Integration of Visuomotor Learning, Cognitive Grasping and Sensor-Based Physical Interaction in the UJI Humanoid Torso

A. P. del Pobil, A. J. Duran,
M. Antonelli, J. Felip, A. Morales
Robotic Intelligence Lab,
Universitat Jaume I, Castellón, Spain
{pobil,abosch,antonell,jfelip,morales}@uji.es

M. Prats
Willow Garage,
Menlo Park, California, USA
mprats@willowgarage.com

E. Chinellato
Imperial College London
South Kensington College, London, UK
e.chinellato@imperial.ac.uk

Abstract

We present a high-level overview of our research efforts to build an intelligent robot capable of addressing real-world problems. The UJI Humanoid Robot Torso integrates research accomplishments under the common framework of multimodal active perception and exploration for physical interaction and manipulation. Its main components are three subsystems for visuomotor learning, object grasping and sensor integration for physical interaction. We present the integrated architecture and a summary of employed techniques and results. Our contribution to the integrated design of an intelligent robot is in this combination of different sensing, planning and motor systems in a novel framework.

1 Introduction

Our contribution to the design of intelligent robots is in a high-level overview of the integration of different cognitive abilities in the UJI Humanoid Torso (Fig.1), resulting from an extended research program. This system integrates research accomplishments of three distinct projects over five years, which individually, by themselves, also comprise additional lower-level subsystems.

The first project is EYESHOTS (EYESHOTS 2008-2011) that started from the idea of investigating the cognitive value of eye movements when an agent is engaged in active exploration of its peripersonal space. In particular, we argued that, to interact effectively with the environment, the agent needs to use complex motion strategies at ocular level and also extended to other body parts, such as head and arms, using multimodal feedback to extract information useful to build representations of the 3D space, which are coherent and stable with respect to time. The second one was GRASP (GRASP 2009-2012), whose aim was the design of a cognitive system capable of performing grasping tasks in open-ended environments, by dealing with novelty, uncertainty and unforeseen situations. Our third challenge was robot manipulation beyond grasping to attain versatility (adaptation to different situations), autonomy (independent robot operation), and dependability (for success under modeling or sensing errors) (Mario Prats 2013). In our research we developed a unified framework for physical interaction (FPI) by introducing task-related aspects into the knowledge-based grasp concept, leading to task-oriented grasps; and similarly, grasp-related issues were also considered during the sensor-based execution of a task, leading to grasp-oriented tasks. This results in the versatile specification of physical interaction tasks, as well as the autonomous planning of these tasks, and the sensor-based dependable execution combining three different types of sensors: force, vision and tactile.

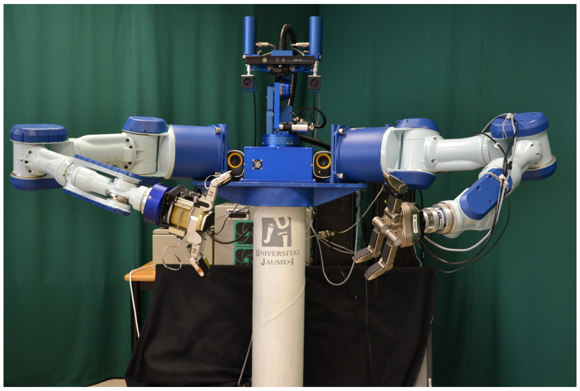


Figure 1: The UJI Humanoid Torso *Tombatossals*.

1.1 Visuomotor Learning

The goal of the EYESHOTS project was to investigate the interplay existing between vision and motion control, and to study how to exploit this interaction to achieve knowledge of the surrounding environment that allows a robot to act properly. Our research relied upon the assumption that a complete and operative cognition of visual space can be achieved only through active exploration, and that the natural effectors of this cognition are the eyes and the arms. The integration in the UJI Torso encompasses state-of-the-art capabilities such as object recognition, dynamic shifts of attention, 3D space perception, and action selection in unstructured environments, including eye and arm movements.

In addition to a high standard in engineering solutions, the development and integration of novel learning rules enables the system to acquire the necessary information directly from the environment. All the integrated processing modules are built on distributed representations in which sensorial and motor aspects coexist explicitly or implicitly. The models resort to a hierarchy of learning stages at different levels of abstraction, ranging from the coordination of binocular eye movements (e.g., learning disparity-vergence servos), to the definition of contingent saliency maps (e.g., learning of object detection properties), up to the development of the sensorimotor representation for bidirectional eye-arm coordination. Through the distributed coding, indeed, it is possible to avoid a sequentialization of sensorial and motor processes, i.e., a hard-coded sequence of discrete events which is certainly desirable for the development of cognitive abilities at a pre-interpretative (i.e., sub- symbolic) level, e.g., when a system must learn binocular eye coordination, handling the inaccuracies of the motor system, and actively measure the space around it.

1.2 Prediction in Cognitive Grasping

To meet the aim of the GRASP project, we studied the problem of object grasping and devised a theoretical and measurable basis for system design that is valid in both human and artificial systems. This artificial cognitive system is deployed in real environments and interacts with humans and other agents. It needs the ability to exploit the innate knowledge and self-understanding to gradually develop cognitive capabilities. To demonstrate the feasibility of our approach, we instantiated, implemented and evaluated our theories and hypotheses on the UJI Humanoid Torso. GRASP goes beyond the classical perceive-act or act-perceive approach and implements a predict-act-perceive paradigm that originates from findings of human brain research and results of mental training in humans where the self-knowledge is retrieved through different emulation principles. The knowledge of grasping in humans is used to provide the initial model of the grasping process that then is grounded through introspection to the specific embodiment. To achieve open-ended cognitive behavior, we use surprise to steer the generation of grasping knowledge and modeling.

1.3 Integrating Vision, Force and Tactile Sensing

The concept of physical interaction has been around since the first works in Robotics and Artificial Intelligence (Del Pobil, Cervera, and Chinellato 2004)

We claim that a unified treatment of grasp and task- related aspects would imply very important advances in intelligent robot manipulation, and advocate a new view of the concept of physical interaction that suppresses the classical boundaries between the grasp and the task. This new view has its foundations in the classical task frame formalism and the concept of grasp preshaping. We proposed several contributions concerning the application of the FPI concept. First, the FPI framework supports a great variety of actions, not only involving direct hand-object manipulation, but also the use of tools or bimanual manipulation. Next, subsystems for autonomous planning of physical interaction tasks are in place. From a high-level task description, the planner selects an appropriate task- oriented hand posture and builds the specification of the interaction task by using the FPI framework. Last, for the dependable execution of these tasks we adopt a sensor-based approach composed of a grasp and task controller running simultaneously, and taking into consideration three different types of sensor feedback which provide rich information during manipulation with robot hands: force, vision and tactile feedback.

2 Integrated system

The UJI Humanoid Torso is the result of the integration of several independent robotic systems that are controlled by a layered architecture. The system was designed in the course of the above projects which shared the goal of integrating the perception of the environment (visual, tactile, etc) with the planning and execution of motor movements (eyes, arms and hands). Also, our group was in charge of the integration of several modules developed by other partners contributing to the projects. Given that the projects focused on different topics, with different people involved and different timing we developed several architectures to integrate the system, each one with a different level of abstraction. In this paper, we describe the unified architecture that we have come up with to merge together all these systems. The reminder of this section describes the UJI Humanoid Torso as well as its software architecture.

2.1 System setup

Tombatossals (Catalan for *mountain-crasher*) is a humanoid torso composed by a pan-tilt stereo head (*Robosoft To40 head*) and two multi-joint arms (*Mitsubishi PA10 Arm*).

The head is endowed with two cameras *Imaging Source DFK 31AF03-Z2* (resolution: 1024×768, frame rate: 30 fps) mounted at a baseline of ≈ 270 mm. This geometrical configuration allows for an independent control of gaze direction and vergence angle in cameras (4 DOF). Moreover, the head mounts a Kinect™ sensor (*Microsoft Corp.*) on the forehead that allows to obtain a three-dimensional reconstruction of the scene. The arms, Mitsubishi PA-10 7C, have seven degrees of freedom each. Both the head and the arms are equipped with encoders that allow gaining access to the motor positions with high precision.The right arm has a 4 DOF Barrett Hand and the left arm has a 7 DOF Schunk SDH2 Hand. Both hands are endowed with tactile sensors (*Weiss Robotics*) on the fingertips. Each arm has a JR3 Force-Torque sensor attached on the wrist between the arm and the hand.

The control system of the robot is implemented on two computers. These are connected by a cross ethernet cable. Each one is devoted to cope with different tasks. The vision computer process the visual pipeline from the system of the cameras and Kinect™sensor. The user interface is running in this computer too. The technical features of this computer are: Intel® Core™i5 CPU 650 @ 3.2 GHz, 8 Gb DDR3 DIMM 1333 MHz, NVidia™580GTX 1Gb. The remaining parts of the system hardware are connected to the control computer. This allows the management and communication with all devices that are part of the robot. The features of this computer are: Intel® Core™2 Quad CPU Q9550 @ 2.83 GHz, 8 Gb DDR2 DIMM 800 MHz, NVidia™9800GT 512 Mb.

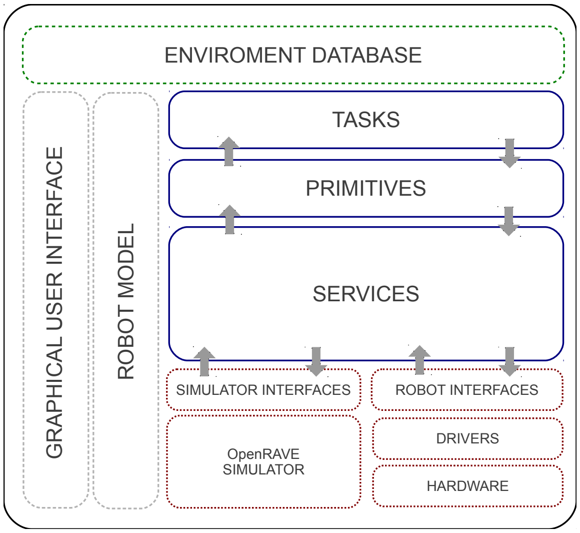


Figure 2: Integration software diagram.

2.2 Software architecture

To control all the components as an integrated platform, we have implemented a layered system that allows us to interact with the robot at different levels of abstraction (see Fig. 2).

Each layer is composed by modules that run in parallel and communicate with each other using three main types of messages:

- Data: input/output of the modules. It contains any type of information, raw or processed. For example, joint status, camera image, object positions, etc.
- Control: changes the parameters (threshold, loop rate, . . .) and the status (run, stop, . . .) of the modules.
- Event: contains information about a detected situation (an object is localized or grasped successfully, . . .).

Each module is wrapped into a node of the robotic operative system (ROS) (Quigley M., et al. 2009) which is in charge of managing the communication and provides other useful tools for development.

Interfaces. As detailed above, our robot is an ensemble of many different hardware components, each one providing different drivers and programming interfaces. The robot interface layer monitors and controls the state of the devices through hardware drivers. Then converts it into ROS messages. In this way, we obtain an abstraction from the hardware of the robot, because the other modules of the system need to know just the type of the data and not how to access it. Table 1 shows the ROS messages used for each device. Simulation interfaces do the same to connect OpenRAVE simulation to the system.

Table 1: ROS messages associated to the robot device.

Device	ROS message
Force	WrenchStamped
Velocity	TwistStamped
6D Pose	PoseStamped
Images	Image
Joint data (position, velocity and torque)	JointState
Point clouds	PointCloud2

Services. They are is a continuous non blocking loop that never stops by itself. Each loop generates at least one output and requires one or more inputs. Services neither generate events nor control messages. Modules in the service layer accept commands such as run, stop, reset or remap. The role of a service is to be a basic process that receives an input and provides an output. In the system, services are mostly used to provide high level information from raw sensor input. This layer provides the building blocks with basic operations that are in general useful for higher layers.

In this layer, inputs and outputs are not platform dependent and the robot model is available to the other layers that configure the services on the basis of the robot embodiment. This layer is not aware of the robot hardware below it, thus using the simulator or the real robot does not affect the modules in this or the upper layers.

An example of service is a blob detector, that receives an image as input, processes it and outputs the position of the detected blob, another example is the inverse kinematics velocity solver that receives a Cartesian velocity and converts it to joint velocity, this module uses the available robot model to do the calculations.

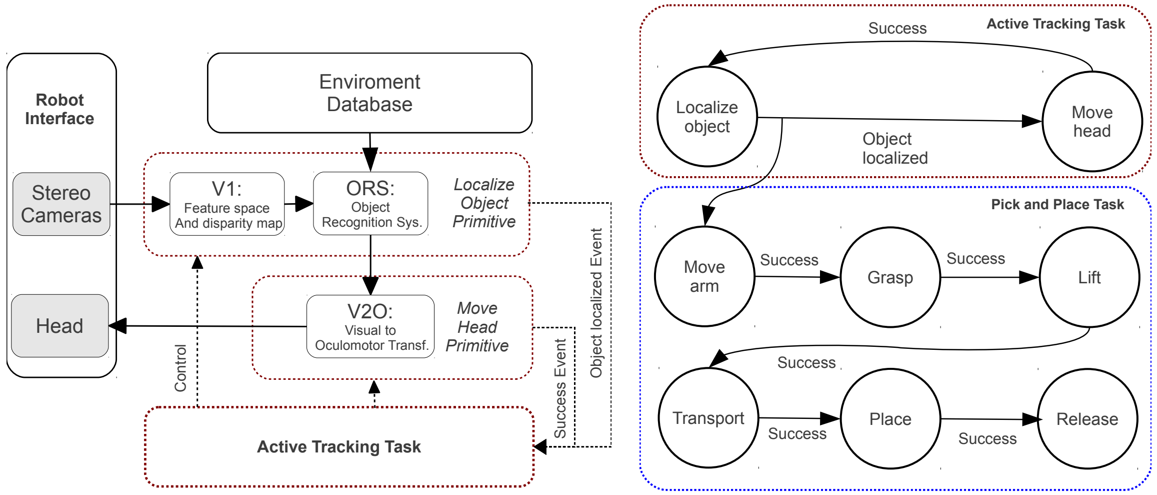
Primitives. They define higher level actions or processes, that may need motion (grasp, move, transport or look at) or may not (detect motion, recognize or localize objects).

As services, primitives are continuous, never stop by themselves and always generate at least one output. A primitive may not have inputs and generates events that can be caught by the task layer. The role of the primitives is to use services to receive data and to send control actions to the robot. Primitives can detect different situations by looking at the data and generate events accordingly. A primitive is a control loop that gets processed data from services, generates events and sends control actions to the service layer.

The primitive layer is more platform independent than the service layer, thus most primitives are platform independent and do not need knowledge about the platform to work.

Tasks. They represent the higher level processes that can be described with this system. Tasks use primitives as building blocks to generate the defined behaviors. In fact a task can be represented as a cyclic, directed, connected and labeled multigraph, where the nodes are primitives and the arcs are events that need to be active. An example of a task that grabs an object while looking at it is depicted in Fig. 3.

Tasks do not need to be continuous and can end. There is no need for a task to generate any output but it can generate events. The role of a task is to coordinate the execution of primitives in the system, tasks generate control messages and change the data flow among primitives and services.



(a) Active tracking task. The task is composed of two primitives, *Localize Object* and *Move Head*. The former is composed of two services that create a distributed representation of the image and localize the object of interest. The latter is composed of a service that converts the retinotopic location of the target into an eye position. Both primitives launch an event when their state changes.

(b) Cooperation among tasks. The robot executes a task which consists of grasping and moving the target object (*pick and place*), and that requires seven primitives. In the meanwhile, the robot actively tracks the moved object.

Figure 3: System description at different levels of abstraction.

Robot model. It is available to all the layers and provides information about the embodiment that is being controlled.

GUI. It is connected to all layers and monitors the whole system. The user is able to interact (send control messages and events) with the system through this interface.

2.3 Tools and documentation

During the development of this integration software, we have prepared several tools to help the programmers and to make uniform the coding style. We consider that the style and documentation of the developed modules is a key point for the integration. Inspired by the ROS tool *ros-create-pkg* [1] we have developed our own scripts to create services and primitives. These scripts allow us to define the inputs, outputs and parameters of the module, then create all the file structure, includes, callbacks and documentation entries that must be filled in. This tools make the module coding more uniform and point out which are the key parts that need to be documented.

3 Achieved results

During the projects carried out by our group, a number of experiments were performed in our robotic platform, in fact, Tombatossals was the only demonstrator for the EYESHOTS project, an one of the demonstrators of the GRASP project. Using our system we have performed experiments focused on different topics such as visual awareness, sensorimotor learning, grasping, physical interaction and simulation.

3.1 Visual awareness and sensorimotor learning

The main goal of the EYESHOTS project was to achieve spatial awareness of the surrounding space by exploiting the interplay that exists between vision and motion control.

Our effort in the project was to develop a model that simulates the neurons of the brain's area V6A involved in the execution of reaching and gazing actions. The main result is a sensorimotor transformation framework that allows the robot to create an implicit representation of the space. This representation is based on the integration of visual and proprioceptive cues by means of radial basis function networks (Chinellato et al. 2011).

Experiments on the real robot shown that this representation allows the robot to perform correct gazing and reaching movements toward the target object (Antonelli, Chinellato, and del Pobil 2011). Moreover, this representation is not hard-coded but it is updated on-line, while the robot interacts with the environment (Antonelli, Chinellato, and Pobil 2013). The adaptive capability of the system and its design that simulates a population of neurons of the primates' brain made possible to employ the robot in a cognitive science experiment, such as saccadic adaptation (Chinellato, Antonelli, and del Pobil 2012).

Another important result of the EYESHOTS project, was the integration on *Tombatossals* of a number of models developed by the other research groups involved in the project. The result of the integration process made available on our robotic system a set of behaviors, such as recognizing, gazing and reaching target objects, that can work separately or cooperate for more structured and effective behaviors.

The system is composed by a hierarchy of modules that begins with a common visual front-end module that models

[1] http://www.ros.org

the primary visual cortex (Sabatini, Gastaldi, and F Solari et al. 2010). On the one hand, the output of this module is used by the model of high level visual areas (V2, V4, IT, FEF) to compute a saliency map and to recognize and localize the target object (Beuth, Wiltschut, and Hamker 2010). On the other hand, the same output is used by a controller that changes the vergence of the eyes to reduce the global disparity of the observed scene (Gibaldi et al. 2010). Finally, our sensorimotor framework is used to gaze to the target or to reach it (Antonelli, Chinellato, and Pobil 2013).

The modules implemented during the project, provided the main building blocks (*services*) to create *primitives* and execute *tasks*. The simplicity by which it is possible to create new behaviors allowed us to employ the robot in a human-robot interaction experiment (Stenzel et al. 2012).

3.2 Grasping and manipulation

Early experiments on sensor based controllers were performed to adapt the robot behavior to the real, uncertain and changing environment (Felip and Morales 2009). In this work we demonstrated, using a simple approach, that using sensors to adapt the robot actions increases the performance and robustness.

Our platform was also used for perception for manipulation experiments, Bogh et.al. (Bohg J. et al. 2011) presented a system that reconstructed the stereo visual input to fill the occluded part of the objects. With the reconstructed objects, the simulator was used to plan feasible grasps and to be executed on the real robot.

The integration of controllers for different platforms was also taken into account and presented in (Felip et al. 2012) where two different robots were performing the same task using abstract definitions. Such implementation of tasks uses the same concepts for high level task definition that were presented in previous section.

A test case of the full manipulation pipeline (i.e. perception-planning-action) is the experiment carried out by Felip et.al. (Felip, Bernabe, and Morales 2012), that achieved the task of emptying a box full of unknown objects in any position, see Fig. 4. Another example of the performance of the full manipulation pipeline was presented by (Bohg J.,et al. 2012) where the robot planned different grasps on household objects depending on the task to be executed.

Using the described system we also performed dual arm coordination and manipulation experiments. Fig. 5 shows the UJI Humanoid torso performing dual arm manipulation of a box.

3.3 Simulation

One of the outcomes of the GRASP project was the implementation of OpenGRASP, a set of plugins for OpenRAVE, that enabled tactile sensing in simulation. Moreover, we have accurately compared to which extent the simulator can be used as a surrogate of the real environment in a work that included a full dynamic simulation for all the robot sensors and actuators (Leon, Felip, and Morales 2012).

The simulator has proven to be a useful tool. Using it as an early test bench has saved a lot of debugging time to the research team. Moreover its tight integration in the system allows us to use the same controllers both for the real and simulated robot (Fig. 6).

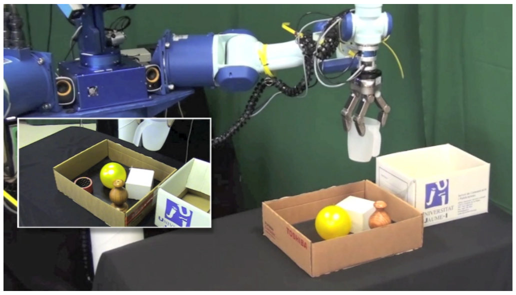


Figure 4: Tombatossals performing the empty-the-box experiment.

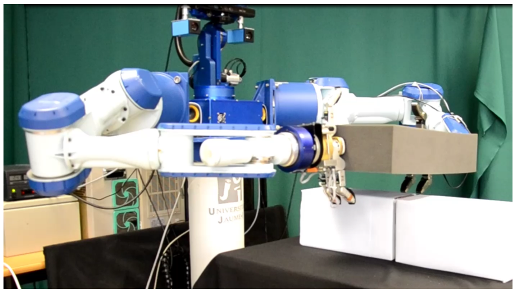


Figure 5: Tombatossals performing a dual arm manipulation experiment.

3.4 Sensor-Based Physical Interaction

We introduced a number of new methods and concepts, such as ideal task-oriented hand preshapes or hand adaptors, as part of our unified FPI approach to manipulation (Mario Prats 2013). The FPI approach provides important advances with respect to the versatility, autonomy and dependability of state-of-the-art robotic manipulation. For instance, the consideration of task-related aspects into the grasp selection allows to address a wide range of tasks far

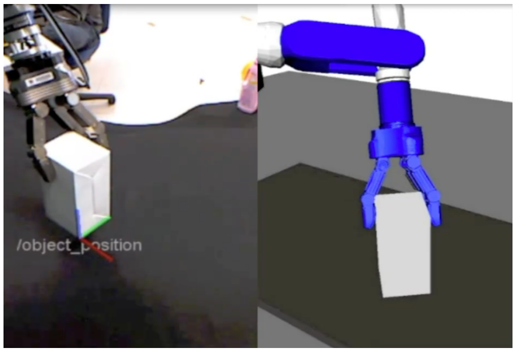


Figure 6: Real and simulated Tombatossals performing a grasping task using the same sensor-based controllers.

beyond those of pick and place that can be autonomously planned by the physical interaction planner, instead of adopting preprogrammed ad-hoc solutions. Most importantly, advances in dependability are provided by novel grasp-task sensor-based control methods using vision, tactile and force feedback. The results of our integrated approach show that the multimodal controller outperforms the bimodal or single-sensor approaches (Mario Prats 2013). All these contributions were validated in the real world with several experiments on household environments. The robot is capable of performing tasks such as door opening, drawer opening, or grasping a book from a full shelf. As just one example of this validation, the robot can successfully operate unmodeled mechanisms with widely varying structure in a general way with natural motions (Mario Prats 2013).

4 Conclusions

We have presented a summary of our research efforts to build an intelligent robot capable of addressing real-world problems with the common framework of multimodal active perception and exploration for physical interaction and manipulation. This system integrates research accomplishments of three distinct projects over five years. We have briefly presented the goals of the projects, the integrated architecture as implemented on Tombatossals, the UJI Robot Torso, and a summary of employed techniques and results, with references to previously published material for further details. We believe this combination of different sensing, planning and motor systems in a novel framework is a state-of-the-art contribution to the integrated design of an intelligent robot.

Acknowledgments

This research was supported by the European Commission (EYESHOTS ICT-217077 and GRASP ICT-215821) , by Ministerio de Ciencia (BES-2009-027151 and DPI2011-27846), by Generalitat Valenciana (PROMETEO/2009/052), by Fundació -Bancaixa (P1-1B2011-54) and Universitat Jaume I FPI program (PREDOC/2010/23).

References

Antonelli, M.; Chinellato, E.; and del Pobil, A. 2011. Implicit mapping of the peripersonal space of a humanoid robot. In *Computational Intelligence, Cognitive Algorithms, Mind, and Brain (CCMB), 2011 IEEE Symposium on*, 1–8.

Antonelli, M.; Chinellato, E.; and Pobil, A. 2013. Online learning of the visuomotor transformations on a humanoid robot. In Lee, and Sukhan et al., eds., *Intelligent Autonomous Systems 12*, volume 193 of *Advances in Intelligent Systems and Computing*. Springer Berlin. 853–861.

Beuth, F.; Wiltschut, J.; and Hamker, F. 2010. *Attentive Stereoscopic Object Recognition*. 41.

Bohg J. et al. 2011. Mind the gap - robotic grasping under incomplete observation. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, 686 –693.

Bohg J.,et al. 2012. Task-based grasp adaptation on a humanoid robot. In *10th Int. IFAC Symposium on Robot Control*.

Chinellato, E.; Antontelli, M.; and del Pobil, A. 2012. A pilot study on saccadic adaptation experiments with robots. *Biomimetic and Biohybrid Systems* 83–94.

Chinellato, E.; Antonelli, M.; Grzyb, B.; and del Pobil, A. 2011. Implicit sensorimotor mapping of the peripersonal space by gazing and reaching. *Autonomous Mental Development, IEEE Transactions on* 3(1):43–53.

Del Pobil, A.; Cervera, E.; and Chinellato, E. 2004. Objects, actions and physical interactions. Anchoring Symbols to Sensor Data, AAAI Press, Menlo Park, California.

Felip, J., and Morales, A. 2009. Robust sensor-based grasp primitive for a three-finger robot hand. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, 1811 –1816.

Felip, J.; Bernabe, J.; and Morales. 2012. Contact-based blind grasping of unknown objects. In *Humanoid Robots, 2012 IEEE-RAS International Conference on*.

Felip, J.; Laaksonen, J.; Morales, A.; and Kyrki, V. 2012. Manipulation primitives: A paradigm for abstraction and execution of grasping and manipulation tasks. *Robotics and Autonomous Systems* (0).

Gibaldi, A.; Chessa, M.; Canessa, A.; Sabatini, S.; and Solari, F. 2010. A cortical model for binocular vergence control without explicit calculation of disparity. *Neurocomp.* 73:1065–1073.

Leon, B.; Felip, J.; and Morales, A. 2012. Embodiment independent manipulation through action abstraction. In *Humanoid Robots, 2012 IEEE-RAS Int. Conference on*.

Mario Prats, Angel P. del Pobil, P. J. S. 2013. *Robot Physical Interaction through the combination of Vision, Tactile and Force Feedback*, volume 84 of *Springer Tracts in Advanced Robotics*.

Quigley M., et al. 2009. Ros: an open-source robot operating system. In *ICRA Workshop on Open Source Software*.

Sabatini, S.; Gastaldi, G.; and F Solari et al. 2010. A compact harmonic code for early vision based on anisotropic frequency channels. *Computer Vision and Image Understanding* 114(6):681–699.

Stenzel, A.; Chinellato, E.; Bou, M. A. T.; del Pobil, A. P.; Lappe, M.; and Liepelt, R. 2012. When humanoid robots become human-like interaction partners: Corepresentation of robotic actions. *Journal of Experimental Psychology: Human Perception and Performance* 38(5):1073–1077.