Reactive Reaching and Grasping on a Humanoid
Towards Closing the Action-Perception Loop on the iCub

Jürgen Leitner, Mikhail Frank, Alexander Förster and Jürgen Schmidhuber
Dalle Molle Institute for Artificial Intelligence (IDSIA) and Scuola universitaria professionale della Svizzera italiana (SUPSI) and Università della Svizzera italiana (USI), Lugano, Switzerland

Keywords: Humanoid Robot, Robot Control, Object Manipulation, Reactive Control, Collision Avoidance, Robot perception, Eye-hand Coordination, Computer Vision.

Abstract: We propose a system incorporating a tight integration between computer vision and robot control modules on a complex, high-DOF humanoid robot. Its functionality is showcased by having our *iCub* humanoid robot pick-up objects from a table in front of it. An important feature is that the system can avoid obstacles – other objects detected in the visual stream – while reaching for the intended target object. Our integration also allows for non-static environments, i.e. the reaching is adapted on-the-fly from the visual feedback received, e.g. when an obstacle is moved into the trajectory. Furthermore we show that this system can be used both in autonomous and tele-operation scenarios.

1 INTRODUCTION

After centuries of imagining automated machines that would help humans, the last few decades brought an ever growing number of these programmable appliances into existence. While there are nowadays thousands of robotic arms fulfilling automation tasks in factories, robots are not yet in wide-spread use outside industrial applications. More and more focus is put on enabling other ares of utilisation, e.g. the use of robots in household or healthcare settings. For robotic systems to perform tasks in such environments, which are not specifically designed for robots, a better understanding – i.e. perception – of the situation and a closer coordination between action and perception – i.e. adapting to changes occurring in the scene – are required (Ambrose et al., 2012; Kragic and Vincze, 2009; Kemp et al., 2007).

The goal of this work is to extend the capabilities of our robot to allow for more autonomous and more adaptive – some would say, more 'intelligent' – behaviours. Our platform is a state-of-the-art, high degree-of-freedom (DOF) humanoid (see Figure 1) and we focus on object manipulation in non-static environments. To facilitate this a close integration of computer vision and control modules was developed and implemented. Drawing inspiration from a wide variety of techniques, our system incorporates, e.g., machine learning approaches to implement vision.

The paper revisits some of the previous approaches in the fields of robot vision and control, as well as previous work on building integrated robotic systems using perception and action subsystems in Section 2. We then describe our approach to building a tightly coupled framework for visually-guided reaching and grasping in detail (Section 3), before we present our high DOF, state-of-the-art robot platform (Section 4). Section 5 explains the experiments and our findings, while conclusions (Section 6) and plans for future research are found at the end.

2 BACKGROUND

Although there exists a rich body of literature in computer vision, path planning, and feedback control, wherein many critical subproblems are addressed

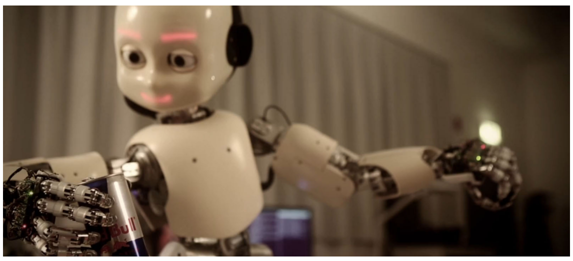

Figure 1: Our research platform, the *iCub* humanoid robot.

Leitner J., Frank M., Förster A. and Schmidhuber J..
Reactive Reaching and Grasping on a Humanoid - Towards Closing the Action-Perception Loop on the iCub.
DOI: 10.5220/0005113401020109
In *Proceedings of the 11th International Conference on Informatics in Control, Automation and Robotics* (ICINCO-2014), pages 102-109
ISBN: 978-989-758-039-0

individually, most demonstrable behaviours for humanoid robots do not effectively integrate elements from all three disciplines. Consequently, tasks that seem trivial to us humans, such as picking up a specific object in a cluttered environment, remain beyond the state-of-the-art in experimental robotics.

Sensory feedback is of critical importance to decide and act in unstructured environments. Robots require the ability to perceive their surroundings, as it is generally not feasible or possible to provide all the information needed a priori (the most striking example being robotic space exploration, where only little about the environment the robot is sent to is known before each mission).

2.1 Perception and Robot Vision

Creating a functional perception system is a hard, but important, problem. It is a prerequisite for robots to act in a purposeful, 'intelligent' way.

A wide range of sensors have been used to build models of the environment. Visual feedback tends to be the most interesting and active research area, as cameras tend to be cheap and in widespread use nowadays. It has extensively been used in mobile robot applications, for obstacle avoidance, mapping and localisation (Davison and Murray, 2002; Karlsson et al., 2005). Furthermore there exists a natural tendency to be inspired by human perception, in which vision plays a primary role, and behaviour, especially when working with humanoid robots. The field of Computer Vision spawned a multitude of research sub-domains, such as, scene reconstruction, event detection, object recognition, motion estimation, etc. An important problem in robot vision is that of determining whether the image data contains some specific object, feature, or activity. The interested reader is referred to a nice survey of the various approaches (Cipolla et al., 2010). This has been researched for quite some time but the task seems harder than expected. No solution for the general case, i.e. detecting arbitrary objects in arbitrary situations, exists (Kemp et al., 2007). Problems arise because of real-life environmental impacts, such as, changing light conditions, incomplete or inconsistent data acquisition (Kragic and Vincze, 2009) (see also Figure 2). Most object detection applications are using SIFT (Lowe, 1999) or SURF (Bay et al., 2006) with extensions for higher robustness (Stückler et al., 2013). For real time object tracking in videos frames contour trackers are often used (Panin et al., 2006). Experimental robotics though still relies heavily on artificial landmarks and fiducial markers to simplify (and speed-up) the detection problem.

Once an object is successfully detected in the camera images, it is important to localise it, so it can be placed into a model of the environment. 'Spatial Perception', as this is known, develops in humans over time from observation and interaction with the world. Research in brain- and neuro-science show clear trends on *what* changes during this development, but *how* these changes happen is not clear (Plumert and Spencer, 2007).

To obtain a distance measure stereo vision systems use two images taken from different angles (Hartley and Zisserman, 2000). In humanoid robot scenarios a method for localisation is required, which is able to cope with motion in the robot's head, gaze and upper body. While projective geometry approaches work well under carefully controlled experimental circumstances, they are not easily transferred to robotics applications though (as seen by the out-of-the-box localisation capabilities of the *iCub* (Pattacini, 2011)). It was previously shown that, even on complex humanoid robots, this stereo localisation can be learned (Leitner et al., 2012c).

2.2 Motion and Actions

Only after the robot knows *which* objects are in the environment and *where* they are located can it start to interact with them. The Path Planning Problem is an important problem in robot motion research. It denotes the issue to find motions that pursue a set goal, e.g. reaching, while deliberately avoiding obstacles and other constraints. Solving this problem is critical to deploying complex robots in unstructured environments. An overview of interesting approaches to motion planning can be found in the textbook "Planning Algorithms" (LaValle, 2006). To solve the path planning problem is generally expensive, therefore robots controlled this way are typically very deliberate and

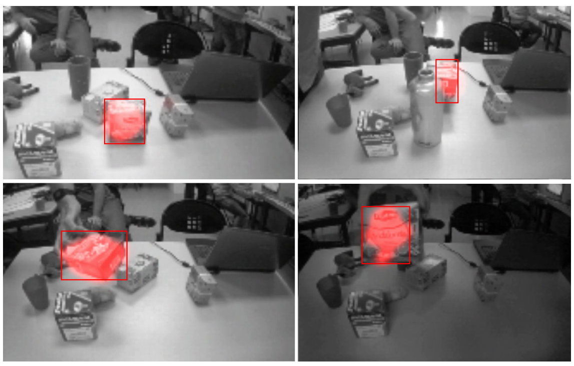

Figure 2: Detection of complex objects, e.g. a tea box, in changing poses, different light and when partially occluded is a hard problem in robot vision.

slow. This is often perceived during robotics demos (e.g. at recent DARPA Robotics Challenge Trials), where one can see them 'think' first, often for quite a bit, before a motion is executed.

A more reactive approach, instead of the 'think first, act later' paradigm, emerged in the 1980s. A variety of approaches have been applied to quickly generate control commands, without searching the robot's configuration space (Khatib, 1986; Brooks, 1991; Schoner and Dose, 1992). All these use local information from the workspace, and transform it into motor commands according to some heuristics. It is therefore not surprising that these approaches excel at fast, reactive obstacle avoidance while they have trouble with global planning tasks. Because of this such approaches have become popular in the context of safety and human-robot interaction (De Santis et al., 2007; Dietrich et al., 2011).

Once a motion is found to reach for a certain object the actual manipulation is not a trivial thing, even for humans. For example, research shows that even a prototypical precision grasp task is not matured until the age of 8-10 years (Forssberg et al., 1991)). Recent results though highlight the progress and advanced state of research in robot grasping (Carbone, 2013).

2.3 Integration

To allow for a variety of objects to be picked up from various positions the robot needs to see, act and react within an integrated control system. For example, methods enabling a 5 DOF robotic arm to pick up objects using a point-cloud generated model of the world and objects are available to calculate reach and grasp behaviours (Saxena et al., 2008). In 2010 a technique for robots to pick up non-rigid objects, such as, towels was presented (Maitin-Shepard et al., 2010). It allows to reliably and robustly pick up a towel from a table by going through a sequence of vision-based regrasps and manipulations-partially in the air, partially on the table.

Even when sufficient manipulation skills are available these need to be constantly adapted by an perception-action loop to yield desired results. A must read for roboticists is 'Robotics, Vision and Control' (Corke, 2011)'. It puts the close integration of these three components into the spotlight and describes common pitfalls and issues when trying to build such systems with high levels of sensorimotor integration. In the DARPA ARM project, which aims to create highly autonomous manipulators capable of serving multiple purposes across a wide variety of applications, the winning team showed an end-to-end system that allows the robot to grasp and pick-up diverse objects (e.g. a power drill, keys, screwdrivers, etc.) from a table by combining touch and LASER sensing (Hudson et al., 2012).

Visual Servoing (Chaumette and Hutchinson, 2006) is a commonly used approach to closed-loop vision based control of robotic systems – i.e. some level of hand-eye coordination. It has been shown to work as a functional strategy to control robots without any prior hand-eye calibration (Vahrenkamp et al., 2008).

3 CLOSING THE ACTION-PERCEPTION LOOP

Our aim is to generate a pick-and-place operation for the *iCub*. For this, functional motion and vision subsystems are integrated to create a closed action-perception loop. The vision-side detects and localises the object continuously, while the motor-side tries to reach for target objects avoiding obstacles at the same time. A grasping of the object is triggered when the hand is near the target.

3.1 Action Side: MoBeE

Modular Behavioral Environment (MoBeE) (Frank et al., 2012) is a software infrastructure to realise complex, autonomous, adaptive and foremost safe robot behaviours. It acts as an intermediary between three loosely coupled types of modules: the Sensor, the Agent and the Controller. These correspond to abstract solutions to problems in Computer Vision, Motion Planning, and Feedback Control, respectively. An overview of the system is depicted in Figure 3. The framework is robot independent, and can exploit any device that controlled via YARP (Metta et al., 2006). It also supports multiple interacting robots, and behavioural components are portable and reusable thanks to their weak coupling. MoBeE controls the robot constantly, according to the following second order dynamical system:

$$M\ddot{q}(t) + C\dot{q}(t) + K(q(t) - q^*) = \sum f_i(t) \qquad (1)$$

where $q(t) \in R^n$ is the vector function representing the robot's configuration, M, C, K are matrices containing mass, damping and spring constants respectively. q^* denotes an attractor (resting pose) in configuration space. Constraints on the system are implemented by forcing the system via $f_i(t)$, providing automatic avoidance of kinematic infeasibilites arising from joint limits, cable lengths, and collisions.

An agent can interact with MoBeE, instead of directly with the robot, by sending arbitrary high-level

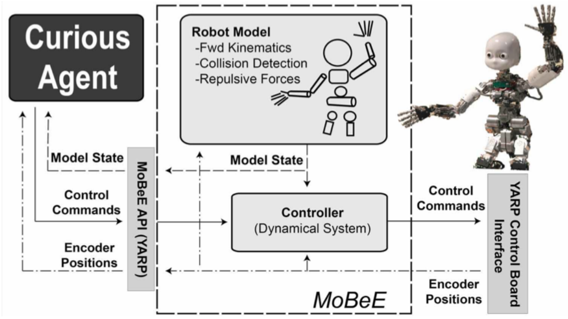

Figure 3: The Modular Behavioral Environment Architecture: MoBeE implements low-level control and enforces necessary constraints to keep the robot safe and operational in real-time. Agents (left) are able to send high-level commands, while a kinematic model (top) is driven by the stream of encoder positions (right). The model computes fictitious constraint forces, which repel the robot from collisions, joint limits, and other infeasibilities. These forces, $f_i(t)$, are passed to the controller (middle), which computes the attractor dynamics that governs the actual movement of the robot.

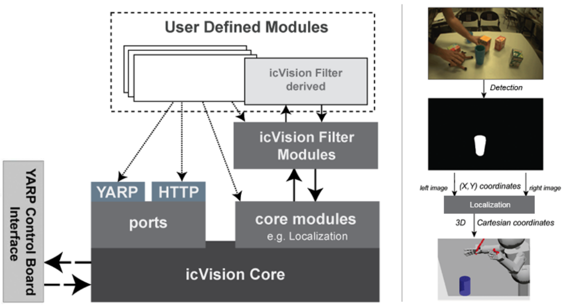

Figure 4: The icVision Architecture: The core module (bottom) is mainly for housekeeping and accessing & distributing the robot's camera images and motor positions. Object detection is performed in the *filter* modules (top), by segmenting the object of interest from the background. A typical work flow is shown (right): input images are retrieved from the cameras, the specific object is detected by a trained *Filter Module*, before the outputs (together with the robot's current pose) are used to estimate a 3D position using a localisation module. The object is then placed in the world model.

control commands. For example, when a new attractor q^* is set to a desired pose by an agent, e.g. by calculating the inverse kinematics of an operational space point, $q(t)$ begins to move toward q^*. The action then terminates either when the dynamical system settles or when a timeout occurs, depending on the constraint forces $f_i(t)$ encountered during the transient response.

3.2 Perception Side: icVision & CGP-IP

icVision (Leitner et al., 2012b) is an open-source[1] framework consisting of distributed YARP modules performing computer vision related tasks in support of cognitive robotics research (Figure 4). It includes the modules for the detection and identification of objects (in the camera images, referred to as *Filters*), as well as the localisation of the objects in the robot's operational space (3D Cartesian space). At the centre is the *icVision* core module, which handles the connection with the hardware and provides housekeeping functionality (e.g., extra information about the modules started and methods to stop them). Currently available modules include object detection, 3D localisation, gazing control (attention mechanism) and saliency maps. Standardised interfaces allow for easy swapping and reuse of modules.

The main part in object detection, the binary segmentation of the object from the background (see Figure 4 on the right), in the visual space, is performed in separate *icVision* filter modules. Each one is trained using our Cartesian Genetic Programming (CGP) implementation called CGP-IP, in combination with the OpenCV (Bradski, 2000) library, to detect one specific object in a variety of real-life situations (see Figure 2) – hence also identifying it. The framework can run multiple filter modules in parallel. A variety of filters have been learnt and most of them are able to perform the object detection in near real-time.

To interact with the objects the robot also needs to know where the object is located. Developing an approach to perform robust localisation to be deployed on a real humanoid robot is necessary to provide the necessary inputs for on-line motion planning, reaching, and object manipulation. *icVision* provides modules to estimate the 3D position based on the robot's pose and the location of object in the camera images.

3.3 Integration

The sensory and motor sides establish quite a few capabilities by themselves, yet to grasp objects successfully while avoiding obstacles they need to work closely together. The continuous tracking of obstacles and the target object is required to create a reactive reaching behaviour which adapts in real-time to the changes of the environment.

We created interfaces between *MoBeE* and *icVision* allowing for a continuous visual based localisation of the detected objects to be propagated into the world model. This basic eye-hand coordination allows for an adaptation while executing the reaching behaviour to changing circumstances, improving our robot's autonomy. Figure 5 gives an overview of

[1]Code available at: https://github.com/Juxi/icVision/

the research in our lab. Various modules (grey boxes) were developed and integrated into a functional system. The perception side consists mainly of object detection and localisation (top area, depicted in green). The bottom area (yellow) shows the action/motion side, while the blue area on the right represents the memory modules, including a world model. The various modules are:

- *Object Models, Detection and Identification:* as mentioned above, the detection and identification of objects is a hard problem. To perform these tasks CGP-IP (Cartesian Genetic Programming for Image Processing) (Harding et al., 2013) is used. It provides a machine learning approach to building visual object models, which can be converted into executable code, in both supervised and unsupervised fashion (Leitner et al., 2012a). The resulting program performs the segmentation of the camera images for a specific object.
- *Object Localisation:* by using the image coordinates of the detected object from the two cameras together with the current robot's pose, the position of the object can be estimated in Cartesian space wrt. the robot's reference frame. Instead of a calibration for each single camera, the stereo system and the kinematic chain, we incorporate a module that learns to predict these from a training set. Our system has been shown to estimate these positions with a technique based on genetic programming (Leitner et al., 2012d) and an artificial neural network estimators (Leitner et al., 2012c). After the object is detected in the camera images the location of an object is estimated and the world model is updated.
- *Action Repertoire:* herein we only use a lightweight, easy-to-use, one-shot grasping system (LEOGrasper[2]), which has been used extensively at IDSIA (Figure 6). It can be configured to perform a variety of grasps, all requiring to close the fingers in a coordinated fashion. The *iCub* incorporates touch sensors on the fingertips, due to the high noise, we use the error reported by the PID controllers of the finger motors to know when they are in contact with the object.
- *World Model, Collision Avoidance and Motion Generation:* the world model keeps track of the robot's pose in space and the objects it has visually detected. Figure 7 shows this model including the robot's pose the static table, and the two objects localised from vision. MoBeE is used to safeguard the robot from (self-)collisions. It furthermore allows to generate motion by forcing the hand in operational space.

[2]Source code available at: https://github.com/Juxi/iCub/

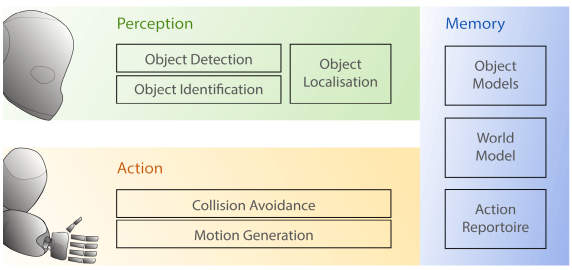

Figure 5: Overview of the modules and the on-going research towards a functional eye-hand coordination system on the *iCub*. The modules are shown by subsystem: perception (green), action (yellow) and memory (blue).

4 THE ROBOT

Our experimental platform is an open-hardware, open-software humanoid robotic system developed within various EU projects, called *iCub* (Tsagarakis et al., 2007) (shown in Figure 1). It consists of two arms and a head attached to a torso roughly the size of a human child. The head and arms follow an anthropomorphic design and provide a high DOF system that was designed to investigate human-like object manipulation. It provides also a tool to investigate human cognitive and sensorimotor development. To allow for safe and 'intelligent' behaviours the robot's movements need to be coordinated closely with feedback from visual, tactile, and acoustic[3] sensors. The *iCub* is an excellent experimental platform for cognitive, developmental robotics and embodied artificial intelligence (Metta et al., 2010). Due to its complexity it is also a useful platform to test the scalability and efficiency of various machine learning approaches when used together with real-life robotic systems.

[3]Acoustic feedback might be used when manipulation objects to know if a grasp was successful.

Figure 6: Grasping a variety of objects successfully, such as, tin cans, plastic cups and tea boxes. The module works for both the right and left hand.

5 EXPERIMENTS & RESULTS

The inverse kinematics problem, i.e. placing the hand at a given coordinate in operational space, can be performed with previously available software on the *iCub*, such as, the existing operational space controller (Pattacini, 2011) or a roadmap-based approach (Stollenga et al., 2013). These systems though require a very accurate calibration the mechanical system, however some mechanical non linearities still cannot be taken into account. In addition they are not yet coupled to visual localisation, meaning there is no easy way to adapt the reach during execution.

The first experiment shows that with the herein presented system we are able to reactively move the arm out of harms way when the environment changes. We then show how we can use this system to reactively reach and grasp these objects when we change their type from 'obstacle' to 'target', therefore changing the fictional forces calculated.

5.1 Avoiding a Moving Obstacle

Static objects in the environment can be added directly into *MoBeE*'s world model. Once, e.g. the table, is in the model, actions and behaviours are adapted due to computed constraint forces. These forces, $f_i(t)$ in (1), which repel the robot from collisions with the table, governs the actual movement of the robot. This way we are able to send arbitrary motions to our system, while ensuring the safety of our robot (this has recently been shown to provide a good reinforcement signal for learning robot reaching behaviours (Pathak et al., 2013; Frank et al., 2014)). The presented system has the same functionality also for arbitrary, non-static objects. After detection in both cameras the object's location is estimated (*icVision*) and propagated to *MoBeE*. The fictional forces

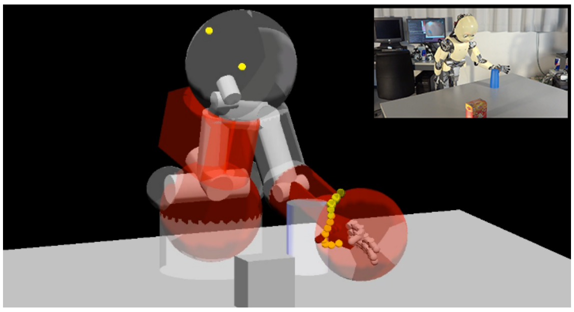

Figure 7: Showing the visual output of the *MoBeE* world model during one of our experiments. Parts in red indicate (an impeding) collision with the environment (or itself). The inset shows the actual scene. (See video: https://www.youtube.com/watch?v=w_qDH5tSe7g).

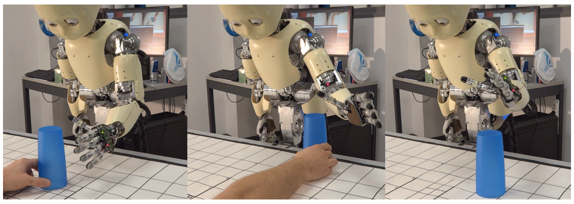

Figure 8: The reactive control of the left arm, permitting the iCub to stay clear of the non-static 'obstacle', as well as the table. (See video: https://www.youtube.com/watch?v=w_qDH5tSe7g).

are calculated to avoid impeding collisions. Figure 7 shows how the localised object is in the way of the arm and the hand. To ensure the safety of the rather fragile fingers, a sphere around the end-effector can be seen. It is red, indicating a possible collision, because the sphere intersects with the object. The same is valid for the lower arm. The forces, calculated at each body part using Jacobians, push the intersecting geometries away from each other, leading to a forcing of the hand (and arm) away from the obstacle. Figure 8 shows how the the robot's arm is avoiding a non-stationary obstacle. The arm is 'pushed' aside at the beginning, when the cup is moved close to the arm. It does so until the arm reaches its limit, then the forces cumulate and the end-effector is 'forced' upwards to continue avoiding the obstacle. Without an obstacle the arm starts to settle back into its resting pose q^*.

5.2 Reaching and Grasping Objects

This next experiment is on a simple reactive pick-and-place routine for the *iCub*. Similarly to the above experiment we are using *MoBeE* to adapt the reaching behaviour while the object is moved. To do this we change the type of the object within the world model from 'obstacle' into 'target'. Due to this change there is no repelling force calculated between the object and the robot parts. In fact we can now use the vector from the end-effector to the target object as a force that drives the hand towards a good grasping position.

MoBeE also allows to trigger certain responses when collisions occur. In the case, when we want the robot to pick-up the object, we can active a grasp subsystem whenever the hand is in the close vicinity of the object. We are using a prototypical power grasp style hand-closing action, which has been used successfully in various demos and videos.[4] Figure 6 shows the *iCub* successfully picking up (by adding an extra upwards force) various objects using our grasping subsystem, executing the same action.

[4] See videos at: http://robotics.idsia.ch/media/

Our robot frameworks are able to track multiple objects at the same time, which is also visible in Figure 7, where it tracks both the cup and the tea box. By simply changing the type of the object within *MoBeE* the robot reaches for a certain object while avoiding the other.

6 CONCLUSIONS

Herein we present our on-going research towards visual guided object manipulation with the *iCub*. A tightly integrated sensorimotor system, based on two frameworks developed over the past years, enables the robot to perform a simple pick-and-place task. The robot reaches to detected objects, placed at random positions on a table.

Our implementation enables the robot to adapt to changes in the environment. Through this it safeguards the *iCub* from unwanted interactions – i.e. collisions. We do this by integrating the visual system with the motor side by using an attractor dynamic based on the robot's pose and a model of the world. This way we achieve a level of eye-hand coordination not previously seen on the *iCub*.

In the future we would like to integrate some machine learning to further improve the object manipulation skills of our robotic system. Improving the predication and selection of actions will lead to a more adaptive, versatile robot. Furthermore it might be of interest to investigate an even tighter sensorimotor coupling, e.g. avoiding translation into operational space by working in vision/configuration space.

ACKNOWLEDGEMENTS

The authors would like to thank the European Commission for supporting this research under grant no. FP7-IST-IP-231722, 'Intrinsically Motivated Cumulative Learning Versatile Robots' (IM-CLeVeR).

REFERENCES

Ambrose, R., Wilcox, B., Reed, B., Matthies, L., Lavery, D., and Korsmeyer, D. (2012). Robotics, tele-robotics, and autonomous systems roadmap. Technical report, NASA.

Bay, H., Tuytelaars, T., and Van Gool, L. (2006). Surf: Speeded up robust features. *Computer Vision – ECCV 2006*, pages 404–417.

Bradski, G. (2000). The OpenCV Library. *Dr. Dobb's Journal of Software Tools*.

Brooks, R. (1991). Intelligence without representation. *Artificial intelligence*, 47(1):139–159.

Carbone, G. (2013). *Grasping in robotics*, volume 10 of *Mechanisms and Machine Science*. Springer.

Chaumette, F. and Hutchinson, S. (2006). Visual servo control, part i: Basic approaches. *IEEE Robotics & Automation Magazine*, 13(4):82–90.

Cipolla, R., Battiato, S., and Farinella, G. M. (2010). *Computer Vision: Detection, Recognition and Reconstruction*, volume 285. Springer.

Corke, P. (2011). *Robotics, Vision and Control*, volume 73 of *Springer Tracts in Advanced Robotics*. Springer.

Davison, A. J. and Murray, D. W. (2002). Simultaneous localization and map-building using active vision. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(7):865–880.

De Santis, A., Albu-Schaffer, A., Ott, C., Siciliano, B., and Hirzinger, G. (2007). The skeleton algorithm for self-collision avoidance of a humanoid manipulator. In *Proc. of the IEEE/ASME International Conferenced on Advanced Intelligent Mechatronics*.

Dietrich, A., Wimbock, T., Taubig, H., Albu-Schaffer, A., and Hirzinger, G. (2011). Extensions to reactive self-collision avoidance for torque and position controlled humanoids. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*.

Forssberg, H., Eliasson, A., Kinoshita, H., Johansson, R., and Westling, G. (1991). Development of human precision grip i: basic coordination of force. *Experimental Brain Research*, 85(2):451.

Frank, M., Leitner, J., Stollenga, M., Frster, A., and Schmidhuber, J. (2014). Curiosity driven reinforcement learning for motion planning on humanoids. *Frontiers in Neurorobotics*, 7(25).

Frank, M., Leitner, J., Stollenga, M., Harding, S., Förster, A., and Schmidhuber, J. (2012). The modular behavioral environment for humanoids and other robots (MoBeE). In *Proc. of the Int'l. Conf. on Informatics in Control, Automation & Robotics (ICINCO)*.

Harding, S., Leitner, J., and Schmidhuber, J. (2013). Cartesian genetic programming for image processing. In Riolo, R., Vladislavleva, E., Ritchie, M. D., and Moore, J. H., editors, *Genetic Programming Theory and Practice X*, Genetic and Evolutionary Computation, pages 31–44. Springer New York.

Hartley, R. and Zisserman, A. (2000). *Multiple view geometry in computer vision*. Cambridge Univ Press.

Hudson, N., Howard, T., Ma, J., Jain, A., Bajracharya, M., Myint, S., Kuo, C., Matthies, L., Backes, P., Hebert, P., Fuchs, T., and Burdick, J. (2012). End-to-end dexterous manipulation with deliberate interactive estimation. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*.

Karlsson, N., Di Bernardo, E., Ostrowski, J., Goncalves, L., Pirjanian, P., and Munich, M. (2005). The vSLAM Algorithm for Robust Localization and Mapping. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*.

Kemp, C., Edsinger, A., and Torres-Jara, E. (2007). Challenges for robot manipulation in human environments [grand challenges of robotics]. *IEEE Robotics & Automation Magazine*, 14(1).

Khatib, O. (1986). Real-time obstacle avoidance for manipulators and mobile robots. *The international journal of robotics research*, 5(1):90.

Kragic, D. and Vincze, M. (2009). Vision for robotics. *Foundations and Trends in Robotics*, 1(1):1–78.

LaValle, S. (2006). *Planning algorithms*. Cambridge Univ Pr.

Leitner, J., Chandrashekhariah, P., Harding, S., Frank, M., Spina, G., Forster, A., Triesch, J., and Schmidhuber, J. (2012a). Autonomous learning of robust visual object detection and identification on a humanoid. In *Proc. of the IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL)*.

Leitner, J., Harding, S., Frank, M., Förster, A., and Schmidhuber, J. (2012b). icVision: A Modular Vision System for Cognitive Robotics Research. In *Proc. of the International Conference on Cognitive Systems (CogSys)*.

Leitner, J., Harding, S., Frank, M., Förster, A., and Schmidhuber, J. (2012c). Learning spatial object localisation from vision on a humanoid robot. *International Journal of Advanced Robotic Systems*, 9.

Leitner, J., Harding, S., Frank, M., Förster, A., and Schmidhuber, J. (2012d). Transferring spatial perception between robots operating in a shared workspace. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

Lowe, D. (1999). Object Recognition from Local Scale-Invariant Features. In *Proc. of the International Conference on Computer Vision*.

Maitin-Shepard, J., Cusumano-Towner, M., Lei, J., and Abbeel, P. (2010). Cloth grasp point detection based on multiple-view geometric cues with application to robotic towel folding. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*.

Metta, G., Fitzpatrick, P., and Natale, L. (2006). YARP: Yet Another Robot Platform. *International Journal of Advanced Robotic Systems*, 3(1).

Metta, G., Natale, L., Nori, F., Sandini, G., Vernon, D., Fadiga, L., von Hofsten, C., Rosander, K., Lopes, M., Santos-Victor, J., Bernardino, A., and Montesano, L. (2010). The iCub humanoid robot: An open-systems platform for research in cognitive development. *Neural Networks*, 23(8-9):1125–1134.

Panin, G., Ladikos, A., and Knoll, A. (2006). An efficient and robust real-time contour tracking system. In *Proc. of the IEEE International Conference on Computer Vision Systems*.

Pathak, S., Pulina, L., Metta, G., and Tacchella, A. (2013). Ensuring safety of policies learned by reinforcement: Reaching objects in the presence of obstacles with the icub. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

Pattacini, U. (2011). *Modular Cartesian Controllers for Humanoid Robots: Design and Implementation on the iCub*. PhD thesis, Italian Institute of Technology, Genova.

Plumert, J. and Spencer, J. (2007). *The emerging spatial mind*. Oxford University Press, USA.

Saxena, A., Driemeyer, J., and Ng, A. (2008). Robotic grasping of novel objects using vision. *The International Journal of Robotics Research (IJRR)*, 27(2):157.

Schoner, G. and Dose, M. (1992). A dynamical systems approach to task-level system integration used to plan and control autonomous vehicle motion. *Robotics and Autonomous Systems*, 10(4):253–267.

Stollenga, M., Pape, L., Frank, M., Leitner, J., Fõrster, A., and Schmidhuber, J. (2013). Task-relevant roadmaps: A framework for humanoid motion planning. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

Stückler, J., Badami, I., Droeschel, D., Gräve, K., Holz, D., McElhone, M., Nieuwenhuisen, M., Schreiber, M., Schwarz, M., and Behnke, S. (2013). Nimbro@home: Winning team of the robocup@home competition 2012. In *Robot Soccer World Cup XVI*, pages 94–105. Springer.

Tsagarakis, N., Metta, G., Sandini, G., Vernon, D., Beira, R., Becchi, F., Righetti, L., Santos-Victor, J., Ijspeert, A., Carrozza, M., and Caldwell, D. (2007). iCub: the design and realization of an open humanoid platform for cognitive and neuroscience research. *Advanced Robotics*, 21:1151–1175.

Vahrenkamp, N., Wieland, S., Azad, P., Gonzalez, D., Asfour, T., and Dillmann, R. (2008). Visual servoing for humanoid grasping and manipulation tasks. In *Proc. of the IEEE International Conference on Humanoid Robots*, pages 406–412.