HAL
open science

Grasping objects localized from uncertain point cloud data

Jean-Philippe Saut, Serena Ivaldi, Anis Sahbani, Philippe Bidaud

To cite this version:

Jean-Philippe Saut, Serena Ivaldi, Anis Sahbani, Philippe Bidaud. Grasping objects localized from uncertain point cloud data. Robotics and Autonomous Systems, 2014, 62 (12), pp.1742 - 1754. 10.1016/j.robot.2014.07.011 . hal-01095300

HAL Id: hal-01095300

https://inria.hal.science/hal-01095300v1

Submitted on 16 Jan 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Grasping objects localized from uncertain point cloud data

Jean-Philippe Saut[a], Serena Ivaldi[a,b], Anis Sahbani[a], Philippe Bidaud[a]

[a] *Institut des Systèmes Intelligents et de Robotique, Université Pierre et Marie Curie, ISIR- CNRS UMR 7222, Boite courrier 173, 4 Place Jussieu, 75252 Paris cedex 05 , France*
[b] *Intelligent Autonomous Systems Lab., TU Darmstadt, Germany*

Abstract

Robotic grasping is very sensitive to how accurate is the pose estimation of the object to grasp. Even a small error in the estimated pose may cause the planned grasp to fail. Several methods for robust grasp planning exploit the object geometry or tactile sensor feedback. However, object pose range estimation introduces specific uncertainties that can also be exploited to choose more robust grasps. We present a grasp planning method that explicitly considers the uncertainties on the visually-estimated object pose. We assume a known shape (e.g. primitive shape or triangle mesh), observed as a –possibly sparse– point cloud. The measured points are usually not uniformly distributed over the surface as the object is seen from a particular viewpoint; additionally this non-uniformity can be the result of heterogeneous textures over the object surface, when using stereo-vision algorithms based on robust feature-point matching. Consequently the pose estimation may be more accurate in some directions and contain unavoidable ambiguities.

The proposed grasp planner is based on a particle filter to estimate the object probability distribution as a discrete set. We show that, for grasping, some ambiguities are less unfavorable so the distribution can be used to select robust grasps. Some experiments are presented with the humanoid robot iCub and its stereo cameras.

Keywords: Robotic Grasping, Multi-fingered Hand, Inverse Kinematics

1. Introduction

In this paper we focus on grasping under uncertainty: precisely, we propose a method that takes explicitly into account the uncertainty in the object observations to compute the best grasping configuration. Our objective in this study is to improve the grasping capabilities of the humanoid iCub that is equipped with two low-resolution stereo cameras and non-sensorized fingers (see Figure 1).

We present a planner that computes grasp configurations from the observation, as a set of 3D points (point cloud), of an object with known shape (e.g. modeled as a primitive shape or a triangle mesh). The point set can be provided by mean of any existing range scanning method. Conducted with a humanoid robot, the presented experiments are based on stereo-vision with robust 2D-features matching, but the approach would apply as well to data obtained with a device like Microsoft's Kinect or a time-of-flight camera. The point cloud is generally not dense enough and too locally-distributed to lead to a unique pose estimation, not talking about noise. The points sparsity can be the result of the small size of the object or of an insufficient number of points to match in case of stereo-vision. The points being not well distributed is because the object is seen from its front with respect to the robot sensing device and also to the non-uniform distribution of features to match in case of stereo-vision. To use the non-uniqueness of the solution of the pose estimation problem, we choose to reason with a probability distribution of the object pose instead of one pose estimation. The pose distribution is estimated through the use of an annealed particle filter. A discrete set of grasps is computed from the object geometry (known by hypothesis) and a set of uniformly-distributed hand poses

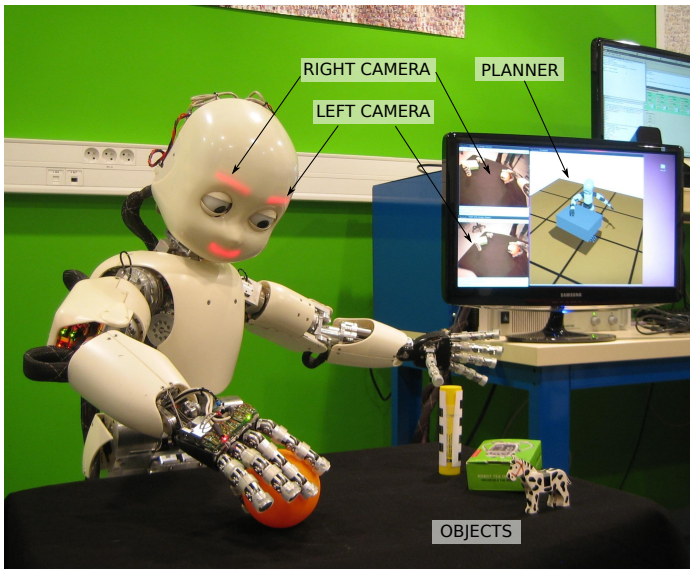


Figure 1: The humanoid robot iCub grasping objects localized by its eyes' cameras.

(with respect to the object). A score is then computed for each grasp that depends on the previously computed pose distribution and is meant to evaluate how robust is the grasp with respect to the uncertainty in the object pose.

The individual components of the method are not new, however their combination is unique to our knowledge. In particular, addressing the uncertainties related to a sparse stereo point cloud together with grasp planning has not been addressed earlier in a similar fashion. Although projected light solutions such as Kinect have gained significant attention lately, stereo processing still has important application areas. For example, in multi-robot systems where the overlap of multiple projected light sensors often causes difficulties, and in humanoid robotics where most vision relies on the embedded robot's eyes/cameras.

Our method addresses a major problem in autonomous robotic grasping. The error in the object pose estimation is inevitable, since the cameras can have low resolution (it is the case for our robot) and their calibration does not compensate perfectly the distortion in the whole image. When an object is observed by matching the features in the two camera images, the resulting point set is small, sparse and the quality of observations is very noisy, especially if compared with the point cloud of an external Kinect. The localization error can be thus greater than the one we would have with such external sensor. In our robot, this error cannot be compensated reactively by tactile feedback, because the fingers are not equipped with tactile sensors, nor visual feedback, because of the visual occlusions of the robot's body during the grasp.

The paper's contribution is explained more in detail in Section 2 that also gives an overview of its principles. Next subsection presents related works proposed in the literature.

1.1. Related Work

Considering uncertainty in robotic grasping strategies has known a revival these last years, due to the availability of many manipulation platforms usable in practical situations. As more and more robots are able to realize manipulation tasks, the goal is now to improve their robustness against all kind of perturbations. The intrinsic sensor inaccuracy leads to uncertain world models, which frequently cause failure of robotic tasks involving interaction with the robot's environment; typically during a grasping task Weisz and Allen [37].

The first works dealing with robotic grasping and uncertainty were published during the 80's. Brost [7] presented a method to grasp a polygonal planar object with a parallel-jaw gripper, in case the object location is not known precisely. A sequence of squeezing operations is planned in order to progressively reduce the possible uncertainty on the pose of the object. Each time the grasp is fully closed on the object, the current aperture of the gripper is used to refine the knowledge on the possible object poses. Lozano-Pérez et al. [24] proposed a planning strategy for

peg-in-hole insertion in the presence of uncertainty on the robot motion. An important idea of Lozano-Pérez et al. [24], which is useful in grasping in general, is the concept of pre-image. A pre-image is a set of points from which the goal can be attained in a single motion, like e.g. a simple translation. From the goal, a set of pre-images can be found, which represents the positions that can reach the goal in one motion. New sets of pre-images are iteratively developed until the start position belongs to a computed pre-image. Another way to deal with uncertainty is to design grasp quality scores that consider robustness against contact positioning error. In this purpose, the notion of *Independent Contact Regions* (ICR) was introduced by Nguyen [27]. ICRs are regions on the object surface –one per contact of the grasp– such that as long as each contact stays in its ICR, the grasp remains stable (i.e., verifies the force-closure property).

In the recent years, the interest in grasping and uncertainty has reappeared with a focus on real applications. Berenson et al. [3] represent the goal of grasping an object by one or several *Task Space Regions* (TSR). A TSR is a volume in SE(3), defined with bounds on the allowable translations and rotations. The uncertainness on the object position is represented as a set of possible poses. These different poses are used to duplicate and rotate the initial TSRs. The intersection of the TSRs defines then a new TSR that is the new goal region. Besides, a new collision model is built by duplicating the object model for each of the possible poses and merging the duplicates into one collision model.

The study of the relation between contact location uncertainty and grasp stability has also been carried on, with the design of more general and more practical algorithms to compute the ICRs of a grasp [30, 21], working with any kind of shapes, simply described as points sets with neighborhood information. The size of the ICRs of a grasp can tell how precise must be the placement of the contacts in order to guarantee the grasp stability.

Other approaches, based on machine learning with a learning database obtained in realistic conditions in terms of variety and noise, were also introduced in the literature. Stulp et al. [36] propose to learn the motion primitives that maximize the chance to successfully reach and grasp an object uncertainly localized. The object is placed on a table according to a particular distribution. Several grasping trials are performed by the robot in order to improve its reaching trajectory via a reinforcement learning procedure. It is shown that the robot adapts its motion to perform better for the considered distribution. Other works based on machine learning have also been proposed to deal with partial information. Saxena et al. [33] use supervised learning to estimate the probability of a point, in an image taken by the robot's camera, to correspond to a good grasping point on the object. A set of visual features is computed for each image point. A logistic regression is used to model the probability of a point to lead to a successful grasp as a function of its visual features. An additional classifier considers a point cloud of the scene, provided by a depth sensor, in order to estimate the quality of a grasp [34].

If humans have some hidden strategy to grasp uncertainly-located objects, it might be interesting to learn from human demonstration. Faria et al. [11] proposed a grasping framework based on learning by demonstration from humans. Objects are perceived as point clouds, which are segmented into three regions. A superquadric shape is then fitted to each region. Grasping experiments are conducted on humans to learn which is the region to grasp, given a task and an object shape (set of superquadrics). However, the grasping choice depends only on the selected task and on the object shape, which does not allow to consider robot perception issues.

If information about the perceived object is incomplete (e.g. hidden parts), it can be completed if some extra assumptions can be made about the object's shape. As many manufactured objects present some plane symmetry, Bohg et al. [6] propose to predict the shape of the whole object from a partial 3D scan. The optimal symmetry plane is found to be the one that leads to the minimal contradiction between the observed images and the points set obtained by mirroring the reconstructed points. From the optimal plane, new points can be inferred in the occluded area behind the visible points. The final object's model is then obtained by meshing the complete set of points.

Another way to cope with errors on object localization is to integrate some tactile exploration. For instance, Hsiao et al. [14] propose a set of simple and efficient heuristics for reactive grasping with application to the robot PR2. Behaviors like compliant gripper closing or grasp adjustment allow to reactively correct a possible misalignment of the gripper. Having a model of the object's shape, a more sophisticated approach is to refine the belief on the object's pose [15]. The robot tries to grasp the object at its most likely position and updates its belief about this position at each trial, using the hand tactile information. A physics simulator is used off-line to estimate the observation function used in the belief updating procedure. The Bayesian reasoning was also used by Laaksonen et al. [23] but with a data-driven approach i.e., where the different probabilistic models come from experiment instead of simulation.

In most aforementioned papers, the ranged estimation (i.e., before any contact) of the object pose is regarded

as unique, obtained from an algorithm that converges to one point (e.g. the *Iterative Closest Point* algorithm used in [14, 1]). This approach ignores unavoidable ambiguities that can occur due to object symmetries or insufficient measurement (e.g. not enough points in the cloud). Additionally this approach cannot discriminate grasps that are equivalent if the error is uniform but are not if the error is higher in some directions (e.g. a ball whose depth position is better known than position in the other directions). [20] recently proposed to model both shape (and consequently pose) uncertainties but were limited to a particular set-up with a 2D shape. The papers that implicitly integrate a non-uniform distribution of the pose error are based on data-driven models and machine learning [36, 34, 4, 23]. The advantage of not using models can however be a drawback in case the motion required to grasp the object is complex and needs a motion planning phase.

Finally, in this work we focus on the uncertainties in the object pose distribution from a noisy point cloud. In our experimental set-up the point cloud is provided by a stereo camera system. For interesting studies on uncertainty propagation for pose estimation from monocular point correspondences, we refer the reader to [13, 22].

2. Overview of Proposed Method and Contribution

In this paper, we are interested in computing grasping configurations of a robotic system. We assume the object to grasp has a known shape and is observed as a point cloud. The point cloud can be provided by a range scanning system, typically a stereo camera system or a lidar. The proposed planning method builds a representation of the object pose distribution as a particle set, computed through an annealed particle filter, from the observation of the object only (i.e., the point cloud). This representation lets us consider several modes in the distribution that can be due to poor observation data. We then choose, among the reachable grasps, the ones that are the most likely to succeed in spite of the incomplete knowledge of the pose. Consequently, the method presented in this paper offers the following contributions:

- The pose error is estimated from ranged data. This can avoid the use of tactile perception or at least give a better starting grasp for a grasping framework based on tactile information. This also makes the approach different and complementary to existing ones that are focused on tactile information (e.g. Laaksonen et al. [23]);
- The use of the annealed particle filter gives full pose estimation (6D) and requires nor a precise initial estimation or an appearance model but only a point cloud. This is an improvement compared to other works that are restricted to 2D poses [14] or require a set of views of the object from different angles [1].

Figure 2 shows the principle and interest of the proposed method. The object to grasp is a box. From a purely geometric perspective, most of the grasps are equivalent (the contacts are on two opposite parallel faces) even if we consider a stability criterion that takes into account error on each contact position like criteria based on ICR areas. However, from a perception point-of-view, the grasps can be very different. Indeed, as the object is observed from a particular viewpoint, the point cloud is not equally distributed over the object's surface. If the cloud is obtained via a stereo-vision algorithm based on 2D-features-matching, the bias can also come from how textured are some parts of the object compared to the others. In the example of Fig. 2, more points are visible on the front of the object, biasing the distribution of the points used to estimate the pose. This leads to a strong anisotropy in the distribution of the pose hypothesis probability (i.e., the particle set). As most of the points are on the object's front, the particles tend to be in a plane with normal pointing towards the viewer (far right image in Fig. 2-b), meaning the depth of the object is well estimated while its position along the other directions and its orientation around the depth-axis is poorly estimated. The anisotropy can be used to favor the grasps that are less sensitive to uncertainties left by the robot's perception (here, the grasps from the top).

The paper is organized as follows. Section 3 presents the pose estimation algorithm, based on an annealed particle filter. Section 4 explains how to choose a grasp from the computed distribution of the possible object poses and also deals with the planning of the collision-free path to grasp the object. Experimental results on the robot iCub as well as a description of the experiment platform are detailed in Section 5.

3. Pose Estimation

Annealed particle filters were recently applied with success in the context of manipulation, to estimate the object pose from contact information [9, 29]. We use a similar approach to estimate the pose of the object from its point

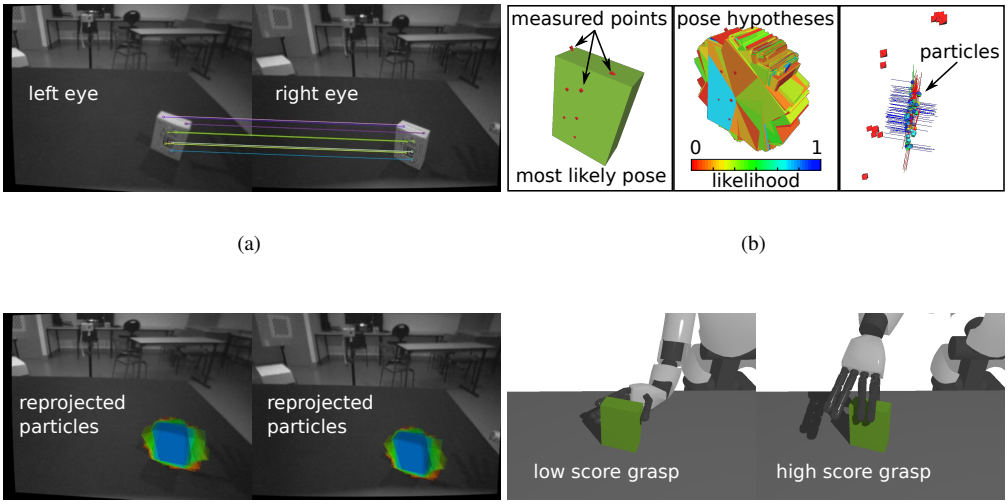


Figure 2: Using pose uncertainty to choose a robust grasp. (a) The points matched for the stereo triangulation. (b) The point cloud and the estimated pose with the highest likelihood; the set of the most likely particles (color gives likelihood); the particles drawn as frames (z axes drawn in blue) (c) The particles re-projected on the images. (d) Two grasps, equivalent in term of stability but the one from the top (right image) is the more likely to succeed for the knowledge the robot has of the object's pose. This is the choice made by our algorithm.

cloud. Point cloud registration is frequently done with the ICP algorithm [5] but this algorithm is not well-suited to deal with uncertainty. It only gives one estimation that corresponds to the convergence to a local minimum, so it is sensitive to the initial guess. Particle filters have the advantage of being able to represent a distribution; with a correct sampling strategy, they do not require a precise start estimation. We use the variant of annealed particle filter that was used for tactile exploration and described in [29]. The following section details the hypotheses and models we used to apply the filter to the pose estimation problem.

3.1. Hypotheses and Models

We consider an object with a known shape (e.g. modeled as a primitive shape or a triangle mesh). Its observation consists of a set of 3D points (point cloud) measured on its visible surface, defined in the observer's coordinates (e.g. in the image frame of a stereo camera system). A particle X is a hypothesis on the object pose i.e., a 6D vector (position + orientation). A measurement d is a point of the cloud. Our goal is to estimate the probability of a given hypothesis for a set of m measurements $(d_1, d_2, \ldots, d_m)$. From the Bayes rule, it can be written as:

$$p(X|d_1, \ldots, d_m) = \frac{p(d_1, \ldots, d_m|X)p(X)}{p(d_1, \ldots, d_m)}. \tag{1}$$

The different measurements are obviously not independent. However, before introducing the object shape and the measurement model, it is not possible to model this dependency so a common assumption is to consider the measurements as independent, like in [29]. Experiments further in the paper have proven this assumption practically holds. This allows simplifying Eq. (1) to:

$$p(X|d_1, \ldots, d_m) = \frac{\left(\prod_{i=1}^{m} p(d_i|X)\right) p(X)}{\prod_{i=1}^{m} p(d_i)}. \tag{2}$$

Without additional knowledge on $p(X)$, we assume it to be uniform in a finite region around the robot and as the $p(d_i)$ are scaling factors only depending on the measurements they can be ignored so:

$$p(X|d_1, \ldots, d_m) \propto \prod_{i=1}^{m} p(d_i|X). \tag{3}$$

To compute the probability or weight of a particle, the computation of the likelihood of a measurement $p(d_i|X)$ is needed. This likelihood computation requires the choice of a measurement model. We use a proximity model [28], frequently used in mobile robot mapping, with an additional constraint of visibility of the observed point:

$$p(d_i|X) = \frac{1}{(2\pi)^{3/2}|C|^{1/2}} e^{-\frac{1}{2}(d_i - x_{0_i})^T C^{-1}(d_i - x_{0_i})}, \tag{4}$$

where x_{0_i} is the visible point of the model that is the closest to d_i (Euclidean distance), under the hypothesis X, C is the covariance matrix of the sensor noise and $|C|$ its determinant. The visibility computation is based on the orientation of the surface at the considered point i.e., the surface normal at the point must be opposite to the viewer direction. If the object is convex it is an exact visibility test otherwise it is an approximation. Combining Eq. (3) with Eq. (4) gives:

$$p(X|d_1, \ldots, d_m) \propto e^{-\frac{1}{2}\sum_{i=1}^{m}((d_i - x_{0_i})^T C^{-1}(d_i - x_{0_i}))}. \tag{5}$$

For a set of measurements, the term in the exponential in Eq. (5), i.e.:

$$\sum_{i=1}^{m}((d_i - x_{0_i})^T C^{-1}(d_i - x_{0_i})) \tag{6}$$

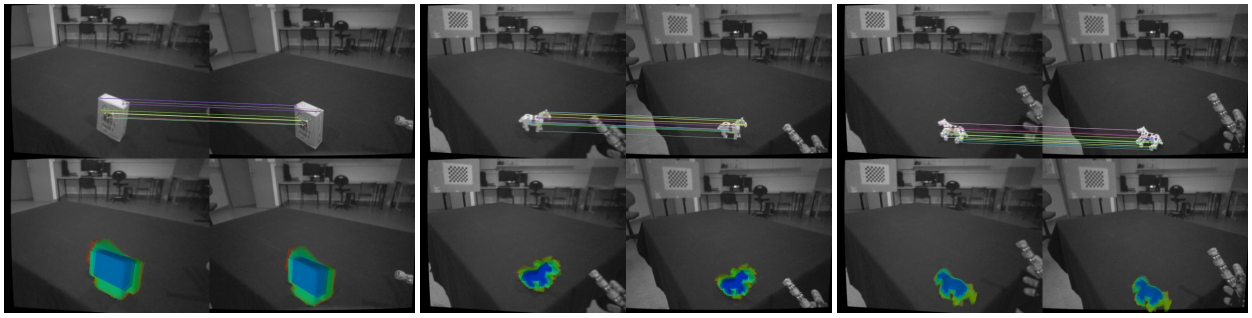
will be referred as the *data potential*. Its value tells how well the hypothesis X fits the measured data. The smallest the potential, the best the hypothesis fits.

3.2. Filter

The general principle of the particle filter applied to the object's pose estimation problem is outlined in Algorithm 1. N and M are parameters of the filter. Many variants of this basic principle exist, which differ mostly in the sampling process. Among the existing variants of the particle filter, we chose the *Scaling Series Algorithm* (SSA) by Petrovskaya and Khatib [29] for two reasons. The first one is that it is an annealed filter i.e., it uses a temperature term T in the measurement model (here in (4)) that flattens the Gaussian model of a measurement by dividing the data potential (6) by T. This reduces the risk of missing a maximum of the distribution because it is too narrow for the sampling resolution. This is very important as we work in a high dimension space that cannot be densely covered with a particle set of reasonable size. The second motivation is the even sampling of the particles in the areas of high-probability particles. As we are interested in the probability distribution of the object pose and not in a single estimation, it is relevant to have a more uniform sampling. The SSA is briefly outlined in Algorithm 2 in Appendix A.

Data: A set of points measured on the object surface $D = (d_1, \ldots, d_m)$;
A model of the object's shape ;
A coarse object's pose estimation X_0 ;
Result: S: a weighted particle set (i.e., pose hypotheses)
Initialization: $S \leftarrow$ *a particle set sampled around* X_0;
for $k = 1$ **to** N **do**
  foreach $X_i \in S$ **do**
    compute the weight $w_i = p(X_i|D)$ *(from Eq. 5)*
  end
  $S \leftarrow$ *keep M best candidate particles*;
  $S \leftarrow$ *re-sample in the neighborhood of S elements*;
  Normalize the particle weights $w_{i, i \in [\![1;M]\!]}$;
end

Algorithm 1: Sketch of the basic particle filter algorithm, applied to the pose estimation problem.



(a) Box: The position estimation is good in the direction of its largest face and the orientation estimations very poor around the normal of this face.

(b) Lego horse: The model is a mesh with 152 triangles.

(c) Lego horse: Its small size makes it difficult to localize but the particle set shows there is a plane to favor to grasp the object.

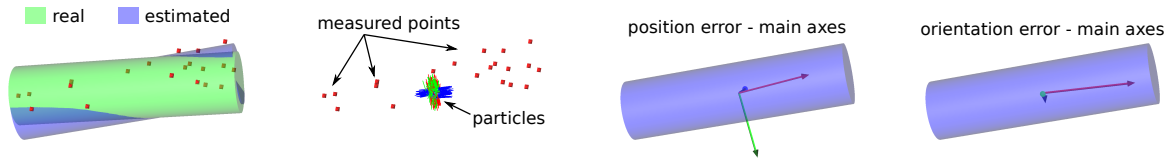
Figure 3: Some examples of estimated pose distributions for two different shapes. They show how the positions of the measurements affect the estimation.

As the point cloud may contain several outliers, a median filter is added to remove such points: for each X, only the 80% of the measurements with the highest likelihood are kept. Details concerning the parameters selected to run the algorithm are given in Appendix A. Figure 3 shows different estimations for different objects. The computation time depends both on the shape complexity (i.e., the number of triangles) and on the number of measurements. For the shown examples it goes from 2s to 10s. As can be seen on Fig. 3, the shape and orientation of the particle set depends on several factors such as the object's shape, its orientation with respect to the observer and the distribution of the measured points on its surface.

3.3. Particle Set

The particle set gives information about the knowledge of the object pose. Next section details its use to select a grasp. Before choosing the grasp, an important point is to decide if the accuracy of the pose estimation is enough to try and grasp the object or if the robot should get closer to the object and change its point of view. Knowing the data potential (Eq. (6)) is not enough as it depends on the number of measurements. For instance, the extreme case of a unique measurement could easily give a zero potential i.e., a perfect match, even though it leaves many uncertainties about the object's pose. A high potential means a bad match but a low potential can be the consequence of few measurements. Therefore the shape of the particle set must be considered. Intuitively, the more it spreads, the less knowledge we have on the object pose as it means that more hypotheses will be consistent with the point cloud[1]. Some indeterminacy however is due to the object symmetries. These ambiguities can be removed via clustering of the particles. Top images of Fig. 4 shows the pose estimation of a cylinder that has two clusters. The clusters are extracted with a neighborhood-graph algorithm and if they are consistent with the object symmetries (in term of numbers and relative transformations) only the most populated one is kept. The last step is to characterize the dispersion of the particles. A principle component analysis (PCA) is performed both on positions and on orientations of the particles. Special care has been taken to analyze the particle orientations (represented by quaternions). A method to find the mean of the quaternion set is given in [19]. The maximum variances of positions and orientations are used as a metric to characterize the uncertainty on the pose. For the orientation, a comparison must also be performed with the direction of possible symmetry axes. Bottom images of Fig. 4 show the result of the two PCAs. The plane symmetry of the cylinder makes the particles have their z-axis aligned with the cylinder axis, with two possible signs, while the plane symmetry makes the particles have all the possible orientations around the z-axis (blue axis in Fig. 4).

[1]Note that the particles spread depends on the number of iterations of the algorithm. This number should then be the same to compare the result of different estimations.



(a) Real (green) and estimated poses (blue)

(b) Particles are represented as frames with red, green and blue segments for the x, y and z axes respectively.

(c) The axes of the principle component analysis for the positions. The axes lengths are scaled to the standard deviation along each axis.

(d) The axes of the principle component analysis for the orientations. The axes lengths are scaled to the standard deviation along each axis.

Figure 4: Results of the pose estimation for a synthetic example: 20 measurements generated randomly on the visible surface of the cylinder with a Gaussian noised added to each measurement.

Consequently two clusters are extracted. The figure also shows the axes of the principle component analysis for position and orientation. The axis of highest variation of the orientation corresponds to the cylinder symmetry axis and should then be discarded as it is intrinsic to the shape and not due to measurement error.

The computed variances can then be used as a measure of the quality of estimation. For instance, the robot could use this measure to decide if it has enough knowledge to go and grasp the object or if it needs to get closer to the object to perform another observation.

How to threshold this value to decide if the robot has enough knowledge to go and grasp the object depends on object shape and hand kinematics. The value should be optimally determined from various and repeated experiments, for instance via a machine learning algorithm. For our robot (see Section 5), with the algorithm parameters of Appendix A, we found empirically that a standard deviation along the axis of highest variance that is about $1cm$ is good enough. The orientation variance is more object-dependent. For some shapes, like a cylinder or the box of Fig. 2, a high uncertainty around one axis is not a difficulty as long as the correct grasp is chosen (grasp from the top in the example of Fig. 2), so the second axis of highest variance is more important. While a standard deviation greater than $20°$ is acceptable for the main axis, we found that an acceptable value for the second axis is less than $5°$.

4. Grasping Strategy

In the previous section we described how to get and represent the object's pose estimation. The current section explains how to exploit the estimation to compute a grasp. The problem of computing the reaching motion will also be considered. We will assume that the grasping action is performed in two steps: the robot moves its open hand (corresponding to a given finger configuration $q_{open_fingers}$) to a target pose T_{grasp} and then closes its fingers on the object. The final grasp (i.e., how the fingers touch the object) will depend on the real object pose with respect to T_{grasp} and the motion law used to close the fingers (which depends in particular on $q_{open_fingers}$). Our goal is then to compute the hand pose T_{grasp} that will provide the highest robustness with respect to uncertainty on the object pose.

4.1. Choice of a grasp

To compute a grasp, we first build a set of various grasp poses (i.e., a set of hand poses expressed in the world coordinates). This set depends only on the object shape and the hand kinematics. Its computation is detailed below (Section 4.1.1). The set is centered on the object most likely pose according to the estimation of Section 3. The hand poses that are not feasible because the hand and the environment collide (e.g. collision with the table) or are not reachable for the robot arm are discarded. Then, the object pose distribution is used to rank the remaining hand poses according to a robustness function. The latter is based on the pose estimation and evaluated in simulation, as described in Section 4.1.2. Figure 5 illustrates the overall decomposition of the grasp computation and in particular at which step the particle set is introduced in the computation pipeline.

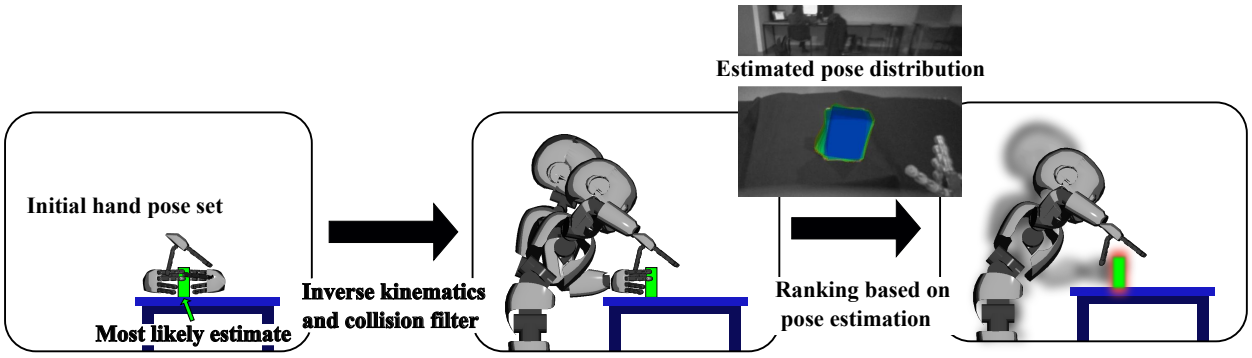


Figure 5: From a set of hand pose uniformly distributed and centered on the object's most likely configuration, the grasps that are colliding or not reachable for the robot are discarded. Then, the object pose distribution is used to rank the grasps according to their likelihood of success for the knowledge of the object pose.

Figure 6: The set of all the grasping directions computed for a box-shaped object. Other poses are obtained by adding small offsets in the main axis directions , for a total of 144 grasps . The fingers are represented closed on the object as they are when the feasibility of a grasp is evaluated.

4.1.1. Sampling the hand poses

The function we use later (Eq. (7)) to characterize the robustness of a grasp cannot be easily interpreted in order to directly find good values of T_{grasp}. Therefore a roughly uniformly-distributed set of T_{grasp} is preferable. The set is obtained by aligning the hand approach direction with the main directions of the object and adding different orientations and offsets to the hand, similarly to Miller et al. [25]. Figure 6 shows some of the grasps of a box computed with our sampling parameters. The other grasps, not shown on the figure, are the same but with small offsets along the box main axes. The overall set consists of 144 grasps. It is then centered on the object's most likely pose.

Once the set of T_{grasp} is computed, the inverse kinematics is applied to remove the unreachable poses, with a closed-loop inverse kinematics algorithm [8]. Then, for each of the computed T_{grasp} and their corresponding robot configuration, only the collision-free ones are kept and their robustness score is computed as explained in next section.

4.1.2. Evaluation of a hand pose

As we assume a discrete probability distribution of the possible object poses (n elements), we can write the probability of success S of a grasp initiated by the hand pose T_{grasp}, by marginalizing $p(S|T_{grasp}, X)$ over the object pose hypotheses:

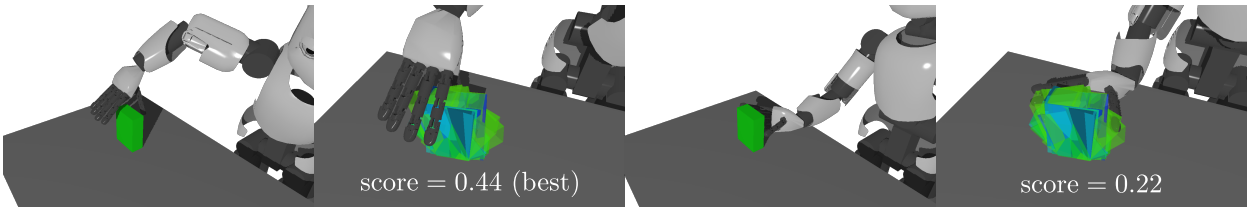


Figure 7: Two grasp configurations (far and close views for each configuration). Their scores according to equation (7) are 0.44 (grasp from the top) and 0.22 (grasp from the front). The grasp from the top is the best among all the computed grasps. Note how it is more robust to slightly shift the hand on the side as the pose hypotheses are not symmetrically distributed. The grasp from the front is not the one with the lowest robustness as some grasps have a null score (e.g. when the hand is too far from all the pose hypotheses or collide the object in all its possible poses).

$$p(S|T_{grasp}) = \frac{1}{n}\sum_{i=1}^{n} p(X_i|d_1,\ldots,d_m)p(S|T_{grasp},X_i) \tag{7}$$

where $p(X_i|d_1,\ldots,d_m)$ is computed by Eq. (5)[2]. To compute Eq. (7), we need $p(S|T_{grasp},X_i)$ i.e., a model to compute the probability that a grasp with hand pose T_{grasp} and the object placed exactly at X_i succeeds. This value is evaluated in simulation. We assume a binary distribution for $p(S|T_{grasp},X_i)$:

$$p(S|T_{grasp},X_i) = \begin{cases} 0, & \text{invalid grasp} \\ 1, & \text{valid grasp} \end{cases} \tag{8}$$

The computation is performed through a common algorithm that closes the fingers (in simulation) until contact with the object or another finger. The closing motion path is the linear interpolation between $q_{open_fingers}$ and a closed-hand configuration. The two configurations are given. The contact points are computed once all the fingers are blocked. This gives configurations like the ones in Fig. 6. Three tests must be passed for a grasp to be considered as successful: there must be no inter-penetration between the different bodies of the hand and the object; the grasp must be stable (force-closure criterion); the fingers must contact the object at similar times (i.e., close values of the interpolation parameter, in our test < 0.05). The last test is not necessary for a robot with tactile or force sensing as it can theoretically ensure a finger will not move the object before the latter is fully grasped.

Figure 7 illustrates an example of the robustness score computation for a box-shaped object. Only grasps from the top and the front of the box are reachable, so the set of hand poses only contains grasps with these two approach directions, with slightly different positions though. The particular particle distribution shown in the figure makes it more robust to grasp from the top that from the front.

4.2. Reaching motion

Once the best grasp is selected according to Eq. (7), a collision-free path is planned. All the care given to choose a grasp that is robust against object pose uncertainty must not be ruined by a reaching motion that goes inconsiderately close to the object. Therefore the path must be optimized so as to avoid configurations that are close to possible poses of the object. In addition, a path should also be far from the environment obstacles. So we introduce a cost over the robot configuration space, by marginalizing over the obstacles poses, considering the object to grasp as an obstacle:

$$cost(q) = \max_{obstacle} \int_{X_{obst}} \frac{p(X_{obst})}{distance(robot(q), obstacle(X_{obst}))} dX_{obst} \tag{9}$$

where q is a robot configuration, X_{obst} is a pose hypothesis for the obstacle $obst$ and $distance(robot(q), obstacle(X_{obst}))$ is the distance between the robot in configuration q and the obstacle $obst$ at pose X_{obst}.

[2]The proportionality relation is replaced with an equality as we are only interested in ranking grasp configurations.

Figure 8: The sample objects used in our study, representatives of particular shapes: a ball (sphere), a glue stick (cylinder), a cardboard box (rectangular prism) and a textured Lego horse (complex object).

We assume the pose of all the obstacles but the object to grasp as being perfectly known. Furthermore the pose distribution of the object to grasp is discrete. We can then rewrite Eq. (9) as:

$$cost(q) = \max\left(\max_{obstacle \setminus object}\left(\frac{1}{distance(robot(q), obstacle)}\right), \sum_{i=1}^{n} \frac{p(X_i)}{distance(robot(q), object(X_i))}\right) \tag{10}$$

where $X_{i \in [\![1;n]\!]}$ is a possible pose of the object and $p(X_i)$ the corresponding probability. State-of-the-art motion planning algorithms bias the exploration towards configurations rated as low cost by a cost function defined in the configuration space. T-RRT [17], for example, has the advantage of an easy implementation and offers a good trade-off between convergence speed and quality of the solution. In particular, the balance can be easily tuned through one parameter only (called $nFail_{max}$) with the version of the algorithm in [16] ("improved temperature tuning" section) that explains how to automatically tune the other parameters. We used (10) as a cost function and set the $nFail_{max}$ parameter to 150.

As the goal configuration of the planning problem is a very high cost configuration because the hand is very close to the object, a standard single-tree exploration would not converge in a reasonable time. We consequently use a bidirectional version with the second tree starting from the goal configuration.

5. Experimental Results

The goal of this section is to show how the proposed method behave with some examples of typical grasping tasks (Section 5.3) and to give an experimental demonstration of its interest compared to a method that would use a single estimate to compute the grasping configuration (Section 5.2).

The experiments were conducted with the humanoid robot iCub [26] in its first version, without any modifications of the platform. Our version is not equipped with any tactile sensor in the hands, as visible in Figure 9. The hand exhibits a certain compliance that allows the robot to hold the objects without any force control. However, the size of the fingers is such that iCub can grasp relatively small objects only. Figure 8 shows the dimensions of the objects used throughout the paper. They all have relatively small size with respect to the robot's hand. Objects were observed by the robot's cameras embedded in the head, as shown in Figure 1; we did not use any external –to the robot– sensing device such as a Kinect. To increase the reaching workspace of the hand, we used the seven degrees of freedom of the arm and the three of the torso.

In the experiments, a table is placed in front of the robot with the object to grasp on the tabletop, placed at a reachable position. The table pose is previously computed by placing a known checker pattern on its top and it does not change during the experiment. Next section gives some technical details concerning the vision processing to compute the point cloud from the stereo images.

5.1. Vision Setup and Point Cloud Computation

We used the pair of stereo cameras mounted on iCub's head (i.e., its eyes). Image resolution is 640×480 for each camera. Stereo calibration of the cameras was performed with functions of the OpenCV library[3]. As we do not use a support for stereo-vision like a projected pattern, the easiest way to segment the object from the table was to assume a table with a uniform color. Here, object extraction is done through the Grabcut algorithm [31], by specifying

[3] www.opencv.org

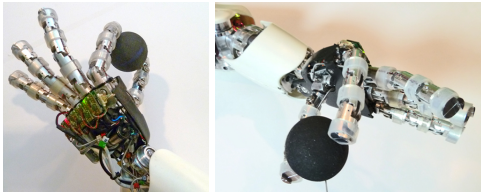
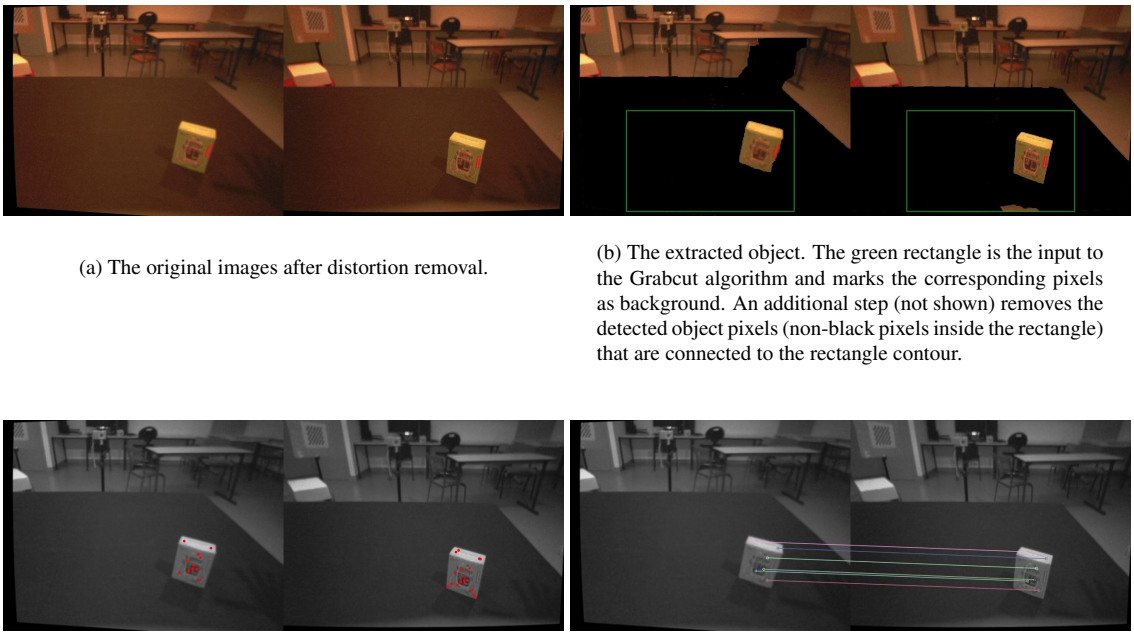


Figure 9: The hands of iCub. The two fingers grasp a small rubber ball (4*cm* diameter).



(a) The original images after distortion removal.

(b) The extracted object. The green rectangle is the input to the Grabcut algorithm and marks the corresponding pixels as background. An additional step (not shown) removes the detected object pixels (non-black pixels inside the rectangle) that are connected to the rectangle contour.

(c) The extracted 2D features.

(d) The matched features.

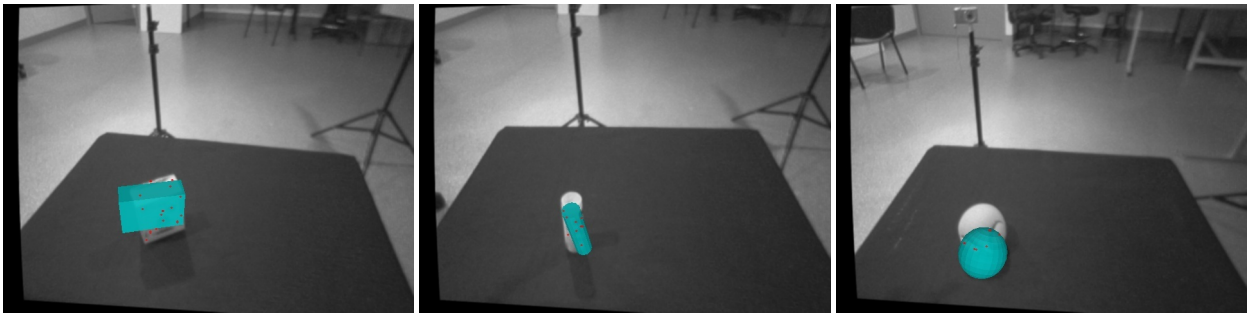
Figure 10: The different steps to obtain the point cloud of the object to grasp.

a rectangle in the images that corresponds roughly to the tabletop. 2D features (SURF [2]) are then extracted in the segmented images. For each feature point p_{feat} on the left image, we match the feature on the right image that is the closer (Euclidean distance) among the features that are less that 1.5 pixels away from the epipolar line of p_{feat}. Matched points are then triangulated. The obtained cloud is in the camera frame. The cloud points are then converted to the robot's reference frame from the knowledge of the eye pose (as provided by the robot's encoders) and the eye-camera transform (obtained from the hand-eye calibration performed with the method by Daniilidis [10]). The covariance matrix C used in Eq. (4) is the diagonal matrix:

$$C = 0.005\ I_3 \tag{11}$$

C corresponds to the noise variance of the overall triangulation process (here empirically set to 5mm). Standard error propagation technique (linearization) applied to stereo vision triangulation alone shows the diagonality of C [35]. We could not estimate experimentally the additional noises (calibration, encoders) so, according to the null hypothesis, we kept C diagonal. Figure 10 shows the different operations to obtain the point cloud of the object.

Once the point cloud is computed, the particle filter is applied to estimate the object pose distribution. The tuning parameters of the particle filter are given in Appendix A.

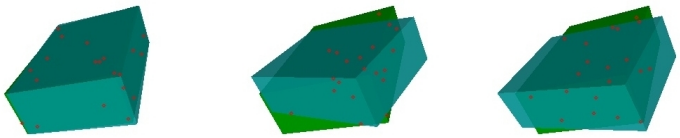


(a) reference model: 902 points, measured: 20 points, computation time: 20s

(b) reference model: 2729 points, measured: 8 points, computation time: 20s

(c) reference model: 1359 points, measured: 9 points, computation time: 5s

Figure 11: Pose estimation results obtained with the ICP algorithm. The number of points of the reference cloud is given for each shape as well as the typical computation time and the number of measured points.



(a) (b) (c)

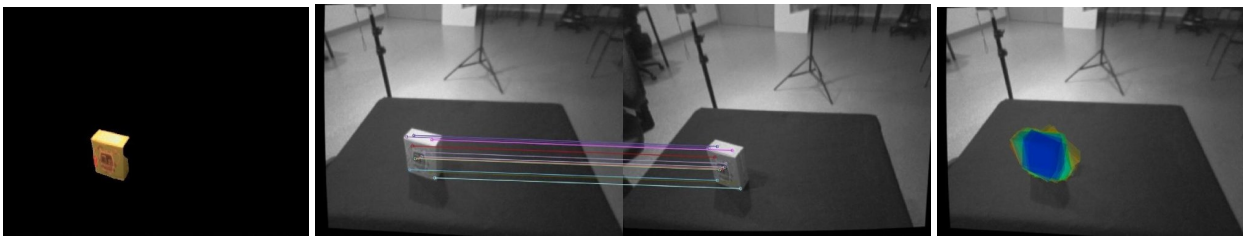
Figure 12: Pose estimation results obtained with the ICP algorithm for a synthetic example with 20 perfectly measured points. The reference cloud has 902 points uniformly distributed over the box surface. Computation times are about 15 seconds.

5.2. Comparison with ICP algorithm

To show the interest of the proposed method, we compare our approach and the Iterative Closest Point algorihtm (ICP) [5] which is commonly used to estimate the pose of an observed point cloud from a model cloud. The same data is used as in the rest of the experiments presented in the paper. The only difference is the object model that has to be a point cloud. Consequently, all the object models were converted into point clouds. All the shapes are first triangulated, then each triangle surface is discretized according to a grid pattern. The implementation of the ICP algorithm is the one of the Point Cloud Library [32]. As shown in Fig. 11, the small number of measured points leads to poor results. Also not taking into account the visibility constraint gives more ambiguities on the object's pose like it can be seen with the ball example of Fig. 11 where the points are found to be on the back (non-visible side) of the ball.

We also considered the case of perfect measurements simulated with synthetic data: n points are chosen randomly at the surface of the perfect model (mesh or primitive shape) in a pose randomly chosen. Depending on n and on the distribution of the points over the object's surface, the ICP algorithm can succeed or fail to find a good pose estimation. For instance, with the box shape and n=20 (typical value in our experiments), we obtained the results of Fig. 12. As can be seen, the computed pose can be very different from the real one. This means that 20 points are not enough to reliably estimate the pose of the object. We found that for this shape, in the perfect-measurement simulation, 40 points are a minimum to obtain poses that are close to the real one.

The above results confirm that the measured data we retrieved with our cameras are not rich enough to lead to a unique pose estimate. Consequently, this result validates our choice of reasoning with a distribution.



(a) The object segmented with the Grabcut algorithm (left image). Note that a part of the object was missed in left image segmentation.

(b) The matched 2D features (13 points).

(c) The computed pose hypothesis particles projected on left image. Note that the most likely pose (the blue one) does not fit perfectly the real object pose. This shows the interest of reasoning with a distribution instead of a single best estimate.

Figure 13: The perception chain of the box example.

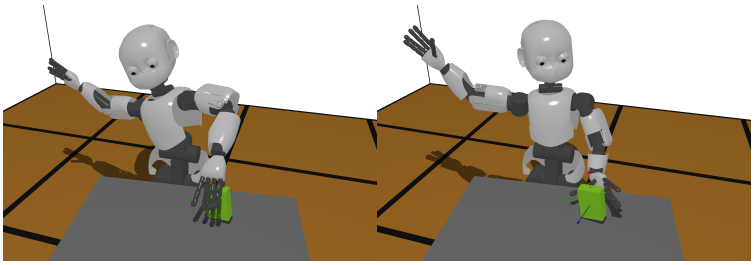


Figure 14: Two of the grasping configurations computed by the grasp planner for the box object. The left one is found to be the best by our algorithm. More configurations are actually found corresponding to different inverse kinematics solutions or slightly different hand poses but are not represented here.

5.3. Grasping task examples

The goal of the first grasping task experiment is to study the influence of the object shape on the computed grasp. Three objects with different shapes are used in this study: a box, a cylinder and a ball. The choice of each shape is motivated below.

5.3.1. Grasping a box

The box shape is particularly interesting because, from a purely geometric and mechanical point of view, grasping the box from its different sides leads to identical grasps[4]. Figure 13 shows the segmented objects and the triangulated points. Most of the triangulated points are on the front face of the box leading to a good knowledge of the face depth whereas its position in the other directions as well as its orientation around the depth axis is poorly known. This is reflected by the distribution of the particle cloud at the bottom of Fig. 13. Figure 14 shows two of the grasp configurations that were computed by the planner. The first one, with a grasp from the top, is chosen as the best one according to Eq. (7). Figure 15 illustrates the motion planned and executed to reach the selected grasp. The last configuration was obtained by lifting the robot arm to check the stability of the grasp.

[4]Considering the light weight of the object, the varying weight direction has no impact on the grasp stability.

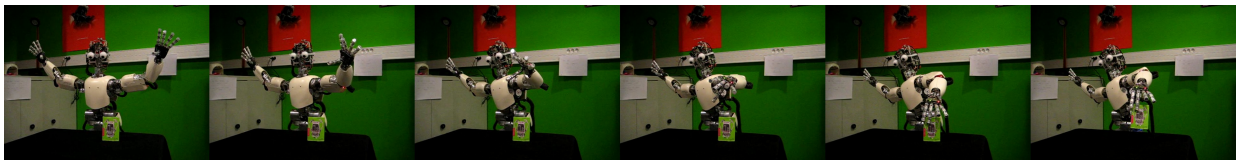
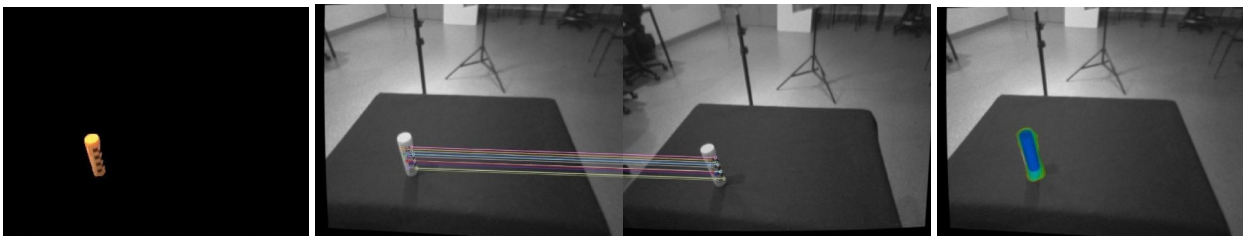


Figure 15: Some snapshots of the motions executed by the robot to grasp an object, by choosing the best ranked grasping configuration. The last snapshot of the sequence shows the robot lifting the object, to check the grasp stability.



(a) The segmented object in left image.

(b) The matched 2D features (18 points).

(c) The computed pose hypothesis particles projected on left image.

Figure 16: The perception chain of the cylinder example.

5.3.2. Grasping a cylinder

The second object is a cylinder. In order to have enough 2D features to match, a patterned sheet was pasted on a side of the object (a checker). The conditions are similar to the first experiment in the sense that the object is seen mainly from the front (Fig. 16). However the front surface of the object contains more information about its orientation (as it is no more flat i.e., no more invariant in rotation around the normal vector). Consequently, the variance of the particles orientation is smaller. Here the variance mainly concerns the cylinder vertical position. The chosen grasp is thus a grasp from the side of the object (Fig. 17). We see that in conditions similar to the previous test with the box, the chosen grasping direction is orthogonal to the one chosen to grasp the box, because of the different front surface. It must be noted that we assume no uncertainty on the table pose, which makes possible to choose a grasp with the hand close to the table. With noise on the table position, the chosen grasping configuration might not be the best choice.

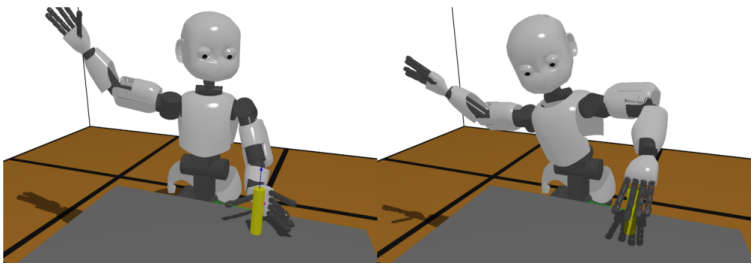
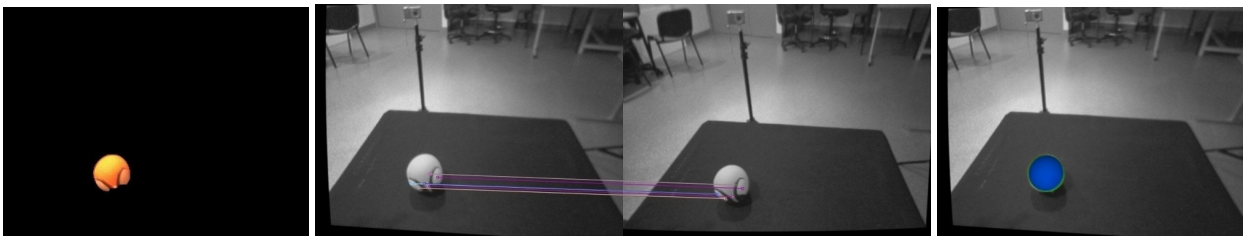


Figure 17: Two of the grasping configurations computed by the grasp planner for the cylinder object. The left one is found to be the best by our algorithm. More configurations are actually found corresponding to different inverse kinematics solutions or slightly different hand poses but are not represented here.



(a) The segmented object (left image).

(b) The matched 2D features (9 points).

(c) The computed pose hypothesis particles projected on left image.

Figure 18: The perception chain of the ball example.

5.3.3. Grasping a ball

The ball is interesting because its shape requires fewer points to be localized. Only three noise-free points are needed. However, as the measurements are noisy, we expect more points to be required for a good estimation. From the experimental results (Fig. 18), we see that there is no anisotropy in the estimation. The difference in the variances along the different axes of highest variance (from the principal component analysis of Section 3.3) does not allow to conclude that there is any direction of major error. Considering the fact that the triangulation error is expectedly higher in the depth direction, we also tried a higher value (twice the initial value) of the noise variance in the camera depth direction in matrix (Eq. (11)). However, it did not lead to a significant change in the estimation. Consequently, for a sphere-shaped object with this number of triangulated points (9 points), all the grasps are equivalent with respect to our robustness criterion.

5.4. Changing object placement

In this experiment, we conducted 8 tests with the box-shaped object. For each test, the box was placed at a different position and orientation. The placements were chosen so as to let different possibilities to grasp the object. The robot had to be able to grasp at least two different sets of opposite faces of the box e.g. from the top and from one side of the box. This constraint was added to ensure that the robot kinematics does not restrict the reachable grasps to one small set of very similar grasps. Figure 19 shows the images of the left camera, for each of the test, with the reprojected pose estimation particles. Some poses looks similar at a first glance in Fig. 19, but the object has actually different orientations making visible different sides of the box that have different patterns (i.e., different features to match). This modifies the perception of the object.

Figure 20 shows the best grasps according to the robustness function of Section 4.1.2. Notably, there is not a unique best grasping direction (e.g. from the top or side) for all the object poses. The particle set is always roughly in the plane of the front face of the box, from the camera perspective, but it extends more or less in the directions orthogonal to the plane's normal. The spread of the particles depends on the number and distribution of the triangulated points and cannot be known in advance.

5.5. Discussion

This study has shown that a possible successful grasping direction can be computed from a noisy point cloud even if the cloud cannot provide a precise estimation of the object pose. Interestingly our study was performed with low resolution stereo cameras.

Over the dozens of experiments that were conducted we found that about 20% of the grasping tasks were successful. This success rate is rather low, but still better than what was obtained by computing the grasp through a single-estimate method like the ICP algorithm. In that case the success rate falls to 0 (see the pose estimation results in Fig. 11). Our results can be improved, especially with a better robotics setup. Among the several possible causes, we think the small success rate is mostly due to the inaccuracy of iCub arm and hands (tendon-driven motion), the

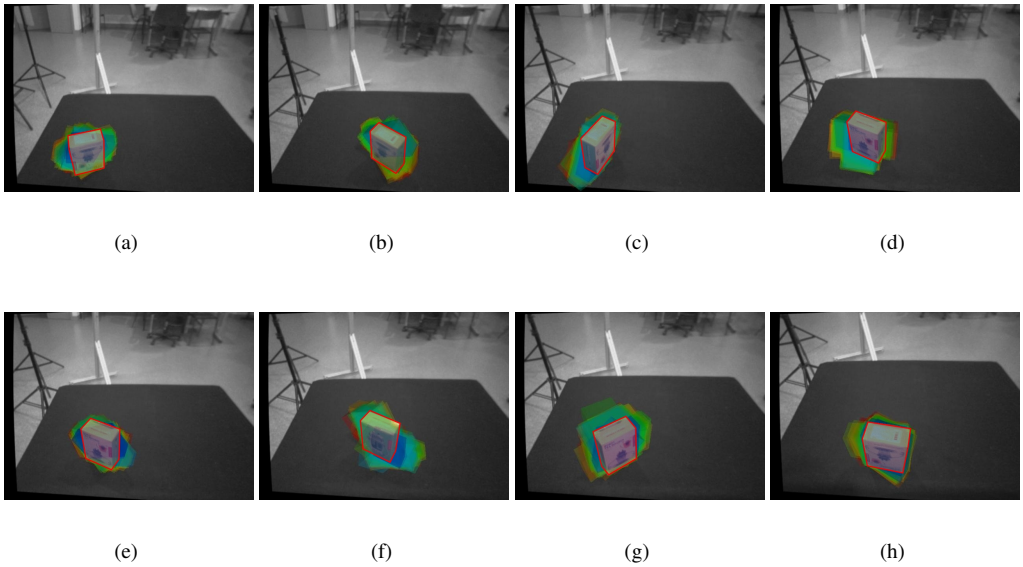


Figure 19: The left image of each test. The object contour is highlighted in red. The particle set of the pose estimation is reprojected over the image with the same color convention as in the rest of the paper (red= low likelihood, blue= high likelihood).

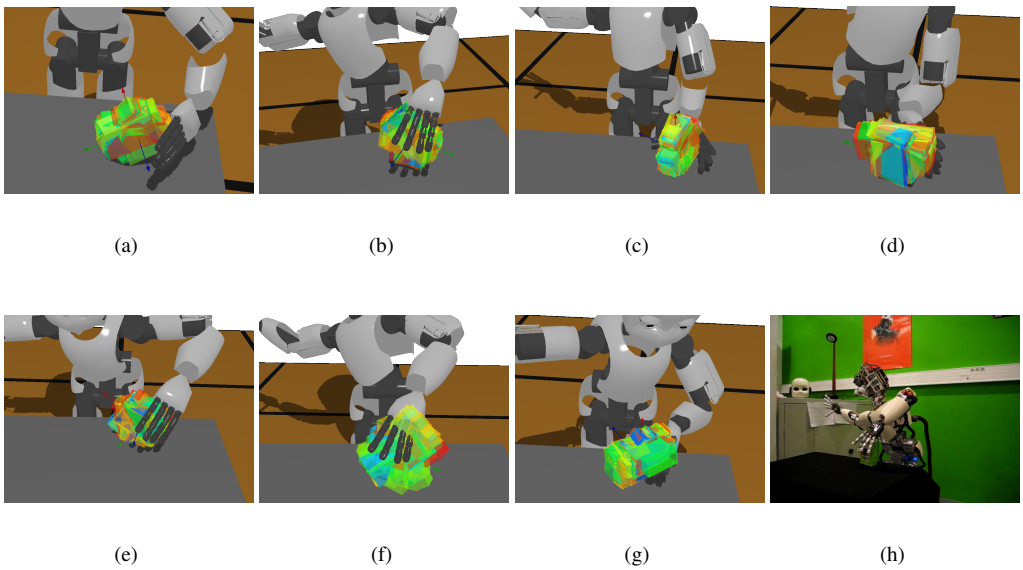


Figure 20: For each of the poses depicted in Fig. 19 the grasp that was ranked as the most likely to succeed. The figures numbering is the same as the one in Fig. 19. The particle sets of the pose estimations are also depicted. The last picture shows the grasp as executed on the real robot in the example of Fig. 5.4.

accumulated kinematic errors, especially in the eyes, that affect the computation of the camera/eyes frame in the robot base frame.

Several aspects of the experimental protocol could be modified in order to adapt to more powerful sensing devices. For instance, the robustness of the grasping tasks sometimes suffered from the small number of points that could be matched in the stereo images. There are several reasons for the possibly small number of points to match: the uniformity of the object's texture, the small size of the object or the cameras' low resolution. Range scanning device like lidars or camera systems with pattern projection (like Microsoft's Kinect or PR2's stereo cameras) would provide a point cloud with higher density. Object and table segmentation could then be done more robustly with the use of combined plane extraction and point clustering [18], which was not possible with our setup that was limited to a single object on the table. It would then be possible to work with a cluttered environment, i.e. many objects on the table.

Grasps could also be improved reactively through the use of tactile sensors [12]. For example, because of the intrinsic imprecision of tendon-actuated fingers, in some situations the robot fails to correctly position the fingertips on the object. Just stopping the motion once a contact is detected would solve this problem. Tactile sensors would also allow the synchronization of the contact times. Having a finger that touches the object before the others can indeed make the object pivot around its position and potentially fall.

The computational cost of the proposed method is tolerable, especially if compared to the ICP algorithm. We executed the experiments on a single computer machine (CPU i5 – 2.1GHz, 4GB RAM). The computational cost of our method is between 2 and 10s for the pose estimation (computation of the pose probability distribution, Section 3.2) plus less than 20s for the remainder of the algorithm (computation of the grasp and planning of reaching motion). As a comparison just estimating the (single) pose with the ICP algorithm takes 20s (Section 5.2).

6. Conclusion

We have presented a method that allows a robot to successfully grasp an object, even when the range perception of its pose leaves many unknown. It is based on the estimation of the probability distribution of the object pose as a set of discrete particles computed through an annealed particle filter. The shape of the particle set is exploited to select the grasps that are the more likely to succeed considering the possible poses. Experimental tests on objects with various shapes confirmed that taking into account the pose error dramatically changes the direction of the best grasp to choose. The proposed approach was applied on the robot iCub with a point cloud obtained through stereo matching of 2D features. Even with a small number of points the algorithm can compute a successful grasp. For range scanning devices that provide a higher density cloud, the method would allow to grasp very small objects. As the proposed approach only considers the grasping task before the contacts, it could easily complete some existing methods for robust grasping via tactile exploration.

Acknowledgements

This work was partly supported by the European Commission within the CoDyCo project (FP7-ICT-2011-9, No. 600716).

Appendix A. Particle Filter Parameters

Some parameters must be selected in order to run the particle filter algorithm used in Section 3. The different terms below refer to the Scaling Series algorithm described in [29] and briefly outlined in Algorithm 2. For more details on the algorithm and its parameters, we refer the interested reader to Petrovskaya and Khatib's paper [29].

- The maximum number of iterations N is 35
- The number of re-sampled particle in each neighborhood M is 15
- The initial neighborhood V_0 wherein to sample particles:

- Its center is taken as the barycenter of the point cloud for the translation part and the identity quaternion for the orientation part
- Its radius is $\sqrt{3} \times 0.1m$ for each translation component and π (radians) for the orientation component

- The terminal value of δ, δ_* is ($\sqrt{3} \times 1mm, 0.1°$)

Data: D: a set of measurements;
V_0: initial uncertainty volume ;
M: number of particles per δ-neighborhood ;
T_0: initial temperature parameter
Result: (X,W) a weighted particle set approximating the posterior
Algorithm initialization : δ_0 = *radius of* V_0;
for $n = 1$ **to** N **do**
  update δ_n;
  update T_n;
  $X_n \leftarrow$ *sample volume* V_{n-1} *with M evenly-distributed particles*;
  $W_n \leftarrow$ *compute normalized weights for each particle (depends on* T_n*)*;
  keep the best (highest weights) particles;
  compute the volume V_n*: union of the balls centered on the particles* X_n *with radius* δ_n;
end
compute the final particle set X and the weights W from (X_n, W_n);

Algorithm 2: Sketch of the Scaling Series algorithm [29].

References

[1] Bagnell, J. A., Cavalcanti, F., Cui, L., Galluzzo, T., Hebert, M., Kazemi, M., Klingensmith, M., Libby, J., Liu, T. Y., Pollard, N., Pivtoraiko, M., Valois, J.-S., Zhu, R., October 2012. An integrated system for autonomous robotics manipulation. In: IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 2955–2962.

[2] Bay, H., Tuytelaars, T., Gool, L. V., 2006. Surf: Speeded up robust features. In: European Conference on Computer Vision. pp. 404–417.

[3] Berenson, D., Srinivasa, S., Kuffner, J., oct. 2009. Addressing pose uncertainty in manipulation planning using task space regions. In: IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 1419–1425.

[4] Bergström, N., Bohg, J., Kragic, D., 2009. Integration of visual cues for robotic grasping. In: Fritz, M., Schiele, B., Piater, J. (Eds.), Computer Vision Systems. Vol. 5815 of Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 245–254.

[5] Besl, P., McKay, H., feb 1992. A method for registration of 3-d shapes. IEEE Transactions on Pattern Analysis and Machine Intelligence 14 (2), 239 –256.

[6] Bohg, J., Johnson-Roberson, M., Leon, B., Felip, J., Gratal, X., Bergstrom, N., Kragic, D., Morales, A., may 2011. Mind the gap - robotic grasping under incomplete observation. In: Robotics and Automation (ICRA), 2011 IEEE International Conference on. pp. 686–693.

[7] Brost, R., apr 1986. Automatic grasp planning in the presence of uncertainty. In: IEEE International Conference on Robotics and Automation. Vol. 3. pp. 1575–1581.

[8] Buss, S. R., 2004. Introduction to inverse kinematics with jacobian transpose, pseudoinverse and damped least squares methods. Tech. rep., IEEE Journal of Robotics and Automation.

[9] Corcoran, C., Platt, R., 2010. A measurement model for tracking hand-object state during dexterous manipulation. In: IEEE International Conference on Robotics and Automation. pp. 4302–4308.

[10] Daniilidis, K., 1998. Hand-eye calibration using dual quaternions. International Journal of Robotics Research 18, 286–298.

[11] Faria, D. R., Trindade, P., Lobo, J., Dias, J., 2014. Knowledge-based reasoning from human grasp demonstrations for robot grasp synthesis. Robotics and Autonomous Systems 62 (6), 794 – 817.
URL `http://www.sciencedirect.com/science/article/pii/S0921889014000347`

[12] Felip, J., Laaksonen, J., Morales, A., Kyrki, V., 2013. Manipulation primitives: A paradigm for abstraction and execution of grasping and manipulation tasks. Robotics and Autonomous Systems 61 (3), 283 – 296.

[13] Haralick, R. M., 1996. Propagating covariance in computer vision. International Journal of Pattern Recognition and Artificial Intelligence 10 (5), 561–572.

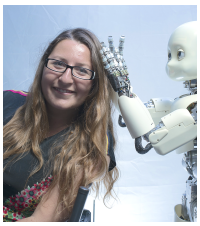
[14] Hsiao, K., Chitta, S., Ciocarlie, M., Jones, E., oct. 2010. Contact-reactive grasping of objects with partial shape information. In: IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 1228–1235.

[15] Hsiao, K., Kaelbling, L. P., Lozano-Perez, T., 2011. Robust grasping under object pose uncertainty. Autonomous Robots: Special Issue on RSS 2010.

[16] Iehl, R., Cortés, J., Siméon, T., july 2012. Costmap planning in high dimensional configuration spaces. In: IEEE/ASME International Conference on Advanced Intelligent Mechatronics. pp. 166 –172.

[17] Jaillet, L., Cortés, J., Simeéon, T., aug. 2010. Sampling-based path planning on configuration-space costmaps. IEEE Transactions on Robotics 26 (4), 635 –646.
[18] Jain, A., Kemp, C., 2010. El-e: an assistive mobile manipulator that autonomously fetches objects from flat surfaces. Autonomous Robots 28, 45–64.
[19] Johnson, M. P., 2003. Exploiting quaternions to support expressive interactive character motion. Ph.D. thesis, Section 6.2.
[20] Kehoe, B., Berenson, D., Goldberg, K., 2012. Toward cloud-based grasping with uncertainty in shape: Estimating lower bounds on achieving force closure with zero-slip push grasps. In: IEEE International Conference on Robotics and Automation.
[21] Krug, R., Dimitrov, D., Charusta, K., Iliev, B., oct. 2010. On the efficient computation of independent contact regions for force closure grasps. In: IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 586–591.
[22] Kyrki, V., Kragic, D., Christensen, H. I., 2006. Measurement errors in visual servoing. Robotics and Autonomous Systems 54 (10), 815–827.
[23] Laaksonen, J., Nikandrova, E., Kyrki, V., 2012. Probabilistic sensor-based grasping. In: IEEE/RSJ International Conference on Intelligent Robots and Systems.
[24] Lozano-Pérez, T., Mason, M. T., Taylor, R. H., 1984. Automatic synthesis of fine-motion strategies for robots. International Journal of Robotics Research 3 (1).
[25] Miller, A., Knoop, S., Christensen, H., Allen, P., sept. 2003. Automatic grasp planning using shape primitives. In: IEEE International Conference on Robotics and Automation. Vol. 2. pp. 1824 – 1829 vol.2.
[26] Natale, L., Nori, F., Metta, G., Fumagalli, M., Ivaldi, S., Pattacini, U., Randazzo, M., Schmitz, A., G. Sandini, G., 2012. Intrinsically motivated learning in natural and artificial systems. Springer-Verlag, Ch. The iCub platform: a tool for studying intrinsically motivated learning.
[27] Nguyen, V.-D., apr 1986. Constructing force-closure grasps. In: IEEE International Conference on Robotics and Automation. Vol. 3. pp. 1368–1373.
[28] Olson, C., feb 2000. Probabilistic self-localization for mobile robots. IEEE Transactions on Robotics and Automation 16 (1), 55 –66.
[29] Petrovskaya, A., Khatib, O., june 2011. Global localization of objects via touch. IEEE Transactions on Robotics 27 (3), 569 –585.
[30] Roa, M., Suàrez, R., may 2011. Influence of contact types and uncertainties in the computation of independent contact regions. In: IEEE International Conference on Robotics and Automation. pp. 3317–3323.
[31] Rother, C., Kolmogorov, V., Blake, A., 2004. Grabcut: Interactive foreground extraction using iterated graph cuts. ACM Transactions on graphics 23, 309–314.
[32] Rusu, R. B., Cousins, S., May 9-13 2011. 3D is here: Point Cloud Library (PCL). In: IEEE International Conference on Robotics and Automation (ICRA). Shanghai, China.
[33] Saxena, A., Driemeyer, J., Ng, A. Y., 2008. Robotic grasping of novel objects using vision. The International Journal of Robotics Research 27 (2), 157–173.
[34] Saxena, A., Wong, L. L. S., Ng, A. Y., 2008. Learning Grasp Strategies with Partial Shape Information. In: AAAI Conference on Artificial Intelligence.
[35] Sibley, G., 2003. Stereo observation models. Tech. rep., University of Southern California.
[36] Stulp, F., Theodorou, E., Buchli, J., Schaal, S., may 2011. Learning to grasp under uncertainty. In: IEEE International Conference on Robotics and Automation. pp. 5703–5708.
[37] Weisz, J., Allen, P. K., 2012. Pose error robust grasping from contact wrench space metrics. In: IEEE International Conference on Robotics and Automation. pp. 557–562.

Jean-Philippe Saut received the M.S. degree in robotics in 2003 from University Pierre et Marie Curie (UPMC), Paris, France. He received the Ph.D. degree in 2007 from UPMC. He continued his work in the field of robotic manipulation during one postdoctoral fellowship at the Laboratoire dAnalyse et dArchitecture des Systemes-Centre National de la Recherche Scientifique (LAAS-CNRS), France, and another at the Institut des Systemes Intelligents et de Robotique (ISIR) in UPMC. His research interests include dexterous manipulation, grasp planning, manipulation planning and assistive robotics.



Serena Ivaldi is a postdoctoral researcher in the Intelligent Autonomous Systems Lab (IAS) in TU Damrstadt, Germany. She received the B.S. and M.S. degree in Computer Engineering, both with highest honors, at the University of Genoa (Italy) and her PhD in Humanoid Technologies in 2011, jointly at the University of Genoa and the Italian Institute of Technology. There she also held a research fellowship in the Robotics, Brain and Cognitive Sciences Department. In 2011 she joined the Institut des Systemes Intelligents et de Robotique (ISIR) in University Pierre et Marie Curie, where she coordinated the experiments of MACSi, EDHHI and CODYCO Projects on iCub. In 2014 she joined the the Technical University of Damrstadt, Germany, to work in the Intelligent Autonomous Systems Lab on the CODYCO Project. Her research is centered on humanoid robots interacting physically with humans and environment.

Anis Sahbani is an Associate Professor at Pierre et Marie Curie University (Paris 6). He holds his research activities at the Intelligent Systems and Robotics Institute (ISIR-CNRS-UMR 7222). He has been the Principal Investigator for several research projects, and has been exploring different topics on robotics control, computer vision and, in particular, manipulation planning. He obtained an Electrical Engineering Degree from Ecole Nationale d'Ingenieurs de Tunis, in 1998, and MS in Automation and Computer Engineering, in 1999. He received his PhD in Robotics and Computer Science from the University Paul Sabatier - Robotics and Artificial Intelligence Group of the LAAS-CNRS in 2003.

Philippe Bidaud is the director of the Institute des Systemes Intelligents et de Robotique at UPMC. He received his Ph.D. degree in Mechanical Eng./Robotics from theUniversity of Poitiers (France) in 1984 and the habilitation for directing research from the University Paris6 in 1996 for his work on Design and Control of Complex Robotics Systems. For 15 years he was as a researcher in the section of Physical Sciences for Engineers at CNRS (Centre National de la RechercheScientifique), France. He became Full Professor at the University of Paris, Pierre et Marie Curie (UPMC) in 1998. Since 1981,his research interests are the field of the design of complex robotics systems, articulated hands, locomotion systems, assembly systems, haptic devices from the view point of modeling and analysis of mechanical systems and of their associated control. He also works in complex interaction control and modeling, control of smart materials, micro-robotics systems, design and optimization of compliant structures, and high mobility systems. In 199798, Philippe Bidaud was a Visiting Professor at the Field and Space Laboratory of the Department of Mechanical Engineering at MIT. He has published more than 150 papers including contributions to books, journals, and international conferences. He holds several patents and has been very active in technology transfer and valorization through the Centre de Robotique Integre Ile-de-France.