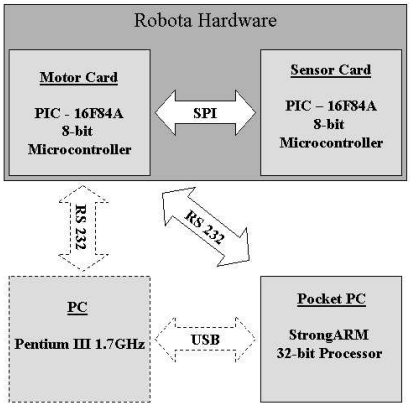

Fig. 3. Interconnectivity across the different microcontrollers and the *PocketPC*, for the mini-humanoid robot *Robota*.

Robota's Hardware *Robota* is a mini-humanoid robot of 5 degrees of freedom (DOFs). It is 45cm high, for a weight of about 1500g. It has five *Maxon-A* DC Motors with clutches to drive its two legs, two arms and head, 5 associated potentiometers, as well as 6 switches (see full description in [3]). *Robota*'s motors are driven by a PIC 16F84A microcontroller. *Robota*'s sensors are monitored by a second PIC 16F84A microcontroller. The *PocketPC* interfaces the motor and sensor cards via RS232 serial connection. Motor and sensor cards are interfaced through SPI protocol. The PC, shown in dotted lines in Figure 3, is used only for programming the *PocketPC*.

Robota's Applications The *Robota* project aims at developing an educational high-tech toy that exploits multi-modal means of human-robot interaction, such as speech and vision. *Robota* is currently being used at the undergraduate level for hands-on robotics practicals and in pilot studies with normal children and children with autism [3, 7].

Studies with autistic children investigate the use of imitation games to help them learn coordinating, and interactive skills fundamental to social interactions

[8]. In these games, the robot mirrors the user's movements of the arms and face (see Figure 4).

2.3 Language Acquisition Game

Fig. 4. *Robota* mirrors the motion of the user's arm. A PDA with a camera, mounted on *Robota*'s belly, tracks the motion of the user's arms and face continuously. A speech processing engine running on the PDA tracks the user's verbal input. The robot learn the meaning of the vocal utterance (e.g. "This is your arm") by associating the user's verbal input with its arms and face motion.

We developed a language learning game for the *iPAQ 3850 PocketPC*. In this application, the robot can learn, through a simple imitation game, a vocabulary to describe its body features and its perception of external objects. A built-in imitation module allows the robot to imitate (in mirror fashion) the user's motion of the arms and the face (see Figure 4 and 1). The robot associates the user's vocal utterance with its visual perceptions of movement and with the motor commands executed during the imitation. Once the robot has correctly learned the meaning of a word, it can then execute the motors commands associated with that word: hence, performing the correct action upon verbal command.

Social interactions have structure that can be exploited to simplify the implementation of the language acquisition game. We have implemented two fundamental means of human social interaction: *imitation* and *turn taking*. Imitation is an attentional mechanism [4, 2]. Through the imitation game, the user can force the robot to go through a specific set of perceptions. In our application, the imitation game is used by the teacher, e.g. to lead the robot to perceive the action of *lifting up an arm* or to watch a specific object by looking in a specific direction. Thus, the imitation game focuses the robot's attention on the relevant visual features, reducing importantly the amount of storage required for visual representation, and, therefore, increasing the speed of learning.

Turn taking allows to clearly separate the act of learning and the act of repeating. In the architecture of our application, *the learning phase*, during which the robot imitates the user and acquires knowledge, and *the retrieval phase*, during which the robot reproduces what it has learned, are clearly separated. The two key-sentences "Listen!" and "Try it now!" are used to switch the robot's controller into either the learning mode or the rehearsal mode, respectively.

2.4 Control Architecture

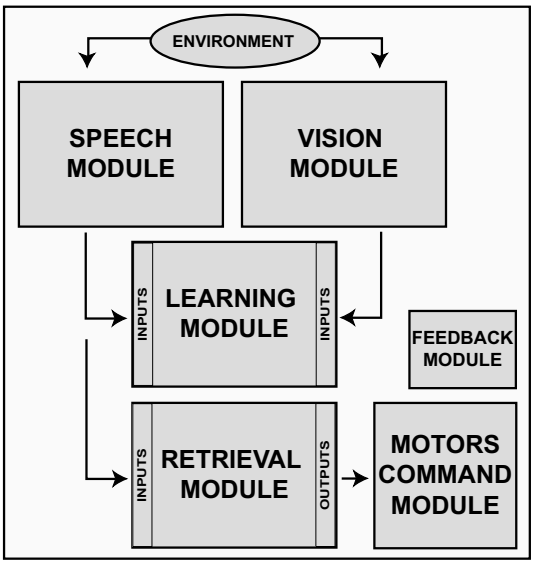

Fig. 5. Control architecture for the PDA language acquisition game.

Figure 5 shows a schematic of the control architecture of the PDA language learning game. A *speech module* and a *vision module* perform a preprocessing of the speech and camera data to simplify the robot's perception of the environment prior to learning. Speech and vision are then associated, and can be retrieved to control the robot.

Speech Module Sentences and words from the speech stream are extracted by the *CONVERSAY* Automatic Speech Recognition (ASR) engine, using preprogrammed syntactic rules. The syntax is described as a set of rules (see Figure 6). Multiple sentences can then describe a same meaning. In our system, only a subset of keywords are kept for further processing by the *learning module*. These keywords are shown in dark gray in Figure 6. With *Robota*, for example,

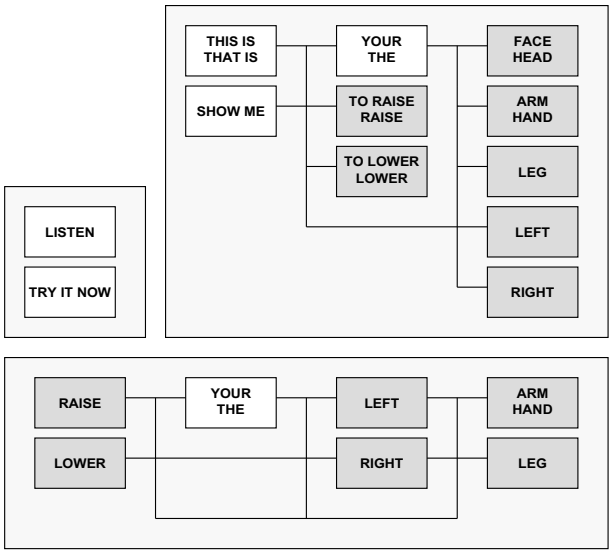

Fig. 6. Representation of the grammatical rules used by the speech recognition system in the *Robota* application. The grammatical rules are defined as rules of transition. The keywords retained for further processing are shown in dark gray.

when the user says "This is your face", the ASR detects the use of an indexed grammar. In the present example, the grammar, encoded by the programmer, specifies that "This is your" is always followed by a noun, here "face". Among the list of nouns that the ASR is programmed to recognize, the word "face" is in this example the keyword that is extracted and processed for learning. The advantage of a syntax definition instead of a list of sentences is that the description is shorter, use less computational power, and can generalize sentences. The user can then omit unimportant words without perturbing the system. For example, both sentences "This is arm" and "That is your arm" will extract the keyword "arm". In Figure 6, three different syntax categories are described. The top-left category gives order to switch between learning (sentence "listen") and retrieval phase (sentence "try it now"). The top-right category shows the grammatical model for learning and retrieval of the keywords. The bottom category show how to combine different keywords, when the system has been sufficiently trained. Ordering "Lower your left arm" to the robot will use at the same time keywords "lower", "left" and "arm".

Vision Module The *vision module* of *Robota* grabs images of the upper part of the user's body, including the head, arms, and shoulders. It tracks the vertical movements of both arms and the horizontal movements or rotation of the head. Tracking of the arms is based on luminosity and optical flow detection. The luminosity is extracted from the pixels' RGB color intensity.

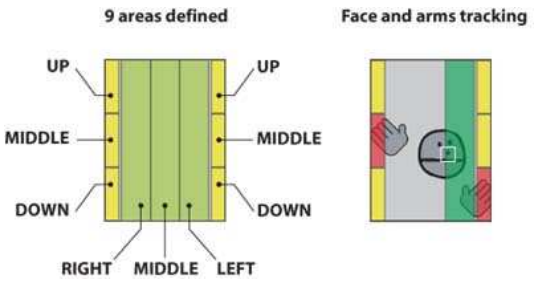

Fig. 7. Defined areas for the arms and face tracking in the *Robota* application. The vision processing is performed by the PDA.

The position of each arm and of the head is determined according to three areas only (see Figure 7), in order to ensure robust detection of movements. The *PocketPC* screen displays the image of the user. The areas, where the tracking of the arms and the face take place, are highlighted in yellow and green. Areas highlighted in red show the current estimated location of the arms (or hands). Figure 2 show screenshots of the application on the *PocketPC* screen. The two pictures show that the distance between the user and the camera can vary, without major disruption of the tracking. Indeed, when the user is close to the camera, the system tracks the user's hands, and when the user is farther away, the system tracks the user's arms.

At the initialization ($t = 0$), a reference value S representing the sum of pixels luminosity l, weighted by a factor w specific to each tracking area, is calculated as follows:

$$S = \sum_{k=1}^{6} \sum_{x=1}^{15} \sum_{y=1}^{52} l_{kxy}^{t=0} \cdot w_k$$

At each time t and for each tracking area $K = \{1, .., 6\}$ of the arms, a luminosity coefficient s_K^t is calculated as follows:

$$s_K^t = \frac{\sum_{x=1}^{15} \sum_{y=1}^{52} l_{Kxy}^t \cdot w_K}{S}$$

where l_{kxy}^t stands for the pixel luminosity of area k, at time t and position x and y in the area (each area is 15 pixels width by 52 pixels height). w_k is a weight value of the area k, that represents a factor to reduce the luminosity inhomogeneity, due to optical effects of the camera (areas on the peripheral part looks darker). w_k is fixed experimentally, equals to 1.0 for the 4 areas at the corner of the images, and equals to 1.2 for the areas in the middle.

When the user raises his arm, the system tracks the upper part of the arm, and the vision system only selects the upper area that the arm has reached. As the lower part of the arm can pass through the other areas, when doing a natural

gesture, the following algorithm has been used. For each arm, if we name the upper area 1, the middle area 2, and the bottom area 3, the selected area j at time t, used for learning, is defined by:

$$j^t = \min_{i \in \{1,2,3\}} \{i \mid s_i^t > T\}$$

The values s_i^t are compared to a threshold value T, fixed experimentally equals to 0.2. The area that exceeds the threshold with the lowest index is selected.

The tracking of the face takes inspiration from the works of Gorodnichy [6]. They show that the nose is the most robust feature for tracking the face, when using a low-resolution camera. The method is based on a local template matching, with an intensity pattern designed to match the tip of the nose. The symmetry and convexity of the nose feature provide a template that is almost the same during head rotation, and that does not change much with the head moving toward and away from the camera. In our application, the nose feature is defined as a small rectangular area of 5 pixels width by 2 pixels height. The size of the tracking area was optimized manually, so as to ensure a sufficient number of pixels for processing, while keeping the speed of processing fast enough for on-line tracking. The position of the nose/face used for the imitation is determined according to three areas (see Figure 7).

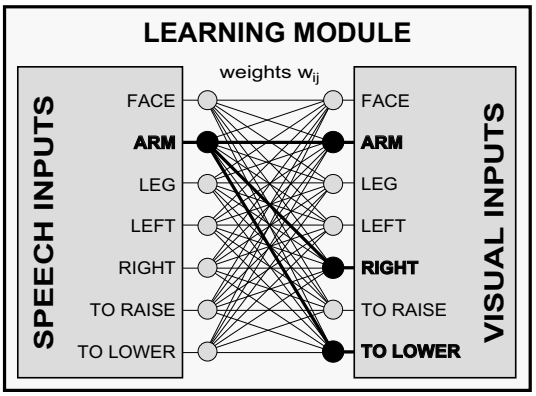

Fig. 8. The *learning module* associates vision and speech inputs in an artificial neural network. In this example, the user has said "This is your arm" and has lowered his left arm. *Robota* imitates, and lowers its right arm (mirror imitation). The connection weights between co-activated inputs are increased.

Learning Module The *learning module* performs associations of the key features extracted by the *speech* and *vision module*s. A 1-layer Artificial Neural

Networks (ANN) performs the association, using Hebbian learning. The weight of the connection between two neurons is increased if these two neurons are co-activated. The state of a neuron can be only 0 or 1.

If we note x_i the state of the speech input i and y_j the state of the vision input j, the increase of the connection weight w_{ij} between inputs i and j is given by:

$$\Delta w_{ij} = x_i \cdot y_j$$

When another learning or retrieval cycle takes place, the ANN is updated, and the neurons states are reset to the null value.

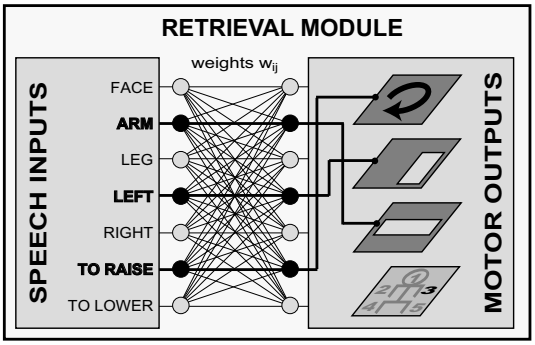

Fig. 9. The *retrieval module* processes the speech inputs, and outputs motors command to control the robot.

Retrieval Module The *retrieval module* is used to test the correctness of the learning, by requesting the robot to demonstrate its new skills upon vocal command. The neurons of the speech part act as inputs, similarly to what is happening in the learning phase. In this mode, the neurons corresponding to the vision inputs in the learning phase, act as outputs to activate motors commands. Based on the neural activity of the inputs, the ANN computes the neural activity of the outputs, following a winners-take-all mechanism.

For each activated input i ($x_i = 1$), we compute the index k where the weight values between i and j are maximum, to extract only the most relevant associations:

$$k = \{j \mid \max_{j \in \{1,2..7\}} w_{ij}\}$$

The output states y_k can then be activated according to a threshold on the weights (the threshold value is fixed by hand, and takes the value 1):

$$y_k = \Theta(w_{ik}, 1)$$

Motors Command Module The *motors command module* interprets the output of the ANN neurons. According to the activity of each neuron, it combines motors masks and basic motors commands, that have been pre-programmed, to produce the final motion. The sentence "Raise left arm" gives 3 output commands (see Figure 9). One that defines the sense of rotation of the motor (e.g. the sense corresponding to the word "raise"). One that applies a mask to let only the arms motors available, and one that applies a mask to let only the left part of the body motors available. The superposition of the two masks implies that only the left arm motor will be rotated to raise the corresponding arm.

Feedback Module The *feedback module* gives a vocal and visual feedback to the user. The vocal feedback is implemented by letting the speech synthesizer repeat the keyword extracted by the ASR from the user's sentence. If the keyword recognized by the ASR is not the correct one, the user can prevent an incorrect learning by repeating the sentence until the correct word is extracted. Visual feedback is given both by the display of the user's image on the *PocketPC* screen, and by the robot imitating the user's gesture. If the imitation is incorrect (for example *Robota* moves the left arm while the user moves the head), the user can adjust her/his posture with respect to the three tracking areas, highlighted on the camera image (see Figure 2). The imitation of the gesture informs the user about a correct or incorrect visual recognition of the arms and face movement.

3 PDA Application for DB Human-Sized Robot

3.1 DB Robot

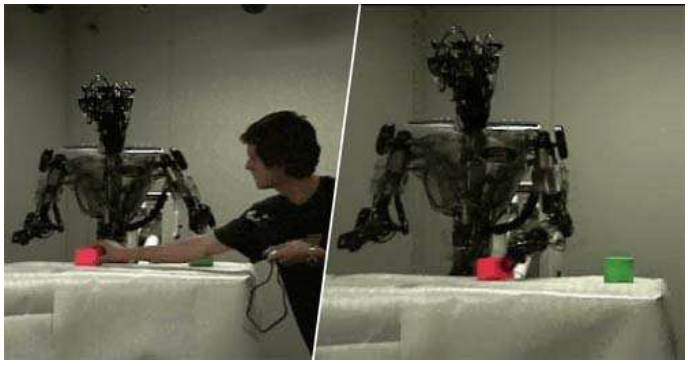

Fig. 10. **Left:** The demonstrator teaches the robot words to label the boxes and the directions in which to move the box. The robot tracks the direction of motion of the boxes and associates the teacher's verbal utterance with its observations. **Right:** After learning and upon verbal request from the teacher, the robot reproduces the motion of the named box in the requested direction.

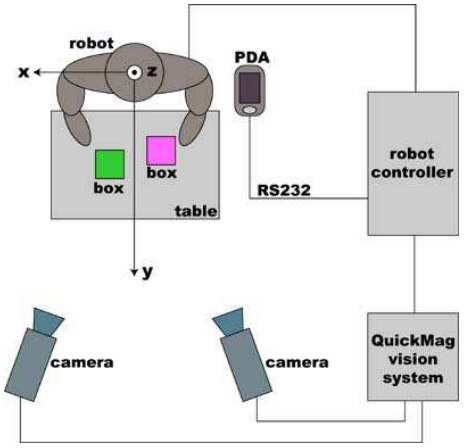

Fig. 11. DB robot application setup.

In collaboration with the *Humanoid Robotics and Computational Neuroscience* department (HRCN) at the *Advanced Telecommunication Research* institute (ATR), we implemented the PDA-based language learning game on *DB*, an anthropomorphic hydraulic robot with 30 DOFs. In these experiments, the robot learns the names of two boxes and of directions of movements (left and right). Once the robot has correctly learned the words, it can push the boxes in a desired direction upon verbal command.

Figures 10 and 11 show the experimental setup for the DB robot, where a table is placed in front of the robot, with two boxes of different colors on it (green and pink, to facilitate color tracking). An external stereo vision system tracks the boxes position.

DB is mounted at the pelvis. It is 1.85 meters tall, weighs 80 kg, and is driven by 25 linear hydraulic actuators and 5 rotary hydraulic actuators. Each arm has 7 DOFs. The vision system consists of 2 cameras fixed on the ceiling and facing the robot (see Figure 11). A color blobs tracking system generates blob position information at 60 Hz. In our application, the *vision module* of *DB* robot extracts relevant changes in the boxes' position and direction of displacement. Similary to what what was done with *Robota*, see Section 2.4, speech and vision inputs are associated in a ANN, computed on-board of the PDA. During retrieval, the output neurons activate absolute goal positions, and relative sequences of movement, such that the robot can push the requested box in the requested direction. The communication from PDA to robot is still assured by a RS232 serial interface in this experiment, because the aim of this first experiment was to evaluate the use of the PDA as a remote control. Wireless capabilities of the *iPAQ* will be used in further experiments.

4 Experiments and Results

	Speech Recognition System			Vision System		
	Recognition rate	Error rate		Recognition rate	Error rate	
		No detection	False detection		No detection	False detection
Experiment 1	85%	13%	2%	-	-	-
Experiment 2	77%	23%	0%	82%	16%	2%
Experiment 3	86%	7%	7%	-	-	-

Table 1. Experiments results for the speech recognition system and the vision system processed by the PDA setup.

Table 1 gives the results from 230 trials over 3 experiments, conducted with the same user. Experiments 1 and 2 have been conducted with *Robota*. The first experiment presents a typical learning scenario where the robot learns the labels for its limbs (head, arm and leg), the difference between left and right, and directions of movement (lifting up and down). This experiment uses the inputs from the potentiometers instead of the vision system for knowing the location of the limbs. Experiment 2 presents the same learning scenario, using the vision system presented in Section 2.4. Results from these two experiments show that the error rate of the speech recognition system increases, when the processing rate of the *iPAQ* is disturbed by another task (i.e. by the vision process in Experiment 1). Experiment 3 has been conducted with the robot *DB*. There, the robot learns names for objects (two color boxes), and for directions in which to move the boxes (left/right). There are no data on the efficiency of the vision system, since this one was an external system, not performed on the *iPAQ* (see Chapter 3.1).

With *Robota*, we conducted trials with adults (native and non-native English speakers), in different rooms and under different lighting conditions, and with two little girls of 4 and 9 years. These trials showed that, when the user is trained and understands the turn-taking model and the vision system principles, and when the environment is constrained (clear background, no moving objects in the background), the system performs correct visual and speech recognition, and the learning is immediate. However, the robustness of speech and visual processing decreases quickly in a moving and noisy environment (e.g. when there are people standing behind the user, or with strong directional light, resulting in a saturation of the camera).

Experiments with children showed that, at nine years old, the child is perfectly able to understand and work with the system. At four years old, the child could understand the game principle but had difficulty playing the game. The visual system is, indeed, too constrained, preventing the kid to play freely the imitation game. The speech recognition is very poor for children voices.

We conducted a first set of trials with autistic children, in collaboration with Jacqueline Nadel at the Hôpital de la Salpêtrière in Paris in May 2003. These trials showed that the system needs to be improved to be successfully used by the children. The vision tracking system must be improved in order to track more robustly the child's movements, even if the child is not centered in the camera field of view.

The experiments conducted with the robot *DB* showed that the PDA offers user-friendly means to control that robot, and that further developments are worth to be conducted, using the *iPAQ* wireless capabilities.

5 Further Developments and Conclusion

This work intended to investigate the feasibility of porting a simple language and imitation game on a PDA. On the one hand, this work allowed to investigate the use of the PDA interface to teach a full size humanoid robot a basic vocabulary, which can then be used to direct the robot in a simple manipulation task.

On the other hand, this work investigated whether this embarked system could replace the previous PC+WebCam setting for *Robota*. Having an all-in-one setup for *Robota* is advantageous for the experiments conducted with disabled and autistic children [8]. It offers flexibility and a hands-free device, more acceptable for these children. First trials with normal and autistic children showed, however, that the robustness of the PDA speech and vision systems have to be improved before more tests are conducted with children younger than 9.

The limited processing power of the *iPAQ* constrained us to implement very simple vision and speech processing. Each module of our system has been developed to be separately optimized. This allows us to easily increase the efficiency of one module, taking thus advantages of the new features, accessories, and processing power of the upcoming PDAs. We are currently investigating the integration of more powerful vision algorithm for skin color and motion tracking on a new PDA device, improving the programming setup, and possibly using the PDA wireless capabilities.

In conclusion, the possibilities offered by the *iPAQ* in terms of computation and multi-modal sensor interfaces show that applications that deal with speech and vision are possible, at an affordable price for toy robots. We have also shown that such handheld devices can be exploited to interact with complex humanoid robots, offering the advantage of a user-friendly device that can be moved freely.

The closest work to ours is that of Okada et al. [14], who investigated a PDA application to provide operator's assistance to a humanoid robot during an object manipulation task. While the manipulation task performed by the *DB* humanoid robot in our work is much simpler than that implemented by Okada et al, our application went farther in exploring the use of the PDA for teaching (and not simply directing) the robot, by combining both vision and speech processing. The learning possibilities (i.e. the vocabulary) were very limited, because the purpose of the experiment was first to evaluate the functionality of a PDA to

remotely control the robot. In future work, we will investigate learning of regular sentences to direct more manipulation tasks.

Acknowledgments

We are very grateful to Dr. Gordon Cheng and Professor Mitsuo Kawato of the *Humanoid Robotics and Computational Neuroscience* department (HRCN), at the *Advanced Telecommunication Research* Institute (ATR), for having given us access to the humanoid robot *DB*. Lots of thanks to Professor Stefan Schaal for providing access to the *DB* robot programming environment. This work was supported by grants from the Swiss National Science Foundation, the EPFL student travel funds, and the ATR travel funds.

References

1. Ronald C. Arkin, Masahiro Fujita, Tsuyoshi Takagi, and Rika Hasegawa. An ethological and emotional basis for human-robot interaction. In *Proceedings of the IEEE/RSJ Int. Conference on Intelligent Robots and Systems (IROS2002)*, 2002.
2. A. Billard. Imitation: a means to enhance learning of a synthetic proto-language in an autonomous robot. In C. Nehaniv and K. Dautenhahn, editors, *Imitation in Animals and Artifacs*, pages 281–311. MIT Press, 2001.
3. A. Billard. Robota: Clever toy and educational tool. *Robotics & Autonomous Systems*, 42:259–269, 2003.
4. A. Billard and K. Dautenhahn. Grounding communication in autonomous robots: an experimental study. *Robotics and Autonomous Systems, Special Issue on Scientific methods in mobile robotics*, 24:1-2, 1998.
5. C. Breazeal and B. Scassellati. Infant-like social interactions between a robot and a human caretaker. *Special issue of Adaptive Behavior on Simulation Models of Social Agents*, 1998.
6. Gorodnichy D. On importance of nose for face tracking. In *Proceedings of the IEEE Int. Conference on Automatic Face and Gesture Recognition (FGR'02)*, Washington, D.C., 2002.
7. K. Dautenhahn. Robots as social actors: Aurora and the case of autism. In *Proc. CT99, The Third International Cognitive Technology Conference, August, San Francisco*, 1999.
8. K. Dautenhahn and A. Billard. Games children with autism can play with robota, a humanoid robotics doll. In *In Proceedings of the 1st Cambridge Workshop on Universal Access and Assistive Technology, March*, 2002.
9. M. Ehrenmann, O. Rogalla, R. Zllner, and R. Dillmann. Teaching service robots complex tasks: Programming by demonstation for workshop and household environments. In *Proc. of the IEEE Int. Conf. on Field and Service Robotics. (FRS), Finnland 2001.*, 2001.
10. J.-O. Eklundh and H. Christensen. Computer vision:past and future. *Informatics: 10 Years Back, 10 Years Ahead (R. Wilhelm, ed.), Lecture Notes in Computer Science*, pages 328–340, 2001.
11. T. Fong, C. Thorpe, and B. Glass. Pdadriver: A handheld system for remote driving. In *IEEE International Conference on Advanced Robotics*, Coimbra, Portugal, July 2003.

12. M. Fujita and H. Kitano. Development of an autonomous quadruped robot for robot entertainment. *Autonomous Robots, 5*, pages 7–18, 1998.
13. H. Hüttenrauch and M. Norman. Pocketcero - mobile interfaces for service robots. In *Proceedings of Mobile HCI 2001: Third International Workshop on Human Computer Interaction with Mobile Devices*, Lille, France, 2001.
14. K. Okada, Y. Kino, M. Inaba, and H. Inoue. Visually-based humanoid remote control system under operator's assistance and its application to object manipulation. In *Proceedings of the IEEE International Conference on Humanoid Robots*, Karlsruhe, Germany, September 2003.
15. S. Schaal. Nonparametric regression for learning nonlinear transformations. In H. Ritter and O. Holland, editors, *Prerational Intelligence in Strategies, High-Level Processes and Collective Behavior*. Kluwer Academic Press, 1999.
16. J. Suomela and A. Halme. Cognitive human machine interface of workpartner robot. In *Intelligent Autonomous Vehicles 2001 Conference (IAV2001)*, Sapporo, Japan, 2001.
17. D. Williams and D.H. Williams. *PDA Robotics: Using Your Personal Digital Assistant to Control Your Robot*. McGraw-Hill Companies, June 2003.