Markerless visual servoing for humanoid robot platforms

Claudio Fantacci\(^1\), Ugo Pattacini\(^1\), Vadim Tikhanoff\(^1\) and Lorenzo Natale\(^1\)

ABSTRACT

Recent surge of interest in humanoid robots and their use in private or public contexts has risen the need for robust and resilient techniques for manipulation and interaction tasks [1]. These contexts present real-world challenges in that the environment is unstructured, complex and time varying. Precise and reliable manipulation and interaction tasks can be achieved when accurate knowledge of the end-effector pose is available. This is possible for industrial settings, where it is required to repeat similar tasks over time, in a fine-calibrated setting and in a well-known and structured environment. Many humanoid robots instead: I) are supposed to act in dynamic and unknown environment wherein object poses and shapes are unknown and II) have unreliable proprioception due to measurement noises, sensor biases, mechanical elasticity of the links and so forth. With these motivations, we propose a framework towards addressing I, II in the context of grasping tasks. In particular, we use vision for compensating for the robot’s proprioception errors and, as a result, such a refined information allows designing of a visual servoing control [2]–[4] for precise reaching and grasping. Our approach is markerless and makes use of stereo vision information and RGB images. As main contribution to our previous work [5], we propose an image-based visual servoing approach with decoupled translation and orientation controls.

The framework we propose for markerless visual servoing consists of the following steps (cfr. Fig. 1):

S1. The grasping goal pose is set by using the 3D point cloud acquired from stereo vision.
S2. An open loop phase brings the robot’s end-effector in the proximity of the object and in the cameras field-of-views, then the 3D model-aided particle filter of [5] estimates the end-effector pose using RGB images.
S3. Visual servoing uses the particle filter output of S4 in order to reach for the pose computed in S2.
S4. Reaching completes and the robot grasps the object.

Within S2, the 3D model-aided particle filter of [5] estimates the 6D pose (position and orientation) of the robot’s end-effector without the use of markers. To yield out estimates we resort to Computer Aided Design (CAD) schematics of the robot in order to render 3D mesh models of the end-effector as they would appear from the robots viewpoints (cfr. Fig. 2). In particular: 1) we render, for each particle, an image of the 3D mesh model of the end-effector as it would appear from the robot’s viewpoints (likewise in augmented reality contexts); 2) we then use this state representation to directly estimate the 6D pose of the end-effector in the robot operative space using 2D image descriptors. In particular, we use Histograms of Oriented Gradients (HOG) [6] to compare the rendered images with the robot’s camera images and, as a result, the particles which are more likely to represent the end-effector will have higher weight. Further details and a pseudo code of the 3D model-aided SIS PF can be found in [5].

During S3, a visual servoing controller commands the robot’s end-effector for accurately reaching a desired pose. A good visual servoing approach for humanoid robots requires the design of a robust and reliable control law and a human-like motion of the upper-body. The two main ingredients to design a visual servo control are the goal pose \(x^g\) and the current pose of the end-effector \(x^e\), which is given by the 3D model-aided particle filter of [5]. The visual servoing objective is to minimize the error

\[
\begin{align*}
e_k & \triangleq s(x^e_k) - s(x^g) = s^e_k - s^g, \\
\end{align*}
\]

where \(s^e_k\) and \(s^g\) are some feature representing, respectively, the manipulator and the goal pose. Once a feature \(s\) is selected, the aim is to design a velocity controller

\[
\dot{x}^e = -K^e J^e \dot{e},
\]

1 Claudio Fantacci, Ugo Pattacini, Vadim Tikhanoff and Lorenzo Natale are with Istituto Italiano di Tecnologia, iCub Facility, Humanoid Sensing and Perception, Via Morego 30, Genova, Italy claudio.fantacci@iit.it, ugo.pattacini@iit.it, vadim.tikhanoff@iit.it, lorenzo.natale@iit.it
where $K^e > 0$ is a proportional gain, $J \in \mathbb{R}^{m \times 6}$ is the feature Jacobian, or simply Jacobian, and $J^\dagger$ its Moore-Penrose pseudo-inverse.

We adopt the image-based visual servoing approach [4], which is preferable over position-based visual servoing, because it allows precise control despite errors in the extrinsic camera parameters. Thus, we use image-plane coordinates of a set of points to define the feature vector $s^*$. Unfortunately, in our settings, popular visual servo controllers [4] provide unsatisfactory camera velocity profiles and unexpected translation motion when a rotation is needed to reach the goal pose. As a result, unsuitable Cartesian trajectory of the end-effector are generated (see the dashed lines in Fig. 3). To tackle this problem, we propose a new image-based visual servoing control that provides satisfactory Cartesian trajectories.

Our approach considers two image-based visual servoing problems to be solved. The first solves for the translation motion assuming the rotation completed. This is equivalent to consider the current pose $x^e_t$ of the end-effector as the combination of the 3D Cartesian component of $x^e$ and of the orientation of $x^g$. Conversely, in the second problem we compute the rotation motion under the assumption of achieved translation, which is equivalent to consider the current pose $x^g_o$ of the end-effector as the combination of the 3D Cartesian component of $x^g$ and of the orientation of $x^e$. We then proceed with the classic approach presented in [4], defining four coplanar 3D points around $x^e_t$ and $x^e_o$ and by computing the two image Jacobians $J^e_t$ and $J^e_o$ to synthesize two velocity controller as defined in (2). The resulting trajectory turns out to be satisfactory, combining a decoupled translation and rotation motion. A comparison view of the trajectories is shown in Fig. 3.

We tested effectiveness and robustness of the proposed framework on the iCub humanoid robot platform. We achieved sub-pixel precision while performing reaching and grasping tasks, decreasing the Cartesian error by an order of magnitude w.r.t. using the pose of the hand provided by the direct kinematics achieving millimeter precision. The C++ implementation of our method is freely available on Github\(^1\).

![Image 1](image1.png)  ![Image 2](image2.png)  ![Image 3](image3.png)

**Fig. 2.** Left: mechanical model of the iCub right arm. Center: image from the left camera of iCub. Right: rendered image of the right end-effector (hand) of the iCub.

**Fig. 3.** Left camera view of four image-plane trajectories performed by the right end-effector using different image Jacobians. The green and red crosses represent, respectively, the initial and final position of the end-effector. The reaching task was carried out in simulation to avoid damaging the robot, and it mainly consists of a translation toward the left and a small positive rotation.

**REFERENCES**


\(^1\) github.com/robotology/visual-tracking-control/tree/master