

Supersurfaces for grasping, scene generation and contact modelling

Joan Badia Torres^a, Lidia Garrido^a, Alba Perez Gracia^{a,b}

^a*Department of Mechanical Engineering, Universitat Politècnica de Catalunya (UPC), Barcelona, Spain*

^b*Institut de Robòtica i Informàtica Industrial (CSIC-UPC), Barcelona, Spain*

Abstract

In this work we study the use of superquadrics and superquartics for grasping and contact detection. We include an overview of the different formulations that have been derived from the superquadrics, and we group all these under the denomination of *supersurfaces*: superquadrics, superquartics, hyperquartics, superfigures, deformed superquadrics and combination of superquadrics. The focus of this work is in the formulation of superellipsoids and supertoroids, with derivations for their differential geometric properties and distance formulations. We apply this to grasp synthesis and to formulations for the contact between objects.

Keywords:

superquadrics, supertoroids, grasping, contact

1. Introduction

Many robotics actions require the knowledge of the shape, position, and dynamic properties of objects found in the robot's environment.

Accurate grasping of possibly unknown objects is one of the main needs for robotic systems in unstructured environments. In order to do so, the robotic system has to go through multiple costly calculations. The object of interest needs to be defined, which includes segmentation, obtaining the suitable surface impression of the object, and identifying the position of the object. Then the grasp needs to be synthesized by calculating the set of points on the object where the fingers can be placed and the preferred approach directions. Typical robotic grasping methods focus on the optimization of

stable grasp metrics. Finally, a safe path of the hand towards the object must be planned.

For all these applications, different approaches have been tested for geometric modeling fitting; among others, implicit polynomials [57], spherical harmonics [41], but mostly simple shapes such as spheres or cylinders [17], [34]. The aim is always to find a model similar enough to the targeted objects, while computationally simple.

Complexity increases when the objects are deformable. In this case, the selected modeling method has to allow for large global or small local deformations, which occur when forces are applied on it.

Here we focus on the use of supersurfaces to model the shape and position of objects. The term *supersurfaces* is used to encompass results that include superquadrics, superquartics, supershapes, superformula and hypersurfaces, as well as their deformed variations.

Superquadrics and hyperquadrics can adapt to a variety of shapes. Early work on superquadrics includes [6], [27], [11]; [30] shows their application for shape and pose recovery. Early work on hyperquadrics can be found in [42]. Superquadrics are being used to capture point cloud geometry in combination with neural networks [39], or to create virtual environments [44]. The inherent symmetry of superquadrics can be overcome by adding coefficients that allow the stretching and deforming of the surfaces [32]. Objects with holes, common in human environments, can be modeled using supertoroids [5], which are derived from quartic equations.

In Section 2, a review on the methods for modeling objects for grasping is presented. Section 3 shows some of the existing formulations and reviews the literature on modeling objects with supersurfaces.

We focus then on superquadrics and superquartics to gather, in Section 4, different methods for fitting supersurfaces, and to show the use of radial distance on different formulations. In Section 5, we apply superquadrics and superquartics for the grasping of objects with different types of grippers. Section 6 presents some results on the use of superquadrics for identifying contact between objects and modeling the contact forces.

Overall, the mathematical description of supersurfaces and their geometric properties, some techniques for finding pose and shape of objects, and the results of some applications, are presented and discussed.

2. Grasping

Out of the subtasks that compose a grasping action -identification of the object or area to grasp, grasp synthesis and robot’s path planning-, we present here a brief review of the first two.

A considerable amount of work exists in the area of computer vision for identifying objects and computing feasible grasping points on known or novel objects. Initial efforts used 2D images [52]. The extensive use of RGB-D cameras allows 3-D object identification from point clouds and derived parameters.

Within the explicit methods, the strategies for identifying the objects or areas to be grasped can be classified as model-based identification, geometric modeling of objects, and model-free grasping surface identification. Geometric modeling is also used for segmentation or as a first step for model-based systems, or within machine learning algorithms [54], [61].

When a finite, known, and well-defined set of objects is to be encountered, a database of these objects with their corresponding preferred grasps can be used. Earlier efforts in this direction can be found for instance in [19]. Databases of objects may include optimized grasping points as a part of the features of the object ([12], [40]). As the environment becomes less structured, the system has to explore and evaluate an increasingly large number of objects and grasps, and observing a novel object may be common. In these cases, a methodology for grasping unknown objects may be preferred.

Model-free grasping surface identification can be found on [43] and [4]. They rely on geometry and certain grasping heuristics, but the information about the object to be grasped is lost.

Different methods have been utilized for geometric model of objects, from blob models [53] and detection of symmetries [60] to implicit polynomials [58] and spherical harmonics [31]; however, the most common representations have used simple geometry, such as generalized cylinders [18] and union of balls [48].

2.1. Computation of grasping points

Most methods for the calculation of feasible grasping points, on a point cloud or a mesh, are computationally expensive: a large set of candidate grasps is generated, and those are evaluated using different methods, such as convolutional neural networks [26]. Such methods avoid the need for robust segmentation but cannot assure the assignment of the grasp to a

target object. A similar approach [20] uses local topographical information from the point cloud to calculate antipodal grasps.

A different set of methods use the fitting of object models to the point cloud to generate smaller or simpler sets of grasp points [22]. Curvature-based grasping using antipodal points on differentiable curves was studied in [29]. In [14], a grasping energy function is used to calculate antipodal grasping points on a local model of the surface.

3. Modeling objects with supersurfaces and combination of supersurfaces

Superquadrics were used from their early research to model a variety of closed, symmetric objects without holes. However, superquadrics and superquartics can also be used to model more irregular objects, objects formed by a combination of supersurfaces, or those objects in which a large local deformation happens at specific points, similar to the corotational approach for modeling deformable objects. In addition, other approaches such as hyperquadrics or the superformula have also been developed.

3.1. Superquadrics and supertoroids

Quadrics are algebraic varieties of affine spaces that define hypersurfaces (surfaces of dimension $n - 1$ in a space of dimension n). When we restrict ourselves to real affine spaces, quadrics define surfaces with a very specific property: these can be expressed by quadratic equations,

$$\mathbf{x}^T[M]\mathbf{x} + [N]\mathbf{x} + P = 0, \quad (1)$$

or, in homogeneous form,

$$\left\{ \mathbf{x}^T \quad 1 \right\} \begin{bmatrix} [M] & [N]^T/2 \\ [N]/2 & P \end{bmatrix} \begin{Bmatrix} \mathbf{x} \\ 1 \end{Bmatrix} = 0, \quad (2)$$

The general equation of a quadric in the Euclidean three-dimensional space,

$$Ax^2 + By^2 + Cz^2 + Dxy + Eyz + Fxz + Gx + Hy + Iz + J = 0, \quad (3)$$

can be transformed to a normal form with a change of coordinates to its principal axes. In the case of an ellipsoid, this normal form will have its

axes centered and along the principal directions of the quadric, to obtain the simplified equation

$$\frac{x^2}{a_1^2} + \frac{y^2}{a_2^2} + \frac{z^2}{a_3^2} = 1, \quad (4)$$

and we would obtain similar expressions for the rest of quadrics.

We can also write a parametric expression for quadrics. Consider again the ellipsoid with its principal reference frame. Each point of the ellipsoid can be expressed with the following functions, parameterized by angles η and ω ,

$$f(x, y, z) = \begin{cases} a_1 \cos \eta \cos \omega \\ a_2 \cos \eta \sin \omega \\ a_3 \sin \eta. \end{cases} \quad (5)$$

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

The implicit equation of a supersurface of superellipse type could be written, in a canonical coordinate system,

$$j \frac{x}{a_1} f^{\epsilon_1} + j \frac{y}{a_2} f^{\epsilon_2} + j \frac{z}{a_3} f^{\epsilon_3} = 1. \quad (6)$$

However we will limit our results to some of these supersurfaces.

The pose of the superquadric, considering its canonical frame, 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 three independent parameters, such as ρ, ψ, θ , to define the orientation. The total set of parameters that fully defines a superquadric's shape and pose is the set $f \mathbf{a}, \epsilon, p_x, p_y, p_z, \rho, \psi, \theta g$, where \mathbf{a} is the vector of parameters defining the dimensions of the major axes of the superquadric and ϵ is the vector containing the exponents.

In the expressions presented below, we mostly follow [6] and our work in [37] and [5].

3.1.1. Superellipsoids

The superellipsoid is the most widely used member of the family of superquadrics, and the object of our study; other figures are the superhyperboloid or superparaboloid. A superellipsoid can be obtained as the spherical

product of two superellipses [51], S_1 and S_2 , to obtain the parametric equations

$$f_e(\eta, \omega) = S_1(\eta) \cdot S_2(\omega) = \begin{Bmatrix} \cos^{\epsilon_1} \eta \\ a_3 \sin^{\epsilon_1} \eta \end{Bmatrix} \cdot \begin{Bmatrix} a_1 \cos^{\epsilon_2} \omega \\ a_2 \sin^{\epsilon_2} \omega \end{Bmatrix} = \begin{Bmatrix} a_1 \cos^{\epsilon_1} \eta \cos^{\epsilon_2} \omega \\ a_2 \cos^{\epsilon_1} \eta \sin^{\epsilon_2} \omega \\ a_3 \sin^{\epsilon_1} \eta \end{Bmatrix}, \quad (7)$$

with $\eta \in [-\frac{\pi}{2}, \frac{\pi}{2}]$ and $\omega \in [-\pi, \pi]$.

The coefficients a_1, a_2, a_3 are the scaling factors of the three principal axes; the real, positive exponent ϵ_1 controls the shape of the superellipsoid's cross-section in the planes orthogonal to (x, y) plane, and ϵ_2 , also real and positive, controls the shape of the superellipsoid's cross-section parallel to the (x, y) plane.

The total set of parameters that fully defines the superellipsoid's shape and pose consists of 11 variables, $\{a_1, a_2, a_3, \epsilon_1, \epsilon_2, p_x, p_y, p_z, \rho, \psi, \theta, g\}$. Figure 1 shows the variation of shapes controlled by the exponents,

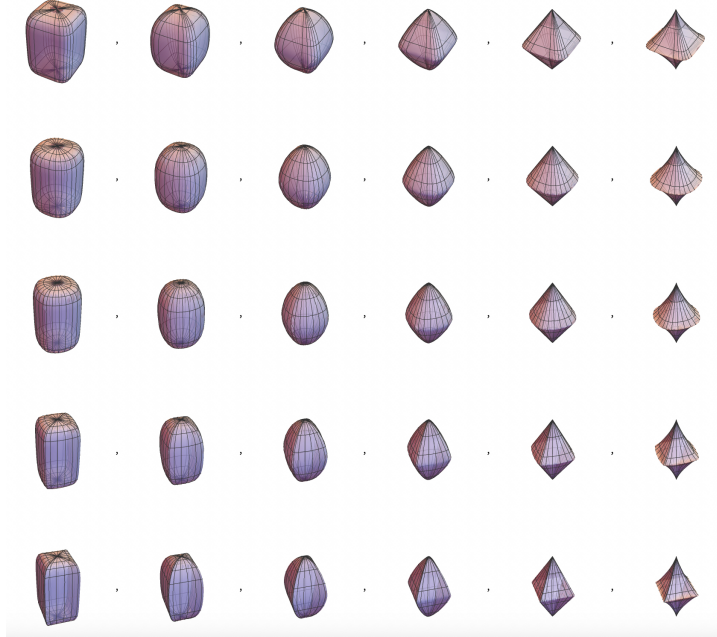


Figure 1: Superellipsoid shapes for different exponents (ϵ_1 along rows and ϵ_2 along columns).

In order to obtain the four symmetric quadrants, take $\cos^{\epsilon_1} \eta = \text{sign}(\cos \eta) |\cos \eta|^{\epsilon_1}$, and similarly for the other trigonometric terms. The parameterized equation

becomes

$$f_e(\eta, \omega) = \left\{ \begin{array}{l} a_1 \operatorname{sign}(\cos \eta) j \cos \eta^{j\epsilon_1} \operatorname{sign}(\cos \omega) j \cos \omega^{j\epsilon_2} \\ a_2 \operatorname{sign}(\cos \eta) j \cos \eta^{j\epsilon_1} \operatorname{sign}(\sin \omega) j \sin \omega^{j\epsilon_2} \\ a_3 \operatorname{sign}(\sin \eta) j \sin \eta^{j\epsilon_1} \end{array} \right\}. \quad (8)$$

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

$$F_e(x, y, z) : \left(\left| \frac{x}{a_1} \right|^{\frac{2}{\epsilon_2}} + \left| \frac{y}{a_2} \right|^{\frac{2}{\epsilon_2}} \right)^{\frac{\epsilon_2}{\epsilon_1}} + \left| \frac{z}{a_3} \right|^{\frac{2}{\epsilon_1}} = 1. \quad (9)$$

It is easy to show that the superquadric is bounded by the planes given by $x = a_1$, $y = a_2$ and $z = a_3$.

More detailed properties of the surface will be studied in subsequent sections.

3.1.2. Supertoroids

The toroid is a quartic surface with the particularity that it is also a surface of revolution. Its interest lies in the fact that, unlike the quadrics, the toroids have a hole. Topologically speaking, while the quadrics have genus zero, the toroids have genus one. There are many objects in human environments with genus one, from objects with handles to rings and frames.

The equation of the torus,

$$(x^2 + y^2 + z^2 + R^2 - r^2)^2 - 4R^2(x^2 + y^2) = 0, \quad (10)$$

in which R and r being the major and minor radii, respectively, can be generalized to toroids of different cross-sections, and to more general shapes having a single hole, if we follow the same approach defined for the superquadrics. The *supertoroids* can change their shape according to four parameters for the lengths along the axes and two exponents, ϵ_1 and ϵ_2 .

Supertoroids were already identified in the classic paper by Barr [6], who worked with both superellipsoids and supertoroids, given the similarity in their parametric equations. The parametric expression of the supertoroid is obtained with the spherical product

$$\begin{aligned} f_t(\eta, \omega) &= \left\{ \begin{array}{l} a_4 + \cos^{\epsilon_1} \eta \\ a_3 \sin^{\epsilon_1} \eta \end{array} \right\} \left\{ \begin{array}{l} a_1 \cos^{\epsilon_2} \omega \\ a_2 \sin^{\epsilon_2} \omega \end{array} \right\} \\ &= \left\{ \begin{array}{l} a_1(a_4 + \cos^{\epsilon_1} \eta) \cos^{\epsilon_2} \omega \\ a_2(a_4 + \cos^{\epsilon_1} \eta) \sin^{\epsilon_2} \omega \\ a_3 \sin^{\epsilon_1} \eta \end{array} \right\}. \end{aligned} \quad (11)$$

Here, as with the superellipses, in order to complete the four quadrants we use $\cos^{\epsilon_1} \eta = \text{sign}(\cos \eta) |\cos \eta|^{\epsilon_1}$, and similarly for the rest of trigonometric variables.

The real positive exponent ϵ_2 controls the shape in the canonical x y plane and the real, positive exponent ϵ_1 controls the shape in the cross sections on perpendicular planes containing the z axis. The coefficients a_i control the dimensions in the x , y , z directions. These intrinsic parameters form a vector $v_i = (a_1, a_2, a_3, a_4, \epsilon_1, \epsilon_2)$. Fig. 2 shows a sample of shapes that can be accomplished by modifying the exponents, from smaller to larger values.

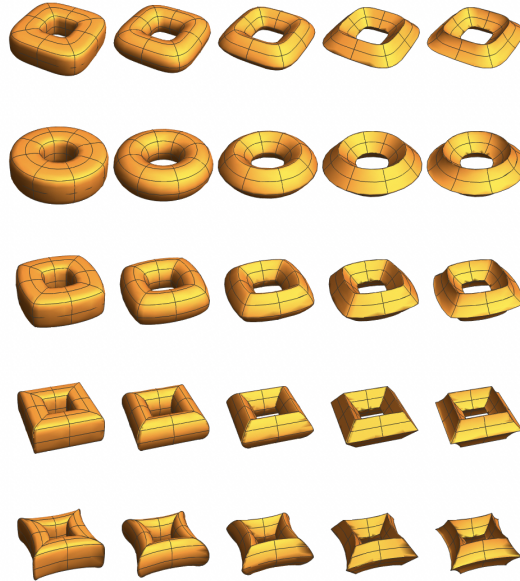


Figure 2: Different supertoroid shapes obtained with the same a_i values and increasing numerical values for the exponents ϵ_1 and ϵ_2 , from 0.5 to 2.5. [5]

The supertoroid has the implicit equation, in its canonical coordinate system,

$$F_t(v_i, x, y, z) : \left| \left(\left| \frac{x}{a_1} \right|^{\frac{2}{\epsilon_2}} + \left| \frac{y}{a_2} \right|^{\frac{2}{\epsilon_2}} \right)^{\frac{\epsilon_2}{2}} a_4 \right|^{\frac{2}{\epsilon_1}} + \left| \frac{z}{a_3} \right|^{\frac{2}{\epsilon_1}} = 1. \quad (12)$$

The supertoroid is bounded by planes $x = (a_1 + a_1 a_4)$, $y = (a_2 + a_2 a_4)$ and $z = a_3$.

If we consider each radial section by a plane containing the z-axis, we obtain superellipses, which we denote as *cross-section superellipse*. The center of all those forms a superellipse f_m , called the *mean superellipse*, which is located on the $x-y$ canonical plane and has implicit equation

$$F_m(v_i, x, y) : \left| \frac{x}{a_1 a_4} \right|^{\frac{2}{\epsilon_2}} + \left| \frac{y}{a_2 a_4} \right|^{\frac{2}{\epsilon_2}} = 1, \quad (13)$$

assuming $a_4 > 0$. The mean superellipse is shown in Figure 3 and its parametric equation is

$$f_m(\omega) = \left\{ \begin{array}{l} a_1 a_4 \cos^{\epsilon_2} \omega \\ a_2 a_4 \sin^{\epsilon_2} \omega \end{array} \right\}. \quad (14)$$

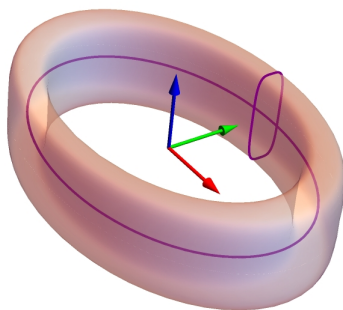


Figure 3: The mean superellipse and a cross-section superellipse.

Other interesting geometric parameters include the radius of the mean superellipse as a function of the angle ω ,

$$\bar{R}(\omega) = a_4 \sqrt{a_1^2 \cos^{2\epsilon_2} \omega + a_2^2 \sin^{2\epsilon_2} \omega}, \quad (15)$$

in which absolute values must be added whenever necessary. The cross-section superellipses can be calculated as parameterized by the angle ω_π of the vertical plane, starting at the x-axis. It is important to realize that the angle of the parameterized expression at that cross section, ω_s , will not correspond to the angle ω_π of the plane, shown in Figure 9. The relation between these two angles is given by

$$\tan^{1/\epsilon_2} \omega_s = \frac{a_2}{a_1} \tan \omega_\pi. \quad (16)$$

The cross-section superellipse f_c can be expressed in a local canonical coordinate frame $f x', z' g$, whose orientation corresponds to a z -rotation by an angle ω_π , $[R_z(\omega_\pi)]$.

The implicit expression of the cross-section superellipse in this local frame is

$$f_c(v_i, x, y, z) : \quad j \frac{x'}{a_{\omega_s}} \Big|_{\epsilon_1}^{\frac{2}{\epsilon_1}} + \Big| \frac{z'}{a_3} \Big|_{\epsilon_1}^{\frac{2}{\epsilon_1}} = 1, \quad (17)$$

where a_{ω_s} is given by

$$a_{\omega_s} = \sqrt{a_1^2 \cos^{2\epsilon_2} \omega_s + a_2^2 \sin^{2\epsilon_2} \omega_s}. \quad (18)$$

The change of coordinates to the global frame allows us to express the local coordinates $f x', z' g$ as

$$\begin{aligned} x' &= \frac{a_1 \cos^{\epsilon_2} \omega_s (x - a_1 a_4 \cos^{\epsilon_2} \omega_s)}{a_{\omega_s}} + \frac{a_2 \sin^{\epsilon_2} \omega_s (y - a_2 a_4 \sin^{\epsilon_2} \omega_s)}{a_{\omega_s}} \\ z' &= z. \end{aligned} \quad (19)$$

This basic formulation is used to calculate surface properties and fitting algorithms. It is also easy to compute differential geometric properties such as tangents, normals and curvatures. Even though the surfaces are not regular for all values of the exponents, if we restrict to $\epsilon_1, \epsilon_2 < 2$, the analysis can be performed in patches that are well behaved at seams and stitches. Figure 4 shows the tangent vectors at a singular point for different values of the exponent. For more details, see [5].

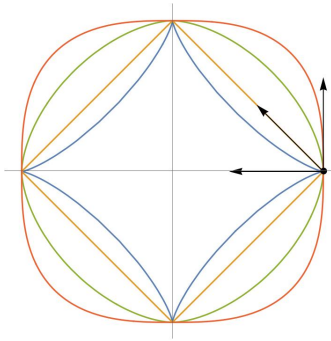


Figure 4: Tangent vector at the seams for a superellipse, depending on the values of ϵ_2 and ϵ_1 . Shown are the $\epsilon_2 > 2$ (cusp), $\epsilon_2 = 2$ (singular point), and $\epsilon_2 < 2$ [5].

3.2. Deformed supersurfaces, hyperquadrics and the superformula

We can extend the range of shapes in which we apply superellipses and supertoroids if we allow the superquadrics to undergo parametric deformations. This was developed by Barr [7] and Solina and Bajcsy [55]. The deformations used included bending, tapering, twisting, and cavity deformation; others can also be defined. In general, the gain in shape diversity implies a loss of efficiency on the fitting process and the stability of the representation [9].

These deformed superquadrics have been used in dynamics and analysis of deformable bodies, by defining a function to relate applied forces to the shape change.

3.2.1. Tapering, bending and twisting

Some of the equations for tapering, bending and twisting are presented below. In each case, it leads to an increase in the number of parameters defining the shape. For instance, tapering along the z axis implies adding two constant slopes K_x, K_y ,

$$\begin{Bmatrix} X \\ Y \\ Z \end{Bmatrix} = \begin{Bmatrix} (\frac{K_x}{a_3}z + 1)x \\ (\frac{K_y}{a_3}z + 1)y \\ z \end{Bmatrix}, \quad (20)$$

while bending is accomplished by defining a curvature k ,

$$\begin{Bmatrix} X \\ Y \\ Z \end{Bmatrix} = \begin{Bmatrix} x + k \frac{\rho}{k^2 + z^2} \\ y \\ z \end{Bmatrix}, \quad (21)$$

and the twisting is parameterized by an angle $\theta = n\pi(1 + \frac{z}{a_3})$ depending on the vertical axis,

$$\begin{Bmatrix} X \\ Y \\ Z \end{Bmatrix} = \begin{Bmatrix} x \cos \theta & y \sin \theta \\ x \sin \theta + y \cos \theta \\ z \end{Bmatrix}. \quad (22)$$

Figure 5 shows tapering applied to a superellipse and a supertoroid.

3.2.2. Local deformation: extended superquadrics, hyperquadrics and the superformula

Local deformation may also be desirable, and that needs a different kind of transformation. One way to accomplish this is using *extended superquadrics*. These were developed by Zhou and Kambhanettu [2], see also [10].

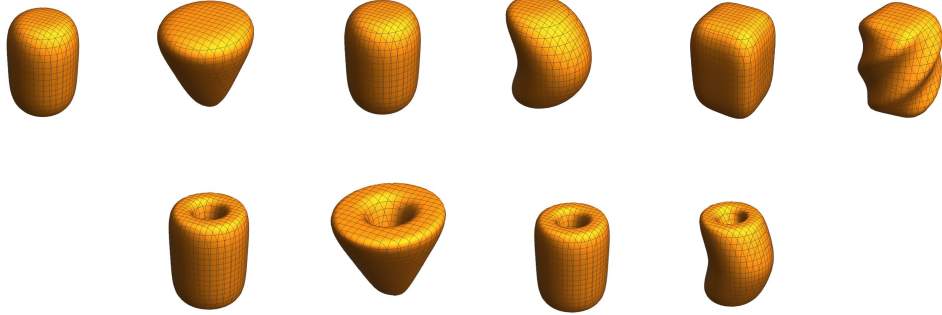


Figure 5: Regular, tapered, bent and twisted superellipsoids and supertoroids

Extended superquadrics generalize the superquadrics in the sense that the exponents no longer have a constant value. For these exponents, they use smooth functions of the parameters, $\epsilon_1 = f_1(\omega)$ and $\epsilon_2 = f_2(\eta)$, and to adapt to the desired deformation, cubic splines are considered.

A similar generalization of quadrics is the *superformula*, which yields *superfigures*. In this case, the constant values a_i are functions of the parameterization angles. The superformula was developed by John Gielis around 1997 [25], for application on botanical modeling. Again, many different shapes, regular and non-regular, can be accomplished. See also [8].

The parametric expression for the extended superellipsoid and the superformula is shown in Eq.(23), where either the exponents or the coefficients are variable in each case,

$$f_e(\eta, \omega) = \left\{ \begin{array}{l} g_1(\eta) \cos^{f_1(\eta)} \eta \quad g_2(\omega) \cos^{f_2(\omega)} \omega \\ g_1(\eta) \cos^{f_1(\eta)} \eta \quad g_2(\omega) \sin^{f_2(\omega)} \omega \\ g_3(\eta) \sin^{f_1(\eta)} \eta \end{array} \right\}, \quad (23)$$

Another generalization of superquadrics is the *hyperquadric*. Hyperquadrics were first defined by Hanson [1] and have the following general expression:

$$F_e(x, y, z) = \sum_{i=1}^N jA_i x + B_i y + C_i z + D_i f^{\epsilon_i} = 1 \quad (24)$$

The sum allows us to create non-symmetric shapes based on the selection of the coefficients.

The variety of shapes obtained using these generalizations is very high, and they are being applied in many areas of science and engineering. However, their computational complexity increases accordingly. It was shown early [13] that the measures that are applied to compute distance to regular superquadrics yield errors and biases on these deformed and general figures.

3.3. Combination of superquadrics

Complex objects can be modeled with a combination of superquadrics, which provides fast convergence on each superquadric and a diversity of shapes. For the combination of supersurfaces the total number of parameters used to define the object will be the sum of the parameters corresponding to each supersurface. In addition, a graph may be needed to indicate the connection, as well as the coordinates of the physical connection between bodies.

In [37] we presented an experiment in which a teddy bear is segmented and superellipses are fitted for each segmented surface, see Figure 6. After this step, adding a spherical joint or a passive spring element between any two connected superquadrics, we obtain an approximation to the behavior of this object with only a few parameters.



Figure 6: Level decomposition, replication and superquadric fitting on asymmetrical object [37]

Chevalier et al. [16] proposed a procedure for the automatic creation of combinations of superquadrics. They perform the segmentation and fitting concurrently by using a split-and-merge procedure, generating several superellipsoids that can be discarded in the fitting process. They use graph theory to encode the connection between superellipses.

Bhabhrawala and Krovi [10] segment objects according to concavity, and fit extended superquadrics to each segmented volume. Fougerolle et al. [23] use a previously-segmented point cloud and fit superformula's supershapes.

R-functions are used to transcribe Boolean information to analytical equations, to describe models composed of several supershapes. In [64], superellipsoids, supertoroids, supercylinders and super cones are merged to generate multiple shapes. Paschalidou et al. [44], use machine learning to parse superellipses together in order to obtain the complex shapes using an unsupervised process.

Sircelj and Solina [54] use a two-step process for segmenting and fitting point cloud data to superellipses, using Mask R-CNN. Alaniz et al. [3] recombine objects using superquadrics as semantic parts, from several 2D views. Instead of learning algorithms, they use more robust optimization of the parameters of the superellipse, and compare the rendered 3D view to the image silhouettes. Liu, Chirikjian et al. [36] iteratively grow superquadric primitives by analyzing connectivity of voxels and perform fitting with a signed distance function, to obtain an efficient system when tested on the ShapeNet dataset. Finally, Park et al. [32] use a set of pre-defined deformable superquadric primitives on point clouds from a single view. A network is trained for segmentation of the cloud and computation of grasping points.

Summarizing, many applications are found where supersurfaces are being used for modeling, and the equations for many of them are presented here. Current computational methods may favor combination of standard superquadrics to the extended formulations of their non-symmetric counterparts for object reconstruction, while the extended formulations may be more valid for organic shapes and biomedical data. .

4. Distance to supersurfaces: point cloud fitting

Supersurfaces, and in particular superquadrics, have been extensively used to fit geometric shapes and poses to point clouds. Some of the early examples were presented in the introduction. A good review of more recent applications can be found in [37], where the issues of partial views and noisy sensors are also discussed.

Here we will focus on finding a metric to assess the proximity of a cloud to a given supersurface, taking into account the physical meaning as well as the convergence.

4.1. Radial distance to a superquadric

The implicit equation of the superquadric defines an inside-outside function. Consider $F_e(x, y, z)$, the left side of Eq.(9). Then it is true, for any points (x, y, z) ,

$$F_e(x, y, z) \begin{cases} < 1 & \text{point inside} \\ = 1 & \text{point on surface} \\ > 1 & \text{point outside} \end{cases} \quad (25)$$

It is possible to directly use the inside-outside function for an optimization procedure. Also the true Euclidean minimum distance between a point and a superquadric can be calculated using optimization. However here we prefer a widely used distance, the *radial Euclidean distance*, in order to calculate distance between points and the superquadric for fitting purposes [28].

The radial Euclidean distance is defined as the distance between the point and the superquadric surface along a line through the point and the origin of the canonical frame of the superquadric [63]. In his work, the non-uniqueness of the fitting according to different metrics was also studied. Several other distances have been defined, for instance the Taubin Method [59].

Here we present the radial distance to the superellipsoid. Given a point of coordinates $p_0 = (x_0, y_0, z_0)$ in the superellipsoid canonical frame, consider a scalar β used to scale vector p_0 so that the tip of this vector, $p_s = \beta p_0$, defines a point on the surface of the superellipsoid. See Figure 7.

This scalar turns out to be easy to calculate using the inside-outside function on point p_s ,

$$F_e(\beta x_0, \beta y_0, \beta z_0) = \beta^{\frac{2}{\epsilon_1}} F_e(x_0, y_0, z_0) = 1. \quad (26)$$

We can calculate the scalar β as

$$\beta^{\frac{2}{\epsilon_1}} = F_e(x_0, y_0, z_0). \quad (27)$$

The radial distance,

$$d = |p_0 - p_s| = |p_0 - \beta p_0| = |1 - \beta| |p_0|, \quad (28)$$

where $| \cdot |$ denotes a given metric. Figure 7 shows the value of the radial distance as the module of the vector defined by $(1 - \beta)p_0$. Notice that the absolute value is necessary, as β could be greater than one for points inside of the superquadric, if we want to keep the distance well defined. Using the Euclidean radial distance,

$$d = \sqrt{(p_0 - p_s)^2} = |p_0| \sqrt{(1 - \beta)^2}. \quad (29)$$

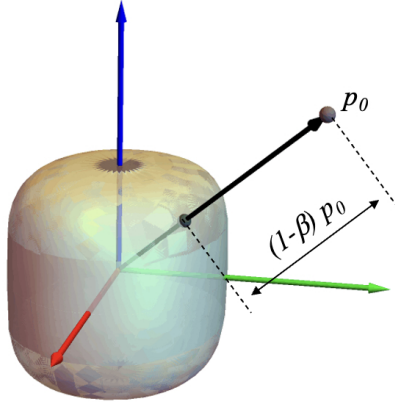


Figure 7: Radial distance to the superellipsoid.

Calculating the radial distance reduces to evaluating the implicit equation of the superquadric at the given point in order to calculate the parameter β .

4.2. Radial distance to a deformed superquadric

An exact or approximate radial distance may be derived for tapered, bent or twisted superquadrics. Here we present an approximation for the tapered superellipsoid with $K_x = K_y = K$, which does not require a numerical method but has limited precision.

Consider a point p_0 , for which we can calculate the vector $p_{es} = \beta p_0$ that locates the point on the original superellipse, according to Eq.(27). The original and deformed superellipsoid coincide for $z = 0$, and we can define the point $p_{e0} = f_e(0, \omega_s)$ on the deformed superellipse. We can also calculate the point obtained applying the deformation in Eq.(20) to point p_{es} , to obtain point p_{ts} .

Now we approximate the tapering so that the line from p_{ts} to p_{es} belongs to the surface, see Figure 8. Then, the vector $p_s = \beta_t p_0$ approximately belongs to the deformed superellipse if

$$\beta_t = \frac{a_3 \beta + K z_0 \beta^2}{a_3}. \quad (30)$$

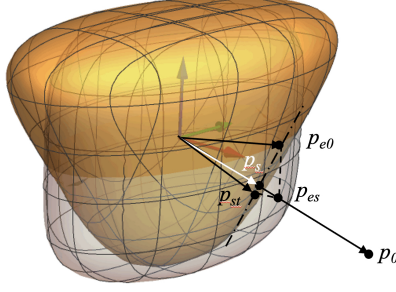


Figure 8: Point p_s lies approximately on the surface of the tapered superellipsoid.

4.3. Meridian radial distance for the supertoroid

Similar to the superellipse, the implicit equation of the supertoroid defines an inside-outside function,

$$F_t(v_i, x, y, z) \begin{cases} < 1 & \text{point inside} \\ = 1 & \text{point on surface} \\ > 1 & \text{point outside} \end{cases} \quad (31)$$

Following the radial distance of the superquadrics, in [5] we define a modification to ensure that the vector intersects the figure.

A point p of the point cloud is projected in the $x-y$ plane of the supertoroid canonical frame (vector p_π). This vector will intersect the mean superellipse defined previously at a point, which we locate using the vector R_π . Create the vector p_R such that $p = R_\pi + p_R$. See Fig. 9.

We define two ratios, the minor ratio β_1 and the major ratio β_2 . The major ratio β_2 locates the intersecting point on the mean superellipse, and the minor ratio β_1 locates the tip of vector p_e on the cross-section superellipse.

The vertical projection plane is located at an angle ω_π from the canonical x -axis,

$$\tan \omega_\pi = \frac{p_y}{p_x}, \quad (32)$$

and the value of vector R_π is found in Eq. 15, taking into account Eq. 16. The major ratio is the fraction of the projected vector p_π that corresponds to the length of the mean radius R_π ,

$$R_\pi = \beta_2 p_\pi = \beta_2 (p_x \cos \omega_\pi, p_x \sin \omega_\pi, z), \quad (33)$$

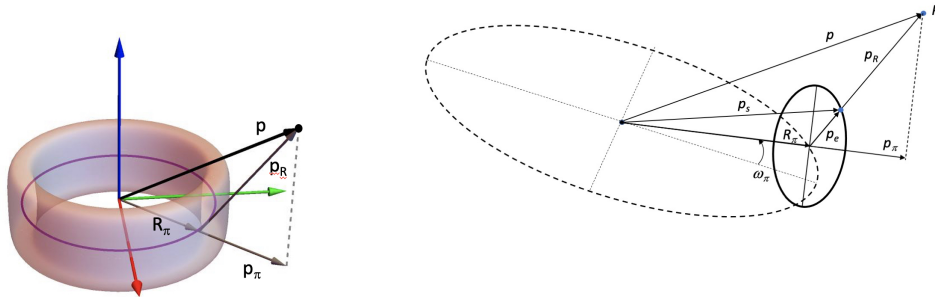


Figure 9: The vector to a point p is the sum of vector R_π (intersection of the mean superellipse with the projection of p on plane $x-y$) and vector p_R . The vector p_s of the closest point on the surface, according to the major and minor radial distance coefficients [5].

The coefficient β_2 can be calculated using the inside-outside function of the mean superellipse for $\beta_2 = 0$,

$$\beta_2^{2/\epsilon_2} = F_m(p_\pi). \quad (34)$$

The minor ratio coefficient yields the vector p_e as a fraction of the length of vector p_R , $p_e = \beta_1 p_R$.

The vector p_e lies on the plane of the cross-section superellipse, so that, in local coordinates, $F_e(p_e) = F_e(\beta_1 p_R) = 1$. Using the transformation to the coordinates of the supertoroid, we calculate

$$\beta_1^{2/\epsilon_1} = F_e(p_R). \quad (35)$$

Write the vector p_s as

$$p_s = \beta_1 p + \beta_2 (1 - \beta_1) p_\pi \quad (36)$$

The length of the difference of these two vectors is what we call the *meridian radial distance* $d_s = |p - p_s|$.

This distance can be used for regular surfaces of revolution for which the generatrix is a closed curve. This distance takes a very simple expression as a function of the minor and major coefficients β_1 and β_2 . Using Euclidean

distance, the meridian radial distance from a point p_0 to the supertoroid can be calculated as

$$d_s = j(1 - \beta_1)j\sqrt{(1 - \beta_2)^2(p_{0x}^2 + p_{0y}^2) + p_{0z}^2}. \quad (37)$$

The fitting procedure consists on minimizing the meridian radial distance from the point cloud to the supertoroid.

5. Supersurface Grasp Synthesis

For grasp synthesis, both the object to be grasped and the embodiment of the grasping technology play an important role.

In this section we briefly describe the steps to obtain a good representation, to then proceed to create quick and robust grasping heuristics for superellipsoids and supertoroids.

This work focuses on the use of two popular end-of-arm tools: suction grippers with one suction cup, and two-fingered grippers with parallel fingers. It is well known that antipodal grasps can be successful in both convex and concave objects in the presence of friction.

A possible efficient grasping strategy is to create some heuristics for the grasping, yielding a grasping decision table. It will be dependent on the supertoroid parameters but also on the type of gripper used.

For the one-cup suction gripper, the key is to find a surface with little curvature and balancing as much as possible the forces so that no moments are created. For the two-fingered grippers, antipodal points must be found with certain properties.

The grasping strategies are defined in the canonical reference frame of the object, as a set of contact points and directions of approach. Given a contact point p_c in the local frame, it will be transformed as

$$P_c = [T_i]p_c, \quad (38)$$

where $[T_i]$ is the 4 × 4 homogeneous transformation from the world frame to the principal frame of supersurface i .

5.1. Partial-view Model Identification

When seeing objects with a single RGB-D camera or structured light sensor, the obtained point cloud represents a segment of the surface of the

object. We also need to take into account the noise introduced in the sensing. Figure 10 shows the partial cloud generated by adding some noise to a sampled supertoroid and cutting it by a given plane, as well as a real point cloud obtained from a simple view of an RGB-D camera.

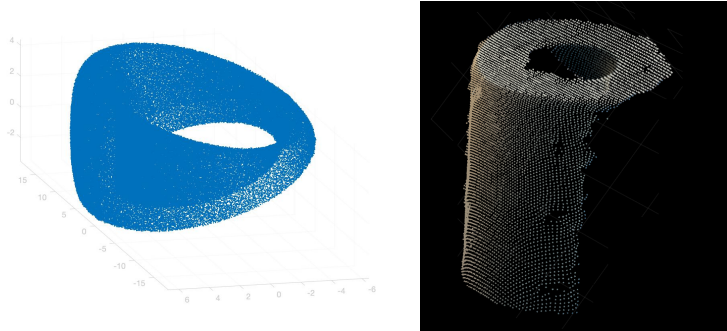


Figure 10: Partial, noisy point cloud for a supertoroid generated with noisy Pilu-Fisher sampling [46], left, and real point cloud for a paper roll captured with an RGB-D camera, right.

In order to identify the camera information with the complete shape of the object, several strategies may be followed. We could combine the point clouds of several cameras or several views of a single camera [45], complete the cloud assuming symmetry ([49], [38] [33]), or we could work with the partial point cloud.

While having a complete cloud is the ideal outcome, generating several views increases the time and complexity of operations. The strategies to complete the cloud assuming some symmetry does not add any additional information to the fitting of superquadrics or supertoroids, which are indeed symmetric.

The strategy of fitting the partial cloud obtained from a single view, skipping the extra step of completing the cloud based on some symmetry assumption, has been proved successful and quick in our tests.

5.2. Segmentation

It is common that the camera captures several objects on a planar surface. Usually the dominant table plane is removed using for instance a RANSAC algorithm [21]. Then a clustering algorithm ([56], [35], [15]) can be used to separate the objects.

Once segmented, initial conditions for the fitting are calculated using PCA and the centroid of the point cloud, which can be pushed toward the occluded region by using some heuristics.

5.3. Grasping strategies for superellipsoids

The grasping heuristics for superellipsoids are based on minimizing wrenches, by grasping closer to the centroid of the object, and grasping at the points of minimum curvature, in order to maximize the contact with fingers or suction cups. To those we add the constraints given by the dimensions and range of the robotic gripper and the task-based direction of approach.

In addition, for parallel grippers, the gripper contact points for antipodal grasping require both minimum curvature and proximity to collinear normals to the surface at the contact points.

The normal to the surface at the first patch $(0 \leq \omega, \eta \leq \pi/2)$ for both superellipsoids and supertoroids is

$$n_{surf} = \frac{1}{m_n} \left\{ \begin{array}{l} a_2 a_3 \sin^{\epsilon_1 - 2} \eta \sin^{\epsilon_2 - 2} \omega \\ a_1 a_3 \sin^{\epsilon_1 - 2} \eta \cos^{\epsilon_2 - 2} \omega \\ a_1 a_2 \cos^{\epsilon_1 - 2} \eta \cos^{\epsilon_2 - 2} \omega \sin^{\epsilon_2 - 2} \omega \end{array} \right\}, \quad (39)$$

with

$$m_n = \left((a_1 a_2 \cos^{\epsilon_1 - 2} \eta \cos^{\epsilon_1 - 2} \omega \sin^{2\epsilon_1 - 2} \omega)^2 + (a_3 \sin^{\epsilon_1 - 2} \eta)^2 (a_1^2 \cos^{2\epsilon_1 - 4} \omega + a_2^2 \sin^{2\epsilon_1 - 4} \omega) \right)^{\frac{1}{2}}, \quad (40)$$

and similarly for the rest of patches, with some sign differences. At the seams and stitches, the normals are only defined for $\epsilon_1, \epsilon_2 < 2$, see Figure 4, and are shown in Table 5.3. In this table the denominators are

$$\begin{aligned} m_{\omega 0} &= \sqrt{a_1^2 \cos^{2\epsilon_2 - 4} \omega + a_2^2 \sin^{2\epsilon_2 - 4} \omega} \\ m_{\omega \pi} &= \sqrt{a_2^2 \cos^{2\epsilon_1 - 4} \eta + a_3^2 \sin^{2\epsilon_1 - 4} \eta} \\ m_{\eta 0} &= \sqrt{a_1^2 \cos^{2\epsilon_1 - 4} \eta + a_3^2 \sin^{2\epsilon_1 - 4} \eta}. \end{aligned} \quad (41)$$

From the previous criterion to minimize wrenches, the line of the grasping points for parallel grippers passes through the origin of the canonical frame, that is, they are antipodal points. We impose collinearity by minimizing

Table 1: Normal directions to the surface at the limits of the first patch

	$\omega = 0$	ω	$\omega = \frac{\pi}{2}$
$\eta = 0$	$\begin{Bmatrix} 1 \\ 0 \\ 0 \end{Bmatrix}$	$\frac{1}{m_{\omega 0}} \begin{Bmatrix} a_2 \sin^{\epsilon_2} \omega \\ a_1 \cos^{\epsilon_2} \omega \\ 0 \end{Bmatrix}$	$\begin{Bmatrix} 0 \\ 1 \\ 0 \end{Bmatrix}$
η	$\frac{1}{m_{\eta 0}} \begin{Bmatrix} a_3 \sin^{\epsilon_1} \eta \\ 0 \\ a_1 \cos^{\epsilon_1} \eta \end{Bmatrix}$	n_{surf}	$\frac{1}{m_{\omega \pi}} \begin{Bmatrix} 0 \\ a_3 \sin^{\epsilon_1} \eta \\ a_2 \cos^{\epsilon_1} \eta \end{Bmatrix}$
$\eta = \frac{\pi}{2}$	$\begin{Bmatrix} 0 \\ 0 \\ 1 \end{Bmatrix}$	$\begin{Bmatrix} 0 \\ 0 \\ 1 \end{Bmatrix}$	$\begin{Bmatrix} 0 \\ 0 \\ 1 \end{Bmatrix}$

the angle between the direction of the contact point and the direction of the normal,

$$\min_{\omega, \eta} ((f(\eta, \omega) \cdot n_{surf})^2 - 1), \quad (42)$$

which will yield the angular values to locate those points in the parameterized equation of the supersurface, $f(\eta, \omega)$. Figure 11 shows the normals for two different sections of a supersurface. Notice that, for some values of the exponents, there are more solutions for parallel grasping with collinear normals not passing through the origin, but in general they don't fulfill the minimum wrench condition.

In the particular case of a vertical grasping of a superellipsoid, for instance, the angular parameter of the approximate antipodal points in the plane reduces to

$$\alpha_{grasp} = \min_{\eta} \left(\frac{a_1 a_3 \cos^{2\epsilon_1} \eta \sin^{2\epsilon_1} \eta}{\sqrt{a_3^2 \cos^{2\epsilon_1+4} \eta \sin^{4\epsilon_1} \eta + a_1^2 \cos^{4\epsilon_1} \eta \sin^{2\epsilon_1+4} \eta}} - 1 \right). \quad (43)$$

For concave supersurfaces, with exponents greater than 2, the negative normal curvature along a line helps in making the grasp closed, and antipodal points are found using the same process.

For convex supersurfaces, or those with negative Gaussian curvature - such as the superellipsoid-, enforcing the condition of minimum curvature implies selecting the curvature that needs to be used.

Normal curvatures along any curve can be defined; for us, it is easier to define them along the coordinate curves. The minimum and maximum normal curvatures are defined for the principal directions, which do not always

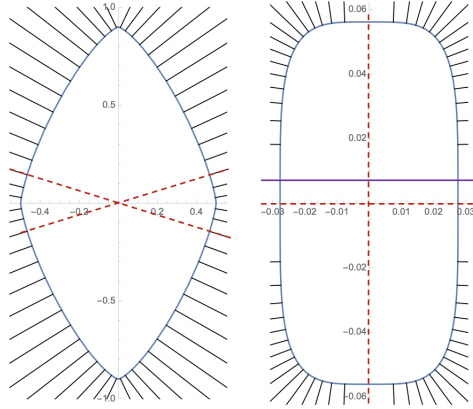


Figure 11: Normals to the surface of the superellipse for $\epsilon_1 = 1.5$, left, and $\epsilon_1 = 0.49$, right. Normal lines for antipodal points intersecting the origin (dotted lines). Some values of ϵ yield more collinear normals, but they don't fulfill the minimum wrench condition (solid line).

coincide with the coordinate curves given by the angles ω and η . Another option is to use the mean curvature; see [5] for details on this discussion.

With these criteria, [37] presents the selection of grasping points for a superellipse shown in Table 5.3. More complete grasping point selection is presented in Tables 3 and 4.

Table 2: Position vector to contact points as a function of the length of the semi-axis and the exponent ϵ ([37]).

Exponent	Dimensions	Position vector p_c
$\epsilon < 1$	$a_1 > a_2$	$p_c = (0, a_2, 0, 1)$
	$a_1 = a_2$	$p_c = (a_1, 0, 0, 1)$ or $p_a = (0, a_2, 0, 0, 1)$
	$a_1 < a_2$	$p_c = (a_1, 0, 0, 1)$
$\epsilon = 1$	$a_1 > a_2$	$p_c = (0, a_2, 0, 1)$
	$a_1 = a_2$	$p_c = (a_1 \cos \omega, a_1 \sin \omega, 0, 1)$ for all ω
	$a_1 < a_2$	$p_c = (a_1, 0, 0, 1)$
$\epsilon > 1$	$a_1 = a_2$	$p_c = (a_1/\sqrt{2}, a_1/\sqrt{2}, 0, 1)$
	$a_1 \neq a_2$	$p_c(\omega)$ as per Eq.(43)

5.4. Grasping strategies for supertoroids

The normal lines and curvature are similar to those of the superellipsoid, and have been studied in detail in [5]. In many cases, the best grasping points correspond to seams and stitches of the surface when we divide it in the 16 patches defined by the values of the angles between 0 and $\pi/2$. For most values of the exponent, those points are regular points, see Figure 4, and their curvature and normal directions can be easily pre-calculated.

The heuristics for grasping supertoroids are based on the following desired outcomes:

- The grasp must be close to the centroid to minimize moments.
- The contact must be at points of minimum positive curvature to increase the surface of contact.
- For pinching grippers, the separation between contact points must be less than the width of the gripper
- For pinching grippers, contact points must be such that the normal directions to the surface are collinear for both fingers.
- For suction grippers, the area around the contact point must be greater than the area of the suction cup.

In addition, we have the option of internal grasping, and of using the negative Euler curvature of the surface.

Figure 12 presents the decision tree for the grasping process. Notice that we assume $a_1 > a_2$, which can always be done when we assign local axes to the superquadric.

The simple criteria presented here are easy to change in order to add or modify some of the heuristics, or to be transformed into a learning-based decision tree; other classifiers may be added. When more than one option is feasible, these can be ranked and prioritized.

Figure 12 directs to several tables, which were developed in [24]. Here we present Table 3 and 4 as examples of the application of the heuristics for the two types of grippers considered.

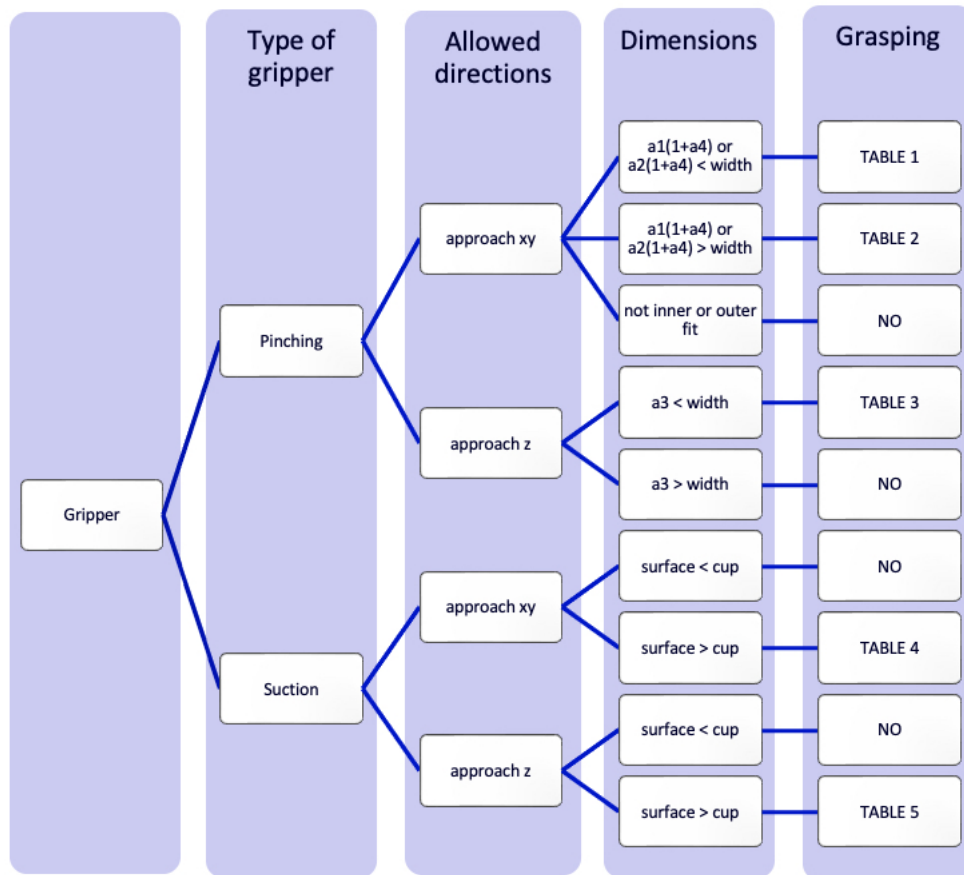








Figure 12: Decision tree for the grasping process. Coordinates x, y, z of the local super-toroid frame.

Table 3: Grasping TABLE 1: Grasping with a parallel finger gripper, approach along x y plane

ϵ_1	ϵ_2	a_1, a_2	Grasping point p_c	View
$\epsilon_1 < 1$	$\epsilon_2 < 1$	$a_1 > a_2$	$p_c = \begin{Bmatrix} a_1(1 + a_4) \\ 0 \\ 0 \end{Bmatrix}$	
		$a_1 = a_2$	$p_c = \begin{Bmatrix} a_1(1 + a_4) \\ 0 \\ 0 \end{Bmatrix}, \begin{Bmatrix} 0 \\ a_1(1 + a_4) \\ 0 \end{Bmatrix}$	
	$\epsilon_2 = 1$	$a_1 > a_2$	$p_c = \begin{Bmatrix} a_1(1 + a_4) \\ 0 \\ 0 \end{Bmatrix}$	
		$a_1 = a_2$	$p_c = \begin{Bmatrix} a_1(1 + a_4) \\ 0 \\ 0 \end{Bmatrix}, \begin{Bmatrix} 0 \\ a_1(1 + a_4) \\ 0 \end{Bmatrix}$	
	$\epsilon_2 > 1$	$a_1 > a_2$	$p_c = 2^{-\epsilon_2} \begin{Bmatrix} a_1(1 + a_4) \\ 3^{\frac{\epsilon_2}{2}} a_2(1 + a_4) \\ 0 \end{Bmatrix}$	
		$a_1 = a_2$	$p_c = \begin{Bmatrix} a_1(1 + a_4) \\ 0 \\ 0 \end{Bmatrix}, \begin{Bmatrix} 0 \\ a_1(1 + a_4) \\ 0 \end{Bmatrix}$	
$\epsilon_2 = 1$	etc			





5.5. Implementation and combined results

The superquadric fitting implementation and grasping can be found in https://github.com/jontromanab/sq_grasp.git, and the supertoroid fitting in <https://github.com/jbadiat/SupertoroidFitting.git>. Grasping results for different objects can be found in [37] and [24].

The initial conditions for the extrinsic parameters are obtained using PCA. For the parameters a_i , the eigenvalues are selected as initial values, while we constrain ϵ_1 and ϵ_2 to be between 0.1 and 1.9. The parameters are optimized using several numerical methods, mainly Levenberg-Marquardt [50].

As the supersurface grasping system does not rely on iteration for finding grasp to execute, it may provide better results in time efficiency, with of

Table 4: Grasping TABLE 5: Grasping with a suction cup, approach along the z axis

ϵ_2	ϵ_1	a_1, a_2	Grasping point p_c	View
$\epsilon_2 < 1$	$\epsilon_1 = 1$	$a_1 > a_2$	$p_c = \begin{Bmatrix} 0 \\ a_2 a_4 \\ a_3 \end{Bmatrix}$	
		$a_1 = a_2$	$p_c = \begin{Bmatrix} a_1 a_4 \\ 0 \\ a_3 \end{Bmatrix}, \begin{Bmatrix} 0 \\ a_2 a_4 \\ a_3 \end{Bmatrix}$	
	$\epsilon_1 > 1$	$a_1 > a_2$	$p_c = \begin{Bmatrix} 0 \\ a_2((\frac{1}{2^{1/2}})^{\epsilon_1} + a_4) \\ 2^{\epsilon_1/2} a_3 \end{Bmatrix}$	
		$a_1 = a_2$	$p_c = \begin{Bmatrix} a_1((\frac{1}{2^{1/2}})^{\epsilon_1} + a_4) \\ 0 \\ 2^{\epsilon_1/2} a_3 \end{Bmatrix}, \begin{Bmatrix} 0 \\ a_2((\frac{1}{2^{1/2}})^{\epsilon_1} + a_4) \\ 2^{\epsilon_1/2} a_3 \end{Bmatrix}$	
$\epsilon_2 = 1$	etc			

maximum force balance and stability.

6. Intersection and contact for supersurfaces

6.1. Contact detection between superquadrics

Superquadrics can also be used for modeling in simulation, including dynamic simulation. In this case, it is important to be able to detect contact between superquadrics. This is the object of study of [47].

They formulate the problem as a nonlinear, constrained optimization problem and solve it numerically. In order to do so, local coordinates are transformed to a global frame, and the distance between one point of each superquadric, \mathbf{r}_A and \mathbf{r}_B , is created and minimized,

$$\|\mathbf{r}_B - \mathbf{r}_A\|^2 = (X_B - X_A)^2 + (Y_B - Y_A)^2 + (Z_B - Z_A)^2, \quad (44)$$

subject to the exact constraints given by the two implicit equations of the superquadrics, with the point coordinates transformed to the global frame.

Then, in order to recognize contact of the minimum point, the dot product of the difference vector with the normal to the surface is created,

$$\mathbf{n}_A \cdot \mathbf{r}_{AB} \leq 0, \quad (45)$$

where there is contact when the dot product is less or equal to zero.

6.2. Non-iterative contact approximation

The minimum distance between superquadrics is always between two points that share a common normal. This can be exploited for collision detection using iterative methods [62] as well as non-iterative approximations. The main benefit of using non-iterative methods is that their implementation can be much simpler and compatible with trajectory optimization problem formulations.

One such approximation that relies on this type of method allows for the approximation of the contact between a mesh and a superquadric.

For shallow contacts, that is, those in which none of the elements experiences significant deformation, the general shape of the volume shared by the two objects can be approximated. To do so, the radial distances of the points of the mesh with respect to the superquadric are calculated. During this process, a value of ω_s is determined for each point by first finding ω_π and then applying the relation shown in Eq. 16. This can be used to find the projection of the point on the superquadric, the distance between the two being the radial distance. Additionally, the normal to the superquadric at the projected point, $n(\omega_s)$, is taken as the direction of the contact force. The normal direction, $n(\omega_s)$, is defined by the relation in Eq. 39.

If the contact is treated as linearly elastic, the pressure at each point is a factor of the material properties of the bodies involved and the indentation of the mesh. In this context, indentation (I_p) is understood as a measure of how deep a point is in the superquadric. I_p has both a magnitude and a direction. For our purposes, it can be determined using Eq. 46.

$$I_p = (p_R \cdot n(\omega_s))n(\omega_s) \quad (46)$$

Thus, the indentation of a point in a superquadric is the vector from the point to its projection on the superquadric projected on the normal associated to its ω_s . The point of application of the contact force depends on the relative stiffness of the bodies. If the meshed body is far stiffer than the superquadric,

the point of application is the point on the mesh. Whereas if the opposite is true, the point on the mesh plus the indentation can be taken instead.

A graphical representation of the procedure is shown in Figure 13. In it, the dotted lines are the distance from the mesh points to the center of the superellipse, the dashed lines are the radial distance to each point, and the arrows show the magnitude and direction of the indentation of each point. The cross markers on the superellipse show the points where each normal is used.

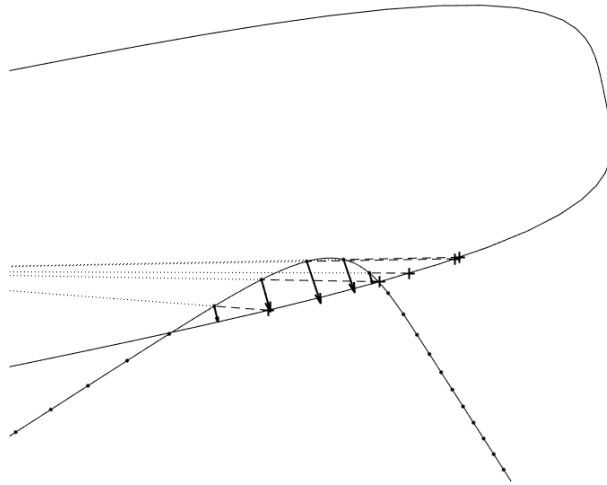


Figure 13: Contact between meshed and unmeshed superellipses. In this case, the meshed superellipse is considered as much stiffer than the unmeshed one.

The indentation can then be used for calculating a pressure associated to each point. The pressure on each point multiplied by an area corresponding to it can in turn be used to compute a set of wrenches acting on the bodies.

By examining the definition of the radial distance it becomes clear that the closer a point of the mesh is to the surface, the more accurate this approximation becomes. Reducing the distance between the point in the mesh to the surface of the superquadric to zero produces an indentation of exactly zero magnitude and direction perfectly normal to the superquadric. The error grows as the distance between the two increases, particularly in elongated bodies, in which the directions of $n(\omega_s)$ and p_s p can be especially dissimilar.

7. Conclusions

This work is intended as an overview on the modeling of shapes using the many variations of superquadrics, originally devised as quadric surfaces whose shapes could be modified by substituting the original quadratic exponents with somewhat arbitrary positive real numbers. Based on this background, some new results on the definition and use of superquadrics and superquartics are presented, for grasping, obstacle and contact detection. The use of simpler equations limits the shapes that they can model but leads to closed expressions for some of the basic characteristics of these surfaces, such as normal lines, curvature and distances to points. These expressions can be used to create algorithms and heuristics for actions in the interaction with real objects.

References

- [1] Hyperquadrics: Smoothly deformable shapes with convex polyhedral bounds. *Computer Vision, Graphics, and Image Processing* **44**(2), 191–210 (1988)
- [2] Extending superquadrics with exponent functions: Modeling and reconstruction. *Graphical Models* **63**(1), 1–20 (2001)
- [3] Alaniz, S., Mancini, M., Akata, Z.: Iterative superquadric recombination of 3d objects from multiple views. In: *International Conference on Computer Vision*. IEEE (2023)
- [4] B. S. Zapata-Impata P. Gil, J.P., Torres, F.: Fast geometry-based computation of grasping points on three-dimensional point clouds. *International Journal of Advanced Robotic Systems* **16**(1) (2019). DOI 10.1177/1729881419831846
- [5] Badia Torres, J., Carmona, E., Makhal, A., Heidari, O., Perez Gracia, A.: Supertoroid fitting of objects with holes for robotic grasping and scene generation (2024)
- [6] Barr, A.H.: Superquadrics and angle-preserving transformations. *IEEE Computer Graphics and Applications* **1**(1), 11–23 (1981). DOI 10.1109/MCG.1981.1673799

- [7] Barr, A.H.: Global and local deformations of solid primitives. *Computer Graphics* **18**(3), 21–30 (1984)
- [8] Beirinckx, B.: From superellipses to superformula and technology. In: *Proceedings of the 1st International Symposium on Square Bamboos and the Geometree (ISSBG 2022)*, pp. 123–142 (2023)
- [9] Bennamoun, M., Mamic, G.: *Object recognition: fundamentals and case studies*. Springer-Verlag London (2002)
- [10] Bhabhrawala, T., Krovi, V.: Shape recovery from medical image data using extended superquadrics. pp. 479–486 (2005)
- [11] Biegelbauer, G., Vincze, M.: Efficient 3d object detection by fitting superquadrics to range image data for robot’s object manipulation. In: *2007 IEEE International Conference on Robotics and Automation (ICRA)*. Roma, Italy (2007)
- [12] Bohg, J., Johnson-Roberson, M., León, B., Felip, J., Gratal, X., Bergström, N., Kragic, D., Morales, A.: Mind the gap - robotic grasping under incomplete observation. In: *2011 IEEE International Conference on Robotics and Automation*, pp. 686–693 (2011). DOI 10.1109/ICRA.2011.5980354
- [13] Boulton, T.E., Gross, A.D.: Recovery Of Superquadrics From 3-D Information. In: D.P. Casasent, E.L. Hall (eds.) *Intelligent Robots and Computer Vision VI*, vol. 0848, pp. 358 – 365. International Society for Optics and Photonics, SPIE (1988). DOI 10.1117/12.942759
- [14] Chen, I.M., Burdick, J.: Finding antipodal point grasps on irregularly shaped objects. *IEEE Transactions on Robotics* **9**(4) (2002)
- [15] Chevalier, L., Jaillet, F., Baskurt, A.: 3d shape coding with superquadrics. In: *Proceedings 2001 International Conference on Image Processing (Cat. No.01CH37205)*, vol. 2, pp. 93–96 vol.2 (2001). DOI 10.1109/ICIP.2001.958432
- [16] Chevalier, L., Jaillet, F., Baskurt, A.: Segmentation and superquadric modeling of 3d objects. In: *International Conference in Central Europe on Computer Graphics and Visualization* (2003)

- [17] Chuang, J.H., Ahuja, N., Lin, C.C., Tsai, C.H., Chen, C.H.: A potential-based generalized cylinder representation. *Computers & Graphics* **28**(6), 907–918 (2004)
- [18] Chuang, J.H., Ahuja, N., Lin, C.C., Tsai, C.H., Chen, C.H.: A potential-based generalized cylinder representation. *Computers & Graphics* **28**(6), 907–918 (2004)
- [19] Corey Goldfeder Matei Ciocarlie, H.D., Allen, P.K.: The columbia grasp database. In: 2009 International Conference on Robotics and Automation (ICRA) (2009)
- [20] Fischinger, D., Weiss, A., Vincze, M.: Learning grasps with topographic features. *The International Journal of Robotics Research* **34**(9), 1167–1194 (2015). DOI 10.1177/0278364915577105
- [21] Fischler, M.A., Bolles, R.C.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* **24**(6), 381–395 (1981). DOI 10.1145/358669.358692
- [22] Fontanals, J., Dang-Vu, B.A., Porges, O., Rosell, J., Roa, M.: Integrated grasp and motion planning using independent contact regions. In: 2014 IEEE-RAS International Conference on Humanoid Robots (2014). DOI 10.1109/humanoids.2014.7041469
- [23] Fougerolle, Y.D., Gribok, A., Foufou, S., Truchetet, F., Abidi, M.A.: Rational supershapes for surface reconstruction. In: D. Fofi, F. Meriaudeau (eds.) Eighth International Conference on Quality Control by Artificial Vision, vol. 6356, p. 63560M. International Society for Optics and Photonics, SPIE (2007). DOI 10.1117/12.736916
- [24] Garrido Reus, L.: *Agarre robòtic con ajuste supercuàrtic*. Ph.D. thesis, UPC, Escola Tècnica Superior d’Enginyeria Industrial de Barcelona, Departament d’Enginyeria Mecànica (2022)
- [25] Gielis, J.: A generic geometric transformation that unifies a wide range of natural and abstract shapes. *American Journal of Botany* **90**(3), 333–338

- [26] Gualtieri, M., ten Pas, A., Saenko, K., Platt, R.: High precision grasp pose detection in dense clutter. In: 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 598–605 (2016)
- [27] Jaklič, A., Leonardis, A., Solina, F.: Segmentation and Recovery of Superquadrics: Computational Imaging and Vision. Kluwer Academic Publishers, Norwell, MA, USA (2000)
- [28] Jaklič, A., Leonardis, A., Solina, F.: Segmentation and Recovery of Superquadrics: Computational Imaging and Vision. Kluwer Academic Publishers, Norwell, MA, USA (2000)
- [29] Jia, Y.: Curvature-based computation of antipodal grasps. In: 2002 IEEE International Conference on Robotics and Automation (2002)
- [30] K. Duncan S. Sarkar, R.A., Dubey, R.: Multi-scale superquadric fitting for efficient shape and pose recovery of unknown objects. In: 2013 IEEE International Conference on Robotics and Automation (ICRA). Karlsruhe, Germany (2013)
- [31] Kazhdan, M., Funkhouser, T., Rusinkiewicz, S.: Rotation invariant spherical harmonic representation of 3 d shape descriptors. In: Symposium on geometry processing, vol. 6, pp. 156–164 (2003)
- [32] Kim, S., Ahn, T., Lee, Y., Kim, J., Wang, M.Y., Park, F.C.: Dsqnet: A deformable model-based supervised learning algorithm for grasping unknown occluded objects. *IEEE Transactions on Automation Science and Engineering* **20**(3), 1721 (2023)
- [33] Kroemer, O., Amor, H.B., Ewerton, M., Peters, J.: Point cloud completion using extrusions. In: 2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012), pp. 680–685 (2012). DOI 10.1109/humanoids.2012.6651593
- [34] L. Baronti, M.C.e.a.: Primitive shape fitting in point clouds using the bees algorithm. *Applied Sciences* **9**(23) (2019). DOI 10.3390/app9235198
- [35] Leonardis, A., Solina, F., Macerl, A.: A direct recovery of superquadric models in range images using recover-and-select paradigm, pp. 309–318.

Springer Berlin Heidelberg, Berlin, Heidelberg (1994). DOI 10.1007/3-540-57956-7_35

- [36] Liu, W., Wu, Y., Ruan, S., Chirikjian, G.S.: Marching-primitives: Shape abstraction from signed distance function. In: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 8771–8780. IEEE Computer Society (2023). DOI 10.1109/CVPR52729.2023.00847
- [37] Makhal, A., Thomas, F., Gracia, A.P.: Grasping unknown objects in clutter by superquadric representation. In: 2018 Second IEEE International Conference on Robotic Computing (IRC). Laguna Hills, CA, USA (2018)
- [38] Marton, Z.C., Pangercic, D., Blodow, N., Kleinehellefort, J., Beetz, M.: General 3d modelling of novel objects from a single view. In: 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3700–3705 (2010). DOI 10.1109/IROS.2010.5650434
- [39] Menendez, E., Martinez, S., Díaz-de María, F., Balaguer, C.: Integrating egocentric and robotic vision for object identification using siamese networks and superquadric estimations in partial occlusion scenarios. *Biomimetics* **9**, 100 (2024). DOI 10.3390/biomimetics9020100
- [40] Miller, A.T., Allen, P.K.: Graspit! a versatile simulator for robotic grasping. *IEEE Robotics Automation Magazine* **11**(4), 110–122 (2004). DOI 10.1109/MRA.2004.1371616
- [41] Mousa, M.H., Chaine, R., Akkouche, S., Galin, E.: Efficient spherical harmonics representation of 3d objects. In: *Computer Graphics and Applications, 2007. PG '07. 15th Pacific Conference on*, pp. 248–255 (2007)
- [42] Ohuchi, M., Saito, T.: Three-dimensional shape modeling with extended hyperquadrics. In: *Proceedings Third International Conference on 3-D Digital Imaging and Modeling*, pp. 262–269 (2001). DOI 10.1109/IM.2001.924449
- [43] ten Pas, A., Gualtieri, M., Saenko, K., Platt, R.: Grasp pose detection in point clouds. *The International Journal of Robotics Research* **36**(13-14), 1455–1473 (2017). DOI 10.1177/0278364917735594

- [44] Paschalidou, D., Ulusoy, A.O., Geiger, A.: Superquadrics revisited: Learning 3d shape parsing beyond cuboids (2019)
- [45] Pascoal, R., Santos, V., Premebida, C., Nunes, U.: Simultaneous segmentation and superquadrics fitting in laser-range data. *IEEE Transactions on Vehicular Technology* **64**(2), 441–452 (2015). DOI 10.1109/TVT.2014.2321899
- [46] Pilu, M., Fisher, R.B.: Equal-distance sampling of superellipse models. In: *Proceedings of the 1995 British Conference on Machine Vision (Vol. 1)*, BMVC '95, pp. 257–266. BMVA Press, Surrey, UK, UK (1995)
- [47] Portal, R.F., Dias, J.P., de Sousa, L.G.: Contact detection between convex superquadric surfaces. *The Archive of Mechanical Engineering* **57**(2) (2010)
- [48] Przybylski, M., Asfour, T., Dillmann, R.: Unions of balls for shape approximation in robot grasping. In: *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1592–1599 (2010). DOI 10.1109/IROS.2010.5653520
- [49] Quispe, A.H., Milville, B., Gutiérrez, M.A., Erdogan, C., Stilman, M., Christensen, H., Amor, H.B.: Exploiting symmetries and extrusions for grasping household objects. In: *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3702–3708 (2015). DOI 10.1109/ICRA.2015.7139713
- [50] Ranganathan, A.: *The Levenberg-Marquardt Algorithm* (2004)
- [51] Rusu, R.B., Cousins, S.: 3D is here: Point Cloud Library (PCL). In: *IEEE International Conference on Robotics and Automation (ICRA)*. Shanghai, China (2011)
- [52] Saxena, A., Driemeyer, J., Ng, A.Y.: Robotic grasping of novel objects using vision. *Int. J. Rob. Res.* **27**(2), 157–173 (2008). DOI 10.1177/0278364907087172
- [53] Shapiro, L.G., Moriarty, J.D.: A generalized blob model for three-dimensional object representation. In: *Proceedings of the Second IEEE Workshop on Data Description and Management*, pp. 109–116 (1980)

- [54] Sircelj, J., Oblak, T., Grm, K., Petkovic, U., Jakli, A., Peer, P., Struc, V., Solina, F.: Segmentation and recovery of superquadric models using convolutional neural networks. ArXiv [abs/2001.10504](https://arxiv.org/abs/2001.10504) (2020)
- [55] Solina, F., Bajcsy, R.: Recovery of parametric models from range images: the case for superquadrics with global deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **12**(2), 131–147 (1990). DOI 10.1109/34.44401
- [56] Stein, S.C., Schoeler, M., Papon, J., Wörgötter, F.: Object partitioning using local convexity. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition, pp. 304–311 (2014). DOI 10.1109/CVPR.2014.46
- [57] Subrahmonia, J., Cooper, D.B., Keren, D.: Practical reliable bayesian recognition of 2d and 3d objects using implicit polynomials and algebraic invariants. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **18**(5), 505–519 (1996)
- [58] Subrahmonia, J., Cooper, D.B., Keren, D.: Practical reliable bayesian recognition of 2d and 3d objects using implicit polynomials and algebraic invariants. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **18**(5), 505–519 (1996). DOI 10.1109/34.494640
- [59] Taubin, G.: Estimation of planar curves, surfaces, and nonplanar space curves defined by implicit equations with applications to edge and range image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **13**(11), 1115–1138 (1991). DOI 10.1109/34.103273
- [60] Terzopoulos, D., Witkin, A., Kass, M.: Symmetry-seeking models and 3d object reconstruction. *International Journal of Computer Vision* **1**(3), 211–221 (1988). DOI 10.1007/BF00127821
- [61] Tomašević, D., Peer, P., Solina, F., Jaklič, A., Štruc, V.: Reconstructing superquadrics from intensity and color images. *Sensors* **22**(14) (2022). DOI 10.3390/s22145332
- [62] Wellmann, C., Lillie, C., Wriggers, P.: A contact detection algorithm for superellipsoids based on the common-normal concept. *Engineering Computations* **25** (2008). DOI 10.1108/02644400810881374

- [63] Whaite, P., Ferrie, F.: From uncertainty to visual exploration. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **13**(10), 1038–1049 (1991)
- [64] Țălu, : Generation of 3d shapes with superellipsoids, supertoroids, supercylinders and super cones. *Journal of Engineering Studies and Research* **18** (2017). DOI 10.29081/jesr.v18i2.230