Pose Estimation for Grasping Preparation from Stereo Ellipses

Giovanni Saponaro[1], Alexandre Bernardino[2]

[1] *Dipartimento di Informatica e Sistemistica "Antonio Ruberti" Sapienza - Università di Roma, via Ariosto 25, 00185 Rome, Italy*

[2] *Institute for Systems and Robotics - Instituto Superior Técnico Torre Norte, Piso 7, Av. Rovisco Pais, 1049-001 Lisbon, Portugal*

giovanni.saponaro@gmail.com, alex@isr.ist.utl.pt

This paper describes an approach for real-time preparation of grasping tasks, based on the low-order moments of the target's shape on a stereo pair of images acquired by an active vision head. The objective is to estimate the 3D position and orientation of an object and of the robotic hand, by using computationally fast and independent software components. These measurements are then used for the two phases of a reaching task: (i) an initial phase whereby the robot positions its hand close to the target with an appropriate hand orientation, and (ii) a final phase where a precise hand-to-target positioning is performed using Position-Based Visual Servoing methods.

Keywords: Reaching, Grasping, 3D Pose Estimation, Stereo, Visual Servoing.

1. Introduction

Grasping and manipulation are among the most fundamental tasks to be considered in humanoid robotics. Like humans distinguish themselves from other animals by having highly skilled hands, humanoid robots must consider dexterous manipulation as a key component of practical applications such as service robotics or personal robot assistants.

The high dexterity present in human manipulation does not come for granted at birth, but it arises from a complex developmental process across many stages. Babies first try to reach for objects, with very low precision; then they start to adapt their hands to the shape of the objects, and only at several years of age they are able to master their skills. Together with the manipulation, perception develops in parallel in order to incrementally increase performance in detecting and measuring the important object features for grasping. Along time, interactions with objects of diverse shapes

are performed, applying many reaching and manipulation strategies. Eventually, salient effects are produced (e.g. the object moves, deforms, makes a sound when squeezed), perceived and associated to actions. An agent learns the object affordances,[10] i.e. the relationships between a certain manipulation action, the physical characteristics of the object and the observed effect. The way of reaching for an object evolves from a purely position-based mechanism to a complex behavior which depends on target size, shape, orientation, intended usage and desired effect.

Framed by the context of the EU project RobotCub,[9] this work aims at providing simple 3D object perception for enabling the development of manipulation skills in a humanoid robot. The objective of the RobotCub project is to build an open-source humanoid platform for original research on cognitive robotics, focusing especially on developmental aspects. Inspired by recent results in neurosciences and developmental psychology, one of the tenets of the RobotCub project is that manipulation plays a key role in the development of cognitive ability.

This work puts itself in an early stage of this developmental pathway and will address the problem of reaching for an object and preparing the grasping action according to the orientation of the objects to interact with. It is not intended to have a very precise measurement of object and hand postures, but merely the necessary quality to allow for successful interactions with the object. Precise manipulation will emerge from experience, by the optimization of action parameters as a function of the observed effects.[10] To have a simple enough model of object and hand shape, they are approximated as 3D ellipses. The only assumption is that objects have a sufficiently distinct color to facilitate segmentation from the background. Perception of object orientation in 3D is provided by the second-order moments of the segmented areas in the left and right images, acquired in the humanoid robot active vision head.

The paper will describe the humanoid robot setup, computer vision techniques, 3D orientation estimation, the strategy to prepare the reaching and grasping phases, and experimental results.

2. Robotics setup

The robotic platform of RobotCub, called the *iCub*, has the appearance of a three-year-old child, with an overall of 53 degrees of freedom (see Fig. 1). However, the iCub's arm-hand system is still under development and for this work the robot *Baltazar*[7] was used: it is a robotic torso built with the aim of understanding and performing human-like gestures, mainly for

biologically inspired research (see Fig. 1).

To reach for an object, two distinct phases are considered:[8] (i) an open-loop ballistic phase is used to bring the manipulator to the vicinity of the target, whenever the robot hand is not visible in the robot's cameras; (ii) a closed-loop visually controlled phase is used to make the final alignment to the grasping position. The open-loop phase (reaching preparation) requires the knowledge of the robot's inverse kinematics and a 3D reconstruction of the target's posture. The target position is acquired by the camera system, where the hand position is measured by the robot arm joint encoders. Because these positions are measured by different sensory systems, the open-loop phase is subject to mechanical calibration errors. The second phase, grasping preparation, operates when the robot hand is in the visible workspace. 3D position and orientation of target and hand are estimated in a form suitable for Position-Based Visual Servoing (PBVS).[4,6] The goal is to make the hand align its posture with respect to the object. Since both target and hand postures are estimated in the same reference frame, this methodology is not prone to mechanical calibration errors.

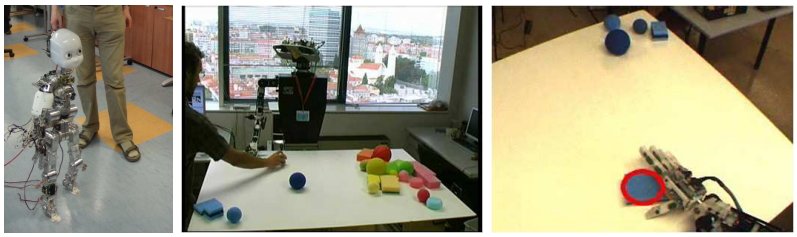

Fig. 1. Left: RobotCub humanoid platform iCub. Middle: humanoid robot Baltazar in its workspace. Right: view from one of Baltazar's eyes during a grasping task.

2.1. *Software architecture*

The software architecture used in this project is based on YARP[a], a cross-platform, open-source, multitasking library, specially developed for robotics. YARP facilitates the interaction with the devices of humanoid robot Baltazar, as well data exchange among the various software components (middleware). Other libraries used are OpenCV[b] for image pro-

[a]Yet Another Robot Platform: `http://eris.liralab.it/yarp`.
[b]Open Computer Vision Library: `www.intel.com/technology/computing/opencv`.

cessing, and GSL[c] for efficient matrix computation, especially in the 3D reconstruction part (see Sec. 3.2).

Particular care was put into designing the several components of the project as distributed. YARP takes care of inter-process communication (IPC), while the several concurrent instances of the CAMSHIFT tracker (left and right view of the target object, left and right view of the robot hand) can run on different machines or CPU cores: as modern processors sprout an increasing number of cores, the code can thus take advantage of the extra power available and improve real-time performance.

3. Visual processing

Using computer vision to control the grasping task is natural, since it allows to recognize and to locate objects (see Ref. 5 and Ref. 6). In particular, stereo vision can help robots reconstruct the 3D scene and perform visual servoing. In this work, the CAMSHIFT tracking algorithm[2,3] was used extensively. A brief outline of it is given in the next section.

3.1. *CAMSHIFT algorithm*

Originally designed for the field of perceptual user interfaces and face tracking,[3] CAMSHIFT is a method based on color histograms and MeanShift,[1] which in turn is a robust, non-parametric and iterative technique that finds the mode of a probability distribution, in a manner that is well suited for real-time processing of a live sequence of images.

A sketch of the algorithm logic and a sample execution are presented in Fig. 2. For this project, a modified version of the CAMSHIFT implementation publicly available in OpenCV was used. The inputs are the current original image obtained from the camera and its color histogram in the HSV (hue, saturation, value) space. The output of each iteration of CAMSHIFT is a "back projected" image, produced by the original image by using the histogram as a lookup table. When it converges, a CAMSHIFT tracker returns not only the position, but also the size and 2D orientation of the best-fit ellipse to the segmented target points. Then, the boundary points in the ellipse along its major axis are computed.

Consider a *target object* placed in front of the robot; tracking is accomplished by running two CAMSHIFT processes. Let points $\{p_1, p_2\}_l^{target}$ and

[c]GNU Scientific Library: `http://www.gnu.org/software/gsl/`.

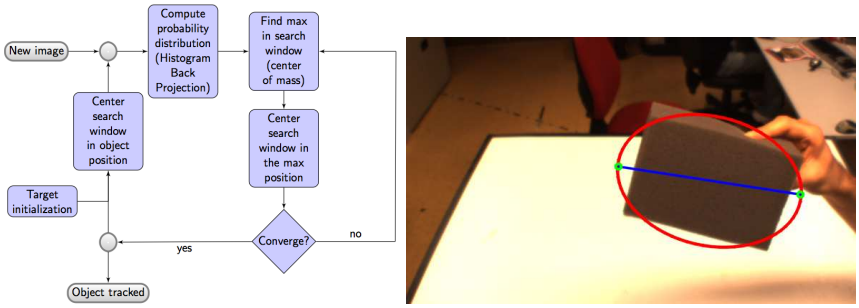

Fig. 2. Left: Flux diagram of the CAMSHIFT object tracking algorithm. Right: CAMSHIFT tracking of an object. The approximating best-fit ellipse is drawn in red, the major axis in blue, and the extremities of the axis are small green circles.

$\{p_1, p_2\}_r^{target}$ be the extremities of the major ellipse axis expressed in the 2D coordinate frame of the left and right tracker, respectively[d].

3.2. *3D reconstruction*

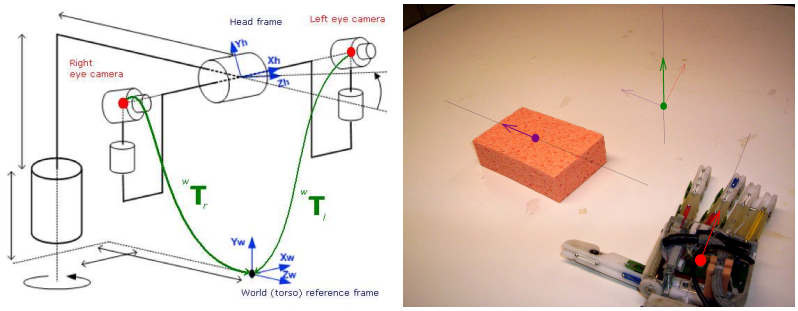

Fig. 3. Left: mechanical structure of Baltazar's head, reference frames and points of interest; transformation matrices are highlighted in green. Right: unit vector of the target object along its orientation axis (purple), versor and orientation of robot hand (red), and third axis resulting from their cross product, and corresponding unit vector (green).

A 3D reconstruction process receives the coordinates of the four points $\{p_1, p_2\}_{l,r}$ as inputs, along with the instantaneous head joint angle values of the robot, used to compute the time-varying extrinsic camera parameter

[d]The same considerations apply for stereo tracking and 3D reconstruction of the robot *hand*, but for the sake of simplicity only the target object case is explained in this paper. From now on, the "target" exponent in the notation is therefore omitted.

matrices: not just the target object, but also the robot cameras may be moving during experiments. Transformation matrices ${}^{w}\mathbf{T}_{l}$ and ${}^{w}\mathbf{T}_{r}$ represent the roto-translations occurring, respectively, from the left and right camera reference frame to the world (torso) reference frame, as shown in Fig. 3.

Once the reconstruction is computed, 3D coordinates of $\{p_1, p_2\}$ are obtained. The difference vector $p_1 - p_2$ encodes the orientation of the target.

4. Reaching and grasping preparation

As mentioned in Sec. 2 and Ref. 8, two distinct phases in reaching and grasping preparation are considered.

Reaching preparation: this first phase aims at bringing the robot hand to the vicinity of the target. It is applied whenever a target is identified in the workplace but the hand is not visible in the cameras. The measured 3D target position is used, in conjunction with the robot arm kinematics, to place the hand close to the target. Inevitably, there are mechanical calibration errors between arm kinematics and camera reference frames, so the actual placement of the hand will be different from the desired one. Therefore, the approach is to command the robot not to the exact position of the target but to a distance safe enough to avoid undesired contact both with the target and the workspace.

Grasping preparation: in this phase, both target and hand are visible in the camera system and their posture can be obtained by the methods previously described. The goal is now to measure the position and angular error between target and hand, and use a PBVS approach to make the hand converge to the target. The features used in such an approach are 3D parameters estimated from image measurements—as opposed to Image-Based Visual Servoing (IVBS), in which the features are 2D and immediately computed from image data. There are, however, two peculiarities in the presented approach:

(1) Normally, PBVS requires the 3D model of the observed object to be known,[4,6] but in this project one gets rid of this constraint: by using the stereo reconstruction technique explained in Sec. 3.2, the only condition to prepare the servoing task is that the CAMSHIFT trackers are actually following the desired objects—whose models are *not* known beforehand.

(2) Classical PBVS applications consider that target and end-effector positions are measured by different means, e.g. target is measured by the camera and end-effector is measured by robot kinematics. This usually leads to problems due to miscalibrations between the two sensory systems. Instead, in this work, target and hand positions are measured by the camera

system in the same reference frame, therefore the system becomes more robust to calibration errors.

Having computed 3D position and orientation of both a target object and of the robot hand, features suitable for the application of the PVBS technique must be obtained. As described in Ref. 4, the robot arm can be controlled by the following law:

$$\begin{cases} \mathbf{v} = -\lambda \left((\mathbf{t}_t - \mathbf{t}_h) + [\mathbf{t}_h]_\times \vartheta \mathbf{u} \right) \\ \omega = -\lambda \vartheta \mathbf{u} \end{cases} \tag{1}$$

where $\mathbf{v}$ and ω are the arm linear and angular velocities, λ establishes the trajectory convergence time, $\mathbf{t}_t$ and $\mathbf{t}_h$ are the target and hand positions; $\vartheta, \mathbf{u}$ are the angle-axis representation of the rotation required to align both orientations. Other control laws can be applied to this problem, but most of them rely on an angle-axis parameterization of the rotation. In this case, it is possible to calculate the required angle ϑ and axis $\mathbf{u}$ by applying a simple cross-product rule between the normalized hand and target orientation vectors:

$$\mathbf{u} = o_{target} \times o_{hand} \qquad \text{and} \qquad \vartheta = \arcsin \|\mathbf{u}\|_{L^2} \tag{2}$$

where $\| \cdot \|_{L^2}$ is the Euclidean norm, o_{target} is a unit vector in the direction of the target object's reconstructed orientation and o_{hand} is a unit vector in the direction of the hand's reconstructed orientation (see Fig. 3).

5. Experiments and results

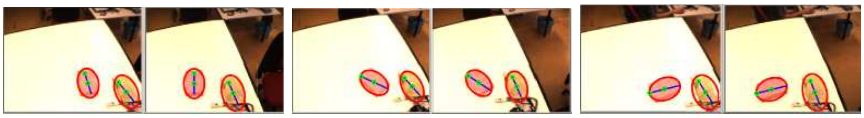

Fig. 4. Evaluated axis $\mathbf{u}$ and angle ϑ between tracked object and hand in several scenarios, in three different stereo pairs. Left: object and hand are parallel – $\mathbf{u} = (X = -0.006, Y = -0.062, Z = -0.041)$, $\vartheta = 4.285°$. Middle: about 45° – $\mathbf{u} = (-0.129, 0.737, -0.129)$, $\vartheta = 49.413°$. Right: orthogonality scenario – $\mathbf{u} = (-0.146, 0.919, 0.362)$, $\vartheta = 87.408°$.

Keeping in mind that the aim of this work is not high accuracy, but good qualitative estimations in order to interact with objects in front of the robot (see Sec. 1 and Ref. 10), the precision obtained is satisfactory. Fig. 4 shows the obtained results, estimated through Eq. (2).

6. Conclusions and future work

A simple algorithm for reaching and grasping preparation in a humanoid robot was presented in this paper. The method does not assume any particular shape model for the hand and objects, and it is robust to calibration errors. Although not relying on high precision measurements, the method will provide a humanoid robot with the minimal reaching and grasping capabilities for initiating the process of learning object manipulation skills from self-experience.

Future work includes evaluating the proposed technique with actual servoing and grasping experiments, as well as improving the pose estimation method by using the minor axis of ellipses in addition to the major one.

Acknowledgments

Work supported by EC Project IST-004370 RobotCub, and by the Portuguese Government - Fundação para a Ciência e Tecnologia (ISR/IST pluriannual funding) through the POS_Conhecimento Program that includes FEDER funds. The authors also want to thank Dr. Manuel Lopes for his guidance on visual servoing.

References

1. A. J. Abrantes, J. S. Marques, *The Mean Shift Algorithm and the Unified Framework*, ICPR, p. I: 244–247, 2004.
2. J. G. Allen, R. Y. D. Xu, J. S. Jin, *Object Tracking Using CamShift Algorithm and Multiple Quantized Feature Spaces*, 2003 Pan-Sydney Area Workshop on Visual Information Processing, Vol. 36, pp. 3–7, 2004.
3. G. R. Bradski, *Computer Vision Face Tracking for Use in a Perceptual User Interface*, Intel Technology Journal, 2nd Quarter 1998.
4. F. Chaumette, S. Hutchinson, *Visual Servo Control, Part I: Basic Approaches*, IEEE Robotics & Automation Magazine, Vol. 13, Issue 4, 2006.
5. Y. Dufournaud, R. Horaud, L. Quan, *Robot Stereo-hand Coordination for Grasping Curved Parts*, BMVC, pp. 760–769, 1998.
6. S. Hutchinson, G. D. Hager, P. I. Corke, *A Tutorial on Visual Servo Control*, IEEE Transactions on Robotics and Automation, Vol. 12, Issue 5, 1996.
7. M. Lopes, R. Beira, M. Praça, J. Santos-Victor, *An anthropomorphic robot torso for imitation: design and experiments*, IROS 2004, Japan, 2004.
8. M. Lopes, A. Bernardino, J. Santos-Victor, *A Developmental Roadmap for Task Learning by Imitation in Humanoid Robots: Baltazar's Story*, AISB 2005 Symposium on Imitation in Animals and Artifacts, UK, 12-14 April 2005.
9. G. Metta *et al.*, *The RobotCub Project: An Open Framework for Research in Embodied Cognition*, IEEE-RAS ICHR, December 2005.
10. L. Montesano, M. Lopes, A. Bernardino, J. Santos-Victor, *Learning Object Affordances: From Sensory-Motor Maps to Imitation*, IEEE Transactions on Robotics, Special Issue on Bio-Robotics, Vol. 24(1), February 2008.