

A Grasping Approach Based on Superquadric Models

Giulia Vezzani¹, Ugo Pattacini² and Lorenzo Natale²

Abstract—This paper addresses the problem of grasping unknown objects with a humanoid robot. Conventional approaches fail when the shape, dimension or pose of the objects are missing. We propose a novel approach in which the grasping problem is solved by modeling the object and the volume graspable by the hand with *superquadric* functions. The object model is computed in real-time using stereo vision. Pose computation is formulated as a nonlinear constrained optimization problem, which is solved in real-time using the Ipopt software package. Notably, our method finds solutions in which the fingers are located on portions of the object that are occluded by vision. The performance of our approach has been assessed on a real robotic system, the iCub humanoid robot. The experiments show that the proposed method computes proper poses, suitable for grasping even small objects, while avoiding hitting the table with the fingers.

I. INTRODUCTION

Industrial robotics shows how high performance in manipulation can be achieved in terms of speed, precision and reliability, if a very accurate knowledge of the environment and the objects to manipulate is provided. However grasping of unknown objects or objects whose pose is uncertain is still an open problem [1].

Our work aims at improving grasping when the shape or pose of objects are uncertain. The first main contribution of our approach consists of reconstructing in real-time a model for the object under consideration and representing the robot hand with proper and mathematically usable models, i.e. *superquadric functions*. Our choice of models exploitation, even if approximated, finds support in behavioral and neurophysiologists studies, that demonstrate how 3D shape information plays a fundamental role in human grasping capabilities [2], [3]. The volume graspable by the hand is represented by an ellipsoid and is defined *a-priori*, because the shape of the hand and its pose are known in advance. The superquadric representing the object is obtained in real-time from partial vision information instead, e.g. one stereo view of the object under consideration, and provides an approximated 3D full model. Some grasping techniques in literature exploit only 3D partial models for objects [4], but the usage of full superquadric models has the advantage of taking into account even *occluded* regions, which may have better grasping properties. In addition, a hand-modeling based approach provides useful information for power grasp computations. In fact, a rough model of the

hand volume can help to identify grasp candidates that are compatible with the maximum opening of the hand and the finger range of movements. The second main contribution of our approach is the optimization problem we formulate for the grasping pose computation, which can be solved online by using the Ipopt software package [5] and, thus, does not require off-line computation or learning. In practice, our approach provides a real-time novel pipeline for object modeling and grasping: given an object in the robot field of view, a representative superquadric is computed and used for grasping pose computation.

The paper is organized as follows. Section II reviews the state-of-art on object grasping and object modeling with superquadric functions. Then, Section III introduces the maths behind the superquadrics and object model reconstruction, together with the description of the method we proposed for grasping pose computation. Section IV validates our approach by showing a set of successful grasps performed by the robot iCub [6] with different every-day objects. Finally, Section V ends the paper with concluding remarks and perspectives for future work.

II. RELATED WORKS

1) *Grasping Approaches*: Grasping of unknown objects is an open problem in literature hence diverse methodologies are still being explored. Moreover, different goals can be included in the generic field of grasping. For instance, grasp actions can be divided into *power* and *precision* grasps [7]. Power grasp involves large areas of contact between the hand and the object, without adjustment of the fingers after contact [4]. On the contrary, precision grasp provides sensitivity and dexterity, since in this case the object is held with the tips of the fingers [8]. In precision grasp tasks, the hand touches the object at small contact points, therefore the study of grasp stability plays an important role. In this work we consider only power grasp actions.

Yet another classification criterion considers how the robot hand needs to approach the object to enable a successful grasp. Especially in the past, the grasping problem has been addressed using *analytical* techniques [9]. Such methods formulate the grasping problem only in terms of force-closure and form-closure, looking for specific conditions on the contact wrenches that ensure a certain hand configuration to firmly hold any object. These approaches usually assumed that contact point locations were given without explicitly relating the hand configuration to the object geometry. Recent works attempt to get around these limitations [10]–[19]; these methods are

¹G. 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 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.

called *empirical* and mimic human grasping in selecting a grasp that best conforms to task requirements and the target object geometry. Our method belongs to this category, because the hand pose is computed by exploiting object shape information.

Several empirical approaches have been proposed in the last decades. Some of them attempt to create a direct mapping between object shape and hand pose [10], [11]. Other techniques generate a certain number of grasp hypotheses on the basis of specific heuristics, and then evaluate them with machine learning algorithms such as Artificial Neural Networks [12], [13], simple maximum likelihood algorithms [14], or kernel density estimation methods [15]. Others simply consider a set of possible grasp configurations and choose them according to some shape properties [16]–[18]. An interesting approach [19] identifies handle-like grasp affordances in 3-D point clouds. 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 [20] where hand-eye coordination for grasping is learned from monocular images and [21], where the planning of a manipulation task is formulated as a structured prediction problem whose resulting strategy is transferred across different objects by embedding point-cloud along with trajectory data into a shared representation using a deep neural network.

Despite the good performance and their growing popularity, the drawback of learning approaches is that they require time consuming data-gathering and off-line training processes. In this regard, another possible grasping problem formulation (even if less common in literature) is the *one-shot* approach, in which the goal is to compute good grasp poses, without learning preprocesses [22].

Our work can be then defined as a *one-shot empirical* method for *power* grasp actions.

2) *Superquadric Functions*: The superquadric model [23], [24] has been introduced in computer graphics by A.H. Barr in 1981, as a generalization of quadrics and has been well studied in graphics and computer vision [25]. Superquadrics and extensions such as hyperquadrics [26] and deformable superquadrics [24] are a convenient representation for a large class of both convex and non-convex objects. The most popular method to determine superquadric parameters for fitting 3D points was proposed by Soline in 1990 [25]. Recently, several works have focused on speeding up computation, refining the model and extending it to approximate complex shapes with a set of superquadrics [27], [28].

Despite their ability to represent a wide range of objects with a small number of parameters, superquadrics have not been widely used as an object representation for contact tasks such as dynamic simulation or dextrous manipulation. This is in part due to the lack of appropriate distance functions and collision detection algorithms. In general, there is no closed-form solution for the distance between two superquadrics (and in fact no closed-form solution exists

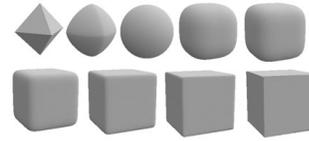


Fig. 1. Superquadric functions can represent several simple objects, ranging from boxes (on the right) to octaedrus (on the left).

even for the distance between a point and an ellipsoid) and only approximated approaches have been proposed [29]. For this reason, several works, such as [10], [30], [31], use superquadrics merely for object modeling, whereas grasp planning is computed using simulators (e.g. GraspIt! [32]). It is worth mentioning the recent work in [33] in which grasping exploits object model information obtained with superquadrics and extrusions patterns. Our approach is different in that it exploits superquadric functions not only for object modeling (as in [10], [30], [31], [33]) and for grasping pose computation, but also for representing the robot hand.

The advantage of our approach is that it allows solving the grasping problem by means of optimization while taking additional constraints into account, such as obstacle avoidance and object penetration.

III. GRASPING POSE COMPUTATION

In this work, we propose a novel technique for computing a suitable grasping pose, which is based on modeling the object and the robot hand. For this purpose, we choose superquadric functions, since they provide a mathematical representation that uses a small number of parameters and, thus, is suitable for near real-time computation. We briefly describe the superquadric function and its properties in Section III-A. Then, we show how we make use of superquadrics for object modeling in Section III-B and, in the end, we explain in detail in Section III-C our method for computing the grasping pose.

A. Superquadrics

Superquadric functions 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. Examples of elementary objects that can be represented with superellipsoids are depicted in Fig 1. The best way to define a superellipsoid – which we will call simply superquadric from now on – in an object-centered coordinate system is the *inside-outside* function:

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

Equation (1) uses only five parameters, i.e. $\boldsymbol{\lambda} = [\lambda_1, \dots, \lambda_5]$, and provides a simple test whether a given point lies inside or outside the superquadric. If $F < 1$, the given point (x, y, z) is inside the superquadric, if $F = 1$ the corresponding point

lies on the surface of the superquadric, and if $F > 1$ the point lies outside the superquadric. 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. $\lambda = [\lambda_1, \dots, \lambda_{11}]$. We choose the RPY (roll-pitch-yaw) notation for the Euler angles.

B. Superquadric Modeling

1) *Object Modeling*: The superquadric \mathcal{O} which best represents the object to be grasped is reconstructed from a single, partial 3D point cloud, acquired by a stereo vision system. In particular, object modeling via superquadrics consists of finding those values of the parameters vector $\lambda \in \mathbb{R}^{11}$, so that most of the N 3-D points $\mathbf{s}_i = [x_i, y_i, z_i]$ for $i = 1, \dots, N$, acquired by means of the stereo vision system, 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:

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

where $(F(\mathbf{s}_i, \lambda) - 1)^2$ imposes the point-superquadric distance minimization, whereas 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.

The generic formulation of superquadrics allows for object modeling by solving a single optimization problem (Eq. (2)), without requiring *a-priori* information or making assumptions about the object shape.

In the literature Eq. (2) is usually solved via Levenberg-Marquardt. In this paper we propose to use the Ipopt instead [5], a software package capable of solving large scale, nonlinear constrained optimization problems.

2) *Hand Modeling*: A fictitious superquadric model is exploited to represent the volume graspable by the hand. The shape and pose of such superquadric are chosen by considering the anthropomorphic shape of the robot hand and its grasping capabilities. A suitable shape for this purpose turns out to be the ellipsoid \mathcal{H} attached to the hand palm, as shown in Fig. 2.

C. Grasp Pose Computation

The solution of the grasping problem consists of a feasible pose of the robot hand, which allows grabbing the object under consideration. The hand pose can be represented with a 6D vector $\mathbf{x} = [x_h, y_h, z_h, \phi_h, \theta_h, \psi_h]$, 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 basic idea of our approach is to compute the solution by looking for a pose \mathbf{x} that makes the hand ellipsoid \mathcal{H}

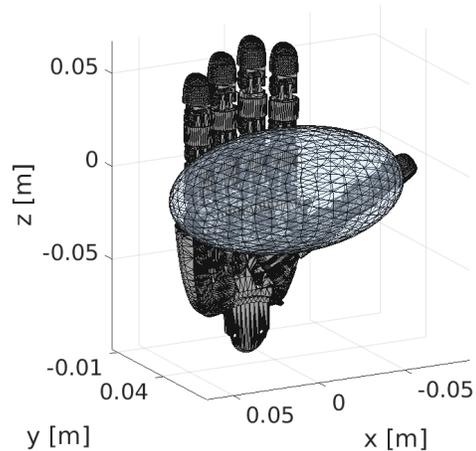


Fig. 2. The volume graspable by the hand is represented as the ellipsoid \mathcal{H} attached to the hand. The right hand of the robot iCub is represented by the CAD model

overlap with the object superquadric \mathcal{O} while meeting a set of requirements that guarantee \mathbf{x} is reachable by the robot hand.

The general formulation we propose can be described by the following nonlinear constrained optimization:

$$\begin{aligned} \min_{\mathbf{x}} \sum_{i=1}^L \left(\sqrt{\lambda_1 \lambda_2 \lambda_3} (F(\mathbf{p}_i^{\mathbf{x}}, \lambda) - 1) \right)^2, \\ \text{subject to:} \\ h_i(\mathbf{a}_i, f_i(\mathbf{p}_1^{\mathbf{x}}, \dots, \mathbf{p}_L^{\mathbf{x}})) > 0, \\ \text{for } i = 1, \dots, M. \end{aligned} \quad (3)$$

Hereafter, we report on the meaning of all the mathematical quantities contained in Eq. (3).

1) *Grasping avoiding Object Penetration*: The cost function in Eq. (3) imposes the minimization of the distance between the object superquadric \mathcal{O} , represented by the inside-outside function $(F(\cdot, \lambda) - 1)$, and L points $\mathbf{p}_i^{\mathbf{x}} = [p_{x,i}^{\mathbf{x}}, p_{y,i}^{\mathbf{x}}, p_{z,i}^{\mathbf{x}}]$ for $i = 1, \dots, L$, sampled on the surface of the hand ellipsoid \mathcal{H} , whose pose is given by vector \mathbf{x} . More precisely, the L points lie on the closest half of the ellipsoid \mathcal{H} to the hand (Fig. 3(b)). This design choice hinders the robot hand from penetrating the object. In fact, if the points were uniformly sampled on the entire \mathcal{H} surface, the point-superquadric distance minimization could place the ellipsoid \mathcal{H} in the center of the object superquadric \mathcal{O} and, consequently, lead to the object penetration by the hand, in case \mathcal{O} is bigger than \mathcal{H} (Fig. 4(a)). By contrast, our choice avoids this scenario. The asymmetric distribution of the L points makes the distance minimization possible only if the portion of the \mathcal{H} surface under consideration lies closer to the \mathcal{O} surface, thus avoiding object penetration

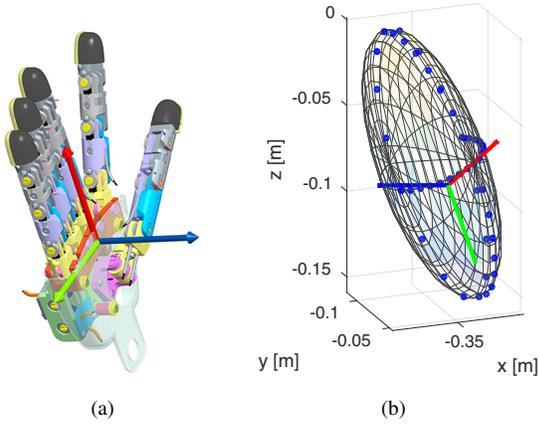


Fig. 3. In Fig. (a), the reference frame $(\mathbf{x}_h, \mathbf{y}_h, \mathbf{z}_h)$ attached to the robot hand in RGB convention (\mathbf{x}_h is coloured in red, \mathbf{y}_h in green, \mathbf{z}_h in blue). In Fig. (b): the L points sampled on the closest half of the hand ellipsoid \mathcal{H} . The RGB frame represents the hand pose, showing how the ellipsoid \mathcal{H} is attached to the hand.

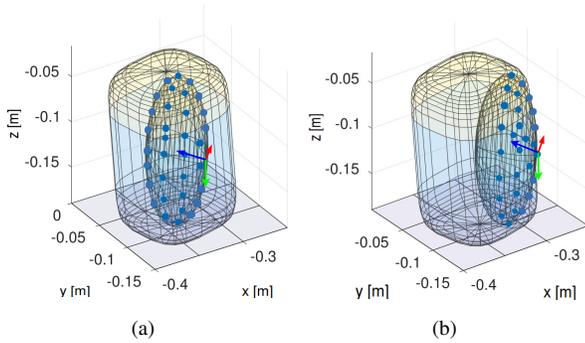


Fig. 4. If the points are uniformly sampled on the ellipsoid \mathcal{H} (i.e. the smallest ellipsoid in the plot), the minimum distance between the points and the object surface can be achieved by placing \mathcal{H} in the centroid of the object superquadric \mathcal{O} . In case (a) \mathcal{O} is bigger than \mathcal{H} , leading to object penetration (the frame attached to the robot palm is inside the object superquadric \mathcal{O}). On the other hand, if the points are sampled only on a portion of \mathcal{H} surface as shown in case (b), the distance minimization is achieved only by placing the \mathcal{H} surface (and then the hand frame) near the surface of \mathcal{O} .

with the robot hand (Fig. 4(b)).

The cost function we introduce is similar to the one exploited for object model reconstruction in Equation (2), although the optimization variable in Eq. (3) is given by the hand pose \mathbf{x} (in the coordinates of the L points \mathbf{p}_i^x), instead of the vector of superquadric parameters $\boldsymbol{\lambda}$, that is given by the object model.

2) *Obstacle Avoidance:* The use of superquadrics and implicit functions helps us define avoidance tasks. If the implicit functions modeling M obstacles under consideration are given, obstacle avoidance can be taken into account by imposing M constraints in the form of (3). Each term $h_i(\mathbf{a}_i, \cdot)$, for $i = 1, \dots, M$, is the implicit function representing the i -th obstacle, such as a support on which the object stands. Each vector \mathbf{a}_i consists of the parameters of the i -th implicit function and each $f_i(\mathbf{p}_1^x, \dots, \mathbf{p}_L^x)$ accounts for a generic dependency on the L points \mathbf{p}_i^x . The formulation displayed

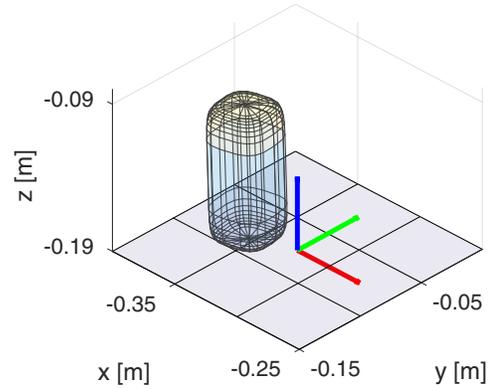


Fig. 5. The frame in RGB convention represents the plane frame $(\mathbf{x}_p, \mathbf{y}_p, \mathbf{z}_p)$. The z_p axis is parallel to the plane normal and it is positive in the space region where the object stands.

in Eq. (3) is general and can be modified according to the problem we aim to address.

In our case, the only obstacle is the table on which the object is located, hence $M = 1$ in Eq. (3). For the sake of simplicity, we refer to $h_1(\mathbf{a}_1, f_1(\cdot))$ as $h(\mathbf{a}, f(\cdot))$. The table is modeled as a plane, whose implicit function is thus linear and given by:

$$h(\mathbf{a}, x, y, z) = a_1 x + a_2 y + a_3 z + a_4, \quad (4)$$

with (x, y, z) a generic point.

We then define the function $f(\mathbf{p}_1^x, \dots, \mathbf{p}_L^x)$ as follows. Let $(\mathbf{x}_p, \mathbf{y}_p, \mathbf{z}_p)$ be the reference system anchored to the plane to be avoided. Let z_p be aligned with the plane normal and positive in the space region where the object lies (Fig. 5). We call $\mathbf{p}_{i,p}^x = [p_{x_p,i}^x, p_{y_p,i}^x, p_{z_p,i}^x]$ for $i = 1, \dots, L$ the points expressed in the plane reference system $(\mathbf{x}_p, \mathbf{y}_p, \mathbf{z}_p)$. Table avoidance can be achieved by forcing $\bar{\mathbf{p}}_p^x$, the point with the smallest z_p -coordinate in the plane frame, to lie in the region above the plane representing the table.

Thus, the constraint of Eq. (3) can be expressed as follows:

$$a \bar{p}_{x_p}^x + b \bar{p}_{y_p}^x + c \bar{p}_{z_p}^x + d > 0, \quad (5)$$

with $(\bar{p}_{x_p}^x, \bar{p}_{y_p}^x, \bar{p}_{z_p}^x) = \arg \min_{\mathbf{p}_{i,p}^x} \mathbf{p}_{i,p}^x$.

An additional advantage of our formulation is the possibility of imposing specifications on the robot pose, by adding further constraints to the optimization problem. For instance, additional K constraints could be formulated in order to handle preferences on the final pose, by defining suitable $h_i(\mathbf{a}_i, \cdot)$ and $f_i(\mathbf{p}_1^x, \dots, \mathbf{p}_L^x)$ functions and increasing the total number of constraints up to $M + K$.

3) *Lifting Objects:* The theoretical formulation we presented thus far does not take into account dynamic constraints to let the robot actually *lift* the object. As an initial approximation, the robot can physically lift an object if it places its hand in proximity of the object geometric centroid (the object is assumed to have a uniform density). In fact, in case the hand is located on an extremity of the object, as in the examples illustrated in Fig. 6(c) and 6(d),

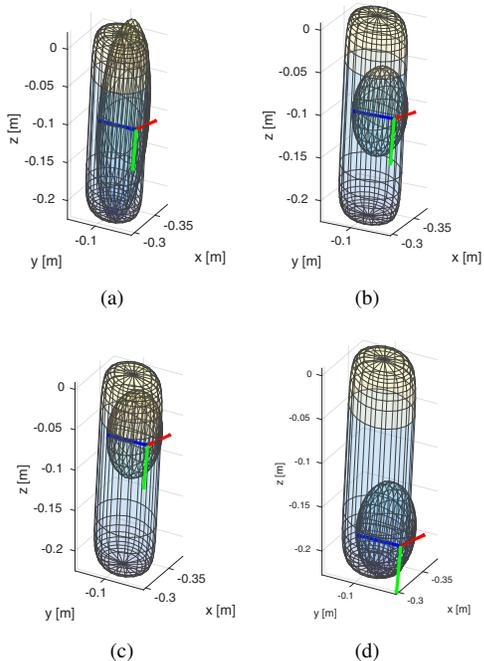


Fig. 6. Stretching the ellipsoid \mathcal{H} so as to amount the longest dimension of the object superquadric \mathcal{O} leads to a grasping pose that eventually enables lifting the object, case (a), without imposing additional constraints in Eq. (3). If a smaller \mathcal{H} was exploited, more solutions were acceptable for the optimization problem, including some poses that do not allow the robot to lift the object properly, such as case (c) and (d).

the probability that the object may fall while it is lifted is large. A possible solution could be adding a constraint to Eq. (3) to require the minimization of the distance between the centroid of \mathcal{O} and the ellipsoid \mathcal{H} . However, instead of a further constraint that might cause the overall execution time to eventually increase, we can alternatively vary the dimensions of \mathcal{H} . Specifically, when the largest dimension of the object superquadric \mathcal{O} (say λ_3) is greater than the corresponding dimension of the ellipsoid \mathcal{H} , (say $\lambda_{h,3}$, thus $\lambda_3 > \lambda_{h,3}$), then we resize \mathcal{H} , so that $\lambda_{h,3} = \lambda_3$. In this way, the dimensions of \mathcal{H} and \mathcal{O} implicitly impose a constraint on the reciprocal position of the centroids of the two entities. A practical proof of the effectiveness of this approach is provided in Fig. 6. If the ellipsoid \mathcal{H} is stretched so as to amount the longest dimension of the object superquadric \mathcal{O} (Fig. 6(a)), the centroid of \mathcal{H} in the computed pose is close to the centroid of \mathcal{O} .

The optimization problem we propose for the computation of proper grasping pose is solved by the Ipopt package efficiently and with execution times compatible with the requirements of real-time applications (see Table III of Section IV). Even if the global solution of a non-linear constrained optimization problem is not generally guaranteed, Ipopt package ensures that a local minimizer is provided to the user, avoiding maximizers and saddle points [34].

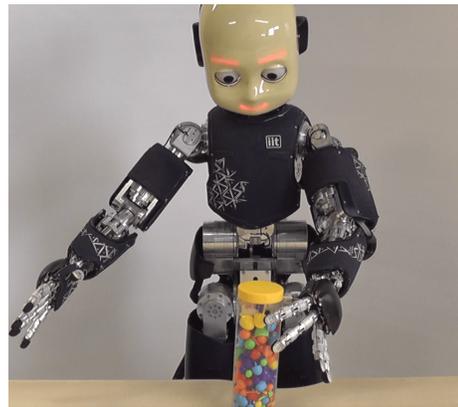


Fig. 7. The iCub humanoid robot has been used as testing platform for testing the proposed approach.

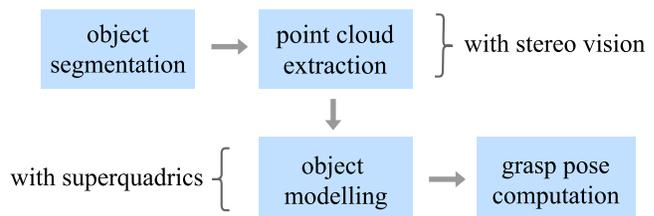


Fig. 8. Object modeling and grasping: complete pipeline. We assume that the object to be grasped is detected using an object recognition system and fixated by the robot. The first two steps of the pipeline involve segmenting the object and constructing a partial 3D point cloud using stereo vision (at this aim we use the cameras mounted on the head of the robot, after calibration). The remaining steps use Eq. (2) to compute the superquadric that better fit the 3D points. The superquadric is finally exploited to compute a reachable pose, by solving the problem in Eq. (3).

Finally, the Cartesian controller available on the iCub [35] is responsible for providing suitable joint trajectories to reach for the grasping pose found by our method.

IV. EXPERIMENTS

In order to validate our approach in a real scenario, we implemented on the iCub humanoid robot (Fig. 7) the pipeline shown in Fig. 8. We assume that the object is identified using an object recognition system. Using stereo vision the object is segmented and a partial 3D point cloud is constructed. Our implementation of superquadric modeling¹ so as grasp pose computation² is publicly available on GitHub.

The dimensions of the ellipsoid we used for representing the volume graspable by the iCub hand are related to the fingers lengths and the palm dimensions. The ellipsoid pose in the hand frame is instead determined by considering the fingers workspace.

We conducted our experiments using a set of 6 objects, shown in Fig. 9. The objects were deliberately selected so as to be different in shape and dimensions (Table I). The number of points L sampled on the ellipsoid \mathcal{H} is chosen to

¹<https://github.com/robotology/superquadric-model>, DOI:10.5281/zenodo.262995.

²<https://github.com/giuliavezzani/superquadric-grasping>, DOI:10.5281/zenodo.263015.

TABLE I
OBJECT DIMENSIONS

Object	Volume [m ³]	Object	Volume [m ³]
Cylinder	$0.06 \times 0.06 \times 0.20$	Ladybug	$0.16 \times 0.08 \times 0.08$
Cat	$0.10 \times 0.10 \times 0.10$	Lettuce	$0.07 \times 0.07 \times 0.07$
Bear	$0.09 \times 0.09 \times 0.12$	Turtle	$0.16 \times 0.10 \times 0.06$

Table I shows the dimensions of the object used in the experiments.

TABLE II
EXECUTION TIME FOR MODEL RECONSTRUCTION

Object	Average time [s]	Object	Average time [s]
Cylinder	1.97	Ladybug	2.39
Cat	1.82	Lettuce	2.22
Bear	2.40	Turtle	2.75

Table II indicates the average execution time across 10 trials for model reconstruction process of each object.

be 46, being this value a good trade-off between algorithm’s performance and speed.

We perform 10 trials for each object (i.e. we ask the robot to grasp each object 10 times by computing a new pose for each trial), in order to evaluate the reliability of our pipeline and the effectiveness of the poses provided by our method. The poses computed in each trial differ because the superquadrics that model the objects vary as a result of variations in the object segmentation and point cloud. In order to reduce such a influence, we implemented the following expedients. First, the least square formulation itself of Eq. (2) makes the model reconstruction approach immune to white noise. Second, we reduce outliers effect on object modeling by pre-filtering the point clouds in order discard points regions with low density. Of course, the use of single-view point clouds leads to rough models whose quality highly depends on the view angle during data acquisition. Fig. 9 shows an example of extracted point clouds and the respective superquadric models \mathcal{O} . Table II indicates the average execution time across 10 trials for model reconstruction. In Table III, we show the percentage of successful grasps, together with statistical information of the execution time required for computing a proper grasp.

The 10 poses computed for each object are compared in Fig. 10, where for the sake of clarity we show only

TABLE III
PERCENTAGE OF SUCCESSFUL GRASPS

Object	No. Trials	Success on Trails [%]	Average Time [s]
Cylinder	10	100%	2.71
Cat	10	85%	1.15
Bear	10	100%	1.70
Ladybug	10	90%	1.89
Lettuce	10	90%	1.72
Turtle	10	85%	1.50

Table III shows the percentage of successful grasps, together with the average execution time required for computing a proper grasp for each object.



Fig. 9. Top: the objects used in the experiments. Bottom: reconstructed point clouds with superimposed superquadric models.

one reconstructed superquadric \mathcal{O} for each object, without showing the (overlapping) ellipsoid \mathcal{H} that represents the hand.

Fig. 10 demonstrates that the desired poses computed with different models of the same object are affected by a small variability, thus guaranteeing a high grade of similarity and therefore underpinning the robustness of our method. For instance, the poses computed for object (a) are all very similar, representing the best grasp for a cylinder shape, that is located on the side at the middle of its height (Fig. 10(a)).

The exploitation of 3D object models in pose computation allows considering even those portions of the object occluded by vision, as illustrated in Fig 10 (a) and (c), where the computed poses bring the hand to touch the objects on the side and thus to place the fingers on the back. This is a remarkable advantage, because using only the visible portion of the object may lead to hand poses that appear not *natural* from the standpoint of human-like reaching or even not easily attainable for a humanoid robot, whose dexterity is often limited compared with that of humans.

Another advantage of our method is that we model the base on which the object is placed (e.g. the table), and impose constraints to avoid it in the optimization problem (the constraint in Eq. (3)). This feature allows the robot to grasp even small objects, without hitting the table with the fingers, as it is the case of the objects in Fig. 10(e) and 10(f) (these objects are approximately 6 [cm] tall).

V. CONCLUSIONS

In this paper, we proposed a novel approach for solving the grasping problem of unknown objects. In short, the idea of our approach is to use superquadrics to model the graspable volume of the hand and the objects from vision. These models are then used to compute a proper grasping pose solving a nonlinear constrained optimization problem. We showed how to add constraints so that the set of possible poses is limited to those that do not cause collisions with obstacles (e.g. the base on which the object stands) and do not lead to object penetration. Our approach is sufficiently generic to deal with objects and obstacles of different shape

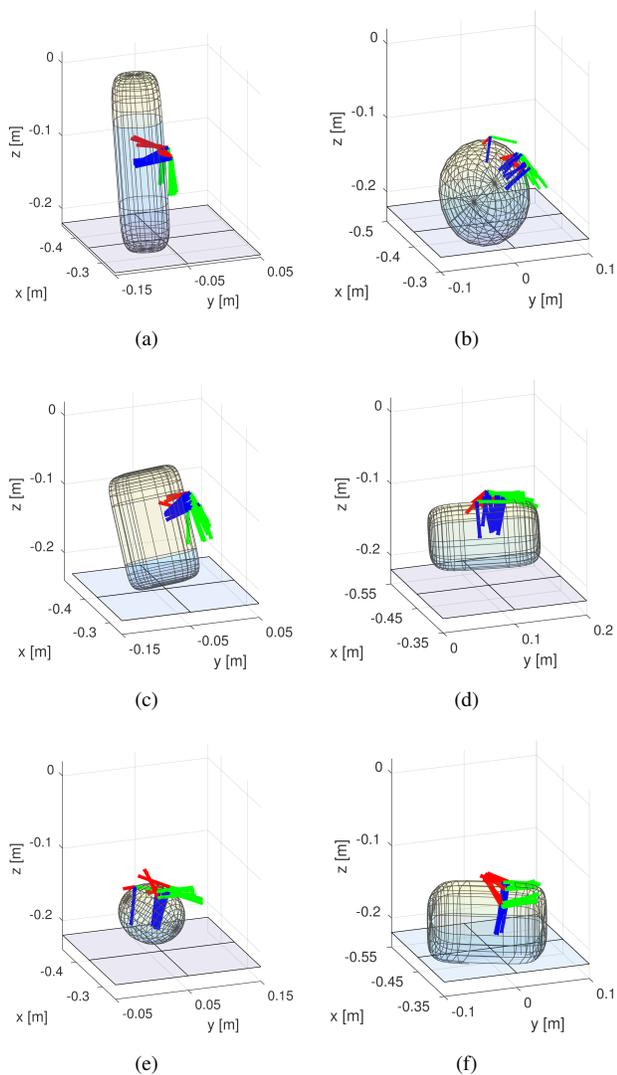


Fig. 10. For each object 10 poses are shown. The letters identifying the different plots ((a) - (f)) correspond to different objects, according to the notation of Fig. 9.

and size, and enables specifying further requirements on the robot pose by adding new constraints.

We evaluated our method experimentally with the iCub humanoid robot, showing that it is reliable, providing a percentage average on 6 objects greater than 90%. In addition our tests demonstrate that the robot can grasp even small objects, without hitting the supporting table with the fingers. Finally, an interesting advantage of our algorithm is that it can compute poses in which fingers are located on object portions that are occluded from vision, if they have better grasping properties.

The work in this paper can be extended in several ways. First of all, it is possible to deal with a larger number of objects with more complex shapes by introducing more accurate models. For this reason we will perform an extensive evaluation of our method on the YCB Object & Model Set [36] and compare the reconstructed superquadric models with the 3D models provided by the dataset. In

this paper we used a single superquadric which provides a rough approximation of the object shape. However, the object model can be refined by using a set of superquadrics [27], [28]. Similarly, the hand model can be improved, for example by taking into account finger shape and position. A viable approach is to approximate them with a set of superquadrics and impose object avoidance constraints for each finger. Finally, the problem formulation could be also extended to compute, in addition to the grasping pose, an appropriate approach trajectory that satisfies additional optimality criteria (such as length of the trajectory, obstacle avoidance).

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] Y. Hu, R. Eagleson, and M. A. Goodale, "Human visual servoing for reaching and grasping: The role of 3-d geometric feature," in *16th IEEE International Conference on Robotics and Automation (ICRA)*, vol. 4, 1999, pp. 3209 – 3216.
- [3] G. Rizzolatti and G. Luppino, "The cortical motor system," *Neuron*, vol. 31, no. 6, pp. 889 – 901, 2001.
- [4] I. Gori, U. Pattacini, V. Tikhonoff, and G. Metta, "Ranking the good points: a comprehensive method for humanoid robots to grasp unknown objects," in *16th International Conference on Advanced Robotics (ICAR)*, 2013.
- [5] A. Wächter and L. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical programming*, 2006.
- [6] 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.
- [7] J. Napier, "The prehensile movements of the human hand," *he Journal of bone and joint surgery*, pp. 902 – 913, 1956.
- [8] I. Gori, U. Pattacini, V. Tikhonoff, and G. Metta, "Three-finger precision grasp on incomplete 3d point clouds," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014.
- [9] A. Sahbani and S. El-Khoury, "An overview of 3d object grasp synthesis algorithms," *Robotics and Autonomous Systems*, vol. 60, no. 3, pp. 326 – 336, 2011.
- [10] R. Pelossof, A. Miller, P. Allen, and T. Jebara, "An svm learning approach to robotic grasping," in *21th IEEE International Conference on Robotics and Automation (ICRA)*, vol. 4, 2004, pp. 3512 – 3518.
- [11] A. Maldonado, U. Klank, and M. Beetz, "Robotic grasping of unmodeled objects using time-of-flight range data and finger torque information," in *23rd International Conference on Intelligent Robots and Systems (IROS)*, 2010, pp. 2586 – 2591.
- [12] S. Geidenstam, K. Huebner, D. Banksell, and D. Kragic, "Learning of 2d grasping strategies from box-based 3d object approximations," in *Robotics: Science and Systems Conference*, 2009.
- [13] L. Bodenhausen, D. Kraft, and M. Popovi'c, "Learning to grasp unknown objects based on 3d edge information," in *International Symposium on Computational Intelligence in Robotics and Automation*, 2009.
- [14] A. Saxena, L. Wong, and A. Ng, "Learning grasp strategies with partial shape information," in *AAAI Conference on Artificial Intelligence*, 2008.
- [15] 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.
- [16] R. Detry, C. Ek, M. Madry, J. Piater, and D. Kragic, "Generalizing grasps across partly similar objects," in *IEEE International Conference on Robotics and Automation*, 2012.
- [17] A. Miller, S. Knoop, H. Christensen, and P. Allen, "Automatic grasp planning using shape primitives," in *IEEE International Conference on Robotics and Automation*, 2013.

- [18] M. Popovic, D. Kraft, L. Bodenhagen, E. Baseski, N. Pugeault, D. Kragic, T. Asfour, and N. Kruger, "A strategy for grasping unknown objects based on co-planarity and colour information," in *Robotics and Autonomous Systems*, 2010.
- [19] A. ten Pas and R. Platt, "Localizing handle-like grasp affordances in 3d point clouds," in *2014 International Symposium on Experimental Robotics (ISER)*, May 2014.
- [20] 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.
- [21] 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.
- [22] S. El Khoury, M. Li, and A. Billard, "Bridging the gap: One shot grasp synthesis approach," in *IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2012.
- [23] A. H. Barr, "Superquadrics and angle preserving transformations," *IEEE Computer Graphics and Applications*, vol. 1, no. 1, pp. 11 – 23, 1981.
- [24] A. H. Barr, "Global and local deformations of solid primitives," *Computer Graphics*, vol. 18, no. 3, pp. 21 – 30, 1981.
- [25] A. Jaklic, A. Leonardis, and F. Solina, "Segmentation and recovery of superquadrics," *Computational imaging and vision*, vol. 20, 2000.
- [26] A. J. Hanson, "Hyperquadrics: smoothly deformable shapes with convex polyhedral bounds," *Computer Vision, Graphics, and Image Processing*, vol. 44, no. 2, pp. 191 – 210, 1988.
- [27] L. Chevalier, F. Jaillet, and A. Baskurt, "Segmentation and superquadric modeling of 3d objects," 2003.
- [28] 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 on Robotics and Automation (ICRA)*, 2013, pp. 4238 – 4243.
- [29] N. Chakraborty, J. Peng, S. Akella, and J. E. Mitchell, "Proximity queries between convex objects: an interior point approach for implicit surfaces," *IEEE Transactions on Robotics*, vol. 24, no. 1, pp. 211– 220, 2008.
- [30] C. Goldfeder, P. K. Allen, C. Lackner, and R. Pelossof, "Grasp planning via decomposition trees," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2007, pp. 4679 – 4684.
- [31] T. T. Cocia, S. M. Grigorescu, and F. Moldoveanu, "Multiple-superquadrics based object surface estimation for grasping in service robotics," in *13th International Conference on Optimization of Electrical and Electronic Equipment (OPTIM)*, 2012, pp. 1471 – 1477.
- [32] A. Miller and P. K. Allen, "Graspit!: A versatile simulator for robotic grasping," *IEEE Robotics and Automation Magazine*, vol. 11, no. 4, pp. 110– 122, 2004.
- [33] A. H. Quispe, B. Milville, M. A. Gutierrez, C. Erdogan, M. Stilman, H. Christensen, and H. B. Amor, "Exploiting symmetries and extrusions for grasping household objects," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, May 2015, pp. 3702– 3708.
- [34] A. Wächter, "Short tutorial: Getting started with ipopt in 90 minutes." [Online]. Available: <https://projects.coin-or.org/CoinBinary/export/837/CoinAll/trunk/Installer/files/doc/Short%20tutorial%20Ipopt.pdf>
- [35] "The YARP cartesian interface," *description available at Cartesian Interface*.
- [36] B. Calli, A. Walsman, A. Singh, S. Srinivasa, P. Abbeel, and A. M. Dollar, "Benchmarking in manipulation research: The ycb object and model set and benchmarking protocols," *preprint available at arXiv:1502.03143*, 2015.