Improving Superquadric Modeling and Grasping with Prior on Object Shapes

Giulia Vezzani[1], Ugo Pattacini[2], Giulia Pasquale[2] and Lorenzo Natale[2]

***Abstract*— This paper proposes an object modeling and grasping pipeline for humanoid robots. This work improves our previous approach based on superquadric functions. In particular, we speed up and refine the modeling process by using prior information on the object shape provided by an object classifier. We use our previous method for the computation of grasping pose to obtain pose candidates for both the robot hands and, then, we automatically choose the best candidate for grasping the object according to a given quality index. The performance of our pipeline has been assessed on a real robotic system, the iCub humanoid robot. The robot can grasp 18 objects of the YCB and iCubWorld datasets considerably different in terms of shape and dimensions with a high success rate.**

I. INTRODUCTION

Grasping of unknown objects or whose pose is uncertain is still an open problem in robotics [1]. The missing or noisy information on object models and poses strongly affects manipulation performance.

In this work, we propose a modeling and grasping pipeline for humanoid robots equipped with two arms, a vision system supplying 3D information (stereo vision or RGBD camera) and tactile sensors on their fingertips. This pipeline is obtained by improving our previous approach based on *superquadric models* [2] and consists of the following steps. An object categorization system [3] provides prior information on the shape of the object to be grasped. Our modeling approach reconstructs a superquadric representing the object by combining the information provided by vision and the prior on object shape. The estimated model is used by our pose computation method to obtain pose candidates for the right and left hand. The best hand for grasping the object is automatically selected according to proper criteria that are summarized in a pose quality index. Once the selected hand reaches the desired pose, grasp stabilization is achieved via tactile feedback [4] so that the robot can robustly lift the object.

The first contribution of this paper is a general improvement of our previous work [2], in terms of reliability and computation time. This is achieved by using prior information on the object shape. We classify the objects to be grasped according to their shape with a visual object categorization stage trained on the fly and integrated in our pipeline. This is achieved by using the recognition system proposed in [3]. Our second contribution is the definition of a pose quality index that, given a pose candidate for the right and the left hand, automatically chooses the best hand for grasping the object according to proper criteria. This leads to a considerable improvement of the performance, since we enlarge the dexterous workspace and the robot grasping capabilities.

We validated our entire pipeline on the iCub humanoid robot [5], demonstrating significant improvements in the number of graspable objects with respect to tests performed in [2].

The paper is organized as follows. Section II reviews the state of the art on object grasping approaches. In Section III, we briefly recall the modeling and grasping approach fully described in [2]. Section IV explains the main contributions of this work. In Section V, we describe the integration in our pipeline of the classification system. Section VI collects the results of the validation experiments we carried out for testing the complete pipeline. Finally, Section VII ends the paper with concluding remarks and perspectives for future work.

II. RELATED WORKS

Grasp synthesis problem consists of finding a pose of the robot hand that satisfy a set of criteria for grabbing a given object. This problem is frequently addressed in the robotics community, since an ultimate approach that works effectively in a wide range of conditions has not been found yet.

Grasp methodologies can be classified according to several criteria. Sahbani et al. [6] divide the techniques into analytic and empirical. Analytic approaches construct force-closure grasps with multi-fingered robot hand that exhibit certain properties, such as stability and dexterity. Grasp synthesis is then formulated as an optimization problem that aim to minimize some given cost functions. Empirical or data-driven techniques, instead, compute grasp candidates relying on dedicated experiments generated in simulation or collected on a real robot by using heuristics.

Bohg et al. [1] classify empirical grasping techniques according to the role of the perception in the process. In particular, they group the approaches based on the *a priori* knowledge about the object: if it is *known*, *familiar* or *unknown*. In case of unknown objects, there are different ways to deal with the information acquired from the sensors, such as stereo cameras. Some methods approximate the full shape of the objects [7]–[9], while others compute grasps by using low-level features and heuristics [10], [11].

[1]G. Vezzani is with the Istituto Italiano di Tecnologia (IIT), iCub Facility, Via Morego 30, Genova, Italy is also with the University of Genova, Via All'Opera Pia, 13, 16145 Genova. giulia.vezzani@iit.it.

[2] U. Pattacini, G. Pasquale and L. Natale are with the Istituto Italiano di Tecnologia (IIT), iCub Facility, Via Morego 30, Genova, Italy. ugo.pattacini@iit.it, lorenzo.natale@iit.it.

There exist techniques that generate grasp hypotheses approximating the objects with shape primitives. Dunes et al. [7] model the object with quadrics whose minor axes are used to infer the wrist orientation. The quadric is estimated from multi-view measurements of the object in monocular images. Marton et al. [8] instead exploit object symmetries for fitting a curve to a cross section of the point cloud of the object. Then, the reconstructed model is used in a simulator for generating pose candidates. Rao et al. [12] sample grasp points from the surface of the segmented object and then exploit geometrical considerations. Bohg et al. [9] reconstruct the full object shape assuming planar symmetry by using the complete object point cloud.

Moreover, other methods generate a certain number of grasp hypotheses on the basis of specific heuristics, which are evaluated with resort to machine learning algorithms [10], [11]. Recently, data-driven approaches have been investigated and large datasets have been used for training a convolutional neural network (CNN). Successful examples are provided by [13] where hand-eye coordination for grasping is learned from monocular images and [14], where the planning of a manipulation task is formulated as a structured prediction problem.

The grasping approach described in [2] - and improved in this work - can be classified as an *empirical* technique for grasping *unknown* objects. Pose computation is formulated as a nonlinear constrained optimization problem, based on geometric considerations. The object is modeled with a *superquadric* function reconstructed in real-time using a single-view point cloud. The superquadric models are introduced in computer graphics by A. H. Barr [15] as a generalization of quadric surfaces and play an important role in graphics and computer vision [16]. The most popular technique for estimating the parameters of superquadrics fitting 3D points is proposed by Solina [17]. Further, a number of works in literature propose also suitable extensions of geometric modeling to complex shapes using a set of superquadrics [18], [19].

III. MODELING AND GRASPING WITH SUPERQUADRIC FUNCTIONS

In this Section, we briefly recall the modeling and grasping approach described in [2].

A. Object and hand modeling

The technique proposed in [2] for computing a suitable grasping pose is based on modeling the object and the volume graspable by the hand with superquadric functions.

Superquadrics are an extension of quadric surfaces and include supertoroids, superhyperboloids and superellipsoids. Superellipsoids are most commonly used in object modeling because they define closed surfaces. The best way to represent a superellipsoid - which we will call simply superquadric from now on - in an object-centered system is the *inside-outside* function:

$$F(x,y,z,\boldsymbol{\lambda}) = \left(\left(\frac{x}{\lambda_1}\right)^{\frac{2}{\lambda_5}} + \left(\frac{y}{\lambda_2}\right)^{\frac{2}{\lambda_5}}\right)^{\frac{\lambda_5}{\lambda_4}} + \left(\frac{z}{\lambda_3}\right)^{\frac{2}{\lambda_4}}. \quad (1)$$

The five parameters of Eq. (1) take into account the superquadric dimensions $(\lambda_1, \lambda_2, \lambda_3)$ and shape (λ_4, λ_5). The values (λ_4, λ_5) are also responsible for the concavity/convexity of the superquadric. In this work we focus on the use of convex superquadrics. Equation (1) provides a simple test whether a given point lies ($F = 1$) or not ($F > 1$ or $F < 1$) on the superquadric surface. Furthermore, the inside-outside description can be expressed in a generic coordinate system by adding six further variables, representing the superquadric pose (three for translation and three Euler angles for orientation), with a total of eleven independent variables, i.e. $\boldsymbol{\lambda} = [\lambda_1, \ldots, \lambda_{11}]$.

Object modeling consists in finding the superquadric $\mathcal{O}$ which best represents the object to be grasped starting from a single, partial 3D point cloud acquired from vision. In other words, we want to find those values of the parameters vector $\boldsymbol{\lambda} \in \mathbb{R}^{11}$, so that most of the N 3-D points $\boldsymbol{s}_i = [x_i, y_i, z_i]$ for $i = 1, \ldots, N$ of the point cloud lie on or close to the superquadric surface. The minimization of the algebraic distance from points to the model can be solved by defining a least-squares minimization problem:

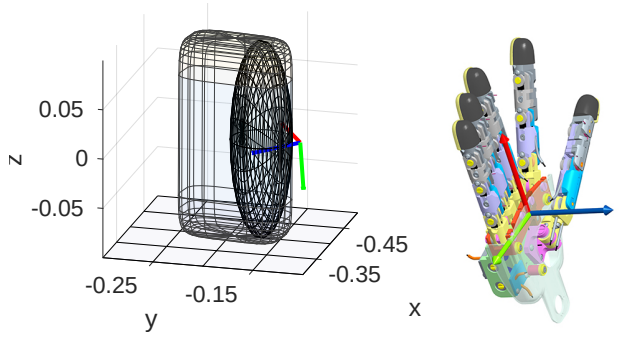
$$\min_{\boldsymbol{\lambda}} \sum_{i=1}^{N} \left(\sqrt{\lambda_1\lambda_2\lambda_3}\,(F(\boldsymbol{s}_i,\boldsymbol{\lambda}) - 1)\right)^2, \quad (2)$$

where $(F(\boldsymbol{s}_i,\boldsymbol{\lambda}) - 1)^2$ imposes the point-superquadric distance minimization and the term $\lambda_1\lambda_2\lambda_3$, which is proportional to the superquadric volume, compensates for the fact that the previous equation is biased towards larger superquadric. This problem can be efficiently solved by the Ipopt [20], a software package capable of solving large scale, nonlinear constrained optimization problems.

The volume graspable by the hand is instead represented by a fictitious superquadric, whose shape and pose are chosen by considering the anthropomorphic shape of the robot hand and its grasping capabilities. A suitable shape for this purpose is shown to be an ellipsoid $\mathcal{H}$ attached to the hand palm (see Fig. 1).

B. Grasping pose computation

The grasping approach proposed in [2] computes a feasible pose $\boldsymbol{x} \in \mathbb{R}^6$ of the robot hand which allows grabbing the object by looking for that pose $\boldsymbol{x}$ that makes the hand ellipsoid $\mathcal{H}$ overlap with the object superquadric $\mathcal{O}$ while meeting a set of requirements. We choose $\boldsymbol{x} = [x_h, y_h, z_h, \phi_h, \theta_h, \psi_h]$, where (x_h, y_h, z_h) are the coordinates of the origin of the hand frame and $(\phi_h, \theta_h, \psi_h)$ are the RPY Euler angles, accounting for orientation. The desired pose is computed by



(a) (b)

Fig. 1. Fig. (a): an example of grasping pose computed for the right iCub hand with the pose computation approach of [2]. The figure shows both the object model $\mathcal{O}$ and the hand ellipsoid $\mathcal{H}$. Fig. (b): the right hand reference frame in RGB convention: red for x, green for y and blue for z axis.

solving the following optimization problem:

$$\min_{\boldsymbol{x}} \sum_{i=1}^{L} \left(\sqrt{\lambda_1 \lambda_2 \lambda_3} \left(F(\boldsymbol{p}_i^{\boldsymbol{x}}, \boldsymbol{\lambda}) - 1 \right) \right)^2, \quad \text{subject to:} \quad h_i(\boldsymbol{a_i}, f_i(\boldsymbol{p}_1^{\boldsymbol{x}}, \ldots, \boldsymbol{p}_L^{\boldsymbol{x}})) > 0, \; for \; i = 1, \ldots, M. \tag{3}$$

Hereafter, we briefly recall the meaning of the most important quantities of Eq. (3). The exhaustive description of Eq. (3) and the pose computation approach is provided in [2].

The cost function in Eq. (3) imposes the minimization of the distance between the object superquadric $\mathcal{O}$, represented by the inside-outside function $(F(\cdot, \boldsymbol{\lambda}) - 1)$, and L points $\boldsymbol{p}_i^{\boldsymbol{x}} = \left[p_{x,i}^{\boldsymbol{x}}, p_{y,i}^{\boldsymbol{x}}, p_{z,i}^{\boldsymbol{x}}\right]$ for $i = 1, \ldots, L$, properly sampled on the surface of the hand ellipsoid $\mathcal{H}$, whose pose is given by vector $\boldsymbol{x}$.

The M constraints of Eq. (3) take into account obstacle avoidance requirements. Each term h_i, for $i = 1, \ldots, M$ is the implicit function representing the i-th obstacle. As is in [2], the only obstacle is the table on which the object is located, hence $M = 1$. The quantity $h_1(\boldsymbol{a}_1, f_1(\cdot)) = h(\boldsymbol{a}, f(\cdot))$ is the implicit function of the plane modeling the table. The vector $\boldsymbol{a}$ consists of the parameters of the plane function and each $f(\boldsymbol{p}_{x1}, ..., \boldsymbol{p}_{x1})$ accounts for a dependency on the L points $\boldsymbol{p}_i$ suitably designed for the grasping task.

The optimization problem of Eq. (3) is efficiently solved by using the Ipopt package, meeting the requirements for real-time applications and providing local minimizer when no global solution is available. Fig. 1 shows an example of grasping pose computed by solving Eq. (3).

Suitable joint trajectories to reach for the grasping pose are provided by the Cartesian controller available on the iCub [21].

IV. IMPROVING PIPELINE RELIABILITY

In this work, we propose an improved modeling and grasping pipeline[1] with respect to the one described in [2] by adding novel features and integrating a visual object categorization system (see Section V).

In order to increase the overall grasping reliability:

- we improve, speed up and stabilize the computation of the superquadric model described in Section III, by means of the use of prior information on the object shape;
- we design a pose quality index to automatically select the best hand for grasping the object, once the right and the left pose candidates are computed.

In the next paragraphs, we detail the improvements we propose, whereas in Section VI we show their effect on the grasping performance.

A. Modeling improvement with prior on object shape

As we recall in Section III, object modeling consists in estimating the parameters vector $\boldsymbol{\lambda} \in \mathbb{R}^{11}$ of a superquadric function through the optimization problem of Eq. (1).

A useful information to reduce the computational time required to solve the optimization problem is to properly set the lower $\boldsymbol{\lambda}_l \in \mathbb{R}^{11}$ and upper bounds $\boldsymbol{\lambda}_u \in \mathbb{R}^{11}$ of the variables to be estimated $\boldsymbol{\lambda} \in \mathbb{R}^{11}$. Reasonable bounds for the object dimensions $(\lambda_1, \lambda_2, \lambda_3)$ and position $(\lambda_6, \lambda_7, \lambda_8)$ can be extracted respectively from the volume occupancy and the centroid of the 3D point cloud.

We can instead obtain proper bounds on the superquadric exponents (λ_4, λ_5), that are responsible for the object shape, by classifying the objects according to their similarity to shape primitives, such as cylinders, parallelepipeds and spheres. Each shape is in fact identified by a specific couple (λ_4, λ_5) in the superquadric representation: $(\lambda_4, \lambda_5)_{cyl}$ = (0.1, 1.0) for cylinders, $(\lambda_4, \lambda_5)_{par}$ = (0.1, 0.1) for parallelepipeds and $(\lambda_4, \lambda_5)_{sph}$ = (1.0, 1.0) for spheres (Fig. 2). Thus, we can use different lower and upper bounds for the superquadric exponents according to the shape primitive of the object. The bounds can be expressed as:

$$\begin{aligned} (\lambda_4, \lambda_5)_{l,shape} &= (\lambda_4, \lambda_5)_{shape} - (\Delta_{l,4}, \Delta_{l,5}) \\ (\lambda_4, \lambda_5)_{u,shape} &= (\lambda_4, \lambda_5)_{shape} + (\Delta_{u,4}, \Delta_{u,5}), \end{aligned} \tag{4}$$

where the label *shape* stands for one of the shape primitives. The bounds shown in Eq. (4) force the superquadric shape to be of the category identified via object classification. The $(\Delta_{l,4}, \Delta_{l,5})$ and $(\Delta_{u,4}, \Delta_{u,5})$ values are positive numbers introduced in order to deal with the noise affecting the point cloud. In fact, an object point cloud might be better represented by a superquadric with softer edges due to its noise. Fig. 3 shows an example of this phenomenon while modeling a box. The noisy point of the box cloud is represented with blue dots and we provide two examples of reconstructed

[1]Our modeling and grasping approach C++ implementation are available on Github:
https://github.com/robotology/superquadric-model/tree/master
https://github.com/robotology/superquadric-grasp/tree/feature-visualservoing.

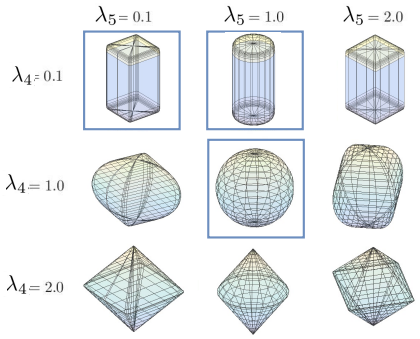


Fig. 2. How superquadric shapes change according to λ_4 and λ_5 values. We are interested only in convex objects, thus $\lambda_{4,min} = \lambda_{5,min} > 0.0$ $\lambda_{4,max} = \lambda_{5,max} < 2.0$. For avoiding difficulties with singularities we use the further bounds $\lambda_{4,min} = \lambda_{5,min} = 0.1$ [17]. In this work we take into account the object shapes highlighted with blue frames. The sharp-cornered shape of parallelepiped and cylinder shapes are caused by $\lambda_4 = 0.1$.

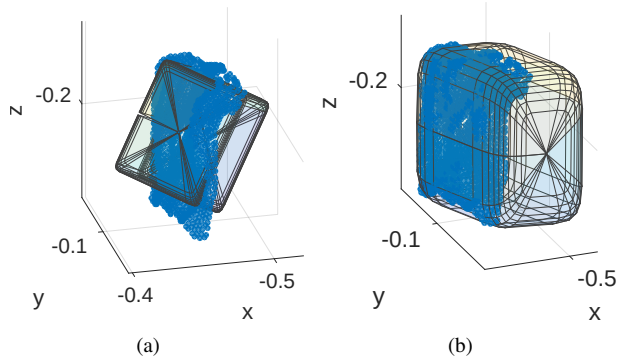


Fig. 3. Two examples of superquadric models overlapped to the acquired object point cloud. Fig. (a) shows the superquadric modeling the object obtained by setting $\Delta_{l,4} = \Delta_{u,4} = \Delta_{l,5} = \Delta_{u,5} = 0.0$. Consequentely, (λ_4, λ_5) are fixed equal to (0.1, 0.1) and they are not estimated by the optimization problem. Fig. (b) shows the superquadric modeling the object by setting $\Delta_{l,4} = \Delta_{l,5} = 0.0$ and $\Delta_{u,4} = \Delta_{u,5} = 0.4$. In this case, λ_4 and λ_5 are computed by solving the optimization problem and corresponds to (0.25, 0.25). The superquadric with softer edges shown in Fig. (b) better fits the noisy object point cloud.

superquadrics. In Fig. 3(a), we force the superquadric to be a sharp-cornered parallelepiped, i.e. $\Delta_{l,4} = \Delta_{u,4} = \Delta_{l,5} = \Delta_{u,5} = 0.0$ and, thus, $(\lambda_4, \lambda_5) = (0.1, 0.1)$. In this case, the optimization problem of Eq. (2) estimates only 9 parameters (instead of 11), since λ_4 and λ_5 are fixed. However, a sharp-cornered shape is not the best one for the point cloud of interest. For this reason, the solution of the optimization problem does not correctly fit the point cloud and provides a wrong model for the object. If we instead set $\Delta_{u,4} = \Delta_{u,5} > 0$, the optimization problem has to estimate also λ_4 and λ_5, since they can range in a non-zero interval. This allows properly fitting the point cloud with a superquadric with softer edges, i.e. $(\lambda_4, \lambda_5) > (0.1, 0.1)$, as result of the optimization problem (Fig. 3(b)).

B. Automatic selection of the hand

In case of robots equipped with two hands, computing grasping poses for both the hands helps enlarge the dexterous workspace and, ultimately, the grasping capabilities.

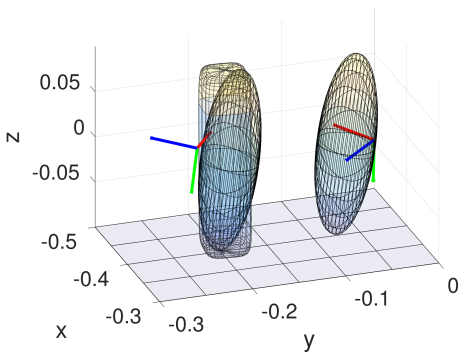


Fig. 4. Example of object graspable only by the left hand. The poses and the ellipsoids shown on the left and on the right are respectively for the left and the right hand. The object is located out of the right hand reachable workspace. For this reason, the optimization problem is not able to find a solution where the right ellipsoid is overlapped on the object model. In this case we obtain $F_{f,left} < F_{f,right}$.

However, the increased redundancy brings about a further complexity due to the need for conceiving a principled way to determine which is the best hand to be used for accomplishing the grasp. Depending on the object pose, the robot can better grasp it with the right or the left hand. The method described in [2] and summarized in Section III can be applied both on the right and left hand. What is still missing in [2] is an automatic way for selecting the hand to be used given pose candidates for the right and the left hand. At this aim, we propose to rely on the following index:

$$\mathcal{I}_{P,hand} = \frac{1}{w_1 F_{f,hand} + w_2 \ (z_{hand} \cdot z_{root})}, \tag{5}$$

that is proportional to the pose quality, i.e. the higher $\mathcal{I}_{P,hand}$, the better the pose is. The index of Eq. (5) takes into account:

- the cost function of Eq. (3) evaluated in the computed grasping pose $\boldsymbol{x}_{f,hand}$,

$$F_{f,hand} = \sum_{i=1}^{L} \left(\sqrt{\lambda_1 \lambda_2 \lambda_3} \left(F(\boldsymbol{p}_i^{\boldsymbol{x}}, \boldsymbol{\lambda}) - 1 \right) \right)^2 \Big|_{\boldsymbol{x}=\boldsymbol{x}_{f,hand}}. \tag{6}$$

  The higher $F_{f,hand}$, the worse the overlapping between the hand ellipsoid and the object superquadric and, thus, the pose are. An example is shown in Fig. 4.
- $z_{hand} \cdot z_{root}$, the inner produt between the z-axis of the frame attached to the hand and the z-axis of the root frame, thus measuring essentially the grade of alignment between the two axes. Given the definition of the hand and root reference frames, this quantity is used to favor lateral or top grasps, that give values < 0.8 in absolute value (Fig. 5).

These two terms are combined together with the proper weights w_1 and w_2 to make the quantities comparable. In particular, w_2 is > 0 or < 0 depending if the index is computed for the left or the right hand in order to take into account for the different orientation of the hand frames.

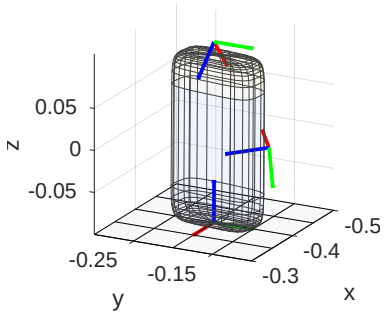


Fig. 5. In this figure, the root reference frame is represented on the horizontal plane representing the table where the object lies. Then, we report an example of top and lateral grasping poses.

In summary, once pose candidates are computed for the right and the left hand, we evaluate the quantities $\mathcal{I}_{P,right}$ and $\mathcal{I}_{P,left}$. The hand chosen for grasping the object is the one providing the maximum index, i.e. $\arg\max(\mathcal{I}_{P,right}, \mathcal{I}_{P,left})$.

V. INTEGRATION WORK

In Section IV-A, we claim that our modeling process improves when prior information on the object shape is available. To this end, we classify the objects according to their similarity to shape primitives. Object classification is formulated as a categorization problem and is achieved by taking advantage of the recognition system described in [3]. We employ the implementation currently in use on the iCub robot[2] and, specifically, we adopt the ResNet-50 Convolutional Neural Network model [22], trained on the ImageNet Large-Scale Visual Recognition Challenge [23] and available in the Caffe framework [24][3], as a feature extractor. A rectangluar region around the object of interest is cropped from the image by using an object segmentation method based on RGB information. Each cropped image is fed to the network and encoded into a 1024-dimensional vector composed of the activations of the 'pool5' layer. In the considered recognition pipeline, the extracted vector representations are fed to a multiclass Support Vector Machine (SVM), which is trained to categorize the objects according to their shape.

The training is performed on the fly in a supervised manner: a human teacher shows a number of example objects to the robot, whose labels represent their shape categories. Fig. 6 depicts the objects used to train the system on the three shapes under consideration. A few example images per objects are sufficient to achieve good prediction accuracy on the test set of 18 objects (see also Fig. 8), thanks to the use of deep representations. At test time, the object is assigned to the class with maximum score produced by the SVM classifier, if this is above a certain threshold (set empirically), otherwise it is considered a generic object. Importantly, the 18 objects used for the graping experiments are not part of the training set, i.e., they are fully novel when presented to the robot.

[2]https://github.com/robotology/himrep.

[3]https://github.com/KaimingHe/deep-residual-networks.

Fig. 6. Training set: 8 parallelepipeds, 8 cylinders and 8 spheres belonging to the YCB dataset and the iCub world dataset [25].

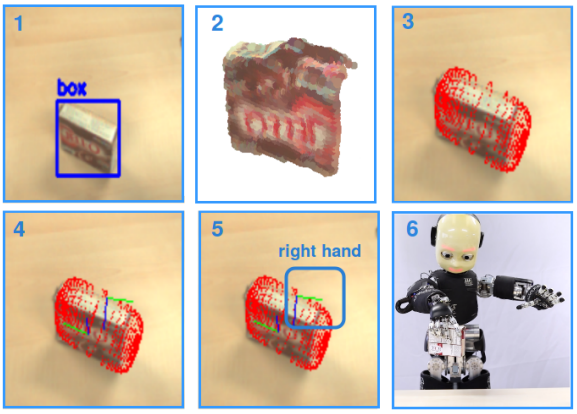


Fig. 7. The complete and improved modeling and grasping pipeline.

With the addition of object categorization step, the entire novel pipeline we propose is the following (Fig. 7):

1) A rectangular crop of the image is extracted from the camera images. The categorization system uses the pipeline proposed in [3] to classify the cropped image according to its similarity to shape primitives. We take into account three possible shapes: cylinder, parallelepiped and sphere. All those objects that cannot be well represented with a shape primitive (e.g. some plushes of Fig. 8) are threated as generic objects.
2) The object segmentation is used for extracting the relative 3D point cloud from stereo vision.
3) The modeling approach computes the superquadric that better fits into the object 3D points.
4) Our grasping approach finds a pose candidate for the right and left hand by solving Eq. (3).
5) The pose quality indices $\mathcal{I}_{P,right}$ and $\mathcal{I}_{P,left}$ are evaluated and the hand with maximum index value is chosen for grasping the object: $hand = \arg\max(\mathcal{I}_{P,right}, \mathcal{I}_{P,left})$.
6) The robot uses the selected hand to reach for the grasping pose. Once the final pose is reached, the robot close the fingers until a contact is detected by the tactile sensors mounted on the fingertips. Then, the tactile feedback on the fingertips is used continuosly to achieve a stable grasp of the object [4] and the robot

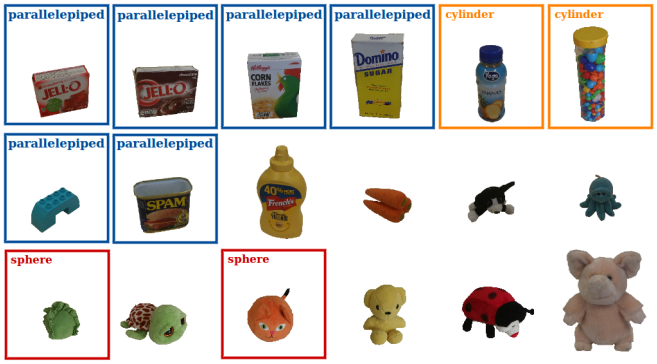


Fig. 8. Classification results on the test set. The objects whose confidence is lower than a threshold for all the shape primitives are not classified and are considered as generic objects from the superquadric modeling process.

lifts the object.

VI. EXPERIMENTS

In this section we detail the experiments we carried out for testing our pipeline for modeling and grasping tasks. We extensively evaluate our approach on the iCub humanoid robot by using the 18 objects shown in Fig 8. The test set consists of the objects used in [2] plus 13 objects, including objects of the YCB dataset [26]. The objects have been selected so as to be graspable by the iCub: we discarded objects with slippery surfaces, that are too large or too heavy for the robot hand and too small for being grasped with a power grasp technique.

A. Modeling results

In this paragraph, we report the results we obtained with our improved modeling approach. In Fig. 8, we show how the objects of the test set are classified. The objects that cannot be modeled with a shape primitive are considered as generic objects in the superquadric reconstruction step. Fig. 9 highlights how the use of prior on the object shape improves the model reconstructed for a box (i.e. a parallelepiped). Table I contains the superquadric computation time we achieve thanks to the use of prior information on the object shape. The computation time is 20 times faster with respect to the values obtained in [2]. In fact, the use of priors on the object shape allows also reducing the number of 3D points sampled from the object point cloud from 300, as in [2], to 50, without loosing accuracy of the overall process.

The use of prior information on object shape increases the reliability of the modeling process. In order to evaluate this improvement, we performed the test we termed *Experiment 1*. We executed 10 grasping trials with the improved modeling approach and with the one described in [2]. In Table II, we report the success rate of the tests. In order to focus only on the improvements given by the new modeling approach, we put all the objects in the left hand workspace, so as to use only one hand for this evaluation.

The experiments show a performance gain when we use prior shape information. The prior information on object shapes helps to obtain a finer and more sharp-cornered model that is crucial for computing better grasping poses for parallelepipeds, cylinders and spheres. For example, the box model shown in Fig. 9 (c) leads to pose candidates on the top or on the lateral surfaces of the box, since the hand ellipsoid better overlaps on those portions of the object superquadric. If, instead, we use the model of Fig. 9 (a), the model made of rounded corners lets the final pose lie also on the box top corners (Fig. 9 (b)).

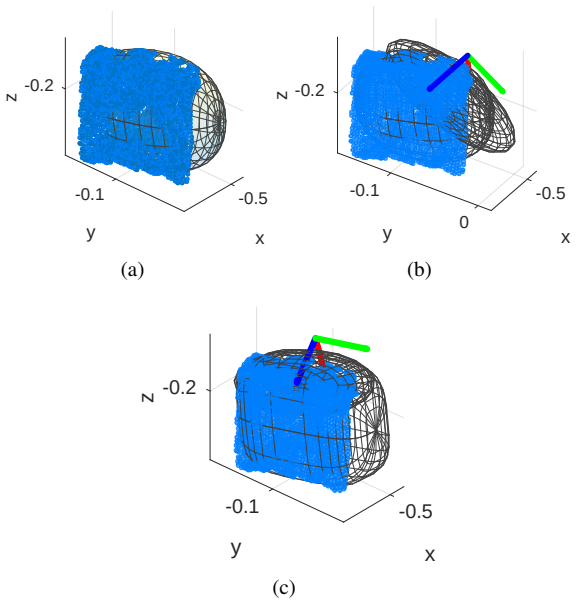


Fig. 9. Superquadric models of the jello box 1 overlapped on the complete object point clouds (represented with blue dots). We show the superquadric obtained without any prior information on the object shape (Fig. (a)) and the relative grasping pose (Fig. (b)) and the same quantities obtained with the prior information (Fig. (c)). The model obtained with prior information has sharp-cornered shapes. The use of prior information enable to significantly downsample the object point cloud used for superquadric estimation (number of points used=50) and to obtain better grasping poses, i.e. located on the top surface of the box (Fig (c)) instead of on the box corners (Fig. (b)).

TABLE I

EXECUTION TIME FOR MODEL RECONSTRUCTION

Object	Average time [s]	Object	Average time [s]
Cylinder	0.09	Pig	0.13
Cat	0.09	Lettuce	0.08
Bear	0.12	Mustard box	0.09
Juice bottle	0.06	Sugar box	0.08
Jello box 1	0.08	Turtle	0.13
Meat can	0.10	Lego brick	0.05
Carrots	0.06	Cereal box	0.09
Jello box 2	0.07	Octopus	0.04
Dog	0.10	Ladybug	0.12

Table I indicates the average execution time across 10 trials for model reconstruction process of each object including prior information and by using 50 points sampled from the object point clouds.

TABLE II

EXPERIMENT 1: PERCENTAGE OF SUCCESSFUL GRASPS

Object	Success on Trials [%] Old modeling approach	Success on Trials [%] **New modeling approach**
Jello box 1	40%	**100%**
Jello box 2	60%	**90%**
Cereal box	60%	**80%**
Sugar box	60%	**90%**
Juice bottle	80%	**90%**
Cylinder	70%	**90%**
Lego brick	50%	**80%**
Meat box	50%	**80%**
Mustard box	60%	**80%**
Carrots	40%	**70%**
Dog	50%	**70%**
Octopus	70%	**80%**
Lettuce	50%	**80%**
Turtle	70%	**80%**
Cat	70%	**80%**
Bear	70%	**90%**
Ladybug	50%	**90%**
Pig	60%	**70%**

Table II shows the percentage of successful grasps, in case the old and the new modeling approach (including prior on object shape) are used. The number of points is 50 in both the experiment.

TABLE III

EXPERIMENT 2: PERCENTAGE OF SUCCESSFUL GRASPS

Object	Success on Trials [%] One hand approach	Success on Trials [%] **Automatic hand selection**
Jello box 1	90%	**100%**
Jello box 2	80%	**90%**
Cereal box	70%	**90%**
Sugar box	70%	**90%**
Juice bottle	80%	**90%**
Cylinder	70%	**100%**
Lego brick	80%	**90%**
Meat box	60%	**80%**
Mustard box	70%	**90%**
Carrots	60%	**80%**
Dog	80%	**90%**
Octopus	90%	**100%**
Lettuce	70%	**90%**
Turtle	60%	**80%**
Cat	70%	**80%**
Bear	60%	**100%**
Ladybug	70%	**90%**
Pig	50%	**70%**

Table III shows the percentage of successful grasps, in case only one hand is used for grasping the hand and the automatic selection of the hand is implemented.

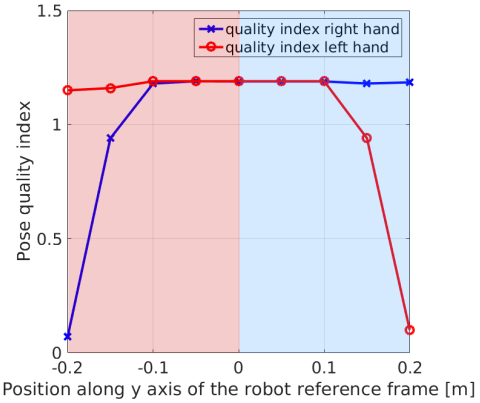


Fig. 10. Pose quality index for right hand and left hand of the jello box 1 in different positions. The indices have been computed sliding the object along the y axis of the robot reference frame from the left (in red) to the right (in blue) workspace.

B. Automatic selection of the hand

In order to evaluate the effectiveness of our automatic approach for selecting the hand for grasping the object, we evaluate the pose quality indices $\mathcal{I}_{P,right}$ and $\mathcal{I}_{P,left}$ by varying the object position (with the same orientation) from the left hand to the right hand workspace. The trend of the indices obtained with the jello box 1 is shown in Fig. 10. As expected, the index for each hand is higher in the hand workspace and decrease while the object position goes towards the other hand workspace.

In addition, we executed the following experiment (*Experiment 2*) to show how the pose computation for both the hands and the selection of the best hand for grasping the object increases the number of successful grasps. We put the object of interest in a fixed position reachable by both the hands and we change only its orientation during each trial. Table III compares the success rate if, respectively, only one hand or two hands are used for grasping the object. Even if the object is in a workspace sufficiently dexterous for the both hands, its orientation and reconstructed model can favor one hand with respect to the other, increasing the success percentage when the best hand is automatically selected for grasping the object.

VII. CONCLUSIONS

In this paper, we improve the object modeling and grasping pipeline described in [2]. In particular, we refine and stabilize the object superquadric modeling technique by using prior information on the object shape. The prior information is provided by a visual object classifier we trained and integrated in the pipeline employing the recognition system of [3]. In addition, we propose a novel pose quality index for automatically selecting the best hand for grasping the object among two pose candidates for the right and the left hand.

We evaluated the improved pipeline on 18 real objects with the iCub humanoid robot, focusing on the effects of the refined modeling process and the automatic selection of the hand. The experiments highlight how these contributions increase the pipeline reliability in terms of number of successful grasps. The overall success rate of the entire pipeline is nearly 85%.

The main source of failures is represented by the uncalibrated eye-hand system of the robot that entails non-negligible misplacements of the robot hand when reaching for the target pose. This problem is peculiar of humanoid robots in that elastic elements lead to errors in the direct kinematics computation. Moreover, robots with moving cameras, such as the iCub platform, need to deal with errors in

the visual estimation of the object pose due to imprecise knowledge of the cameras extrinsic parameters. These errors can be compensated by closed loop control techniques of the end-effector resorting to a visual feedback. At this regard, we could improve the grasping reliability by integrating the visual servoing technique described in [27], where desired poses computed from stereo vision are accurately reached by the robot end-effector thanks to the use of a precise end-effector pose estimate over time.

The pipeline we propose in this paper can be extended in several ways. At first, we are aware of the fact that a trajectory plan for reaching the final pose is still missing in our work. A viable solution is to use our approach also for computing a set of waypoints, together with the final grasping pose. At this aim, superquadrics could be used to model obstacles, and their avoidance added as optimization constraints as shown in Section III-B. Another extension is the formulation of a supervised learning method for automatically discriminate good grasping poses. The approach in this paper in fact only selects the best pose between two candidates, even if neither of them is suitable for grasping the object. Moreover, we aim in the future at refining the estimation of the object model by using a set of superquadrics in place of only a single superquadric [18], [19]. This way, we will be able to accurately model more complex and concave objects and, thus, locate the point of contact of the fingers onto specific parts of the objects.

REFERENCES

[1] J. Bohg, A. Morales, T. Asfour, and D. Kragic, "Data-Driven Grasp Synthesis - A Survey," *IEEE Transactions on Robotics*, vol. 30, no. 2, pp. 289–309, 2015.

[2] G. Vezzani, U. Pattacini, and L. Natale, "A grasping approach based on superquadric models," in *34th IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 1579–1586.

[3] G. Pasquale, C. Ciliberto, F. Odone, L. Rosasco, and L. Natale, "Teaching icub to recognize objects using deep convolutional neural networks," vol. 43, 2015, pp. 21–25. [Online]. Available: http://www.jmlr.org/proceedings/papers/v43/pasquale15

[4] M. Regoli, U. Pattacini, G. Metta, and L. Natale, "Hierarchical grasp controller using tactile feedback," in *IEEE-RAS 16th International Conference on Humanoid Robots*. IEEE, 2016, pp. 387–394.

[5] G. Metta, L. Natale, F. Nori, G. Sandini, D. Vernon, L. Fadiga, C. Von Hofsten, K. Rosander, M. Lopes, J. Santos-Victor, *et al.*, "The iCub humanoid robot: An open-systems platform for research in cognitive development," *Neural Networks*, vol. 23, no. 8, pp. 1125 – 1134, 2010.

[6] A. Sahbani, S. El-Khoury, and P. Bidaud, "An overview of 3d object grasp synthesis algorithms," *Robotics and Autonomous Systems*, vol. 60, no. 3, pp. 326–336, 2012.

[7] C. Dune, E. Marchand, C. Collowet, and C. Leroux, "Active rough shape estimation of unknown objects," in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*. IEEE, 2008, pp. 3622–3627.

[8] Z.-C. Marton, D. Pangercic, N. Blodow, J. Kleinehellefort, and M. Beetz, "General 3d modelling of novel objects from a single view," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*. IEEE, 2010, pp. 3700–3705.

[9] J. Bohg, M. Johnson-Roberson, B. León, J. Felip, X. Gratal, N. Bergström, D. Kragic, and A. Morales, "Mind the gap-robotic grasping under incomplete observation," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 686–693.

[10] A. Saxena, L. Wong, and A. Ng, "Learning grasp strategies with partial shape information," in *AAAI Conference on Artificial Intelligence*, 2008.

[11] R. Detry, R. Baseski, M. Popovic, Y. Touati, N. Kruger, O. Kroemer, J. Peters, and J. Piater, "Learning object-specific grasp affordance densities," in *IEEE International Conference on Development and Learning*, 2009.

[12] D. Rao, Q. V. Le, T. Phoka, M. Quigley, A. Sudsang, and A. Y. Ng, "Grasping novel objects with depth segmentation," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*. IEEE, 2010, pp. 2578–2585.

[13] S. Levine, P. Pastor, A. Krizhevsky, and D. Quillen, "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection," *preprint available at arXiv:1603.02199*, 2016.

[14] J. Sung, S. Hyun Jin, I. Lenz, and A. Saxena, "Robobarista: learning to manipulate novel objects via deep multimodal embedding," *preprint available at arXiv:1601.02705*, 2016.

[15] A. H. Barr, "Superquadrics and angle preserving transformations," *IEEE Computer Graphics and Applications*, vol. 1, no. 1, pp. 11 – 23, 1981.

[16] A. H. Barr, "Global and local deformations of solid primitives," *Computer Graphics*, vol. 18, no. 3, pp. 21 – 30, 1981.

[17] A. Jaklic, A. Leonardis, and F. Solina, "Segmentation and recovery of superquadrics," *Computational imaging and vision*, vol. 20, 2000.

[18] L. Chevalier, F. Jaillet, and A. Baskurt, "Segmentation and superquadric modeling of 3d objects," 2003.

[19] K. Duncan, S. Sarkar, R. Alqasemi, and R. Dubey, "Multi-scale superquadric fitting for efficient shape and pose recovery of unknown objects," in *IEEE International Conference om Robotics and Automation (ICRA)*, 2013, pp. 4238 – 4243.

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

[21] "The YARP cartesian interface," *description available at Cartesian Interface*.

[22] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[23] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, Dec 2015. [Online]. Available: https://doi.org/10.1007/s11263-015-0816-y

[24] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the 22nd ACM international conference on Multimedia*. ACM, 2014, pp. 675–678.

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

[26] B. Calli, A. Walsman, A. Singh, S. Srinivasa, P. Abbeel, and A. M. Dollar, "Benchmarking in manipulation research: Using the yale-cmu-berkeley object and model set," *IEEE Robotics & Automation Magazine*, vol. 22, no. 3, pp. 36–52, 2015.

[27] C. Fantacci, U. Pattacini, V. Tikhanoff, and L. Natale, "Visual end-effector tracking using a 3D model-aided particle filter for humanoid robot platforms," in *accepted at 2017 IEEE Conference on Intelligent Robots and Systems (IROS), preprint: http://arxiv.org/abs/1703.04771*. IEEE, 2017.