Grasping Unknown Objects in Clutter by Superquadric Representation

Abhijit Makhal*
Department of Mechanical Engineering
Idaho State University
Pocatello, Idaho 83209
Email: makhabhi@isu.edu

Federico Thomas
Institut de Robòtica i Informàtica
industrial: IRI (CSIC/UPC)
Barcelona, Spain 08028
Email: fthomas@iri.upc.edu

Alba Perez Gracia
Department of Mechanical Engineering
Idaho State University
Pocatello, Idaho 83209
Email: perealba@isu.edu

***Abstract*—In this paper, a quick and efficient method is presented for grasping unknown objects in clutter. The grasping method relies on real-time superquadric (SQ) representation of partial view objects and incomplete object modelling, well suited for unknown symmetric objects in cluttered scenarios which is followed by optimized antipodal grasping. The incomplete object models are processed through a mirroring algorithm that assumes symmetry to first create an approximate complete model and then fit for SQ representation.**

The grasping algorithm is designed for maximum force balance and stability, taking advantage of the quick retrieval of dimension and surface curvature information from the SQ parameters. The pose of the SQs with respect to the direction of gravity is calculated and used together with the parameters of the SQs and specification of the gripper, to select the best direction of approach and contact points. The SQ fitting method has been tested on custom datasets containing objects in isolation as well as in clutter. The grasping algorithm is evaluated on a PR2 and real time results are presented. Initial results indicate that though the method is based on simplistic shape information, it outperforms other learning based grasping algorithms that also work in clutter in terms of time-efficiency and accuracy.

I. INTRODUCTION

Accurate grasping of possibly unknown objects is one of the main needs for robotic systems in unstructured environments. In order to do so, robotic systems have to go through multiple costly calculations. First, the object of interest needs to be identified, which includes a segmentation step resulting in a suitable surface impression of the object. In the next stage, the precise position of the object in the world should be determined. Then, grasps needs to be synthesized by calculating the set of points on the object where the fingers can be placed followed by determining preferred approach directions. Finally, the system needs to plan a safe collision-free path of the manipulator towards the object.

Typical robotic grasping methods focus on optimization of stable grasp metrics. One common approach is to maintain a database of objects with their corresponding preferred grasps. When the system encounters an object that can be identified with an entry from the database, the corresponding suitable referred grasp is pulled out. In this case, the system has to explore and evaluate an increasingly large number of objects and grasps for a given scenario. One of the limitations of such kinds of systems is that the database has to be quite large in order to contain all of the common objects that a robot may encounter. If the system has to work in less structured environments, observing a novel object may be common. A methodology for grasping unknown objects is preferred here, rather than relying on the database.

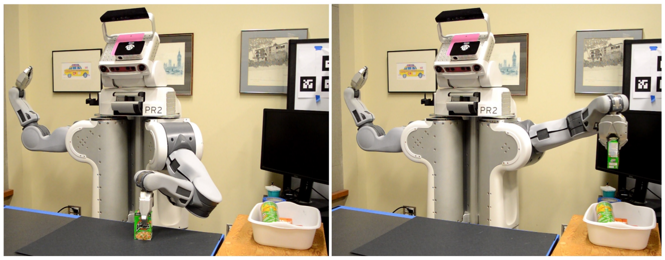

Fig. 1. Experimental setup.a) The system has to perceive the object on the table, model it by fitting superquadric and execute suitable grasp to pick up the object and b) drop it in the bin to the right of the image

The rest of the paper is organized as follows: Section II presents a brief summary of previous work performed for grasping known and unknown objects. Section III explains the methodology for representation of Superquadrics (SQ). Completion of object models from perceived point cloud is explained in section IV. Section V describes the process of fitting SQ on completed point clouds, while finding antipodal grasps on those SQ is presented in section VI. The experimental results of fitting SQ on point cloud in real time for objects in isolation and clutter with an evaluation with other methods are presented in section VII. We conclude with the limitations and bottlenecks of the method in section VIII.

II. RELATED WORK

A considerable amount of work exists in the area of finding feasible grasping points on novel objects using vision based systems. Initial efforts using 2D images to find grasping points in presented in [1]. The arrival of economic RGB-D cameras allowed interpreting objects in 3D, which led to object identification as point clouds and derived parameters. Different models have been utilized for this task, such as representing objects by implicit polynomial and algebric invariants [2], spherical harmonics [3] [4], geons [5], generalized cylinders

[6], symmetry-seeking models [7] [8], blob models [9], union of balls [10] and hyperquadrics [11], to name a few.

The representation of objects as SQ is found in [12] and later in [13]. The work in [14] shows fast object representation with SQ for household environments. An approach infast and efficient pose recovery of objects using SQ has been presented in [15]. In the work of [16] the authors try to fit SQ to piled objects.

Recent literature suggests that using a single-view point cloud to fit (SQ) can lead to erroneous shape and pose estimation. In general, the 3D sensors are noisy and obtain only a partial view of the object, from a single viewpoint. In order to obtain a full model of the object, several strategies have been developed. In [17], the shape of the object is completed from the partial view and a mesh of the completed model is generated for grasping. Generating a full model from a partial model can be approached in several ways, mainly symmetry detection [8] and symmetry plane [18] [19] [20]. Extrusion-based object completion has also been proposed in [21]. [22] presents a different strategy of changing the viewpoint in a controlled manner and registering all the partial views to create a complete model. This technique provides good results, yet it is not very suitable for real time systems and is prone to errors due to registration of several views. In addition, results are not satisfactory when the working environment becomes densely cluttered.

The calculation of feasible grasping points on a point cloud or a mesh is by nature iterative, hence computationally expensive. In [23], a large set of grasps is generated directly from the point cloud and evaluated using convolutional neural networks, obtaining good grasp success results. Such methods avoid the need for robust segmentation but cannot assure the assignment of the grasp to a target object. The method, denoted Grasp Pose Detection (GPD), has been combined with object pose detection in [24]. A similar approach uses Height Accumulated Features (HAF) [25], where local topographical information from the point cloud is retrieved to calculate antipodal grasps.

A different set of methods use the fitting of object models to the point clouds to generate smaller or simpler sets of grasp points. In [26], the calculation of grasp regions is combined with path planning. Curvature-based grasping using antipodal points for differentiable curves, in both convex and concave segments, was studied in [27], while in [28], a grasping energy function is used to calculate antipodal grasping points using local modelling of the surface.

A third set of methods relies on the recognition of objects and comparison to a database of objects with optimized grasping points already included as features of the object [20] [29].

The main contributions of the paper are as follows, 1) a novel method is presented to mirror the partial view point cloud obtaining an approximated full model of the object, 2) SQ fitting on the point cloud as part of the online process which is robust to change in the environment and 3) a novel shape based grasping algorithm which can obtain antipodal grasping points for frictional two-fingered grippers satisfying properties of maximum force balance and stability in a time efficient manner.

III. Superquadric Representation

A SQ can be defined as a generalized quadric in which the exponents of the implicit representation of the surface are arbitrary real numbers, allowing for a more flexible set of shapes while keeping the symmetric characteristics of the regular quadric.

The superellipsoid belongs to the family of SQ; other members of the family are the superhyperboloid and the supertoroid. A SQ can be obtained as the spherical product of two superellipses [12], S_1 and S_2, to obtain the parametric equation

$$r(\eta,\omega) = S_1(\eta) \oplus S_2(\omega) = \left(\begin{array}{c} a_1 \cos^{\epsilon_1}\eta \cos^{\epsilon_2}\omega \\ a_2 \cos^{\epsilon_1}\eta \sin^{\epsilon_2}\omega \\ a_3 \sin^{\epsilon_1}\eta \end{array} \right), \quad (1)$$

with $\eta \in [\frac{-\pi}{2}, \frac{\pi}{2}]$ and $\omega \in [-\pi, \pi]$, where a_1, a_2, a_3 are the scaling factors of the three principal axes. The exponent ϵ_1 controls the shape of the SQ's cross-section in the planes orthogonal to (x, y) plane, and ϵ_2 controls the shape of the SQ's cross-section parallel to the (x, y) plane. The pose of the SQ with respect to a world frame is specified by the six parameters that define a rigid motion, p_x, p_y, p_z for the position vector and ρ, ψ, θ for defining a rotation matrix, for instance using roll, pitch, and yaw angles. The total set of parameters that fully defines the SQ consists of 11 variables, $\{a_1, a_2, a_3, \epsilon_1, \epsilon_2, p_x, p_y, p_z, \rho, \psi, \theta\}$.

The SQ can also be expressed using an implicit equation in normal form as

$$f(a, x, y, z): \quad \left(\left|\frac{x}{a_1}\right|^{\frac{2}{\epsilon_2}} + \left|\frac{y}{a_2}\right|^{\frac{2}{\epsilon_2}} \right)^{\frac{\epsilon_2}{\epsilon_1}} + \left|\frac{z}{a_3}\right|^{\frac{2}{\epsilon_1}} = 1. \quad (2)$$

It is easy to show that the SQ is bounded by the planes given by $-a_1 \leq x \leq a_1$, $-a_2 \leq y \leq a_2$, and $-a_3 \leq z \leq a_3$.

In [30], a method is proposed for uniform spatial sampling of points on SQ models. The arc length between two close points on $x(\theta)$ and $x(\theta + \Delta_\theta\theta)$ on a curve can be estimated by Euclidean distance,

$$D(\theta) = |x(\theta) - x(\theta + \Delta_\theta\theta))| \quad (3)$$

where we can use a first-order approximation for $\Delta_\theta\theta$,

$$\Delta_\theta\theta = \frac{D(\theta)}{\epsilon} * \sqrt{\frac{\cos(\theta)^2\sin(\theta)^2}{a_1^2\cos(\theta)^{2\epsilon}\sin(\theta)^4 + a_2^2\sin(\theta)^{2\epsilon}\cos(\theta)^4}} \quad (4)$$

By setting $D(\theta)$ to a constant sampling rate, the incremental updates of the angular parameters are

$$\theta_i = \begin{cases} \theta_{i-1} + \Delta_\theta(\theta_i), & \theta_0 = 0, i \in \{1....N\}, \theta_N < \frac{\pi}{2}, \\ \theta_{i-1} - \Delta_\theta(\theta_i), & \theta_0 = \frac{\pi}{2}, i \in \{1....N\}, \theta_N > 0, \end{cases} \quad (5)$$

where

$$\Delta_\theta \theta = \left(\frac{D(\theta)}{a_2} - \theta^\epsilon \right)^{\frac{1}{\epsilon}} - \theta \tag{6}$$

for $\theta \to 0$, and

$$\Delta_\theta \theta = \left(\frac{D(\theta)}{a_1} - (\frac{\pi}{2} - \theta) \right)^{\frac{1}{\epsilon}} - (\frac{\pi}{2} - \theta) \tag{7}$$

for $\theta \to \frac{\pi}{2}$.

This approach has been implemented for the SQ sampling in our work.

IV. Partial-view Model Completion

In this section, the procedure for completion of object models from the partial view point cloud is described. A typical point cloud view from a point cloud capturing device is shown in Fig. 2a, where only a part of the object is captured. The other parts of the object are on the occluded side, so they can only be captured by a device situated on the other side of the table. Reconstruction of the full object model can then be performed by registering the two views from two individual devices. As most of the objects in our daily lives are symmetric by nature, by the law of symmetry it can be asserted that the occluded side of the object is merely a reflection of the visible part. By using only this assumption, the mirroring algorithm presented here creates the occluded part of the point cloud and reconstructs the object model online.

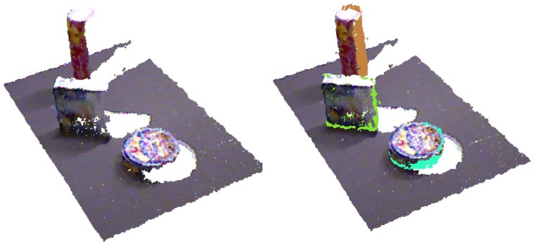

Fig. 2. Point cloud mirroring on individual objects of the scene. From left to right a) point cloud of three objects on the table (the depth cloud capturing device is at the left of the image), b) the point cloud is mirrored and the approximated complete model is constructed. The mirrored points for different objects are shown in distinct colors.

The input to the system is an organized point cloud captured from a single view 3D point cloud capturing camera. The scenes can be captured by an RGB-D camera such as the Kinect, time-of-flight cameras, or stereo cameras. In our experiments, a single Kinect camera is used. The process is divided into three steps, a first segmentation step, second pose estimation step, and finally a mirroring step.

A. Segmentation

The initial scene contains objects situated on a table plane. After the dominant table plane is removed, we assume that the clusters present on the table plane correspond to objects. The Voxel Cloud Connectivity Segmentation (VCCS) [31] algorithm is used to separate the clusters. The 3D point clouds can be over-segmented into surface patches, which are considered as supervoxels. The algorithm utilizes a local-region growing variant of K-Means Clustering to generate individual supervoxels, represented as $\bar{p}_i = (\bar{x}_i, \bar{n}_i, N_i)$, where $\bar{x}_i$ is the centroid, $\bar{n}_i$ is the normal vector and $e \in N_i$ are the edges to adjacent supervoxels. The connections of supervoxels are checked for convexity and those belonging to same connectivity graph are segmented into individual clusters. The results of the segmented point clouds are visualized in Fig. 3c.

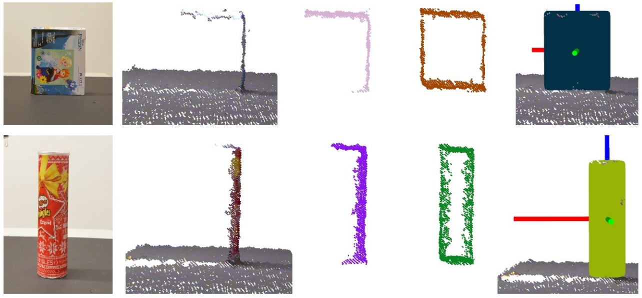

Fig. 3. The complete mirroring pipeline. From top to bottom, the mirroring process with a box and a cylinder object (The capturing device can be assumed at the right side of the image): a) RGB side view of the object, b) captured point cloud of the scene, c) segmented point cloud of the object, d) mirrored point cloud with the segmented cloud, e) SQ representation of the object.

B. Pose estimation

Estimating correct pose is one of the most important characteristics of any grasping scenario. The most common approach to estimate pose from point clouds where ground truth orientation of the object model is not known is by using Principal Component Analysis (PCA) for orientation and centroid information for positions. The dominating eigen vectors from PCA are considered as the orientation of point clouds. But this approach utilizes the point cloud on the surface of the object. While the centroid from these methods may be the center of the point cloud, it is not the centroid of the object. Also, to deal with noisy point clouds from real sensors, the axes cannot be determined from PCA, as for every iteration, the number of points perceived varies rapidly.

Algorithm 1 Generate Mirrored Cloud

```
input (organized point cloud $Cloud$)
procedure $CompleteCloud$
Segment out the Table
Segment Objects
    for each $i$ in $Objects$ do
        Find pose $(c_x, c_y, c_z, \rho, \psi, \theta)$
        Generate local frame at $c_x, c_y, c_z$
        for each point $p$ in $i$ do
            Calculate Euclidean distance $(c, p)$
            Negate distance and find point $m$
            Add $m$ to $i$
        end for
    end for
end procedure
```

The center of the object (c_x, c_y, c_z) can be estimated by taking the mean of the values of the points in all three dimensions. For completely unknown objects, the ground truth orientation is also unknown, so an assumption is asserted that the orientation of all the objects perceived are z-axis upwards. The problem of obtaining pose information (ρ, ψ, θ) is now limited to finding only the θ which control the rotation in z-axis. The volume V_{cloud} can be calculated by maximum distances in each dimension(d_x, d_y, d_z), where the maximum distance is the difference between the maximum value and the minimum value in each plane. All the points in the point cloud are projected onto the table plane and a 2D convex hull is created around the projected cloud. The volume of a bounding box can also be estimated from length l and width w of the convex hull and h as the value of p_{Zmax}. By rotating the convex hull in small iterations and minimizing error (ϵ) between the volume of the bounding box and the volume of the point cloud, the rotation θ in z-axis can be obtained.

$$\epsilon = min(V_{cloud} - [l * d * h]) \tag{8}$$

C. Creating a mirrored surface

To obtain a complete approximated model of the perceived object, the points perceived are mirrored based on the pose $(c_x, c_y, c_z, 0, 0, \theta)$ found in the previous step. A local reference frame is assumed at the center of the object and the euclidean distance of all the points from the centroid is calculated. All the points are reflected on the occluded side by scaling all of them by the negative value of 1 unit to the distance. So for every single point P_i (p_x, p_y, p_z), a mirrored point M_i $(-p_x, -p_y, -p_z)$ is created where p_x, p_y, p_z are distances in x, y, z directions from the center of the object. Though the pose does not contain any ρ and ψ values, this mirroring method can work on any arbitrary orientation of the object. To deal with noisy data, all the point clouds are processed through a series of outlier removal algorithms.

As shown in Fig. 2b, the objects are captured from a point of view (POV) where only a part of the object can be seen. After removing the points belonging to the table surface and segmenting out the object of interest from the environment, the object point clouds are depicted in Fig. 2c. The mirrored point cloud with object points are shown in Fig. 2d. It is worth noticeable that the pose of the box in Fig. 2 is such that only the smallest part of the box can be perceived, but the algorithm still creates the approximate model of the box. A scene with mirrored point clouds are shown in Fig. 2, where the points with distinct colors are mirrored from the object point cloud by the mirroring algorithm presented in Algorithm 1.

V. Superquadric Fitting

To fit the SQ model to point cloud data, distance from the points to the function $f(a, x, y, z)$ must be minimized. Radial distance $|OP|$, which is the distance between the points and the center of the SQ [13], is used instead of Euclidean distance,

$$d(a, x, y, z) = \min \sqrt{a_1 a_2 a_3 \sum_{k=0}^{n} ||OP|| * (f(a, x, y, z)^{\epsilon} - 1)} \tag{9}$$

VI. Superquadric Grasp Synthesis

This work focuses on the use of two-fingered grippers for grasping the objects. It is well known that antipodal grasps can be successful in both convex and concave objects in the presence of friction. Finding antipodal points on generally-shaped object models usually returns many candidates, which are ranked according to some metric. In order to make the grasp more stable and balanced, we impose two conditions: a) grasping closer to the centroid of the object and b) at the points of minimum curvature. In addition, we have constraints given by the depth and width of the robotic gripper and the task-based direction of approach.

A. Minimum curvature contact points

The gripper contact points for antipodal grasping with minimum curvature can be calculated from the SQ parameters to be

$$p_W = [T_{WK}][T_{KS_i}](\pm \begin{Bmatrix} p_a \\ 1 \end{Bmatrix}), \tag{10}$$

where $[T_{WK}]$ is the 4×4 homogeneous transformation from the world frame to the camera frame, $[T_{KS_i}$ is the 4×4 homogeneous transformation from the camera frame to the principal frame of SQ i, and p_a is the position vector of the contact points in the SQ frame, to be selected according to the criteria explained below and summarized in Table I.

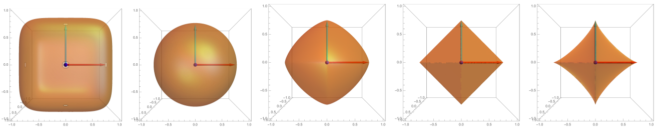

Fig. 4. Convex SQ for $\epsilon = 0.4$, $\epsilon = 1$, $\epsilon = 1.4$ and flat for $\epsilon = 2$; concave SQ for $\epsilon = 2.4$.

The SQ is convex for values of the exponent $0 < \epsilon < 2$; for larger values, it becomes concave at the corresponding cross section, as shown in Fig.4.

Curvature in SQ can be easily calculated from either their implicit or their parametric expression. Considering the implicit expression in eq (1) at the $x - z$ plane obtained when $\omega = 0$, the curvature is

$$k(\eta) = \frac{(\epsilon_1 - 2) a_1 a_3 \cos^{\epsilon_1 - 4} \eta \sin^{\epsilon_1 - 4} \eta}{\epsilon_1 \sqrt{(a_1^2 \cos^{2\epsilon_1 - 4} \eta + a_3^2 \sin^{2\epsilon_1 - 4} \eta)^3}}, \tag{11}$$

and similarly for the other two planes passing through the origin of the SQ's frame. Depending on the value of the exponent ϵ_1, the minimum curvature will be found at the intersection with the axes or at 45^o from them. Fig.5 shows the effect of the exponent on the location of the minimum curvature; the curvature is constant for $\epsilon_1 = 1$.

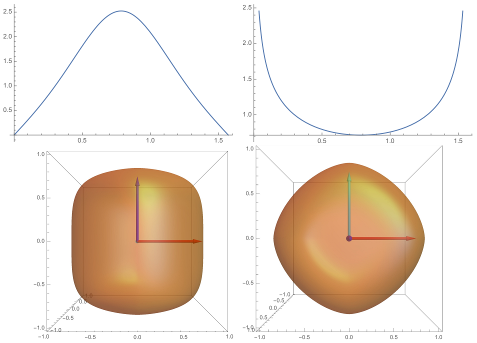

Fig. 5. From top to bottm: a) curvature as a function of the angular parameter and the value of the exponent ϵ in curvature plot, b) cross-sectional cut of the SQ for same semi-axes length and values $\epsilon = 0.5$ and $\epsilon = 1.2$.

For substantially different lengths of the semi-axes of the convex SQ, the points of minimum curvature at $\epsilon > 1$ are not antipodal. Here, normal directions to the SQ are found with minimum angular deviation. Disregarding its sign, the normal to the SQ can be calculated for a coordinate plane passing through the origin as the second derivative of the parametrized curve. For $x - z$ plane at $\omega = 0$ normal n is

$$n(\eta) = \left\{ \begin{matrix} \frac{a_3 \cos^{\epsilon_1+2} \eta \sin^{2\epsilon_1} \eta}{\sqrt{a_3^2 \cos^{2\epsilon_1+4} \eta \sin^{4\epsilon_1} \eta + a_1^2 \cos^{4\epsilon_1} \eta \sin^{2\epsilon_1+4} \eta}} \\ \frac{a_1 \cos^{2\epsilon_1} \eta \sin^{\epsilon_1+2} \eta}{\sqrt{a_3^2 \cos^{2\epsilon_1+4} \eta \sin^{4\epsilon_1} \eta + a_1^2 \cos^{4\epsilon_1} \eta \sin^{2\epsilon_1+4} \eta}} \end{matrix} \right\} \tag{12}$$

Fig. 6 shows an example of the antipodal points with minimum curvature on the $x - z$ plane; if a solution exists, then there will always be another symmetric solution. In addition, the contact points at the major axes are antipodal, but depending on the value of the exponent, they may present higher curvature or a cusp for extreme cases.

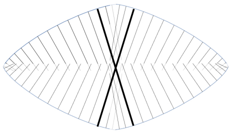

Fig. 6. Normals to the surface of the SQ for $a_1 = 0.5$, $a_3 = 0.9$ and $\epsilon_1 = 1.5$. Normal lines of approximate antipodal points marked with thicker lines

The angular parameter of the approximate antipodal points in the plane can be found by imposing that the normal line passes close enough to the origin,

$$p_a = \min_{\eta}\left(\frac{a_1 a_3 \cos^{2\epsilon_1} \eta \sin^{2\epsilon_1} \eta}{\sqrt{a_3^2 \cos^{2\epsilon_1+4} \eta \sin^{4\epsilon_1} \eta + a_1^2 \cos^{4\epsilon_1} \eta \sin^{2\epsilon_1+4} \eta}} - 1\right). \tag{13}$$

For concave SQs, the curvature helps make the grasp closed. Antipodal points are sought using eq (13).

B. Direction of approach

Let us define the plane of grasping as the plane containing the contact points and normal to the fingers of the gripper.

TABLE I
POSITION VECTOR TO CONTACT POINTS AS A FUNCTION OF THE LENGTH OF THE SEMI-AXIS AND THE EXPONENT ϵ.

Exponent	Dimensions	Position vector
$\epsilon < 1$	$a_1 > a_2$ $a_1 = a_2$ $a_1 < a_2$	$p_a = (0, a_2, 0, 1)$ $p_a = (a_1, 0, 0, 1)$ or $p_a = (0, a_2, 0, 1)$ $p_a = (a_1, 0, 0, 1)$
$\epsilon = 1$	$a_1 > a_2$ $a_1 = a_2$ $a_1 < a_2$	$p_a = (0, a_2, 0, 1)$ $p_a = (a_1 \cos\omega, a_1 \sin\omega, 0, 1)$ for all ω $p_a = (a_1, 0, 0, 1)$
$\epsilon > 1$	$a_1 = a_2$ $a_1 \neq a_2$	$p_a = (a_1/\sqrt{2}, a_1/\sqrt{2}, 0, 1)$ $p_a(\omega)$ as per Eq.(13)

The direction of approach of the gripper is then perpendicular to this plane. Because of the condition of entered grasp, at least one of the dimensions of the superellipsoid at the plane of grasping, a_i, needs to be smaller than the width of the gripper, w. In addition, the dimension of the major axis in the direction perpendicular to that plane needs to be smaller than the depth of the gripper, h.

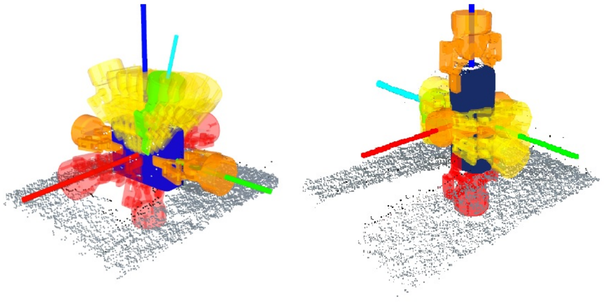

Fig. 7. Grasps on the (SQ). From left to right, a) on a box, b) on a cylinder. The red grasps are invalid, yellow grasps are potential grasp on different axes, light yellow grasps are potential grasp on current axes and green grasp in the best grasp. The approach direction is shown by cyan bar.

The plane of grasping to consider can be at an angle from the coordinate planes in order to accommodate verticality or the above conditions. As before, the direction of approach is calculating as

$$s_W = [T_{WK}][T_{KS_i}](\pm \begin{Bmatrix} s_a \\ 0 \end{Bmatrix}), \tag{14}$$

where $s_a = x, y$, or z or a coordinate rotation about one of those axes.

Algorithm 2 shows the summary of the process to select the plane of grasping and the contact points. It is intuitive that the algorithm always tries to approach object from the minimum $\pm a_i$ direction. It shown in Fig. 8, where the first and last images shows grasping of the box from the minimum $\pm a_i$ direction.

VII. Experimental Results

This section discusses the results of the replication and SQ fitting method, and the results from the grasping algorithm. Our dataset consists of simple household items of various shapes: balls, cubes, paper rolls etc. For experiments, the

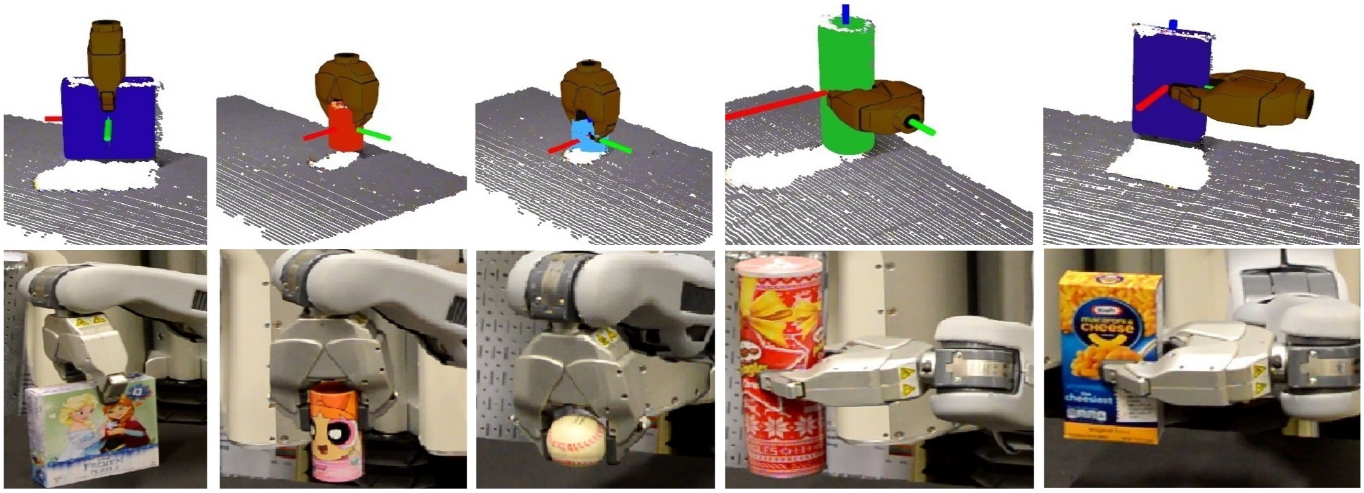

Fig. 8. Experiments with the PR2 robot with 5 different objects. Top: Visualization of the gripper on the obtained SQ before executing. Bottom: Grasp execution on objects in real time

Algorithm 2 Grasp synthesis

input (SQ parameters, depth of the gripper h, width of the gripper w)

```
procedure $SelectContactPlanes$
    for $s_a$ SQ direction do Calculate $s_W$
        if angle($s_W$,$Z_W$) $> \alpha$ or length($s_a$) $> h$ then
            Discard $s_a$
        end if
    end for
    for remaining $s_a$ do
        if $2 * a_i > w$ for perpendicular directions then
            Discard $s_a$
        end if
    end for
end procedure
procedure $CalculateContactPoints$
    Select $p_a$ (Table I)
    Calculate $p_W$
end procedure
```

approach direction s_W in negative z direction is always discarded, as by our pose estimation process, we assumed that the pose is z-axis upwards. So the object can not be grasped through the table. Fig. 7 shows the visualization of potential grasps on a box and a cylinder SQ. The red grasps are easily discarded as the object can not be grasped from those directions. The dark yellow grasps are from those directions from where the object can be grasped. By incorporating motion planning and robot reachability with grasps, we decide on the grasp to be executed. The first grasp is at the $min(a_i)$ direction. If the motion planner fails to reach the main grasp at direction $min(a_i)$, we sample the grasps with varying angles at the same direction (yellow grasps in Fig.7). If all the grasps at this direction fails, we move to second $min(a_i)$ direction.

We have performed the grasping experiments on a PR2 robot that uses a Kinect 1 to perceive objects. A grasp is considered to be successful if the object is been picked up from the table and placed in a bin situated on the other side of the setup (refer to Fig. 1 for the locations of the objects and bin). To merge the fitting process with grasping as an online system, only the five parameters $\{a_1,\ a_2,\ a_3,\ \epsilon_1,\ \epsilon_2\}$ are optimized by Levenberg Marquardt[32] algorithm implemented by Ceras library. The other pose parameters $\{p_x,\ p_y,\ p_z,\ \rho,\ \psi,\ \theta\ \}$ are optimized separately by the process described above in the pose estimation section of mirroring algorithm. While the range of ϵ_1 and ϵ_2 is constrained between the range of 0.1 and 1.9, the initial values of a_1, a_2 and a_3 are eigenvalues of the covariance matrix of the point cloud. A cartesian planner has been implemented for the system to reach the grasp by defined approach and pick up the object. For the placing task, a joint controller is implemented.

A. *Superquadric fitting*

Our object dataset mostly consists of small symmetric objects, such as small boxes, cylinders and balls with varied parameters in x, y, z directions. For calculating the errors, in an offline scenario, we obtained the ground truth $(a1, a2, a3)$ parameters of all the objects. The ϵ_1 and ϵ_2 are obtained by iterating over the values between 0.1 and 1.9 and choosing the volume that best fits the object. The error is radial distance error between the point cloud generated by ground truth information and the point cloud generated by obtained SQ parameters. For every object, 5 different poses are defined on the table and the objects are perceived from all 5 poses. The results shown in II are a comparison of recovered parameters between our method and [17]. As the result suggests, our method shows better improved results is SQ fitting in terms of time-efficiency and accuracy, as our method determines the pose on the approximated center of the object. Also, instead of optimizing 11 parameters of the object, we only optimize 5 parameters related to object shape. While the size of the

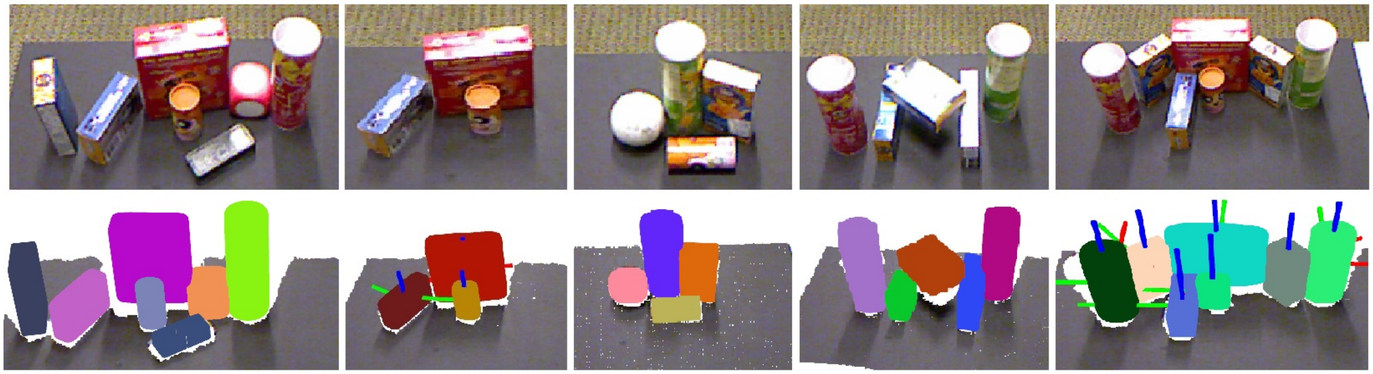

Fig. 9. SQ fitting in cluttered scenario. The top images show the RGB view of the image perceived by the robot. The bottom images shows the fitted SQ on those objects. The poses of the SQ are also visualized in Fig.b and Fig.e

object increases, the time duration of the optimization process also increases.

TABLE II

SINGLE OBJECT PARAMETERS. METHOD 1 IS MIRRORING AND SQ FITTING, METHOD 2 BY QUISPE ET AL. [17]. AVERAGE TIME IN SECONDS.

Object	Meth	a_1	a_2	a_3	e_1	e_2	Avg. time	Avg. error
toy	1	0.023	0.024	0.15	0.389	1.031	0.535	9.2%
cylinder	2	0.022	0.022	0.15	0.561	0.928	1.788	9.78%
ball	1	0.69	0.09	0.45	1.003	0.95	0.78	5.23%
	2	0.43	0.04	0.043	0.986	0.972	1.23	6.24%
Cheese	1	0.14	0.26	0.43	0.1557	0.319	0.82	8.41%
Box	2	0.12	0.12	0.45	0.1329	0.417	1.18	9.6%
dice	1	0.05	0.05	0.479	0.603	0.606	0.56	6.77%
	2	0.054	0.054	0.052	0.629	0.613	0.92	6.28%
Pringles	1	0.012	0.012	0.168	0.355	1.28	0.95	7.48%
Box	2	0.016	0.016	0.162	0.372	1.315	1.24	8.21%

B. Grasping

For grasping objects both in single and dense cluttered scenarios, we compare our results with [23] and [25] in Table II and Fig. 10, as these two methods are well-recognized grasping methods for unknown objects. These two methods have certain advantages over our method as they are learning-based methods. For single objects, the objects are kept in 3 different positions in the environment and the same process of grasping is been iterated 3 times. The average results are presented. For dense cluttered scenes, 5 experiments are performed and for all the experiments and the objects are kept the same for individual experiments. The objects chosen for each experiments are shown in Fig. 9.

As our system does not rely on iteration for finding grasps to execute, it provides better results in time efficiency than other methods. For single objects, location 2 is the place where every method struggled. The location is chosen deliberately far away from the camera frame, where the segmentation process is prone to failure. As the Agile method is not dependent on segmentation, it provides a better result for only one object (ball). It failed significantly for the rest of the objects. For cluttered scenes in Fig. 9 our system consumes much more time for the first and the last scenarios than the other scenarios, as those contains the highest number of objects in a given scene. In experiment 1, though our method calculated all the grasp poses, the dice slipped from the gripper, affecting the accuracy. The vertical placement of the camera where it perceives the environment from the top explains the reason for the significantly poor performance of the Haf grasping method. As for the Agile grasping, the execution system of agile grasping is not based on most vertical grasps, so though in most of the cases, the objects are perceived properly for good grasps, it still failed to execute them.

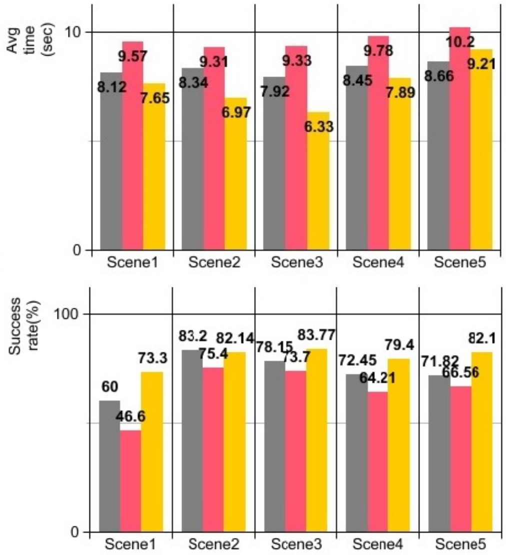

Fig. 10. Grasping result compared with methods presented in [23] and [25]. From left to right: a) avg. time and b)success rate for individual objects. Gray bars are [23], Red bars are [25] and while yellow bars are by Mirroring and SQ fitting

VIII. CONCLUSIONS

In this work, we propose a fast antipodal grasping algorithm based on the properties of the SQ, which are fitted to single-view objects. The generation of contact points on the SQ is very fast and does not require a second phase

TABLE III
OBJECT IN ISOLATION

Object	Method	Loc. 1	Loc. 2	Loc. 3	Avg .Time	succeess Rate (%)
Ball	Agile	3/3	3/3	3/3	4.24s	66.67
	Haf	3/3	1/3	2/3	6.71s	66.67
	HyperGrasp	3/3	1/3	2/3	3.02s	66.67
Box	Agile	2/3	1/3	2/3	4.57s	55.56
	Haf	2/3	1/3	1/3	6.32s	44.45
	HyperGrasp	2/3	1/3	2/3	3.21s	55.56
Toy Cylinder	Agile	2/3	1/3	3/3	3.78s	66.67
	Haf	2/3	0/3	2/3	6.45s	44.45
	HyperGrasp	3/3	2/3	3/3	3.252s	88.89
Screw Driver	Agile	1/3	1/3	2/3	4.11s	44.45
	Haf	2/3	0/3	1/3	6.23s	33.34
	HyperGrasp	3/3	1/3	3/3	3.16s	77.78

of analysis of the potential contact points to select the optimal candidate. The method has been tested in regular objects and the results show good potential for real time and good grasping success. This method and its implementation is available and updated for open-source community at https://github.com/jontromanab/sq_grasp

ACKNOWLEDGMENTS

The authors are grateful to Prof. Maya Cakmak for providing us with a PR2 robot we could use for the experiments.

REFERENCES

[1] A. Saxena, J. Driemeyer, and A. Y. Ng, "Robotic grasping of novel objects using vision," *Int. J. Rob. Res.*, vol. 27, pp. 157–173, Feb. 2008.

[2] J. Subrahmonia, D. B. Cooper, and D. Keren, "Practical reliable bayesian recognition of 2d and 3d objects using implicit polynomials and algebraic invariants," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, pp. 505–519, May 1996.

[3] M. H. Mousa, R. Chaine, S. Akkouche, and E. Galin, "Efficient spherical harmonics representation of 3d objects," in *Computer Graphics and Applications, 2007. PG '07. 15th Pacific Conference on*, pp. 248–255, Oct 2007.

[4] M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz, "Rotation invariant spherical harmonic representation of 3 d shape descriptors," in *Symposium on geometry processing*, vol. 6, pp. 156–164, 2003.

[5] K. Wu and M. D. Levine, *3-D object representation using parametric geons*. Citeseer.

[6] J.-H. Chuang, N. Ahuja, C.-C. Lin, C.-H. Tsai, and C.-H. Chen, "A potential-based generalized cylinder representation," *Computers & Graphics*, vol. 28, no. 6, pp. 907–918, 2004.

[7] D. Terzopoulos, A. Witkin, and M. Kass, "Symmetry-seeking models and 3d object reconstruction," *International Journal of Computer Vision*, vol. 1, no. 3, pp. 211–221, 1988.

[8] N. J. Mitra, L. Guibas, and M. Pauly, "Partial and approximate symmetry detection for 3d geometry," *ACM Transactions on Graphics (SIGGRAPH)*, vol. 25, no. 3, pp. 560–568, 2006.

[9] L. G. Shapiro and J. D. Moriarty, "A generalized blob model for three-dimensional object representation,"

[10] M. Przybylski, T. Asfour, and R. Dillmann, "Unions of balls for shape approximation in robot grasping," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1592–1599, Oct 2010.

[11] M. Ohuchi and T. Saito, "Three-dimensional shape modeling with extended hyperquadrics," in *Proceedings Third International Conference on 3-D Digital Imaging and Modeling*, pp. 262–269, 2001.

[12] A. H. Barr, "Superquadrics and angle-preserving transformations," *IEEE Computer Graphics and Applications*, vol. 1, pp. 11–23, Jan 1981.

[13] A. Jaklič, A. Leonardis, and F. Solina, *Segmentation and Recovery of Superquadrics: Computational Imaging and Vision*. Norwell, MA, USA: Kluwer Academic Publishers, 2000.

[14] G. Biegelbauer and M. Vincze, "Efficient 3d object detection by fitting superquadrics to range image data for robot's object manipulation," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pp. 1086–1091, April 2007.

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

[16] D. Katsoulas and A. Jaklic, "Fast recovery of piled deformable objects using superquadrics," in *Proceedings of the 24th DAGM Symposium on Pattern Recognition*, (London, UK, UK), pp. 174–181, Springer-Verlag, 2002.

[17] A. H. Quispe, B. Milville, M. A. Gutirrez, 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)*, pp. 3702–3708, May 2015.

[18] S. Thrun and B. Wegbreit, "Shape from symmetry," in *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, vol. 2, pp. 1824–1831 Vol. 2, Oct 2005.

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

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

[21] O. Kroemer, H. B. Amor, M. Ewerton, and J. Peters, "Point cloud completion using extrusions," in *2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012)*, pp. 680–685, Nov 2012.

[22] R. Pascoal, V. Santos, C. Premebida, and U. Nunes, "Simultaneous segmentation and superquadrics fitting in laser-range data," *IEEE Transactions on Vehicular Technology*, vol. 64, pp. 441–452, Feb 2015.

[23] M. Gualtieri, A. ten Pas, K. Saenko, and R. Platt, "High precision grasp pose detection in dense clutter," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 598–605, 2016.

[24] A. ten Pas, K. Saenko, and R. Platt, "Combining grasp pose detection with object detection," in *R:SS 2016 Workshop on Bootstrapping Manipulation Skills*, 2016.

[25] D. Fischinger, A. Weiss, , and M. Vincze, "Learning grasps with topographic features," *The International Journal of Robotics Research*, vol. 34, no. 9, pp. 1167–1194, 2015.

[26] J. Fontanals, B.-A. Dang-Vu, O. Porges, J. Rosell, and M. Roa, "Integrated grasp and motion planning using independent contact regions," in *2014 IEEE-RAS International Conference on Humanoid Robots*, 18-20 November 2014.

[27] Y. Jia, "Curvature-based computation of antipodal grasps," in *2002 IEEE International Conference on Robotics and Automation*, 2002.

[28] I.-M. Chen and J. Burdick, "Finding antipodal point grasps on irregularly shaped objects," *IEEE Transactions on Robotics*, vol. 9, no. 4, 2002.

[29] A. T. Miller and P. K. Allen, "Graspit! a versatile simulator for robotic grasping," *IEEE Robotics Automation Magazine*, vol. 11, pp. 110–122, Dec 2004.

[30] M. Pilu and R. B. Fisher, "Equal-distance sampling of superellipse models," in *Proceedings of the 1995 British Conference on Machine Vision (Vol. 1)*, BMVC '95, (Surrey, UK, UK), pp. 257–266, BMVA Press, 1995.

[31] J. Papon, A. Abramov, M. Schoeler, and F. Wrgtter, "Voxel cloud connectivity segmentation - supervoxels for point clouds," in *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2027–2034, June 2013.

[32] A. Ranganathan, "The Levenberg-Marquardt Algorithm," 2004.