To appear in *Advanced Robotics*
Vol. 00, No. 00, Month 201X, 1–18

FULL PAPER

A Novel Pipeline for Bi-manual Handover Task

G. Vezzani[a,b*], M. Regoli[a,b], U. Pattacini[a] and L. Natale[a]

[a] *iCub Facility, Istituto Italiano di Tecnologia, Genova, Italy*; [b] *University of Genova, Italy*;

(*v1.0 released March 2017*)

This paper addresses the problem of bi-manual object handover with a humanoid robot, i.e. the task of passing objects from one hand to the other. Bi-manual coordination is fundamental for improving manipulation capabilities of humanoid robots. We propose a novel and effective pipeline that tackles the problem by using visual and tactile information. Given the object in one of the robot hand (first hand), the object in-hand pose is estimated by a localization algorithm, which makes use of vision and tactile information. Then, the estimated pose is used in order to automatically choose a suitable pose for the second hand among a set of candidates *a-priori* annotated on the object model. The selected pose is finally used to accomplish the handover task. The performance of our approach has been assessed on a real robotic system, the iCub humanoid robot, on a set of objects from the YCB dataset. Experiments show that the proposed method allows performing proper and reliable handovers with different every-day objects.

Keywords: Handover; Manipulation planning; Tactile sensors; Object localization.

1. Introduction

The manipulation capabilities of humanoid robots can be significantly improved in many common tasks by using two hands at the same time. Bi-manual approaches double the robot workspace and allow solving problems that are unfeasible with a single arm, such as object re-grasping and manipulation of large and heavy objects that is impossible with one hand alone.

In this paper, we focus on the bi-manual handover task, for which the robot is asked to pass an object from one hand to the other hand. This ability is fundamental in diverse manipulation contexts. For instance, in a pick-and-place scenario if the robot is given an object in its left hand and is asked to put it on a target location in right hand workspace, the most reasonable movement requires passing the object from the left to the right hand. Bi-manual transfer allows also re-grasping objects, which can be useful to achieve task-specific grip.

The first main contribution of our work consists in a novel and reliable pipeline that allows performing the handover task with a real humanoid robot (the iCub humanoid robot [1]) and with different every-day objects. The pipeline takes advantage of our previous work on tactile-driven object manipulation and localization as well as self-touch, conveniently adapted and connected together for tackling the entire handover problem. The second main contribution is represented by a pose selection method we designed and integrated in the pipeline for selecting the best pose for the handover task among a set of *a-priori* poses. The chosen pose maximizes the distance between the two hands and the manipulability index of a two-arms kinematic chain.

The paper is organized as follows. Section 2 reviews the state-of-art on bi-manual handover

*Corresponding author. Email: giulia.vezzani@iit.it

systems. Then, Section 3 introduces the pipeline we designed, together with a detailed description of all its steps. Section 4 validates our approach by analyzing the results of each pipeline steps and showing a set of successful handovers performed by the robot with different every-day objects. Finally, Section 5 ends the paper with concluding remarks and perspectives for future work.

2. Related work

One of the earliest contributions on bi-manual coordination is Koga *et al.* in [2], in which the authors address the problem of planning the path of two cooperating robot arms to carry an object from an initial configuration to a goal configuration amidst obstacles. The paper compares three 2D planning techniques with different arms (2-DOFs[1] and 3-DOFs) in different scenarios (without and with obstacles). This is a seminal work since it was the first to formalize the concept of transit and transfer motion planning (later defined as multi-motion planning [3]). These results were later extended to 3D planning [4, 5].

Another interesting work is described in [6], tackling pick-and-place task by planning the motion of a dual-arm robot. The starting and the goal configurations of the object constrain the robot to grasp the object with one hand, to pass it to the other hand, before placing it in its final configuration. The framework proposed in [6] deals with the complete pipeline providing the grasping poses for arbitrarly-shaped objects, the solution of handover configurations and the actual control of robot's movements. In order to improve the planner performance, a context-independent grasp list is computed offline for each hand and for the given object as well as an offline trajectory that will be adapted according to the environment. In [7], bi-manual re-grasping is cast as an optimization problem, where the objective is to minimize execution time. The optimization problem is supplemented with image processing and a uni-manual grasping algorithm based on machine learning that jointly identifies two good grasping points on the object and the proper orientations for each end-effector. The optimization algorithm exploits this data by finding the proper re-grasp location and orientation to minimize execution time.

Motion planning for bi-manual tasks on humanoid robots with a high number of DOFs requires computationally efficient approaches to determine the robot full joint configuration at a given grasping position. In other words, we require to solve the Inverse Kinematics (IK) problem for one or both hands of the robot. This problem is described in [8] where the inverse kinematics problem and motion planning is achieved by combining a gradient-descent approach in the robot pre-computed reachability space with random sampling of free parameters. This strategy provides feasible IK solutions at a low computational cost without resorting to iterative methods that could be trapped by joint-limits. The work described in [9] provides another approach dealing with innovative IK solutions for bi-manual task. In that paper, instead of performing a bi-manual task with a parallel control of two kinematic chains, the authors transform the task into the control of a single chain including both the arms and a new inverse kinematic solver is proposed.

More recently, Dobson *et al.* implemented multi-arm handover using the transit and transfer motion planning framework proposed by Koga. A similar work was proposed by Cohen et al. in [10], with the main difference that Cohen's work strongly relies on heuristics. In [11], a hierarchical approach to planning sequences of non-prehensile and prehensile actions was proposed by splitting the problem into three stages (object contacts, object poses and robot contacts), thereby reducing the size of search space.

In this paper, we propose a novel pipeline for object transfer, which makes use of the IK solution proposed in [9] and visual and tactile information for maintaining a stable grip and localizing the object in the hand. We also propose a new strategy for selecting the best pose for positioning the other hand.

[1]Degrees of freedom.

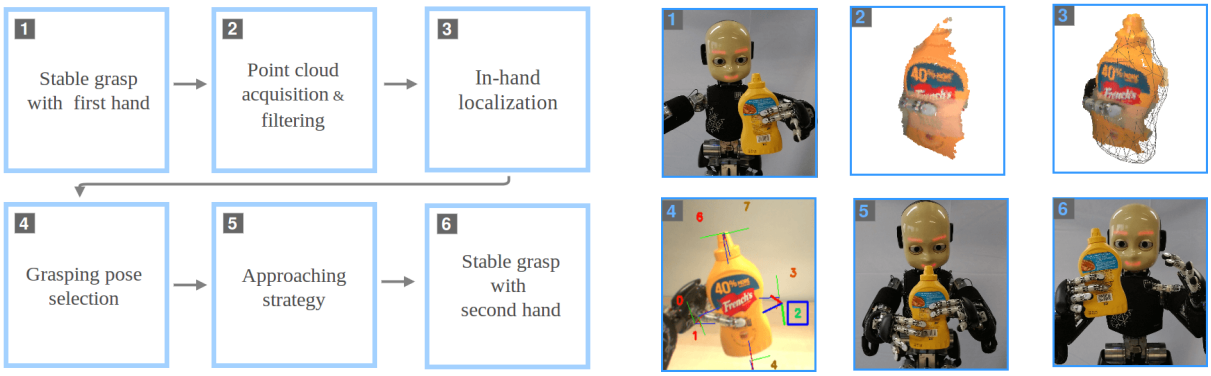


Figure 1. On the left: a sketch of the proposed pipeline. On the right: some snapshots from the execution of a real handover. 1) The robot grasps the object with the first hand by using tactile stabilization. 2) A set of 3D points of the object is acquired and filtered. 3) The point cloud is used by the localization algorithm for estimating the object pose. 4) Thus, the pose for the second hand which best satisfies to specific criteria is selected among a set of poses *a-priori* annotated on the object model. 5) Finally, both arms move and, once the second hand reaches the selected pose 6) the second hand grasps the object, which is contemporary released by the first hand.

3. Pipeline

The pipeline for bi-manual object transfer we propose is outlined in Figure 1. In practice, we ask the robot to pass a known object from one hand (that we refer to as *first hand*) to the other hand (named *second hand*).

The entire pipeline can be divided in the following steps:

- *Stable grasp with the first hand*: the robot grasps the desired object with the first hand by reaching a stable grasp using tactile feedback.
- *Point cloud acquisition and filtering:* the robot vision system provides 3D points of the closest blob in the field of view. Then, we properly filter the point cloud in order to extract only points belonging to the object surface, discarding instead those points that belong to the background or the hand.
- *In-hand localization:* a localization algorithm estimates the object in-hand pose by using the filtered 3D points.
- *Grasping pose selection:* the object model is *a-priori* annotated with a set of grasping poses reachable by the second hand. After the object is localized, we rank the candidates according to the distance from the first hand and the manipulability index of the two-arms kinematic chain. We then select the best pose for performing the handover.
- *Approaching strategy:* Both the robot arms move until they reach the selected pose. The approaching strategy is designed to avoid collisions between the second hand and the object.
- *Stable grasp with the second hand*: the robot grasps the object with the second hand and, once the grasp is stabilized, the first hand releases the object. The bi-manual handover is finally achieved.

In the next paragraphs we fully illustrate each step together with the methodology we implemented.

3.1 *Stable grasp with tactile feedback*

A stable grasp of the object is essential throughout the execution of the entire handover task. In fact, the movements of the object can compromise the localization reliability or cause the object to fall while the arms are moving. To this end, we adopted the grasp stabilization strategy described in [12], performing a three-finger precision grasp with the thumb, the index and the middle fingers. In this section we briefly revise this method. Fig. 2 shows the overall control schema, which is made of three main components:

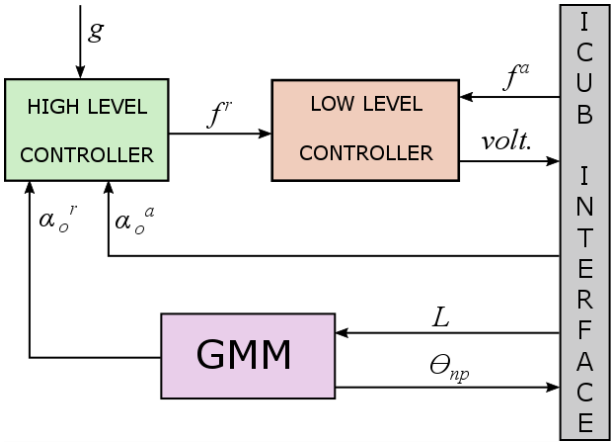


Figure 2. Grasp stabilizer control schema. While grasping an object, the set $\mathbf{L}$ of lengths of the edges of the triangle defined by the points of contact is used by the Gaussian mixture model to compute the reference values of the non-proximal joints $\mathbf{\Theta_{np}}$ and the object position α_o^r. In order to reach α_o^r and g, the high-level controller sets the appropriate force references $\mathbf{f^r}$ of the low-level controller for each finger. The low-level force controller, in turn, sends voltage to the motors actuating the proximal joints to compensate the force error. The actual object position and the actual forces at the fingertips are represented by, respectively, α_o^a and $\mathbf{f^a}$.

- The *low-level controller* is a set of P.I.D. force controllers, one per finger, which maintain a given force at the fingertips by sending an appropriate voltage signal to the motors actuating the proximal joints Θ_p. We estimate the force at each fingertip by taking the magnitude of the vector obtained by summing up all the normals at the sensor locations weighted by the sensor response.
- The *high-level controller* acts on top of the low-level controller and stabilizes the grasp by regulating the object position while delivering a specified grip strength. The object position α_o, defined in Fig. 3, is controlled resorting to a P.I.D. controller that adjusts the set-points of the forces at each finger to reduce the final position error.

  The grip strength is a weighted average between the force applied by the thumb and the forces applied by the index and middle as follows:

$$g = \frac{2}{3} \cdot f_{th} + \frac{1}{3} \cdot (f_{ind} + f_{mid}), \tag{1}$$

  where f_{th}, f_{ind} and f_{mid} are the forces at the thumb, the index and the middle fingers, respectively. This is obtained by exploiting some specific assumptions on the force model and the noise distribution, whose details can be found in [12]. The target grip strength is kept constant by choosing set-points of the forces that satisfy (1).
- The *gaussian mixture model* is a stable grasp model trained by demonstration. We collected several stable grasps using objects of different size and shape. The stability of a grasp was determined by visual inspection, avoiding non-zero momentum, unstable contacts between the object and the fingertips and grasp configurations that are close to joint limits. Given the set $\mathbf{L}$ of lengths of the edges of the triangle defined by the points of contact, which are related to the object shape, the model estimates the target object position α_o^r and the target set of non-proximal joints $\mathbf{\Theta_{np}}$ that improve grasp stability as well as the robustness. The target α_o^r is used as the set-point of the high-level controller, while the $\mathbf{\Theta_{np}}$ is commanded directly using a position controller.

The grasp stabilizer is triggered when all the fingertips are in contact with the object, which happens by closing all the fingers at constant speed. The fingers stop when they all exceed a given force threshold at the fingertip.

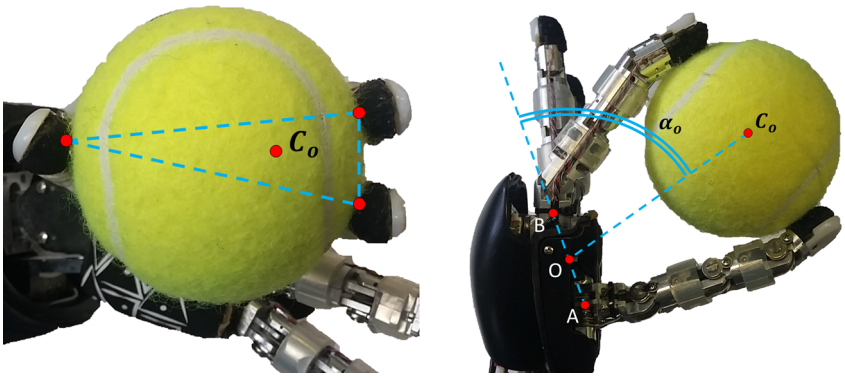


Figure 3. The object center C_o is defined as the centroid of the triangle identified by the three points of contact (left). The object position α_o is defined as the angle between the vectors $O\vec{C}_o$ and $\vec{OB}$ (right). A and B are set at the base of, respectively, the thumb and the middle finger, while O lies at middle distance between A and B.

(a) (b)

Figure 4. On the left: Point cloud obtained after the application of coarse filter. The point cloud includes also point belonging to the robot hand. On the right: the blue points represent the selected points after the hand filter. Notice that the points belonging to the hand are removed.

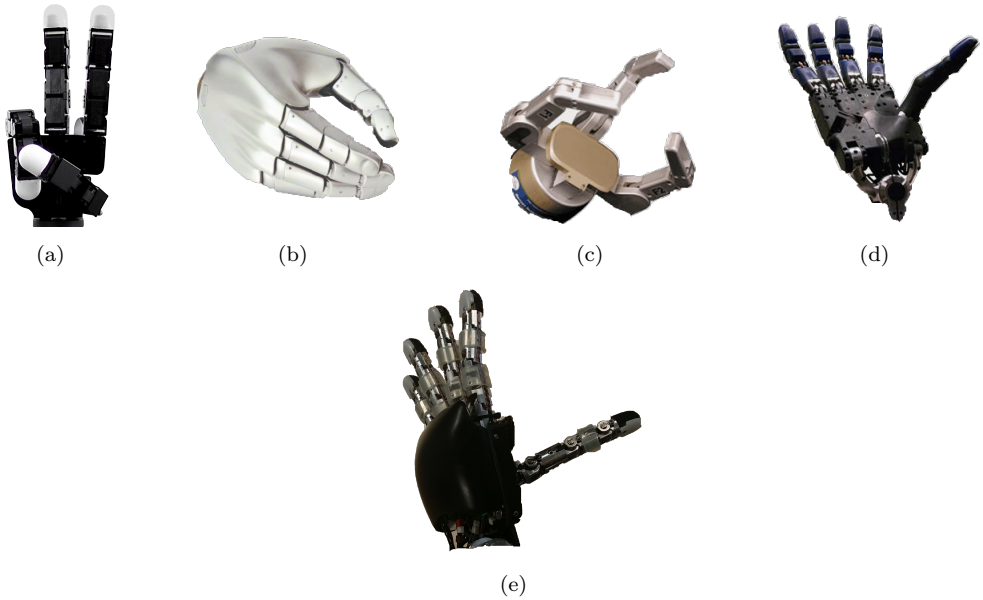
3.2 *Point cloud acquisition and filtering*

Once the object is stably held by the first hand, the nearest blob in the robot visual field is acquired from the stereo vision system. Such a blob contains 3D points belonging both to the visible portion of the object and to the robot hand (see Fig. 4(a)). Using point clouds including parts of the robot's hand would adversely affect the initial information about the object pose. For this reason, a pre-filtering process is required. We implemented two filters that are consequently applied to the point cloud:

a) the *coarse filter* removes possible points outside a volume *a-priori* defined around the robot hand. This filter is necessary in case of noisy initial point clouds, e.g. when the selected blob includes also portions of the background scene.
b) the *hand filter* is applied in order to discard the 3D points belonging to the robot hand. At this aim, the filter removes all points with a specific color property.

An example of filtered point cloud is shown in Fig. 4(b). The blue points represent the final point cloud after the filtering process. Hereafter, we describe in the detail the filters we designed.

The coarse filter implementation simply consists of an inside/outside test on the points coordinates. If the 3D point lies outside a 3D box built around the first hand, the point is discarded,



(a) (b) (c) (d)

(e)

Figure 5. Some examples of grayish robotic hands: (a) Allegro hand, (b) Wessling hand, (c) Barret hand, (d) Shadow hand and (e) the iCub hand, which is the platform we used for testing our approach.

otherwise is selected.

A principled way to remove the hand from the point cloud is to use the robot kinematics to project the hand model on the point cloud. In the case of the iCub robot, this approach is unsuitable because the forward kinematics is affected by errors due to the elasticity of the tendons [13]. To overcome this problem we propose to use a color filter. This solution assumes that the hand is gray and it corresponds to pixels with low saturations. Such an approach can be applied to other robotic hands (see Fig. 5). The filter selects all points for which a measure of saturation:

$$S = \frac{\sum_{l=1}^{l=L}(|R_l - G_l| + |R_l - B_l| + |B_l - G_l|)/3}{L} > \mu, \tag{2}$$

where R_l, G_l and B_l are the RGB values of point $l \in 1, \ldots, L$ and L is the number of points lying in a certain volume of radius r. The value of μ is chosen experimentally to deal with variability in light condition.

3.3 *In-hand localization*

In order to estimate the object in-hand pose, we adapted to our problem the Memory Unscented Particle Filter (MUPF) proposed in [14].

The MUPF is a recursive Bayesian estimation algorithm, capable of localizing real objects, whose models are known, through 3D points, with good performance. Such an algorithm relies on the Unscented Particle Filter (UPF) suitably adapted to the localization problem. The UPF jointly exploits the potentials of the particle filter for approximating multimodal distributions and of the unscented Kalman filter for efficiently generating the proposal distribution. The standard UPF algorithm has been modified with the inclusion of a suitable sliding memory (hence the name MUPF) of past measurements in the update of the particle importance weights.

Even though the MUPF was designed for tactile object localization, we can suitable adopt it for object in-hand localization of the handover pipeline for two reasons. First, the object pose can be considered time-invariant (as required by the MUPF) with respect to the hand pose since our grasp approach (Section 3.1) stabilizes the object in the robot hand (as validated in Section 4). Second, the filter is agnostic about measurements nature, as long as they consists of

Figure 6. An example of object model in the estimated pose, computed via the MUPF algorithm. The algorithm uses the filtered point cloud shown in Fig. 4(b).

the Cartesian positions of points lying on the object surface. For this reason, we can feed the algorithm with a subset of points belonging to the filtered point cloud (Section 3.2). Fig. 6 shows an example of the object model in the pose correctly estimated by the MUPF overlapped to the relative point cloud.

For the sake of completeness, we provide a brief explanation of the MUPF algorithm adapted to our case.

The Memory Unscented Particle Filter tackles the problem of object localization as a peculiar filtering problem. In the assumption that the object does not move, the entity to be estimated consists in the object pose and does not depend on time. For this reason, in our problem, the system state $\boldsymbol{x}$ is in R^6 and is defined by:

$$\boldsymbol{x} = \left[x,\, y,\, z,\, \phi,\, \theta,\, \psi\right]^T, \tag{3}$$

where x, y, z are the coordinates of the center of the reference system attached to the object model and ϕ, θ, ψ are the three Euler angles representing orientation.

The observations $\{\boldsymbol{y}_t\}_{t=1}^{L}$ exploited to localize the object consist of the Cartesian positions of the 3D points lying on the object surface and obtained from the vision system:

$$\{\boldsymbol{y}_t = (x_t, y_t, z_t)\}_{t=1}^{L}. \tag{4}$$

In order to correctly formulate the localization problem in the filtering framework, we need to define other mathematical quantities, as follows:

- Since the state to be estimated is stationary, the state transition equation can by expressed as:

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t + \boldsymbol{\omega}_t, \tag{5}$$

  where $\boldsymbol{\omega}_t$ is a small artificial noise [15]. This term is introduced in order to allow the filtering technique to change the estimate of $\boldsymbol{x}$ and then converge to the final solution.
- The likelihood function $\ell_t(\boldsymbol{y}_t|\boldsymbol{x}_t)$ is based on the so-called *proximity model* [16], in which the measurements are considered independent of each other and corrupted by Gaussian

noise. For each observation, the likelihood function depends on the distance between the measurement and the object model, hence the name "proximity". The likelihood is defined as:

$$\ell_t(\boldsymbol{y}_t|\boldsymbol{x}) \propto \max_i \ell_{t,i}(\boldsymbol{y}_t|\boldsymbol{x}), \tag{6}$$

where $\ell_{t,i}(\boldsymbol{y}_t|\boldsymbol{x})$ is assumed to be Gaussian, with variance σ_p^2 and amounts to:

$$\ell_{t,i}(\boldsymbol{y}_t|\boldsymbol{x}) = \frac{1}{\sqrt{2\pi}\sigma_p} \exp\left(-\frac{1}{2}\,\frac{d_i(\boldsymbol{y}_t,\boldsymbol{x})^2}{\sigma_p^2}\right), \tag{7}$$

with $d_i(\boldsymbol{y}_t,\boldsymbol{x})$ the shortest Euclidean distance of $\boldsymbol{y}_t$ from the face f_i of the object model when the object is in the pose $\boldsymbol{x}$.

The exploited measurements are relatively uninformative if used individually, since they are three-dimensional vectors in a 6D space. This fact implies that the standard UPF is not well suited to this problem, since it exploits only the current measurement $\boldsymbol{y}_t$ at each time step. Such a behavior is somewhat critical, because the algorithm might end up limiting the search within wrong sub-regions, thus ruling out potential representative solutions.

In order to overcome this drawback, [14] proposes the use of a limited number of past measurements during each iteration. The importance weights $\{w_t^i\}_{i=1}^N$ are updated by resorting also to past observations. In case the number of acquisitions remains limited, a growing memory strategy can be adopted by computing the importance weights with all the measurements collected up to time t. On the other hand, in case of a larger number of measurements, the computational burden can be reduced by following a moving window strategy, where only a given number m of the most recent acquisitions are used at each time instant. In our application, where the number of measurements L is limited to a subset of the 3D points acquired by vision ($L = 100$), we use $m = L$, because it does not seriously affect the computational cost and all the L measurements are acquired at the same time from the robot vision.

After all the L measurements have been processed, i.e. when $t = L$, the algorithm outputs, as final estimate of the object configuration, the corrected particle $\bar{\boldsymbol{x}}_L^i$ corresponding to the highest value of the estimated posterior distribution $\hat{p}_{L|L}(\cdot)$. The adoption of a maximum *a posteriori* probability (MAP) criterion is motivated by the strongly multimodal nature of the density, due to the fact that, in the presence of symmetries in the object, there might exist multiple values of $\boldsymbol{x}$ compatible with the measurements. In fact, in a multimodal case, taking the expected value as estimate is not meaningful. Hence, the particle with the MAP probability [17] can be readily obtained as:

$$\hat{\boldsymbol{x}} = \arg \max_{j\in\{1,\dots,k\}} \hat{p}_{L|L}(\bar{\boldsymbol{x}}_L^j), \tag{8}$$

where $\hat{p}_{L|L}(\cdot)$ is the estimated posterior.

3.4 *Pose selection*

The object mesh model is *a-priori* annotated with N pose candidates for the second hand. These poses represent the minimum set of stable grasps feasible for the object. We compute them by using the Grasp-Studio library provided with the Simox toolbox [18], that offers the possibility to obtain grasping poses for several robotic end-effectors, including the iCub hand. We provide more details about the model *a-priori* annotation during the experimental evaluation of Section 4.

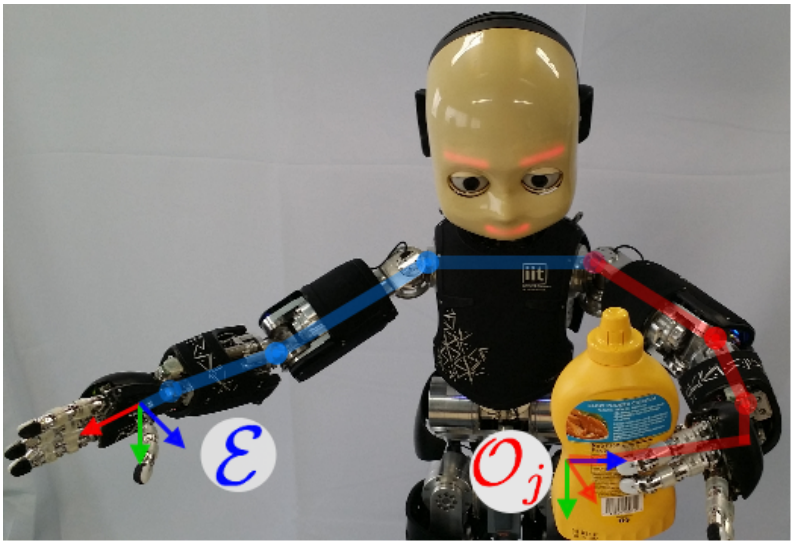


Figure 7. An outline of the two-arms chain we exploit for the handover task. The new chain origin $\mathcal{O}_j$ is located in correspondence to the j-th pose on the object, held by the left hand. The first part of the chain is reversed with respect to a standard chain for a robotic arm (colored in red). The remaining part (in blue) is a direct chain. The new chain end-effector $\mathcal{E}$ is located on the right palm.

Once the object is localized by the MUPF, its model – together with the annotated poses – is considered attached to the first hand according to the estimated pose (Fig. 8(a)).

Our approach consists in selecting that pose among the N candidates, which allows performing the best handover tasks according to some evaluation criteria and given the estimated object pose and the current robot arms configurations. At this aim, instead of modeling the arms as two kinematic chains with n-DOFs each, we represent them as a single $2n$-DOFs chain. In fact, classical approaches, where the arms are controlled as two separate chains, lead to several difficulties, as remarked in [9]; in particular, they require heuristics for defining a common region easily reachable by both arms.

All these reasons suggest the use of a two-arms chain with the *origin* $\mathcal{O}$ located at the end-effector of the first hand and the end-effector $\mathcal{E}$ of the two-arms chain located on the end-effector of the second hand. Specifically for the handover task, we refer to the origin with $\mathcal{O}_j$, since it coincides with the j-th pose candidate, with $j \in 1, \dots N$. This situation is depicted in Fig. 7. Under this formulation, the handover task involves moving the end-effector $\mathcal{E}$ to the origin $\mathcal{O}_j$ of the two-arms kinematic chain. From a mathematical viewpoint, we need to compute the values of joint angles $\boldsymbol{q}_j^*$ such that the end-effector $\mathcal{E}$ reaches the origin $\mathcal{O}_j$ both in orientation and position. Such a problem requires the reversal of the serial chain of the first arm with floating base. In particular, the first part of the kinematic chain is reversed with respect to the standard chains of robot arms, because it is traversed upside down from the origin $\mathcal{O}_j$ to the shoulder. The description of this type of kinematic chain in Denavit Hartenberg (DH) convention [19] is proposed in [9], where the authors provide the algorithm to derive the corresponding DH transformation matrix for each reversed link. We exploit this result for modeling the two-arms kinematic chain.

The joint angles $\boldsymbol{q}_j^*$ for performing the handover task with the j-th pose can be obtained as follows:

$$\boldsymbol{q}_j^* = \arg \min_{\boldsymbol{q} \in \mathrm{R}^{2n}} (\|I - K_\alpha^{\mathcal{O}_j}(\boldsymbol{q})\|^2)$$

subject to: (9)

$$\begin{cases} \|K_x^{\mathcal{O}_j}(\boldsymbol{q})\|^2 < \epsilon \\ \boldsymbol{q}_l < \boldsymbol{q} < \boldsymbol{q}_u \end{cases},$$

where $I \in \mathrm{R}^{3\times3}$ is the identity matrix, $K_x^{\mathcal{O}_j}(\cdot) \in \mathrm{R}^3$ and $K_\alpha^{\mathcal{O}_j}(\cdot) \in \mathrm{R}^{3\times3}$ are the forward kinematic functions that represents the position and the orientation of the end-effector $\mathcal{E}$ with respect to the origin $\mathcal{O}_j$; $\boldsymbol{q}_l$ and $\boldsymbol{q}_u \in \mathrm{R}^{2n}$ are vectors describing the joints lower and upper limits; ϵ is a parameter for tuning the precision of the reaching movements, tipycally $\epsilon \in [10^{-5}, 10^{-4}]$. The cost function of Eq. (9) imposes the minimization of the error between the end-effector $\mathcal{E}$ and the origin $\mathcal{O}_j$ orientations. The constraints take into account the error between the end-effector $\mathcal{E}$ and the origin $\mathcal{O}_j$ positions and require the solution $\boldsymbol{q}_j^*$ to lie between a set of lower and upper bounds of physically admissible values for the joints. As fully explained in [20], this formulation gives higher priority to the control of the position with respect to the orientation. The former is in fact handled as a nonlinear constraint and is evaluated before the cost function. We require a perfect reaching in position, whereas we can handle small errors in orientations relying on the robustness of our grasp controller.

We solve the problem described in Eq. (9) for each pose candidate j, $for\ j = 1, \ldots, N$ using Ipopt [21], thus obtaining the desired joints values to perform the handover with all the possible poses, i.e. $\{\boldsymbol{q}_j^*\}_{j=1}^{j=N}$. The latter N solutions do intrinsically encode suitable configurations for both the arms, without the need for dedicated heuristics to specify *a-priori* reachable regions where to perform the handover. Then, we execute two sequential rankings on the N poses in order to select the best one for the handover task.

First, the N candidates are ranked according to:

- the **distance d_j of the first hand from the origin** $\mathcal{O}_j$ (which represents the target pose):

$$d_j = \|\boldsymbol{p}_{h,j}\|, \tag{10}$$

  where $\boldsymbol{p}_{h,j} \in \mathrm{R}^3$ is the first hand position in the reference frame of the origin $\mathcal{O}_j$. The origin $\mathcal{O}_j$ represents the pose the second hand should reach during the handover. Thus, pose candidates j with larger values of d_j are given a higher score in the ranking, since their probability of collision with the first hand is lower.

Then, we update the first pose ranking by taking into account:

- the **manipulability index** of the **two-arms chain** m_j, in order to favor poses easily reachable by the robot arms:

$$m_j = \sqrt{det(J(\boldsymbol{q}_j^*)J(\boldsymbol{q}_j^*)^T)}, \tag{11}$$

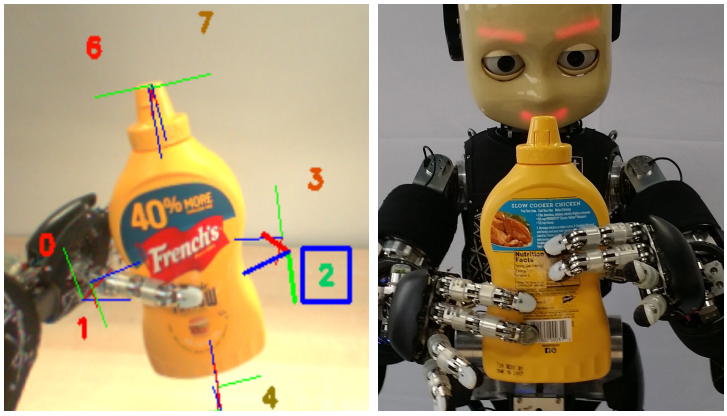
  where J is the jacobian of the kinematic chain, $\boldsymbol{q}_j^* \in \mathrm{R}^{2n}$ are the joints values of the two-arms chain which allow performing the handover with the j-th pose and n is the number of DOFs of a single arm.

In summary, the two sequential rankings applied on the N candidates provide as best pose that pose j^* with the maximum distance from the first hand and with the higher manipulability index of the two-arms chain (see Fig. 8(a)).

3.5 *Approach and handover*

The robot exploits the joints values $\boldsymbol{q}_{j^*}^*$, computed solving Eq. (9) which corresponds to the selected pose j^*, to move the arms toward the handover pose. In addition, the second hand passes by an intermediate waypoint so that to avoid its fingers hitting the object during the approach. The waypoint is simply obtained by shifting the final pose at a fixed distance from the object, along the x and z axis of the hand reference frame.

When the arms reach the final pose, the second hand grasps the object by using the approach



(a) (b)

Figure 8. On the left: an example of pose ranking. The numbers associated to each pose are colored according to the pose score, ranging from red for the worst pose up to green for the best one. In this case, our method correctly selects pose no. 2, i.e. $j^* = 2$.
On the right: the hands holding the object before passing the object from the first to the second hand.

Figure 9. The iCub platform has been used for testing the proposed approach for the handover task.

described in Section 3.1 (Fig. 8(b)). The second hand pose j^*, in fact, aims at suitably locate the hand close to the object surface with a proper orientation, leaving the actual grasping task to the grasp controller. Finally, the first hand opens and leaves the object in the second hand.

4. Results

In order to validate our approach, we tested the pipeline shown in Fig. 1 on the iCub humanoid robot (Fig. 9). Our implementation of the handover pipeline is availble on GitHub[1].

We carried out our experiments using a set of 5 objects, shown in Fig. 10. The objects were deliberately selected among the YCB Object & Model set [22] so as to be different in shape, dimensions and surface texture. We extracted the mesh models of the objects by applying the Poisson Surface Reconstruction algorithm [23] to the merged point clouds provided by the YCB dataset.

Without loss of generality, we illustrate the results obtained in case the left hand and the

[1]https://github.com/tacman-fp7/handover, DOI:105281/zenodo.437739.

Figure 10. The set of objects used in the experiments belonging to the YCB Object & Model set. We refer to the objects as: Sugar box, Chocolate box, Mustard box, Chips tube and Little cup.

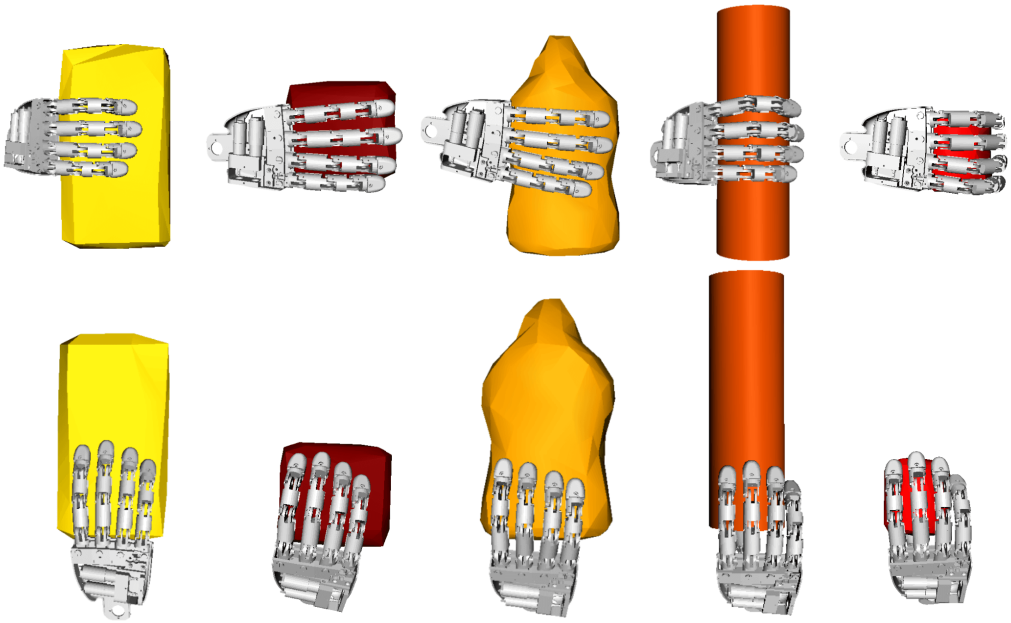
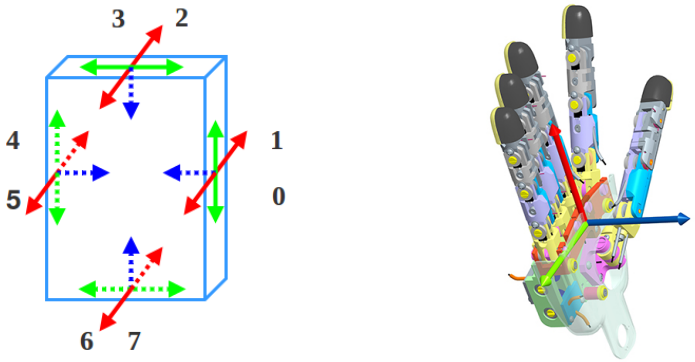


Figure 11. Some examples of poses generated with Grasp-Studio. The final set of poses is obtained by multiplying the basic poses according to the symmetry axes of each object.

right hand are, respectively, the first and the second hand of the handover task.

We annotated the mesh model by using the Grasp-Studio library. In particular, we used the implemented planner for computing the candidate poses. We selected a subset of poses (Fig. 11) among the planner solutions, discarding those that were visibly unstable. Then, we duplicated and rotated the selected poses in order to deal with the object symmetries. This is a crucial point, because multiple correct solutions of the localization problem are available for symmetric objects. All the models we generated have several symmetries due to their shape. In addition, we only consider the geometric properties of the models, without exploiting information about surfaces color or texture. Fig. 12(a) provides an example of the minimum set of poses (for the iCub right hand) we have to consider for box-like objects. Pose annotation shown in Fig. 12(a) is in fact invariant with respect to 180-degree rotations of the object along x-, y- or z- axis



(a) (b)

Figure 12. For box-like objects, 8 poses are enough to take into account all the grasping scenarios that might happen, due to the object symmetries (a). Fig. 12(b) illustrate the reference frame of the right hand of the robot iCub.

of its reference frame (located in the object barycenter). Fig. 12(b) illustrates how the hand reference frame is attached to the iCub right hand. More poses are necessary for cylindrical objects, due to their major symmetry. In conclusion, we generated 8 poses for box-like object (Sugar, Chocolate and Mustard box), 24 for the long tube (Chips tube) and 16 for the small cup (Little cup).

In the following paragraphs, we show the results obtained with the three main steps of our algorithm: *point cloud filtering*, *in-hand localization* and *pose selection*. Then, we evaluate the reliability of the entire pipeline computing the success rate of our approach for each object and in different poses.

Point cloud filtering

During filtering process we make use of the coarse and the hand filter with RGB coding, $r = 0.001$ and $\mu = 25$. Fig. 13 shows the point clouds after the coarse filter, on the top, and after the hand filter, on the bottom, for all the objects.

Although we obtain good results with the hand filter, this is a heuristic approach. The main weak point of the method arises when it is applied to grayish objects (see gray portions of Sugar box and Chips tube of Fig. 13). In these cases in fact, the point cloud saturation is not informative enough for distinguishing among the points belonging to the object or to the robot hand. A discussion on how we could eventually deal with grayish objects is presented in Section 5.

In-hand localization

We select a subset of 100 points (i.e. $L = 100$) of the filtered point clouds for the localization step. The object models in the estimated pose overlapped to the corresponding point clouds are collected in Fig. 14. The average MUPF execution time for each object is approximately 45 [s].

Pose selection

Fig. 15 shows the results obtained with our pose selection approach. For each object, the N candidates are overlapped on the camera image according to the estimated pose. The poses are labeled with numbers. Each number j is colored according to its score in the ranking, ranging from red (worst pose) up to bright green (best pose). The selected pose is indicated with a blue square. Table 1 collects the execution time required by the pose selection process.

The tests demonstrate the effectiveness of our approach since the best poses are located on the surfaces farther from the first hand and are better reachable by the second hand with two-arms movements. For example, in the bottom left image of Fig. 15, poses no. 0, 1, 6 and 7 have distances larger from the first hand. However, only poses no. 1 and 6 (colored in green)

Figure 13. An example of filtered point clouds after the coarse filter, on the top, and after the hand filter, on the bottom. The blue dots represent the final selected points. The hand is correctly removed from the point clouds of all the objects. Sugar box and Chips tube are examples of objects with grayish surface portions. Coherently with the filter behavior, those parts are discarded together with the robot hand. Nevertheless, the filtered point cloud is still representative of the object for the localization algorithm.

Figure 14. An example of estimated object poses for all the objects. Each mesh model is overlapped to the relative point cloud.

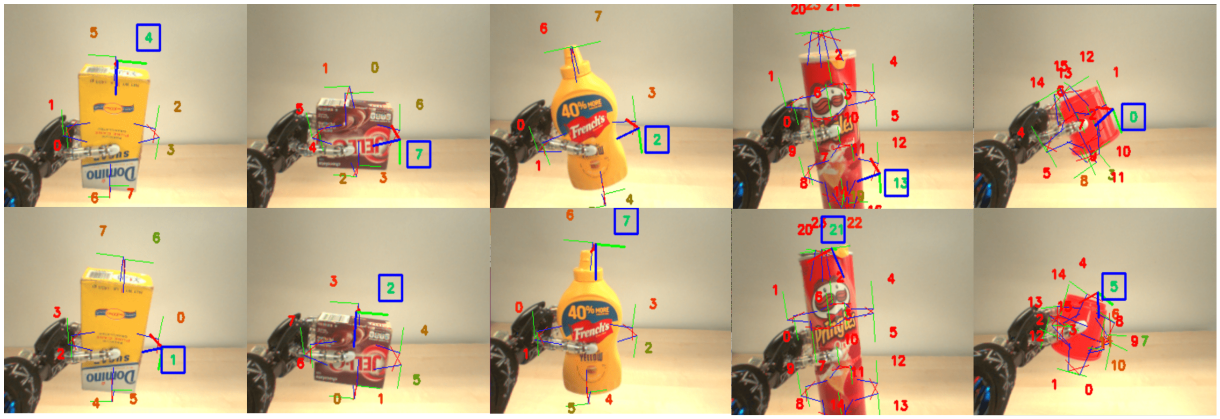


Figure 15. Some example of grasping pose selection results for the set of objects. Each images shows the pose annotated on the object model in the estimated pose. The poses are labeled with numbers, which are colored according to the pose score, ranging from red for the worst pose up to green for the selected pose.

Table 1. Computation time for pose selection step.

Object	Computation time [s]	Object	Computation time [s]
Sugar box	3.10	Chocolate box	3.10
Mustard box	3.10	Chips tube	9.45
Little cup	7.02		

Table 2. Success percentage of the handover task for each object and for different poses. We consider a handover successfully achieved if the object is held by the second hand without falling while the second arm is moving.

Object	Pose	Success rate	Pose	Success rate
Sugar box	Lateral	90%	Top	90%
Chocolate box	Lateral	90%	Top	100%
Mustard box	Lateral	80%	Bottom	80%
Chips tube	Lateral	80%		
Little cup	Lateral	50%		

can be selected because they can be reached with a better joints configuration of the robot arms.

Pipeline reliability

We executed 10 trials for each object and for different poses. Table 2 reports the percentage of success of the handovers (greater than 80% for the majority of the experiments). We consider the task successfully achieved if the object is held by the second hand without falling while the arm is moving. Some snapshots of successful handovers are shown in Fig. 16. We do not take into account the performance obtained with poses in which the second hand is located on the top of the Mustard box, the Chips tube and the Little cup. Due to the shape of their upper part and their slippery surface, even very small errors in the final reached pose compromise the success/outcome of the handover. These errors are mostly due to the errors in kinematics of the robot and noise in the point cloud.

In Table 3, we detail the three main causes of the tests failures. First, uninformative point clouds can be source of errors in object localization. For instance, if only one of the 8 faces of the Sugar box or the Chocolate box is acquired, multiple and wrong solutions of the localization algorithm can properly fit the point clouds. The second critical issue is represented by slippery objects, such as the Mustard box and the Chips tube. In this case, little displacements between the desired and the reached pose can lead to an unstable grasp and to failure. Finally, object size is crucial for the handover success. In case the object size is comparable with the hand dimensions, a fingers avoidance approach is necessary in order to prevent the second hand from blocking the first hand while grasping the object. The lack of such an avoidance approach in our method is the reason of the low reliability of the handover task with the Little cup.

5. Conclusions

This paper proposes a pipeline for the handover task by integrating various modules that make use of visual and tactile feedback, namely:

- a grasp controller, stabilizing the object in the robot hand using tactile feedback;
- a point cloud filter, extracting 3D points laying on the object surface from the closest blob

Table 3. Main causes of handover failures.

Object	Pose	Failure source
Sugar box	Lateral	Uninformative point cloud
	Top	Uninformative point cloud
Chocolate box	Lateral	Uninformative point cloud
	Top	None
Mustard box	Lateral	Slippery object surface
	Bottom	Slippery object surface
Chips tube	Lateral	Slippery object surface
Little cup	Lateral	Fingers overlapping

Figure 16. Examples of successful handovers.

in the robot view;

- an object localizer, named Memory Unscented Particle Filter, capable of reliably estimating the object in-hand pose by using the 3D points coming from vision;
- a pose selection strategy, which chooses the best pose for performing the handover by maximizing the distance between the first and the second hand and the manipulability index of a two-arms kinematic chain.

We evaluated our method experimentally with the iCub humanoid robot, showing that it provides a success percentage greater than 80% on 4 objects of the YCB Object & Model Set, different in shape, texture and dimensions.

The contributions of this paper and the experimental evaluation we carried out suggest perspectives for future work. One of the limitations of this work is that it makes use of a saturation filter to separate the hand from the object. This approach may fail when the object and the hand have similar color distributions (specifically, low saturation). To overcome this limitation, we could rely on an accurate kinematic model of the hand or hand tracking techniques [24]. In the particular case of the iCub hand, such an approach could allow performing the removal of the robot hand also from point clouds of grayish objects. In some cases, the robot had only a partial view of the object, which correspond to an ambiguous point

cloud and wrong localizations. If the positions of the hand is known with great accuracy, this can be fixed by fusing in the localization 3D points coming from the fingers in touch with the object. Another extension consists in recognizing the object autonomously. This could be done using techniques from computer vision [25] or by fitting object models on the point cloud acquired from the cameras and the contact points between the fingers and the objects [26].

Acknowledgment

This research has received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement No. 610967 (TACMAN).

References

[1] Metta G, Natale L, Nori F, Sandini G, Vernon D, Fadiga L, Von Hofsten C, Rosander K, Lopes M, Santos-Victor J, et al.. The iCub humanoid robot: An open-systems platform for research in cognitive development. Neural Networks. 2010;23(8):1125 –1134.
[2] Koga Y, Latombe JC. Experiments in dual-arm manipulation planning. In: IEEE International Conference on Robotics and Automation. 1992. p. 2238–2245.
[3] Hauser K, Latombe JC. Multi-modal motion planning in non-expansive spaces. The International Journal of Robotics Research. 2010;29(7):897–915.
[4] Koga Y, Latombe JC. On multi-arm manipulation planning. In: IEEE International Conference on Robotics and Automation. 1994. p. 945–952.
[5] Koga Y, Kondo K, Kuffner J, Latombe JC. Planning motions with intentions. In: Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques. 1994. p. 395–408.
[6] Saut JP, Gharbi M, Cortés J, Sidobre D, Siméon T. Planning pick-and-place tasks with two-hand regrasping. In: IEEE/RSJ International Conference on Intelligent Robots and Systems. 2010. p. 4528–4533.
[7] Balaguer B, Carpin S. Bimanual regrasping from unimanual machine learning. In: IEEE International Conference on Robotics and Automation. 2012. p. 3264–3270.
[8] Vahrenkamp N, Berenson D, Asfour T, Kuffner J, Dillmann R. Humanoid motion planning for dual-arm manipulation and re-grasping tasks. In: IEEE/RSJ International Conference on Intelligent Robots and Systems. 2009. p. 2464–2470.
[9] Roncone A, Hoffmann M, Pattacini U, Metta G. Automatic kinematic chain calibration using artificial skin: Self-touch in the icub humanoid robot. In: IEEE International Conference on Robotics and Automation. 2014. p. 2305–2312.
[10] Cohen B, Phillips M, Likhachev M. Planning single-arm manipulations with n-arm robots. In: The 8th Annual Symposium on Combinatorial Search. 2015.
[11] Lee G, Lozano-Pérez T, Kaelbling LP. Hierarchical planning for multi-contact non-prehensile manipulation. In: IEEE/RSJ International Conference on Intelligent Robots and Systems. 2015. p. 264–271.
[12] Regoli M, Pattacini U, Metta G, Natale L. Hierarchical grasp controller using tactile feedback. In: IEEE-RAS 16th International Conference on Humanoid Robots. 2016. p. 387–394.
[13] Fanello SR, Pattacini U, Gori I, Tikhanoff V, Randazzo M, Roncone A, Odone F, Metta G. 3d stereo estimation and fully automated learning of eye-hand coordination in humanoid robots. In: 14th IEEE-RAS International Conference on Humanoid Robots. 2014. p. 1028–1035.
[14] Vezzani G, Pattacini U, Battistelli G, Chisci L, Natale L. Memory unscented particle filter for 6-DOF tactile localization. In: preprint available at http://arxiv.org/abs/1607.02757v1. 2016.
[15] Simon D. Optimal state estimation. Hoboken, New Jersey: Wiley. 2006.
[16] Petrovskaya A, Khatib O. Global localization of objects via touch. IEEE Transactions on Robotics. 2011;27(3):569 – 585.
[17] Saha S, Boers Y, Driessen H, Mandal PK, Bagchi A. Particle based MAP state estimation: A

comparison. In: 12th International Conference on Information Fusion. 2009. p. 278 – 283, Seattle, USA.

[18] Vahrenkamp N, Kröhnert M, Ulbrich S, Asfour T, Metta G, Dillmann R, Sandini G. Simox: A Robotics Toolbox for Simulation, Motion and Grasp Planning. In: International Conference on Intelligent Autonomous Systems (IAS). 2012. p. 585–594.

[19] Denavit J. A kinematic notation for lower-pair mechanisms based on matrices. Transactions of the ASME Journal of Applied Mechanics. 1955;22:215 – 221.

[20] Pattacini U, Nori F, Natale L, Metta G, Sandini G. An experimental evaluation of a novel minimum-jerk cartesian controller for humanoid robots. In: IEEE/RSJ International Conference on Intelligent Robots and Systems. 2010. p. 1668–1674.

[21] Wächter A, Biegler L. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. Mathematical programming. 2006;.

[22] Calli B, Walsman A, Singh A, Srinivasa S, Abbeel P, Dollar AM. Benchmarking in manipulation research: The ycb object and model set and benchmarking protocols. In: preprint available athttps://arxiv.org/abs/1502.03143. 2015.

[23] Kazhdan M, Bolitho M, Hoppe H. Poisson surface reconstruction. In: 4th Eurographics Symposium on Geometry Processing. Vol. 7. 2006. p. 61 – 70.

[24] Fantacci C, Pattacini U, Tikhanoff V, Natale L. Visual end-effector tracking using a 3D model-aided particle filter for humanoid robot platforms. In: available online on: http://arxiv.org/abs/1703.04771. 2017.

[25] Pasquale G, Ciliberto C, Rosasco L, Natale L. Object identification from few examples by improving the invariance of a deep convolutional neural network. In: IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE. 2016. p. 4904–4911.

[26] Vezzani G, Jamali N, Pattacini U, Battistelli G, Chisci L, Natale L. A novel Bayesian filtering approach to tactile object recognition. In: 16th IEEE-RAS International Conference on Humanoid Robotics. Cancun, Mexico, 2016. p. 256 – 263.