2010 IEEE International Conference on Robotics and Automation

Anchorage, Alaska, May 3-8, 2010

Proceedings of the ICRA 2010 Workshop on Multimodal Human - Robot Interfaces

Editors:

José M. Azorín

José L. Pons

Anchorage, Alaska, May 3, 2010

Objectives

Haptic interfaces, natural language and gestures have traditionally been used to interact with robots. However, in last years, new modalities of interaction have emerged, like EMG and EEG interfaces. The current scenario is one of transition from the industrial workplace towards increasing interaction with the human operator in other scenarios. This means that interaction with humans is expanding from a mere exchange of information (in teleoperation tasks) and service robotics to a close interaction involving physical and cognitive modalities. It is in this context where multimodal interfaces combining different kind of interaction modalities play a crucial role. Multimodal interfaces increase usability (the weaknesses of one modality are offset by the strengths of another) and they have implications for accessibility (a well-designed multimodal application can be used by people with a wide variety of impairments).

This workshop will provide an overview of the most recent advances about human-robot multimodal interfaces and it will explore new directions in the field, with a particular focus on interfaces for disabled people. The workshop will form an ideal environment for the emerging community to meet and exchange ideas.

Organizers

José M. Azorín (IEEE Member)
Biomedical Neurorobotics Group
Universidad Miguel Hernández de Elche
Avda. de la Universidad, s/n
03202 Elche, Alicante, Spain
jm.azorin@umh.es

José L. Pons (IEEE Member)
Bioengineering Group
Instituto de Automática Industrial, CSIC
Ctra. Campo Real, km. 0,200
28500 Arganda del Rey, Madrid, Spain
jlpons@iai.csic.es

Homepage

http://nbio.umh.es/vr2/eventos/icra2010/home.html

Program

9.00 - 9.15: Introduction: Multimodal Human-Robot Interfaces. José M. Azorín, José L. Pons (Organizers of the workshop).

PART I: TECHNOLOGIES

9.15 - 9.55: From haptic human-human to human-robot interaction - challenges and selected results. Angelika Peer (Institute of Automatic Control Engineering, Muenchen, Germany).

9.55 - 10.30: Modular haptic device for bimanual virtual manipulation. Ignacio Galiana (Politechnical University of Madrid, Madrid, Spain)

10.30 - 10.50: Coffee Break

10.50 - 11.25: Graphical and semantic interaction by means of gestures. Alícia Casals (Universidad Politécnica de Cataluña, Barcelona, Spain).

11.25 - 11.55: Haptic and ocular human-robot interface. José M. Azorín (Biomedical Neuroengineering Group, Universidad Miguel Hernández de Elche, Elche, Spain).

11.55 - 12.30: From Rehabilitation robots to Neuroprosthetics and neurobototics. Issues pending in BMI Systems (José L. Pons, Bioengineering Group, Instituto de Automática Industrial, CSIC, Arganda del Rey, Madrid, Spain).

12.30 - 14.00: Lunch Time

PART II: APPLICATIONS

14.00 - 14.30: Robotic wheelchair controlled by a multimodal interface (Teodiano Freire-Bastos, Departamento de Engenharia Elétrica, Universidade Federal do Espírito Santo, Vitoria, Brazil).

14.30 - 15.00: Multimodal HMI interface in Neuroborotic management of Tremor (Eduardo Rocon de Lima, Bioengineering Group, Instituto de Automática Industrial, CSIC, Arganda del Rey, Madrid, Spain)

15.00 - 15.30: A BNCI-driven rehabilitation of gait in stroke patients (Juan C. Moreno, Bioengineering Group, Instituto de Automática Industrial, CSIC, Arganda del Rey, Madrid, Spain).

15.30 - 15.50: Coffee Break + Poster Exhibition

PART III: POSTER PRESENTATION

15.50 - 17.30: Short presentations of the participants:

1. An Ontology-based Multimodal Communication System for Human-Robot Interaction in Socially Assistive Domains. Ross Mead, Jerry B. Weinberg, and Maja J Mataric (University of Southern California, Southern Illinois University Edwardsville, USA).

2. Exploring Multimodal Interfaces For Underwater Intervention Systems. J. C. Garcia, P. J. Sanz, Member, R. Marin, and O. Belmonte (Universitat Jaume I, Spain)

3. Enhancing Collaborative Human-Robot Interaction Through Physiological-Signal Based Communication. Susana Zoghbi, Chris Parker, Elizabeth Croft and H.F. Machiel Van der Loos (University of British Columbia, Canada).

4. Towards an Enabling Multimodal Interface for an Assistive Robot, Martin F. Stoelen, Alberto Jardon, Fabio Bonsignorio, Juan G. Victores, Concha Monje and Carlos Balaguer (Universidad Carlos III de Madrid, Spain).

5. Humanoid robot skill acquisition through balance interaction between human and humanoid robot. Jan Babič, Erhan Oztop (Jožef Stefan Institute, Slovenia, ATR Computational Neuroscience Laboratories, Japan).

6. Robot Reinforcement Learning using EEG-based reward signals. I. Iturrate, L. Montesano and J. Minguez (Universidad de Zaragoza, Spain).

7. Temporal gesture recognition for human-robot interaction, Markos Sigalas, Haris Baltzakis and Panos Trahanias (Foundation for Research and Technology - Hellas, Heraklion, University of Crete, Greece).

8. BCIs and Mobile Robots for Neurological Rehabilitation. practical applications of remote control. Luigi Criveller, Emanuele Menegatti, Franco Piccione, Stefano Silvoni (Università degli Studi di Padova, I.R.C.C.S. San Camillo Venice, Italy).

9. Safe Human-Robot Interaction based on a Hierarchy of Bounding Volumes. Juan Antonio Corrales, Fernando Torres, Francisco Andrés Candelas (University of Alicante, Spain).

10. Hand Gesture Recognition for Human Robot Interaction in Uncontrolled Environments, Jong Lee-Ferng, Javier Ruiz-del-Solar, Mauricio Correa, Rodrigo Verschae (Universidad de Chile, Chile).

11. Vision-based gesture recognition interface for a social robot. J.P. Bandera, A. Bandera, L. Molina-Tanco, J.A. Rodríguez (University of Málaga, Spain).

Table of contents

Proceedings of the IEEE ICRA 2010
Workshop on Multimodal Human-Robot Interfaces
Anchorage, Alaska, May, 2010

From Haptic Human-Human to Human-Robot Interaction - Challenges and Selected Results

Angelika Peer, Zheng Wang, Jens Hölldampf, and Martin Buss
Institute of Automatic Control Engineering
Technische Universität München
D-80290 München, Germany
Angelika.Peer@tum.de, wang@tum.de, jens.hoelldampf@tum.de, mb@tum.de

***Abstract*— Adding physicality to virtual environments is considered a prerequisite to achieve natural interaction behavior of the user and can be achieved by usage of appropriately designed and controlled haptic devices, as well as by implementation of sophisticated haptic rendering algorithms. While in the past a variety of haptic rendering algorithms for the interaction with passive environments were developed, the interaction with active environments like the physical interaction with a virtual character is rarely investigated. Such kind of physical interactions pose a number of new challenges compared to the interaction with passive environments as the human expects to interact with a character that shows human-like behavior, i.e. it is able to estimate human intentions, to communicate intentions, and to adapt its behavior to its partner. On this account, algorithms for intention recognition, interactive path planning, and adaptation are needed when implementing such interactive characters. In this paper two different approaches for the synthesis of interactive behavior are reviewed, an engineering-driven and an experimental-driven approach. Following the experimental-driven approach haptically interacting partners are synthesized following a three step procedure record-replay-recreate. To demonstrate the validity of this approach it is applied to two prototypical application scenarios, handshaking and dancing.**

***Index Terms*— haptic human-robot interaction, physical human-robot interaction, haptic rendering**

I. Introduction

While today's virtual environments are able to provide high quality visual and auditory feedback, most of them still lack physicality, the state or quality of being physical and follow physical principles: People can penetrate into walls, objects, and characters, lift objects without feeling their weight, stroke objects without feeling their texture and socially interact with characters, without feeling forces when being in physical contact.

Since virtual environments are used for simulation, training, rehearsal, and virtual gatherings, they should provoke natural interaction behavior of the user to attain their expected effect. Physicality is considered one of the main prerequisites to achieve this. Thus, high-quality haptic feedback is desired, which calls for appropriately designed and controlled haptic devices, as well as a sophisticated haptic rendering algorithms.

In the past years a variety of haptic interfaces have been developed and presented in literature, see [1], [2] for an overview. Systems either provide kinesthetic or tactile feedback and are ceiling, floor, desktop, or body-grounded. Most devices, however, have been designed and optimized for a specific application only. Thus, several devices would be necessary to cover the various types of interactions required when realizing immersive virtual environments.

When bringing physicality to virtual environments high-quality haptic rendering algorithms are needed. Haptic rendering has been a very active field of research and a variety of algorithms for interaction with passive environments have been developed in the past. This includes geometric rendering algorithms for single point contact with polygonal [3], [4], parametric [5], and implicit surfaces [6] or volumetric objects. Advanced versions also consider point interaction with deformable objects, line interaction or interaction between polygons. Recently, also direct haptic rendering from measurements has been studied intensively [7]. Beside rendering algorithms for kinesthetic feedback also a number of texture rendering algorithms exist, see [8], [9]. Interested readers please refer to [10] for a comprehensive overview of state-of-the-art haptic rendering algorithms.

While all the aforementioned rendering algorithms assume interactions with passive environments only, rendering of active environments has been rarely studied in literature. Simulation of actuated systems or virtual characters that haptically interact with the human user are typical examples for such active environments. In literature only a few examples for such systems exist, e.g. [11], where an actuated car door is rendered.

We aim for rendering a virtual, interactive character like a handshaking or dancing partner. Compared to haptic rendering of passive environments and the rendering of actuated systems, rendering of a virtual character that can physically interact with humans poses a variety of new challenges, because the human expects to interact with a character that shows human-like behavior, i.e. it is able to estimate human intentions, to communicate intentions, and to adapt its behavior to its partner. Since human intention (the way how the human desires to carry out the task) is hidden in the human mind, it must be inferred by analyzing measured force and motion data. When analyzing this data, however, we have to call into attention that the human can change his behavior by either changing his execution plan or by adapting his mechanical impedance. Both, execution

plan and mechanical impedance can significantly vary over time and thus lead to changes in the force and motion data. Intelligent intention recognition algorithms, however, need to be able to distinguish between these two cases, because only then, the behavior of the virtual character can be adapted as desired by either changing the desired path or the implemented compliance provided by the haptic interface. Taking into account these challenges arising when rendering physical interactions with interactive characters, it is clear that this requires modules that are not necessary when rendering passive environments or actuated systems, namely modules for intention recognition, interactive path planning, and adaptation.

First approaches in this direction can be found in the field of physical human-robot interaction where a robot is supposed to assist the human operator while jointly performing a joint transporting or social interaction task. Starting from purely passive followers as presented in [12], control schemes with varying impedance parameters [13], [14] and controllers that introduce additional virtual constraints [15] were developed, leading finally to active robot partners that can estimate human intention [16]–[18] and based on these estimations change their interaction behavior by taking different roles [19].

In the following section we will present our approach to realize a haptically interacting partner, while Sections III and IV exemplarily demonstrate this approach for two prototypical physical interaction tasks, handshaking and dancing.

II. Approaches to synthesize interactive behavior

When synthesizing interactive behavior, two completely different approaches can be adopted: an engineering-driven and an experimentally-driven approach. A typical engineering-driven approach implements control strategies which are e.g. based on heuristics, optimality criteria or stability criteria. One of the drawbacks of such an approach is that it does not necessarily guarantee that the resulting interaction patterns are human-like which complicates the recognition of the virtual partner's intention, the building of its mental model and thus prediction of its action. Beside this, also natural communication of ones own intentions can be affected by the usage of this approach as the artificial virtual partner often lacks the ability to understand and interpret them.

These limitations can be overcome by an experimentally-driven approach which aims at i) recording data during human-human interaction and using this data to replay [20] and synthesize interactive behavior [21], [22], or ii) by recording and studying human-human interaction and transferring knowledge gained from the analysis of this data to human-robot interaction as recently explored in [18], [23]–[28].

In the EU project Immersence[1], we adopt the experimentally-driven approach to synthesize interactive behavior following a three step procedure record-replay-recreate. In the recording phase force and motion signals resulting from physical interaction of two humans are recorded. In the second step, this data is simply replayed by using a haptic interface. Since, a pure replay lacks the ability to adapt to the human partner and thus natural interaction behavior cannot be achieved [20], a third phase, the recreation phase is introduced which aims at synthesizing real interactive behavior. The following two sections will demonstrate this approach for two prototypical application scenarios, handshaking and dancing.

[1]www.immersence.info

III. Application Scenario: Handshaking

In [21] we studied handshaking with a virtual, visually and haptically rendered character, see Fig. 1. So far, only few authors investigated handshaking: In [29] the first tele-handshake using a simple one degree-of-freedom (DOF) device was created while [30] generated handshake animations from a vision system. Remarkably, only few people viewed handshaking from a force/motion interaction aspect: in [31] the authors took the oscillation synchronization approach to realize human-robot handshaking and in [32] the authors focused on the approaching and shaking motions of a handshaking robot.

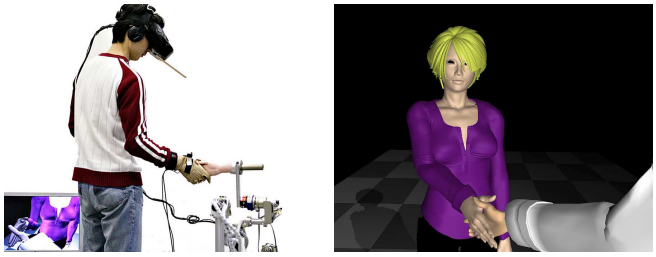


Fig. 1. Handshaking with a virtual interactive character

In the following sections we will review our approach to synthesize an interactive handshaking partner following the three-step procedure record-replay-recreate delineated above.

A. Record

In the recording phase in total a number of 900 human-human handshakes of 24 male college students were performed and position and force data was recorded during interaction, see Fig. 2. To measure the interaction force special data gloves [33] were used while position was recorded by an optical tracking system. No instructions how to perform the handshakes were given to achieve natural interaction behavior.

B. Replay

In the replay phase trajectories recorded during human-human handshakes were replayed by the haptic interface ViSHaRD10 [34]. To improve naturalness of interaction, a compliant controller was additionally used to imitate human arm stiffness, see Fig. 3. If the human performs similarly

Fig. 2. Recording of human-human handshakes

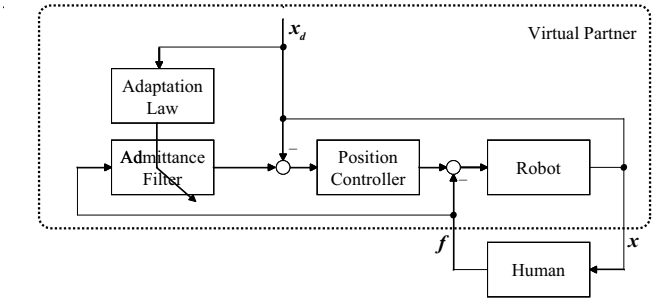


Fig. 3. Replay of human-human handshakes using a position-based admittance controller with time varying parameters

at each handshake, the same desired trajectory leads to similar force profiles for each handshake and hence provides the human a similar handshaking experience compared to the human-human case where the reference trajectory was recorded. Thus, provided that human participants are good repetitors, this replay strategy leads to natural handshakes. However, it has a fundamental limitation for realizing full interactive handshakes as it lacks the ability to alter the reference trajectory, therefore the robot can only playback motions as predefined, with the human input applied on top of it. This is clearly different from human-human handshaking, where the arms can provide compliance during interaction, while in the human mind different strategies can be selected about whether to adapt to the partner or not. On this account, an advanced more interactive handshaking controller has been developed.

C. Recreate

To synthesize an interactive behavior of the virtual handshaking character we propose a double-layered control scheme consisting of a low-level and a high-level controller, see Fig. 4. The low-level controller (LLC) implements position-based admittance control and the high-level controller (HLC) updates the admittance parameters and adapts the reference trajectory depending on the actual estimated human intention. Interactive behavior is consequently achieved by three modules, the intention estimation module, the adaptation law, and the trajectory planning algorithm.

Using measured force and position data an online parameter estimator identifies human behavior parameters,

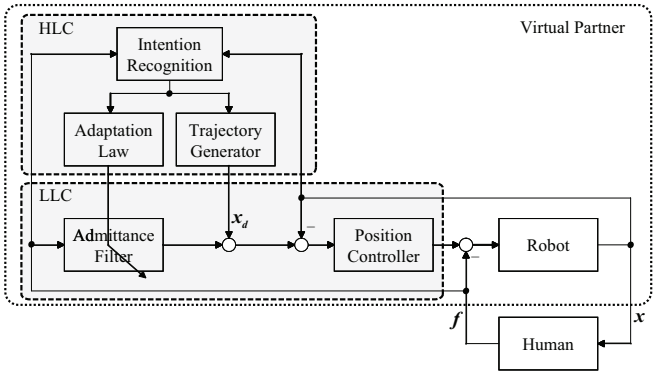


Fig. 4. Recreation of human-human handshakes using an HMM-based handshaking controller

abstracts them into symbols and feeds them into an Hidden-Markov-Model(HMM)-based intention recognition module which outputs an estimate of the current human intention. Two HMMs are defined for the estimator relecting the two opposite roles *active* and *passive*. Active indicates that the human is trying to lead the handshake, while passive means the human is following the lead of the robot. Depending on the estimated intention the robot is programmed to take opposite roles to the human partner by generating appropriate reference trajectories and altering the displayed compliance according to the implemented adaptation law. Experiments showed that the roles active and passive are not fixed for the whole interaction, but that partners switch between them several times. Using the aforementioned approach the number of switching events between the states active and passive can be easily modified by changing the thresholds of the discrete HMM. In doing so, a more or less dominant interaction partner can be realized. Interested readers are referred to [21] for a detailed description of the implemented algorithms and achieved results.

IV. Application Scenario: Dancing

In [35] we studied dancing as haptic interaction scenario, see Fig. 5. Like handshaking, dancing requires a physical coupling between partners, but differs from handshaking as i) dominance is by definition distributed unequally between partners, ii) the basic form of dancing steps is predefined, and iii) dancing figures represent cyclical movements.

Several studies concerning the analysis of human behavior while dancing are known from literature. In [36] the sensimotory coordination is examined while dancers were performing small-scale tango steps. In [37] the haptic coordination between dancers with PHANToMs is investigated. Finally, [38], [39] built a female dancing robot which follows the male and tries to estimate the next dancing step and adapts the step size on demand [40]. Our aim was to implement a haptic enabled male dancer that imitates the behavior of a real human partner. In contrast to female dancers which require the ability to understand intentions, male dancers need the ability to communicate intentions and to adapt to their partner's behavior.

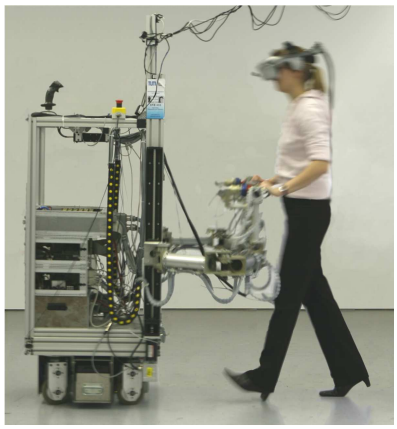


Fig. 5. Dancing with a virtual interactive character

In order to achieve this, again the three step approach record-replay-recreate was adopted as illustrated in the following subsections.

A. Record

In the recording phase semi-professional dancing couples were recorded using a motion-capture system, see Fig. 6. As dance the discofox has been chosen as it allows to have only two interaction points. In order to measure the interaction force, special adapters were constructed which connect two handles over a 6 DOF force-torque sensor and consequently allow dancing-like hand postures.

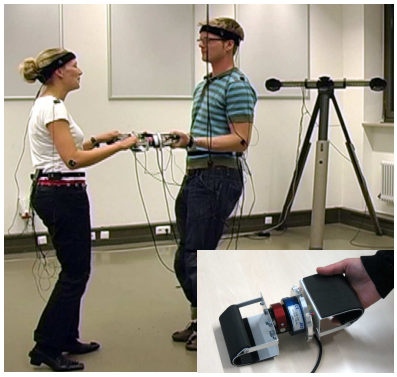


Fig. 6. Recording of dancing couples

B. Replay

In the replay phase the recorded motion data was replayed on a mobile robotic platform [41] equipped with two robotic arms [42]. The mobile platform was position controlled and the robotic arms were programmed to follow the measured positions of left and right hand recorded during human-human interaction. Adopting this approach leads to a fully dominant male dancer, which does not estimate human intentions and thus cannot adapt to the female partner.

C. Recreate

To recreate the behavior of an interactive dancing partner we implemented the vector field approach of Okada et al. [43]. It allows to synthesize a nonlinear system dynamics based on a given trajectory

$$\boldsymbol{x}[k+1] = \boldsymbol{x}[k] + \boldsymbol{f}(\boldsymbol{x}[k]) \tag{1}$$

while reducing the amount of data points needed for a typical dancing scenario.

Thus, only a few hundred parameters need to be stored, which results in a tremendous reduction of memory requirements. A further advantage is that dancing steps are parametrized and thus can easily be transformed in size and shape. This attribute will be utilized in the recreation phase described below.

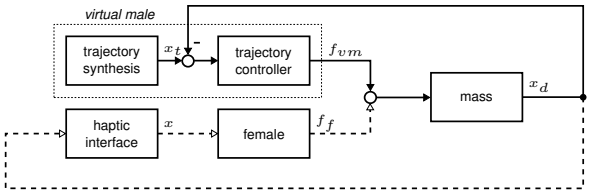


Fig. 7. Replay of human-human dancing trajectories using a trajectory generator

Switching between different dancing steps is achieved by using different dynamic subsystems for each dancing step. This can be simply realized by adding an input signal $\boldsymbol{u}$ to the nonlinear system dynamics

$$\boldsymbol{x}[k+1] = \boldsymbol{x}[k] + \boldsymbol{g}(\boldsymbol{u}[k], \boldsymbol{x}[k]). \tag{2}$$

After segmenting the recorded trajectory and introducing different levels of the input signal, synthesis can be easily performed by applying the appropriate signal level. When switching between dancing steps they are smoothly faded into each other by means of the attracting vector fields.

Since replay of generated trajectories results into partners which lack the ability to estimate human intentions and to adapt to their partner, the recreation phase was used to implement an interactive virtual partner which estimates human intention from measured interaction forces, see Fig. 8, and adapts the step size depending on them by applying the following transformation

$$\boldsymbol{y}[k] = \boldsymbol{a}[k]\boldsymbol{x}[k] + \boldsymbol{d}[k] \tag{3}$$

to $\boldsymbol{x}$[k] in (2). For a detailed description of the implemented algorithms and achieved results please refer to [35].

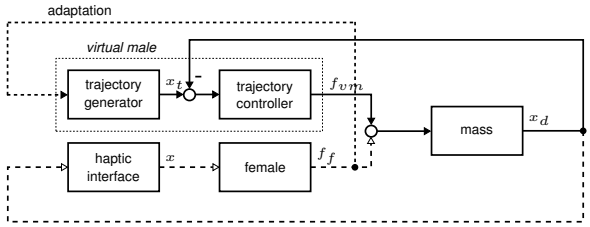


Fig. 8. Recreation of human-human dancing trajectories using an adaptive trajectory generator

V. Conclusions

Today's virtual environments often lack physicality although it is considered a prerequisite to achieve natural user behavior. Adding physicality to virtual environments requires high-quality haptic interfaces, but also advanced haptic rendering techniques that are able to render realistic haptic interactions. While in the past a variety of haptic rendering algorithms for the interaction with passive environments were developed, interaction with active environments like the physical interaction with a virtual character are still rarely studied. Such physical interactions were shown to pose a number of new challenges compared to the rendering of passive environments as the human expects to interact with a character that shows human-like behavior, i.e. it should be able to estimate human intentions, to communicate intentions, and to adapt its behavior to its partner. On this account, algorithms for intention recognition, interactive path planning, and adaptation are needed when implementing haptic interaction partners.

Two principal approaches to synthesize interactive behavior were reviewed: an engineering-driven approach and an experimental-driven approach. While the engineering-driven approach clearly lacks the ability to realize human-like interaction behavior, this can be achieved when following an experimental-driven approach which uses human-human interaction as a reference. In the presented work we followed the experimental-driven approach and adopted a three-step procedure implementing a record, replay, and recreate phase to realize a haptic interaction partner. Finally, to demonstrate the validity of the introduced approach two prototypical application scenarios, handshaking and dancing, were introduced.

Acknowledgments

This work is supported in part by the ImmerSence project within the 6th Framework Programme of the European Union, FET - Presence Initiative, contract number IST-2006-027141. For the content of this paper the authors are solely responsible for, it does not necessarily represent the opinion of the European Community, see also www.immersence.info.

References

[1] G. Burdea, *Force and Touch Feedback for Virtual Reality*. John Wiley & Sons, 1996.

[2] J. Martin and J. Savall, "Mechanisms for Haptic Torque Feedback," in *Proc. of the First Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, 2005, pp. 611–614.

[3] C. Zilles and J. Salisbury, "A constraint-based god-object method for haptic display," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1995.

[4] C. Basodogan and M. A. Srinivasan, "Haptic rendering in virtual environments," in *Handbook of virtual environments: Design, implementation, and applications*, K. Stanney, Ed. Lawrence Erlbaum Associates, 2002, pp. 117–134.

[5] T. Thompson II, D. E. Johnson, and E. Cohen, "Direct haptic rendering of sculputured models," in *Proceedings Symposium on Interactive 3D Graphics*, 1997.

[6] L. Kim, A. Kyrikou, G. Sukhatme, and M. Desbrun, "An implicit-based haptic rendering technique," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2002.

[7] A. M. Okamura, K. Kuchenbecker, and M. Mahvash, "Measurement-based modeling for haptic rendering," in *Haptic Rendering: Foundations, Algorithms, and Applications*. Ak Peters Series, 2008.

[8] C. Basdogan, C. Ho, and M. Srinivasan, "A Ray-Based Haptic Rendering Technique for Displaying Shape and Texture of 3D Objects in Virtual Environments," in *Dynamic Systems and Control Division*, 1997.

[9] M. Otaduy and M. Lin, "Rendering of textured objects," in *Haptic Rendering: Foundations, Algorithms, and Applications*. Ak Peters Series, 2008.

[10] M. Lin and M. Otaduy, Eds., *Haptic Rendering, Foundations, Algorithms, and Applications*. Ak Peters Series, 2002.

[11] M. Strolz and M. Buss, "Haptic rendering of actuated mechanisms by active admittance control," in *Haptics: Perception, Devices and Scenarios*, ser. LNCS 5024, M. Ferre, Ed. Springer, 2008, pp. 712–717.

[12] Y. Hirata and K. Kosuge, "Distributed robot helpers handling a single object in cooperation with humans," in *Proc. of the IEEE International Conference on Robotics and Automation*, 2000, pp. 458–463.

[13] T. Tsumigawa, R. Yokogawa, and K. Hara, "Variable impedance control with regard to working process for man-machine cooperation-work system," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2001.

[14] V. Duchain and C. Gosselin, "General model of human-robot cooperation using a novel velocity based variable impedance control," in *Worldhaptics*, Tsukuba, Japan, 2007, pp. 446–451.

[15] H. Arai, T. Takubo, Y. Hayashibara, and K. Tanie, "Human-robot cooperative manipulation using a virtual nonholonomic constraint," *IEEE Proc. of the International Conference on Robotics and Automation*, vol. 4, pp. 4063–4069, 2000.

[16] Y. Maeda, T. Hara, and T. Arai, "Human-robot cooperative manipuation with motion estimation," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Maui, Hawaii, 2001, pp. 2240–2245.

[17] T. Takeda, Y. Hirata, and K. Kosuge, "Dance step estimation method based on hmm for dance partner robot," *IEEE Transactions on Industrial Electronics*, vol. 54, no. 2, pp. 699–706, 2007.

[18] B. Corteville, E. Aertbelien, H. Bruyninckx, J. de Schutter, and H. van Brussel, "Human-inspired robot assistant for fast point-to-point movements," in *IEEE International Conf. on Robotics and Automation*, Roma, Italy, 2007, pp. 3639–3644.

[19] P. Evrard, E. Gribovskaya, S. Calinon, A. Billard, and A. Kheddar, "Teaching physical collaborative tasks: Object-lifting case study with a humanoid," in *9th IEEE-RAS International Conference on Humanoid Robots*, 2009, pp. 399–404.

[20] K. B. Reed and M. A. Peshkin, "Physical collaboration of human-human and human-robot teams," *IEEE Transactions on Haptics*, vol. 1, p. 108120, 2008.

[21] Z. Wang, A. Peer, and M. Buss, "An HMM approach to realistic haptic human-robot interaction," in *Proceedings of World Haptics 2009, the Third Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, Salt Lake City, USA, 2009, pp. 374–379.

[22] J. Hölldampf, A. Peer, and M. Buss, "Virtual partner for a haptic interaction task," in *Human Centered Robot Systems, Cognitive Systems Monographs*, H. Ritter, G. Sagerer, R. Dillmann, and M. Buss, Eds. Springer, 2009, pp. 183–191.

[23] M. Rahman, R. Ikeura, and K. Mizutani, "Cooperation characteristics of two humans in moving an object," *Machine Intelligence & Robotic Control*, vol. 4, pp. 43–48, 2002.

[24] K. B. Reed, M. J. Peshkin, M.and Hartmann, J. Patton, P. M. Vishton, and M. Grabowecky, "Haptic cooperation between people, and between people and machines," in *Proceedings of the 2006 IEEE/RSJ Conference on Intelligent Robots and Systems*, Beijing, China, 2006.

[25] S. Miossec and A. Kheddar, "Human motion in cooperative tasks: Moving object case study," in *Proceedings of the 2008 IEEE International Conference on Robotics and Biomimetics*, 2008.

[26] D. Feth, R. Groten, A. Peer, S. Hirche, and M. Buss, "Performance related energy exchange in haptic human-human interaction in a shared virtual object manipulation task," in *Third Joint EuroHaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, 2009.

[27] R. Groten, D. Feth, H. Goshy, A. Peer, D. A. Kenny, and M. Buss, "Experimental analysis of dominance in haptic collaboration," in

The 18th International Symposium on Robot and Human Interactive Communication, 2009.

[28] R. Groten, D. Feth, R. Klatzky, A. Peer, and M. Buss, "Efficiency analysis in a collaborative task with reciprocal haptic feedback," in *The 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009.

[29] Y. Kunii and H. Hashimoto, "Tele-handshake using handshake device," in *Proceedings of the 21st IEEE International Conference on Industrial Electronics, Control, and Instrumentation*, vol. 1, USA, 1995, pp. 179–182.

[30] N. Pollard and V. Zordan, "Physically based grasping control from example," in *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, USA, 2005, pp. 311–318.

[31] T. Sato, M. Hashimoto, and M. Tsukahara, "Synchronization based control using online design of dynamics and its application to human-robot interaction," in *Proceedings of the IEEE International Conference on Robotics and Biomimetics*, 2007, pp. 652–657.

[32] Y. Yamato, M. Jindai, and T. Watanabe, "Development of a shake-motion leading model for human-robot handshaking," in *Proceedings of the SICE Annual Conference 2008*, Japan, 2008, pp. 502–507.

[33] Z. Wang, J. Hoelldampf, and M. Buss, "Design and performance of a haptic data acquisition glove," in *Proceedings of the 10th Annual International Workshop on Presence*, Spain, 2007, pp. 349–357.

[34] M. Ueberle, N. Mock, and M. Buss, "Vishard10, a novel hyper-redundant haptic interface," in *Proceedings of HAPTICS '04, 12th International Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, 2004, pp. 58–65.

[35] J. Hölldampf, A. Peer, and M. Buss, "Virtual Dancing Partner," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010, submitted.

[36] S. Brown, M. J. Martinez, and L. M. Parsons, "The neural basis of human dance," *Cerebral Cortex*, vol. 16, no. 8, pp. 1157–1167, 2006.

[37] S. Gentry and R. Murray-Smith, "Haptic dancing: human performance at haptic decoding with a vocabulary," in *Proc. IEEE International Conference on Systems, Man and Cybernetics*, vol. 4, 5–8 Oct. 2003, pp. 3432–3437.

[38] Y. Hirata, T. Hayashi, T. Takeda, K. Kosuge, and Z. Wang, "Step estimation method for dance partner robot "ms dancer" using neural network," in *IEEE International Conference on Robotics and Biomimetics*, 2005, pp. 523–528.

[39] T. Takeda, Y. Hirata, and K. Kosuge, "Dance step estimation method based on hmm for dance partner robot," *IEEE Transactions on Industrial Electronics*, vol. 54, no. 2, pp. 699–706, April 2007.

[40] ——, "Dance partner robot cooperative motion generation with adjustable length of dance step stride based on physical interaction," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2007, pp. 3258–3263.

[41] U. Unterhinninghofen, T. Schauß, and M. Buss, "Control of a mobile haptic interface," in *Proc. IEEE International Conference on Robotics and Automation ICRA 2008*, 2008, pp. 2085–2090.

[42] A. Peer and M. Buss, "A New Admittance Type Haptic Interface for Bimanual Manipulations," *IEEE/ASME Transactions on Mechatronics*, vol. 13, no. 4, pp. 416–428, 2008.

[43] M. Okada, K. Tatani, and Y. Nakamura, "Polynomial design of the nonlinear dynamics for the brain-like information processing of whole body motion," in *Proc. IEEE International Conference on Robotics and Automation ICRA '02*, vol. 2, 11–15 May 2002, pp. 1410–1415.

Proceedings of the IEEE ICRA 2010
Workshop on Multimodal Human-Robot Interfaces
Anchorage, Alaska, May, 2010

Modular Haptic Device for Bimanual Virtual Manipulation

Ignacio Galiana, Manuel Ferre, Jorge Barrio, Pablo García-Robledo, Raúl Wirz
Universidad Politécnica de Madrid, Centro de Automática y Robótica UPM-CSIC
Jose Gutierrez Abascal 2, 28006 Madrid, Spain.
Email: (ignacio.galiana,m.ferre,jordi.barrio,p.grobledo, r.wirz)@upm.es
Telephone: (34) 91 336 30 61, Fax: (34) 91 336 30 10

***Abstract*—In this article, the mechanical and electronic design of a Multifinger haptic interface is described. This interface can be used for bimanual manipulation of virtual scenarios. Due to the complexity of the system, we have decided to design a modular device. The basic element is the haptic interface for one finger, and both the electronic and the mechanical design are independent for each module. A distributed architecture has been developed so as to be able to simulate virtual manipulation scenarios by using more than one haptic devices for bimanual or collaborative tasks. The designed haptic interfaces is called MasterFinger-2 (MF-2). The user inserts his thumb and index fingers to manipulate virtual objects. An application calculates the force exerted to the objects, and these forces are reflected to the user by the haptic device. Some examples of bimanual manipulation of virtual scenarios are shown in this article.**

***Index Terms*— Bimanual, haptic, multifinger, virtual manipulation, advanced manipulation.**

I. Introduction

Haptic devices provide the user with force and tactile information during manipulation or exploration of virtual environments [1]. Devices that feedback tactile and force sensation to the user, have been used in teleoperation, design of virtual reality environments, educational training and so on[2].

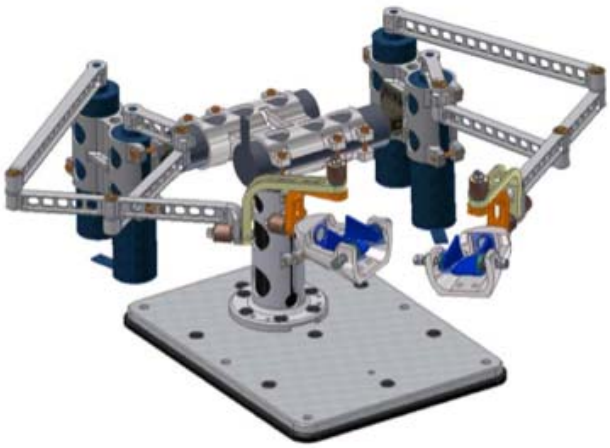


Fig.1. MasterFinger-2 is made up of two modules. Thumb and index fingers are inserted into a thimble respectively.

A great advancement has been done in the manipulation of virtual objects with only one contact point that simulates palpation or exploration of a virtual object surface. However, at least two contact points per hand are required in advanced manipulation tasks for grasping and properly handling objects [3]. Relevant examples of this advanced manipulation can be found in telerobotics [4] and surgical applications [5][6][7].

In this paper, a haptic device interface called MasterFinger-2 (MF-2) is presented. A setup for bimanual and cooperative tasks is also described. The setup consists of two MF-2 devices that allow users to manipulate the virtual environment by using his index and thumb fingers of both hands. The resulting workspace of this system is explained. It is also described the distributed architecture that has been designed for haptic scenarios development and some examples of bimanual virtual manipulation are described.

II. Haptic Device Description

The designed haptic interface can be managed by the user by inserting his index and thumb fingers in two adjustable thimbles that have been designed for that purpose, more details about this thimble can be found at [10]. These thimbles are connected to a mechanical structure with seven actuators, three actuators per finger plus and additional actuator that allows rotating all the mechanical structure on a vertical axis. The mechanical design has been conceived with the purpose of facilitating object manipulation, in particular, for grasping objects. This mechanism is based on a modular configuration in which each finger has its own mechanical structure and electronic components. Figure 1 shows the haptic device developed which is called MasterFinger-2 [8][9].

A. Mechanical Design

The two-finger haptic device has 7 actuators and 13 Degree of Freedom (DoF) in total. Each finger has its own mechanical structure with 6 DoF, the firsts 3 of them actuated and the last 3 only measured. This configuration allows any position and orientation for the fingers into the device workspace. The three actuators are located close to the base device in a serial-parallel configuration. It allows reflecting

forces in any direction with a small inertia. Actuators are linked to a five-bar mechanical structure which is connected to a gimble with 3 rotational DoF. Finally, the last gimble rotation axis is linked to the thimble.

Thimble position is calculated from the encoders included in the actuators, orientation is obtained from three absolute encoders, which are placed in the gimbal rotational axis. The three rotational axis of the gimbal intersect at the user's fingertip. This geometrical configuration avoids torque reflection, meaning that only forces are reflected to the user's finger. The thimble and the gimbal are shown in Fig. 1.

The thimble has been designed so that it can be adjusted to any user finger by adjusting some screws. Each thimble incorporates four contact sensors. These contact sensors are used to estimate the force exerted by the user during the virtual object manipulation. This information contains some inaccuracy since contact sensors only detect normal forces. Tangential forces are estimated by contact sensors located in the sizes of the fingers, details about the contact sensor configuration can be found at [10].

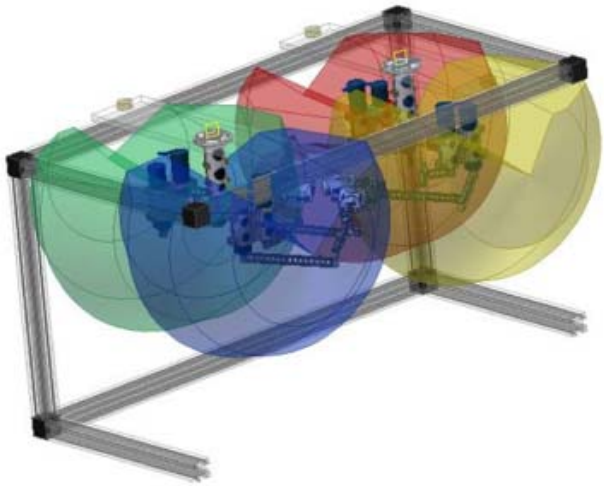


Fig.2. Bimanual configuration and resulting workspace.

B. Controller

MasterFinger-2 has a modular and scalable design. The hardware needed to control one of these devices is based on three different electronic boards:

– Acquisition system board which serves as an interface between the controller and the mechanical parts and sensors. This board not only acts as a bridge between the encoders and the control board but also translates the analog signals into digital data.

- Power electronic board to feed the motors and measure the current.

– A Virtex-5 FPGA (ML-505 Board) that has the low level control of the system programmed on the PowerPC.

The signals and measurements used for controlling the system are mainly the position of the fingertips and the current of the motors. Every motor has an encoder in order to determine its angular position. The end position and orientation of each finger can then be calculated. Current of the motors is low in order to control direction and the amount of force exerted over the fingers.

This controller also computes gravity pre-compensation and includes an antiwindup subsystem as a safety measurement for human users.

The PowerPC runs at the low level controller under a VxWorks real time operating system to assure a constant frequency rate. Only kinematics and Jacobian calculus have been moved to the Scenario Server due to high computational cost.

C. Distributed Architecture for Haptic Scenarios Development

MasterFinger-2 is based on a distributed control architecture this architecture allows the user to use as many MasterFingers and Graphic servers as he might need to interact with virtual objects. The designed architecture is described in Fig.3. It consists of four different components: the haptic interface (MF-2), the control module of the haptic device, the scenario server, and the graphic server.

The mechanical structure and the control module have been described in the previous section, we will now briefly describe the other two components of the designed architecture: The Scenario Server and the Graphic Server.

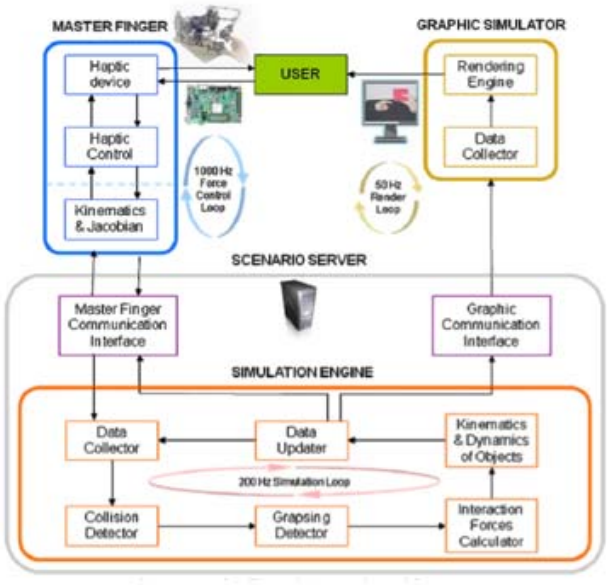


Fig.3. Distributed architecture scheme for haptic scenarios development .

C.1 Scenario Server

The Scenario Server is responsible for the interaction between every object in the virtual scenario and the virtual

user's hand.

The Scenario Server integrates all the data given by the devices and provides the information needed to the Graphic Simulators and to the devices. Different modules compose this server: an interface with the MasterFinger, an interface with the Graphic Communication Server and the Simulation Engine.

Fingertips have been modeled as spheres inside the collision detector module. Diameter of these spheres is 2cm. This length represents the average size of the distal phalanx in a real finger. Spheres are used due to its simplicity to be computed for collision processes. The Simulation Engine also has one module that evaluates if the user is grasping an object with one or two hands or if he is just colliding with it.

C.2 Graphic Simulator

The Graphic Simulator is used to show the virtual scenario; this simulator receives the position and orientation of all the objects in the scenario and shows them on a screen. The Graphic Simulator that we designed is an OpenGL application running on Windows operative system that needs a XML document with all the objects listed. The object's list includes information related to object size, initial position and orientation, its color, the number of segment for its representation and the object transparency rate. When running it receives position data of the objects and of the user (The hand model representation for the MF-2 only needs the position and orientation of the index and thumb finger MasterFinger-2 is mainly used for grasping applications) in the scenario at a frequency rate of 50 Hz; this rate is enough so as to see objects move fluently.

The simulator can be running on different computers at once; this is an important functionality for collaborative scenarios where more than one user needs to see the virtual scenario from his remote location.

The graphic simulator has two modes of operation. One of them uses a wired view of the objects so the user can see through them in the scenario and the fingers are represented as two points that represent the fingertip. The second one uses a solid representation of the objects and a hand model. This last mode of view is much more realistic but less useful when performing difficult tasks because solid objects occult part of the scenario implying that the user has to change the point of view all the time.

D. Communications between Different subsystems of the Distributed Architecture

Communications between different subsystems is based on UDP protocol so that users can add as many devices as needed only by attaching them to an Ethernet port. All the information exchanged by the devices is filtered at MAC level so that the Scenario Server is more robust when using an internet connection. However, common robotics protocols do not provide complete solutions for teleoperation through Internet, there are many applications that require the "Haptic Loupe" to be in the same segment of a LAN and requires low time delay. A novel BTP protocol [11] has been designed to improve performance of bilateral flow tasks for real-time robot teleoperation, it can be easily integrated into sophisticated control algorithms making systems more reliable. BTP is an end-to-end congestion protocol whose main objective is to minimize the round trip tie (RTT) while maximizing the transmission frequency. To achieve this, it performs a network congestion control by means of avoiding congestion signals (Timeouts and packet losses).

E. Examples of virtual manipulation

As was stated before, at least two points of contact per hand are needed to carry out object manipulation and grasping. Some experiments where the user interacted with virtual scenarios were developed with the bimanual mechanical disposition shown in Fig.4.a.

1) Bimanual Box Manipulation

Design of haptic interfaces for precise bimanual manipulation should take into account how weight simulations is implemented when manipulation switches between one and two hands. The importance of this is apparent in tasks where the user requires to apply vertical forces to penetrate a surface with a tool or splice a fragile object using one and/or two hands.Accurate perception of simulated weight should allow the user to execute the task with high precision [12].

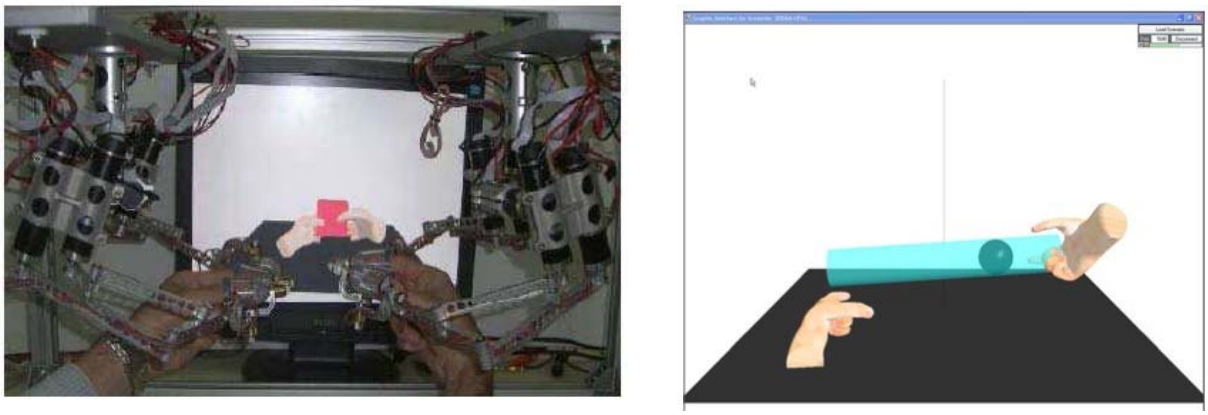
As a first approach to the bimanual problem, a weight discrimination scenario was developped. The scenario consists of a box that can be lifted using one or both hands by a user as shown in Fig.10.a. The goal of this task is to lift one box with one hand, and then lift again another box using both hands (with a different weight) and decide wether it felt heavier or lighter. Results of this experiment were that similar weight discrimination performance between unimanual and bimanual lifting can be observed with real and virtual weights generated by MF-2. The bimanually lifted virtual weights tended to feel lighter than unimanually lifted weights. However, the effect was not as prominent as that observed using real weights; more details of this experiment can be found at [12].

2) Bimanual or Cooperative Manipulation of a Cylindrical Object.

In this experiment the scenario consists on a cylinder that has a sphere that can roll inside as shown in Fig10.b and Fig.2. The goal of this experiment is trying to equilibrate the sphere in the middle of the cylinder. To achieve this goal, there are two possible scenarios: the first one is carried out by just one person by using both hands and the second one is performed by two people (each user using his preferable hand).

The interest of the cooperative scenario is by means of knowing what the other user (which can be placed in a different room) is intending to do and try to coordinate

movements with no more communication between them than just forces and visual information to achive a common goal.



a) b)

Fig. 4. Examples of advanced virtual manipulation: bimanual grasping of a box (a) left and right hands are used by the user; and cooperative manipulation of a cylinder (b) where two users are using their right hand.

During this experiment it was seen that MasterFinger-2 interface can be used for a wide range of cooperative and bimanual tasks due to its considerably large workspace.

III. Conclusion

Most of the haptic devices in market provide the user with just a contact point to do virtual manipulation tasks. The designed haptic device provides the user with two points of contact per hand, this permits a wider variety of manipulating object's tasks and grasping simulations.

In this article, a modular design in which each finger has its own mechanical structrure, its own processing signals hardware and controllers. All the information asociated to each finger is transmitted via Ethernet to an application called Scenario Server that calculates the interaction forces while the user is interacting with the virtual environment.

Simulation of virtual manipulation scenarios requires real time operative systems that guarantee the stablished execution in processes like kinematic calculus, collisions detection that should be calculated at a frequency rate close to 200 Hz.

The combination of all the described hardware and software permits bimanual grasping of a virtual object and other kind of advanced manipulation tasks.

Acknowledgment

This work has been partially funded by the European Commission under the project IMMERSENCE (FP6-IST-027141), and the Spanish *Ministerio de Ciencia e Innovación* under the project TEMAR (DPI2009-12283).

References

[1] Srinivasian, M.A., Haptic Interfaces in Virtual Reality: Scientific and Technical Challenges, EDS: N.I. Durlach and A.S. Mavor, Report of the Committee on Virtual Reality Research and Development, National Research Council, National Academy Press, 1995.

[2] Haruhisa Kawasaki, Jun Takai, Yuji Tanaka, Charfeddine Mrad, and Tetsuya Mouri. "Control of Multi-Fingered Haptic Interface Opposite to Human Hand", Proceedings of the 2003 IEEE/RSJ Conference on Intelligent Robots and Systems. Las Vegas, Nevada. October 2003.

[3] Matsumoto,Y. ; Katsura, S. ; Ohnishi,K. "Dexterous Manipulation in Constrainted bilateral Teleoperation Using Controlled Supporting Point". Transactions on Industrial Electronics, VOL. 54, NO. 2, pp. 1113-1121. APRIL 2007.

[4] A.Peer, M.Buss, "A New Admittance-Type Haptic Interface for Bimanual Manipulations" in IEEE/ASME Transactions on Mechatronics, vol.13, No.4. pp. 416-428, 2008.

[5] K.J. Waldron, K. Tollon, "Mechanical Characterization of the Inmersion Corp. Haptic, Bimanual, Surgical Simulation Interface" in 8th International Symposium on Experimental Robotics (ISER02), vol5, pp. 106-112. 2003.

[6] J. Rosen, B. Hannaford, M. P. MacFarlane, and M.N. Sinanan, "Force controlled and teleoperated endoscopic grasper for minimally invasive surgery-Experimental performance evaluation" IEEE Trans. Biomed. Eng., vol 46, no. 10, pp. 1212-1221. Oct. 1999.

[7] L.Sun, F. Van Meer, Y. Bailly, C.K. Yeung, "Design and Development of a Da Vinci Surgical System Simulator" in Proceedings of the 2007 IEEE International Conference on Mechatronics and Automation, pp. 1050-1055, 2007.

[8] Garcia-Robledo, P.; Ortego, J.; Galiana, I.; Ferre,M.; Aracil,R. "MultiFinger haptic interface for bimanual manipulation of virtual objects" in IEEE International Workshop on Haptic Audio Visual Environments and Games. Have 2009. Pisa, Italy.

[9] M. Monroy, P. Garcia-Robledo,M. Ferre, J.Barrio, J. Ortego, "Advanced virtual manipulation based on modular haptic devices" in 9th International IFAC Symposium on Robot Control, in press.

[10] 10Mary Monroy, Manuel Ferre, Jorge Barrio, Victor Eslava and Ignacio Galiana. "Sensorized thimble for haptics applications". Proceedings of the 2009 IEEE International Conference on Mechatronics. Málaga,Spain. April 2009.

[11] 11Wirz, R. ; Marin, R. ; Ferre, M. ; Barrio, J. ; Claver, J.M. ; Ortego, J. Bidirectional Transport Protocol for Teleoperated Robots. In IEEE Transactions on Industrial Electronics, VOL. 53, NO. 6, December 2006.

[12] Christos Giachritsis, Jorge Barrio, Manuel Ferre, Alan Wing, and Javier Ortego."Evaluation of Weight Perception During Unimanual and

Bimanual Manipulation of Virtual Objects”Third Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. Salt Lake City,UT,March 18-2009.

Proceedings of the IEEE ICRA 2010
Workshop on Multimodal Human-Robot Interfaces
Anchorage, Alaska, May, 2010

Graphical and Semantic Interaction by means of Gestures

Alícia Casals *Senior Member, IEEE*, Josep Amat *Member, IEEE*, Jordi Campos

***Abstract*—Human-Machine interfaces constitute a key factor to guarantee the effective use of technological equipment. In the field of image guided surgery and surgical robots, the availability of an adequate interaction means determines the suitability or not of a given technological aid. This work focuses on the problems surgeons find in planning and executing a robot assisted intervention. Analyzing the potential of computer graphics, together with the surgeons needs during, first, the planning and later on the development of a surgical intervention, the specifications and the implementation of an interface is described. In the design of this interface, main attention has been put on the gesture and attention capabilities surgeons can devote to the interface.**

I. Introduction

Graphical interfaces are very common in different fields of Human-Machine interaction, being the medical field an area in which they have a significant relevance. Graphical information can complement the semantic contents of control orders, especially when dealing with robotic systems. Free hand interfaces are of special interest in the surgical field since surgeons have their hands busy with instruments and the gloves they wear constitute an additional inconvenience to deal with classical interfaces. Most interfaces require physical contact and mechanical interaction through a master device. They, together with those interfaces that use gloves for gesture recognition are unacceptable in the surgical environment.

Free hand operation is usually related to interaction systems relying on oral interfaces. However, although voice communication can be very useful in some application areas, it can result inefficient in others due to its limited semantics, when restricted to a short vocabulary or reduced set of commands. These limitations affect even more in robot assisted orthopedic surgery, where stronger interaction requirements appear. This kind of surgery can take advantage of systems operating with virtual fixtures (VF), which constitute computer tools that alleviate surgeons from the pressure they suffer in some interventions and to facilitate their work. VF are useful, either to protect critical areas or to assist surgeons in trajectory guidance. VF have to be defined by the surgeon *a priori*, or even on-line, if the interface offers this facility.

Manuscript received March 5, 2010. This work was supported in part by the Spanish Research Agency under the Program Plan Nacional I+D, under the projects: DPI 2007- 63762 and DPI2008-06857-C02-01

A. Casals is with the Institute for Bioengineering of Catalonia, and Professor of the Universitat Politècnica de Catalunya, Barcelona Tech. Barcelona, Spain (e-mail: acasals@ibec.pcb.ub.es).

J. Amat is with the Universitat Politècnica de Catalunya, Barcelona Tech. Barcelona, Spain (e-mail: josep.amat@upc.edu).

J. Campos is with the Universitat Politècnica de Catalunya, Barcelona Tech. Barcelona, Spain (e-mail: jordi.campos@upc.edu).

Referring to oral communication, although its use in MIS has proved to have some limitations due to the sensitiveness to the speaker's emotions, some efforts have been done to make oral recognition independent of stress, fatigue or other causes of voice modulation. In [1] some robustness is achieved using a high dimensional acoustic feature space. Nevertheless, referring to surgical robots, only simple operations have been reported, as camera guidance in laparoscopic surgery, using oral orders. In [2] a study is done on pros and cons of current interfaces and their suitability in surgery, considering not only the need, or not, of the surgeon hands to interact, but also the attention the interface requires from the user. Among existing interface techniques, gestures constitute an alternative means to communicate with a machine in a natural and intuitive way. A multimodal system is described in [3] combining oral local communication to guide the camera with simple qualitative orders; a mobile interface that offers a Graphic User Interface (GUI) that can even be controlled remotely; and finally a remote interface conceived for an experienced surgeon that can assist the local physician. The possibility of focusing the attention to speech only when required is tackled in [4], based on eye contact and contextual speech recognition. In [5] the free hand concept is tackled considering the needs to be solved using gestures: gesture detection, action generation and the association between gestures and actions.

To cope with the limitations of the above mentioned interfaces, and focusing on the needs encountered in robot assisted surgery, mainly orthopedics, a gesture based interface system has been developed. This interface combines two modalities: one semantic, based on the use of menus, and a second one graphic, which complements the former by improving its capabilities and efficiency by reducing the time required to define the actions and orders to be given.

The developed system is oriented to the control of tools such as: grippers, scissors, holders, catheters… which are controlled by electromechanical or robotic systems, providing a means to operate with more ergonomic control capabilities. Thus, the main goal of the interface is reducing tiredness and stress to surgeons and at the same time increase patient's safety.

II. Typology of Commands with Graphic Support

Some simple orders can be given operating "free-hand", by means of oral communication (voice recognition), but they can become useless to perform guidance tasks in which the orders necessary for the control of the system have a much wider pass band than that achievable using oral commands. Furthermore, oral communication would become very noisy under these conditions. Other devices

can complement oral information, but for complex orders no good enough solutions are commonly found.

Alternatively, gesture based orders are very efficient and intuitive to the user. Gesture, as voice, also constitutes a natural language. Gesture commands can either rely on a mechanical support such as joysticks, 6D devices or so, or operate "hands free". In this case, either inertial sensors or those relying on magnetic or optical sensors are considered [6, 7]. However, in spite of their good performances they have some drawbacks as they demand sometimes uncomfortable postures from the user, mainly if the task to be carried out takes place around complex geometries.

In order to deal with such limitations, a graphical complement allows the user to rotate, move, approach or move away the visualized working space, or generate and edit surfaces within this workspace.

The former actions are oriented to have the best point of view available at any moment, while the latter is aimed to generate virtual fixtures that behave as protection surfaces or as guiding surfaces to help in instrument guidance.

The most common 3D graphical interfaces which allow the user to move the controlled element in X,Y and Z directions and to rotate them over these three axes, fig. 1, are used without much difficulties by users of CAD systems. However, they become extremely tough to users less accustomed to such computer systems. Moreover, the enormous versatility of these interfaces usually brings such users to desperation, when he or she sees something rotating undesirably, or when due to bad luck an object rotates erroneously, or if by mistake a rotation is produced in two consecutive axes. In these situations, in most interfaces, there is no way to return to the "stable" initial position without an enormous effort that takes much time and requires high attention. All these factors produce unacceptable situations that surgeons cannot admit during an intervention.

Fig. 1 Usual 3D commands in CAD environments

III. 3D Graphic Commandment System with Imposed Constraints to Facilitate their Operability

After a common work together, engineers and surgeons, a graphical interface has been established which is highly simplified compared to those used by CAD designers. The interface provides an easy interaction with a much reduced menu than those commonly used. This menu, based on flip flop operating icons, type activated/deactivated, consists of two independent blocks, which are independently activated.

They are:

- Visualization block (change of the observation point)
- Commands block, for the generation and edition of constraints

These constraints are oriented to define or modify the limitations of the working space during the robotized actuation of the surgical instruments, with two main goals; to guarantee patient safety and at the same time reduce the stress that surgeons suffer when operating close to critical zones. Fig. 2 shows the interface designed in common agreement with the medical team. It has been conceived to be operated by means of gestures.

This gesture based interaction relies on a stereoscopic vision system developed within the research team, [6] that locates the operator hand in front of the screen, emulating a mouse. Besides its location, the system models the hand in such a way that it can determine two different states: open-hand or closed-hand. These two states emulate the mouse click, as well as the double click, with the close-open-close hand, at speeds that can be adjusted to each user.

The developed system allows the user moving any element in space and visualizes the sticking point (when closing the hand). This action is indicated with a yellow point (picking point). If the picked element has available a constraint in one degree of freedom (for instance a rotation over a point) these constraints are visualized with a blue point. Therefore, with the yellow points-blue points code, the selected object can be moved, either in the free space or leaning on a point or an edge.

To improve the efficiency of the vision system (gesture based mouse) and be able to lock the cursor movement at the user's will, a pedal is used that activates the vision when pressed and deactivates it when released.

In what refers to computer facilities, the interface offers the user the possibility of generating any kind of constraints. Thus, the software tools allow the user to generate lines (straight or curves) that can be used as trajectories, the user can convert or generate surfaces (flat or curved) and by aggregation of surfaces the user can configure three-dimensional limits, which bound the working space that contains the robot end-effector.

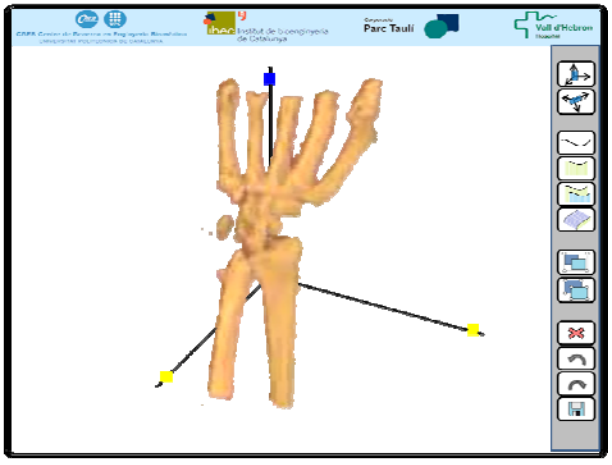


Fig. 2 Menu used in the graphic interface

IV. Reduced Set of Commands for Interactive Surface Generation

The accessible working spaces that a surgeon generates are composed of elemental surfaces that can be further composed together so as to generate the desired volume. Later on, they can be modified at the surgeon's will. This task, operating in normal conditions in 3D CAD environments, takes some learning time; a too long and tough process for non specialized users. The large range of possibilities these systems offer carry with them the need of spending great learning and training efforts that most of health professionals are not prone to undertake.

For these reasons, the interface designed has been conceived to be very simple and intuitive, allowing the user to generate all the characteristic restrictions usually needed in orthopedic surgery. The criterion that has been followed to generate such constrained spaces is to provide tools to configure them from surfaces, being these surfaces generated from generatrix lines.

Under these premises, if it is necessary to define a simple trajectory corresponding to a cut, the surgeon has to trace a trajectory over a 3D model (MR, …) fig. 3a. This trajectory, using n reference points will generate n-1 segments, calculated trigonometrically from a space dimension R^2, the composition of which will give us a spline line.

$$[X]=[A]+t\cdot[B]$$

Once the trajectory is described, the surgeon can convert that line to a plan π, when dealing with straight lines or a surface otherwise, Fig. 3b. That conversion is provided by equally increasing the same coordinate value of the n points of an existing line (duplicating it), and composing the m segments between them.

$$\pi \Rightarrow ax+by+cz=d$$

The generation of a surface that contains a line constitutes an undetermined problem, and consequently, it is necessary to define additional conditions, such as determining a passing point in space or defining an orientation from which a growing process starts. In order to simplify this operation, the developed system presupposes that this surface has to be formed perpendicular to the visualization plan. This simplification is possible since it has been observed that surgeons place the vision plan in a position frontal to the cut to be done. This assumption is also extended to the election of the semi plan defined by the generatrix line, since usually the direction of the cut to be performed is from outside to inside the screen and any observation has indicated that it happens in the opposite direction. Anyway, this automatic assignation of an orientation in the 3D space is equivalent to an assignation by default, since the user can correct this initial orientation by "picking" the generated plan by any point external to the generatrix line (then appearing a yellow point as indicator). This point can also be moved in space.

These corrections are equivalent to a reconfiguration of this plan in the 3D space according to its new orientation, using a single-axis rotational matrix and its movement configuration and composition.

$$(X' \quad Y')=[X \quad Y]\begin{bmatrix}\cos a & \sin a\\ -\sin a & \cos a\end{bmatrix}$$

Therefore, this semi surface generation is assisted by the designed interactive interface, since it minimizes the number of actions to perform, which are tedious and even unacceptable by qualified personal staff, which do not belong to the world of geometry and informatics. Once the semi surface has been traced, the surgeon can convert this constraint to a bilateral constrain, by clicking the duplication function and positioning it, if necessary the new surface, which is parallel to the previous one, fig. 3c.

Usually, this visual protector of the cut to be done, is also complemented with a new surface, the depth limit. A final click action over the "grouping" function, forms an "allowed" work space.

In this way, and through a successive aggregation of limiting surfaces, it is possible to configure a restricted work space that impedes the access of the surgical instrument to the protected parts, when a robot is controlled in comanipulation mode.

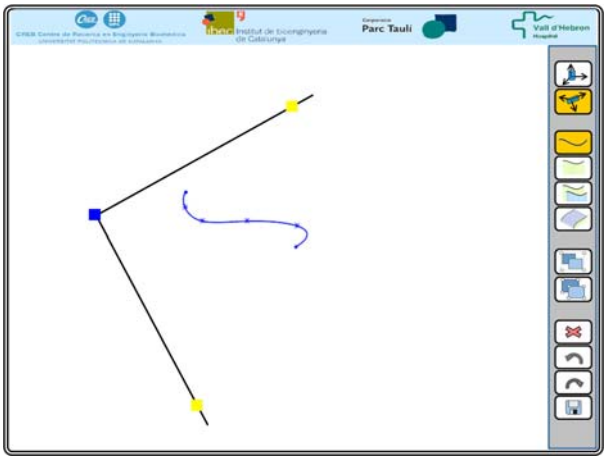


Fig. 3 a) Frontal view of a line with some passing points

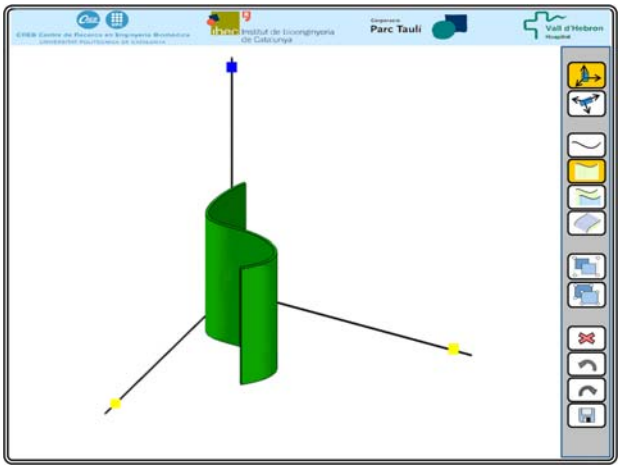


Fig. 3 b) View of the image with a line that has grown down to the X,Y axis to become a surface.

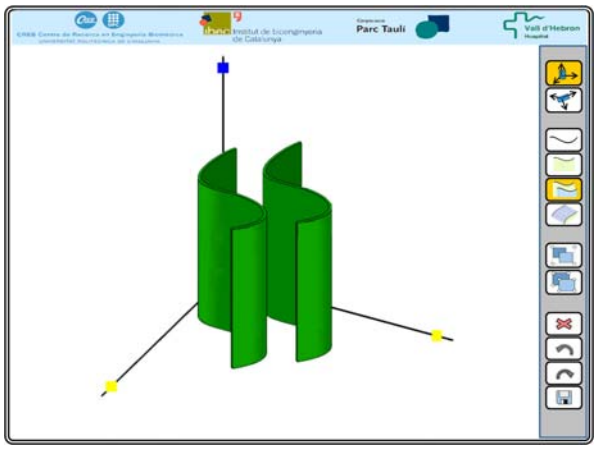


Fig. 3 c) Bilateral constrain generated by plane duplication

V. Application to the Generation of Virtual Fixtures in Orthopedic Surgery

This interface has been evaluated in the implementation of different operations in the laboratory, using animal skulls, Fig. 4, having achieved very precise cuts, Fig. 5.

Fig. 4 Test bed for experimentation

Fig. 5 Detail of a cut operation

In maxillofacial surgery, bone reconstruction in oncology is a good candidate intervention type that can benefit from this interactive assistance. The procedure consists in extracting a piece of the tibia bone to graft it in the affected jaw. In this case, the formation of a cut line can be done in a much precise way. Using the same defined pattern, both in the tibia where the bone tissues are extracted and in the jaw where they are grafted, the extracted tissue can be adjusted to the shape and size of the volume of the affected jaw, and thus, the graft fits better.

These VF can be, as well, of utility as a safety protection over critical elements as can be the facial nerve. For such applications the benefit of VF is mainly the reduction of stress that the surgeon suffers when approaching such elements, and indirectly, gaining in patient's safety and efficiency.

VI. Evaluation of Acceptability

The evaluation of the interface by different professional staff has provided some inputs to estimate its operability and acceptability. A significant parameter evaluated has been the time spent in the definition of a cutting restriction defined over a plan, as shown in fig. 6, programmed in a previous planning phase. Two issues are evaluated; first, the difficulties each operator founds in converting the described plan or surface into a bilateral space, that is, a corridor comprised between two surfaces, and second, in defining a bounding surface that limits the depth of the cut to be performed. With the commands available in the interface, shown in fig. 1, managed from gestures, the operation times obtained from different users are shown in fig. 7. It can be clearly observed that the learning factor, for task implementation (not the commands), does not represent a dramatic time reduction. Thus, it can be seen as an index of the simplicity in its use.

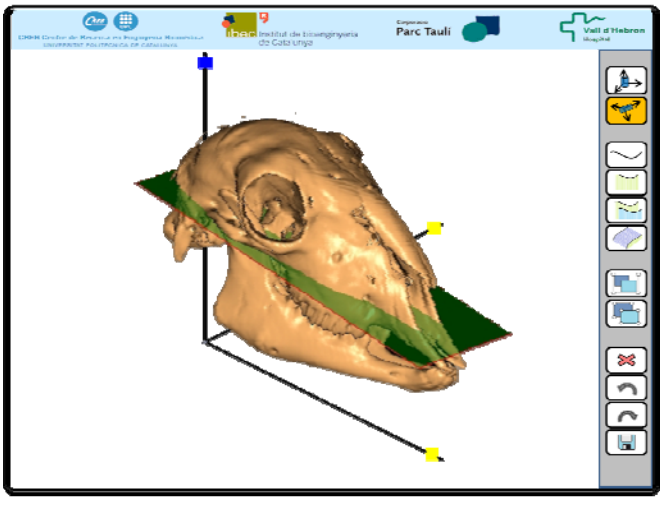


Fig. 6 Anatomic image with a restriction plan inserted

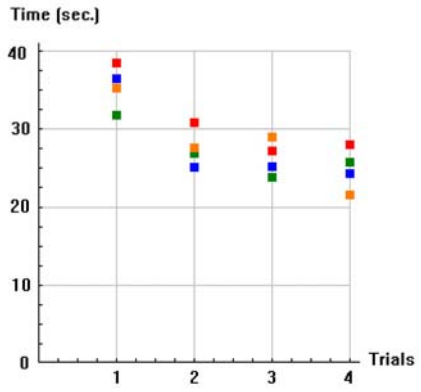


Fig. 7. Time spent in successive test trials by four different users

VII. Conclusion

From the conviction that the interface is a critical part of robot and computer assisted systems, this work has focused on the needs of a particular kind of robotic or teleoperated (or comanipulated) systems. The interface has considered a limited number of actions to be performed with the hand and has designed an interface that facilitates an ergonomic operation. Placing and orienting adequately the elements to be visualized and the movements to be carried out by the surgeon it is possible to avoid too large turns or rotations, thus improving ergonomics. In what refers to the required user's attention, and thanks to the reduced number of remaining actions, the identification of the minimum number of icons and their type, oriented to this application field, the interaction becomes friendly, intuitive and easy to learn.

Acknowledgment

The authors want to thank the contribution of the surgeons Dr. Enric Laporte, Head of the Centre of Experimental Surgery, of the Consorci Sanitari del Parc Taulí and Dr. Juan A. Hueto, Head of the Maxilofacial Surgery Dep. of the Hospital de la Vall d'Hebron. Both of them have contributed to this work with their advice and continuous evaluation of robotic interfaces.

References

[1] B. Schuller, et al., "Emotion Sensitive Speech Control for Human-Robot Interaction in Minimal Invasive Surgery", *17th IEEE International Symposium on Robot and Human Interactive Communication*, 2008 pp. 453-458

[2] A. Casals, M. Frigola, J. Amat, E. Laporte, "Quasi Hands Free Interaction with a Robot for Online Task Correction", *Springer Tracts in Advanced Robotics: Experimental Robotics IX*, 2006

[3] J. Fernández-Lozano, et al, "Human-Machine Interface Evaluation in a Computer Assisted Surgical System", *IEEE Int. Conference on Robotics adn Automation*, 2004

[4] G. J Lepinski and R. Vertegaal, "Using Eye Contact and Contextual Speech Recognition for Hands-Free Surgical Charting", *Int. Conference on Pervasive Computing Technologies in Health Care,* 2008

[5] Y. Mohammad, T.Nishida and S. Okada, "Unsupervised Simultaneous Learning of Gestures, Actions and their Associations for Human-Robot Interaction", *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009

[6] M. Frigola, A. Rodriguez, J. Amat, and A. Casals, "Computer Vision Body Modeling for Gesture Based Teleoperation", *Springer Tracts in Advanced Robotics: Advances in Telerobotics*, Vol, 31, 2007

[7] N. Berci and P. Szolgay, "Vision Based Human-Machine Interface via Hand Gestures", *18th European Conf. on Circuit Theory and Design*, 2007

Proceedings of the IEEE ICRA 2010
Workshop on Multimodal Human-Robot Interfaces
Anchorage, Alaska, May, 2010

Haptic and Ocular Human-Robot Interface

Andrés Úbeda, Eduardo Iáñez, Carlos Pérez and José M. Azorín

***Abstract*— These paper provides a brief description of a multimodal interface for robot controlling. The man-machine multimodal interface is based on the fusion of visual control, through electrooculography, and haptic control, using a desktop device operated with the hand. The paper describes several control strategies to move a Fanuc robot with the multimodal interface and some applications are proposed in order to test the improvement in human-robot communication.**

I. INTRODUCTION

Recent technological advances are opening new human-machine interaction ways. They allow an intuitive interaction and remove any physical or technical limitation of the user, providing a more accessible machine control. In this sense, multimodal interfaces let people improve their ways of communication with external devices such as computers or robots.

Multimodality consists of using different ways of human communication: voice, eyes, gesture or movement, in order to perform a more natural man-machine communication. A multimodal interface is the device which mixes these different ways of communication to achieve the objectives of multimodality. A typical example of multimodality is a personal computer, where the use of the mouse and the keyboard is combined. Nevertheless, there are many other ways of integrating man-machine communication devices, for example using pens, sounds, gestures, tactile screens, voice recognition or even eye recognition [1]-[5].

This paper describes a multimodal human-robot interface that uses haptic and ocular information. The electrooculography technique is used to detect the eyes motion. A desktop haptic device is used to provide force feedback. Both devices can be used to control the robot: a Fanuc LR Mate 200iB. This kind of interfaces are usually used separately in other works related to man-machine communication.

Haptic interfaces are based on recognizing objects through touch by transmitting forces, vibrations or movements to the user. These devices increase the interaction between the user and the machine by perceiving virtual objects or receiving feedback forces from the user actions. This technology is used on many fields such as surgery training or spare time activities like videogames augmenting the feelings perceived by the user [4], [6].

This work has been supported by the Ministerio de Ciencia e Innovacion of the Spanish Government through project DPI2008-06875-C03-03.

Andrés Úbeda, Eduardo Iáñez, Carlos Pérez and José M. Azorín are with Virtual Reality and Robotics Lab, Industrial Systems of Engineering, University Miguel Hernández, Elche, Spain, aubeda@umh.es, eianez@umh.es, carlos.perez@umh.es, jm.azorin@umh.es

Ocular interfaces consists of obtaining the eye gaze or direction in order to perform all sort of tasks. These devices are usually easy to use for people with some disability, but also improve the natural communication with the environment. This devices can be used in many man-machine interfaces [7]-[10].

The remainder of these paper is organized as follows. In section II the multimodal interface is described. Section III explains different strategies for multimodal controlling. The applications proposed are shown in section IV. Finally, section V contains some conclusions.

II. HUMAN-ROBOT MULTIMODAL INTERFACE

The architecture of the multimodal interface is shown in Fig.1. It consists of an ocular interface and a haptic interface. Both interfaces are connected to a computer where the control strategies are implemented. The multimodal interface is used to control a 6 DOF arm robot (right) which is able to perform a wide range of applications.

A software application to integrate both interfaces has been developed in C++ language. It has been separated in two threads, see Fig.2. One thread (thread 1) performs the communication with the acquisition card that captures the electrooculographic signals. The other thread (thread 2) controls the haptic and graphic contexts, executes the control strategies and controls the robot movement.

The electrooculographic signal is captured at a frequency of 50Hz per channel. That signal is analyzed and processed in blocks of 50 samples per channel, so the analysis and processing frequency will be 1Hz. The haptic device owns an independent loop with a frequency of 1000Hz. This loop controls the force feedback in the device.

A. Ocular Interface

The ocular interface uses electrooculography (EOG) to detect the movement of the eyes. Next, the ocular interface is briefly described. See [9], [10] for more information.

EOG is based on the fact that the eye acts as an electrical dipole between the positive potential of the cornea and the negative potential of the retina. Thus, in normal conditions, the retina has a bioelectrical negative potential related to the cornea. For this reason, the rotations of the ocular globe cause changes in the direction of the vector corresponding to this electric dipole, Fig.3. The recording of these changes requires placing some small electrodes on the skin around the eyes, Fig.3. The EOG value varies from 50 to 3500 V with a frequency range of about DC-100Hz between the cornea and the Bruch membrane located at the rear of the

Fig. 1. Multimodal Interface (left): Haptic Interface (Phantom Omni of Sensable), Control Device (PC), Graphic Interface and EOG Interface (Nicolet Viking IV D). Robot environment (right): Fanuc LR Mate 200iB

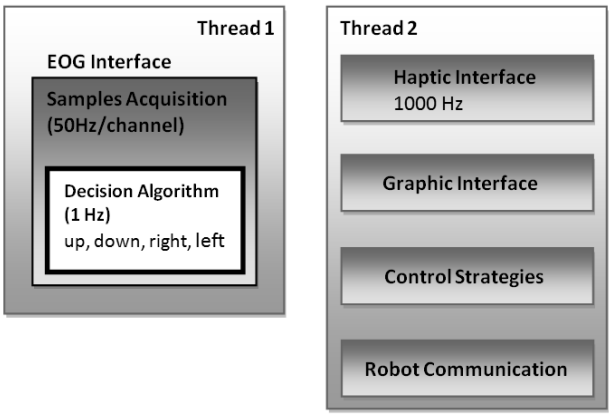


Fig. 2. Multimodal interface threads. EOG interface is managed by the thread 1; haptic interface, graphic interface, control strategies and robot communication are managed by the thread 2.

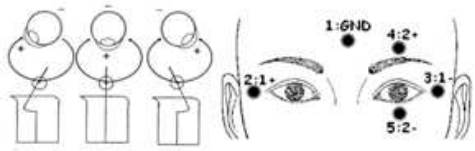


Fig. 3. Ocular dipole (right part). Position of the electrodes on the face (left part). 1: ground electrode, 2 and 3 for horizontal movement, 4 and 5 for vertical.

eye. Its behavior is practically linear for gaze angles of ±50°horizontal and ±30°vertical.

The ocular interface uses the Nicolet Viking IV D device and a National Instruments (NI) card to obtain the EOG signals with a sample frequency of 50 Hz in a computer.

In order to generate the device command, the human must perform a fast movement of the eyes in the desired direction and he/she must later return his/her eyes to the center position. The algorithm to obtain the device command from the eye movement is shown in Fig.4. The phases of this algorithm are the next:

1) The EOG signals are acquired from the NI card.
2) The moving average is calculated to eliminate the noise and to obtain a cleaner signal.
3) The derivative is calculated to detect the change in the eyes direction. If the person looks toward one direction, the signal abruptly changes. This fast change followed by a slow fall is detectable by the derivative, generating a high value (positive and negative) in the moment that happens.
4) A threshold is used to distinguish the detection of the eyes movement from noise and/or the signal obtained when there is not eyes movement. This threshold can be different for horizontal and vertical movements and it depends of the human.
5) The maximums and minimums are searched in order to know the direction of the eyes movement. In the control strategy developed, the max/min/max or min/max/min sequences are searched.
6) Finally, the direction of the gaze is obtained.

Electrodes impedances must be lower than 50kΩ to assure the quality of the signals. In the tests, the impedances were about 30kΩ for positive electrodes and about 20kΩ for negative ones.

The decision algorithm is executed for each channel. The result will be -1, 0 or 1 that corresponds to left, nothing or right (down, nothing or up) respectively.

The interface analyzes and processes the eye movements, while the haptic device button is pressed. Thus the user can move the eyes freely when he/she wants to generate a command and the interface comfort is increased.

The accuracy of the algorithm depends on the election of the thresholds of the signal. As it has been mentioned, there are two channels: vertical and horizontal. For both channels, two threshold values are chosen (max and min) which eliminate all the signal between them and prevent the algorithm from getting wrong positives. It has been proved

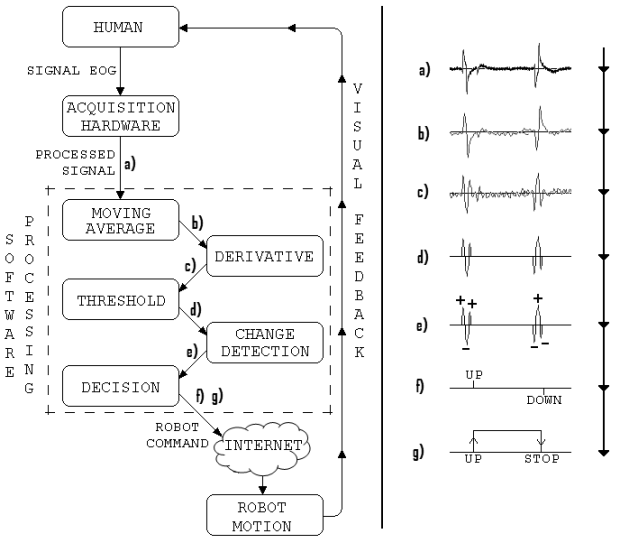


Fig. 4. EOG-processing algorithm to obtain the robot arm command from the eye movement (left) and evolution of the EOG signal during the processing algorithm (right).

that the value of these numbers is different depending on the user. Therefore, a training must be performed before working with the interface for every user. This training includes a series of predefined movements of the eye that allow obtaining the proper threshold for each channel.

The series of training movements consists of looking at each direction twice in the same order and waiting two seconds between each movement. If the algorithm does not work properly, in other words, the direction is not detected, all the information is saved in an spreadsheet where the data can be represented and the right threshold can be easily changed.

B. Haptic Interface

The haptic interface is the Phantom Omni Device from SensAble, that has 6 degrees of freedom and force feedback in 3 degrees of freedom. This means the user will be able of moving the robot arm in any three dimensional direction. The device can be easily controlled by holding the pen and moving it to the desired direction.

The OpenHaptics toolkit from SensAble has been used for the software development. The control strategies will get the position and orientation from the haptic device and then they will control the force feedback.

III. CONTROL STRATEGIES

Ten control strategies have been developed. They can be grouped into three control philosophies: Non Simultaneous Control (4 strategies developed), Shared Control (3 strategies developed) and Sensorial Fusion (3 strategies developed). Next, the control strategies are explained.

A. Non Simultaneous Control Strategies

In Non Simultaneous Control Strategies each interface controls a robot feature (or robot environment feature) non-simultaneously.

- **Non Simultaneous Control Strategy 1:**
  In this control strategy, first the user gets closer to an object using the EOG interface and then, the user touches that object using the haptic interface.
  The distance between the robot end effector and objects is evaluated in each iteration. The strategy estimates if they are near or far according to preset parameters. If the robot end effector is far from objects, it is controlled by the EOG interface. In this case the haptic interface will automatically be moved like the robot end effector. When the robot end effector is near from an object, it is controlled by the haptic interface.
- **Non Simultaneous Control Strategy 2:**
  In this control strategy, the robot is controlled by the haptic interface and the EOG interface controls the environment camera.
  The user can move, rotate and zoom the environment camera. That changes affect to the workspace of the haptic device, e.g., if the zoom tool of the environment camera is used to get closer to an object, the user will see bigger the object and the graphic scene and, in the same way, the workspace of the haptic scene will be reduced.
- **Non Simultaneous Control Strategy 3:**
  In this control strategy, the 2 DOF tasks are controlled by the EOG interface and the 6 DOF tasks are controlled by the haptic interface. Thus the user can realize tasks in a plane without fixing the others DOF of the haptic device.
  In this strategy the user selects the interface that will control the robot end effector. In the same way like the first control strategy, when the robot is controlled by the EOG interface, the haptic interface will automatically be moved like the robot end effector.
  When the EOG interface is selected, the user can choose the plane where the robot end effector will be moved between a list of preset planes (XY, YZ, XZ, ...)
- **Non Simultaneous Control Strategy 4:**
  In this control strategy, first the user defines the EOG movement plane using the haptic interface. Then, the EOG interface controls the movement of the robot end effector in that plane.
  Like the previous strategies, when the robot is controlled by the EOG interface, the haptic interface will automatically be moved like the robot end effector.

B. Shared Control Strategies

In the Shared Control Strategies each interface controls simultaneously a different robot feature. The following strategies have been developed:

- **Shared Control Strategy 1:**
  In this control strategy the haptic interface controls the velocity of the robot end effector while the EOG interface controls the direction of the robot end effector. The further the haptic device end effector is from the center of its workspace (horizontally), the higher speed

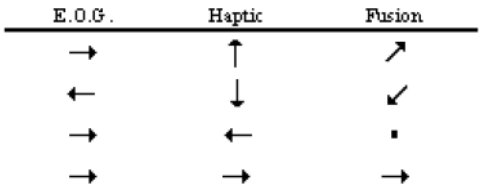


Fig. 5. Sensorial Fusion Strategy 1. Right & Up: Diagonal movement. Left & Down: Diagonal movement. Right & Left: Nothing. Right & Right: Right movement

is gotten. If the user moves right, speed is scaled ten times.

- **Shared Control Strategy 2:**
  In this control strategy the EOG interface controls the movement of the robot end effector in a plane while the haptic device receives the force feedback of that movements. In other words, haptic interface does not generate control actions.
  The haptic device will automatically be moved like the robot end effector. If the robot end effector touches an object, it will stop. At this moment, the haptic interface will be blocked.
  In this state, the EOG interface can only carry out control actions that move the robot end effector away from the object surface.
- **Shared Control Strategy 3:**
  In this control strategy the EOG interface controls the movement of the robot end effector in a plane. This plane is defined by the haptic interface.
  The haptic device will automatically be moved like the robot end effector. The plane is defined with the last 3 DOF of the haptic device.

C. Sensorial Fusion Strategies

Finally, in the Sensorial Fusion Strategies the information of both interfaces is merged providing a unique control action. Three sensorial fusion control strategies have been developed:

- **Sensorial Fusion Strategy 1:**
  Both interfaces control the movement of the robot end effector in a plane.
  The user has four control actions for both interfaces: up, down, left an right.
  The haptic interface is going to behave like the EOG interface. The user must perform a movement with the haptic device from an initial position to a position on left, right, up or down and then return to the initial position. This action will be the new control action of the haptic device.
  Thus the user can send two control actions at the same time, e.g. if the EOG interface send up and the haptic interface send right simultaneously, the result will be a diagonal movement, see Fig. 5.
- **Sensorial Fusion Strategy 2:**

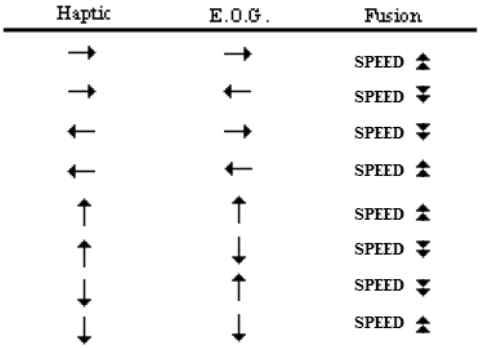


Fig. 6. Sensorial Fusion Strategy 3. The haptic device controls de direction of the movement. EOG controls speed on each direction

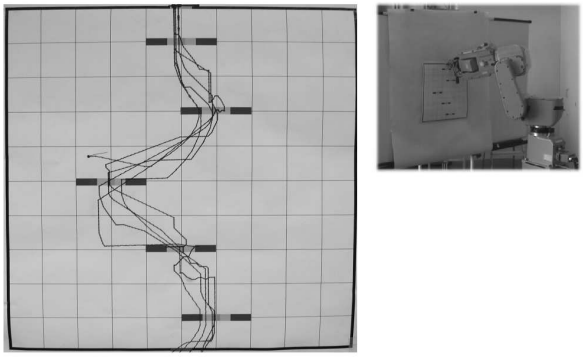


Fig. 7. Example of circuit panel (left). Robot environment (right)

  Both interfaces control the movement of the robot end effector in a plane.
  The EOG interface controls the direction of the movement, while the haptic device controls the movement itself. This way, the robot end effector is moved by the haptic interface, but along the direction given by the EOG interface, which acts as a supervisor.
- **Sensorial Fusion Strategy 3:**
  Both interfaces control the movement of the robot end effector in a plane.
  The haptic device controls the movement, while the EOG interface controls the speed in each direction. The movement can be totally controlled by the haptic interface, but user is able to freely speed up or down the end effector of the robot. The combination of each control action is described in Fig. 6.

IV. APPLICATIONS

Once the different strategies are defined, several applications can be proposed. Some applications aimed at motor disabled users have been tested in previous works, see [11], [12]. But there are also applications that improve the classical human-robot interaction by combining these different interfaces. As an example, a panel has been designed to test

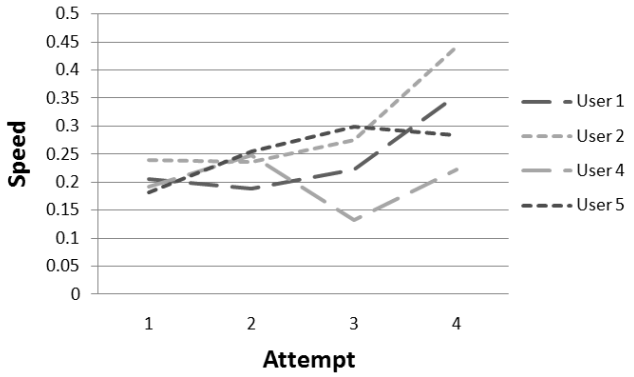


Fig. 8. Speed evolution in each attempt

the speed of the combination of the haptic and ocular device (see Fig. 7).

In this application, the user controls the end effector of the robot, which holds a marker to draw a trajectory in a previously designed panel. The objective is to pass through each goal (a total of five) and get the maximum score in as little time as possible. EOG controls the direction and the haptic device controls speed.

A protocol was designed to evaluate properly the results. 5 users tested the application and each one drew up to 4 paths. For each one, the speed was measured. This speed can be calculated as the score divided by the time spent on each path. Each goal has a value of 10 points if the marker passes through the center. If not, the score decreases from 10 to 0 in the left and right end side.

As it can be seen in Fig. 8, speed increases in each attempt. Notice that User 3 was out of the study because he/she was extremely tired and the results were not conclusive. For the rest of the users, it can be seen that the system has a period of training and the users get used to it throughout time.

V. CONCLUSIONS

This paper has described a multimodal interface based on electrooculography and haptics. Several control strategies have been developed in order to exploit the advantages of multimodality for human-robot interaction. The haptic and ocular multimodality may be useful to improve this interaction. For this purpose, one application with the robot has been shown. In this test, it is proved that the communication with external devices, such as robots, enhances substantially with the combination of both devices and some training.

REFERENCES

[1] M. Kolsch, R. Bane, T. Hollerer and M. Turk, "Multimodal Interaction with a Wearable Augmented Reality System," *IEEE Computer Graphics and Applications*, vol. 26, no. 3, pp. 62-71, May-Jun. 2006.

[2] A. Corradini and P. R. Cohen, "Multimodal speech-gesture interface for handfree painting on avirtual paper using partial recurrent neural networks as gesturerecognizer," *Intenational Joint Conference On Neural Networks*, Honolulu, USA, vol. 3, pp. 2293-2298, 2002

[3] A. Timofeev, A. Nechaev, I. Gulenko, V. Andreev, S. Chernakova and M. Litvinov, "Multimodal man-machine interface and virtual reality for assistive medical systems," *International Journal: Information, Theories & Applications*, vol.14, no. 2, pp. 133-138, 2007

[4] D. Wang, Y. Zhang and Z. Sun, "Multi-modal Virtual Reality Dental Training System with Integrated Haptic-Visual-Audio Display," *Robotic Welding, Intelligence and Automation*, vol. 362, pp. 453-462, 2007

[5] M. Beitler, Z. Kazi, M. Salganicoff, R. Foulds, S. Chen and D. Chester, "Multimodal User Supervised Interface and Intelligent Control (MUSIIC)," *AAAI Technical Report FS-95-05*, pp. 5-11, 1995

[6] K. Park, B. Bae and T. Koo, "A haptic device for PC video game application,", *Mechatronics*, vol. 14, pp. 227-235, Mar. 2004

[7] A. E. Kaufman, A. Bandopadhay and B. D. Shaviv, "An eye tracking computer user interface," *IEEE Symposium on Research Frontiers*, pp. 120-121, 1993

[8] Y. Chen and W. S. Newman, "A human-robot interface based on electrooculography," *IEEE International Conference on Robotics and Automation*, vol. 1, pp. 243-248, Apr-May. 2004

[9] E. Iáñez, J. M. Azorín, R. Morales and E. Férnandez, "Electrooculography-based Human Interface for Robot Controlling," *Proceedings of the 13th Annual Conference of the International Functional Electrical Stimulation Society (IFESS)*, Freiburg, Germany, vol. 53, sup. 1, pp. 305-307, Sep. 2008

[10] E. Iáñez, J. M. Azorín, E. Férnandez and J. M. Sabater, "Control Strategies for Human-Robot Interaction using EOG," *Proceedings Book 40th International Symposium on Robotics (ISR 2009)*, Barcelona, Spain, pp. 183-187, Mar. 2009

[11] J. P. Pinar, J. M. Azorín, A. Úbeda and Eduardo Iáñez, "Human-Robot Multimodal Interaction using Haptics and Electrooculography," *Eurohaptics 2010*, Submitted

[12] Andrés Úbeda, Eduardo Iáñez and José M. Azorín, "Haptic and Ocular-based Human-Robot Multimodal Interface for Disabled and Non-Disabled People," *IEEE Transactions on Systems, Man, and Cybernetics* , Submitted

Proceedings of the IEEE ICRA 2010
Workshop on Multimodal Human-Robot Interfaces
Anchorage, Alaska, May, 2010

Robotic Wheelchair Controlled by a Multimodal Interface

Teodiano F. Bastos, André Ferreira, Wanderley C. Celeste, Daniel C. Cavalieri,
Mário Sarcinelli-Filho, Celso De La Cruz, Carlos Soria, Elisa Pérez, Fernando Auat

***Abstract*—This work presents the development of a robotic wheelchair which offers the user (adult or children) with flexibility of either supervised or fully automatic unsupervised navigation. It offers the user with multiple command options to provide support for people with different levels of disabilities. User may command the chair based on eye blinks recorded using electromyographic signals (EMG), eye movements using videooculogram, head movements using accelerometer or video camera, and using electroencephalogram (EEG signals). The wheelchair also is equipped with a communication system that allows the user to communicate with people in the close proximity. The user is provided with an easy to use and flexible Graphical User Interface (GUI) on a Personal Digital Assistant (PDA) that allows the users to communicate the commands, needs or emotions.**

I. Introduction

DISABLED people often lack mobility and subsequently face several hardships. Powered wheelchairs help these patients overcome some these limitations and provide for them with some level of mobility and freedom. While extremely useful, the wheelchairs require the user to have intact manipulation ability to use a joystick to command the wheelchair. Unfortunately, number of disabled people do not have the manipulation ability to control a joystick or similar mechanical device and are unable to use such a wheelchair [1].

If such low mobility is due to Amyotrophic Lateral Sclerosis (ALS), the patient would have lost communication capabilities. In this case, the patient is locked in his own body, with low quality of life. Frustration, anxiety and depression are common for these patients [2].

Robotic systems can improve the personal autonomy of disabled people through the development of some devices that allow displacement and communication of those patients. Robotic wheelchair can be used for mobility by people who are unable to manipulate the controllers. These can be equipped with sensors to detect obstacles, and such a wheelchair can follow a predefined path or allow the user a free path. Such systems can obtain commands based on biological signals generated by the wheelchair user, such as eye blinks, eye movements, head movements and brain signals (this last one is necessary for patient with ALS [3]).

If the patient is able to maintain good head posture and controlled head movements, an accelerometer attached to his head or a video camera installed onboard the wheelchair can capture the head movements and use it to command the wheelchair or for the purpose of communication. However if the patient does not have head movements but can control his eye blinks (Electromyographic signal – EMG), these signals can be used.

People with ALS who are unable to blink eyes may use eye movements (captured with a video camera – video-oculography). Another option is the use of Electroencephalographic signals (EEG). Generally EEG is acquired non-invasively [4-6], though embedded EEG based systems have recently been proposed.

In the past, different groups around the world have worked on providing a modality to help the patients. Unfortunately, each of the solutions are stand-alone and not integrated together to provide one common platform that can provide the user with a choice of modalities for commanding the robotic wheelchair. While each of the modalities are useful, without such integration, each of these are able to support only a small group of patients and do not provide the user with the desired flexibility and reliability.

This paper reports the development of a robotic wheelchair that integrates the different modalities and provides the user with flexibility to choose from a number of command options. The robotic wheelchair can be commanded using eye blinks, eye movements, head movements and brain signals. The wheelchair has onboard a Human Machine Interface (HMI) integration system that provides the intelligence to the chair. The system identifies the different commands and communicates these to a PDA that identifies the user movement command. This interface also provides means for the user to communicate with other people.

The wheelchair also has an autonomous mode, where the user does not have to provide a series of commands. In this mode, the user identifies the destination and the wheelchair identifies the best and safe path to reach the destination while avoiding the obstacles.

This work is structured as follows: Section 2 presents the Human-Machine Interface. Section 3 presents details of the acquisition system and processing system. Section 4 deals with experiments carried out and, finally, Section 5 presents conclusions of this work.

II. Human-Machine Interface

Fig. 1 shows the structure of the general purpose Human-Machine Interface that has been developed and installed

Manuscript received March 4, 2010. This work was supported in part by FACITEC/Brazil (Vitoria City Hall Foundation under Grant 03/2009).

Teodiano F. Bastos, André Ferreira, Wanderley C. Celeste, Daniel C. Cavalieri, Mário Sarcinelli-Filho and Celso De La Cruz are with Departamento de Engenharia Elétrica, Universidade Federal do Espírito Santo, Vitoria, 29075-910 Brazil (+5527 40092077; fax: +5527 40092644; e-mail: teodiano@ele.ufes.br).

Carlos Soria, Elisa Pérez and Fernando Auat are with Instituto de Automática, Universidad Nacional de San Juan, San Juan, Argentina (e-mail: csoria@inaut.unsj.edu.ar).

onboard the robotic wheelchair. This consists of an acquisition system that includes amplification, filtering, digitization, recording and processing of different biological signals. The signal is recorded on the onboard computer which in real-time classifies these signals and interfaces with the PDA to generate command signals to the wheelchair, feedback signals for the user (Fig. 1), and performs automatic symbols scanning with the PDA. These symbols are associated to movements (arrows or places) or communication (characters and iconic symbols representing needs or feelings). After a valid command has been identified, a movement command is sent to the wheelchair or an acoustic signal is generated for the audio speakers onboard the wheelchair.

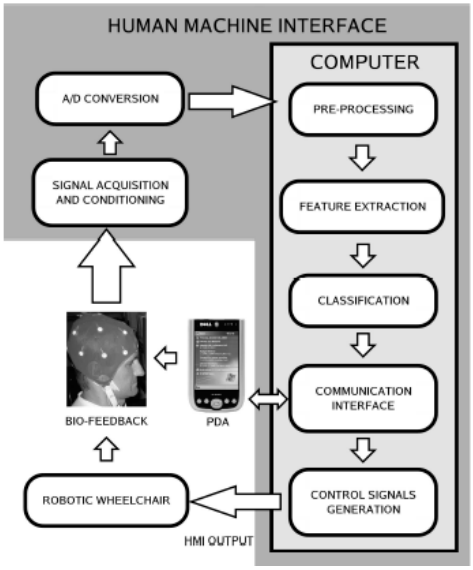


Fig. 1. Structure of the Human-Machine Interface developed.

III. ACQUISITION AND PROCESSING SYSTEM

The system being reported in this paper is a non-invasive system. It determine eye blinks based on surface electromyogram (sEMG) of the associated muscles. This requires placement of surface electrodes on the temporal muscles around the eyes. The eye movements are determined based on video data acquired through a small video camera attached to purpose built easy to wear system resembling commonly worn spectacles (videooculography). For obtaining the head movements, two modalities have been provided in this system. An accelerometer attached to the head using a cap is one modality and a video camera installed in front of the wheelchair user is the second modality. EEG signals are acquired using a pair of surface electrodes placed on the visual cortex (occipital region).

A. *Commanding the Robotic Wheelchair by Eye Blinks*

To command the robotic wheelchair using eye blinks, two channel EMG signals are captured through electrodes located on temporal muscles (Fig. 2). One channel is used for the right eye muscle and the other for the left eye muscle. The signals shown in Fig. 2 are the differential signal between both channels, using a reference at the ear lobe.

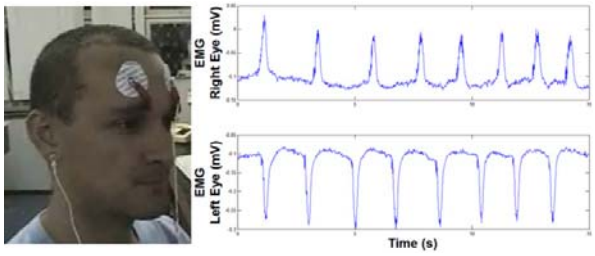


Fig. 2. EMG Acquisition (eye blinks).

To detect the eye blink signal, a simple threshold based algorithm was used. The threshold is determined from the sample data, being 35% of the maximum of the EMG signal. Eye blink is detected with peaks above 35% of the maximum peak because this avoids false detection [8].

The next step in confirming the eye-blink is based on the duration. The algorithm developed is based on the angular variation of each sample point of the data. The tangent to the left and to the right of the signal peak derivate is computed. In case of the tangent value is smaller than an empirically determined threshold value (0.0025 was used for our experiments), this is considered the beginning or the end of the valid signal.

After identifying the eye blinks signals as detailed above, supervised Artificial Neural Network (ANN) was used to recognize the eye blink and ignore the noise. As the first step, the data was downsampled to 20 samples/ second. This was then normalized to improve the speed of convergence of the ANN.

252 test signals were obtained (84 eye blinks of the left eye, 84 of the right eye and 84 randomic noise). Several supervised ANN algorithms were evaluated, and Resilient Backpropagation algorithm, with 4 neurons in the hidden layer and 3 neurons in the output layer was selected based on its performance. With this algorithm, the accuracy for the test data was 99.6% in the cases. Fig. 3 shows the robotic wheelchair commanded by eye blinks.

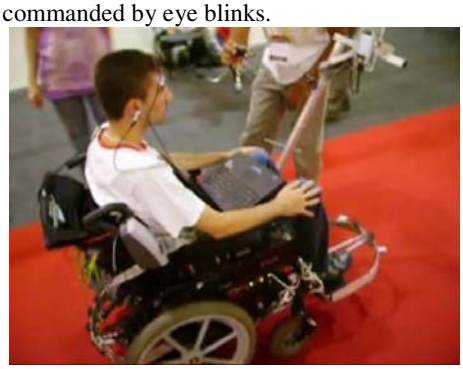


Fig. 3. Robotic wheelchair commanded by eye blinks.

B. *Robotic Wheelchair Commanded by Head Movements*

Two options have been provided for enabling the user to give head movement commands. One option uses an accelerometer type inclination sensor attached to a cap (or

other device attached to the head). The second option is video based and uses a video camera mounted in front of the wheelchair and trained towards the head.

- *Using an accelerometer*

A two axis accelerometer has been incorporated to provide a voltage proportional to the head inclination. This signal is processed by a microcontroller which uses Bluetooth to communicate with the onboard computer. Fig. 4 shows this sensor attached to the purpose built circuit to measure head movements. Moving the head forward, to the right or to the left, commands the robotic wheelchair to go forward, turn to right or turn to the left, respectively. Moving the head to the rear makes the wheelchair stop.

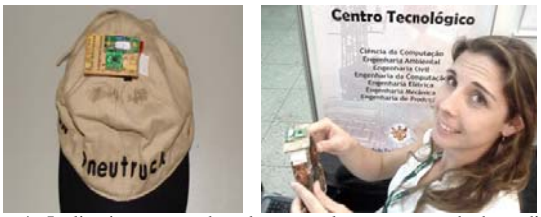


Fig. 4. Inclination sensor based on accelerometer attached to different devices.

The principle behind determining the head inclination angles is based on associated gravitational accelerations. To obtain the head movement, two independent angles need to be determined; α and β angles. α is the forward inclination while β is the angle to the left and to the right Fig. 5 shows the process of obtaining the α angle, which is given by:

$$\alpha = \cos^{-1}(Gy/G)$$

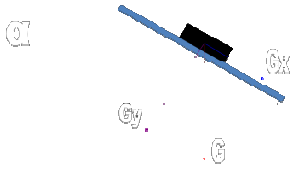


Fig. 5. Obtaining inclination angle α.

Fig. 6 shows the robotic wheelchair commanded by head movements (captured by accelerometer).

Fig. 6. Robotic wheelchair commanded by head movements (captured by accelerometer).

- *Using video camera*

To detect head movements, a standard light weight fixed focus webcam video camera can also be used. It is necessary to install the camera in front of the wheelchair such that it captures images of the user face (Fig. 7). It is necessary for all the processing to take place in real time.

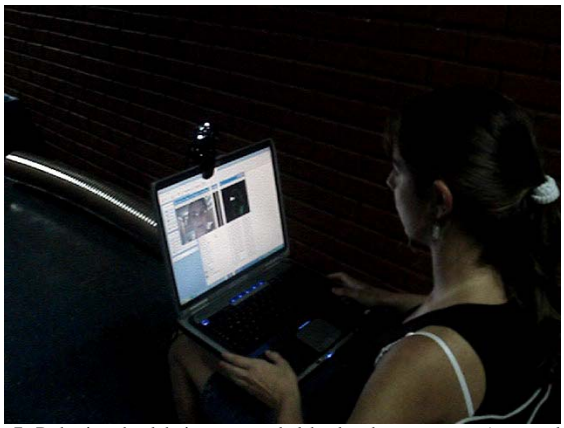


Fig. 7. Robotic wheelchair commanded by head movements (captured by video camera).

The first step in the analysis of the video data is histogram equalization of the RGB video data to improve contrasts and overcome lighting variations. This is transformed to YCbCr space to detect the skin color. The image is segmented to identify the skin using a Cb and Cr threshold obtained from the training data. An elliptical region of interest (ROI) is generated and centered at the first image moment of the segmented image. An operation AND is executed between the ellipsis generated and the negative of the component Y.

The next step is identification of the centroids of the regions associated to both eyes and the mouth. These are filtered using a Kalman filter to improve the position estimate. Three non-collinear points in the camera coordinates define a triangle in the image plane (Fig. 8). Changes in space points, due to head movements, will be projected on the image plane, changing the points in the image.

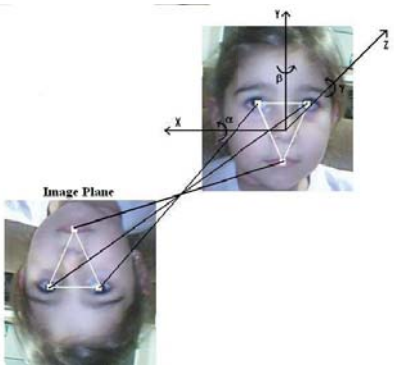


Fig. 8. Facial features.

From the point projections on the image, the different angles of the head movements can be obtained: rotation about axis Z, rotation about axis Y and rotation about axis X, given, respectively, by

$$\gamma = \tan^{-1}\left(\frac{yr - yl}{xr - xl}\right)$$

$$\beta = 2\tan^{-1}\left(\frac{a_1 \pm \sqrt{a_1^2 - f^2\left(a_1^2/a_0^2 - 1\right)}}{f\left(a_1/a_0 + 1\right)}\right)$$

$$\alpha = 2\tan^{-1}\left(\frac{c_1 \pm \sqrt{c_1^2 - f^2\left(c_1^2/c_0^2 - 1\right)}}{f\left(c_1/c_0 + 1\right)}\right)$$

C. *Robotic Wheelchair Commanded by Eye Movements*

To command the robotic wheelchair by eye movements, a webcam attached to a purpose built commonly used spectacle shaped support has been used. Fig. 9 shows the Human-Machine Interface used to track the eye movements.

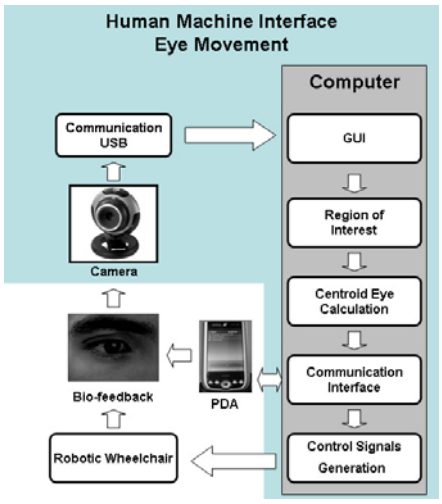


Fig. 9. Human-Machine Interface used to track eye movements.

To obtain the eye movement, the ocular globe has to be first identified. The first step requires identifying a threshold to distinguish the iris from other parts of the face. However, this technique can be influenced by the presence of the eyebrow and eyelash. To overcome this shortcoming, a Hough Circular Random Transform and a Canny filter have been applied to the image. The next step is to define a region of interest around the eye to allow tracking the eye movements. Due to illumination variations, a Kalman filter is used to reduce the error in the calculus of the eye center. This way, a robust system is obtained allowing determining the eye position and tracking it. To select a symbol in the PDA, the wheelchair user must gaze the eye to the symbol desired. For instance, to command the robotic wheelchair to go ahead, the user must gaze his eye to the arrow indicating go ahead movement. Thus, after some seconds, the PDA will send a control signal to the computer onboard the wheelchair in order to start the movement desired. Fig. 10 shows the steps necessaries to detect and track the eye movements, and Fig. 11 shows the system adapted to the wheelchair.

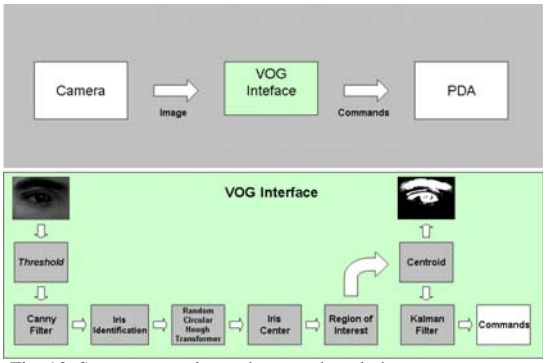


Fig. 10. Steps necessaries to detect and track the eye movements.

Fig. 11. Robotic wheelchair commanded by eye movements.

D. *Robotic Wheelchair Commanded by Brain Signals*

To command the robotic wheelchair by brain signals, it is necessary to put a pair of electrodes on the occipital cortex (visual region), position O_1 and O_2, according to Standard International 10-20 (Fig. 12).

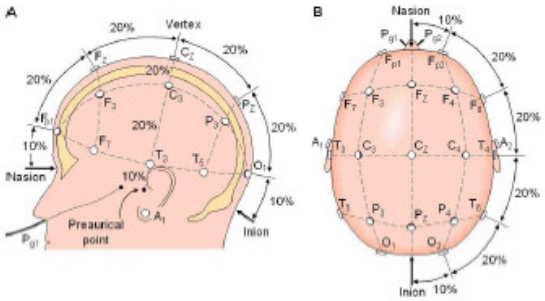


Fig. 12. Standard international 10-20 to locate electrodes.

The brain patterns used are suppression and activation of alpha rhythm, which are related to concentration and visual excitation (which is more intense to open eyes), and visual relaxation (which is more evident to closed eyes) [7].

Fig. 13 shows the electrode location on the occipital cortex and the brain signals generated when the wheelchair user has visual excitation (suppression of alpha rhythm) or visual relaxation (activation of alpha rhythm).

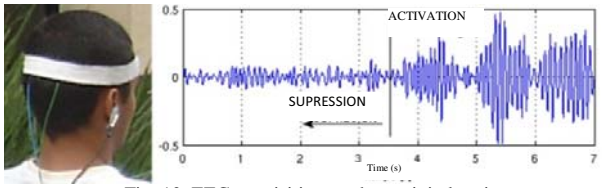


Fig. 13. EEG acquisition on the occipital region.

The signal acquisition system is composed of a conditioning board, where signals are amplified, filtered and digitized. Filtering is necessary to extract the CC level and attenuate the 60 Hz noise and other artifacts (from muscles, heart beats and electrode movements). A bandpass filter from 8 to 13 Hz is used to obtain the alpha rhythm.

To identify the command from the EEG signal, the signal variance (VAR) has been used. This has the advantage of giving a measure of the signal intensity and density while being suitable for near real-time application and is given by

$$s = \frac{1}{N}\sum_{k=1}^{N}(x_k - \mu)^2$$

where N is number of samples of the EEG signal, x_k is the k-th simple of the signal and μ is the simple average, given by

$$\mu = \frac{1}{N}\sum_{k=1}^{N} x_k$$

In Fig. 13 a signal that contains the patterns of activation and suppression is shown. This is generated by the wheelchair user by opening the eyes to generate suppression, and by closing the eyes, to generate activation. It can be observed from this figure that the signal variation is very significant. It can also be observed from Fig. 13 that the signal variance changes significantly.

Signal classification has been performed using a threshold technique. Only if the variance is greater than the higher thresholds, the signal is considered to be a command. If the signal is between the two thresholds, the signal is considered in a dead zone, and if lower than the lower threshold, it is discarded. These two thresholds are determined during training.

Fig. 14 shows the robotic wheelchair commanded by brain signals. Besides the biosignals based human computer interface for commanding the chair, this wheelchair is equipped with multiple sensors to ensure the safety of the user. An encoders determines its location; ultrasonic sensors are for obstacle avoidance and magnetic sensors detect metallic tracks (in case the wheelchair is operating as an auto-guided vehicle). A set of speakers are provided to help the user communicate using pre-recorded voice.

Fig. 14. Robotic wheelchair commanded by brain signals.

Kinematics and dynamic models based control architecture has been used to control the wheelchair movement [10], [11] (Fig. 15). The kinematics controller receives the reference signals associated to movement commands, and generates linear and angular velocities, that are transmitted to the dynamic controller. This controller then generates another pair of linear and angular velocities, which are transmitted to the low level controller (PID controller) onboard the wheelchair. This low level controller is responsible for controlling the linear and angular velocities of the wheelchair.

The dynamic controller was designed based on the nominal wheelchair dynamic, which represents the medium estimate dynamic of the wheelchair.

The kinematics controller manages the changes needed to the wheelchair orientation and its linear and angular velocities. On the other hand, the dynamics controller improves the wheelchair navigation, generating smooth movements.

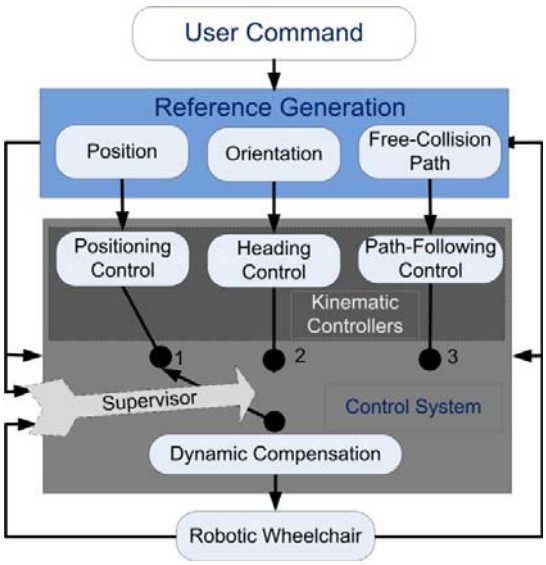


Fig. 15. Control system of the robotic wheelchair.

IV. Auto-Guided Navigation

The robotic wheelchair also provides an auto-guided option to the user for indoor environment. This maybe

appropriate for a highly dependent user, or a user who may desire this option to go to some specific locations. It is also a desired option for indoor environment when navigating the doors may not be easy for some of the users.

For this option, metal tape on the floor is provided to define the navigation path. Magnetic sensors are installed on the wheel chair and these detect the metallic tracks. In the auto-guided option, the pathway for the wheelchair along the metallic strips from the current location to the desired location is determined by the computer. RFID (Radio Frequency Identification) targets are also installed in suitable locations such as the door to monitor the wheelchair for controlling the speed. It also provides acoustic feedback to the user for location awareness. Fig. 16 shows the magnetic sensors and the RFID reader installed onboard the wheelchair.

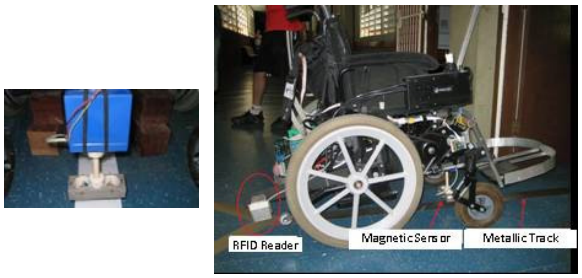


Fig. 16. Details of the magnetic sensor (left), its location on the wheelchair, the RFID reader and the metallic tracks.

V. Navigation Using Laser Sensor

Athough we have used metalic stripes along the way and using magnetic sensors to detect them in order to allow the wheelchair to navigate through narrow ways, we also have used laser sensor for navigation through SLAM (Simultaneous Localization and Mapping Algorithm) [12, 13]. The problem is how to drive the wheelchair to successfully cross a door without using reactive behavior. No map is previously loaded neither. The solution using SLAM allows simultaneously build a map of the environment while the door is detected and the wheelchair robot tries to reach it. Fig. 17 shows the laser sensor onboard the wheelchair, and Fig. 18 shows the general system architecture used in this work.

The SLAM is implemented by an EKF (Extended Kalman Filter) algorithm which estimates both: the wheelchair's position and the environmental features parameters. The door is considered as part of the SLAM system state. Once the door is detected, a path is generated between the wheelchair and the door. The path is then time constrained and an adaptive trajectory controller drives the wheelchair to the door. The controller receives the wheelchair's pose estimation within the environment (Fig. 20).

The door is detected by an adaptive cluster algorithm, based on laser histogram measurements. The parameters that define a door are the coordinates of its middle point at the SLAM reference system. Lines and corners surrounding the door are features used to recognize the door. The laser sensor acquires 181 measurements from 0 to 180º.

The path planning algorithm used is based on variation of the Frontier Points Method. This method finds empty spaces at the limits of the range sensor measurements and directs the motion of the robotic wheelchair to these spaces. The nodes generated by this algorithm are obtained by an angle windowed search of the frontier point associated to the laser range. Fig. 19 shows the determination of the frontier points. A path is generated joining the nodes by a spline. It is dynamically maintained and updated during the wheelchair's driving. This situation helps in avoiding collisions, generating a safety zone along each node of the path. This ensures that the path does not have obstacles.

The SLAM system state is composed by the wheelchair's pose estimation, the parameters of the middle point of the door and the parameters of the corners (concave and convex of the environment) and lines (associated with lines). The algorithm starts when a door is detected.

Fig. 17. Laser sensor installed onboard the robotic wheelchair.

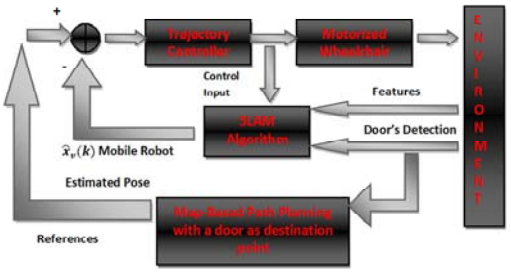


Fig. 18. General architecture system.

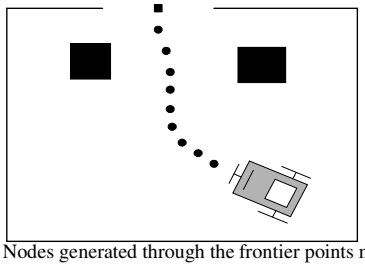


Fig. 19. Nodes generated through the frontier points method.

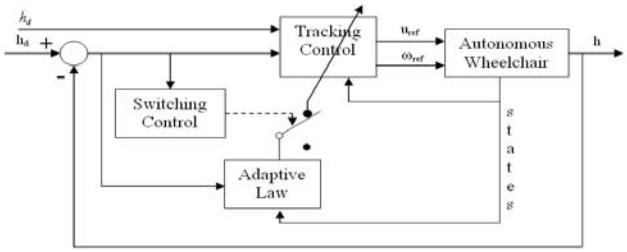


Fig. 20. Adaptive trajectory controller to drive the wheelchair to the door.

VI. COMMUNICATION SYSTEM

The PDA onboard the wheelchair provides the graphical user interface (GUI) with icons for movement commands (arrows or icons of places) and communication symbols (characters and icons expressing needs or feelings), Fig. 17. These are organized in a hierarchical fashion, and scanned serially. The user can make a selection at the temporal location when the desired symbol is scanned. A suitable pre-recorded acoustic emission is produced according to the symbol, word or sentence selected.

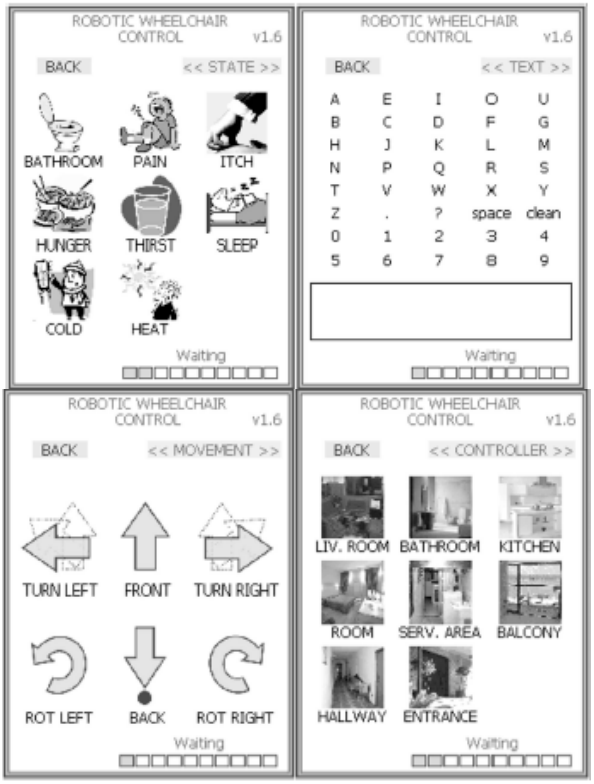


Fig. 17. Different options of communication (symbols representing needs or feelings, or characters) and movement (arrows of movement or symbols representing places) present on the PDA.

VII. CONCLUSIONS

A multi-option robotic wheelchair is presented in this paper. This wheelchair can be commanded by eye blinks, eye movements, head movements and brain signals. It also provides for autonomous control option. The chair has a user friendly GUI using which the wheelchair user can issue movement commands to the wheelchair, or use the onboard communication system to communicate with people in the proximity. The GUI uses easy to recognize icons organized in a hierarchical fashion to help the user express emotions and feelings or select characters to compose words or sentences. A set of pre-recorded acoustic signals and a speaker are provided for this purpose.

The wheelchair can navigate in an autonomous style by taking the user from the current location to the desired location, or in an auto-guided style by following metallic tracks. RFID is used to determine the location. This is also used to provide location awareness to the user.

The wheelchair uses a kinematics and dynamics controller to minimize navigation error and perform smooth movements.

Several experiments were conducted with this robotic wheelchair, with the subjects using the different command options. The wheelchair was evaluated by healthy and severe disabled people (adult and children). The next step is to extend the brain signals option using EEG based motor mapping. This option would allow the user to command the chair with the intent of the movement of the left and right hand. Preliminary experiments have been conducted using Power Spectral Density (PSD) and Adaptive Autoregressive (AAR) parameters as feature inputs to a classifier based on Support Vector Machine (SVM). Results indicate the command identification accuracy to be 98%.

REFERENCES

[1] Cassemiro, C. R. y Arce, C.G., "Visual communication by computer in amyotrophic lateral sclerosis" (in Portuguese), Arquivos Brasileiros de Oftalmologia, 2004, Vol. 67, No. 2, pp. 295.300.

[2] Borges, C.F., "Dependency and die of the "family mother": the helping from family and community to take care of a pacient with amyotrophic lateral sclerosis", (in Portuguese), Psicologia em Estudo, 2003, Vol. 8, pp. 21.29.

[3] Hori, J., Sakano, K. y Saitoh, Y., "Development of communication supporting device controlled by eye movements and voluntary eye blink", Proceedings of the 26th Annual International Conference of the IEEE EMBS, San Francisco, 2004.

[4] Wolpaw, J.R., Birbaumer, N., McFarland D.J., Pfurtscheller, G. y Vaughan, T.M., "Brain computer interfaces for communication and control", Clinical Neurophysiology, 2002, Vol. 113, No. 6, pp. 767.791.

[5] Millán, J.R., Renkens, F. Mouriño, J., y Gerstner,W., "Non-invasive brain-actuated control of a mobile robot", Proceedings of the 18th International Joint Conference on Artificial Intelligence. Acapulco, México, 2003.

[6] Bastos, T.F., Sarcinelli-Filho, M., Ferreira, A., Celeste, W.C, Silva, R.L., Martins, V.R., Cavalieri, D.C., Filgueira, P.S., Arantes, I. B., "Case study: Cognitive control of a robotic wheelchair", Wearable Robots: Biomechatronic Exoskeletons, Ed. Pons, J.L., Wiley, 2008, Ch. 9, Sec. 9.6, pp. 315-319.

[7] Ferreira, A. Celeste, W.C., Cheein, F.A., Bastos, T.F., Sarcinelli-Filho, M. y Carelli, R., "Human-machine interfaces based on EMG and EEG applied to robotic systems", Journal of NeuroEngineering and Rehabilitation, 2008, Vol. 5, pp. 1-15

[8] Cavalieri, D.C., Brandão, A.S., Ferreira, A., Benevides, A.B., Bastos, T.F., Sarcinelli-Filho, M., "Redes neuronales artificiales aplicadas a la detección de parpadeos de ojos", XII Reunión de Trabajo en Procesamiento y Control, Río Gallegos, Argentina, 2007.

[9] A. Ferreira, R.L Silva, W.C. Celeste, T.F. Bastos, y M. Sarcinelli Filho, "Human-Machine Interface Based on EMG and EEG Signals Applied to a Robotic Wheelchair," Journal of Physics. Conference Series, v. 1, 2007, p. 1/012094-8.

[10] F.N. Martins, W.C. Celeste, R. Carelli, M. Sarcinelli Filho, T.F. Bastos, "An Adaptive Dynamic Controller for Autonomous Mobile Robot Trajectory Tracking," Control Engineering Practice, v. 16, p. 1354-1363, 2008.

[11] C. De La Cruz, R. Carelli and T. F. Bastos, "Switching Adaptive Control of Mobile Robots", IEEE International Symposium on Industrial Electronics – ISIE08, Cambridge, UK, 2008.

[12] G. Dissanayake, P. Newman, S. Clark, H.F. Durrant-Whyte and M. Csorba. "A solution to the simultaneous localisation and map building (SLAM) problem." IEEE Trans. Robotics and Automation, 2001.

[13] K. Kouzoubov, D. Austin. "Hybrid Topological/Metric Approach to SLAM". Proc. of the IEEE International Conference on Robotics and Automation. 2004.

Proceedings of the IEEE ICRA 2010
Workshop on Multimodal Human-Robot Interfaces
Anchorage, Alaska, May, 2010

An Ontology-based Multimodal Communication System for Human-Robot Interaction in Socially Assistive Domains

Ross Mead, Jerry B. Weinberg, *Member, IEEE*, and Maja J Matarić, *Senior Member, IEEE*

***Abstract*—An agent's ability to perceive the world and its physical capabilities impact its communicative modalities. Moreover, the sensing and processing (i.e., interpretation) abilities of the user determine the channels and dynamics of communication that should be utilized by the robot to transmit and receive information. In previous work, we proposed an ontology-based communication and coordination system for the formation of impromptu teams of heterogeneous robots. We extend this work to consider the capabilities of both the robot and a human user in the production of multimodal communicative behaviors to facilitate user needs and preferences in an interaction.**

I. Introduction

As humans, we know that our abilities to sense the world and our physical capabilities to interact with the world shape the way we ground concepts and communicate them [1]. The capabilities and sensing modalities of a robotic agent have analogous impact on its abilities to communicate. This is evident in the coordination of impromptu teams of heterogeneous robots [2]. Moreover, the characteristics of a human user influence the way that the robot should interact; specifically, the sensing and processing (i.e., interpretation) abilities of the user determine the channels and dynamics of communication that should be utilized by the robot to transmit and receive information.

Verbal and nonverbal modalities make up what can be considered "typical" human communication. However, there are some populations whose circumstances impair such social interaction [3]. For example, children with autism spectrum disorder (ASD) tend to avoid eye contact and, thus, often miss communicated intentions and emotions expressed in the face and body; the early-to-moderate stages of Alzheimer's disease often limit a patient's vocabulary; post-stroke rehabilitation patients frequently have reduced motor activity, thus limiting social expressiveness. We consider the capabilities of both the robot and the user in the production of multimodal communicative behaviors to facilitate the specific preferences and needs of the user in an interaction.

Manuscript received February 15, 2010. This work is supported in part by the National Science Foundation under Grants CNS-0709296, IIS-0803565, and IIS-0713697.

R. Mead is with the University of Southern California, Los Angeles, CA 90089 USA (phone: 213-740-6245; fax: 213-821-5696; e-mail: rossmead@usc.edu).

J. B. Weinberg is with Southern Illinois University Edwardsville, Edwardsville, IL 62026 USA (e-mail: jweinbe@siue.edu).

M. J. Matarić is with the University of Southern California, Los Angeles, CA 90089 USA (e-mail: mataric@usc.edu).

II. Background

In previous work, we presented an ontology-based symbolic communication protocol and Agent Interaction Manager for coordinating impromptu teams of heterogeneous robots [2]. Ontological reasoning provides a sense of meaning to information. By representing a robot's abilities, perceptions, and goals as symbols relating to concepts in ontologies, a robot is able to meaningfully share its symbols with other agents via network communication.

Tejada *et al.* [4] and Browning *et al.* [5] discuss mechanisms for humans to coordinate with homogeneous and heterogeneous robots, respectively. Each of the proposed techniques incorporates a traditional computer interface for communication. However, humans rely on verbal and nonverbal modalities, such as speech and body language, to convey information. Likewise, a social robot should utilize similar modes of communication.

III. Approach

We revisit our ontology-based Agent Interaction Manager (AIM) [2], and extend it for interactions with humans. In the previous implementation, robots communicated concepts over a network using the AIM protocol; however, the channels utilized in human interaction require the robot to communicate concepts via verbal and nonverbal modalities.

A. AIM Server—Agent Profiles and Templates

At initialization, a robot AIM client (Fig. 1a) uses the AIM protocol to communicate its concepts to an AIM server (Fig. 1b). These concepts define the ontologies necessary to form the robot's *agent profile*—an ontology generated dynamically using the concepts expressed by the robot [2].

In the presence of another agent, the robot sends an AIM message requesting the instantiation of a new profile unique to the agent. This profile is initialized using an *agent template*, a generic representation of the concepts and modalities utilized by a category of agent. For example, if the robot identifies the agent as another robot, then the local area network is an assumed medium for communication; however, if the agent is identified as human, then verbal and nonverbal modalities are assumed, as well as some understanding of concepts represented by the ontologies. AIM is then responsible for maintaining the profiles of all known agents, recognizing common concepts and modalities between them. This is inspired by computational models for the theory of mind [6], and has implications with regard to special-needs populations, such as children with ASD [3].

B. AIM Client—Agent Model and Interfaces

The *agent model* refers to the agent's knowledge representation of the world, as well as necessary meta-knowledge to communicate concepts to AIM through agent interfaces. An *agent interface* converts the representation with respect to an interaction modality into ontological concepts that can be exchanged in the AIM server (Fig. 1a).

In AIM, information is presented in a structured form, similar to a descriptive sentence, containing a subject, a predicate, and an object [2]. By enforcing this strict syntax, concepts can be broken down and related to the traits of others. This embedded grammar can be used to produce an interface for *verbal communication* (e.g., text-to-speech [7]).

A robot must utilize its own embodiment to communicate to a human. There are two types of nonverbal behavior: speech-independent and speech-dependent [8].

Speech-independent nonverbal communication requires that the robot physically convey the essence of a concept. For example, facial expressions can be used to express emotion; deictic gestures, such as eye gaze or pointing, can be used to indicate a point of interest; shape-related gestures can be used to illustrate the form of an object [9]. Balch & Parker [10] suggest that agents exchange concepts in three ways: (1) *iconically*, expressed physically similar to the concept itself; (2) *indexically*, establishing connections between iconic representations; and (3) *symbolically*, providing semantic relationships among all three representations. We are investigating physical manifestations of these representations.

Speech-dependent nonverbal communication requires that the robot produce socially expressive behaviors that compliment the verbal channel. Lee & Marsella [11] describe a rule-based NonVerbal Behavior Generator (NVBG) that converts the function of a communicative act (i.e., speech content and emotion) to "coverbal" (i.e., synchronized verbal and nonverbal) behaviors for embodied conversational agents; a "behavior realizer" is then used to carry out the coverbal act. We are in the process of integrating NVBG and implementing behavior realizers for a variety of anthropomorphic and non-anthropomorphic robots (http://robotics.usc.edu/interaction/?l=Laboratory:Facilities).

IV. Implementation

A series of agent templates and interfaces will be validated on various typically-developed/unaffected user groups, and then extended to focus on special needs populations, such as children with ASD, people with Alzheimer's disease, and post-stroke rehabilitation patients, in an effort to improve or optimize human task performance.

References

[1] G. Lakeoff and M. Johnson, *Metaphors We Live By*, Chicago, IL: University of Chicago Press, 1980.

[2] R. Mead and J. B. Weinberg, "Impromptu teams of heterogeneous mobile robots," *Student Abstract and Poster Program of the 22nd AAAI National Conference on Artificial Intelligence (AAAI-07)*, Vancouver, B.C., pp. 1890-91, July 2007.

[3] A. Tapus, M. J. Matarić, and B. Scassellati, "The grand challenges in socially assistive robotics," *IEEE Robotics and Automation Magazine*, vol. 14, no. 1, pp. 35-42, 2007.

[4] S. Tejada, A. Cristina, P. Goodwyne, E. Normand, R. O'Hara, and S. Tarapore, "Virtual synergy: a human-robot interface for urban search and rescue," *Proceedings of the AAAI 2003 Robot Competition*, Acapulco, Mexico, 2003.

[5] D. H. Browning, M. B. Dias, T. K. Harris, B. Browning, E. G. Jones, B. Argall, M. Veloso, A. Stentz, and A. Rudnicky, "Dynamically formed human-robot teams performing coordinated tasks," *Proceedings of the 20th AAAI National Conference on Artificial Intelligence (AAAI-05)*, Pittsburgh, PA, July 2005.

[6] B. Scassellati, "Theory of mind for a humanoid robot," *Autonomous Robots*, vol. 12, pp. 13-24, 2002.

[7] G. Dorai and Y. Yacoob, "Facilitating semantic web search with embedded grammar tags," *IJCAI Workshop on E-Business and Intelligent Web*, Seattle, WA, pp. 40-45, August 2001.

[8] M. L. Knapp and J. A. Hall, *Nonverbal Communication in Human Interaction*, 7th edition, Boston, MA: Wadsworth Publishing, 2009.

[9] S. Kopp, T. Sowa, and I. Wachsmuth, "Imitation games with an artificial agent: from mimicking to understanding shape-related iconic gestures," *Gesture-Based Communication in Human-Computer Interaction (LNAI 2915)*, Springer, Berlin, pp. 436-447, 2004.

[10] T. Balch and L. Parker, *Robot Teams: From Diversity to Polymorphism*. Natick, Massachusetts: A K Peters, Ltd, 2002.

[11] J. Lee and S. Marsella, "Nonverbal behavior generator for embodied conversational agents," *6th International Conference on Intelligent Virtual Agents*, Marina del Rey, CA, 2006, pp. 243-255.

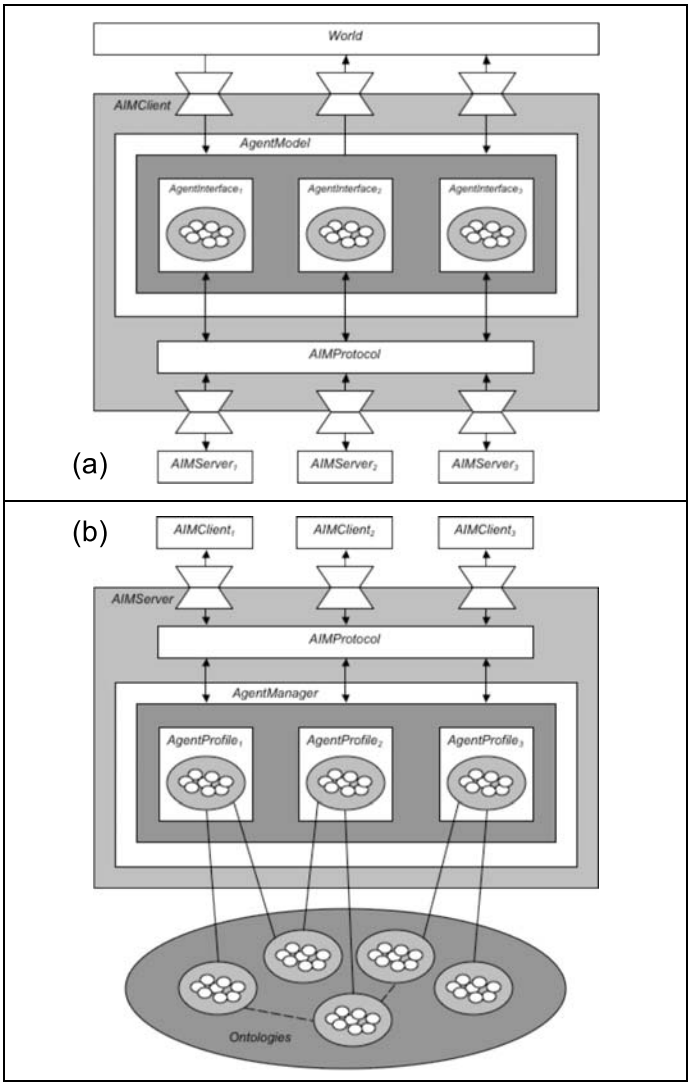


Fig. 1: The Agent Interaction Manager (a) client and (b) server.

Proceedings of the IEEE ICRA 2010
Workshop on Multimodal Human-Robot Interfaces
Anchorage, Alaska, May, 2010

Exploring Multimodal Interfaces For Underwater Intervention Systems

J. C. Garcia, M. Prats, P. J. Sanz, *Member, IEEE,* R. Marin, *Member, IEEE, and* O. Belmonte

***Abstract*— Graphical User Interfaces play a very important role in the context of Underwater Intervention Systems. Classical solutions, specially concerning Remotely Operated Vehicles, frequently require users with an advanced technical level for controlling the system. In addition, continuous human feedback in the robot control loop is normally needed, thus generating a significant stress and fatigue to the pilot.**

This paper shows work in progress towards a new multimodal user interface within the context of autonomous underwater robot intervention systems. We aim at providing an intuitive user interface that can greatly improve the non-expert user's performance and reduce the fatigue that operators normally experiment with classical solutions. For this, we widely adopt advanced interaction systems such as haptic devices, projectors, Head-Mounted Display and more.

***Keywords*— Graphical User Interface (GUI), Autonomous Underwater Vehicle for Intervention (I-AUV), multimodal interface, simulator.**

I. Introduction

CURRENTLY Remotely Operated Vehicles (ROVs) are commercially available to develop all kind of intervention missions. These systems are underwater robots tethered to a mother ship and controlled from onboard that ship. Here the control is assumed by an expert user, called the ROV pilot, by means of a special Graphical User Interface (GUI) with specific interaction devices like a joystick, etc. The main drawback in this kind of systems, apart from the necessary expertise degree of pilots, concerns the cognitive fatigue inherent to master-slave control architectures [1].

On the other hand, the best underwater robotics labs around the world are recently working for the next technology step, trying to reach new levels of autonomy far beyond those present in current ROVs. These technologies have lead to Autonomous Underwater Vehicles for Intervention (I-AUVs), which represent a new concept of undersea robots that are not tethered to a mother ship. In fact, the history about I-AUVs is very recent, and only a few laboratories around the world are currently trying to develop this kind of systems [2].

One of the most well-known research projects devoted to develop an I-AUV is SAUVIM [3]. Along its life, this project has implemented a GUI combining all kind of sensor data inside a common simulation environment. Their GUI uses its own programming language and allows for high level interaction of the user and the underwater robot in text mode. In addition, virtual reality (VR) is available within the GUI, thus showing the evolution of the complete system along the intervention mission, and assisting the user in the high-level control. This very complete interface has shown to be very suitable for users with an advanced previous expertise, but might be too complex for a new user without technical knowledge.

Our research group is working on this kind of underwater intervention systems in general, and more concretely in specific multimodal interfaces that allow an intuitive use by non-expert users. In fact, because of the impossibility to implement a complete I-AUV autonomy level with available technology, we design a two steps strategy [4], guaranteeing the "intelligence" in the system performance including the user in the control loop when strictly necessary, but not in a continuous way like in ROV's. Thus, in a first step, our I-AUV is programmed at the surface, and then navigates through the underwater Region of Interest (RoI) and collects data under the control of their own internal computer system. After ending this first step, the I-AUV returns to the surface (or to an underwater docking station) where its data can be retrieved. A 3D image mosaic is constructed, and by using a specific GUI, including virtual and augmented reality, a non-expert user is able to identify the target object and to select the suitable intervention task to carry out during the second step. Then, during this second step, the I-AUV navigates again to the RoI and runs the target localization and the intervention modules onboard. Our I-AUV system concept, currently under construction in Spain (i.e. RAUVI's Spanish Coordinated Project), can be observed in Figure 1, where the vehicle, developed in the University of Girona (Spain) and the arm, under responsibility of Univerisity Jaume I (Spain), that is an adaptation of the "arm 5E" from CSIP Company (UK) must be assembled in the next months. Moreover, it is noticeable that just now we are starting out the coordination of a European Project named TRIDENT within the same context but with a bit more challenging long term objectives.

Thus, this paper shows our ongoing research on

This research was partly supported by the European Commission's Seventh Framework Programme FP7/2007-2013 under grant agreement 248497 (TRIDENT Project), by Ministerio de Ciencia e Innovación (DPI2008-06548-C03), and by Fundació Caixa Castelló-Bancaixa (P1-1B2009-50).

J.C. García, M. Prats, P.J. Sanz and R. Marin are with the Department of Computer Science & Engineering, at the Universitat Jaume I, Spain ([garciaju,mprats,sanzp,rmarin]@uji.es).

O. Belmonte is with the Department of Computer Languages & Systems, at the Universitat Jaume I, Spain (Oscar.Belmonte@uji.es).

multimodal user interfaces for enabling the aforementioned kind of underwater intervention missions, initially focused on recovery object tasks. We aim to provide an intuitive user friendly interface improving the non-expert user's performance and reducing the inherent fatigue within traditional ROV interaction ways. Section II describes our recent efforts for building such an interface, including our ongoing work on immersive underwater simulation, facilities for target identification and task specification, and recent progress in grasp simulation. Section III clarifies the main drawbacks and advantages of our solutions when compared with the state of the art technologies, and also discusses the results obtained so far and the long list of challenges that need to be addressed. Finally, Section IV concludes this paper.

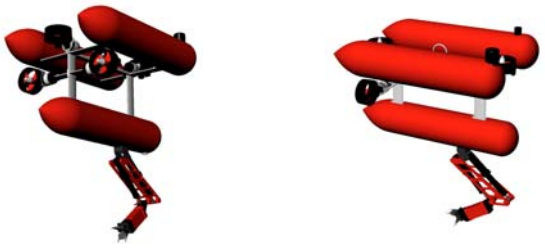


Fig. 1. The I-AUV envisioned concept currently under construction within the RAUVI's Spanish Coordinated Project.

II. TOWARDS A NEW MULTIMODAL INTERFACE

The whole mission specification system is composed of three modules: a GUI for object identification and task specification, a grasp simulation and specification environment, and the I-AUV Simulator. After target identification and specifying the intervention task, all the information is displayed into another 3D environment where the task can be simulated and the human operator can either approve it or specify another strategy by means of some facilities addressed within the interface. Finally, another environment is used for simulating and supervising the overall intervention mission. The ongoing work on these three modules is detailed in the following.

A. GUI for target identification and task specification.

Two main tasks must be solved in the underwater intervention context: the target identification and the specification of the suitable intervention to carry out over the target. Initially, a GUI is used for specifying the task to perform. Once the desired task has been selected, the GUI provides facilities for detecting interesting objects and identifying the target.

We are currently trying to expand the facilities available through the GUI for enabling a more intuitive level of interaction. In this way, the developed GUI (Figure 2) tends to be *user-friendly* with few requirements from the user side. Some examples of the intervention tasks to specify could be hooking a cable, pressing a button, etc. Currently we are focused on a specific task related with object recovery, where a suitable grasp has to be performed in order to manipulate in a reliable manner the target object.

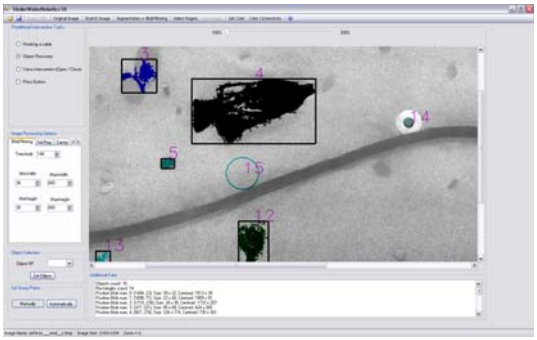


Fig. 2. An example of GUI screenshot: the object detection process

Looking for easy-to-use interaction ways, the GUI assists the user adapting its interface depending on the task to perform. Once the user has loaded the input image (i.e. first step in the process) and selected the intervention task, the user identifies the object and selects the target. For that, the GUI provides methods for object characterization and also for assisting in the grasping determination problem. The planned grasp will be later used in the grasping simulator and finally, in the real system. The general process can be observed in Figure 3. Due to the pour visibility conditions in the underwater environment and so, in the input image, the user could have problems to identify correctly the target. Low-level details about the different interaction ways currently available within thee GUI under development can be found elsewhere [5].

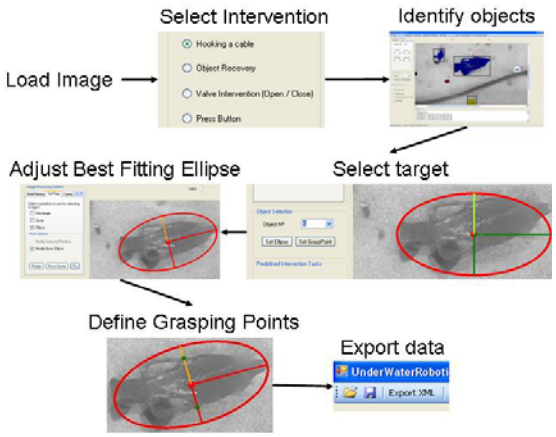


Fig. 3. Main steps through the GUI under development during the object characterization process.

The underwater scenario provides a hostile and very changing environment, including poor visibility conditions, streams and so on. So, the initial input compiled during the survey mission will be always different to the final conditions arising during the intervention mission. Thus, a predictive interface ensuring realistic task simulation is more than convenient before the robot be able to carry out the intervention defined by the user in the GUI.

B. Grasp simulation and specification.

Our most recent work is focused on an intuitive grasp simulation and supervision system that allows the user to visually check and validate the candidate grasps or to intuitively refine them in case they are not suitable. The grasping simulator will get data from the XML file generated by the previous object detection and task specification GUI. This data will include candidate grasping points and other target object properties that will be displayed in the simulator following augmented reality techniques (e.g. grip opening, joint angles, planned contact points, etc.).

The user's hand will be covered by a data glove with a tracking system that will allow replicating the human hand motion in the simulated environment. This will be used for specifying the required elements of a grasp (e.g. the hand configuration, grip opening, etc.), and also for indicating predefined actions through specific gestures (see Figure 4).

Fig. 4. Detail of the P5 data glove during a simple test: "grasp a virtual cube".

Our research team has a long experience in robotic grasping using the knowledge-based approach [6]. This approach defines a set of hand preshapes, also called hand postures or prehensile patterns, which are hand configurations that are useful for a grasp on a particular shape and for a given task. Several hand preshapes taxonomies have been developed in robotics, being the one proposed by Cutkosky [7] the most widely accepted. Since the publication of the Cutkosky's taxonomy, several researchers in the robotics community have adopted the grasp preshapes as a method for efficient and practical grasp planning in contrast to contact-based techniques.

One of our recent contributions in the field of robotic grasping is the concept of ideal hand task-oriented hand preshapes [8], which are a set of hand preshapes defined for an ideal hand and extended with task-oriented features. The ideal hand is an imaginary hand able to perform all the human hand movements. Our approach is to plan or define grasps by means of ideal preshapes, and then define hand adaptors as a method for the instantiation of the ideal preshapes on real robotic hands. The main advantage of this approach is that the same grasp specification can be used for different hands, just by defining a suitable mapping between the ideal hand and the real one. This concept is illustrated in Figure 6, which shows three different ideal preshapes and their mapping to a robotic Barrett Hand.

We plan to adopt this approach for the grasp specification and execution in the context of our grasp simulator. The human operator will specify a grasp using its own hand covered with a data glove. The finger joint angles captured by the data glove tracking system will be passed to a standard classifier (e.g. like in [9]) that will select the ideal hand preshape that best suites the human hand posture. The grasp will be specified by the ideal hand preshape and the part of the object where it is applied. For its execution by a robotic hand, the corresponding hand adaptor will transform the ideal preshape into a real posture depending on the robotic hand. The grasp will be finally simulated with the real robotic system as shown in Figure 5.

1) Low level details for the grasp simulator.

In order to develop the grasping simulation, some of the most common and used game and physics engine software, have been explored. A *game engine* is a software system designed for the creation and development of video games. The core functionality typically provided by a game engine includes a rendering engine for 2D/3D graphics, a physics engine or collision detection and response, and so on. On the other hand, a *physics engine* is used to model the behaviors of objects in space, using variables such as mass, velocity, friction, and wind resistance. It can simulate and predict effects under different conditions that would approximate what happens in real life or in a fantasy world. They are also used to create dynamic simulations without having to know anything about physics.

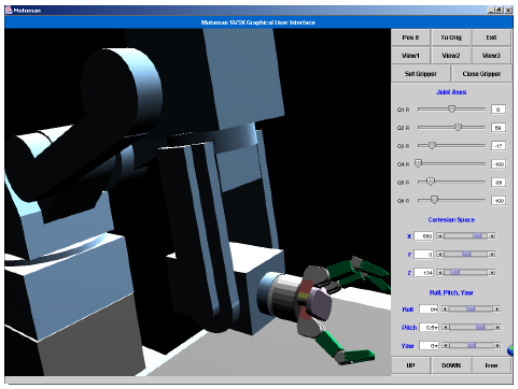
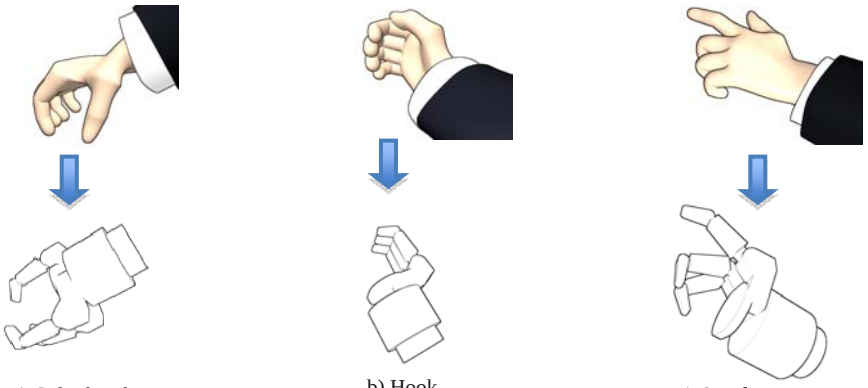


Fig. 5. GUI integrating the Barrett Hand 3D-model simulator

Despite both software platforms seems to be similar, a very important difference exists between them. The *physics engine* uses the Physics Processing Unit (PPU), which is a dedicated microprocessor designed to handle the calculations of physics, (e.g. rigid and soft body dynamics, collision detection or fracturing of objects). Using this dedicated microprocessor the CPU is off-loaded of high time-consuming tasks.



a) Cylindrical b) Hook c) One finger

Fig. 6. Three different ideal preshapes and their mapping to a Barrett Hand

In this way, the software compared is the jMonkeyEngine [10] (JAVA game engine) and PhysX [11] (physic engine). jME is a high performance scene graph based graphics API and is completely open source under the BSD license. A complete feature list can be found in [12]. On the other hand, PhysX is a proprietary solution of NVIDIA, but its binary SDK distribution is free under the End User License Agreement (EULA). A complete feature list can be found in [13].

The main difference between both engines is the platform compatibility and PC performance. Whereas jME is available for PCs (Windows, Linux and MacOS), PhysX is available for PCs (Windows and Linux) and all the actual videogames platforms (PS3, Xbox360, Wii). This justifies the number of more than 150 title games using PhysX technology. In PC performance terms, the use of a NVIDIA graphic card compatible with PhysX increases the general PC performance. Of course, with a SLI [14] schema with one dedicated graphic card, PhysX would deliver up to twice the PC performance (in frames per second). We should notice that PCs with an ATI graphic card would not get all the advantages of this technology, due to PhysX is a proprietary solution of NVIDIA, although they could still run the program.

Thus, in our first approach developing the grasping simulator, we are considering the NVIDIA physics engine. Besides the advantages explained before, we will try to take profit of the latest NVIDIA graphic card features, even using its 3D Vision technology [15]. This technology enables 3D vision over every single application, and only needs a *3dReady LCD monitor* and a NVIDA GeForce 3D Vision glasses.

C. I-AUV Simulator.

Previous research in this context has been developed in our Laboratory since 2008, starting with the cooperation with the University of Bologna, Italy, in order to implement a complete simulator [4]. This simulator includes a complete I-AUV 3D model, and emulates the physics of both the underwater environment and the robotic system. Currently, we are improving the user interaction capabilities by using a *Head Mounted Display* with an accelerometer, enabling to control the virtual cameras by means of the human head's movements. Further development could also include data gloves for gesture recognition, as can be observed in Figure 7.

Fig. 7. The initial simulator under development.

On the other hand, another I-AUV simulator is being developed at our laboratory, as observed in Figure 8. Its main features are the distributed and collaborative properties, as well as the use of advanced Virtual Reality (VR) devices. Low-level details can be found elsewhere [16]. This simulator uses a distributed and collaborative system, which enables to combine remote data coming from different PCs that can be placed in different locations. Thus, different users can work in cooperation with this kind of interface achieving simultaneously task specification missions/simulations that can be observed by different users in real time.

Fig. 8. The I-AUV is teleoperated by the user by means of special VR hardware, including immersive 3D vision and a virtual joystick controlled with data gloves.

In particular, this kind of cooperative interface opens new capabilities for personal training, enabling the possibility of sharing the VR interface among several remote experts and

non expert's users. In this way, researchers on different disciplines can focus on the simulation aspects that are most interesting for their research, either if they are not physically present in the ship.

However, this cooperative VR interface has a serious drawback: the high costs underlying the specific hardware resources included in such a system.

III. Discussion

After exploring different possibilities of interfaces including all kind of VR devices, simulators and the potential of cooperative work, it is clear that significant benefits can be achieved. Probably one of the main advantages is what concerns the user training. In fact, the interaction by means of more intuitive and user friendly interfaces would allow reducing the pilot training period. In particular, the use of the developed VR technology, including distributed, collaborative and multimodal components, allows the user to interact in a very realistic way with the intervention scenario, promoting prediction actions. In addition, it allows appreciating the nature of the problems in case the simulation of the mission plan fails.

The most important difference between our approach and other existing solutions is that we put a special emphasis on the use of advanced technologies and functionalities making easier the human robot interaction for non-expert users.

For instance, the SAUVIM's GUI integrates several modules into one single interface, so the overall user interface provides a very powerful and flexible solution for monitoring the state of the robot during the mission, and provides advanced mechanisms for the low-level control. However, the interface has been designed for expert users that require an advanced technical background, including very specific and intensive training periods.

In contrast, our GUI is being developed focusing basically on the user experience. In fact, the GUI is divided in three different applications: the object identification & task specification GUI, the grasping simulator and the general I-AUV simulator. All of them make use of advanced devices for human-computer interaction (e.g. data gloves, Head-mounted Displays, etc.) and enabling an immersive 3D environment where interaction is more satisfactory for the end-user.

However, this project is still in a preliminary stage and needs further research for a complete validation. So, in the work developed so far, we have analyzed several human-computer interaction devices that could potentially improve the way humans currently interact with underwater robotic systems. We have explored and implemented different possibilities that have to be carefully analyzed, having into account the end-user's requirements and preferences, before its final implementation. Therefore, future lines will mainly focus on a thorough analysis of the different options and the selection and complete implementation of the most suitable solution.

IV. Conclusions and Future lines

This work has presented the first steps towards the development of a *user-friendly* GUI for autonomous underwater intervention missions. We are considering an interface composed of three different applications for object detection and task specification, task simulation, and for the overall supervision of the mission. We claim that the use of new graphics technology and VR devices can greatly increase the overall immersive sensation of the user in the virtual world, thus facilitating its interaction with the robotic system even with little technical knowledge. Therefore, our explored solutions combine different interaction devices such as data gloves for the grasp specification and Head-mounted Displays for immersive visualization.

Our long-term objective is to reach new levels of human-robot interaction in the context of autonomous underwater intervention missions, thus improving the user's satisfaction and performance while using the system.

References

[1] Sheridan, T.B. "Telerobotics, Automation and Human Supervisory Control". MIT Press. 1992.

[2] Yuh. "Design and Control of Autonomous Underwater Robots: A Survey". In Int'l J. of Autonomous Robots 8, 7–24. 2000.

[3] Yuh, J.; Choi, S.K.; Ikehara, C.; Kim, G.H.; McMurty, G.; Ghasemi-Nejhad, M.; Sarkar, N.; Sugihara, K., "Design of a semi-autonomous underwater vehicle for intervention missions (SAUVIM)", Underwater Technology, 1998. Proceedings of the 1998 International Symposium on , vol., no., pp.63-68, 15-17 Apr 1998.

[4] De Novi, G., Melchiorri, C., García, J. C., Sanz, P. J., Ridao, P., Oliver, G., "A New Approach for a Reconfigurable Autonomous Underwater Vehicle for Intervention", In Proc. of *IEEE SysCon 2009 —3rd Annual IEEE International Systems Conference, 2009*, Vancouver, Canada, March 23–26, 2009.

[5] García, J. C., Fernández, J. J., Sanz, P. J., Marin, R., "Increasing Autonomy within Underwater Intervention Scenarios: The User Interface Approach". In Proc. of *IEEE International Systems Conference —4th Annual IEEE International Systems Conference, 2010*, San Diego, CA, USA, April 5–8, 2010. Accepted, pending of publishing.

[6] Stansfield, S.A. "Robotic Grasping of Unknown Objects: A Knowledge-based Approach". International Journal of Robotics Research, 10(4), 314–326. 1991

[7] Cutkosky, M., & Wright, P. "Modeling manufacturing grips and correlations with the design of robotic hands". Pages 1533–1539 of: IEEE International Conference on Robotics and Automation, vol. 3. 1986.

[8] M. Prats, P.J. Sanz and A.P. del Pobil. "A Framework for Compliant Physical Interaction: the grasp meets the task". Autonomous Robots, 28(1), pp. 89-111, 2010.

[9] S. Ekvall and D. Kragic, "Grasp Recognition for Programming by Demonstration". In IEEE Intl. Conf. On Robotics and Automation (ICRA), pp. 748-753, Barcelona, Spain, 2005.

[10] http://www.jmonkeyengine.com

[11] http://www.nvidia.com/object/physx_new.html

[12] http://www.jmonkeyengine.com/wiki/doku.php/complete_features_list

[13] http://developer.nvidia.com/object/physx_features.html

[14] http://www.slizone.com/page/slizone_learn.html

[15] http://www.nvidia.es/object/GeForce_3D_Vision_Main_es.html

[16] O. Belmonte, M. Castañeda, D. Fernández, J. Gil, S. Aguado, E. Varella, M. Nuñez, J. Segarra. In Int. Journal of Future Generation Computer Systems 26, 308-317. 2010.

Proceedings of the IEEE ICRA 2010
Workshop on Multimodal Human-Robot Interfaces
Anchorage, Alaska, May, 2010

Enhancing Collaborative Human-Robot Interaction Through Physiological-Signal Based Communication

Susana Zoghbi, Chris Parker, Elizabeth Croft and H.F. Machiel Van der Loos

Abstract— **In order to develop a friendly and safe interaction between humans and robots, it is essential for the robot to evaluate users' affective states and respond accordingly. This paper investigates the use of physiological signals to estimate human affective states during a Human-Robot Interaction (HRI) task. We focus on characterizing physiological responses and understanding how affective states evolve in a collaborative human-robot task. We propose to both design a model that maps physiological signals to affective states in real time and design a methodology for the robot to exhibit an appropriate behavior during the task in response to estimated changes in affective states.**

I. Introduction

There has been a long standing interest in designing robotic systems to help people with their daily activities: completing chores, caring for the elderly, etc. Today, robots primarily are found in industrial settings - isolated from human workers - to automate tasks that are either too dangerous or that require a greater throughput or precision than a human worker can provide. Safe and robust human-robot interaction (HRI) in shared workspaces, however, has yet to be realized outside of the laboratory.

When people cooperate on a load-sharing task (*e.g*, carrying a table together) they use explicit and implicit cues to communicate with each other the actions they intend to take, their perception of task progress, and how the shared goal should be modified. A portion of interpersonal communication relies on implicit cues [4]. Furthermore, the communication/recognition of affective states is important to and expected by cooperating humans [5]. Therefore, robots intended to work with humans on shared tasks should be able to perceive their human partners' actions and intentions conveyed in both explicit and implicit modes. Additionally, a robot should use these modes to communicate its own plans and intentions. The ultimate goal of our research is to develop strategies for safe and intuitive interaction between humans and robots by enabling the robot to recognize affective state changes in its human partner and respond accordingly.

Several explicit and implicit cues can be used to estimate affective states in a partner, *e.g.*: characteristics of speech, facial expressions, gestures, postures, and physiological signals. This research will focus on the last of these cues to infer affective states during HRI. Physiological signals provide quantifiable measures that tend to be involuntary as well as age and culture independent.

E. Croft, H. Van der Loos, C. Parker and S. Zoghbi are with the Department of Mechanical Engineering, University of British Columbia, 6250 Applied Science Lane, Vancouver, Canada ecroft@mech.ubc.ca

A. Specific Objectives

To study physiological-signal based implicit communication between robots and people, we will focus on the following four research objectives.

1) Characterize real-time human physiological responses elicited by the robot partner in an interactive task: The time and frequency domain physiological responses of the human in an HRI task will allow us to determine whether or not statistically significant changes in these parameters can be made online and if relevant features can be extracted from them in our HRI context.

2) Develop a model of the dynamics of the real-time evolution of affective states during a human-robot interaction task: While interacting with a robot, a person's affective state may change in response to different robot motions. We plan to study how these changes occur over time and which robotic stimuli elicit them.

3) Design a model to map physiological signals to affective states in real time: A sufficiently reliable model to map physiological responses to affective states would enable a robot to decode important implicit cues displayed by a human partner. Moreover, it will help validate the suitability of extracted physiological features while eliminating redundant or useless ones. Machine-learning techniques will be used to design such a model in real time.

4) Design a methodology for the robot to exhibit an appropriate behavior during a collaborative task: We will investigate appropriate responses of a robot to its human user in specific task-oriented scenarios in response to changes in the user's affective state. For example, should the robot alter its behavior if it infers a decrease in its user's affect, and if so, how? We will consider how responses should be defined and evaluated given a prescribed task context.

II. Literature Review

There is a rich body of psychophysiological literature related to affect estimation and the use of affect for Human-Computer and Human-Robot Interaction domains. However, very few studies have used the robot as the primary elicitor of physiological responses and changes in affect. Picard et al. [6] identified patterns in four physiological signals - electromyography (EMG), blood volume pressure, galvanic skin response (GSR) and respiration) - from an actor expressing eight different emotions, and were able to develop an emotion classifier that achieved 83% accuracy. Rani et al. [7], were able to recognize anxiety in five users based on several physiological signals (i.e., cardiac, electrodermal and electromyographic activity as well as temperature) using

regression trees and fuzzy logic. Estimated affective states were compared to the subjects' self reports. Their earlier work was used to drive mobile robot behavior in simulated rescue domain [8].

Liu et al. [3], using a large set of physiological features and support vector machines, designed an affect recognition model achieving a success rate of 83% with children with autism spectrum disorders. Kulic and Croft [2] used Hidden Markov Models (HMM) to estimate affective state in response to various robot motions. Physiological signals including heart rate, GSR, and EMG. Offline questionnaires were used to assess users' affective states, represented using the valence-arousal model. User-specific HMMs successfully recognized valence and arousal better than 80% of the time.

III. Proposed Methodology

To achieve our first two specific objectives, we propose to perform a series of experimental trials in which a person and a robot perform a task together. The robot holds one end of an object with a pointing device attached (i.e., a laser pointer) and the person holds the other end. The human and the robot then trace a 2-dimensional path that has been drawn on a horizontal surface with the pointer. This task is analogous to many real-life scenarios: for example, in industry, a robot holds a heavy tool and the person guides the motion; in hospitals or care homes, for the elderly or patients who have diminished limb strength; in space station assembly as astronauts sometimes lose tools during repair missions. In all these tasks, the user decides when the task is done and the robot provides assistance for the task.

A CRS A460 robotic arm (human-sized) will be used with an ATI 6-axis force/torque sensor attached to the gripper. Robot behavior is set by an impedance controller (similar to that of [1]). During each trial the virtual impedance (i.e., mass and viscous damping), are changed randomly. The user is instructed to trace the path in each trial for one of the following two conditions: i) as fast as possible (speed), ii) as accurately as possible (accuracy). It is hypothesized that different values of the virtual parameters and/or random disturbances elicit changes in affect, as the task becomes easier or harder to perform. Throughout the experiment, several physiological signals are collected: i.e., electrocardiography, EMG, GSR, skin temperature, respiration rate and electroencephalography. After each trial, the user is asked to fill in a questionnaire to report their level of performance, effort, frustration, comfort, engagement, boredom and perceived helpfulness of the robot. Additionally, subjects will report their affective state based on video recordings of themselves during the trial.

To design a model to map physiological signals to affective states in real time (objective 3), a dynamic Bayesian inference network will be used. In this model, we will consider affective states as hidden variables and physiological signals as a high-dimensional vector of observations. We assume a first-order Markov process with observation variables depending only on the current hidden state. The parameters of the probability density function (pdf) of transition and observation are estimated using Maximum Likelihood. A recursive algorithm is used to make estimations in the Bayesian network [9].

Given affective state estimations, we will propose models of how the robot should respond (objective 4). The aim is to provide a decision-making process for the robot to appropriately adjust its behavior. Machine-learning algorithms for supervised learning will be investigated in this stage. These methods will be evaluated through user trials. It is expected that this will be an iterative process in which desirable behavior of the robot in response to estimated affect will be elucidated first from "Wizard of Oz" experiments. Outcomes of these trials will provide input to the decision making system that will then be evaluated in a series of trials to explore system effectiveness as well as the effect of the robot's failure to respond appropriately.

IV. Summary

In order to develop a comfortable and effective interaction between humans and robots, we focus on incorporating physiological measures as implicit cues for a robot to both recognize affective states in its human partner and behave appropriately. To this end, we focus on four specific objectives: 1) Characterize real-time human physiological responses elicited by the robot partner in an interactive task, 2) Understand the dynamics of the real-time evolution of affective states during a human-robot interaction task, 3) Design a model to map physiological signals to affective states in real time and 4) Design a methodology for the robot to exhibit an appropriate behavior during a collaborative task. This paper has presented a brief outline on how we propose to achieve these objectives.

References

[1] D. De Carli, E. Hohert, C. A. C. Parker, S. Zoghbi, S. Leonard, E. Croft, and A. Bicchi. Measuring intent in human-robot cooperative manipulation. In *Proc. IEEE International Workshop on Haptic Audio visual Environments and Games HAVE 2009*, pages 159–163, November 7–8, 2009.

[2] D. Kulic and E. Croft. Estimating robot induced affective state using hidden markov models. In *Proc. 15th IEEE International Symposium on Robot and Human Interactive Communication ROMAN 2006*, pages 257–262, September 6–8, 2006.

[3] C. Liu, K. Conn, N. Sarkar, and W. Stone. Physiology-based affect recognition for computer-assisted intervention of children with autism spectrum disorder. *International Journal of Human-Computer Studies*, 66(9):662 – 677, 2008.

[4] A. Mehrabian and S. R. Ferris. Inference of attitudes from nonverbal communication in two channels. *Journal of Consulting Psychology.*, 31(3):248–252, 1967.

[5] R. W. Picard. *Affective Computing*. MIT Press, 1997.

[6] R.W. Picard, E. Vyzas, and J. Healey. Toward machine emotional intelligence: Analysis of affective physiological state. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23:1175–1191, 2001.

[7] Pramila Rani, Nilanjan Sarkar, and Julie Adams. Anxiety-based affective communication for implicit human-machine interaction. *Advanced Engineering Informatics*, 21(3):323 – 334, 2007. Applications Eligible for Data Mining.

[8] Pramila Rani, Nilanjan Sarkar, Craig A. Smith, and Leslie D. Kirby. Anxiety detecting robotic system - towards implicit human-robot collaboration. *Robotica*, 22(01):85–95, 2004.

[9] S. Russell, P. Norvig, and J. Canny. *Artificial Intelligence*. Prentice Hall, 2003.

Proceedings of the IEEE ICRA 2010
Workshop on Multimodal Human-Robot Interfaces
Anchorage, Alaska, May, 2010

Towards an Enabling Multimodal Interface for an Assistive Robot

Martin F. Stoelen, Alberto Jardon, Fabio Bonsignorio, Juan G. Victores, Concha Monje and Carlos Balaguer

Abstract— **The development of assistive robots for elderly and disabled people is currently an active field of research in the robotics community. An important part of making these systems usable is to allow for multimodal Human-Robot Interaction (HRI). However, the overall human-machine system is complex. The user and the robot are operating in a closed loop and both are potentially capable of adapting to the other. The work presented here has attempted to approach the problem from three different perspectives, investigating methods for analyzing, implementing, and testing an enabling multimodal interface for the ASIBOT assistive robot. It was proposed to use principles from Information Theory as the basis for the analysis, with the goal of increasing the information capacity of the human-machine channel. Multimodality was identified as one possible approach for achieving this. Methods for performing information fusion and machine learning that might be of interest for the implementation were identified. It was speculated that reinforcement learning could serve as an on-line adaptive component in the interface. Finally, the use of standard movement models and tasks as the basis for testing multimodal HRI was discussed and linked to typical tasks for assistive robots.**

I. INTRODUCTION

Assistive robots are currently being developed to support disabled and elderly people inside their own homes and in other everyday environments. One of the interesting challenges for the Human-Robot Interaction (HRI) in assistive robotics is the broad range of users and disabilities that needs to be catered for. Approaches to this problem range from a simple scanning interface requiring only the actuation of a single button to a wide range of input devices which can be personalized for each user. However, there has been less focus on true multimodal interaction so far. This involves fusion of different types of input modalities in order to, among other objectives, improve error handling and reliability [1]. Some exceptions do exist however, for example in the related field of rehabilitation robotics [2]. The ultimate goal of the work presented here is to achieve an "enabling" interface for the ASIBOT assistive robot, which can adapt to and compensate for the disabilities of a given user through the use of learning, contextual information, and multimodal interaction.

ASIBOT is a portable assistive robot for elderly and disabled people, which aims to give these users more freedom in daily tasks [3]. It is a manipulator robot with 5 degrees of freedom which can climb between special docking stations mounted in the environment as well as mount itself to a user's wheelchair. The current version of ASIBOT has a set of input devices that allow the user to employ different modalities to control the robot in joint-by-joint mode, Cartesian mode, and to execute simple pre-programmed tasks. This includes a joystick, a speech recognition system, a PDA menu system, and a flexible acceleration-based pointer input device that can be used with different parts of the body [4]. More input device modalities are currently in development and planning, including chin joysticks and eye tracking devices. In addition, the current interface architecture is being extended to allow for true multi-modal interaction. This will begin with low-level teleoperation commands and will later be extended to higher-level commands when a more mature level of autonomy is available.

The work presented here is part of a framework that has a focus on user-in-the-loop development. This aims to increase the shared knowledge between the users and developers of assistive robots and will hopefully lead to a higher degree of usability for these devices. An important aspect of this is HRI that maximizes the flow of useful information between the user and the robot. This paper attempts to identify methods for analyzing, implementing, and testing an enabling multimodal interface for the ASIBOT robot as a first step towards this goal.

II. ANALYSIS

Fig. 1 is a simplified representation of the complete human-machine system for multimodal assistive HRI. As can be seen in the figure, the model assumes that the user has some intentional commands for the robot, **h**, that are actuated through a set of input devices. The disabilities of the user are modeled as sources of noise, **z**, which can be independent for each input modality. The multimodal signals received by the enabling interface, **d**, are thus noisy representations of the user's true intention. The goal of the enabling interface is to use these noisy signals and time copies thereof, together with information from the context (**e**), to produce robot commands (**m**) that are as close as possible to the user's original intention. The user receives noisy feedback about the state of the robot, **x**, closing the loop. Feedback from the state of the input devices (visual and/or proprioceptive) is omitted for clarity in Fig. 1. Both the user and the interface are assumed to potentially be able to perform some form of adaptation and learning.

One interesting approach to analyzing complex closed loop systems like the one shown in Fig. 1 can be found

Martin F. Stoelen, Alberto Jardon, Juan G. Victores, Concha Monje and Carlos Balaguer are members of the RoboticsLab research group within the Department of System Enginnering and Automation, Universidad Carlos III de Madrid, (mstoelen, ajardon, jcgvicto, cmonje, balaguer)@ing.uc3m.es

Fabio Bonsignorio is a member of the RoboticsLab as Santander Chair of Excellence in Robotics at Universidad Carlos III de Madrid and is CEO/founder of Heron Robots of Genova, Italy, fabio.bonsignorio@uc3m.es

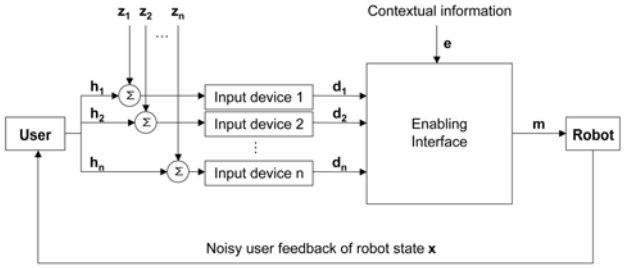


Fig. 1. The problem statement for multimodal assistive HRI.

in [5]. This is based on representing a complete control system as a directed acyclic graph of random variables and analyzing it using concepts from Information Theory [6]. The system includes the current state X, with values $x \in \mathcal{X}$, and the future state X'. The random variable representing the controller, C, then senses the current state and actuates to achieve the future state. This can be represented by conditional probabilities, $p(c|x)$ and $p(x'|x,c)$. These can be viewed as representing a sensor and actuation channel, respectively. The authors were further able to derive the conditions for observability, controllability, and optimality using this method.

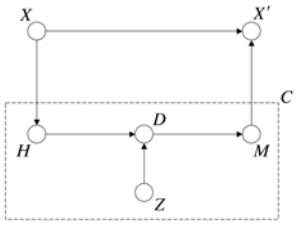


Fig. 2. The human-machine system as a directed acyclic graph.

Fig. 2 depicts our extension of this method to the human-machine system. The controller C here includes both the user (more generally the Human, H) and the assistive robot (more generally the Machine, M). The goal of the human-machine system in the most general sense is then to maximize the flow of useful information between the human and the machine over a noisy medium. Thus, we are interested in the communication channel existing between a source H and a receiver M, which will be denoted the "human-machine channel" in the following discussion and which has channel capacity C_{HM}. The information available in the source can be represented by the Shannon entropy per second of the random variable representing the human, here denoted as H_{Human}. The definition for entropy used here is shown in (1).

$$H(X) = -\sum_{x \in \mathcal{X}} p(x) \log p(x). \tag{1}$$

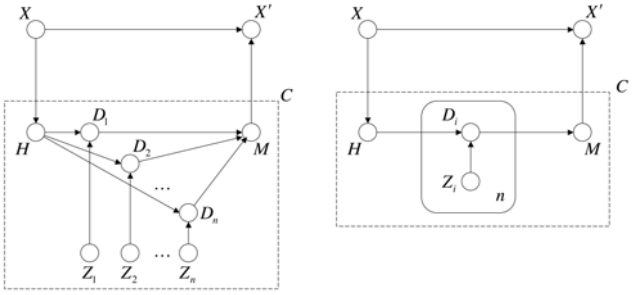
In Information Theory terminology the stated goal is then equivalent to transmitting this information over the human-machine channel with a minimum of errors. Chan and Childress [7] also applied information theory principles to analyze the information transmission in the human-machine system for tracking tasks. The analysis here differs in that it entails multimodality as well as learning, and is applied on the directed acyclic graph representing the system.

More specifically, the goal of the work presented here can be defined as maximizing the flow of useful information between the user and the assistive robot, given the user's physical disability. As can be seen in Fig. 2, the disability is also here modeled as a source of noise, Z. The random variable representing a given input device, D, will then depend probabilistically on both the user's true intentions, H, and the noise Z. A model with a user both mentally and physically healthy will not include this noise. Assuming that input devices with sufficient performance are available to the user, we would then have $H_{Human} \leq C_{HM}$. As stated in the channel coding theorem [6], there exists a coding system for this situation such that the information from the source, the user's intended commands, can be transmitted with an arbitrarily low error.

The interpretation of a mentally healthy, but physically disabled user attempting to control a complex system like an assistive robot is then that of a source rich in information, but acting over a human-machine channel with limited channel capacity. In other words, a situation where $H_{Human} > C_{HM}$. From the channel coding theorem we know there is no way to transmit information over the human-machine channel with errors smaller than $H_{Human} - C_{HM}$. However, we also know that encoding can keep the errors close to this value. Thus, there are in reality only two ways of augmenting the flow of information between the user and the robot. One is to increase the capacity of the human-machine channel. Another is to enable an efficient encoding. There are several potential approaches to achieve the former. One is to increase the number of input devices, enabling multimodal interaction with the user. This situation is depicted in Fig. 3. Two other might be using information about the context and information from past user inputs. These will not be described further here.

The main purpose of multimodality is then here to reduce the number of errors introduced by the disability of the user. The most obvious way to achieve this from an engineering standpoint would be through highly synchronized and redundant commands from the different modalities. That is, the user will simultaneously coordinate the different modalities so as to make his/her intention clearer to the system. A question that immediately arises is whether this increased need for organization would also reduce the output of the user, in effect reducing the information available in the source of the human-machine channel. The answer will probably depend on the type of modalities to coordinate. In fact, many forms of multimodal interaction do not involve redundancy of content nor simultaneous signals [8].

However, the procedure followed here is to begin with



(a) Normal notation. (b) Compact "plate" notation.

Fig. 3. Increasing the information capacity of the human-machine channel through input device redundancy.

low-level commands and teleoperation. The most typical input devices for this purpose will involve physical movements made by the user, although distinct modalities like electromyography (EMG), electroencephalography (EEG) and speech are also possible, see [9] for an insightful comparison. Looking at the literature on bimanual human movements, there are indications that symmetrical and synchronous movements of the two hands are not much slower than those using just one hand. For example [10], where there was no significant difference in the time taken to reach to two distinct targets (but with similar difficulty) than to one, although the reaction time increased. This is related to the idea that such bimanual movements share a common planning in the nervous system, at least for movements of equal duration of the two limbs [11]. As the user here would in essence be controlling the system on the same task with the different modalities, this finding might be applicable. Whether it also extends to movements of other parts of the body (for example head and shoulder movements) is not clear, but not entirely unlikely.

In the end, a multimodal interface for an assistive robot will need to cater for both redundant and complementary information coming from the user, with different degrees of synchronization between them. The requirements will likely differ with high- and low-level commands, with the latter having a higher degree of synchronization and simultaneity of commands. It will also likely vary with the specific disability of each user. Compared with users that have no disabilities, the users of assistive robots might require a higher degree of redundancy of modalities to overcome the extra errors introduced by the disability.

III. IMPLEMENTATION

A. Requirements

This section attempts to outline methods for addressing the requirements for the multimodal enabling interface. From our analysis we can identify the following capabilities that would be beneficial to include in the final system:

- Allow for multimodal interaction
- Make use of contextual information
- Learn and adapt to user, online and offline
- Be easily adjusted to different users
- Be easily verified experimentally

From these requirements it can be seen that both an element of information fusion and learning will be required. These are addressed in section III-B and III-C, respectively.

B. Information Fusion

Multimodal interfaces require the fusion of information from a range of input devices, where the data may arrive at different times and be very different from device to device. This is a similar problem to that faced in the field of sensor fusion, where a range of disparate sensors readings arriving at different times are used to improve the estimate of the state of one or several systems. For example the location and orientation of a mobile robot with respect to its environment. Sensor fusion in general is an expansive field and includes many different methods. One of the approaches used is Bayesian sensor fusion. For a system to be considered Bayesian it must have three characteristics [12]. One is a prior distribution, $p(A)$, representing the state of the system, A, before receiving the current set of data from a sensor, B. Another is a likelihood function, $p(B|A)$ that characterizes the information in the sensor. Bayes' theorem then allows for the calculation of the posterior, $p(A|B)$, the probability of the system having state A given the current sensor reading B. Assuming *n* independent sensor readings to be fused, $\mathbf{B}$, the posterior can be written as in (2).

$$p(A|\mathbf{B}) \propto p(A) \sum_{i}^{n} p(B_i|A) \tag{2}$$

Applied to a multimodal interface, the sensor readings can be interpreted as the noisy data coming from the different input devices and the state as the true intention of the user. For the graphs in Fig. 3 it could be visualized as updating the belief of H with knowledge of the values of the *n* input device signals D. This is a simple example of the insights that can be gained with Bayesian sensor fusion but does not necessarily represent an algorithm for achieving this in practice. It is interesting to note that Bayesian integration of information has also been suggested as the driving principle behind sensorimotor learning in humans [13].

C. Learning

Learning is another important aspect of assistive HRI. There is ongoing work on using machine learning to attempt to reduce the effect of a user's disability in daily tasks. This includes adaptive filters, for removing the effects of tremorous movements from the control signal or for physically counteracting the tremor [2]. Tremor reduction is also important in medical robotic applications [14]. There are also approaches that aim to adapt to other disabilities that cannot easily be separated from the user's intended signal in the frequency domain. For example, a weak left driving signal or inaccurate commands. This work has previously

been focused mainly on the problem of assisted wheelchair driving.

For example, using information on the context of operation to better understand the intent of the user [15]. This work was based on global trajectories to locations of interest in the environment. Each time step a set of potential user plans were calculated, based on the most likely trajectory to each of the points of interest. However, knowing the likelihood of a user plan given the user's current and previous input is difficult to infer directly. By using Bayes' theorem the problem was reduced to estimating the likelihood of the current user input given that the user had a specific plan in mind. This could readily be estimated from how different the user's commands were from those estimated for that plan at that moment. The commands corresponding with the most likely plan was then fused with the user's commands in a shared control scheme.

A related approach has attempted to also learn a general model of the user's disabilities [16]. This work utilized a form of supervised learning, Gaussian processes, to learn the way a specific wheelchair user performed a set of local trajectories (not to global points of interest but rather a set of integrated velocity commands). This knowledge could then be utilized in estimating the probability that this specific user would give the current input, given that he/she had a specific local plan in mind. The resulting system was trained on recorded user data and tested for plan recognition performance on a set of the same data with encouraging results.

It may also be possible to extend this method to higher Degree-Of-Freedom (DOF) movements, although there are several potential issues. In the original work one Gaussian process was used for the linear and one for the rotational velocity, which could mean 6 Gaussian processes if extended to typical robot manipulator control. The number of possible user trajectories would also increase dramatically. In addition, learning and adapting to the user is complicated by the fact that a user's behavior depends on the behavior of the system under control. Thus, the complete human-robot system can be seen as two learning systems attempting to adapt to each other. This may not necessarily favor off-line learning. On-line simultaneous learning by the human and the robot should probably also be allowed for. This could also include learning the way a given user combines the different input devices of the multimodal interface. Usable adaptive multimodal interfaces have indeed been identified as important future work in the field of multimodal interaction [1].

Reinforcement learning [17] is a form of learning more geared towards online applications. This method is in essence based on a learning agent with a trial-and-error behavior, exploring the environment by performing actions on it and learning from the rewards (and penalties) returned. These rewards can be of a very simple and high-level nature, for example large and positive if the system has achieved a goal state and negative every step until that time. It could be interesting to also explore this learning method for the enabling interface envisioned here. The learning could be performed on simple tasks online with the user (and possibly offline), where the reward could be provided to both the user (by displaying a visual representation of the goal state) and the system (a numerical reward for achieving the goal state). There are several open questions about the application of this type of learning for interacting with a real user however. This includes the number of learning episodes required and the continuous nature of the user inputs and robot commands. Reinforcement learning with continuous states and actions is still active research, although successful applications do exist [18].

IV. TESTING

A. Task Selection

Multimodal human-robot interfaces like the one investigated here require testing. During development it can be used to effectively evaluate changes to the system. It is also required to document the potential benefit of the interface to the rest of the research community. The selection of the task to use as a basis for this testing can influence the value of the results obtained. Application-specific tasks are typically used for this purpose at the moment. For example, the work of Tsui et al. on combining a pointing device and a camera-screen pair [19]. This system was tested with disabled users on a typical domestic task for a wheelchair-mounted robot system, selecting and approaching an object (from several) in a bookshelf. Measures included the time to completion but also expert- and user-completed evaluations.

For the lower-level commands and teleoperation considered here comparisons of interfaces are typically also done on application-specific tasks. This may be sufficient if the tasks are limited and well known. However, assistive robots are typically intended for use in a user's daily environment. This environment can thus vary from user to user and is difficult to specify at design time. This makes it harder to come up with a representative set of tasks for a comparison of assistive HRI.

There also seem to be limitations on the conclusions that can be drawn from the results obtained with application-specific tasks. How can we be sure that the interface with the best performance on picking up a bottle of soda from a desktop is also the best on opening the refrigerator door? In other words, how can we know that our results are generalizable? In addition, application-specific tasks can in many cases include a component of user interpretation, for example the many ways to approach and grasp an object. How can we ensure that an increase (or decrease) in performance after a change in the interface is due to the change itself and not of the user interpreting the task differently?

Another approach is to use a more general task that is still similar to the one intended, like a peg-in-the-hole task. However, It would still be beneficial to have a reliable measure of the difficulty of the task. One potential solution could be to simplify the tasks into a set of primitives which have such a measure, and that can then be used as the basis for performance evaluation. Section IV-B outlines some of the ISO standard simplified tasks that can be adopted for

this purpose, while section IV-C discusses how these relate to typical assistive robot tasks.

B. Simplified Tasks

Since its original publication, Fitts' law [20] has been an important tool in modeling the speed/accuracy trade-off in simple human movements. As seen in (3a), the model predicts that the Mean Time (*MT*) to complete a movement varies linearly with the Index of Difficulty (*ID*). This index is a function of the distance moved and the accuracy requirement, or tolerance, on the movement. These are denoted as the distance of movement, *D*, and the width of the target area, *W*, respectively. The standard formulation for *ID* is shown in (3b).

$$MT = a + b \cdot ID \tag{3a}$$

$$ID_{Fitts} = log_2\left(\frac{D}{W} + 1\right) \tag{3b}$$

ID has units of *bits* and has roots in Information Theory. In fact the original interpretation of the law was as a measure of the information capacity of the human motor system. Typically the application of Fitts' law is in simple left to right movements of the hand. An example can be seen in Fig. 4. The coefficients a and b are determined experimentally using a linear regression analysis. The slope coefficient *b* then becomes a measure of the rate of change of completion time with change in the difficulty of the task. The reciprocal, *1/b*, is known as the Index of Performance (*IP*) and has units of *bits/second*. In other words, human performance for a task with a given distance and accuracy requirement can be predicted on the basis of observations of other such combinations.

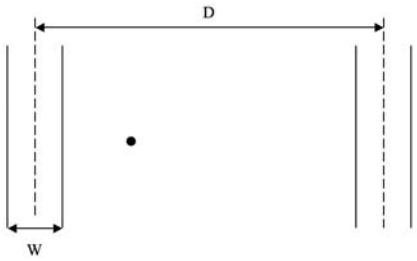


Fig. 4. Standard task for Fitts' law.

The version of Fitts' law used here was first proposed by [21] and is the basis for performance testing in ISO 9241-9 [22], which covers ergonomic requirements for non-keyboard computer input devices. Fitts' law has been used extensively in the field of Human-Computer Interaction (HCI) to quantify performance and drive graphical user interface design. The law is also commonly used in comparisons of input devices, where it provides the capability to generalize about results beyond a specific task. In fact, Fitts' law remains one of very few hard quantitative tools available to designers of human-machine interfaces, even though it is today considered more as an empirical regularity than as a model of the underlying mechanics of human movement. Multidimensional versions of the law have been developed, including pointing in 2D and 3D. Rotational movements based on Fitts' law tasks have also been explored.

$$\begin{aligned} &ID_{Steering} = \frac{d}{w}, \\ &where: \\ &w = k - b. \end{aligned} \tag{4}$$

Equation (4) shows the related steering law [23], which is an extension of the reciprocal or discrete movements in Fitts' law to continuous trajectories. By in essence applying an infinite number of Fitts' law task goals along a given trajectory, human performance for steering down 2D corridors of different shapes with computer input devices can successfully be modeled. See Fig. 5 for the standard straight task.

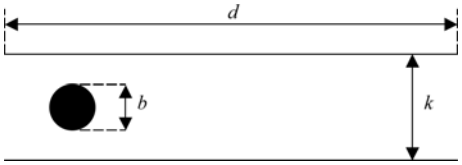


Fig. 5. Standard straight task for the steering law.

C. Relating Simplified Tasks to Robot Tasks

The simple models presented in the previous section has been used successfully in a wide range of simple tasks and for a wide range of users. However, it is not clear if they are also representative of the movements required of an assistive robot. The first issue that arises is whether a complex task like placing a can of soda in a kitchen cabinet can be approximately represented by a set of such movement primitives. As both Fitts' and the steering law are typically applied in planar environments, it is useful to begin the discussion with a planar simplified representations of an assistive task, see Fig. 6. Assuming there are relatively few items in the cabinet, the above task can probably be considered planar, as the out-of plane restrictions are relatively loose in comparison to those in-plane. An interpretation of the task could be that of an initial gross Fitts' law movement to the edge of the opening between two shelves (*A* to *B*). This would then be followed by a steering-based movement between the two shelves and possibly with a Fitts' law requirement for the final placement to avoid hitting the end-wall (*B* to *C*).

There are few guarantees that the time taken to perform a complex movement could be accurately modeled by such a decomposition into simpler movement primitives. However, even an approximate model is probably better than the alternative of comparing performance on tasks without any measure of the difficulty of the task. The better the model of the task the less variability not related to the task will be observed, reducing the effect of external factors on the results and increasing the amount of "control" in the experiment. Another potential benefit is that the difficulty of the task can be approximately quantified in units of information,

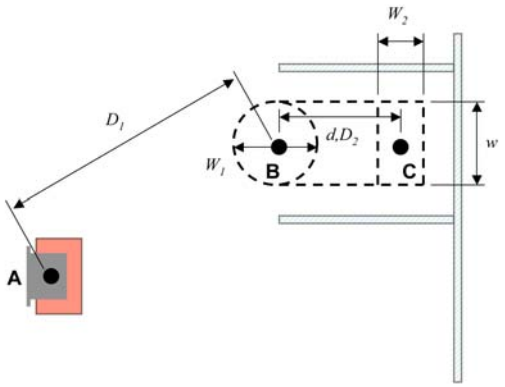


Fig. 6. Example of a planar, simplified representation of an assistive robot task; placing a can on a shelf in a cabinet.

bits, given the Information Theory roots of the two laws. The amount of information flowing between the human and the machine (*bits/second*) could then perhaps be said to be maximized when the user has minimized the time (in *seconds*) to complete the movement of a given number of *bits*.

V. CONCLUSIONS AND FUTURE WORKS

An attempt was made to identify methods for analyzing, implementing, and testing a future multimodal enabling interface for the ASIBOT robot. Information Theory concepts were found to be of interest in analyzing the human-machine system. For example, by defining the goal of the human-machine interface as transmitting the information representing the intention of the user over the noisy human-machine channel with a minimum of errors. The disability of the user was represented as the noise in the channel. Multimodality was identified as one of the potential approaches to achieve this, in particular synchronized and redundant signals from several input devices. A brief investigation into potential methods for implementing the requirements from the analysis was also performed. This included information fusion and learning methods. For the latter, supervised learning in the form of Gaussian processes was identified as a potential candidate. However, reinforcement learning is also of interest for its online applicability. The requirements for being able to test a multimodal enabling interface were also explored. It was speculated that using application-specific tasks for testing the system without a good measure of the difficulty of the task could make the results difficult to generalize upon. Leaning on experimental comparisons of computer input devices a suggestion was made for using well-proven experimental paradigms such as Fitts' law for the testing. It was speculated that more complex application-specific tasks could be approximately modeled by an appropriate set of such simple movement primitives. The future work will include implementation of a set of modules for achieving an enabling multimodal interface for ASIBOT and pilot studies to evaluate the methods highlighted and the assumptions made.

REFERENCES

[1] B. Dumas, D. Lalanne and S. Oviatt, "Multimodal Interfaces: A Survey of Principles, Models and Frameworks," *Human Machine Interaction: Research Results of the Mmj Program*, Springer, 2009.

[2] E. Rocon, J.M. Belda-Lois, A.F. Ruiz, M. Manto, J.C. Moreno, J.L. Pons, "Design and validation of a rehabilitation robotic exoskeleton for tremor assessment and suppression," *IEEE transactions on neural systems and rehabilitation engineering*, vol. 15(3), pp. 367-78, 2007.

[3] C. Balaguer, A. Giménez, A. Jardón, "The MATS robot: Service Climbing Robot for Personal Assistance," *IEEE Robotics & Automation Magazine*, vol. 13, no. 1, pp. 51-58, 2006.

[4] V.L. Vaquero, A. Jardón and C. Balaguer "Dispositivo Inalambrico para Facilitar el Acceso al Ordenador," *In Proceedings of International Congress on Domotics, Robotics and Remote-Assistance for All (DRT4All)*, 2009.

[5] H. Touchette and S. Lloyd, "Information-theoretic approach to the study of control systems," *Physica A: Statistical Mechanics and its Applications*, vol. 331, pp. 140-172, 2004.

[6] C. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, pp. 379-423, 1948.

[7] R. Chan and D. Childress, "On information transmission in human-machine systems: channel capacity and optimal filtering," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 20, pp. 1136-1145, 1990.

[8] S. Oviatt, "Ten Myths of Multimodal Interaction," *Communications of the ACM*, vol. 42, 1999.

[9] O. Tonet, M. Marinelli, L. Citi, P.M. Rossini, L. Rossini, G. Megali, and P. Dario, "Defining brain-machine interface applications by matching interface performance with device requirements," *Journal of neuroscience methods*, vol. 167, pp. 91-104, 2008.

[10] J. Kelso, D. Southard, and D. Goodman, "On the nature of human interlimb coordination," *Science*, vol. 203, pp. 1029-1031, 1979.

[11] S. Riek, J.R. Tresilian, M. Mon-Williams, V.L. Coppard, and R.G. Carson, "Bimanual aiming and overt attention: one law for two hands," *Experimental brain research*, vol. 153, pp. 59-75, 2003.

[12] D.L. Hall and J. Llinas, "Handbook of multisensor data fusion," CRC Press, Boca Raton, FL; 2001.

[13] K.P. Kording and D.M. Wolpert, "Bayesian integration in sensorimotor learning," *Nature*, vol. 427, pp. 244-7, 2004.

[14] K. Veluvolu, U. Tan, W. Latt, C. Shee and W. Ang, "Adaptive filtering of physiological tremor for real-time compensation," *2008 IEEE International Conference on Robotics and Biomimetics*, pp. 524-529, 2009.

[15] E. Demeester, A. Huntemann, D. Vanhooydonck, G. Vanacker, H. Brussel and M. Nuttin, "User-adapted plan recognition and user-adapted shared control: A bayesian approach to semi-autonomous wheelchair driving," *Journal of Autonomous Robots*, vol. 24, pp. 193-211, 2008.

[16] A. Huntemann, E. Demeester, M. Nuttin and H. Van Brussel, "Online user modeling with Gaussian Processes for Bayesian plan recognition during power-wheelchair steering," *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 285-292, 2008.

[17] R.S. Sutton and A.G. Barto, "Reinforcement learning: An introduction," The MIT Press, Cambridge, MA; 1998.

[18] Y. Engel, P. Szabo, and D. Volkinshtein, "Learning to control an octopus arm with gaussian process temporal difference methods," *Advances in neural information processing systems*, vol. 18, pp. 347-354, 2006.

[19] K. Tsui, H. Yanco, D. Kontak, and L. Beliveau, "Development and evaluation of a flexible interface for a wheelchair mounted robotic arm," *Proceedings of the 3rd international conference on Human robot interaction - HRI '08*, pp. 105-112, 2008.

[20] P.M. Fitts, "The Information Capacity of the human Motor System in Controlling the Amplitude of Movement," *Journal of Experimental Psychology*, vol. 47, pp. 381-391, 1954.

[21] I.S. MacKenzie, "A note on the information-theoretic basis for Fitts law," *Journal of Motor Behavior*, vol. 21, pp. 323-330, 1989.

[22] ISO 9241-9:2000(E), "Ergonomic requirements for office work with visual display terminals (VDTs) - Part 9 - Requirements for non-keyboard input devices," *International Organisation for Standardisation (ISO)*, vol. February 15, 2002.

[23] J. Accot and S. Zhai, "Beyond Fitts' law: models for trajectory-based HCI tasks," *Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM New York, NY, pp. 295-302, 1997.

Proceedings of the IEEE ICRA 2010
Workshop on Multimodal Human-Robot Interfaces
Anchorage, Alaska, May, 2010

Humanoid robot skill acquisition through balance interaction between human and humanoid robot

Jan Babič and Erhan Oztop

***Abstract*— Humanoid robots are intrinsically unstable mechanisms. To achieve a desirable full-body skill of the humanoid robots we propose a framework where we use the human demonstrators real-time action to control the humanoid robot and to sequentially build an appropriate mapping between the human and the humanoid robot. This approach requires the state of the humanoid-robot balance to be transfered to the human as the feedback information. Two example skills obtained by the proposed framework are described.**

I. INTRODUCTION

Human ability to imitate skills and tasks demonstrated by other humans is an important method of learning. If the robots were able to imitate human motions in the same fashion, acquisition of complex robot skills would become very straightforward. One could simply transfer the motion of a human demonstrator to a humanoid robot using a type of real-time motion capture system but due to the different dynamical properties of the humanoid robot and the human, the success of this approach largely depends on the ad-hoc mapping implemented by the researcher [1]. As the humanoid robots are intrinsically unstable mechanisms, such mapping would have to be a sort of full-body balance algorithm which would modify the desired motion of the human demonstrator to ensure the postural stability of the robot. Needless to say, the design of such algorithms is a very demanding task.

Here we propose an alternative approach where we use the human demonstrators real-time action to control the humanoid robot and to consecutively build an appropriate mapping between the human and the humanoid robot. This approach can be seen as a closed loop system where the demonstrator actively controls the humanoid robot motion in real time with the requirement that the robot stays balanced. This setup requires the state of the humanoid-robot balance to be transferred to the human as the feedback information. We implemented two different types of feedback.

The proposed closed-loop approach exploits the human capability of learning to use novel tools in order to obtain a motor controller for complex motor tasks [2], [3]. The robot that is controlled by the demonstrator can be considered as a tool such as a car or a computer mouse when one uses it for the first time. The construction of the motor controller is a two phase process. In the first phase a human demonstrator performs the desired task on the humanoid robot. In the second phase the obtained motions are acquired through machine learning to yield an independent motor controller. The two phases are shown on Fig. 1.

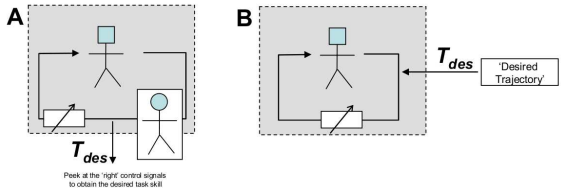


Fig. 1. (A) Human demonstrator controls the robot in closed loop and produces the desired trajectories for the target task. (B) These signals are used to synthesize a controller for the robot to perform this task autonomously.

Herewith we present two example skills that were obtained by the described framework.

II. FULL-BODY REACHING

The proposed approach can be considered as a closed loop approach where the human demonstrator is actively included in the main control loop as shown on Fig. 2. The motion of the human demonstrator was acquired by the contact-less motion capture system. The joint angles of the demonstrator were fed forward to the humanoid robot in real-time. In effect, the human acted as an adaptive component of the control system. During such control, a partial state of the robot needs to be fed back to the human subject. For statically balanced reaching skill, the feedback we used was the rendering of the position of the robot's centre of mass superimposed on the support polygon of the robot which was presented to the demonstrator by means of a graphical display. During the experiment the demonstrator did not see the humanoid robot.

The demonstrator's task was to keep the center of mass of the humanoid robot within the support polygon while performing the reaching movements as directed by the experimenter. With a short practice session the demonstrator was able to move his body and limbs with the constraint that the robot's center of mass was within the support polygon. Hence the robot was statically stable when the demonstrator generated motions were either imitated by the robot in real-time or played back later on the robot. The robot used in the study was Fujitsu HOAP-II small humanoid robot.

The motion of the humanoid robot was constrained to the two dimensions; only the vertical axis and the axis normal

This work has been suported by Japanese Society for Promotion of Science and Slovenian Ministry of Higher Education, Science and Technology

Jan Babič is with Jozef Stefan Institute, Ljubljana, Slovenia jan.babic@ijs.si

Erhan Oztop is with ATR Computational Neuroscience Laboratories, JST-ICORP Computational Brain Project, NICT Biological ICT Group, Japan erhan@atr.jp

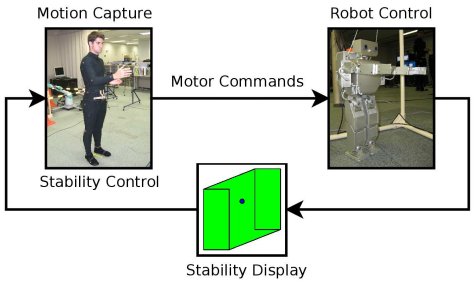


Fig. 2. Closed-loop control of the humanoid robot. Motion of the human is transfered to the robot while the robot's stability is presented to the human by a visual feedback.

to the trunk were considered. The light wiggly curve on Fig. 3 shows the robot end-effector position data which was generated by the demonstrator. One can imagine the humanoid robot from its left side standing with the tips of the feet at the centre of the coordinate frame and reaching out outwards with its right hand gliding over the curve. The long straight segment of the curve connects the beginning and the end of the reaching motion.

For each data point of the obtained end-effector trajectory, the robot joint angles were recorded. Assuming rows of the humanoid robot end-effector position $\mathbf{X}$ is formed by the data points taken from the obtained end-effector trajectory and the robot joint angles $\mathbf{Q}$ is formed by the corresponding joint angles we get a non-linear relation of the form

$$\mathbf{Q} = \Gamma(\mathbf{X})\,\mathbf{W}. \tag{1}$$

By performing a non-linear data fit and solving for $\mathbf{W}$ we can afterwards make prediction with

$$\boldsymbol{q}_{pred} = \Gamma(\boldsymbol{x}_{des})\,\mathbf{W} \tag{2}$$

where $\boldsymbol{q}_{pred}$ is a vector of the predicted joint angles and $\boldsymbol{x}_{des}$ is a vector of the desired end-effector position. Using the prediction we can afterwards ask the humanoid robot to reach out for a desired position without falling over.

For non-linear data fitting the recorded positions $\mathbf{X}$ are mapped into an N dimensional space using the Gaussian basis functions given by

$$\varphi_i(\boldsymbol{x}) = e^{\frac{\boldsymbol{x}-\mu_i}{\sigma^2}} \tag{3}$$

where μ_i and σ^2 are open parameters to be determined. Each row of $\mathbf{X}$ in converted into an N dimensional vector forming a data matrix

$$\mathbf{Z} = \Gamma(\mathbf{X}) = \begin{bmatrix} \varphi_1(\boldsymbol{x}_1) & \varphi_2(\boldsymbol{x}_1) & \dots & \varphi_N(\boldsymbol{x}_1) \\ \varphi_1(\boldsymbol{x}_2) & \varphi_2(\boldsymbol{x}_2) & \dots & \varphi_N(\boldsymbol{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \varphi_1(\boldsymbol{x}_m) & \varphi_2(\boldsymbol{x}_m) & \dots & \varphi_N(\boldsymbol{x}_m) \end{bmatrix}. \tag{4}$$

Assuming we have a linear relation between the rows of $\mathbf{Z}$ and $\mathbf{Q}$, we can solve (2) for $\mathbf{W}$ in the sense of the minimum least squares by

$$\mathbf{W} = \mathbf{X}^{+}\,\mathbf{Q} \tag{5}$$

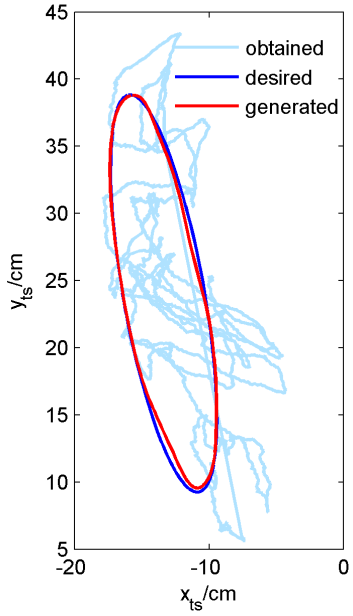


Fig. 3. The obtained end-effector trajectory generated by the demonstrator (light wiggly curve) with the desired end-effector trajectory that was used as the input for the joint angle prediction and the generated end-effector trajectory obtained by playing back the predicted joint angle trajectories on the humanoid robot.

where $\mathbf{X}^{+}$ represents the pseudo-inverse of $\mathbf{X}$. The residual error is given by

$$tr\left((\mathbf{XW}-\mathbf{Q})\,(\mathbf{XW}-\mathbf{Q})^{T}\right). \tag{6}$$

In effect, this establishes a non-linear data fit; given a desired end-effector position $\boldsymbol{x}$, the joint angles that would achieve this position are given by

$$\boldsymbol{q}_{pred} = \begin{pmatrix} \varphi_1(\boldsymbol{x}_{des}) & \varphi_2(\boldsymbol{x}_{des}) & \dots & \varphi_N(\boldsymbol{x}_{des}) \end{pmatrix}\mathbf{W}. \tag{7}$$

The open parameters are N as the number of basis functions which implicitly determines μ_i and the variance σ^2. They were determined using cross-validation. We prepared a Cartesian desired trajectory that was not a part of the recording data set and converted it into a joint trajectory with the current set values of (N, σ^2). The joint trajectory was simulated on a kinematical model of the humanoid robot producing an end-effector trajectory. The deviation of the resultant trajectory from the desired trajectory was used as a measure to choose the values of the open parameters.

Fig. 3 shows the desired end-effector trajectory and the generated end-effector trajectory obtained by playing back the predicted joint angle trajectories on the humanoid robot. The light wiggly curve on Fig. 3 represents the end-effector trajectory that was generated by the human demonstrator in the first phase and subsequently used to determine the mapping $\mathbf{W}$ between the joint angles and the end-effector position.

The reaching skill of the humanoid robot we obtained was statically stable which means that the robot's centre of mass was inside the robot's support polygon. However, when the robot was asked to track a trajectory at speeds

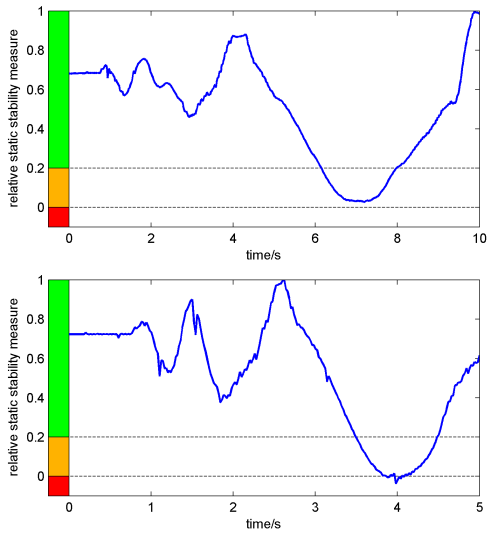


Fig. 4. The stability of the humanoid robot when the circular trajectory shown in Figure 3 is played back at different speeds. When the robot moved fast, the dynamics effects are no longer negligible as underlined by the ditch below 0 at around 4 seconds in the lower plot.

significantly higher than the speed of the demonstrator, the dynamics played a non-negligible effect. This can be seen from Fig. 4 where the upper plot shows the stability when the circular trajectory tracking was performed at $1/10\,Hz$. When the motion was performed at twice speed, the robot became unstable as shown in the lower panel of Fig. 4. The robot could still track the desired trajectory without falling over, but just barely.

A sequence of video frames representing the statically stable autonomous trajectory tracking obtained with our method is shown on Fig. 5.

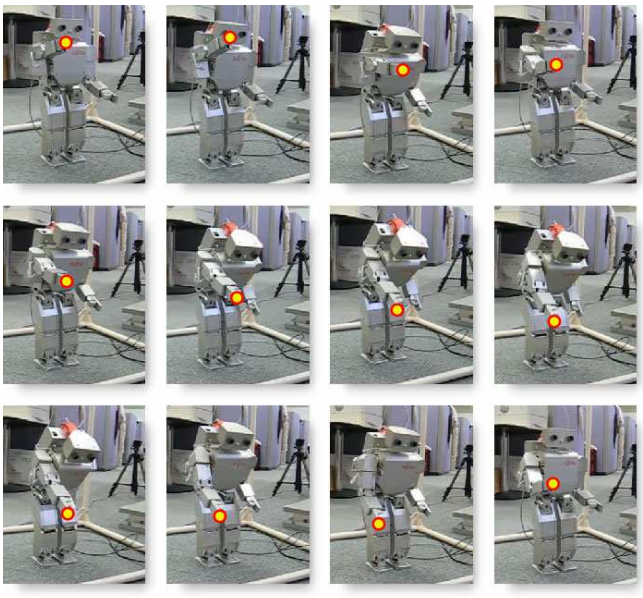


Fig. 5. Video frames representing the statically stable reaching motion of the humanoid robot obtained with the proposed approach.

III. IN-PLACE STEPPING

In this section, we present our preliminary work on performing a statically stable in-place stepping of the humanoid robot. In-place stepping is a task that requires an even stricter balance control than the reaching experiment described in the previous section. In order for the humanoid robot to lift one of its feet during the statically stable in-place stepping, the robot's centre of mass needs to be shifted to the opposite leg before the lifting action occurs. As the robot's centre of mass is relatively high and the foot is relatively small, it is crucial that the position of the centre of mass of the robot can be precisely controlled. For humans, to maintain the postural stability is a very intuitive task. If one perturbs the posture of a human, he/she can easily and without any concious effort move the body to counteract the posture perturbations and to stay in a balanced posture. The main principle of our approach is to use this natural capability of humans to maintain the postural stability of the humanoid robots. In order to do so, we designed and manufactured an inclining parallel platform on which a human demonstrator is standing during the closed-loop motion transfer (Fig. 6).

Instead of using visual information for the robot's stability as previously explained in the reaching experiment, the state of the humanoid robot's postural stability is feed-back to the human demonstrator by the inclining parallel platform. When the humanoid robot is statically stable, the platform stays in a horizontal position. On the contrary, when the centre of mass of the robot leaves its support polygon and therefore becomes statically unstable, the platform moves in a way that puts the human demonstrator standing on the platform in an unstable state that is directly comparable to the instability of the humanoid robot. The human demonstrator is forced to correct his/her balance by moving the body. Consecutively,

Fig. 6. Inclining parallel platform that can rotate around all three axes. The diameter of the platform is $0.7m$ and is able to carry an adult human.

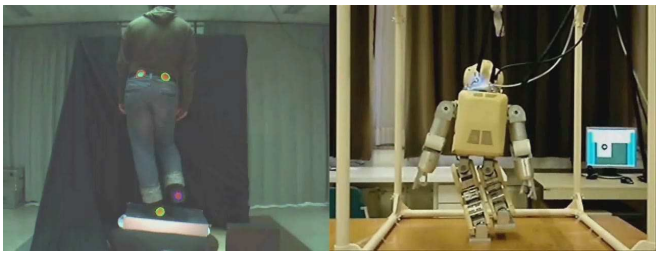


Fig. 7. The human demonstrator and Fujitsu Hoap-3 humanoid robot are shown during the in-place stepping experiment. The video frame on the left side shows the human demonstrator performing in-place stepping on the inclining parallel platform. The right side frame shows the humanoid robot during the one foot posture.

as the motion of the human demonstrator is fed-forward to the humanoid robot in real-time, the humanoid robot gets back to the stable posture together with the demonstrator. Using some practice, human demonstrators easily learned how to perform in-place stepping on the humanoid robot. The obtained trajectories can afterwards be used to autonomously control the in-place stepping of the humanoid robot. Our future plans are to extend this approach and use it for acquiring walking of the humanoid robots. Fig. 7 shows the human demonstrator and Fujitsu Hoap-3 humanoid robot during the in-place stepping experiment.

IV. CONCLUSIONS

A goal of imitation of motion from demonstration is to remove the burden of robot programming from the experts by letting non-experts to teach robots. The most basic method to transfer a certain motion from a demonstrator to a robot would be to directly copy the motor commands of the demonstrator to the robot [4] This approach proved to be very efficient for certain open-loop tasks. However, this simple approach is generally not possible to implement. Either the motor commands may not be available to the robot or the differences between the demonstrator's body and the robot are so big that a direct transfer of motor commands is not possible. One way of solving this problem is to modify the motor commands produced by the demonstrator with a sort of a local controller. The situation in our approach is different because the correct motor commands for the robot are produced by the human demonstrator. For this convenience, the price one has to pay is the necessity of training to control the robot to achieve the desired action. Basically, instead of expert robot programming our method relies on human visuo-motor learning ability to produce the appropriate motor commands on the robot, which can be played back later or used to obtain controllers through machine learning methods as in our case of reaching.

The main result of our study is the establishment of the methods to synthesize the robot motion using human visuo-motor learning. To demonstrate the effectiveness of the proposed approach, statically stable reaching and in-place stepping was implemented on a humanoid robot using the introduced paradigm.

V. ACKNOWLEDGMENTS

The authors gratefully acknowledge Blaž Hajdinjak for carrying out the in-place experiment.

REFERENCES

[1] S. Schaal, Is imitation learning the route to humanoid robots?, *Trends in Cognitive Science*, vol. 3, 1999, pp 233-242.

[2] E. Oztop, L.-H. Lin, M. Kawato, G. Cheng, "Dexterous Skills Transfer by Extending Human Body Schema to a Robotic Hand", *in IEEE-RAS International Conference on Humanoid Robots*, Genova, Italy, 2006.

[3] G. Goldenberg, S. Hagmann, Tool use and mechanical problem solving in apraxia, *Neuropsychology*, vol. 36, 1998, pp 581-589

[4] C.G. Atkeson, J.G. Hale, F.E. Pollick, M. Riley, S. Kotosaka, S. Schaal, T. Shibata, G. Tevatia, A. Ude, S. Vijayakumar, M. Kawato, Using Humanoid Robots to Study Human Behavior, *IEEE Intelligent Systems*. vol. 15, 2000, pp 45-56.

Proceedings of the IEEE ICRA 2010
Workshop on Multimodal Human-Robot Interfaces
Anchorage, Alaska, May, 2010

Robot Reinforcement Learning using EEG-based reward signals

I. Iturrate, L. Montesano, J. Minguez

Abstract— **Recent works suggest that several human cognitive processes elicited during the observation and monitoring of tasks developed by others could be used for robot task learning. These works have demonstrated that human brain activity can be used as a reward signal to teach a simulated robot how to perform a given task in very controlled situations. The open question is whether this activity is potentially usable in a real robot context. This paper gives evidence that: (a) a brain discriminative response is also elicited during the observation of a correct/incorrect operation of a real robot, (b) this response is consistent along different subjects, (c) it is possible to learn a classifier that provides online categorization with enough accuracy (between 85-90%) to implement simple reinforcement learning algorithms based on these signals. Experimental results have been obtained with 3 subjects observing the operation of a 5 dof robotic arm performing correct/incorrect reaching tasks, while an EEG system recorded their brain activity. The presence of these brain patterns during the observation of real robot operation opens the possibility to use human brain activity for developing learning robots able to adapt themselves to the task and user's preferences based on the implicit judgment of the task made by the human.**

I. Introduction

Robot learning has recently emerged as a paradigm where robots acquire new skills during operation, and improve their performance with experience. One of the most common frameworks is reinforcement learning methods (RL) [1], which have been successfully applied to learn motor behaviors and motion primitives among many other skills [2]. RL methods are based on an optimization process to compute a policy that maximizes the long-term reward while acting in the environment. This is done through an iterative process where the robot executes a sequence of actions, receives the corresponding reward signal and uses it to improve the current estimate of the policy.

The reward signal encodes the degree of accomplishment of the executed action. In practice, the engineer has to define this reward and, in some situations, build an ad-hoc system (e.g. a tracking system) to compute it or supervise the task to manually provide it. There are two shortcomings with this approach. The first one is that the programmer has to define the reward signal and devise a way to measure it, what can be challenging when the evaluation of the robot operation is subjective or when dealing with very complex systems. The second one is that the skills learned by the robot are the result of the programmer's experience, while in human-centered robotics the programmer is not the final user. Under this last condition, the operation of the robot needs to adapt its operation to the user preferences and not to engineer's ones.

An alternative approach to achieve this end user adaptation is to use brain activity to capture the final-user perception of the robot operation and compute the reward based on this activity. The advantage of this new concept is: (a) the definition of the correct/incorrect operation is provided directly by the human understanding of the task. Furthermore, it occurs in a transparent manner even in complex systems where it would be difficult to model the reward; (b) the system exploits the activation of the areas related to monitoring and error processing and, therefore, is in theory scalable and applicable to a variety of different robot tasks; and (c) this robot learning concept captures task subjective aspects that depend on each user, which is a firm step towards the individualized operation (human-centered robotics). The work in [3] demonstrated that human brain activity can be used as a reward signal to teach a simulated robot how to perform a given task in very controlled situations. However, the open question is whether this framework is potentially usable in a real robot context, that is, if the error mechanisms of the brain are also elicited by observing a real robot operation. This paper addresses this crucial question for the applicability of this type of techniques within the field of robotics.

Based on an experiment with a real robot, this paper provides evidence that: (a) the brain areas that play a role in detection and monitoring of errors also play a role when observing the operation of a real robot; (b) a brain discriminative response is elicited during the observation of a correct/incorrect operation of a real robot, (c) this response is consistent along different subjects, (d) it is possible to learn a classifier that provides online categorization with enough accuracy to implement simple reinforcement learning algorithms based on these signals.

The remainder of the paper is organized as follows. Next section discusses related work from neurophsysiopsychology, brain-computer interfaces and robotics. In Section III, we describe the experiment that allow us to record the EEG data. Section IV analyzes the data from a neurophysiologyical perspective to evaluate the presence of error-related brain responses. Section V and Section VI present the signal processing and machine learning algorithms used to classify the signal and the results obtained from the experiment dataset. Section VII demonstrates the applicability of these results to a real robot learning task. In Section VIII, we

Iñaki Iturrate, Luis Montesano and Javier Mínguez are with the Instituto de Investigación en ingeniería de Aragón (I3A) and Dpto. de Informática e Ingeniería de Sistemas (DIIS), Universidad de Zaragoza, Spain. E-mail: iturrate@unizar.es, montesano@unizar.es and jminguez@unizar.es.
This work has been partially supported by projects HYPER-CSD2009-00067, DPI2009-14732-C02-01 funded by the Spanish Government and the Portuguese FCT project PTDC/EEA-ACR/70174/2006.

draw the conclusions and comment on future developments.

II. Related Work

This paper studies the applicability of human cognitive processes related to the error monitoring to robot learning. This study has three main axes. The first one is to address the neurophysiological and cognitive mechanisms underlying the human detection and monitoring of errors. The second one is the application of these principles within a brain-computer interface framework, which integrates online signal processing and machine learning for the detection of these brain processes. The third one is to show the applicability of this error recognition for a robot reinforcement learning task. We next discuss related work on each of these three axes.

Roughly speaking, there are two types of brain electrical activity. The Event-Related Potentials (ERP) are signals that are elicited by means of an internal or external event, while the rest of the activity is usually referred to spontaneous rhythms. In cognitive neuroscience and neuropsychology, it is well known the usage of the ERP to study the underlying mechanisms of the human error processing, sometimes referred to Error-related Potentials (ErrPs) [4]. This is because the observation/execution of an incorrect action for the user is the event that triggers a particular activity or potential, which codifies the cognitive error information when the human is expecting a given outcome or performance.

Different ErrPs have been described, for instance, when a subject performs a choice reaction task under time pressure and realizes that he has committed an error [5] (response ErrPs); when the subject perceives an error committed by another person (observation ErrPs) [6]; when the subject delivers an order and the machine executes another one [4] (interaction ErrP); and recently when the subject perceived an error committed by a simulated robot [3]. These error-related processes involve the activation of a specific area of the brain, called anterior cingulate cortex (ACC), Brodmann areas[1] 24, 32 and 33. Evidence for the role of the ACC as been involved in the error detection process comes from consistent observations of error potentials uniquely generated within the ACC upon error occurrences [8]. One of the objectives of this paper is to give evidence that the observation of the operation of a real robot involves the activation of the ACC during the error pontentials associated to the robot erroneous operations.

The next objective is to build a real-time system to measure, identify and use this error information, usually known as Brain-Computer Interface (BCI). These systems acquire the brain activity and convert it into external actions or signals that can be used to perform several tasks. Usually, the signal is recorded with a non-invasive method called electroencephalography (EEG), which uses several sensors placed on the scalp. EEG-based BCIs have succesfully been used in communication tasks such as a speller [9], to move an arm prosthesis [10], or to drive a robotic wheelchair [11], [12]. The key of the success of these systems is to determine the appropriate neurophysiological response that can be identified and used to achieve a particular goal. The nature of the EEG measurements (noise, artifacts, poor spatial resolution, inter-subject variation) makes this a difficult task for most of the brain processes and requires the use of signal processing and machine learning algorithms. In the context of this paper several works have shown that it is possible to perform automatic single trial classification of several of the error-related potentials mentioned above [4], [13]. In this paper, the second objective is to show that it is possible to learn a classifier that provides online categorization of the errors potentials elicited during the observation of the robot with enough accuracy (i.e. it is feasible to build a BCI that discriminates online the robot operation).

Finally, the last relevant aspect of this work is the application of these potentials to develop adaptive systems. Up to our knowledge, there are only two works that have addressed this problem and both of them did it in simulation. The work in [14] designed a two-actions scenario where a cursor moved right or left towards a target. Interaction potentials were detected online and used to modify the probability of each action. In particular, the probability of an action was increased (decreased) when a correct (wrong) action was detected. In [3], the authors proposed the use of online error-related potentials as a reward signal for a Q-learning RL algoritm. The setup of this experiment is somehow more interesting for robotics, since the human was observing a simulated robot arm performing a discrete number of actions. Their results suggest that there may be information within the EEG measurements to differentiate more subtle aspects such as the laterality and degree of errors. Unfortunately, as the authors pointed out, their analysis was limited due to the presence of artifacts that hinder the evaluation of the activation areas involved in the process and limited the temporal window that could be used for automatic classification. This paper makes a step forward and shows that this type of activity is also present when observing a real robot and that can be automatically detected and used in a reinforcement learning context.

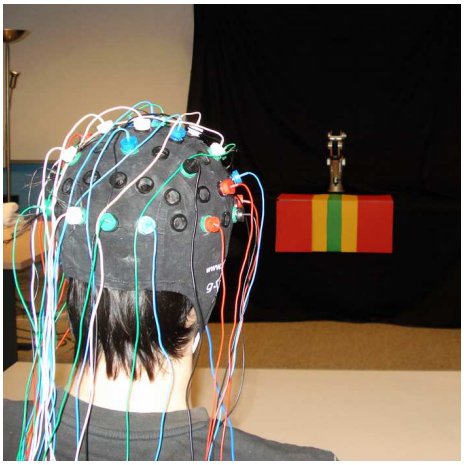
III. Protocol and Design of the Experiment

This section describes the design of the main experiment of the paper. The objective is to collect the EEG to determine: (a) if a specific brain potential is elicited during the observation of a correct/incorrect operation of a real robot, and if this response is consistent among different subjects[2]; and (b) if it is possible to learn a classifier that provides online categorization with enough accuracy, to evaluate the feasibility of a online brain-computer interface.

In the experiment, it was used a Katana300 robot arm with 5 degrees of freedom. The instrumentation used to record

[1]The brain cortex can be divided in areas or regions defined according to its cytoarchitecture (the neurons' organization in the cortex). These zones are called Brodmann Areas (BA) [7], and are numerated from 1 to 52.

[2]Notice that the objective here is not to characterize the Event-Related Potential as it is usually performed Neuropsychology, but to provide evidence that the potential exists and that is consistent for all the participants of the study.



(a)

(b)

Fig. 1. (a) General view of the set up. The subject observes the robotic arm motion while the EEG system records the brain activity. (b) The robot arm performs consecutive reaching tasks to five predefined positions, which are colored in green (correct position), yellow (small incorrect position), and red (large incorrect position).

the EEG brain activity was a gTec system (an EEG cap, 32 electrodes, and a gUSBamp amplifier). The location of the electrodes was selected following previous ErrP studies [15], [3] at FP1, FP2, F7, F8, F3, F4, T7, T8, C3, C4, P7, P8, P3, P4, O1, O2, AF3, AF4, FC5, FC6, FC1, FC2, CP5, CP6, CP1, CP2, Fz, FCz, Cz, CPz, Pz and Oz (according to the international 10/20 system). The ground electrode was positioned on the forehead (position FPz) and the reference electrode was placed on the right earlobe. The EEG was amplified, digitized with a sampling frequency of 256 Hz, and power-line notch-filtered and bandpass-filtered between 0.5 and 10 Hz. As usually done in this type of recordings, a Common Average Reference (CAR) Filter was applied to remove any offset component detected on the signal. The signal recording and processing and the synchronization between the robot arm and the EEG were developed under BCI2000 platform [16].

The general setting of the experiment was a user observing the operation of a robot arm while the EEG was recorded (Figure 1a). The robot continuously operated by developing reaching tasks to five predefined positions (Figure 1b). The participants were instructed to judge the robot motion as follows: (a) a motion towards the center was a correct operation, (b) a motion towards the locations placed just on the side (left or right) of the center was small operation error, and (c) a motion towards the furthest locations from the center (left or right) was a large operation error. The reaching positions were marked with colors to facilitate the participants the identification of the operations, where green was the correct movement, yellow the small operation errors, and red the large operation errors. Notice that this experimental design includes different functional operations (error-correct), different degrees of error (small-large), and different laterality of error (left-right).

Three male, right-handed, 24-aged persons selected from the research team participated in the experiments. The participants were informed about the experiment. Furthermore, they were instructed to avoid as much as possible any muscular movement (artifacts) to avoid the contamination of the EEG. One subtle but important artifact to avoid, as mentioned in previous works [3], was the lateral eye movement. This is because this artifact is very prominent in the EEG of frontal and fronto-central areas, and could lead to erroneous conclusions about the laterality of the EEG potentials. The protocol was adapted to minimize the motion of the eyes by placing the robot arm far enough from the subject (4 meters), so that the participants did not need to significantly move their eyes to observe the final position of the robot. Notice that, although there is literature related to the automatic filtering of these artifacts [17], at this stage of the research is always better to avoid its occurrence (rather than to rely in filtering techniques that could eliminate important aspects of the brain potential).

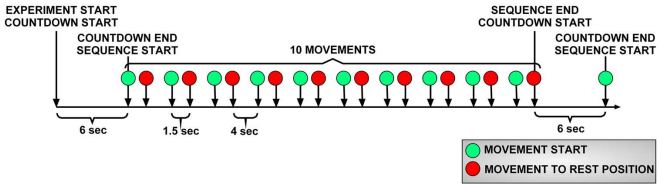


Fig. 2. Temporal diagram of a sequence of robot actions.

For each participant, an experiment consisted of 10 trials of 5 sequences each, where each sequence was composed by 10 random reaching actions carried out by the robot arm. A total number of 500 operations were executed. Each sequence was designed as follows (Figure 2): firstly there was a 6 seconds countdown with auditory signals associated (to inform the subject that the sequence was starting) and then ten random actions were executed by the robot. The reaching action lasted 1.5 seconds (the effective motion was between 0.8 and 1.1 seconds). The returning to the initial position was 4 seconds (the effective motion was between 0.8 and 1.1 seconds) providing the participants some time to relax between robot motions. The total time of the experiment was 51 minutes plus approximately 5

minutes of breaks distributed between trials (depending on the user). The experiment was designed in such a way that the 500 operations were equally distributed as 100 times per possible action. Then, 100 brain responses of each action were recorded, which is the typical amount of samples used in ERP literature to have a good signal to noise ratio using grand averages techniques to study the responses [17].

IV. Neuro-physiologyical analysis

After recording the EEG data, the first step is to characterize the brain response as a possible ERP. The analysis was developed as follows. Firstly, the averaged ERP potentials were constructed, which are simply the averaged sum of the individual responses for each condition at each electrode (to improve the signal-to-noise ratio and, as a consequence, filter background noise and occasional artifacts). The averaged ERP were computed for the three participants for three different cases: (i) error versus correct responses, (ii) left versus right errors and (iii) small versus large errors. Next, a statistical analysis (ANalysis Of Variance, ANOVA test) was performed for all the ERPs of the three cases, with a significance level of 95% ($p < 0.05$). Finally, in order to speculate about the brain areas involved in the generation of the potentials, an EEG Source Localization technique was used. Concretely, we used sLORETA [18]. This type of techniques estimates the neural generators within the brain given the EEG at the surface of the scalp. Figure 3 shows the results of the averaged ERPs and the statistical analysis in the Cz electrode (usually selected to display error-related potentials), and the result of the source localization technique.

The first observation is that the averaged ERPs resulting from the robot operation correct/incorrect are different, which implies that in mean, there are different brain processes involved. Secondly, the difference of the ERP average correct and incorrect reflects a large negativity around 400 milliseconds with great statistical difference. This result agrees with the previous works that describe a negativity around this timing in error observation tasks [4]. Thirdly, the shape of the response in Cz elicited in the incorrect operations is similar to the response of other protocols that involve the human monitoring of errors, concretely the interaction errors (see [4] for some examples): they have a sharp positive potential at around 0.3 seconds, followed by a wide prominent negativity around 0.4 second. Fourthly, the main active areas at the time of the prominent negativity of the difference signal (error minus correct response average) were Brodmann 6, 24 and 31 (Figure 3). These areas are in the close neighborhood of the ACC, which is the brain area involved in the error processing. Furthermore, this finding agrees with several results that obtained the same areas in the most prominent negativity in reaction, observation and interaction errors [5], [6], [4]. Their hypothesis is that these associative areas (somatosensory association cortex) could be related to the fact that the subject becomes aware of the error. All these results push forward the hypothesis that a discriminative (correct/incorrect) Event-Related Potential is elicited during the human monitoring of the robot operation, which belongs to the family of error-related potentials.

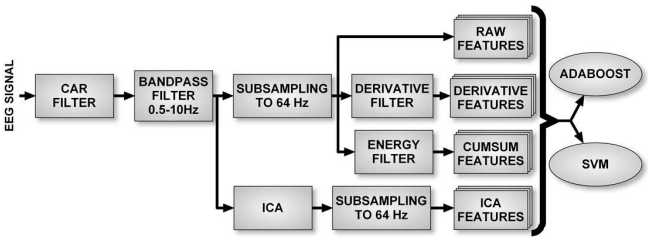


Fig. 4. General classification scheme. The measured EEG signal undergoes a set of different filters before feeding the classifier. Four different types of features were used to train the classifier: raw signal, derivative of the signal, cummulative energy or the M components obtained from ICA. Two different classifiers were compared, one based on Boosting and the other one using Support Vector Machines.

Finally, another interesting result is the ANOVA test between small and large errors. As opposed to the other ANOVA comparisons, where the statistical difference was mostly focused on the 300ms-800ms, this ANOVA showed that the statistical difference was mainly in the 800-900ms time range. This could be due to the fact that the movement is not instantaneous. Thus, the user starts understanding the fact that an error occurred almost at the start of the movement because of the initial turn of the robot to reach the error positions. However, the differences in small and large errors were only present almost at the end of the movement, which occured between 800-900ms.

V. Classification of EEG activity

The previous section shows that there exists a discriminative (correct/incorrect) response in the human brain elicited during the observation of the robot operation. The next objective is to develop a single trial classification of these processes to use it as feedback, for instance, for robot learning or robot supervision. The main difficulty here is that, despite on average the different conditions of the ERPs look very different, single EEG measurements are very noisy and this classification becomes challenging. This section describes the techniques used to obtain an automatic classification.

Let $\mathbf{x}_t \in \mathcal{R}^N$ denote the EEG signal at time t where N is the number of EEG channels recorded. For a given robot motion, the EEG signal is a sequence of measurements $\mathbf{x}_{1:T}$ over a fixed window. During the analysis of the signal, the ANOVA test was used to evaluate the statistical difference among the different conditions. The results of this analysis were used to reduce the temporal window $[1..T]$ for classification to those intervals where the ANOVA test found significant differences.

The classification process is composed of two different phases as illustrated in Figure 4. The first one is the computation of the features that will be used by the classifier. Recall from previous section that, during acquisition, the EEG signal has already been processed to remove the offset (CAR-filter) and to keep those frequencies relevant to the ERP (0.5-10Hz). Previous studies have shown that, for ERPs,

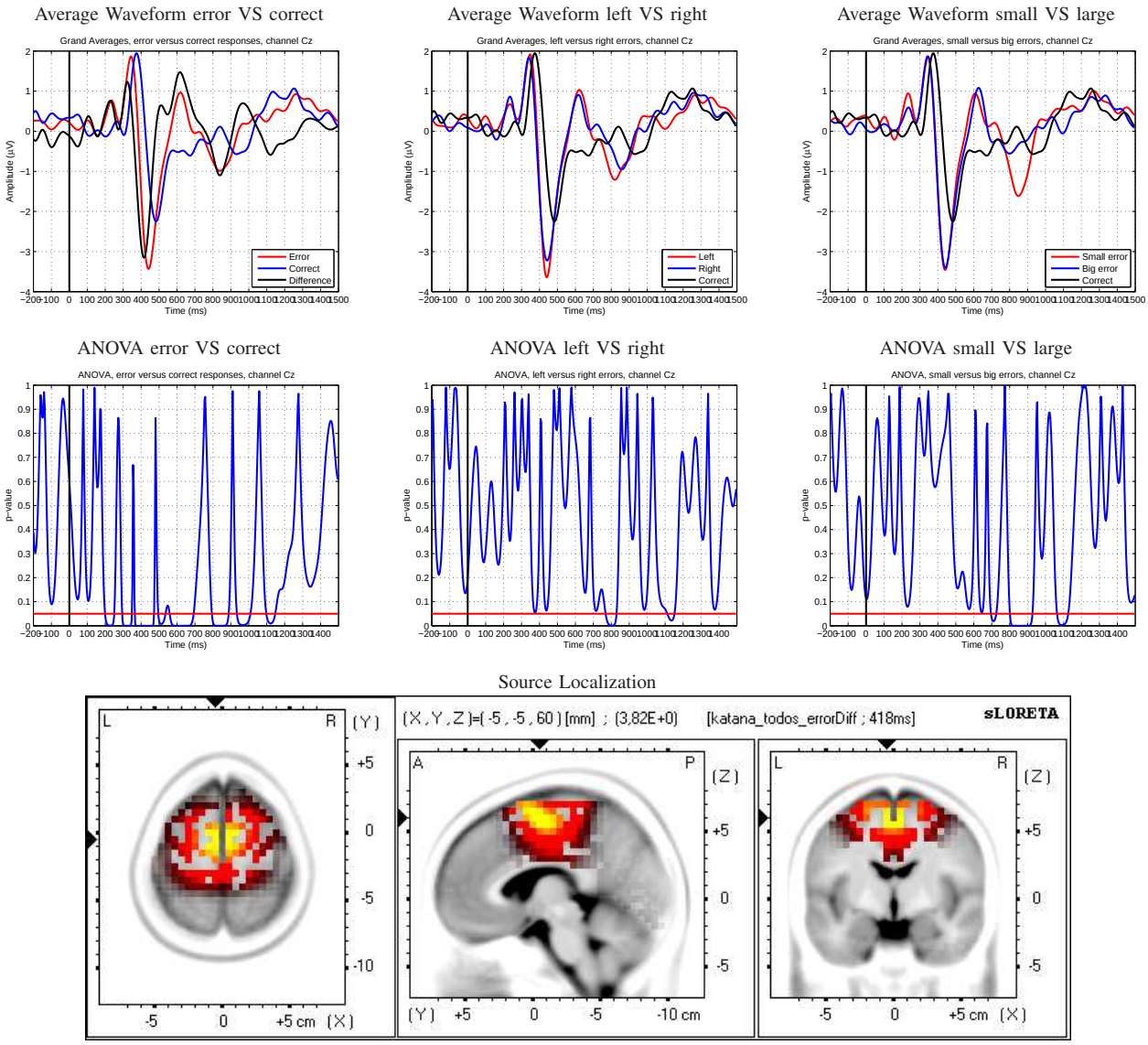


Fig. 3. Average waveforms (top), ANOVA analysis (middle) and Source Localization at $400ms$ (bottom) averaged over all the participants in channel Cz. The average waveforms clearly indicate that an ERP was elicited. A baseline of 200 ms before the movement started is also shown. The robot motion occured between the 0-1500 ms. For the ANOVA figure, the vertical axis correspond to the p-values at each instant of time for the same time window as before. The horizontal red line shows the p-value of 0.05. The source localization shows the activity of the brain. The figure is better understood in color where yellow areas indicates those areas with high activity.

the frequency domain does not contain enough information to perform the classification [19]. Therefore, we focus on the analysis of temporal features that have been used in the literature as well as some designed from our own analysis of the signal. In all cases, the features are computed from a $64Hz$ subsampled version of the signal to reduce the computational cost (we have experimentally verified that this subsampling does not affect the classification accuracy). The following characteristics were computed:

- RAW signal: In this case, the feature vector is the concatenation of all the $\mathbf{x}_t$ with t within the window defined by the ANOVA analysis of the signal.
- First derivative: The ERPs are usually described by their number of components and their value (positive or negative). The derivative is a natural way to obtain a description of the (positive and negative) modes of the signal. The discrete derivative vector is computed as $\dot{\mathbf{x}}_t = \frac{\mathbf{x}_{t+1} - \mathbf{x}_t}{\Delta T}$ over the same window as the RAW signal. The classifier receives as feature vector the concatenation of all the derivatives.
- Cumulative energy of the signal: In the grand averages, a characteristic of the error ERPs is that the error signal has more energy than the correct responses. We explore this idea by approximating the cumulative energy of the signal over the temporal window of channel i, where $e_{it} = \sum_{j=1}^{t} |x_{ij}|^2, \quad \forall t \in [0..T]$. The vector $\mathbf{e}_t = (e_{1t}...e_{Nt})^T$ contains the cumulative energy of each channel at time t. As in the previous cases, the window is set according to the ANOVA results and the concatenation of all $\mathbf{e}_t$ is used as the feature vector for the classifier.

- The Independent Components: Independent Component Analysis [20] (ICA) is a spatial-filtering technique whose aim is to retrieve unobserved signals (the so-called Components) from observed mixtures (in our case, N channels), using the mutual independence among signals as a fundamental initial assumption. The use of ICA is motivated by the fact that the different conditions may trigger different types of brain activity that can be tracked independently. We used the Maximum Likelihood method [20], to compute M Independent Components from the EEG measurements without any subsampling. The method also carried out a dimensionality reduction with Principal Component Analysis (PCA). The resulting demixing matrix transforms the original signal $\mathbf{x}_t$ to a $\mathbf{y}_t \in \mathcal{R}^M$. Finally, we performed an ANOVA test on each component so as to verify the existence of components with statistical difference among the different conditions. In addition to this, the ANOVA test also allowed us to fix the temporal window as we have done with the other types of features.

The previous features were used to train and compare the performance of two different state-of-the-art classifiers used in the ERPs literature:

- AdaBoost: A meta-classifier that combines several weak classifiers and iteratively assings weights to them [21]. As weak classifier, we chose the Functional Decision Tree [22] due to the multi-variate nature of the EEG data. Functional trees allow to use linear combinations of attributes. This combination of boosting and functional decision trees was successfully used in the past for classifying ERPs [3].
- Support Vector Machines: The main idea of this family of classifiers is to search for the maximum hyperplane separation of classes in a self-constructed kernel space, obtained applying a non-linear function (usually called kernel) to the initial feature space [23]. As the previous classifier, this one was succesfully used in the past for classifying ERPs [13]. Among all the possible versions of SVM, we used the ν-SVM classifier with a radial basis function kernel.

VI. Classification Results

This section presents the classification results obtained with the dataset acquired according to the protocol described in Section III, and a comparison of the four different types of features and the two classifiers described in Section V. For a fair comparison, we tried to tune the parameters of each classifier and feature. However, the best performance was obtained in all cases with very similar parameters. The configuration for each classifier was:

- AdaBoost: we used the Weka implementation [24]. The number of iterations for the Functional Tree classifier and the boosting was 10 and 3, respectively.
- SVM: we used the libSVM implementation [25]. The ν parameter was set to 0.5 and the γ parameter of the radial basis function was set to $\frac{1}{\sharp features}$.

The EEG time window (common for all the participants) was $[0.3...0.9]$ seconds for the RAW, derivative and energy features, resulting in a total of 1248, 1216 and 1248 features per robot operation, respectively. For the ICA features, the best results were obtained with 3 independent components. In this case, the time window provided by the ANOVA analysis of the components varied from one participant to another, being the final number of features between 50 and 70. In order to minimize the overfitting effect, we used a ten-fold cross-validation strategy to train the classifier. Artifacts were not removed prior to the classification so as to have more realistic data.

There were two classification tasks: error versus correct responses, and a five-class task (left-large error, left-small error, correct, right-small error, right-large error). For the first task the data is strongly unbalanced (we have four times more errors than correct responses). Thus, we duplicated the correct responses to balance the dataset, having a total of 400 errors versus 400 correct responses. For the five-class case, the dataset was balanced and contains 100 examples per case.

We discuss first the results for the two-class case (error versus correct responses). Table I shows the average for all the participants of the recognition performance for each class, feature and classifier. The results illustrate that both classifiers achieved good classification rates (always $> 75\%$), where the AdaBoost always had better performances (always $> 80\%$). The best features were raw data for both the AdaBoost classifier and for the SVM (both $> 90\%$). In the latter case, the derivative features were better for the correct class and worse for the error one. The results show that ICA was unable to separate very discriminative components. On the other hand, the results using only 3 components point out that that the actual dimensionality of the data can be reduced through PCA without compromising much of the efficiency.

TABLE II
SUBJECT AVERAGE CLASSIFIER-FEATURES ACCURACIES COMPARISON: 5 CLASSES

	RAW	Derivative	Energy	ICA
AdaBoost	46.27%	38.73%	26.00%	33.53%
SVM	51.07%	42.20%	26.33%	29.73%

The average classification results for all the participants in the five-class case are shown in Table II (average of correct detection over all the classes for each pair of feature and classifier). The best combination is the SVM with raw data followed by AdaBoost with raw data too. As in the previous case, the other features provided systematically worse results. The best classification rate is on average 51.07%.

The performance of the classifier for the five-class case, despite being better than chance, degrades significantly with respect to the two-class one. However, let us discuss the confusion matrix for each participant (see Tables III, IV, and V for the SVM classifier[3]). We have labeled the different classes described in Section III as follows: left-2 (large left error), left-1 (small left error), correct, right-1 (small right

[3]The conclusion is also valid for the AdaBoost classifier.

TABLE I

SUBJECT AVERAGE OF CLASSIFIER-FEATURES ACCURACIES: ERROR VS CORRECT

	RAW		Derivative		Energy		ICA	
	Error	Correct	Error	Correct	Error	Correct	Error	Correct
AdaBoost	91.42%	100.00%	90.58%	100.00%	80.00%	98.00%	85.58%	100.00%
SVM	89.25%	95.08%	87.25%	96.33%	76.50%	88.58%	76.50%	84.17%

error) and right-2 (large right error). The matrices show that misclassifications tend to group by blocks, that is, right errors tend to be confused among them and left errors among them. This fact reinforces the idea that the error signal may have some laterality component as pointed out in previous works. The same effect appears between large and small errors. Although in this case the differences are smaller and vary a bit more among participants, one can verify that small errors tend to be misclassified more with small errors of different side than with large ones. For example, right-2 is misclassified more often as left-2 than as left-1 for every participant.

TABLE III

PARTICIPANT 1: SVM CLASSIFIER ACCURACY WITH FIVE CLASSES

	Left-2	Left-1	Correct	Right-1	Right-2
Left-2	42%	21%	7%	12%	18%
Left-1	22%	47%	7%	13%	11%
Correct	5%	1%	86%	4%	4%
Right-1	10%	12%	10%	46%	22%
Right-2	13%	11%	5%	26%	45%

TABLE IV

PARTICIPANT 2: SVM CLASSIFIER ACCURACY WITH FIVE CLASSES

	Left-2	Left-1	Correct	Right-1	Right-2
Left-2	37%	21%	8%	14%	20%
Left-1	19%	47%	6%	19%	9%
Correct	6%	6%	77%	5%	6%
Right-1	10%	20%	12%	41%	17%
Right-2	16%	13%	12%	19%	40%

TABLE V

PARTICIPANT 3: SVM CLASSIFIER ACCURACY WITH FIVE CLASSES

	Left-2	Left-1	Correct	Right-1	Right-2
Left-2	45%	19%	4%	14%	18%
Left-1	20%	42%	5%	22%	11%
Correct	3%	2%	85%	5%	5%
Right-1	15%	18%	4%	45%	18%
Right-2	18%	8%	7%	26%	41%

Finally, we have studied the number of trials needed for good classification performance. This is important for real applications, since the EEG data acquisition is a consuming and tiring process. The analysis was performed selecting sets as follows: using the first (in time) 10% of the data as train set and the 90% of the last (in time) data as the test set was labeled as 10%-90%. We performed this comparison for the two classification tasks, for the cases 10%-90%, 20%-80%, 30%-70%, 40%-60%, 50%-50%, 60%-40%, 70%-30%, 80%-20% and 90%-10%. Figure 5 shows the recognition rate for each class averaged over the three participants using the SVM classifier. The results show that for the two-class problem, the recognition rate reached a stable value with 60 examples (around 35 minutes of data collection). On the five-class task, the behavior is different and it seems that 100 examples is not enough and that we could improve the results using more data.

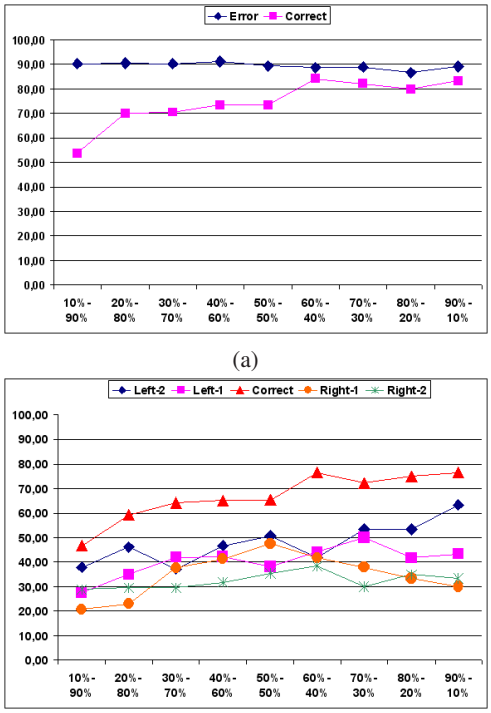


Fig. 5. Detection rates with different percentages of data of train and test sets. (a) two-class and (b) five-class tasks.

Summarizing, we have shown that it is possible to distinguish between error and correct robot operations with a high accuracy (over 90%). For the five-class problem, the performance is not as good, but the confusion matrix has a structure that can be used to obtain laterality and magnitude information. Among all the features, raw data provided consistently the best results. The differences between AdaBoost and SVM were not large and depended on the particular task. In the case of the data set, for error detection a data collection of 30 minutes with 60 examples is required to have a classification rate $> 90\%$.

VII. APPLICATION TO Q-LEARNING

Finally, we have implemented a simple Q-learning algorithm [1] similar to the one presented in [3] to provide a

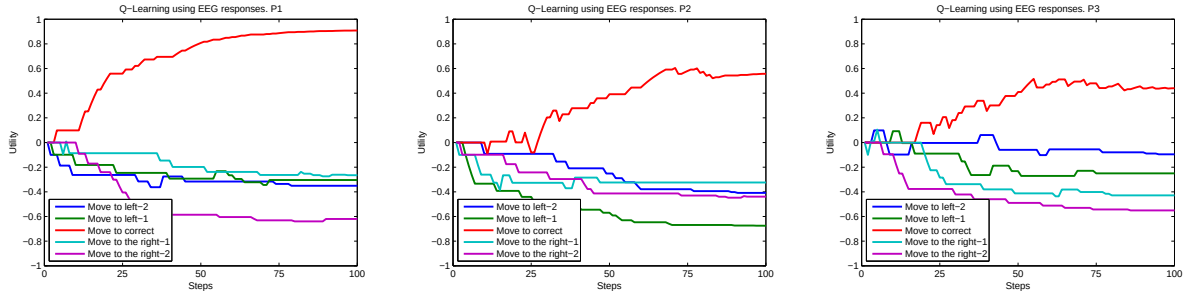


Fig. 6. Q-Learning applied to participant 1 (right), 2 (middle) and 3 (left).

proof of concept of the whole system. For this study, we used the learned SVM classifier for the two class problem (error/correct) to classify online the brain activity during Q-learning. We used the same time window as in the previous section, 0.3-0.9 seconds, and RAW features. The rewards were computed from the classification results as follows: a response classified as error was a -1 reward, and a response classified as correct was a +1 reward. The system started with null Q-function values. Actions were selected according to the ε-greedy policy with decreasing ε and learning rate. The Q-learning algorithm was run up to 100 robot actions.

For the problem at hand, we simply have five different values for each of the possible actions. The Q-values for each participant are shown on Figure 6. The results show how the best action was always the movement towards the center, whereas the other actions gradually decrease in utility.

VIII. Conclusions and Future Work

In this paper, we have demonstrated the existence of a brain response during the observation of a real robot action. The results show that the brain areas involved in this brain activity are those related with prior work on human error processing. The nature of this response, together with the ability to classify single-trial EEG measurements, opens the door to develop robot learning algorithms that use brain activity directly as reward signals.

Our future work focuses on better classification algorithms for the laterality and magnitude of the error. This information may play an important role to implement RL algorithms in more complex settings such as continuous domains and more degrees of freedom. Furthermore, we also plan to explore the detection of errors on a continuous EEG signal to incorporate this on more complex robot actions.

References

[1] Richard S. Sutton and Andrew G. Barto, *Reinforcement Learning: An Introduction*, MIT Press, 1998.

[2] J. Peters and S. Schaal, "Reinforcement learning of motor skills with policy gradients," *Neural Networks*, vol. 21, no. 4, pp. 682–697, 2008.

[3] I.Iturrate, L.Montesano, and J.Minguez, "Robot Reinforcement Learning using EEG-based reward signals," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2010.

[4] P.W. Ferrez and J.d.R. Millán, "Error-related eeg potentials generated during simulated brain-computer interaction," *IEEE Transactions on Biomedical Engineering*, vol. 55, no. 3, pp. 923–929, March 2008.

[5] M. Falkenstein, J. Hoormann, S. Christ, and J. Hohnsbein, "ERP components on reaction errors and their functional significance: A tutorial," *Biological Psychology*, vol. 51, pp. 87–107, 2000.

[6] H.T. van Schie, R.B. Mars, M.G.H Coles, and H. Bekkering, "Modulation of activity in medial frontal and motor cortices during error observation," *Neural Networks*, vol. 7, pp. 549–554, 2004.

[7] K. Brodmann, *Vergleichende Lokalisationslehre der Grosshirnrinde in ihren Prinzipien dargestellt auf Grund des Zellenbaues*, Johann Ambrosius Barth Verlag, 1909.

[8] G. Bush, P. Luu, and M.I. Posner, "Cognitive and emotional influences in anterior cingulate cortex," *Trends in Cognitive Science*, vol. 4, no. 6, pp. 215–222, 2000.

[9] N.Birbaumer et al., "A spelling device for the paralyzed," *Nature*, vol. 398, pp. 297–298, 1999.

[10] C. Guger, W. Harkam, C. Hertnaes, and G. Pfurtscheller, "Prosthetic control by an EEG-based braincomputer interface (BCI)," in *Proceedings of AAATE 5th European conference for the advancement of assistive technology*, 1999.

[11] J.d.R. Millán, F. Renkens, J. Mouriño, and W. Gerstner, "Noninvasive Brain-Actuated Control of a Mobile Robot by Human EEG," *IEEE Transactions on Biomedical Engineering*, vol. 51, no. 6, June 2004.

[12] I. Iturrate, J. Antelis, A. Kuebler, and J. Minguez, "Non-Invasive Brain-Actuated Wheelchair based on a P300 Neurophysiological Protocol and Automated Navigation," *IEEE Transactions on Robotics*, vol. 25, no. 3, pp. 614–627, 2009.

[13] B. Dal Seno, *Toward An Integrated P300- And ErrP-Based Brain-Computer Interface*, Ph.D. thesis, Politecnico di Milano, 2009.

[14] R. Chavarriaga, P.W. Ferrez, and J.d.R. Millán, "To Err is Human: Learning from Error Potentials in Brain-Computer Interfaces ," *1st International Conference on Cognitive Neurodynamics*, 2007.

[15] P.W. Ferrez, *Error-Related EEG Potentials in Brain-Computer Interfaces*, Ph.D. thesis, École Polytechnique Fédérale de Laussane, 2007.

[16] G. Shalk, D.J. McFarland, T. Hinterberger, N. Birbaumer, and J.R. Wolpaw, "BCI2000: A General-Purpose Brain-Computer Interface (BCI) System," *IEEE Transactions on Biomedical Engineering*, vol. 51, no. 6, May 2004.

[17] T. Handy, Ed., *Event-Related Potentials A Methods Handbook*, The MIT Press, 2005.

[18] R.D. Pascual-Marqui, "Standardized low resolution brain electromagnetic tomography (sLORETA): Technical details.," *Methods and Findings in Experimental and Clinical Pharmacology*, pp. 5–12, 2002.

[19] J.M.Bollon, R.Chavarriaga, J.d.R. Millán, and P. Bessiére, "Using dynamic time warping to find patterns in time series," in *4th International IEEE EMBS Conference on Neural Engineering*, 2009.

[20] A.Hyvärinen, J.Karhunen, and E.Oja, *Independent Component Analysis*, Wiley Interscience, 2001.

[21] Y. Freund and R.E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *European Conference on Computational Learning Theory*, pp. 23–37, 1995.

[22] J. Gama, "Functional trees," *Machine Learning*, vol. 55, no. 3, pp. 219–250, November 2004.

[23] V. Vapnik, *The Nature of Statistical Learning Theory*, Springer Verlag, 1995.

[24] I.H.Witten, E.Frank, L.Trigg, M.Hall, G.Holmes, and S.J.Cunningham, "Weka: Practical machine learning tools and techniques with Java implementations," in *ICONIP/ANZIIS/ANNES*, 1999, vol. 99, pp. 192–196.

[25] C.C.Chang and C.J.Lin, "LIBSVM: a library for support vector machines," 2001.

Proceedings of the IEEE ICRA 2010
Workshop on Multimodal Human-Robot Interfaces
Anchorage, Alaska, May, 2010

Temporal gesture recognition for human-robot interaction

Markos Sigalas[†‡], Haris Baltzakis[†] and Panos Trahanias[†‡]

† Institute of Computer Science,
Foundation for Research and Technology - Hellas,
Heraklion, Crete, Greece

‡ Department of Computer Science, University of Crete,
P.O.Box 1470, Heraklion, 714 09 Crete, Greece
{msigalas,xmpalt,trahania}@ics.forth.gr

Abstract— **This paper describes a novel hand gesture recognition system intended to support natural interaction with autonomously navigating robots that guide visitors in museums and exhibition centers. The proposed system utilizes upper body part tracking and two neural network-based classifiers, one for each arm. Tracking is performed in a 9-DoF configuration space and it is facilitated by means of a probabilistic approach which combines particle filters with hidden Markov models in order to enable the simultaneous tracking of several hypotheses for the body orientation and the configuration of each of the two arms.**

Given the arm trajectories in the configuration space, classification is facilitated separately for each arm by means of a combined MLP/RBF neural network structure. The MLP is trained as a standard classifier while the RBF neural network is trained as a predictor for the future state of the system. By feeding the output of the RBF back to the MLP classifier, we achieve temporal consistency and robustness to the classification results.

I. INTRODUCTION

Gesture recognition is an important, yet difficult task. It is important because it is a versatile and intuitive way to develop new, more natural and more human-centered forms of human-machine interaction. Moreover, it is difficult because it involves the solution of many challenging subtasks such as robust identification of hands and other body parts, motion modeling, tracking, pattern recognition and classification.

Early psycholinguistic studies [1], [2], initially targeting sign language gestures, revealed that gestures can be characterized based on four different aspects: shape, motion, position and orientation. All gesture recognition approaches try to approach the problem by concentrating one way or another on one or more of the above four aspects. Posture-based approaches, for example, utilize static images, concentrating only on the shape of the hand to extract features such as hand contours, fingertips and finger directions [3], [4], [5], [6]. Temporal approaches, on the other hand, not only make use of spacial features but also exploit temporal information such as the path followed by the hand, its speed, etc [7], [8], [9], [10].

A category of approaches utilize 3D hand models for the detection of hands in images. One of the advantages of these methods is that they can achieve view-independent detection. The employed 3D models should have enough degrees of freedom to adapt to the dimensions of the hand(s) present in an image. Different models require different image features to construct feature-model correspondences. Point and line features are employed in kinematic hand models to recover angles formed at the joints of the hand [11], [12]. In [13], a 3D model of the arm with 7 parameters is utilized. In [14], a 3D model with 22 degrees of freedom for the whole body with 4 degrees of freedom for each arm is proposed. In [15], the user's hand is modeled much more simply, as an articulated rigid object with three joints comprised by the first index finger and thumb.

In this paper we present a specific approach for vision-based hand gesture recognition, intended to support natural interaction with autonomously navigating robots that guide visitors in public places such as museums and exhibition centers. The operational requirements of such an application challenge existing approaches in that the visual perception system should operate efficiently under totally unconstrained conditions regarding occlusions, variable illumination, moving cameras, and varying background. Recognizing that the extraction of features related to hand shape may be very difficult task, we propose a gesture recognition system that emphasizes on the temporal aspects of the task. More specifically, the proposed approach takes into account information conveyed in the trajectory followed by user's arms while the user performs gestures in front of a robot.

The proposed gesture recognition system builds on our previous work in model-based visual tracking of human arms and body [16]. According to this tracking approach, a nine parameter model is employed to track both arms (4 parameters for each arm) as well as the orientation of the human torso (one additional parameter). In order to reduce the complexity of the problem and to achieve real-time performance, the model space is split into three different partitions and tracking is performed separately in each of them. More specifically, a Hidden Markov Model (HMM) is used to track the orientation of the human torso in the 1D space of all possible orientations and two different sets of particles are used to track the four Degrees of Freedom

(DoF) associated with each of the two hands, using a particle filter-based approach.

Given the arm trajectories in the configuration space, classification is facilitated separately for each arm by means of a combined Multi Layer Perceptron/Radial Basis Function (MLP/RBF) Neural Network structure. The MLP is trained as a standard classifier while the RBF neural network is trained as a predictor for the future state of the system. By feeding the output of the RBF back to the MLP classifier, we achieve temporal consistency and robustness in the classification results.

Sample experimental results presented in this paper, confirm the effectiveness and the efficiency of the proposed approach, meeting the robustness and performance requirements of this particular case of human-robot interaction.

II. APPROACH OVERVIEW

A block diagram of the proposed gesture recognition system is illustrated in Figure 1.

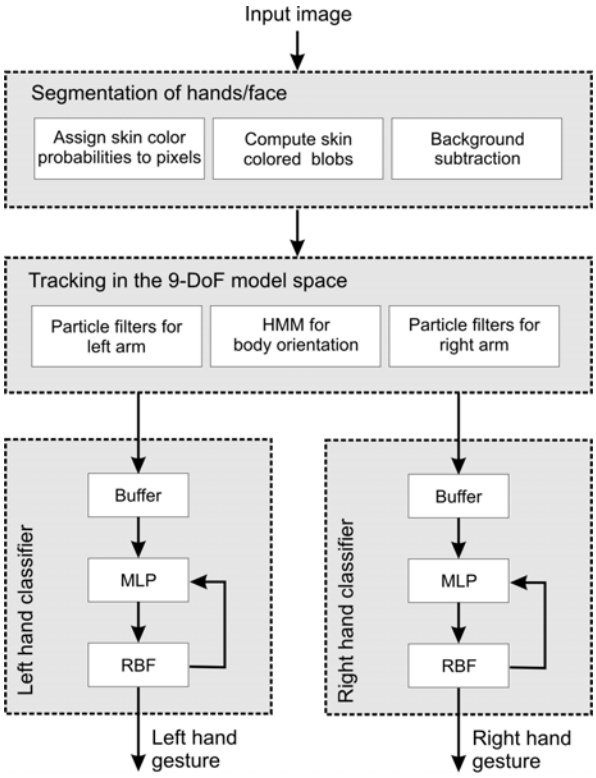


Fig. 1. Block diagram of the proposed approach for hand tracking and gesture recognition. Processing is organized into three layers.

The first step of the approach is to extract hand and face regions as skin-colored foreground blobs.

Assuming a 4 DoF kinematic model for each arm and one additional degree of freedom for the orientation ϕ of the user around the vertical axis (see Fig. 2), the pose of the user is tracked in a 9 DoF model space. The resulting 9-parameter tracking problem is tackled in realtime by fragmenting the 9-dimensional space into three sub-spaces; a 1D parameter space for body orientation angle and two 4D spaces, one for each hand. The body orientation angle ϕ is appropriately

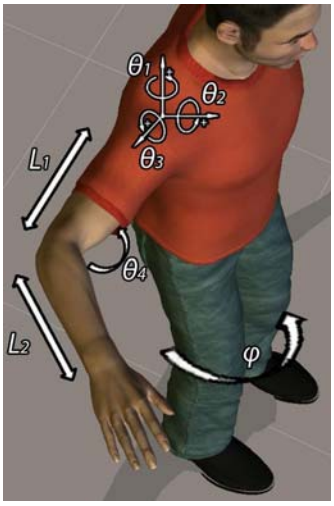


Fig. 2. The 9-parameter model used for the rotation of the body and the pose of the user's arms

quantized and tracked over time by means of an HMM. For every possible solution, a separate particle filter set is employed for each arm. The result of each particle filter is used to estimate the observation probability, which is subsequently employed to update the HMM.

Classification is achieved by buffering the trajectory of each arm (in its 4D configuration space) and feeding it to a feed-forward MLP Neural Network which is trained to recognize between five system states: idle (no gesture), preparation (hand moving towards a gesture), pointing gesture, hello (waiving) gesture, and retraction (hand retracting from a gesture). The output of the MLP is passed though an RBF which is trained as a predictor for the next state of the system and fed back to the MLP in order to improve temporal consistency and robustness of the achieved results.

More details regarding each of the above described modules are provided in the following sections.

III. DETECTION OF HAND AND FACE BLOBS

The first step of the proposed approach is to detect skin-colored regions in the input images. For this purpose, a technique similar to [17], [18] is employed. Initially, background subtraction [19] is used to extract the foreground areas of the image. Then, for each pixel, $P(s \mid c)$ is computed, which is the probability that this pixel belongs to a skin-colored foreground region s, given its color c.

This can be computed according to the Bayes rule as:

$$P(s \mid c) = \frac{P(s)}{P(c)} P(c \mid s) \tag{1}$$

where $P(s)$ and $P(c)$ are the prior probabilities of foreground skin pixels and foreground pixels having color c, respectively. Color c is assumed to be a 2D variable encoding the U and V components of the YUV color space. $P(c \mid s)$ is the prior probability of observing color c in skin colored foreground regions. All three components in the right side of Eq.1 can be computed via offline training.

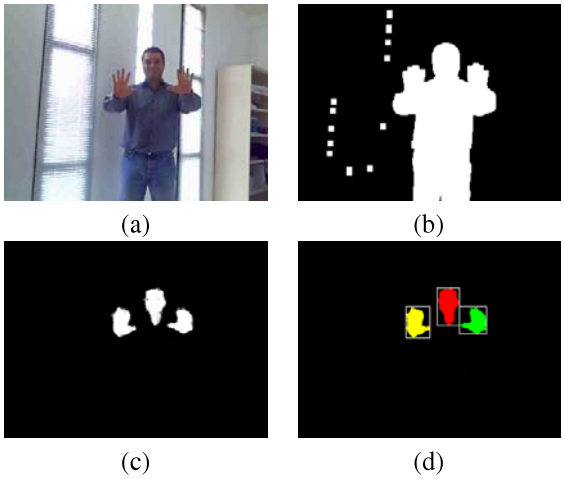


Fig. 3. Blob detection. (a) Initial image, (b) Foreground pixels, (c) skin-colored pixels, (d) resulting skin-colored blobs.

After probabilities have been assigned to each image pixel, hysteresis thresholding is usand connected components labeling are used to extract solid skin color blobs. Pixel probabilities are initially thresholded by a "strong" threshold T_{max} to select all pixels with $P(s \mid c) > T_{max}$. This yields high-confidence skin-colored pixels that constitute the seeds of potential blobs. A second thresholding step, this time with a "weak" threshold T_{min} is performed. During this step, pixels with probability $P(s \mid c) > T_{min}$ where $T_{min} < T_{max}$, that are immediate neighbors of already classified skin-colored pixels, are recursively added to each blob.

A connected components labeling algorithm is then used to assign different labels to pixels that belong to different blobs. Size filtering on the derived connected components is also performed to eliminate small, isolated blobs that are attributed to noise.

A set of simple heuristics based on location and size is used to characterize blobs as hand blobs and face blobs.

Results of the intermediate steps of this process are illustrated in Fig. 3. Figure 3(a) shows a single frame extracted out of a video sequence that shows a man performing various hand gestures in an office-like environment. Fig. 3(b) shows the result of the background subtraction algorithm and Fig. 3(c) shows skin-colored pixels after hysteresis thresholding. Finally, the resulting blobs (i.e. the result of the labeling algorithm) are shown in Fig. 3(d).

IV. TRACKING IN THE MODEL SPACE

A. Kinematic model

As already mentioned, for modeling the human body and arms, a nine-DOF model, has been employed. This model, which is similar to the one proposed in [20] is depicted in Figure 2. According to this model, the human body, with the exception of the arms, is assumed to be a rigid object with only one degree of freedom corresponding to its orientation ϕ. Both arms are assumed to be attached to this rigid body at fixed locations (i.e. the shoulders) and they are modeled by a 4-DoF kinematic model each. The kinematics of each arm are defined as Denavit-Hartenberg parameters, shown in table I. θ_1,θ_2 and θ_3, are the angles corresponding to the three DoFs of the human shoulder and θ_4 corresponds to the angle of the elbow. L_1 and L_2 are the lengths of the upper arm and the forearm, respectively. They are assumed fixed in our implementation.

TABLE I

DENAVIT-HARTENBERG PARAMETERS FOR THE 4-DOF MODEL OF THE HUMAN ARM EMPLOYED IN OUR APPROACH.

i	α_{i-1}	a_{i-1}	d_i	θ_i
1	$+\pi/2$	0	0	$\theta_1 - \pi/2$
2	$-\pi/2$	0	0	$\theta_2 + \pi/2$
3	$+\pi/2$	0	L_1	$\theta_3 + \pi/2$
4	$-\pi/2$	0	0	$\theta_4 - \pi/2$
5	0	L_2	0	0

B. Model space partitioning and tracking

To track in the 9-DoF model space presented in the previous section, the approach presented in [16] has been assumed. According to this approach, in order to reduce the complexity of the problem and meet the increased computational requirements of the task at hand, the model space is split into three different partitions and tracking is performed separately in each of them. More specifically, a Hidden Markov Model (HMM) is used to track the orientation ϕ of the human body in the 1D space of all possible orientations and two different sets of particles are used to track the four DoFs associated with each of the two arms using a particle filtering approach.

To facilitate the implementation of the HMM, the body orientation angle ϕ is appropriately quantized (50 quantization levels were used in our implementation). For every possible solution, a separate particle filter set is employed for each arm. The result of each particle filter is used to estimate the observation probability, which is subsequently employed to update the HMM. This means that the weights of the particles are used to calculate the observation likelihood for a particular body orientation state.

To facilitate the implementation of likelihood function which is necessary in order to evaluate hypotheses in the particle filter-based trackers, the kinematic model defined in the previous section is used, along with the camera perspective transformations. More specifically, forward kinematic equations are used to transform the rotation of the human body and the angles of the arm joints to 3D coordinates for each joint (shoulder, elbow and hand). Accordingly, camera projection transformations are used to project the resulting 3D coordinates of the joints on the image frame. The projected joint locations are evaluated by comparing them with actual observations according to two different criteria: (a) Projected hand locations should be close to observed skin-colored blobs, and (b) projected elbows and shoulders should be within foreground segments of the image.

Figures 4(a) and 4(b) demonstrate the operation of the particle filter trackers that correspond to a specific value of the orientation angle ("0" in both cases). On the right

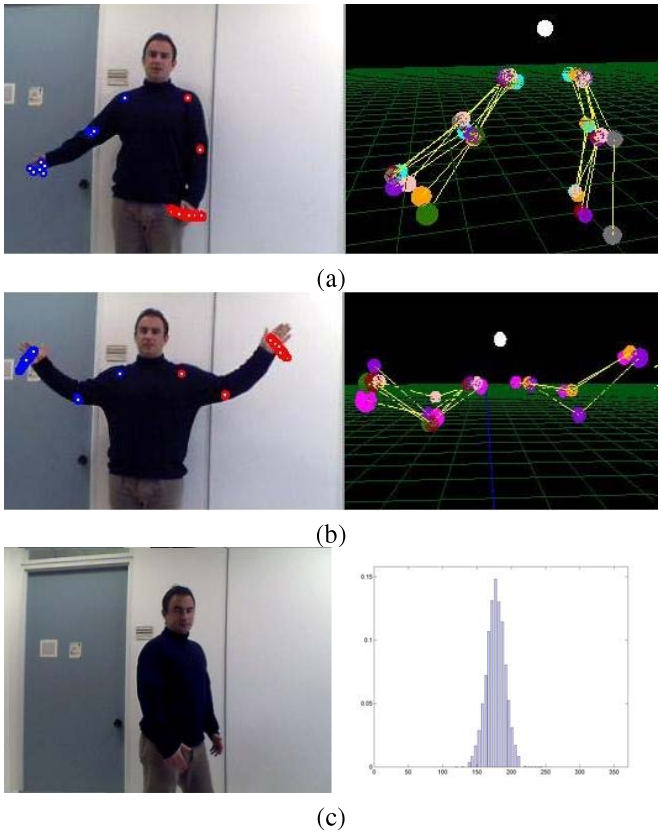


Fig. 4. Operation of the tracker; (a,b) Particle filter sets for a specific orientation angle, (c) A HMM histogram corresponding to a specific frame.

parts of the two images are the samples projected on the 3D space (using forward kinematics, as described above). The corresponding sample projections on the image plane are depicted on the left. Figure 4(c) depicts a sample orientation histogram as tracked by the HMM. The values of each histogram cell correspond to the probability of this specific orientation being the correct orientation.

V. GESTURE CLASSIFICATION

As observed in [21], gestures are dynamic processes that typically consist of three phases: preparation, stroke and retraction. The preparation and retraction phases consist of arm movement from and towards the resting position, before and after the gesture, respectively. These phases have been found to be similar in content between many common gestures and therefore contribute little to the gesture recognition process. The stroke phase is the one that contains most of the information that characterizes a gesture.

Based on the above observations our system has been designed to recognize five different gesturing states:

- Idle. No gesture is performed,
- Preparation phase.
- Pointing gesture,
- Hello gesture. The user is waiving using his hand.
- Retraction phase.

The mentioned states correspond to two different strokes (pointing and hello gestures), the accompanying phases (preparation and retraction) and the idle phase. The transitions between the above-mentioned states are illustrated in Figure 5.

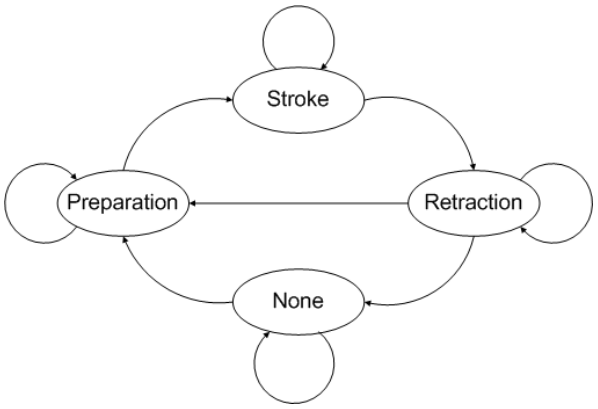


Fig. 5. Gesture state transitions.

Classification is achieved by buffering the trajectory of each arm (in its 4D configuration space) and feeding it to a feed-forward MLP Neural Network which is trained to recognize between the five system states. The output of the MLP is passed though an RBF neural network which is trained as a predictor for the next state of the system and fed back to the MLP in order to improve temporal consistency and robustness of the achieved results.

This classification structure is graphically illustrated in the lower part of Figure 1. As implied in Figure 1, the same structure is employed separately for gesture recognition in each arm.

A. The MLP

The MLP neural network is mainly responsible for gesture recognition and classification, according to the trained patterns. It consists of the input layer, the output layer and two hidden layers. The output layer consists of 4 neurons encoding the five possible states of the system. The input layer consists of 44 neurons; 40 neurons are used to provide input about the trajectory of the arm (4 parameters per frame, 10 frames history) and the rest 4 neurons are used to provide the prediction for the next state of the system, which is fed back by the RBF neural network.

B. The RBF

The input layer of the RBF network consists of four neurons which are connected to the output of the MLP. The output of the RBF Network also consists of four neurons and it is fed back to the MLP. Given the output of the MLP, the RBF is trained to provide a prediction for the next state of the system.

The intuition behind this is simple: one can think of a gesture as a state transition process with acceptable and unacceptable state transitions (Figure 5). However, due to possible discontinuities in the MLP input data (caused by erroneous tracking or lost frames in the video), the output (of the MLP) can itself present discontinuities, translated

to unacceptable state transitions, as well. The RBF network restricts unacceptable transitions and smooths out outliers at the output of the MLP.

C. Network training

For training the proposed classifier, a dataset consisting of 12 sequences was used. This dataset contains six examples of each of the two cosidered gestures. In each of these sequences all three phases of a gesture appear, together with cases where none of the phases is performed or when both hands are acting simultaneously. The dataset was divided into two subsets, of 6 sequences each. The first subset contained 3 sequences from each of two gestures and it was used to train the MLP neural network while the second subset was used to train the RBF network. Using the two subsets, the training of the system was done in two steps.

Training of the MLP was performed by minimizing the mean of the squared error using the Levenberg-Marquardt algorithm. To train the RBF network, the sequences used for training the MLP cannot be used because they are known to the classifier. Thus, the second training set is used.

VI. RESULTS

During our experiments, 3 sequences for each gesture, different from the ones used during the training step, have been tested. The examined scenarios contained both gestures performed by one arm only and by both hands simultaneously. Our main target was to study whether sequences of arm kinematic configurations contain enough information to describe a gesture, given the fact that no other information about the location of the arm has been used.

The proposed approach performed very well in all test cases. Four illustrative examples are depicted in Figure 6. In Figures 6(a) and 6(b) the user performs a right hand gesture that is correctly classified by the employed Neural Network structure. Figures 6(c) and 6(d) present two additional examples where the user gestures with both hands simultaneously.

Table II presents quantitative results obtained with the employed datasets. The TP figures shown in table II correspond to the percentage of correctly classified frames (True Positive classifications). Similarly, FP and FN figures correspond to percentages of false positive and false negative classified frames.

As can easily be observed, the successful recognition ratio does not drop below 86% while the false negative percentage remains in low levels as well. Further experiments have been conducted by eliminating the RBF neural network from the classification structure. In these cases the percentage of false positive decisions for the preparation and retraction phase was higher than 15%. Evidently, the utilization of the RBF network has greatly contributed to the robustness of the classifier by filtering out temporal inconsistencies in the output of the MLP.

VII. CONCLUSION

In this paper, we have presented a novel temporal gesture recognition system intended for natural interaction with

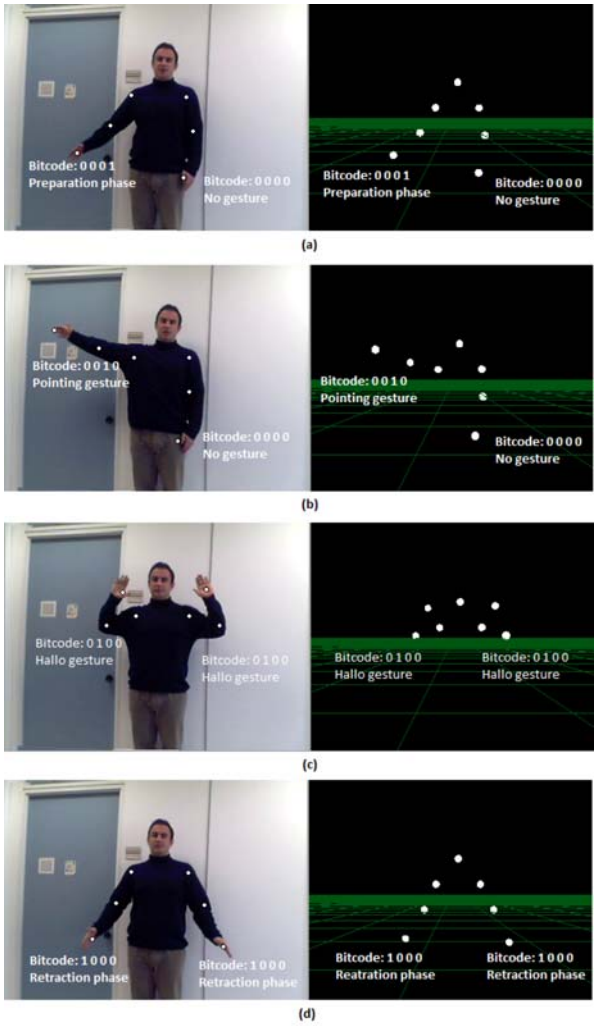


Fig. 6. Recognition and classification of gestures performed by one or both hands simultaneously. The left image depicts a 2D view from one camera of the stereo pair, while the right image shows the 3D representation (of the left image). The output of the classifier has been superimposed on the images for the sake of clarity. (a)The right arm prepares to perform a gesture. (b)The right hand performs a "pointing" gesture. (c)Both hands perform a "hello" gesture. (d)Both hands retract from the stroke phase.

TABLE II
GESTURE CLASSIFIER QUANTITATIVE RESULTS. TP:TRUE POSITIVES, FP: FALSE POSITIVES, FN: FALSE NEGATIVES.

Preparation			Pointing		
TP	FP	FN	TP	FP	FN
88.46%	11.54%	6.47%	86.48%	13.52%	2.08%
Hello			**Retraction**		
TP	FP	FN	TP	FP	FN
96.91%	3.09%	1.41%	86.04%	13.96%	6.21%

autonomous robots that guide visitors in museums and exhibition centers. The proposed gesture recognition system builds on our previous work on vision based tracking and more specifically on a probabilistic tracker capable to track both hands and the orientation of the human body on a nine-parameter configuration space.

Dependable tracking, combined a novel, two-stage neural network structure for classification, facilitates the definition of a small and simple hand gesture vocabulary that is both robustly interpretable and intuitive to humans. Experimental results presented in this paper, confirm the effectiveness and the efficiency of the proposed approach, meeting the run-time requirements of the task at hand.

Nevertheless, and despite the vast amount of relevant research efforts, the problem of efficient and robust vision-based recognition of natural hand gestures in unprepared environments still remains open and challenging, and is expected to remain of central importance in human-robot interaction in the forthcoming years. In this context we intend to continue our research efforts towards enhancing the current system. At first we plan to redesign the classification structure in order to take into account the multiple hypotheses provided by the employed tracker. This is expected to increase classification accuracy since errors in the early processing stages (tracking) are not propagated to later stages (classification). Additionally the training and test datasets will be expanded to include richer gesture vocabularies and larger intra-gesture variation. Finally, we intend to include a more sophisticated algorithm to classify skin colored blobs to hands and faces. This will allow our system to cope with more complex cases where multiple users simultaneously interact with the robot.

VIII. ACKNOWLEDGMENTS

This work has been partially supported by the EU Information Society Technologies research project INDIGO (FP6-045388).

REFERENCES

[1] W. C. Stokoe, *Sign Language Structure*. Buffalo, NY: Buffalo Univ. Press, 1960.

[2] R. Battison, "Phonological deletion in american sign language," *Sign language studies*, vol. 5, no. 1, pp. 1–19, 1974.

[3] V. Pavlovic, R. Sharma, and T. S. Huang, "Visual interpretation of hand gestures for human-computer interaction: A review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 677–695, 1997.

[4] J. J. LaViola, "A survey of hand posture and gesture recognition techniques and technology," Department oc Computer Science, Brown University, Providence, Rhode Island., Tech. Rep. CS-99-11, 1999.

[5] Y. Wu and T. S. Huang, "Vision-based gesture recognition: A review," *Lecture Notes in Computer Science*, vol. 1739, pp. 103+, 1999.

[6] X. Zabulis, H. Baltzakis, and A. Argyros, "Vision-based hand gesture recognition for human-computer interaction," in *The Universal Access Handbook*, ser. Human Factors and Ergonomics, C. Stefanides, Ed. Lawrence Erlbaum Associates, Inc. (LEA), to appear.

[7] A. Wilson and A. Bobick, "Parametric hidden markov models for gesture recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 21, no. 9, pp. 884–900, Sept. 1999.

[8] S. Wang, A. Quattoni, L. Morency, and D. Demirdjian, "Hidden conditional random fields for gesture recognition," in *Proc. IEEE Computer Vision and Pattern Recognition (CVPR)*, 2006, pp. II: 1521–1527.

[9] L. Morency, A. Quattoni, and T. Darrell, "Latent-dynamic discriminative models for continuous gesture recognition," in *Proc. IEEE Computer Vision and Pattern Recognition (CVPR)*, 2007, pp. 1–8.

[10] H. Suk, B. Sin, and S. Lee, "Robust modeling and recognition of hand gestures with dynamic bayesian network," in *Proc. International Conference on Pattern Recognition (ICPR)*, 2008, pp. 1–4.

[11] J. Rehg and T. Kanade, "Model-based tracking of self-occluding articulated objects," in *Proc. International Conference on Computer Vision (ICCV)*, 1995, pp. 612–617.

[12] N. Shimada, Y. Shirai, Y. Kuno, and J. Miura, "Hand gesture estimation and model refinement using monocular camera - ambiguity limitation by inequality constraints," in *IEEE Int. Conf. on Face and Gesture Recognition*, Nara, Japan, 1998, pp. 268–273.

[13] L. Goncalves, E. di Bernardo, E. Ursella, and P. Perona, "Monocular tracking of the human arm in 3D," in *Proc. International Conference on Computer Vision (ICCV)*, Cambridge, 1995, pp. 764–770.

[14] D. Gavrila and L. Davis, "3-D model-based tracking of humans in action: a multi-view approach," in *Proc. IEEE Computer Vision and Pattern Recognition (CVPR)*, 1996, 1996, pp. 73–80.

[15] J. MacCormick and M. Isard, "Partitioned sampling, articulated objects, and interface-quality hand tracking," in *Proc. European Conference on Computer Vision*, 2000, pp. 3–19.

[16] M. Sigalas, H. Baltzakis, and P. Trahanias, "Visual tracking of independently moving body and arms," in *Proc. IEEE/RSJ Intlernational Conference on Intelligent Robotics and Systems (IROS)*, St. Louis, MO, USA, Oct. 2009.

[17] A. A. Argyros and M. I. A. Lourakis, "Real-time tracking of multiple skin-colored objects with a possibly moving camera." in *Proc. European Conference on Computer Vision*, Prague, Chech Republic, May 2004, pp. 368–379.

[18] H. Baltzakis, A. Argyros, M. Lourakis, and P. Trahanias, "Tracking of human hands and faces through probabilistic fusion of multiple visual cues," in *Proc. International Conference on Computer Vision Systems (ICVS)*, Santorini, Greece, May 2008, pp. 33–42.

[19] W. E. L. Grimson and C. Stauffer, "Adaptive background mixture models for real time tracking," in *Proc. IEEE Computer Vision and Pattern Recognition (CVPR)*, Ft. Collins, USA, June 1999, pp. 246–252.

[20] D. Tsetserukou, R. Tadakuma, H. Kajimoto, and N. Kawakami, "Development of a whole-sensitive teleoperated robot arm using torque sensing technique," in *WHC '07: Proceedings of the Second Joint EuroHaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*. Washington, DC, USA: IEEE Computer Society, 2007, pp. 476–481.

[21] A. Kendon, "Current issues in the study of gesture," *The Biological Foundations of Gestures: Motor and Semiotic Aspects*, pp. 23–47, 1986.

Proceedings of the IEEE ICRA 2010
Workshop on Multimodal Human-Robot Interfaces
Anchorage, Alaska, May, 2010

BCIs and Mobile Robots for Neurological Rehabilitation: practical applications of remote control.

Luigi Criveller[1], Emanuele Menegatti[1], Francesco Piccione[2], Stefano Silvoni[2].

[1] Dept. of Information Engineering, University of Padua, Italy.
[2] IRCCS San Camillo Hospital, Venice, Italy.

***Abstract*- This project aims at testing the possible advantages of introducing a mobile robot as a physical input/output device in a Brain Computer Interface (BCI) system. In the proposed system, the actions triggered by the subject's brain activity results in the motions of a physical device in the real world, which we believe are more compelling than just a changement in a graphical interface on a screen. A goal-based system for destination detecting and the high engagement level offered by controlling a mobile robot are hence main features for actually increase patients' life quality level.**

I. INTRODUCTION

A BCI system enables control of devices or communication with other persons through cerebral activity. Because they don't depend on neuromuscular control, BCIs can provide communication and control for people with devastating neuromuscular disorders, such as amyotrophic lateral sclerosis, brainstem stroke, cerebral palsy, and spinal cord injury. BCIs for control applications can be based on goal selection. which means the BCI simply indicates the desired outcome, and downstream hardware and software handle the continuous kinematic control that achieves the outcome. Goal selection is much less demanding in terms of the complexity and rate of the control signals the BCI must provide with respect of direct control of a device.

Our project aims at testing the possible advantages of introducing a mobile robot as a physical input/output device in a system of Brain Computer Interface (BCI). In the proposed system, the actions triggered by the subject's brain activity results in the motions of a physical device in the real world (i.e. the robot), and not only in a modification of a graphical interface. Moreover, our robot is fitted with a camera for real-time video feedback and a microphone for telepresence applications. Joining the low complexity of a goal-based system for detecting a suitable destination and the engagement level generated in the patient by a mobile robot allows the reliable (goal-based) control of a mobile robot platform. With such a device, patients are allowed to overcome, at least partially, their physical disorders and to interact with real world with a certain degree of freedom.

The software modules we developed for interfacing the brain to the robot is called NEVRAROS. NEVRAROS is a non-Invasive P300-based BCI system. It presents to patient a captivating, but simple graphical interface which provides specific visual stimuli for destinations selection; the patient answers to such that stimuli with a specific EEG amplitude alteration. NEVRAROS acquires brain signals, extracts key features from them, and translates the features into goal-commands for remote mobile robot. Goal-commands are then converted in low level commands for navigation. In the meantime, the camera and the microphone mounted on the robot send a perceptive stream to patient's display, so he can "experience" the navigation environment.

NEVRAROS's architecture is divided into logical modules: a set of reusable electrodes (located in Oz, Pz, Cz, Fz, EOG positions), and a EEG amplifier from Compumedics Neuroscan™ are arranged for raw cerebral signal acquisition. HIM module from BCI++™ system and a Matlab™-based classifier extract and classify useful features from raw signal. AEnima module from BCI++™ is responsible both for user display interface, synchronize triggers for HIM features extraction and convert logical control signals into semantic control signal for robotic device. User's visual stimulation is provided by a suitable user interface and allows users to express P300 peaks in their EEG.

Main related works are presented in Chapter II. A concise excursus of base concepts concerning goal-based selection is presented in Chapter III. Chapter IV introduces NEVRAROS system's design, while implementation and system's architecture is presented in Chapter V. Chapter VI is for showing system evaluation, both from performances and usability point of view. Description of result in testing NEVRAROS over patients will be presented as well. References can be found in Chapter VII.

II. RELATED WORKS

Advances in neuroscience and computer technology have made possible a number of recent demonstrations of direct brain control of devices such as a cursor on a computer screen and various prosthetic devices, and experiments in using non-invasive BCI system for controlling a remote autonomous robotic device already took place. A P300-based BCI system offering a web GUI for controlling remote mobile robots was tested by A. Chella, E. Pagello, E.

1Manuscript received March 7, 2010. This work was supported in part by the IRCCS San Camillo Hospital, Venice, Italy.
Luigi Criveller and Emanuele Menegatti are with the Dept. of Information Engineering, University of Padua, Italy, 35100, PD, ITA
(e-mail: crivelle@dei.unipd.it, emg@dei.unipd.it)
Francesco Piccione and Stefano Silvoni are with the IRCCS San Camillo Hospital, VE, 30126, ITA
(e-mail: francopiccione@libero.it) , stefano.silvoni@libero.it).

Menegatti, K. Prifitis et al. [1]. Noninvasive EEG signals recorded over sensorimotor areas (corresponding to three mental states classified using spectral features from the EEG as input) were used on BCI system to give human users control of a mobile robot by J. del R. Millan et al. [26]. A humanoid robot equipped with a video camera that displayed objects in its environment was partially controlled by a human operator's P300 response thanks to a recent work of C. J. Bell et al. [18]. A noninvasive BCI with a shared control system that helps users in driving an intelligent wheelchair by estimating users' steering intentions from EEG was tested by F. Gàlan, J. del R. Millan et al. [32]. The conviction that non-invasive EEG-driven BCIs offer a realistic perspective for communication in paralyzed patients initially was demonstrated by N. Birbaumer, J. R. Wolpaw et al. [15] and then confirmed by N. Birbaumer. [12].

Uses of P300 as neurological signal for brain activities were discovered first by S. Sutton et al. [33], and then used as signal for cursor control by D.J. McFarland, W.A. Sarnacki, and J.R. Wolpaw [20]. Recently, P300 signal were used as base in systems for disabled subjects by Piccione et al. [7], Y. Wang et al. [16] and Sellers and Donchin [5].

Moreover, the use of P300 as simple and non tiring paradigm was cover by Hiroyuki Ishita et al. [34]. The characteristics of not demanding specific training to the user on P300 and SSVEP based BCIs were underlined by F. Piccione, F. Beverina, G. Palmas, S. Silvoni [27].

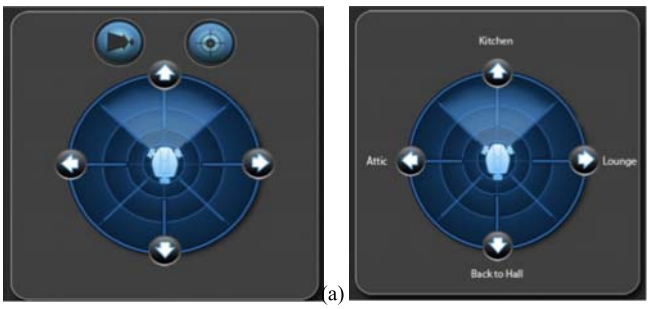
III. GOAL-BASED SELECTION

Although concurrent studies on BCI show many important results in controlling vehicles suitable to carry human being (like robotic wheelchairs), their complexity is high, both in terms of implementation and easy of usage by patients. In most cases, non-Invasive BCI for controlling prosthetic objects requires the ability to activate specific patterns of brain signals that can be achieved by healthy patients only after weeks or even months of training, and time of learning for disabled patients would be longer or even infeasible. The idea of developing a robotic platform to offer a remote telepresence experience, allowing patients to view and listen what the robot perceives and to send high level commands to the robot is an application which is simpler to implement than other solutions and can be more easily realized in practice, using methods that require patients little learning effort, and extremely limited training time.

Abstracting movement concepts from a human-like prospective, and idealizing the hypothetical movements device in a non specified mobile device, it is easily demonstrable that user is able to control this device in navigating into 2D surfaces (i.e. a flat surface in a 3D environment), using only six high-level commands: four of them are related to selecting movement direction (right, left, forward, backward); the other two are used for actually impress the movement through selected direction (go, stop). The number of high-level command can be reduced to four without reducing navigation freedom: the two command for impress the movements can be absorbed by the four commands for direction selection as follow: user select one of the directional commands for starting movement in that direction, and then select opposite movement direction command for stopping movement (i.e. "left" command device to navigate in left direction if device is in "stop" status, while stop the device if it is in "right" navigation status). Such these schemes are simple but complete, only if high-level commands can be imparted to device in real time by the user. A solution of this kind is hence not still acceptable if considering that device's users will be patients with neurological diseases and elder average age.

Overcoming this problem is possible, if one implement a certain degree f autonomy in the robot navigation system. So the user is able to select the robot's destination without worrying about tight deadlines in controlling navigation: once he select the favorite destination, automatic routines will autonomously select appropriate paths, and the robot will reach such that destination with no further user's effort.

IV. SYSTEM DESIGN



(a) (b)

Fig. 1. (a) controlling a mobile device with six high-level commands: four directional arrows for direction decision, and two movements buttons for starting and stopping movement; (b) controlling a mobile device with four high-level destination commands: three select new destinations, one for going back to last destination.

NEVRAROS functional model recalls functional model for a typical brain-computer interface: patient, or user, is connected to the system by a set of electrodes properly attached on the cranial skin outside the skull. Brain activity is hence detected and transmitted to an opportune amplifier as electrical signal. Once signal is amplified to opportune level, the enhanced signal is transmitted to the classifier. The patient is looking at a graphical user interface in which blinking arrows represent the choices the patient can pick. When, the intended choice is blinking the patient's recognition process activate in the brain the so called P300 brain wave. In the mean time, to double check the appropriate timing of the graphical user interface, an external analogical blink sensor acquire interval time of stimulation provided by user interface, and transmit to the classifier (via amplifier, again) a pseudo-periodic square wave, which indicates when stimulations take place (high voltage level means visual stimulation is on, low voltage level represents visual stimulation is off). When the classifier receives both signals from blink sensor and electrodes, starts to classifier the P300 waves resident into brain signal using square wave as selector of the time

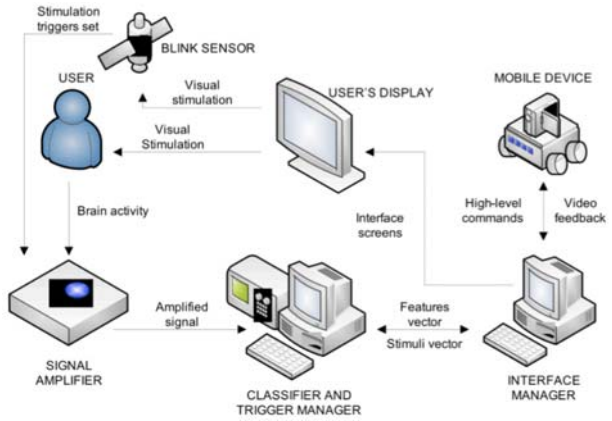


Fig. 2. Proposed functional model for NEVRAROS.

window to analyze. Once classification is finished, results are transmitted to interface manager which updates user interface status (providing new stimuli or calling required module) and sends, if necessary, high level commands to the navigation module of the mobile robot. During all its execution lifetime, the mobile device sends back to the interface manager a video stream that is displayed into user interface screen.

Fig. 2 shows an high-level diagram of components for NEVRAROS system. Note that the blink sensor is used only for debugging purposes. Once we assessed the appropriate timing between all system hardware and software components, the same function provided by blink sensor can be implemented in the software of the graphical user interface which can send an acknowledgment signal to the classifier. The motivation for using two different methods to communicate triggers of classification is for test purpose: the use of software for sending triggers is more compact and require less devices, but delays can occur during transmission, because acknowledgment signal is transferred through a socket into the network that connect classifier and interface manager. Such those delays are not deterministic, and can vary in a little range. The use of external device allows quantification of delays by simple confronting arrival time of square waves with arrival times of acknowledgment sockets. Once the actual delay is quantified, and the classifier is set up for managing it, blink sensor can be removed.

V. SYSTEM IMPLEMENTATION

A. Components

Main components necessary for implementing a NEVRAROS-like BCI system are listed below; note that some of such these components, even if perfectly functioning for the scope of this project, can surely be superseded with other technologies, especially those devices concerning medical and neurological equipment.

One of the keys to recording good EEG signals is the type of electrodes used. Electrodes that make the best contact with a subject's scalp and contain materials that most readily conduct EEG signals provide the best EEG recordings: IRCCS San Camillo Hospital provided a set of shielded reusable disks made of silver chloride (Ag-AgCl).

Once electrodes are set up, the amplifier gets input signal directly from an examinee head. Amplitude of brain potentials measured directly on a scalp is about 100uV. The bioelectric signals are hence small, in terms of voltage, and require considerable amplification. IRCCS San Camillo Hospital provided a Synamps amplifier from Compumedics Neuroscan, with a SCAN Acquisition software as interface to the amplifier.

In order to provide a solid structure for the entire system, a framework from Sensibilab was used: BCI++ (*http://www.sensibilab.campuspoint.polimi.it*). BCI++ is dedicated to the development and fast prototyping of Brain-Computer Interface systems, pc-driven protocols for a variety of bio-signal acquisition paradigms and BCI-based applications.

The BCI++ features two main modules communicating via TCP/IP connnection: HIM, a module dedicated to signal acquisition, storage and visualization, real-time execution and management of custom algorithms (developed using C/C++ or Matlab®) and AEnima, a Graphic User Interface module dedicated to pc-driven protocols development based on a high-level 2D/3D Graphic Engine (Irrlicht).

NEVRAROS display interface represents the core of all the system. It is an AEnima dependent dynamic library loading which contains algorithms and code for user stimulation, mobile device control, video feedback from mobile device control and trigger generator for HIM. It is written in C++ language, for complete integration with AEnima framework. It offers to patient three different use modes: learning (simulated environment) where user replicates interface's choices filling database with his own data set; testing (simulated environment) where data set goodness is tested asking user to reach specific targets; free riding (simulated and real environment), where user has

Fig. 3. BCI++ main components: HIM and AEnima launchers, EEG acquiring window and logos.

Fig. 4. Main windows of NEVRAROS Display Interface. (a) Primary module, which also load graphic elements such as mini map of the overall environment. (b) Selection module. Upper arrow is blinking for stimulating patient, and central cursor is moving for reaching desired destination. (c) Navigation module. While mobile device is reaching selected destination, a video stream is presented to user, as well as an intuitive indication of path selected (the room where the target resides is highlighted) and the overall progress of the path (little arrows on top indicate percentage of path already crossed. (d) Supervisor module, for tuning interface's parameters up.

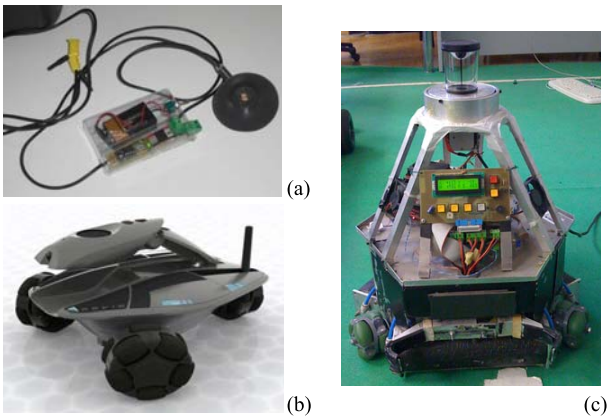


Fig. 5. Blink sensor and robotic devices. (a) Blink sensor. The black cap store the photodiode and assure protection against external artifacts. (b) WowWee Rovio holonomic robot. (c) Team Artisti Veneti Fred holonomic robot.

complete control of interface and device robot over concrete environment.

The blink sensor is a custom made electronic device created in order to detect visual stimulation offered by the user interface. This device is composed by a photodiode, which observes light emission changing frequencies, and an electrical circuit that transforms the information given by the photodiode in a suitable electrical signal for the amplifier. The blink sensor is an analogical device, and it can distinguish between only two light emission levels. Low level is set to zero (equivalent to no significant light emission detected), while a potentiometer allows user to manually select suitable range for high level. A led is also provided for external visual feedback, and works as follow: a solid green represents an high light emission, while no reaction represent no light emission. The device is battery powered, and is provided with suitable external cables terminating with a jack that transmit output signal, for connecting it to the amplifier.

Concerning robot device to use within the system, two main choice are available. The first is Fred: an holonomic robot produced by "Team Artisti Veneti" and builded with hexagonal structure and three omnidirectional wheels. In this way the robot is able to move to any position in the plane without rotate itself. The robot is also fit with a framegrabber for video acquisition and an audio board and WiFi connectivity. The robot also has an omni-directional camera with an hyperbolic mirror. WowWee RovioTM is a mobile wireless IP camera with a three-wheeled drive system. The second choice is Rovio: an inespensive personal robot available in the market. Rovio is equipped with a microphone and a camera to remotely transmits its perceptions and IR sensors on the front for basic obstacle avoidance.

B. Architecture

Previously, a general system description from components point of view was described. Now the system is presented from activities point of view. A general scenario for system behavior can be describe as follow. The set of electrodes is placed over user head skin, in position Oz, Pz, Cz, Fz. For the scope of P300 classification, 4 sites are enough for signal acquisition. Midline is choose for equalize different hemisphere contribute to resulting signal. EOG site (left eye) is also included for acquiring eyes muscles activity and removing it from brain signal. Once electrodes are set up, user brain activity from visual stimulation is sent through amplifier, for pre processing, to SCAN and HIM, for classification. Classification algorithm in use is based into SVM and ICA, evolution of previous algorithm created by F. Piccione et al. [27], here adapted for execution into HIM module. Together with brain signals, HIM also receive from Blink sensor a square wave, and from AEnima a set of triggers for stimulation timing. Once classification is finished, Aenima receives from HIM results from classification, and use them for update user interface and preparing new set of stimuli (or other activities).

User Interface manages goal-based target selection. It is composed by four logical modules:

- Primary module: Starting module that control the overall subsystem. It calls Supervisor module for setting up specific parameters of selection and navigation, and manages Selection and Navigation module execution and activity. It also loads all necessary resources and graphical elements of the interface.
- Selection module: module that control the destination selection paradigm. A central ring containing a cursor is presented to patient. In each edge of the ring a directional arrow represents one of the four possible destinations achievable at each iteration. Near each arrow a small picture representing the destination is also available. The four arrows blink in casual sequence, and their blinking represents the visual stimulation for the patient. He must express his cognitive act of concentration every time the arrow indicating the destination he decided to reach blinks. Each time a cognitive act of concentration is recorded and connected to a particular blink, the cursor

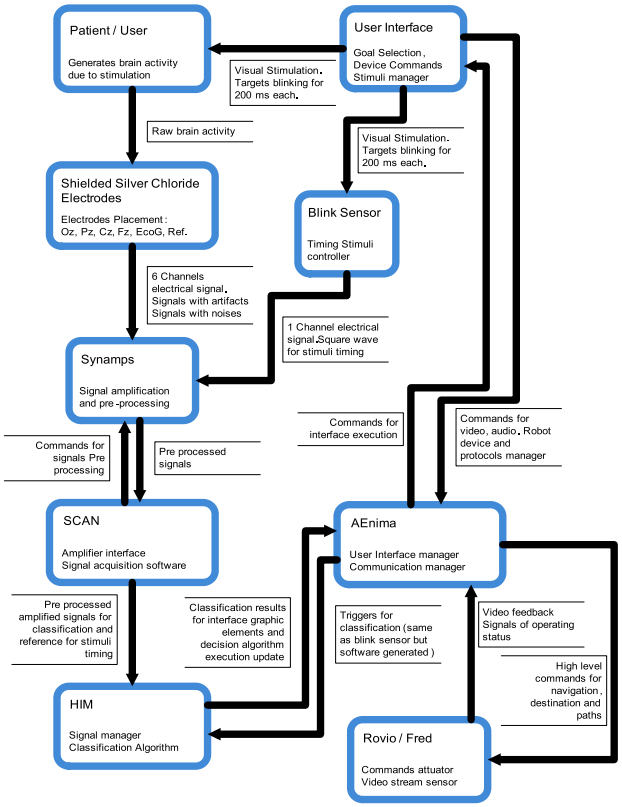


Fig. 6. NEVRAROS components and activities reference diagram

moves for one step in the corresponding arrow's direction. Once one of the edges is reached by the cursor, the Interface choose that destination as selected destination, sends to mobile device opportune commands for reach it in the real environment and calls Navigation module for execution. Since visual stimuli are limited in number, reaching no target determines no Navigation module calling nor commands to mobile device sending. In such this case, Selection module wait for a little time and then ask for a new selection session.

- Navigation module: module that control the navigation process of the mobile device and presents to user the video feedback in coming. A central window shows the video stream from the mobile device. Upper indicators shows overall progress in the path. Once mobile device reach selected destination, a binary choice is offered to user, with the same paradigm of previous selection task. Here user can decide if reach a new target or else take a look again to current destination achieved. Once such this decision is made, Navigation module calls back Selection module for a new selection session.
- Supervisor module: this module is hidden from user interface, and is visible only from secondary screen dedicated to system supervisor. Here supervisor can tune the interface up for the navigation session, selecting which modality use for next session (learning, testing or real navigation), how many elementary steps the central cursor must do for reach external edges, and more. From here a blinking square is also selectable, if use of Blink sensor is expected.

Paths and targets are controlled by device robot itself. NEVRAROS only keeps some information about possibly targets and overall environment composition for updating graphic elements and proposing new target picture near stimulation arrows. So, robot device only receives from AEnima high-level commands for navigation (target-to-reach identity number). Together with the video stream, the robot also sends status messages concerning its execution.

VI. SYSTEM EVALUATION

First test sessions have brought useful information for characterize the system performance. At first, the system has been tested over a healthy patient, without utilization of the robotic device (and therefore operating solely on virtual environment). 8 training sessions and 4 testing sessions has been prepared, plus a subsequent session of free navigation, in order to test the quality of general communication system components. Subsequent tests on 2 healthy subjects were performed for evaluating

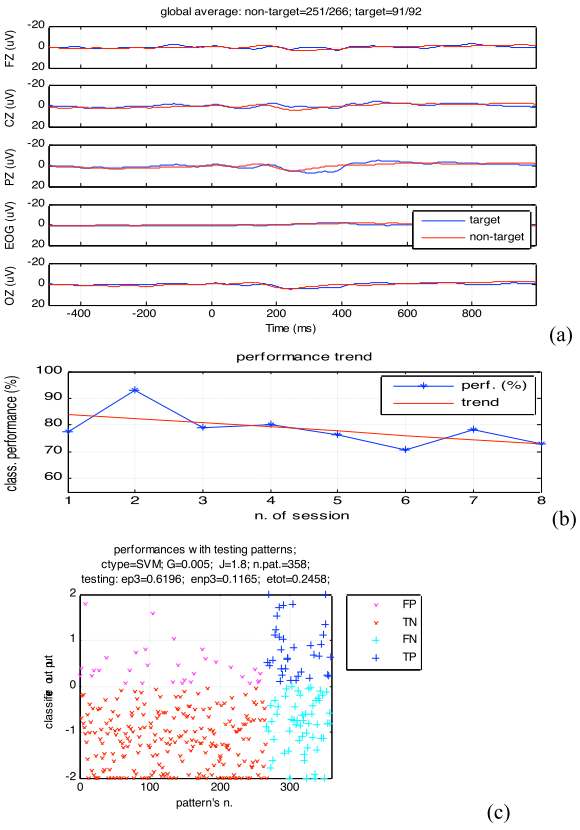


Fig. 7. NEVRAROS performances evaluation summary concerning one of the healthy patients. (a) traces average diagram; (b) classification performance trend; (c) classification output.

performance level of the algorithms for classification and selection of targets: for each of the two patients were prepared 6 to 8 sessions of learning and testing. Since these tests were prepared for check on the efficiency of algorithms for classification, the robotic support has not been used, (test limited to virtual environment). Latter evaluation rounds results show a classification accuracy that is close to 80%, and an robustness index greater than

TABLE I
EVALUATION RESULTS (HEALTHY SUBJECT)

BCI-skill	**(mean±std)**
Classification accuracy (performance %)	78.5±6.6
Transfer bit rate (bit/min)	7.99±4.51
Percentage of sessions successfully completed (%)	875
Training Number of Stimuli (TNS)	231
Performance trend (%/session)	-158

85%.

Further testing are planned with a patient suffering from ASL (subject:male, 50 y.o, white, italian). On that occasion the navigation system of the robot will be tested out of the lab, i.e. in the hospital, preparing a navigable environment adjacent to the patient's room. Moving such a patient to the laboratory would in fact be very difficult.

VII. CONCLUSIONS

The NEVRAROS system described in this paper proved to be effective and reliable in healthy subjects. Experiments with ASL patients are planned in the near future to asses the rehabilitation effectiveness of such robotic BCI system. The modularity of the described system and the open-source solutions adopted enable the continuous improvement of the system by integrating new modules and new solutions as soon as they will be developed by other research groups.

REFERENCES

[1] A. Chella, E. Pagello, E. Menegatti, K. Prifitis et al. *A BCI teleoperated museum robotic guide*. Presented at 2009 International Conference on Complex, Intelligent and Software Intensive Systems. 2009.

[2] S.G. Mason, G.E. Birch. *A General Framework for BCI Design*. IEEE Transaction of Neural Systems and Rehabilitation engineering. Vol. 11, No. 1, March. 2003.

[3] P. Perego et al. *A Home Automation Interface for BCI application validated with SSVEP protocol.* Presented at 4th Internationa Brain-Computer Interface Workshop 2008, Graz. 2008.

[4] E. W. Sellers, E. Donchin. *A P300-based brain–computer interface: Initial tests by ALS patients.* IEEE Transactions of Clinical Neurophysiology. Vol. 117. 2006.

[5] N. Birbaumer N. Ghanayim, T. Hinterberger, I. Iversen, B. Kotchoubey, et al. *A spelling device for the paralysed.* Nature. Vol. 398. 1999.

[6] F.Piccione, K. Priftis et al. *Amyotrophic Lateral Sclerosis patients are able to direct a computer screen cursor using a P300-based BCI*. Proceedings of 4th International Brain-Computer Interface Workshop and Training Course. Graz. 2008.

[7] U. Hoffmann, J.M. Vesin, T. Ebrahimi, K. Diserens. *An efficient P300-based brain-computer interface for disabled subjects*. Journal of Neuroscience Methods. Vol. 167, No. 1. 2008.

[8] J. R. Wolpaw, E. Donchin et al. *BCI Meeting 2005. Workshop on Signals and Recording Methods.* IEEE Transactions on Neural Systems and Rehabilitation Engineering. Vol. 14, No. 2. 2006.

[9] D. J. McFarland, J. R. Wolpaw. *BCI Operation of Robotic and Prosthetic Devices*. IEEE Computer. Vol. 41, No. 10. 2008.

[10] N. Birbaumer. *Brain-Computer Interfaces research, Coming of age.* Clinical Neurophysiology, Vol. 117, No. 3. 2006.

[11] T.W. Berger, J.K. Chapin et al. *Brain-Computer Interfaces, an international assessment of research and developments trends*. Springer. 2008.

[12] N. Birbaumer, J.R. Wolpaw et al. *Brain–Computer Communication Unlocking the locked-in.* Institute of Medical Psychology and Behavioral Neurobiology, University of Tübingen, German. 2001.

[13] Y. Wang, X. Gao. *Brain–Computer Interfaces Based on Visual Evoked Potentials.* IEEE Engineering in Medicine and Biology Magazine. Vol. 27. No. 5. 2008.

[14] J.R. Wolpaw, N. Birbaumer, D. McFarland, G.Pfurtscheller, T. Vaughan. *Computer-interfaces for communication and control.* Clinical Neurophysiology. Vol. 113. 2002.

[15] C.J. Bell et al. *Control of a Humanoid Robot by a Noninvasive Brain-Computer Interface in Humans.* Jounral of Neural Engineering. Vol. 5, No. 2. 2008.

[16] D.J. McFarland et al. *Emulation of Computer Mouse Control with a Noninvasive Brain-Computer Interface.* Journal of Neural Engineering. Vol. 5, No. 2. 2008.

[17] D.J. McFarland, W.A. Sarnacki, and J.R. Wolpaw. *Electroencephalographic (EEG) Control of Three-Dimensional Movement.* Society for Neuroscience Abstract. 2008.

[18] L. Tonin, E. Menegatti. *Integrazione di un sistema BCI ed un robot olonomo.* Padova. 2008.

[19] J.R. Millan et al.. *Noninvasive Brain-Actuated Control of a Mobile Robot by Human EEG.* IEEE Transactions for Biomedical Engineering. Vol. 51, No. 6. 2004.

[20] F. Piccione, F. Giorgi, P. Tonin, K. Priftis, S. Giove, S. Silvoni, G. Palmas, F. Beverina. *P300- based brain computer interface: Reliability and performance in healthy and paralised participants.* Clinical Neurophysiology. Vol. 117, No. 3. 2006.

[21] G. Pfurtscheller, G. R. Müller-Putz et al. *Rehabilitation with Brain-Computer Interface Systems.* IEEE Computer Society. 2008.

[22] Bruce H. Dobkin. *The Clinical Science of Neurologic Rehabilitation,* 2nd edition. Oxford University Press. 2003.

[23] J.R. Wolpaw, G.E. Loeb, B.Z. Allison, E. Donchin, et al. *BCI Meeting 2005, Workshop on Signals and Recording Methods.* IEEE Transactions on Neural Systems and Rehabilitation Engineering. Vol. 14. No 2. 2006.

[24] Galán, F., Nuttin, M., Lew, E., Ferrez, P.W., Vanacker, G., Philips, J., and Millán, J. del R.. *A Brain-Actuated Wheelchair: Asynchronous and Non-Invasive Brain-Computer Interfaces for Continuous Control of Robots.* Clinical Neurophysiology, 119:2159–2169. 2008.

[25] Sutton, S., Braren, M., Zubin, J., & John, E. R. *Evoked-Potential Correlates of Stimulus Uncertainty.* Science, 150(3700), 1187-1188. 1965.

[26] M. Sakai, H. Ishita, Y. Ohshiba, W. Chen, D. Wei. *P300 Detection for Brain-Computer Interface from Electroencephalogram Contaminated by Electrooculogram.* Appears in: Computer and Information Technology. CIT '06. The Sixth IEEE International Conference on. Publication Date: Sept. 2006, on page(s): 256 - 256. 2006.

[27] F. Piccione, F. Beverina, G. Palmas, S. Silvoni. *User adaptive BCIs. SSVEP and P300*. 2003 Psychology Journal,1(4). 2003.

Proceedings of the IEEE ICRA 2010
Workshop on Multimodal Human-Robot Interfaces
Anchorage, Alaska, May, 2010

Safe Human-Robot Interaction based on a Hierarchy of Bounding Volumes

J. A. Corrales, F. Torres and F. A. Candelas

Abstract—**This paper presents a novel human-robot interaction system which guarantees the safety of human operators which cooperate with robotic manipulators. In particular, this system computes on realtime the minimum distance between the human operator and the robotic manipulator and modify the trajectory of the robot accordingly. The system is integrated by two elements: a human tracking system and a hierarchy of bounding volumes. The tracking system combines the measurements of a motion capture system and an UWB localization system by a modified Kalman filter and thus the movements of the human are registered with accuracy. The hierarchy of bounding volumes covers the bodies of the human and the robot so that the minimum distance between them can be computed efficiently. A new distance computation algorithm based on this hierarchy has been developed. Finally, the system has been applied on a real task.**

I. Introduction

HUMAN-ROBOT interaction is becoming more and more widespread in robotics because of the benefits of combining the precision and the repeatability of robots with the intelligence and the dexterity of humans. This cooperation between humans and robots is mainly applied in service robotics applications [1] but it is scarcely used in industries. In industrial environments, robotic manipulators are usually isolated in fenced workspaces where humans do not enter in order to avoid collisions [2]. Therefore, industrial tasks cannot benefit from the human-robot cooperation. In order to overcome this limitation, the current paper develops a new human-robot interaction system which avoids the risk of collision by tracking precisely not only robotic manipulators but also human operators who collaborate in the task. This human-robot interaction system is composed of two main components: a human tracking system and a hierarchy of bounding volumes. The human tracking system registers with high precision all the movements of the human operator while the hierarchy of bounding volumes covers the bodies of the human and the robot so that the minimum distance between them can be computed efficiently.

The human tracking system integrates two subsystems: an inertial motion capture system and an Ultrawideband (UWB) localization system. Their measurements are combined by a modified Kalman filter which obtains not only precise relative rotation measurements between the limbs of the body (such as previous systems [3][4] but also a precise global positioning of these limbs in the workspace. In addition, the positions of the robotic manipulator's links are computed by forward kinematics from the joint angles obtained from the robot controller. Thereby, the tracking system enables the human-robot interaction system to find the spatial relations between the human and the robot on realtime. In particular, the measurements from the tracking system can be used to compute a precise approximation of the human-robot distance and modify the robot's movements accordingly.

However, the tracking system considers the human and the robot as wire skeletons and do not take into account the real dimensions of their limbs and links. In order to develop a more realistic model of them, these skeletons have been covered by a set of bounding volumes which represent the surface of their bodies. A three-level hierarchy of bounding volumes has been implemented to improve the efficiency of the human-robot distance computation in comparison to previous systems [5][6]. Each level is composed of a different group of bounding volumes, which cover the human and robot bodies more precisely than the previous level but increases the required number of pairwise distance tests. A new minimum distance computation algorithm has been developed in order to reduce the number of pairwise distance tests by combining the three levels of this hierarchy of bounding volumes.

This paper presents the developed human-robot interaction system. Section 2 describes the components and the fusion algorithm of the human tracking system. Section 3 describes the hierarchy of bounding volumes and the implemented human-robot distance computation algorithm. In section 4, the human-robot interaction system is applied on a real task where the human helps the robot to disassembly a small electrical appliance. Finally, section 5 presents the conclusions and the future work.

Manuscript received March 7, 2010. This work was supported in part by the Spanish Ministry of Education under grant AP2005-1458 and by the Spanish Ministry of Science and Innovation under research projects DPI2005-06222 (Design, Implementation and Experimentation of Intelligent Manipulation Scenarios for Automatic Assembly/Disassembly Applications) and DPI2008-02647 (Intelligent Manipulation through Haptic Perception and Visual Servoing by Using and Articulated Structure situated over a Robotic Manipulator).

J. A. Corrales is with the Physics, Systems Engineering and Signal Theory Department, University of Alicante, Spain (phone: +34-965-909-491; fax: +34-965-909-750; e-mail: jcorrales@ua.es).

F. Torres is with the Physics, Systems Engineering and Signal Theory Department, University of Alicante, Spain (e-mail: fernando.torres@ua.es).

F. A. Candelas is with the Physics, Systems Engineering and Signal Theory Department, University of Alicante, Spain (e-mail: francisco.candelas@ua.es).

II. HUMAN TRACKING SYSTEM

A. Components of the Human Tracking System

The human tracking system integrates an inertial motion capture system and a UWB localization system. The inertial motion capture system [7] is composed of 18 IMUs (Inertial Measurement Units) which are attached to the main body parts of the human operator. Each IMU contains 3 MEMS (Micro-Electro-Mechanical Systems) gyroscopes, 3 accelerometers and 3 magnetometers whose measurements are combined [8] in order to obtain the orientation (relative rotation angles) of the limb to which the IMU is attached. These rotation measurements are applied over the bones of a skeleton (see Fig. 1a) which represents the structure of the human's body. Similarly, the joint angles of the 7 D.O.F robotic manipulator obtained from its controller are also applied over a skeleton (see Fig. 1b).

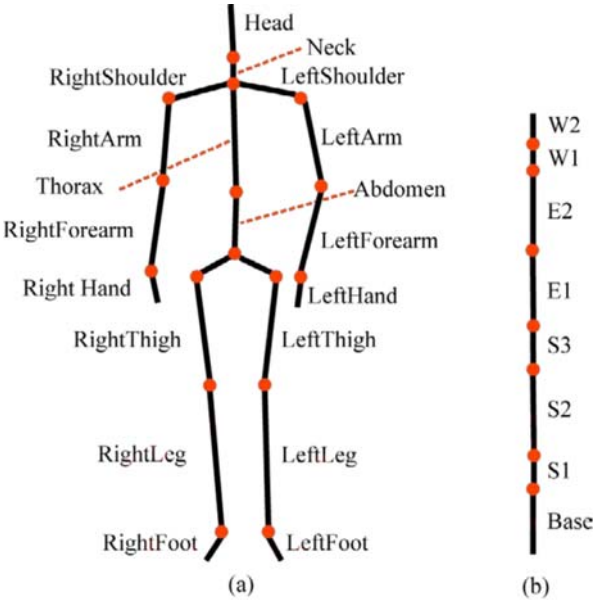


Fig. 1. Skeletal structures obtained by the tracking system for: (a) the human operator and (b) the 7 D.O.F robotic manipulator.

Whereas the robotic manipulator is fixed at a predefined position in the environment, the human operator can move around the workplace. Therefore, not only the relative rotation angles between the limbs are required, but also the global position of the human has to be known. The inertial motion capture system determines the human's global position by applying a foot step extrapolation algorithm to the legs' rotation data. Nevertheless, this algorithm sometimes does not detect steps correctly and accumulates some errors. In order to solve this problem, an additional localization system based on UWB signals has been used.

The UWB localization system [9] is composed of five devices: a small tag which is carried by the human operator and four sensors which are installed at fixed positions of the workplace. These four sensors compute the position of the human by triangulating the UWB pulses which are sent by the tag. Finally, the tracking system combines these positions measurements from the UWB system with the measurements from the inertial motion capture system in order to localize precisely the human operator.

B. Inertial-UWB Fusion Algorithm

Although the UWB localization system provides a precise positioning of the human operator, its sampling rate is much smaller than the motion capture system's frequency (5-10Hz and 30-120Hz, respectively). Therefore, the global position measurements of the motion capture system should be used between each pair of UWB measurements in order to keep a suitable sampling rate. This combination of measurements is implemented by a fusion algorithm based on a Kalman filter which is shown in Table I.

TABLE I
INERTIAL-UWB FUSION ALGORITHM

01: Initialize Kalman Filter's parameters: $\mathbf{P}_1$, **Q, R**.
02: Initialize ${}^{U}\mathbf{T}_{G}$ with the first two measurements: ${}^{G}\mathbf{p}_1$, ${}^{U}\mathbf{p}_2$
03: ***for each*** measurement $\mathbf{p}_t$
04: ***if*** $\mathbf{p}_t$ is from the inertial motion capture system *G*
05: ***if*** $\mathbf{p}_{t-1}$ is from the UWB localization system *U*
06: Recalculate ${}^{U}\mathbf{T}_{G}$ from $\mathbf{p}_t$ and $\mathbf{x}_{t-1}$
07: ***end if***
08: Transform $\mathbf{p}_t$ from *G* to *U* by applying ${}^{U}\mathbf{T}_{G}$
09: $[\mathbf{x}_t, \mathbf{P}_t]$ = KalmanFilterPrediction $(\mathbf{p}_t, \mathbf{P}_{t-1}, \mathbf{Q})$
10: Store $\mathbf{x}_t$ as position estimate for time *t*.
11: ***else if*** $\mathbf{p}_t$ is from the UWB localization system *U*
12: $[\mathbf{x}_t, \mathbf{P}_t]$ = KalmanFilterCorrection $(\mathbf{p}_t, \mathbf{P}_{t-1}, \mathbf{R})$
13: Store $\mathbf{x}_t$ as position estimate for time *t*.
14: ***end if***
15: ***end for***

The state of the implemented Kalman filter is composed by the 3D coordinates $\mathbf{x}_t=(x_t, y_t, z_t)$ of the global position of the human operator with respect to coordinate system *U* of the UWB system. Each time a measurement $\mathbf{p}_t$ from one of the tracking systems is registered; one step of the Kalman filter is performed in order to obtain the state estimate $\mathbf{x}_t$ and its corresponding error covariance $\mathbf{P}_t$. In particular, when a measurement from the motion capture system is registered (line 4, Table I), it is transformed to the *U* frame by applying the matrix ${}^{U}\mathbf{T}_{G}$ (line 8, Table I) and then it is used as input for the prediction step of a standard Kalman filter (line9, Table I) together with the previous error covariance $\mathbf{P}_{t-1}$ and the mean error covariance **Q** of the inertial motion capture system. When the measurement $\mathbf{p}_t$ is from the UWB localization system, it is used as input to the correction step of the Kalman filter (line 12, Table I) together with $\mathbf{P}_{t-1}$ and the mean error covariance **R** of the UWB system.

In addition, each time the correction step of the Kalman filter is executed, the transformation matrix ${}^{U}\mathbf{T}_{G}$ is recalculated through the computed position estimate $\mathbf{x}_{t-1}$ and the next measurement $\mathbf{p}_t$ from the motion capture system (line 6, Table I). Since this transformation matrix is applied to every measurement from the motion capture system, the error accumulated until this moment by the motion capture system is corrected.

III. Hierarchy of Bounding Volumes

A. Components of the Hierarchy of Bounding Volumes

As stated above, the tracking system considers the bodies of the human operator and the robot as linear skeletons but it does not take into account the real dimensions of the surfaces which cover the bones. Therefore, it is necessary to model these surfaces in order to compute more precisely the human-robot distance. Although a mesh of polygons is one of the most standard and detailed representations, it is not suitable for realtime distance computation due to its high computational cost. A representation based on bounding volumes is more suitable provided that it fulfills two requirements: thigh fitting to the body and efficient distance computation. Previous similar human-robot interaction systems [5][6] develop bounding volumes approaches based on spheres due to their inexpensive distance computation. Nevertheless, one sphere per limb does not fit tightly the bodies of humans and robots and thus these sphere-based approaches need to increase substantially the number of bounding volumes per limb. Unfortunately, this increase in the number of required spheres reduces the performance of the distance computation.

This paper presents a new approach based on Sphere-Swept Lines (SSLs) which overcomes these limitations of previous spherical models. On the one hand, a SSL is a bounding box volume obtained from the Minkowski sum of a sphere and a segment [10], which fits more tightly the human's and robot's bodies with only one bounding volume per limb. On the other hand, the computational cost of the distance computation between two SSLs is quite small because it is based on the distance computation between the inner segments of the SSLs [11]. Since this geometric modeling involves 18 SSLs for the human operator and 8 SSLs for the robotic manipulator, 144 pairwise distance tests are required. A three-level hierarchy of bounding volumes is proposed for each agent (human and robot) in order to reduce the number of pairwise distance tests and improve the computational efficiency of the distance computation.

The first level of this hierarchy is composed by one global AABB (Axis-Aligned Bounding Box) for each agent. The second level of the hierarchy is composed by a set of local AABBs which cover the main limbs of their bodies. Finally, the third level of the hierarchy covers each bone of their skeletons (see Fig. 1) with a SSL. Table II shows the components of the hierarchy of bounding volumes which cover the human operator. In particular, the second level of this hierarchy is composed by 5 AABBs and the third level contains 18 SSLs. Table III shows the components of the hierarchy which cover the robotic manipulator. In particular, the second level of this hierarchy contains 3 AABBs and the third level contains 8 SSLs.

TABLE II
Hierarchy of Bounding Volumes for the Human

Level 1	Level 2	Level 3
Global AABB	Left Lower Limb AABB	Left Thigh SSL Lef tLeg SSL Left Foot SSL
	Right Lower Limb AABB	Right Thigh SSL Right Leg SSL Right Foot SSL
	Torso-Head AABB	Abdomen SSL Thorax SSL Neck SSL Head SSL
	Left Upper Limb AABB	Left Shoulder SSL Left Arm SSL Left Forearm SSL Left Hand SSL
	Right Upper Limb AABB	Right Shoulder SSL Right Arm SSL Right Forearm SSL Right Hand SSL

TABLE III
Hierarchy of Bounding Volumes for the Robot

Level 1	Level 2	Level 3
Global AABB	Base AABB	Base SSL S1 SSL
	Arm AABB	S2 SSL S3 SSL
	Forearm AABB	E1 SSL E2 SSL W1 SSL W2 SSL

As shown in Tables II and III, each level of the hierarchy contains several bounding volumes of the immediately inferior level. Therefore, each level provides a more efficient distance computation than the following level but the calculated distance is less accurate. This fact implies that the upper levels (levels 1 and 2) will be used when the human and the robot are far away from each other and no accurate distance computation is needed. However, when the human and the robot collaborate too close, the third level is compulsory because its bounding volumes provide a more precise distance value. In order to reduce the number of pairwise tests which are performed when the third level is applied, a new distance computation algorithm which combines all the levels of these hierarchies has been developed and explained in the following section.

B. Human-Robot Distance Computation

The algorithm implemented for the computation of the human-robot distance combines the three levels of the hierarchy of bounding volumes described above with two main goals. On the one hand, it calculates the human-robot distance with the sufficient degree of accuracy required by the task. On the other hand, it optimizes the number of pairwise distance tests so that the final performance of the algorithm is maximum. Table IV presents a pseudo-code summary of the algorithm's implementation.

The selection of one level of the hierarchy of bounding volumes for the distance computation depends on two

distance threshold values (**DIST$_{12}$** and **DIST$_{23}$**) which identify the required degree of accuracy. Firstly, the human-robot distance is computed from the two global AABBs of the first level (line 3, Table IV). If this distance is greater than the threshold **DIST$_{12}$**, no further computation is needed and the AABB distance **mdist1** is used as the human-robot distance (line 5, Table IV).

In other cases, the algorithm computes the AABBs of the second level (line 7, Table IV) by looking for the maximum and minimum coordinates of the contained bones of the tracking system's skeletons and adding to them the maximum radius of the contained SSLs. Afterwards, all the distances **dist2[]** between each pair of AABBs are computed (line 8, Table IV) and sorted out in ascending order (line 9, Table IV). The minimum value **mdist2** of all these distances (line 10, Table IV) is used as the final human-robot distance if it is greater than the threshold **DIST$_{23}$** (line 12, Table IV).

Finally, if the computed distance **mdist2** is smaller than **DIST$_{23}$**, the third level of the hierarchy is needed because the human and the robot are too close to each other. The algorithm generates the SSLs (lines 19-20, Table IV) contained by the two closest AABBs of level 2 which are in the first position of the array **dist2[]** since it is ordered in ascending order. If the minimum distance **mdist3** (line 21, Table IV) between these SSLs is smaller than the distance between the following AABBs in **dist2[]**, this value is used as the final human-robot distance (line 23, Table IV). If this condition is not verified, the algorithm will continue looking for the closest SSLs by visiting each element of the **dist2[]** array and generating the corresponding SSLs. Thereby, this algorithm calculates the minimum human-robot distance between the SSLs and avoids executing all the pairwise tests in most cases.

TABLE IV
HUMAN-ROBOT DISTANCE COMPUTATION ALGORITHM

```
01: Initialize distance thresholds: DIST12, DIST23
02: Generate AABBs of Level 1: AABB1H, AABB1R
03: mdist1= MinimumDistance(AABB1H, AABB1R)
04: if (mdist1 > DIST12)
05:   minDist= mdist1
06: else
07:   Generate AABBs of Level 2: AABB2H[], AABB2R[]
08:   dist2[]= PairwiseDistance(AABB2H[], AABB2R[])
09:   dist2[]= SortInAscedingOrder(dist2[])
10:   mdist2= MinimumValue(dist2[])
11:   if (mdist2 > DIST23)
12:     minDist= mdist2
13:   else
14:     mdist3= FLOAT_MAX_VALUE
15:     for each element i in dist2[]
16:       if (mdist3 < dist2[i])
17:         break
18:       end if
19:       Generate SSLs inside AABB2H[i]: SSL3H[]
20:       Generate SSLs inside AABB2R[i]: SSL3R[]
21:       mdist3= MinimumDistance(SSL3H[], SSL3R[])
22:     end for
23:     minDist= mdist3
24: end if
25: return minDist
```

IV. EXPERIMENTAL RESULTS

The human tracking system and the hierarchy of bounding volumes are combined in order to build a human-robot interaction system which enables the development of safe tasks where humans and robots collaborate. The tracking system provides a realtime localization of all the bones of the human and the robot. These bones are covered by the hierarchy of bounding volumes in order to obtain an efficient and precise estimate of the minimum human-robot distance. This distance value can be used by the human-robot interaction system in order to modify the behaviour of the robot when a risk of collision is detected.

In order to verify the correctness of the proposed system, it has been applied on a real task where a human operator and a 7 D.O.F *Mitsubishi PA-10* robotic manipulator cooperate in the disassembly process of a small electric appliance (a fridge). The human operator wears the *GypsyGyro-18* inertial motion capture suit [7] and an *Ubisense* UWB tag [9] which compose the human tracking system.

In this task, the robot removes the screws from the rear lid of the fridge and leaves them inside a storage box. Meanwhile, the human operator opens the door of the fridge and empties its contents. Thereby, the repetitive subtasks are performed by the robot while the subtasks which require more intelligence and dexterity are performed by the human. Furthermore, the human-robot interaction system computes on realtime the minimum human-robot distance and activates a safety strategy if this distance is smaller than a threshold (1m). This strategy stops the current trajectory of the robot and moves the robot away from the human so that the safety distance is kept.

Fig. 2 shows the evolution of one execution of the developed disassembly task. Each subfigure shows a photograph of the real scenario and a 3D representation of the corresponding SSLs which cover the skeletons of the human and the robot. Figs. 2a and 2b depict how the robot begins to follow the trajectory towards the rear lid of the fridge to remove the screws. However, when the human approaches the fridge, the human-robot distance goes below the safety threshold and the safety strategy is activated. The robot moves away from the human (Fig. 2c) in order to maintain the safety distance. Meanwhile, the human operator opens the fridge's door (Fig. 2c) and extracts all its contents (Fig. 2d) in order to carry them to a storage box which is outside the workplace. When the human operator moves away and the human-robot distance is again bigger than the safety threshold, the safety behaviour is deactivated and the robot continues tracking the original trajectory (Figs. 2e and 2f) to unscrew the rear lid of the fridge.

Fig. 3 depicts the evolution of the computed human-robot distance along the development of the task. From instant 3.6s to instant 7.4s, the human operator approaches the fridge and the human-robot distance is reduced until it goes below the safety threshold. Then, the safety strategy is

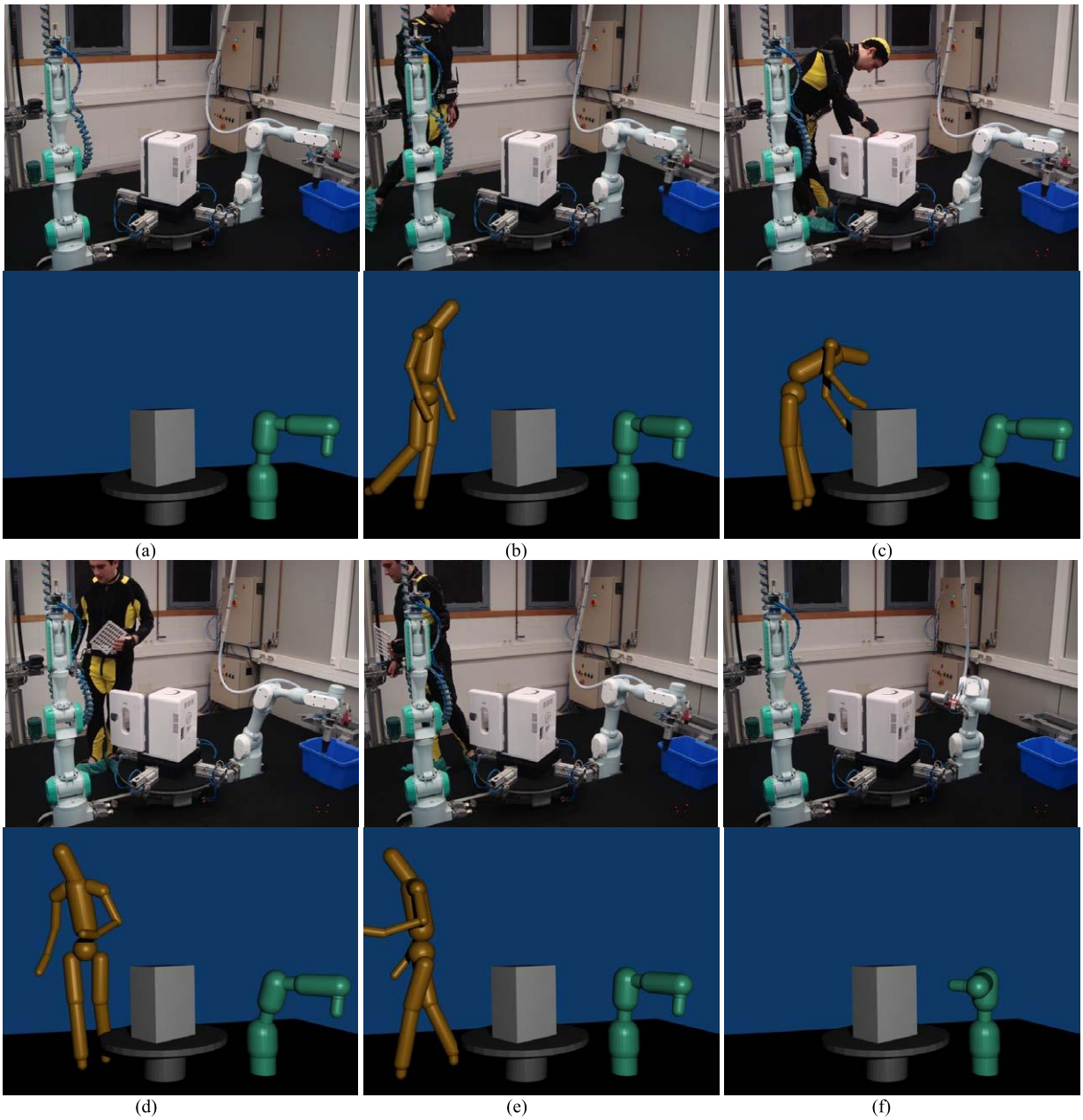


Fig. 2. Evolution of the disassembly task where the human and the robot collaborate with the surveillance of the human-robot interaction system.

activated and the robot maintains the safety distance between instants 7.4s and 18.3s. Afterwards, the human-robot distance is again greater than the safety threshold when the human takes away the fridge's contents to an external storage box. The normal behaviour of the robot manipulator is taken up again. The thresholds for the distance computation algorithm have been set to the following values in this task: $DIST_{12}$ = 2m and $DIST_{23}$ = 1m.

Fig. 4 shows a histogram of the number of pairwise distance tests which are executed during the task. In the 76.8% of the executions of the distance algorithm, the number of performed pairwise tests is less than 20. Between 20 and 80 pairwise tests are required in 12% of cases. Finally, only in 11.2% of cases, the distance algorithm needs

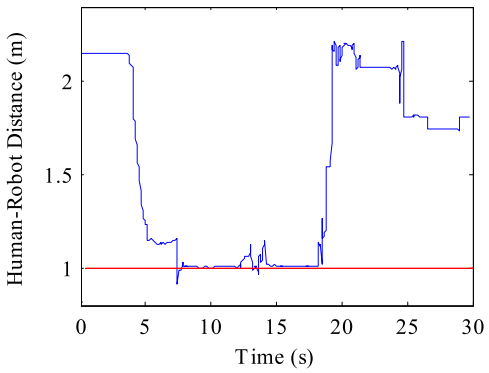


Fig. 3. Evolution of the human-robot distance during the development of the disassembly task.

to perform more than 80 pairwise tests. These results prove that the developed algorithm reduces substantially the number of required pairwise distance tests in comparison with a simple bounding volume approach without any kind of hierarchy which will always need 144 tests for the SSLs.

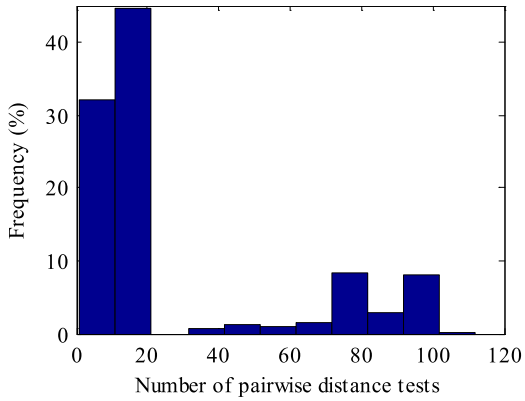


Fig. 4. Histogram of the number of pairwise distance tests which are required to compute the human-robot distance.

V. CONCLUSIONS

This paper presents a new human-robot interaction system which computes the distance between human operators and robotic manipulators that work together in the same workspace. The main goal of the system is to adapt the movements of the robots to the movements of the humans and thus any collision between them is avoided. Therefore, this system needs to track on realtime all their movements. The movements of the robot are easily registered by obtaining its joints' angles from the controller. In the case of the human operator, a tracking system which registers the movements of all the human's limbs is required. This human tracking system has been implemented by fusing the measurements of an inertial motion capture system and a UWB system. The inertial motion capture system provides the relative rotation values between the human's limbs. The UWB system is used to correct the global position measurements of the inertial motion capture system by applying a modified Kalman filter. This fusion algorithm is based on a new approach which relates the two steps of the filter (prediction and correction) with the complementary features of these systems (the high sampling rate of the inertial motion capture system and the localization accuracy of the UWB system).

The ends of the human's limbs are localized precisely with this tracking system and the ends of the robot's limbs are also localized by applying forward kinematics to the controller's joint angles. Nevertheless, a group of bounding volumes is required to model the surface of the human's and robot's bodies and get a better estimate of the human-robot distance efficiently. This paper presents a novel hierarchy of bounding volumes which integrates three levels composed by AABBs and SSLs. This hierarchy in conjunction with the proposed distance computation algorithm reduces the number of pairwise tests between bounding volumes which need to be completed before the distance is obtained. Finally, this distance computation algorithm has been implemented for a real human-robot interaction tasks and the improvements in the distance computation performance have been quantified.

The calculated human-robot distance has been used as the trigger for a safety strategy which stops the robot tracking the normal trajectory and creates a new one which moves the robot away from the human. In future research, the authors will develop more complex strategies which avoid stopping the normal robot's trajectory.

REFERENCES

[1] S. A. Green, M. Billinghurst, X. Chen and J. G. Chase, "Human-robot collaboration: A literature review and augmented reality approach in design," *Int. J. Adv. Robot. Syst.*, vol. 5, pp. 1-18, 2008.

[2] R. M. Ahmed, A. V. Ananiev and I. G. Kalaykov, "Compliant motion control for safe human robot interaction," in *Robot Motion and Control*, vol. 396/2009, M. Thoma, F. Allgöwer, M. Morari, Eds., Berlin: Springer, 2009, pp. 265-274.

[3] D. Vlasic, R. Adelsberger, G. Vannucci, J. Barnwell, M. Gross, W. Matusik and J. Popovic, "Practical motion capture in everyday surroundings," *ACM Trans. Graph.*, vol. 26(3), 2007.

[4] D. Roetenberg, P. J. Slycke and P. H. Veltink, "Ambulatory position and orientation tracking fusing magnetic and inertial sensing," *IEEE Trans. Biomed. Eng.*, vol. 54(5), pp. 883-890, 2007.

[5] L. Balan and G. M. Bone, "Real-time 3D collision avoidance method for safe human and robot coexistence," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, 2006, pp. 276-282.

[6] B. Martinez-Salvador, M. Perez-Francisco and A. P. Del Pobil, "Collision detection between robot arms and people," *J. Intellig. Robot. Syst.*, vol. 38(1), pp. 105-119, 2003.

[7] Animazoo, "GypsyGyro-18," [Online]. Available in: http://www.animazoo.com.

[8] E. Foxlin, "Inertial head-tracker sensor fusion by a complementary separate-bias Kalman filter", in *Proc. Virtual Reality Annual International Symposium*, 1996, pp. 185-196.

[9] Ubisense, "Ubisense v.1 Real-Time Location System," [Online]. Available in: http://www.ubisense.net.

[10] C. Ericson, *Real-time collision detection*, Elsevier, San Francisco, 2005.

[11] P. J. Schneider and D. H. Eberly, *Geometric tools for computer graphics*, Elsevier, San Francisco, 2003.

Proceedings of the IEEE ICRA 2010
Workshop on Multimodal Human-Robot Interfaces
Anchorage, Alaska, May, 2010

Hand Gesture Recognition for Human Robot Interaction in Uncontrolled Environments

Jong Lee-Ferng, Javier Ruiz-del-Solar, Mauricio Correa, Rodrigo Verschae

Abstract—**This article describes a robust and real-time hand gesture recognition system meant to allow a natural interaction with a service robot in dynamic environments. The proposed approach uses context information to detect hands robustly and in real time on a low-end processing unit (standard notebook). Static gestures are recognized using boosted classifiers, which are built using training techniques that decrease the burden on the human annotator, such as active learning and bootstrap. Dynamic gestures are recognized using a novel method that extracts geometrical features from the hand trajectory, thus avoiding explicit temporal variability analysis as in traditional Hidden Markov Models. Simultaneous gesture segmentation and recognition is carried out using a standard Naïve Bayes classifier for finding candidate subsequences that give high scores when matched to a gesture. The system's performance is validated on two applications involving Bender, a service robot: playing rock-paper-scissors and giving simple commands.**

Index Terms— **static hand gesture recognition, dynamic hand gesture recognition, human robot interaction, RoboCup @Home.**

I. Introduction

Hand gestures are extensively employed in human non-verbal communication. They allow to express orders, mood state, and to transmit some basic cardinal information. In some special situations they can be the only way of communicating, as for instance in the cases of deaf people communication (sign language), police's traffic coordination in the absence of traffic lights, and in general, communication in noisy environments.

Thus, it seems convenient that human-robot interfaces incorporate hand gesture recognition capabilities. Such interfaces allow building interfaces for disabled people, as well as implementing ubiquitous multimedia mobile control for social/personal robots. For instance, we would like to have the possibility of transmitting simple orders to personal robots using hand gestures. The recognition of hand gestures requires both hand detection and gesture recognition. Both tasks are very challenging, mainly due to the variability of the possible hand gestures (signs), and because hands are complex, deformable objects (a hand has more than 25 degrees of freedom, considering fingers, wrist and elbow joints) that are very difficult to detect in dynamic environments with cluttered backgrounds and variable illumination.

In this context, we are proposing a robust and real-time hand gesture recognition system to be used in the interaction with personal robots. We are especially interested in dynamic environments such as the ones defined in the *RoboCup @Home league* [15], with the following characteristics: variable illumination, cluttered backgrounds, (near) real-time operation, large variability of hands' pose and scale, and limited number of gestures (they are used for giving the robot some basic orders).

The developed system is able to recognize static and dynamic gestures, and its most innovative features include:

- The use of context information to achieve, at the same time, robustness and real-time operation, even when using a low-end processing unit (standard notebook), as in the case of humanoid robots. The use of context allows adapting continuously the skin model used in the detection of hand candidates, to restrict the image's regions that need to be analyzed, and to cut down the number of scales that need to be considered in the hand-searching and gesture recognition processes.

- The employment of boosted classifiers for the detection of faces and hands, as well as the recognition of static gestures. The main novelty is in the use of innovative training techniques –active learning and bootstrap–, which allow obtaining a much better performance than similar boosting-based systems, in terms of detection rate, number of false positives and processing time.

- The use of temporal statistics of the hand positions and velocities and a Bayes classifier to recognize dynamic gestures. This approach is different from the traditional ones, based on Hidden Markov Models.

The hand gesture recognition system has been adapted to be used in Bender [32], an innovative social robot that has been employed as personal robot for home environments [33], lecturer for school children [34], referee for robot soccer [35] and natural interface for Internet access [36].

Manuscript received March 7, 2010. This work was supported in part by CONICYT (Chile) under Grant FONDECYT 1090250.

J. Ruiz-del-Solar is with the Elec. Eng. Dept. and the Adv. Mining Tech. Center - Universidad de Chile (e-mail: jruizd@ing.uchile.cl).

R. Verschae is with the Elec. Eng. Dept. - Universidad de Chile (e-mail: rodrigo@verschae.org).

J. Lee-Ferng is with the Elec. Eng. Dept. - Universidad de Chile (e-mail: jongbor@gmail.com).

M. Correa is with the Elec. Eng. Dept. - Universidad de Chile (e-mail: mauricio.knight@gmail.com).

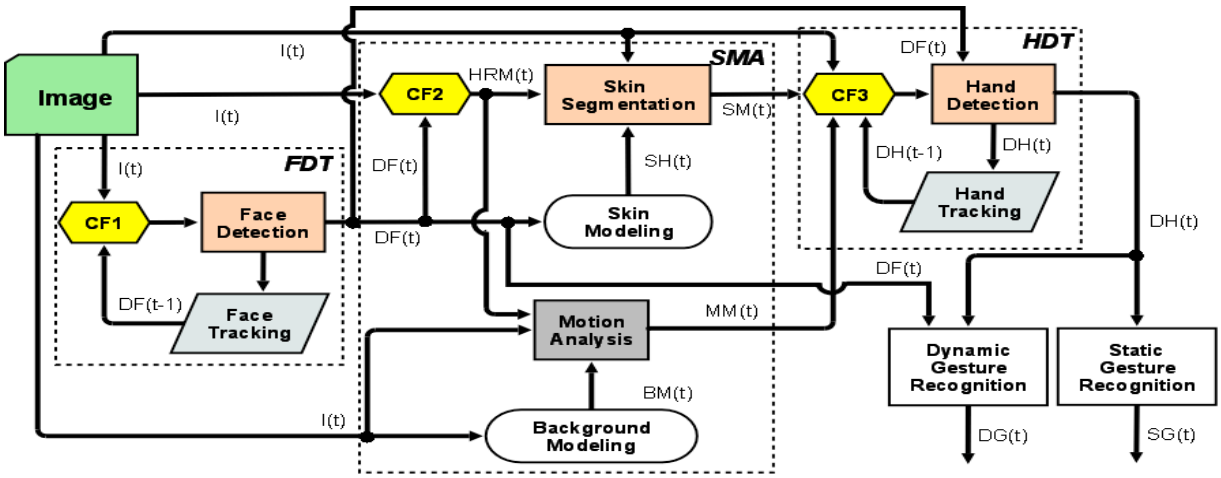


Fig. 1. Proposed hand gesture recognition system. CT*i*: *Context Filter i*. I: Image. DF: *Detected Face*. HRM: *Hand Region Mask*. SH: *Skin Histogram*. SM: *Skin Mask*: BM: *Background Model*. MM: *Motion Mask*. DH: *Detected Hand*. DG: *Dynamic Gesture*. SG: *Static Gesture*. **t**: *Frame index*. See main text for a detailed explanation.

Two applications of the hand gesture recognition abilities of the Bender robot are presented. In the first application, the static gesture module recognition is used to implement the rock-paper-scissors game. The application has been validated with several users in uncontrolled indoor environments. The obtained gesture recognition rate (rock, paper and scissors gestures) is 72.6%. As a second application, we have implemented a system to interact with the robot and give him simple orders in noisy environments. We have designed a set of dynamic gestures that allows moving the robot, waving its hand, and saying him "yes" and "no" (see fig. 3). The system has been validated with a database of 150 gestures, articulated by 5 different users. The obtained recognition rate is 78.5%.

This article is focused on the description of the static and dynamic gesture recognition approaches. In section II we present related work. In section III we present an overview of the gesture recognition system as a whole. The static and dynamic gesture recognition approaches are described in sections IV and V. Results of the application of these approaches are presented and analyzed in section VI. Finally, some conclusions of this work are given in section VII.

II. Related Work

Several hand detection and hand gesture recognition systems have been proposed. Early systems usually require markers or colored gloves to make the recognition easier. Second generation methods use low-level features as color (skin detection) [4][5], shape [8] or depth information [2] for detecting the hands. However, those systems are not robust enough for dealing with dynamic conditions; they usually require uniform background, uniform illumination, a single person in the camera view [2], and/or a single, large and centered hand in the camera view [5].

Boosted classifiers allow the robust and fast detection of hands [3][6][7]. In addition, the same kind of classifiers can be employed for detecting static gestures, as shown by Kolsch and Turk [3], who based their work on Viola and Jones' cascade of boosted classifiers[13]. Our main contribution over previous work are the use of a much powerful learning machine (nested cascade with boosted domain-partitioning classifiers), and the use of better training procedures, which increase the performance of the classifiers.

As for dynamic gestures, in this work, we are interested in purely visual recognition of gestures that are determined only by the trajectory of the hand (not its pose). Dynamic gesture recognition faces two challenges.

The first challenge is modeling the gesture itself, accounting for spatial and temporal variability. HMMs have become the predominant approach in dynamic gesture recognition systems[20]-[24][26]. Following [16], we use simple features such as hand position and hand velocity in order to represent the hands detected in each frame. However, our approach is based on a quite different strategy, which involves computing overall geometric and kinematic information that is independent of the length of the performed gesture. This draws comparisons to other techniques, such as Dreuw[30], Cui et al. [19], Motion History Images [18][25], Time Delay Neural Networks [27], and bag of words approaches [28].

The second challenge is to extract (segment) the gesture out of a continuous stream of hand movement information. Start and end points of a gesture are unknown beforehand, and the simplest approach to gesture segmentation involves using low-level information such as hand velocity, acceleration and angle variations [26][3][17][29]. A second category of methods performs segmentation and detection simultaneously, by finding intervals that match gesture models with high probability, as the CDP method in [16] or HMM-based methods in [31][20][23]. In our system we adopt both methods: gesture segmentation and recognition are achieved simultaneously by finding candidate subsequences that give high scores when matched to a gesture, while low level information –such as hand velocity and undetected (probably out of range) hands– is also used to detect boundaries.

The proposed approach relies on accurate hand detection and tracking as well as face detection and tracking. Face position and size are needed in order to gain invariance to

translation and scale. The hand and face detection module is based on Viola and Jones' cascade classifiers [13], while tracking is achieved with the mean-shift technique. Details can be found in previous works [10][11].

III. Hand Gesture Recognition System: System Overview

The hand gesture recognition system as a whole consists of five modules: Face Detection and Tracking (FDT), Skin Segmentation and Motion Analysis (SMA), Hand Detection and Tracking (HDT), Static Gesture Recognition, and Dynamic Gesture Recognition (see figure 1).

The FDT module is in charge of detecting and tracking faces. These functionalities are implemented using boosted statistical classifiers [11], and the *mean shift* algorithm [1], respectively. The information about the detected face (DF) is used as context in the SMA and HDT modules. Internally the CF1 (Context Filter 1) module determines the image area that has to be analyzed in the current frame for face detection, using the information about the detected faces in the past frame.

The SMA module determines candidate hand regions to be analyzed by the HDT module. The Skin Segmentation module uses a skin model that is adapted using information about the face-area's pixels (skin pixels). The module is implemented using the *skindiff* algorithm [9]. The Motion Analysis module is based on the well-known background subtraction technique. CF2 (Context Filter 2) uses information about the detected face and the human-body dimensions to determine the image area (HRM: Hand Region Mask) where a hand can be present in the image. Only this area is analyzed by the *Skin Segmentation* and *Motion Analysis* modules.

The HDT module is in charge of detecting and tracking hands. These functionalities are implemented using boosted statistical classifiers and the *mean shift* algorithm, respectively. CF3 (Context Filter 3) determines the image area where a hand can be detected in the image, using the following information sources: (i) skin mask (SM) which corresponds to a skin probability mask, (ii) motion mask (MM) that contains the motion pixels, and (iii) information about the hands detected in the last frame (DH: Detected Hand).

The Static Gesture Recognition module is in charge of recognizing static gestures. The module is implemented using statistical classifiers: a boosted classifier for each gesture class, and a multi-class classifier (C4.5 pruned tree [14]) that makes the final decision. The Dynamic Gesture Recognition module spots and recognizes dynamic gestures. This module computes temporal statistics of the hand positions and velocities, which are fed a Bayes classifier that recognizes the gesture.

IV. Static Gesture Recognition

In order to detect hands in the image, a nested cascade of boosted classifiers is applied on whichever skin blobs have been found. Static gestures are recognized using Boosted classifiers. Since it is difficult to build a generic hand detector, we have switched the problem by stipulating that the hand is first detected when performing any of the predefined gestures, and it is tracked afterwards using the mean-shift algorithm. Once the hand is being tracked, in order to determine which static gesture is being expressed, a set of gesture detectors is applied in parallel over the regions of interest delivered as an output of the tracking procedure. Each gesture detector is implemented using a cascade of boosted classifiers. We have implemented detectors for the following gestures: pointing, five, palm, fist and victory (figure 2).

Due to noise or gesture ambiguity, it could be the case that more than one gesture detector will have positive results for a given region of interest. In order to discriminate among these gestures, a multi-gesture classifier (J48 tree) is built and applied to the output of the gesture detectors as whole. Each gesture detector delivers the following attributes:

- *conf*: sum of the cascade confidence's values of windows where the gesture was detected (a gesture is detected at different scales and positions),
- *numWindows*: number of windows where the gesture was detected,
- *meanConf*: mean confidence value given by *conf*/*numWindows*, and
- *normConf*: normalized mean confidence value given by *meanConf*/*maxConf*, with *maxConf* being the maximum possible confidence that a window could get.

The gesture detectors are trained in a learning framework which seeks to decrease the cost of having a human expert who annotates training examples. To achieve this, we use two procedures. The *bootstrap procedure* strives to acquire negative training samples that look alike the target object (a gesture) by iteratively adding negative samples that have been misclassified by the classifier that has been trained up to the moment. The *active learning* procedure strives to acquire positive training samples using the system being built to guide the selection of new samples. In this procedure, a user is asked to perform the gesture of interest for a given time while moving the hand until training examples of the desired variability (illumination, background, rotation, scale, occlusion, etc.) are obtained. The human operator only has to verify that the detected hand gestures are correctly detected and adjust the alignment of the windows if necessary.

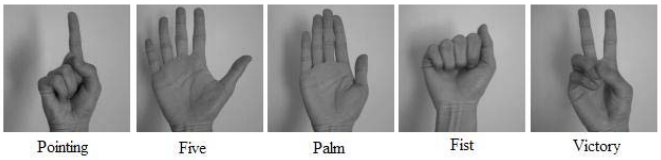


Fig. 2. Static gestures recognized by the system.

V. Dynamic Gesture Recognition

In this work, dynamic gestures are recognized (classified) using standard statistical classifiers. Considering that a given dynamic gesture is composed by a sequence of hand's positions and its corresponding dynamics, feature vectors that characterize both positions and dynamics are defined. Gesture segmentation (i.e., determination of the gesture start and end) and classification are carried out simultaneously, by finding gestures that have a high probability during many frames.

A. Representation

Each detected hand is represented as a vector (x, y, v_x, v_y, t), with (x,y) the hand's position, (v_x, v_y) the hand's velocity, and t the frame's timestamp. In order to achieve translation and scale invariance, coordinates (x,y) are measured with respect to the face, and normalized by the size of the face. Using this hand's vector, statistics (features) that characterize the subsequence of detections (a list of (x,y,v_x,v_y,t) vectors) are calculated. The components of the feature vector are:

- DELTA_X: difference between maximal and minimal position in the *x* axis.
- DELTA_Y: difference between maximal and minimal position in the *y* axis.
- SLOPE0, ..., SLOPE(*c-1)*: so called "direction codes", these features are extracted by approximating the slope of each of *c* segments of equal length in which the hand trajectory is divided. This approximation is expressed as one of eight possible values for each "basic" orientation in space (up, up-left, left, left-down, etc.).
- HIST2D00, ..., HIST2D(*n-1)(m-1)*: a binary image drawn on a *n*x*m* grid that spans the tightest bounding box that includes all hands detected in the observed frames, though some adjustments are allowed in order to minimize distortion of the gesture). The value of HIST2D*xy* is 1 if a hand has passed through cell (x,y), 0 otherwise.

In this work, the feature vector is composed of all the mentioned features (no feature selection is performed).

B. Classification of Segmented Gestures

Segmented gestures are characterized using the feature vector defined in the former section, and classified using a set of standard Naïve Bayes classifiers trained using a one-against-all decomposition. That is, for each gesture of interest *g*, we train a binary classifier using all instances of *g* as positive samples and every other sample in the training set as a negative sample.

C. Gesture Segmentation and Classification

The algorithm for online dynamic gesture recognition is divided in three stages:

1. Candidate generation: in which the aforementioned binary classifiers are continually applied to the incoming frame sequence. When the probabilistic score output by any of these classifiers, averaged along a fixed number of frames, is higher than a threshold, this module declares the existence of a gesture candidate.
2. Candidate evaluation: the gesture candidates generated in the previous stage are checked (compared with templates) in order to make sure they are sound candidates of the gestures they supposedly represent. Subgesture reasoning –discarding candidate gestures that are included in a larger gesture– is also performed in this stage.
3. Purpose evaluation: which examines the frame buffer (updated in stage 1) and the candidate list (updated in stage 2) in order to decide whether the user is still moving his hand in order to perform a gesture or if he has decided to end the gesture.

VI. Results

The developed system has been used in two applications. In the first application, the static gesture module recognition is used to enable Bender to play rock-paper-scissors. As a second application, we have implemented a system to interact with the robot and give him simple orders in environments with uncontrolled illumination and various users.

A. Static gesture recognition

We evaluated static gesture recognition as applied to implementing rock-paper-scissors (using gestures "fist", "palm" and "victory" shown in figure 2). We made a live test, with the system running while the users made gestures in front of the camera instead of using a video. This allows the user to have instantaneous feedback of the gesture recognition and adjusting his or her hand in order to achieve a better recognition. A total of 4 users participated in the test, of which 3 did not have any prior experience with the system. All of them were told to perform the gestures for rock, paper and scissors during 30 seconds each, and this was repeated 3 times. Users were told to slowly move the hand, so that it appeared against varying background objects with different illumination conditions and from different angles. The illumination conditions were those of a typical household. Table 1 shows a confusion matrix. The global recognition rate is 72.6%. While false positives do exist, this problem can be dealt with by determining which gesture is recognized more times in a given time interval. In figure 4 is shown an example of a human user playing rock-paper-scissors with Bender.

B. Dynamic gesture recognition

An uncontrolled environment includes variable illumination, multiple users and a dynamic background. To evaluate the performance of the dynamic gesture recognition system, we recorded a database which has two of these characteristics: uncontrolled illumination settings and five users. The background, however, is static, and including this factor is left for future experiments. Each user recorded three videos in which he or she executed the gestures in figure 3 in sequence, wearing short sleeves and taking the hand away after each gesture. The setting in this database is a realistic interaction with Bender: users stand in front of him, at the sight of the camera, and they can see themselves on the tablet PC that is installed on his chest. Most of the users were untrained and they were given minimal instructions about the speed and size with which the gestures should be performed. The users did not try out the gestures before the recording started, and the first attempt was, with few exceptions, the definitive attempt. The results are given in Table 2, where it can be seen that the mean recognition rate is 78.5%.

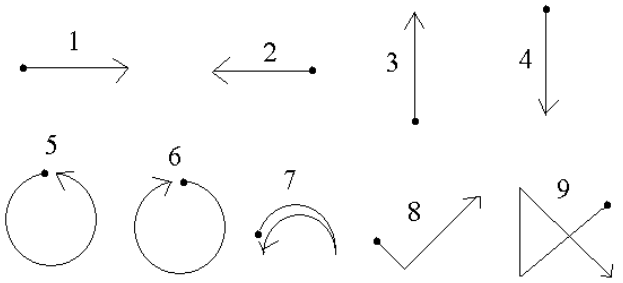


Fig. 3. Dynamic gestures recognized by the system: 1. RIGHT, 2. LEFT, 3. UP, 4. DOWN, 5. CCW, 6. CW, 7. WAVE, 8. CHECK, 9. CROSS.

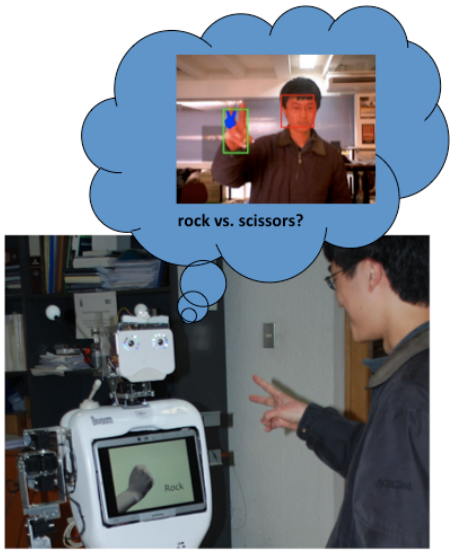


Fig. 4. Playing rock-paper-scissors with Bender.

VII. Conclusions

In this article we described a hand gesture recognition system that allows interacting with a service robot, in dynamic environments and in real-time. The system detects hands and static gestures using a cascade of boosted classifiers, and recognizes dynamic gestures by computing temporal statistics of the hand's positions and velocities, and classifying these features using a Bayes classifier. The support for dynamic environments comes mainly from the vision system, which uses context information to achieve robustness. The system performance is validated in two applications: static gestures used in rock-paper-scissors and dynamic gestures for giving commands to the robot. The performance allows for natural interaction despite illumination variability and multiple users; variable backgrounds have not been extensively tested yet. It is emphasized that many of the users that participated in the tests were inexperienced and given a minimal set of instructions on how to perform the gestures. The size of the video frames is 320x240 pixels, and the on board robot computer where the gesture recognition system runs is a standard notebook (Tablet HP 2710p, Windows Tablet SO, 1.2 GHz, 2 GB in RAM). Under these conditions, once the system detects the user's face, it is able to run at a variable speed of 4-8 frames per second.

Class\Predicted	Fist	Palm	Victory	Unknown	Detection rate
Fist	1098	71	9	71	87.9%
Palm	156	974	107	351	61.3%
Victory	61	167	1118	211	71.8%

Table 1. Results for recognition of static gestures used in rock, paper and scissors.

Gesture	Total	Correct	Inserted	Deleted	Substituted	% detected	% reliability
RIGHT	15	14	0	1	0	93.3	93.3
LEFT	15	15	0	0	0	100	100
UP	15	15	1	0	0	100	93.8
DOWN	15	12	2	3	0	80	70.6
CCW	15	12	0	1	2	80	80
CW	15	11	0	2	2	73.3	73.3
WAVE	15	8	0	6	1	53.3	53.3
CHECK	15	14	0	0	1	93.3	93.3
CROSS	15	5	0	0	10	33.3	33.3
Total	135	106	3	13	16	78.5	76.8

Table 2. Results for dynamic gesture recognition. A total of 15 instances were tested for each gesture. Inserted gestures are gestures that were detected despite not being executed in reality. Deleted gestures are those that failed to be recognized at all. Substituted gestures are gestures that were confused with other gesture. Reliability is computed as correct/(total+inserted), thus taking the inserted gestures into account.

REFERENCES

[1] D. Comaniciu, V. Ramesh, and P. Meer, Kernel-Based Object Tracking, *IEEE Trans. on Pattern Anal. Machine Intell.*, vol 25, no. 5, (2003) pp. 564 – 575.

[2] X. Liu and K. Fujimura, Hand gesture recognition using depth data, *Proc. 6th Int. Conf. on Automatic Face and Gesture Recognition*, (2004) pp. 529 – 534, Seoul, Korea.

[3] M. Kolsch, M.Turk, Robust hand detection, *Proc. 6th Int. Conf. on Automatic Face and Gesture Recognition*, (2004) pp. 614 – 619, Seoul, Korea.

[4] N. Dang Binh, E. Shuichi, T. Ejima, Real-Time Hand Tracking and Gesture Recognition System, *Proc. GVIP 05*, (2005) pp. 19-21 Cairo, Egypt.

[5] C. Manresa, J. Varona, R. Mas, F. Perales, Hand Tracking and Gesture Recognition for Human-Computer Interaction, *Electronic letters on computer vision and image analysis*, Vol. 5, Nº. 3, (2005) pp. 96-104.

[6] Y. Fang, K. Wang, J. Cheng, H. Lu, A Real-Time Hand Gesture Recognition Method, *Proc. 2007 IEEE Int. Conf. on Multimedia and Expo*, (2007) pp. 995-998

[7] Q. Chen, N.D. Georganas, E.M. Petriu, Real-time Vision-based Hand Gesture Recognition Using Haar-like Features, *Proc. Instrumentation and Measurement Technology Conf. – IMTC 2007*, (2007)Warsaw, Poland

[8] A. Angelopoulou, J. García-Rodriguez, A. Psarrou, Learning 2D Hand Shapes using the Topology Preserving model GNG, *Lecture Notes in Computer Science* 3951 (*Proc. ECCV 2006*), (2006) pp. 313-324

[9] J. Ruiz-del-Solar, and R. Verschae, Skin Detection using Neighborhood Information. *6th Int. Conf. on Face and Gesture Recognition* – FG 2004, (2004) pp. 463 – 468, Seoul, Korea.

[10] H. Francke, J. Ruiz-del-Solar, R. Verschae, Real-time Hand Gesture Detection and Recognition using Boosted Classifiers and Active Learning, *Lecture Notes in Computer Science* 4872 (*Proc. PSIVT 2007*), (2007) pp. 533-547.

[11] M. Correa, J. Ruiz-del-Solar, R. Verschae, J. Lee-Ferng, N. Castillo, Real-Time Hand Gesture Recognition for Human Robot Interaction. *Lecture Notes in Computer Science* 5949 (*RoboCup Symposium 2009*), (2010) pp. 46-57.

[12] R. Verschae, J. Ruiz-del-Solar, M. Correa, A Unified Learning Framework for object Detection and Classification using Nested Cascades of Boosted Classifiers, *Machine Vision and Applications*, Vol. 19, No. 2, (2008) pp. 85-103.

[13] P. Viola, M. Jones, Rapid object detection using a boosted cascade of simple features, *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, (2001) pp. 511 – 518.

[14] I.H. Witten and E. Frank (2005) "Data Mining: Practical machine learning tools and techniques", 2nd Edition, Morgan Kaufmann, San Francisco, 2005.

[15] RoboCup @Home Official website. Available in March 2010 in http://www.robocupathome.org/

[16] J. Alon, V. Athitsos, Q. Yuan, and S. Sclaroff, A Unified Framework for Gesture Recognition and Spatiotemporal Gesture Segmentation, *IEEE Trans. on Pattern Anal. Machine Intell.* Vol. 31, No. 9, (2009) pp. 1685-1699.

[17] M.K. Bhuyan, D. Ghosh, and P.K. Bora. Continuous hand gesture segmentation and co-articulation detection, *Lecture Notes in Computer Science 4338*, (2006) pp 564-575.

[18] A.F. Bobick, and J.W. Davis, The recognition of human movement using temporal templates. *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 23, No. 3, (2001) pp 257-267.

[19] Y. Cui, and J. Weng, Appearance-based hand sign recognition from intensity image sequences. *Comput. Vis. Image Underst.*, Vol. 78, No. 2, (2000) pp. 157-176.

[20] J.W. Deng, and H.T. Tsui, An HMM-based approach for gesture segmentation and recognition, *Proc. of the Int. Conf. on Pattern Recognition*, (2000) pp. 3683.

[21] S. Eickeler, A. Kosmala, and G. Rigoll, Hidden Markov model based continuous online gesture recognition, *Proc. of the 14th Int. Conf. on Pattern Recognition*, vol. 2, (1998) pp.1206-1208.

[22] D. Kim, and S. Jinyoung Song, Simultaneous gesture segmentation and recognition based on forward spotting accumulative HMMs, *Pattern Recogn*ition, Vol. 40, No. 11, (2007) pp. 3012-3026.

[23] H-K. Lee, and J-H. Kim, An HMM-based threshold model approach for gesture recognition, *IEEE Trans. Pattern Anal. Mach. Intell*., Vol. 21, No. 10, (199) pp. 961-973.

[24] S-W. Lee, Automatic gesture recognition for intelligent human-robot interaction, *Proc. of the 7th Int. Conf. on Automatic Face and Gesture Recognition*, (2006) pp 645-650.

[25] C. Shan, Y. Wei, X. Qiu, and T. Tan, Gesture recognition using temporal template based trajectories, *Proc. Int. Conf. on Pattern Recognition*, Vol. 3, (2004) pp. 954-957.

[26] T. Wang, H. Shum, Y. Xu, and N. Zheng, Unsupervised analysis of human gestures, *Proc. of the 2nd IEEE Pacific Rim Conf. on Multimedia*, (2001) pp. 174-181.

[27] M. Yang, N. Ahuja, and M. Tabb, Extraction of 2D motion trajectories and its application to hand gesture recognition, *IEEE Trans. Pattern Anal. Mach. Intell*., Vol. 24, No. 8, (2002) pp. 1061-1074.

[28] Z. Zhao. and A. Elgammal, Spatiotemporal pyramid representation for recognition of facial expressions and hand gestures, *Proc. 8th IEEE Int'l Conf. on Automatic Face and Gesture Rec.*, (2008).

[29] W. Kong, and S. Ranganath, Automatic hand trajectory segmentation and phoneme transcription for sign language, *Proc. 8th IEEE Int'l Conf. on Automatic Face and Gesture Rec.*, Washington, (2008).

[30] P. Dreuw, Appearance-Based Gesture Recognition. Diploma thesis, Rheinisch-Westfälische Technische Hochchule, Aachen.

[31] P. Morguet, and M. Lang, Spotting Dynamic Hand Gestures in Video Image Sequences Using Hidden Markov Models, *Proc.* In *IEEE Int'l Conf. on Image Processing*, Vol. 3 , (1998) pp. 193.

[32] Bender's official Website: http://bender.li2.uchile.cl/

[33] J. Ruiz-del-Solar, M. Correa, M. Mascaró, F. Bernuy, S. Cubillos, I. Parra, J. Lee-Ferng, R. Verschae, R. Riquelme, UChile HomeBreakers 2009 Team Description Paper, *RoboCup Symposium 2009*, June 29 – July 5, (2009) Graz, Austria (CD Proceedings).

[34] J. Ruiz-del-Solar, M. Mascaró, M. Correa, F. Bernuy, R. Riquelme, R. Verschae, Analyzing the Human-Robot Interaction Abilities of a General-Purpose Social Robot in Different Naturalistic Environments. *Lecture Notes in Computer Science* 5949 (*RoboCup Symposium 2009*), (2010) pp. 308–319.

[35] M. Arenas, J. Ruiz-del-Solar, S. Norambuena, S. Cubillos, A robot referee for robot soccer. *Lecture Notes in Computer Science* 5399 (*RoboCup Symposium 2008*), (2009) pp. 426-438.

[36] J. Ruiz-del-Solar, Personal Robots as Ubiquitous-Multimedial-Mobile Web Interfaces, *5th Latin American Web Congress LA-WEB 2007*, (2007) pp. 120 – 127.

Proceedings of the IEEE ICRA 2010
Workshop on Multimodal Human-Robot Interfaces
Anchorage, Alaska, May, 2010

Vision-based gesture recognition interface for a social robot

J.P. Bandera, A. Bandera, L. Molina-Tanco and J.A. Rodríguez

Abstract— **Social robots are designed to cooperate with people in their everyday activities. Thus, they should be able to adapt to uncontrolled environments, learn new tasks and become engaging companions for people to interact with. While other sensory inputs are also very important, in order to provide the social robot with the ability to interact with people using natural and intuitive channels, it may be interesting to consider the design and development of vision-based perceptual modules. In this sense, stereo vision systems appear as an useful option as they can provide 3D information, and can be mounted on the head of a social robot. On the other hand, stereo vision-based interfaces have to deal with limited resolution and frame rate, occlusions, noisy images and depth ambiguity. This paper describes the main aspects of a stereo vision-based gesture recognition and learning interface designed to be integrated in a social robot. This system captures, recognizes and learns upper-body social gestures at human interaction rates.**

I. INTRODUCTION

Robots have been massively used in industrial environments for the last fifty years. Industrial robots are designed to perform repetitive, predictable tasks that may be dangerous, disengaging or boring for human workers. These tasks are performed in controlled environments, where human presence is limited and controlled, if allowed [1]. These characteristics of both performed tasks and environmental conditions allow to provide industrial robots with a complete *a priori* knowledge database. Industrial robots are reprogrammed only if the task they are performing changes. On the other hand, these robots have to be aware only of a constrained set of environmental parameters that are directly related with programmed task. Thus, perceptual systems mounted on industrial robots are usually simple, practical and task-oriented.

While their usefulness is evident, industrial robots are strongly limited. More than thirty five years ago, a new generation of robots began to appear [2]. These robots would no more be considered useful tools. They are instead designed to cooperate with people in everyday activities. These robots should be able to work in uncontrolled environments and become engaging companions for people to interact with. It will not be possible to predict all possible situations they will have to face, thus these robots should also be provided with the capability to adapt to new situations, tasks and human companions. While motor capabilities were usually the main specification for industrial robot, for these new robots perception becomes a key element.

This work has been partially granted by the Spanish Junta de Andalucía, Projects P06-TIC-2123 and P07-TIC-03106, and by the Spanish MICINN and FEDER funds, Project TIN2008-06196.

The authors are with the Department of Electronic Technology, University of Málaga. Málaga, 29071, Spain jpbandera@uma.es

Robots that have to interact with people in everyday environments may benefit from sharing certain perceptual and motor human abilities. This idea influenced the use of the term *humanoid robot* to name these agents, and moved robotic researchers to address complex, engaging objectives such as bipedal walking, multi-fingered manipulators or stereo-based vision systems. Despite important contributions in these fields, the idea of a robot that resembles people in perceptual, motor and knowledge capabilities is still a long term objective. In fact, in the last decade the more generic term *social robot* has been introduced to define this new generation of robots. Social robots are agents designed to cooperate with people in everyday tasks. They may be humanoid or not, but in any case they have to be *social*. Following previous works [3][4], in this paper the definition given in [5] for social robot is adopted: *robots that work in social environments, and that are able to perceive, interact with and learn from other individuals, being these individuals people or other social agents.*

In order to become useful companions, social robots should use natural and intuitive interaction and perception channels. Speech recognition [4] and tactile sensors [6] are important features for a social robot. But vision represents the sensory input that usually provides more information to people. Face expression, hand movements and social gestures are key elements in social interactions [4][5]. Some interaction processes (e.g. those involving crowded or noisy environments) may even rely only on vision to achieve communication. It is desirable for a social robot, then, to include a vision-based interface. When dealing with the design of such a system two important questions have to be answered: (i) how to capture visual data from the environment; and (ii) how to process these data in order to provide *on-line* response to the human user.

While other options may be possible, stereo-vision systems are the most common solution to capture visual data [7]. These systems can be easily mounted on the head of a robot, and they provide 3D information. Besides, they are more similar to human eyes than other solutions, and thus they may find less difficulties in adapting to everyday environments, that are designed to be perceived by people. Stereo vision systems also present drawbacks that have to be considered: they have limited resolution, field of view and frame rate, they are very sensitive to occlusions, and they have to deal with noisy images and depth ambiguity [7][8].

Capturing images is just the first step in the perceptual process. It is necessary to extract only relevant information from the huge amount of visual input data if *on-line* response is required. Biological entities filter perceived information

by attention [9]. Social robots focus also attention only in certain relevant features of the environment. Thus, Breazeal [4] considers controlled scenarios in which only certain defined objects are detected and tracked. Hecht et al. [8] label different body parts and track them using particle filters. While their particular implementation may vary, attentional mechanisms are present in all perceptual components proposed for social robots [10][11][12][13].

According to the given definition, social robots are not only aware of their surroundings, but they are also able to learn from, recognize and communicate with other individuals. Robots that could learn from its observations and experiences, and from human teachers, would be able to adapt to new situations and perform new tasks, or improve already known ones. While other strategies are possible, *robot learning by imitation* (RLbI) represents a powerful, natural and intuitive mechanism to teach social robots new tasks [10]. In RLbI scenarios, a person can teach a robot by simply demonstrating the task that the robot has to perform. There are many issues that have to be addressed regarding RLbI. It is desirable to avoid invasiveness and controlled environments, the robot should count with return channels that can provide *on-line* feedback for the human teacher, it may be required to research methods for sharing attention [4], etc. One of the main of these issues is the translation from human to robot activities. This problem is more important as the differences from human to robot bodies grow [5]. Despite all these issues, the important advantages of RLbI systems over other learning methods [10] have moved many researchers in the last decade to address the objective of providing social robots with RLbI architectures [10][11][4][12][13]. These architectures will allow social robots to perceive, recognize, learn and imitate behaviours exhibited by human companions.

In this paper, a new RLbI architecture is proposed that provides a social robot with the ability to learn and imitate upper-body social gestures. This architecture, that is the main topic of the first author's Thesis [5], uses an interface based on a pair of stereo cameras, and a model-based perception component to capture human movements from input image data. Perceived human motion is segmented into discrete gestures and represented using features, that are subsequently employed to recognize and learn gestures. Finally, imitation is achieved by using a translation module that combines different strategies. The rest of the paper is organized as follows: Section II describes the proposed RLbI system. Section III details the different components that conforms the architecture. Section IV presents the results of the experiments performed to test the system, while section V discusses several key topics related with the system. Finally, section VI concludes the paper and defines future research lines.

II. SYSTEM OVERVIEW

There have been many proposals of RLbI architectures in the last decade, due to the increasing interest in autonomous agents, humanoid robots and social robotics [10][11][4][12][13]. These architectures differ in their objectives, considered perceptual inputs, number of modules, levels of abstraction or structure. However, it has been possible to identify some key components that are common to these architectures. It is in the elements inside each of these components, and the relations that are established between these elements, where the differences between RLbI architectures lie. A brief description of these components, that are deeply explained in [14], is given below:

- **Input**. This component includes all the sensory inputs that are available for the architecture. While visual input is necessary to achieve imitation in RLbI scenarios, some authors propose the use of additional perceptual channels, such as auditive or proprioceptive -own state- perception.
- **Perception**. The perception component contains all modules that are used to extract useful information from available perceptual channels.
- **Knowledge**. The knowledge component represents the memory of the social robot. This component contains all elements that are used to store information units, both learnt or preprogrammed. It also includes elements used to process these data.
- **Learning**. Social robots work in everyday environments where it is not possible to predict all possible situations they may face. Thus, social robots are provided with some learning mechanisms that allow them to adapt and learn from these new situations. The learning component is a key component of a RLbI architecture. It mainly affects the knowledge component, adding new items to the knowledge database, but also modifying already stored items or deleting old ones.
- **Motion generation**. RLbI requires the social robot to be able to imitate behaviours. Imitation involves translating the perceived or learnt motion to the robot, and generating a sequence of motion commands. The motion generation component contains all elements that are responsible of generating a motion output in the robot.
- **Output**. Motion commands are received by this component of the RLbI architecture, that uses the abilities of the social robot to execute them.

In this paper a novel architecture based on these components is proposed. This architecture captures, recognizes and learns upper-body human gestures, and it is depicted in Fig. 1. It is influenced by previous approaches, but it also incorporates new elements. Thus, as many previous RLbI architectures [10][11][12], it is divided into two main parts, one regarding perception, and the other action. It also includes a filtering process in the perceptual component. Finally, it relies on a database of known gestures to conform the core of the knowledge component [4][10][11], as Fig. 1 depicts.

As commented above, the proposed architecture also presents important differences respect to most previous proposals. Thus, many of these proposals appeal for a unified

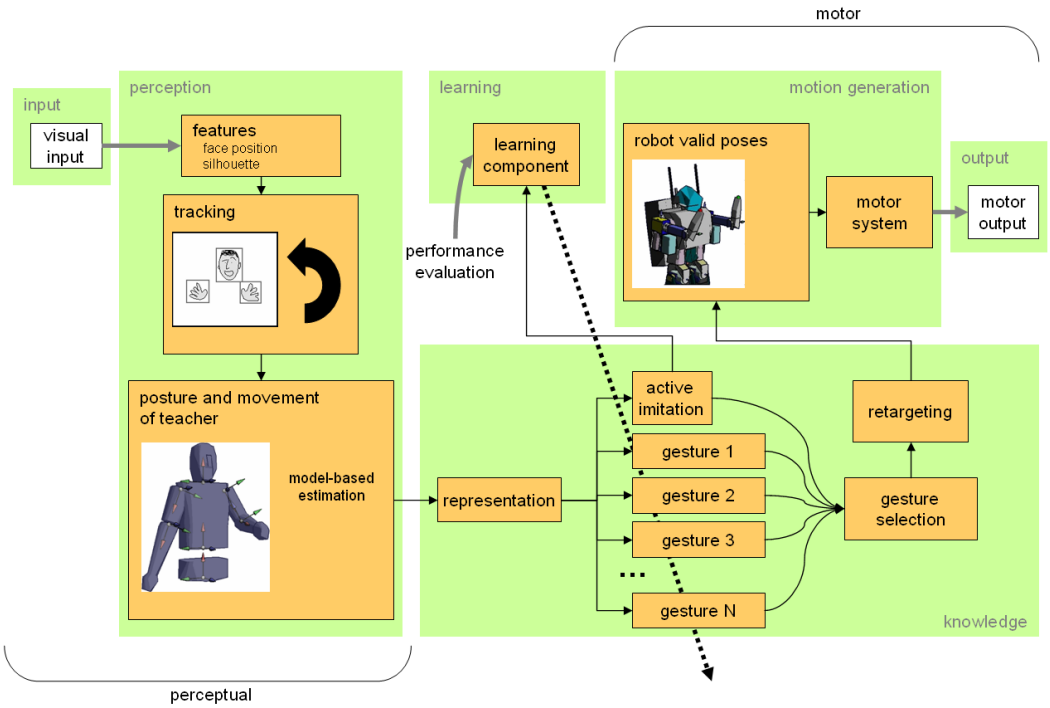


Fig. 1. Proposed RLbI architecture.

representation of perception and action in the *robot* motion space [10][11], following the idea that such a representation is present in biological entities [15]. But biological observers perform imitation and social learning from demonstrators of the same species [9], or who are perceived as belonging to the same species. Small children may find difficulties in learning from imitation when the demonstrator is a machine [15]. It may be reasonable to suppose that the opposite situation will find similar difficulties. Using the robot motion space to represent perceived human motion constraints the perceptual capabilities of the robot due to its motor limitations, as the robot will not recognize gestures it can not imitate. In this paper the idea of 'using a human model to perceive human movements' is proposed as an alternative to previous approaches. It has the additional advantage that translation -or *retargeting*- modules are executed only if imitation is required.

The explicit presence of a retargeting module is another contribution of the proposed architecture. It is considered here that human and robot may be very different, and thus translating motion from human to robot may become a complex issue, that requires a careful design. Finally, it is important to emphasize that the proposed RLbI architecture has been designed not as a theoretical framework, but as a system to be mounted in a real social robot. Thus, experiments have been performed in real RLbI scenarios, in which limited perception, uncontrolled environments and untrained users are present.

III. COMPONENTS

The different components of the proposed RLbI architecture have been implemented as detailed below.

A. Input and Perception

The input component for the proposed architecture is composed by a pair of stereo cameras. These cameras take colour images and disparity maps from the environment. They will be typically mounted on the head of a social robot, and in any case their baseline is set close to average human eye-to-eye distance. This configuration allows the stereo cameras to capture upper-body human motion at standard social interaction distances, that usually vary from 1.5 to 2 meters [5].

Captured images are processed in the perception component (Fig. 2), that extracts human pose from them. It can be seen in Fig. 2 that the first step of this process is to locate a human face that is close enough to the cameras. The 'feature detection' module performs this operation as detailed in [5]. Once the face is detected, the motion of the person begins to be captured. Face 3D position is employed to extract human silhouettes from disparity maps [17], while hands are detected as skin color regions in certain parts of the silhouette. As depicted in Fig. 2, once the 3D positions of the face and hands have been detected, these body parts are tracked *on-line* by the tracking element depicted in Fig. 1, that also implements the 'inhibition of return' mechanism [16].

Disparity silhouettes and tracked 3D positions of head and hands are the features used to estimate upper-body pose. A

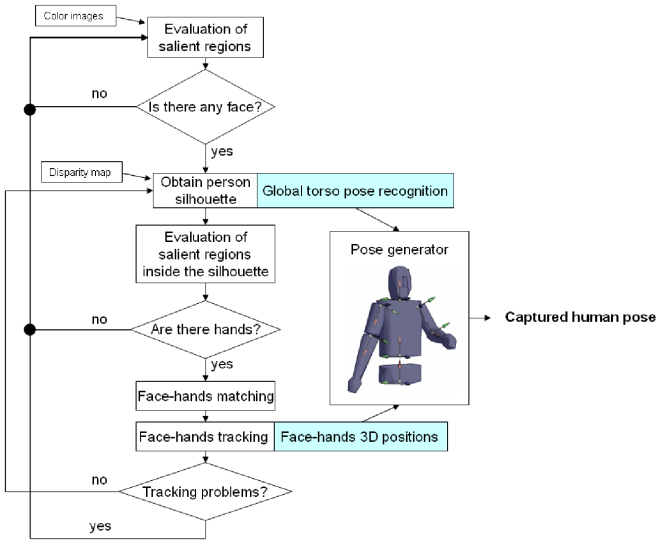


Fig. 2. Flow diagram of the perception component.

model-based human motion capture (HMC) algorithm is employed that firstly estimates torso flexion and rotation angles from disparity silhouettes, using anthropometric relations [17]. Once torso has been posed, the proposed algorithm uses an analytic method based on inverse kinematics to obtain valid arm poses from the 3D positions of the hands [18]. See [5] for a complete description of the whole HMC system.

B. Knowledge

The knowledge component of the proposed RLbI architecture contains the gestures the social robot has already learnt. But before perceived motion can be compared against stored gestures, it has to be segmented into discrete gestures. In this paper this segmentation is based on dynamic time thresholds [5]. Perceived discrete gestures are composed by sets of 3D trajectories followed by different body parts. In order to achieve *on-line* recognition, these trajectories are translated to a more efficient and compact representation. In this paper a novel gesture representation is proposed that considers two different types of features to characterize each gesture: global features and local features. Global features are defined against an external reference and thus they are more robust against noise and outliers. The global features used in the proposed system are simple absolute and relative motion amplitudes. Local features, on the other hand, are based on differential measures and are superior in discriminating fine details of the trajectory [19]. The proposed architecture uses sequences of dominant points as local features for each trajectory. These dominant points are extracted from the adaptive curvature functions associated to the perceived trajectories, as detailed in [20].

Once gestures have been represented as sets of local and global features, they are compared against the gestures stored in the knowledge database of the social robot. This process is deeply explained in [20], and can be summarized as follows: Local features are compared using the Dynamic Time Warping (DTW) algorithm. Then, the resulting local distances are reinforced by a global similarity value, obtained by applying analytic relations to the global features of compared gestures. Thus, both local and global features are considered to obtain the final distances, expressed as confidence values [5]. These values indicate the degree of similarity between each gesture in the knowledge database and the perceived gesture. They will be further used in the learning component to decide whether the gesture is recognized or not.

As Fig. 1 depicts, the whole representation and recognition processes are performed in the human motion space. Thus, even social robots which body is very different to the human body will be able to correctly perceive, understand and learn human gestures when using the proposed architecture.

The last element of this component is the retargeting module. This module translates the resulting motion to the robot motion space if imitation is required. The translation process considers both end-effector positions and joint angle values in a combined strategy, that tends to preserve the former for location movements, and the latter for configured movements [21][5]. As Fig. 1 depicts, the retargeting process does not need to be executed if the robot is not going to imitate the perceived motion.

C. Learning

The learning component of the proposed RLbI architecture uses the confidence values, obtained in the recognition stage, to add new gestures to the repertoire of the robot. The dataflow of this component is depicted in Fig. 3. It can be seen that learning is based on a double threshold. The first threshold Ω allows to directly recognize gestures that are very similar to a stored one (i.e. the biggest obtained confidence value C_{i1} is over Ω). Recognized gestures do not modify the knowledge database. On the other hand, as Fig. 3 shows, the second threshold ω is relative. Gestures that do not satisfy this second threshold are candidates to be included in the repertoire as new gestures. Human supervision is required when adding new gestures to the database. Besides, the first experiments involving this algorithm showed that the first steps in the learning process were critical, thus the amount of human supervision was incremented in this stage of the process (Fig. 3).

D. Motion generation and Output

The retargeted motion is not directly sent to the motors of the robot. A virtual model of the robot is used before to check that the resulting poses are valid. The robot model adopts desired poses using the same algorithms employed in the perception component to pose the human model. Once valid poses have been obtained for the robot, they are sent to its motor system, thus it is able to physically imitate perceived or recognized gestures.

IV. EXPERIMENTAL RESULTS

The different components of the proposed RLbI architecture were individually tested before integrating them in the complete system. The setup and results of these prior experiments are briefly commented below:

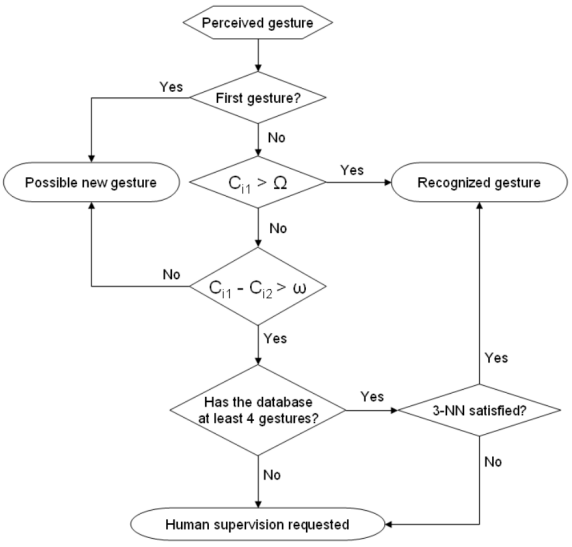


Fig. 3. Flow diagram of the learning component.

- The HMC system conformed by the input and perception components was quantitatively tested in a set of experiments, in which a certain human motion was perceived using both the proposed vision-based HMC system and a Codamotion CX1 HMC system from Charnwood Dynamics Ltd., based on active markers. The latter was used to obtain a reliable ground-truth [5]. Stereo images for the proposed system were captured using a STH-DCSG-VAR-C stereo pair provided by Videre Design. Extensive tests were conducted, in which the positions of the real markers used by the Codamotion CX1 system were compared against the positions of virtual markers, located on the virtual human model used by the proposed vision-based HMC system. Table I depicts obtained errors [5].

TABLE I
TRACKING ERRORS AVERAGED OVER 5300 FRAMES.

Marker	Left Shoulder	Left Elbow	Left Hand
Mean Error (cm)	5.74	12.53	11.51
Standard Deviation (cm)	3.13	6.06	6.55
Marker	Right Shoulder	Right Elbow	Right Hand
Mean Error (cm)	6.72	12.41	11.47
Standard Deviation (cm)	5.01	6.94	7.63
Marker	Left Head	Abdomen	Right Head
Mean Error (cm)	7.03	7.76	6.51
Standard Deviation (cm)	5.41	1.18	5.13

- Perceived motion is segmented into discrete gestures in the knowledge component. Then, this component firstly represents gestures in an efficient way and then uses these representations to match perceived and stored gestures. In order to independently test the representation and recognition modules, it was decided to use reliable motion data provided by the Codamotion CX1 system to fed the knowledge component. Extraction of features as dominant points of the adaptive curvature functions was compared against other representation methods, such as PCA, CSS or extraction of dominant points from fixed curvatures. The proposed method outperformed these approaches, as detailed in [20]. As commented above, distances between local features have been computed using DTW. This algorithm was compared against different dynamic programming techniques [20]. The results of these comparisons showed that DTW offered better recognition rates, and it was also more robust against outliers and noise.
  The combined retargeting strategy was also tested for both location and configured movements [21]. The results of these tests showed that it was able to adequately adapt to each particular situation. They were also useful to detect some undesired limitations in the robot arm motion, that will be corrected in further implementations [21].
- Finally, the learning algorithm based on double threshold was firstly tested over gestures captured using the Codamotion CX1 system. Obtained results were adequate, and emphasize the importance of human supervision in the first steps of the learning process.

After these experiments, the complete RLbI architecture was integrated and tested in real human-robot interaction scenarios. These scenarios involved dynamic, uncontrolled environments, untrained users, limited perception and *on-line* response. Unfortunately NOMADA, the social robot that is being developed in our research group [5], is not yet finished. However, prior versions of its perceptual system and one of its arms are available. A complete virtual model of this robot is also available. Thus, in the experiments of the complete system, stereo images are captured using the same cameras detailed before. The available arm of the robot imitates the movements of the human right arm. The retargeting module and the motion generation component, on the other hand, use the virtual model of the complete robot to check validity of imitated upper-body pose considering real joint limits and collisions.

Experiments use a dataset consisting of 53 upper-body gestures performed by six different people. For each of these gestures, the motion of the left and right hands is recorded at an average sampling rate of 15 Hz. The average amount of samples per gesture is 103.5. The gestures in the dataset are different executions of 8 upper-body gestures, that are commonly found in social interaction scenarios. The people who perform the previously detailed gestures stand in front of the vision-based system at a distance from 1.30 to 1.80 meters. No specific clothes are used to perform the experiments. Tests are performed in real indoor environments, that change dynamically. Thus, lighting changes, people walking around during experiments or environment variations (i.e. chairs or objects moved from one place to another) occur during the execution of the gestures. Two sets of experiments are conducted. In the latter the robot has no prior knowledge about performed gestures. The results of these experiments show that the robot is correctly able to recognize, imitate

and learn upper-body human gestures in these scenarios [5].

V. DISCUSSION

After experiments conducted in real RLbI scenarios, it is clear that the main limitation of the proposed system lies in its perceptual capabilities. *On-line* response imposes the use of a limited resolution. Thus, it is currently not possible to capture face expressions or detailed finger movements. But even although higher resolutions may be possible, limited field of view, noisy images and, specially, disparity errors drastically affect the quality of perceived motion [8]. Average errors listed in Table I are in the range of the pixel errors associated to the stereo cameras. Significantly, these errors are higher as the tracked item approach image borders, due to lens distortion, the pin-hole model used for the cameras and the perspective effects. As discussed in [5], it may be very difficult to improve HMC results by modifying the perceptual component, that behaves correctly if perceived data are accurate. It is in the input component where modifications may be more useful. Thus, the use of better cameras or lenses, but also the inclusion of additional sensory inputs, different from vision (e.g. speech, laser range finders, etc.), should be considered in order to improve the perceptual capabilities of the social robot.

Definitions for *social robot* [3][4][5] include agents that may be very different from people, thus it is important to consider RLbI architectures in which these differences are explicitly considered. As detailed above, one of the main contributions of the proposed system is the execution of the gesture representation, recognition and learning processes in the human motion space, while the retargeting module only translates motion to the robot body if imitation is required. Experimental results show that this architecture, that makes perception independent from the particular motor abilities of the robot, is able to efficiently produce adequate results [5]. While it may be argued that RLbI is simply not possible in agents that are not able to imitate perceived motion to a certain degree, in the proposed architecture a more practical approach is followed, that provides a social robot with the ability to capture human motion as accurately as possible, regardless its physical body.

VI. CONCLUSIONS AND FUTURE WORKS

The main contribution of this paper is a vision-base gesture recognition interface that can be integrated in a social robot. This interface works *on-line*, it is non-invasive and can be used in uncontrolled environments, and by untrained users. Conducted tests show that upper-body gestures can be efficiently perceived, recognized and learnt using the proposed architecture. Limitations detected in the stereo vision system suggest that future work should mainly focus on increasing the perceptual capabilities of the social robot, most probably using multimodal interfaces. More precisely, speech recognition should be incorporated to the robot as it is a key element in most social interactions. Other sensory inputs such as laser range finders or infrared sensors should also be considered. Finally, the complete RLbI architecture will be integrated in a more complex system, that will include higher level decision layers [5].

REFERENCES

[1] J. Craig, *Introduction to robotics: Mechanics and control*, Addison-Wesley, Boston, MA, USA; 1986.

[2] H. Inoue, S. Tachi, Y. Nakamura, K. Hirai, N. Ohyu, S. Hirai, K. Tanie, K. Yokoi and H. Hirukawa,"Overview of humanoid robotics project of meti", *in Proceedings of the 32nd international symposium on robotics*, Seoul, Korea, 2001, pp 1478-1482.

[3] K. Dautenhahn and A. Billard,"Bringing up robots or -the psychology of socially intelligent robots: From theory to implementation", *in Proceedings of the third annual conf. on autonomous agents*, Seattle, Washington, USA, 1999, pp 366-367.

[4] C. Breazeal, Toward sociable robots, *Robotics and Autonomous Systems*, vol. 42 (3-4), 2003, pp 167-175.

[5] J.P. Bandera, Vision-based gesture recognition system in a robot learning by imitation framework, Ph.d. dissertation, Department of Electronic Technology, University of Málaga, Spain; 2009.

[6] T. Asfour, K. Regenstein, P. Azad, J. Schröde and R. Dillmann, "Armar III: A humanoid platform for perception-action integration", *in 2nd international workshop on human-centered robotic systems (hcrs'06)*, 2006.

[7] T. Moeslund, A. Hilton and V. Krüger, A survery of advances in vision-based human motion capture and analysis, *Computer Vision and Image Understanding*, vol. 104, 2006, pp 90-126.

[8] F. Hecht, P. Azad, and R. Dillmann, "Markerless human motion tracking with a flexible model and appearance learning", *in Proceedings of the 2009 IEEE international conference on robotics and automation (ICRA 2009)*, Kōbe, Japan, 2009, pp 3173-3179.

[9] A. Bandura, *Handbook of socialization theory and research*, in D. A. Goslin (Ed.), Rand-McNally, Chicago, IL, USA; 1969, pp 213-262.

[10] S. Schaal, Is imitation learning the route to humanoid robots?, *Trends in Cognitive Sciences*, vol. 3 (6), 1999, pp 233-242.

[11] J. Demiris and G. Hayes, "Imitation as a dual-route process featuring predictive and learning components: A biologically plausible computational model", in K. Dautenhahn and C. Nehaniv (Eds.), Imitation in animals and artifacts, MIT Press, Cambridge; 2002.

[12] M. Mülihg, M. Gienger, S. Hellbach, J.J. Steil and C. Goerick, "Task-level imitation learning using variance-based movement optimization", *in Proceedings of the 2009 IEEE international conference on robotics and automation (ICRA 2009)*, Kōbe, Japan, 2009, pp 1177-1184.

[13] Y. Mohammad and T. Nishida, Interactive perception for amplification of intended behavior in complex noisy environments, *AI & Society*, vol. 23 (2), 2009, pp 167-186.

[14] J.P. Bandera, L. Molina-Tanco, J.A. Rodríguez and A. Bandera, "Architecture for a robot learning by imitation system", accepted in *Proceedings of the 15th IEEE mediterranean electrotechnical conference (Melecon 2010)*, Valletta, Malta, 2010.

[15] A. Meltzoff and M. Moore, Imitation in newborn infants: Exploring the range of gestures imitated and the underlying mechanisms, *Developmental Psychology*, vol. 25, 1989, pp 954-962.

[16] J.P. Bandera, R. Marfil, L. Molina-Tanco, A. Bandera, J.A. Rodríguez and F. Sandoval, Visual tracking of human activity for a social robot working on real indoor scenarios, *International Journal of Factory Automation, Robotics and Soft Computing*, vol. 3, 2008, pp 120-128.

[17] A. Cruz, J.P. Bandera and F. Sandoval, "Torso pose estimator for a robot imitation framework", *in Proceedings of the 12th international conference on climbing and walking robots and the support technologies for mobile machines*, Istanbul, Turkey, 2009, pp 901-908.

[18] J.P. Bandera, R. Marfíl, L. Molina-Tanco, J.A. Rodríguez, A. Bandera and F. Sandoval, *Robot Learning by Active Imitation*, I-Tech Education and Publishing, Vienna, Austria; 2007, pp 519544.

[19] N. Alajlan, I.E. Rube, M. Kamel,M. and G. Freeman, Shape retrieval using triangle-area representation and dynamic space warping, *Pattern Recognition*, vol. 40, 2007, pp 19111920.

[20] J.P. Bandera, R. Marfil, A. Bandera, J.A. Rodríguez, L. Molina-Tanco and F. Sandoval, Fast gesture recognition based on a two level representation, *Pattern Recognition Letters*, vol. 30 (13), 2009, pp 1181-1189.

[21] J.P. Bandera, R. Marfil, R. López, J.C. del Toro, A. Palomino and F. Sandoval, "Retargeting system for a social robot imitation interface", *in Proceedings of the 11th international conference on climbing and walking robots and the support technologies for mobile machines*, Coimbra, Portugal, 2008, pp 1233-1241.