A Comparative Study of Optimization Models for Superellipsoid Fitting in 3D Object

Maria Silva[1], Fernanda Costa[1,2], and Eliana Costa e Silva[3,4(✉)]

[1] Department of Mathematics, University of Minho, Braga, Portugal
[2] Center of Mathematics (CMAT), University of Minho, Braga, Portugal
[3] CIICESI, ESTG, Instituto Politécnico do Porto, Felgueiras, Portugal
eos@estg.ipp.pt
[4] Algoritmi Center, University of Minho, Guimarães, Portugal

Abstract. This work presents an in-depth analysis of superquadrics and their application in the modeling of three-dimensional objects. The study focuses on object modeling from point cloud data, a key problem in robotics and computer vision. Two optimization models are proposed, based on implicit surface fitting and volume minimization, respectively. Both formulations are solved using the SLSQP method. Results are presented through graphical visualizations, correlation analysis for different upper limits of the upper bound of the parameters of the superquadrics, followed by a comparative analysis using volume-based metrics to assess the quality of fit. The results show that, for the studied objects, the best results are obtained for the volume-based model. Further, the influence of upper limits is significant for the implicit surface fitting model but not for the volume-based. These findings reinforce that the volume-based formulation offers a more robust and reliable approach for superellipsoid fitting across the tested objects.

Keywords: Superellipsoids · Optimization · SLSQP · 3D Modeling

1 Introduction

Three-dimensional (3D) object representation is a fundamental problem in several scientific and technological domains, including robotics, computer vision, and computer graphics. The ability to accurately model object geometry is essential for enabling autonomous systems to interact effectively with the physical world. In robotics, this capability is particularly relevant for tasks such as object manipulation, grasp planning, navigation, and collision avoidance, where the quality of geometric representations directly impacts performance and safety.

Robotic systems typically rely on sensor data, such as point clouds acquired from depth cameras or LiDAR, to perceive their environment. Converting this raw data into compact and meaningful geometric models remains a challenging task. On one hand, dense representations such as meshes or raw point clouds

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2027
O. Gervasi et al. (Eds.): ICCSA 2026 Workshops, LNCS 16763, pp. 241–254, 2027.
https://doi.org/10.1007/978-3-032-30530-5_15

provide high fidelity but are computationally expensive. On the other hand, simpler geometric primitives offer efficiency but often lack the flexibility required to accurately describe complex shapes. Therefore, there is a need for representations that balance expressiveness and computational tractability.

Superquadrics have emerged as a promising solution to this problem. As a family of parametric surfaces, superquadrics generalize classical quadrics by introducing shape parameters that control curvature and surface sharpness. This allows them to represent a wide range of geometries—from smooth, rounded objects to more angular and box-like shapes—using a relatively small number of parameters. In addition to shape, superquadrics can encode position, orientation, and scale, making them suitable for modeling objects in arbitrary configurations within a 3D space.

Their compact representation and analytical formulation make superquadrics particularly attractive for robotics applications. Tasks such as collision detection, distance computation, and grasp planning can benefit from the mathematical properties of superquadrics, which enable efficient geometric reasoning. Moreover, their parametric nature facilitates optimization-based fitting to sensor data, allowing objects to be reconstructed directly from point clouds.

Despite these advantages, most existing approaches have focused primarily on superellipsoids, a specific subclass of superquadrics. While effective for modeling convex objects, superellipsoids are limited in their ability to represent more complex geometries, such as objects with holes or non-convex features. This limitation motivates the exploration of additional superquadric forms, including supertoroids and superhyperboloids, which provide greater flexibility in shape representation.

In this work, we aim to extend the use of superquadrics for 3D object modeling by exploring a broader family of shapes and developing optimization-based methods for fitting these models to point cloud data. The proposed approach seeks to improve both the accuracy and generality of superquadric-based representations, enabling their application in more complex and realistic scenarios.

2 Related Work

The problem of 3D object representation has been widely studied, with approaches ranging from explicit geometric models to data-driven methods. Traditional representations include polygonal meshes and voxel grids, which provide detailed descriptions of object geometry but often incur high computational costs. More compact representations based on geometric primitives, such as spheres, cylinders, and ellipsoids, have been proposed to address this issue, although they typically lack the flexibility required for complex shapes.

Superquadrics were introduced as a generalization of quadric surfaces, providing a unified framework for representing a wide variety of shapes using a small set of parameters [2]. Early work demonstrated their effectiveness in computer graphics and vision, particularly for object recognition and modeling tasks. Their ability to smoothly interpolate between different geometric forms makes them a versatile tool for shape approximation.

In robotics, superquadrics have been applied to object modeling and scene understanding using point cloud data. For instance, Costa e Silva et al. [8] demonstrated the use of superquadrics to model objects in robotic workspaces, showing improvements over simpler representations such as ellipsoids in terms of both accuracy and compactness. Their approach highlighted the potential of superquadrics for tasks such as grasp planning and collision detection. However, their study was limited to superellipsoids, restricting the range of shapes that could be effectively modeled.

Subsequent research has explored optimization techniques for fitting superquadrics to data. These methods typically formulate the fitting problem as a non-linear optimization task, where the objective is to minimize the distance between the model surface and the observed data points [7]. Common approaches include least-squares formulations and variants based on implicit surface representations. While effective, these methods can be sensitive to initialization and may struggle with noisy or incomplete data.

Other works have investigated alternative objective functions and constraints to improve robustness. For example, volume-based metrics have been proposed to ensure that the fitted superquadric provides a tight approximation of the object while avoiding excessive overestimation [10]. These approaches aim to balance accuracy with stability in the optimization process.

Despite these advances, the literature still lacks comprehensive studies that consider a wider range of superquadric families, such as supertoroids and superhyperboloids, in a unified framework. These shapes offer additional modeling capabilities, particularly for objects with non-convex geometries or complex topologies. Furthermore, there is a need for methodologies that can generalize across different object types and operate effectively under real-world conditions, including noise, occlusions, and varying sensor quality.

The present work addresses these gaps by systematically exploring multiple superquadric forms and comparing different optimization strategies for fitting them to point cloud data. By extending beyond superellipsoids and incorporating more expressive shape families, this study aims to advance the state of the art in superquadric-based object modeling and contribute to the development of more capable robotic perception systems.

Superquadrics stand out in the literature as a versatile and efficient mathematical approach for representing a wide range of three-dimensional geometric shapes [5,7,8]. Due to their closed-form equations, they provide a compact and computationally efficient description, which makes them particularly relevant in areas such as robotics, computer vision, and object modeling.

Studies such as those by [8] and [7] demonstrate the applicability of variants such as superellipsoids, super hyperboloids, and supertoroids, highlighting the advantages of implicit parameterization, which facilitates both modeling and subsequent geometric analysis. In turn, in [5] the authors propose a generalization to hyperquadrics, increasing the representation capability of shapes, albeit with greater mathematical complexity.

However, traditional superquadrics present limitations, especially when applied to modeling objects with irregular contours or non-convex geometries. They are particularly well-suited to representing symmetric or regular-shaped objects but struggle to handle structures that contain cavities or holes [5,7,8].

In order to overcome these limitations, [11] proposes the use of supertoroids—a fourth-degree generalization of superquadrics—capable of modeling objects with a single hole. Their approach allows for a straightforward derivation of the intrinsic and extrinsic parameters of the surface, based on direct expressions for the major and minor radii, while maintaining a computationally efficient distance function. This advance significantly expands the range of shapes that can be geometrically modeled, showing promise for applications in robotic manipulation and complex scene generation.

Additionally, [6] demonstrates the practical applicability of superquadrics in real-world scenarios, proposing a fast and effective method for grasping unknown objects in unstructured environments. The approach is based on partial reconstruction of objects from incomplete views, assuming symmetry to generate approximate models to which superquadrics are fitted. Using the obtained geometric parameters, the system determines the contact points and the optimal approach direction for an optimized antipodal grasp. Tests on PR2 (Personal Robots 2) robots showed that this solution, despite using simplified representations, outperforms algorithms based on neural networks in terms of temporal efficiency and accuracy in cluttered environments.

In parallel, the automation of the superquadric fitting process has been explored using deep learning techniques. [9] presents an approach based on convolutional neural networks, operating in two stages: in a first step, a Mask R-CNN (Region-based Convolutional Neural Network) architecture identifies regions with structures compatible with superquadrics in depth scenes; in a second step, a CNN regressor (Convolutional Neural Network) fits the parametric models to the segmented regions. This method has proven effective on both synthetic and real data, enabling the description of complex shapes with a reduced set of interpretable parameters and with significantly lower processing time compared to conventional techniques.

This evolution, from classical formulations to practical applications and automated neural network-based approaches, highlights the growing importance of superquadrics in modern robotics, both for perception and representation tasks, as well as for physical actions such as grasping.

3 Methodology

This section presents the methodology adopted for modeling three-dimensional objects using superquadrics. Motivated by the need for accurate and efficient object representations in robotic applications, the proposed approach focuses on fitting superquadric surfaces, more specifically superellipsoids, to point clouds with high accuracy, a key requirement for perception, collision detection, and object manipulation.

For modeling three-dimensional objects using superquadrics two nonlinear optimization models are considered. Model 1, based on the approach proposed in [8], employs the inside–outside function as the objective. Model 2, estimates the parameters of the superellipsoid using a volume-based objective function.

The optimization method used to solve both models is presented in Subsect. 3.3.

3.1 Model 1

In the first model, the objective function is based on the implicit formulation of the superquadric, inspired by the approach proposed in [8]. Specifically, the objective function is defined as the mean fitting error of each point relative to the surface, multiplied by a volume based normalization term. This normalization prevents the objective function from favoring solutions with disproportionate scales, effectively introducing a regularization term proportional to the approximate volume of the fitted superquadric. The imposed constraints ensure that all N data points lie inside or on the superquadric surface.

The exact formulation of the optimization problem is:

$$\min_{\lambda \in \mathbb{R}^5} \; F(\boldsymbol{\lambda}) = \frac{\sqrt{\lambda_0 \lambda_1 \lambda_2}}{N} \sum_{i=1}^{N} \left[f^{\frac{\lambda_3}{2}}(\boldsymbol{\lambda}, \mathbf{C}_i) - 1 \right]^2 \tag{1}$$

$$\text{s.t.} \;\; f(\boldsymbol{\lambda}, \mathbf{C}_i) \leq 1, \quad \forall i = 1, \ldots, N. \tag{2}$$

$$\boldsymbol{\lambda}^{(L)} \leq \boldsymbol{\lambda} \leq \boldsymbol{\lambda}^{(U)}, \tag{3}$$

with

$$f(\boldsymbol{\lambda}, \mathbf{C}_i) = \left(\left(\frac{x}{\lambda_0} \right)^{\frac{2}{\lambda_4}} + \left(\frac{y}{\lambda_1} \right)^{\frac{2}{\lambda_4}} \right)^{\frac{\lambda_4}{\lambda_3}} + \left(\frac{z}{\lambda_2} \right)^{\frac{2}{\lambda_3}},$$

where $\boldsymbol{\lambda}$ is the parameter vector of the superellipsoid. Specifically, λ_0, λ_1, and λ_2 are the scale parameters along the x, y, and z axes, respectively, λ_3 and λ_4 are the shape parameters that control the "roundness" or "squareness" of the surface.

3.2 Model 2

The second model, aims at fitting a superquadric that encloses the point cloud, which minimizes the volume of the enclosing superquadric. Therefore, the constraints are the same as in Model 1 but the objective function is given as the volume of the superquadric. Specifically, the objective function is:

$$F(\boldsymbol{\lambda}) = 2\lambda_0 \lambda_1 \lambda_2 \lambda_3 \lambda_4 \, B\left(\tfrac{\lambda_3}{2} + 1, \lambda_3\right) \, B\left(\tfrac{\lambda_4}{2}, \tfrac{\lambda_4}{2}\right), \tag{4}$$

where $B(x, y)$ is the beta function defined as:

$$B(x, y) = \int_0^1 t^{x-1}(1-t)^{y-1} \, dt.$$

This formulation makes it possible to find the superellipsoid with the smallest possible volume that still satisfies the condition of containing all the points, resulting in more compact solutions with restricted volume, which can be particularly useful in applications that require minimal boundaries or spatial optimization

3.3 The SLSQP Method

The Sequential Least Squares Quadratic Programming (SLSQP) algorithm was originally developed by Dieter Kraft in the 1980s [4], based on the sequential quadratic programming (SQP) method. SLSQP is particularly well suited for nonlinear constrained optimization problems, provided that both the objective and constraint functions are sufficiently smooth (i.e., continuously differentiable), as established in Kraft's original formulation [4]. The current Python implementation - PySLSQP- was designed to handle medium to large-scale problems and has been successfully applied in complex industrial contexts [3].

The implementation used in this work is based on the PySLSQP package developed by Joshy and Hwang [3]. In this version, subproblems are solved using quadratic approximations with linearized constraints, and the Hessian matrix is updated through quasi-Newton approximations, specifically the BFGS (Broyden–Fletcher–Goldfarb–Shanno) algorithm, as described in the implementation's documentation [3]. At each iteration, SLSQP performs the following main steps:

1. computing the *gradients* of the objective and the constraint functions;
2. building a local quadratic approximation of the objective function;
3. linearization of the nonlinear constraints;
4. solving a quadratic programming subproblem with linear constraints;
5. updating the solution based on the subproblem results.

Convergence of the method is guaranteed under continuity and differentiability of the functions involved. The method may also employ quasi-Newton approximations to avoid the explicit computation of the Hessian of the objective function. The selection of SLSQP as the optimizer was motivated by: **(a)** its ability to naturally handle nonlinear constraints, both equality and inequality; **(b)** its support for simple bound constraints on variables, which simplifies modeling; **(c)** its numerical stability and robust convergence behavior; **(d)** its availability in optimization libraries such as `SciPy` and other widely used scientific tools.

4 Results

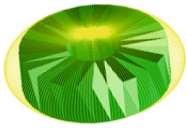
In this section, the numerical results are presented and discussed. The goal is to assess the performance of two models in fitting superquadrics to various 3D objects represented as point clouds. The objects used in the experiments are first described, followed by the experimental design, optimization method, and evaluation criteria. Results for Model 1 and Model 2 are reported separately,

including final parameter values and their influence on the objective function using Pearson correlation. A comparative analysis based on the estimated volumes of the fitted superquadrics identifies which model provides more accurate and efficient representations of the studied objects.

4.1 Experimental Setup

In the numerical experiments, three objects were used in robotic experiments, namely wheel, nut, and column, previously considered in [1]. For each object, a set of surface points is known. The characteristics of each object, specifically, the diameters along the x, y and z axes, corresponding to the maximum distance between points in the Ox, Oy and Oz planes, respectively, as well as the number of known points, are presented in Table 1.

Table 1. Characteristics of the objects used in the numerical experiments.

Parameter	nut	wheel	column
			
Diameter x	100	140	80
Diameter y	86.6	140	80
Diameter z	40	40	760
Number of Points	28	362	54

It is possible to observe that the number of available points varies significantly. Some objects have a more detailed geometric description, while others are represented more sparsely. In fact, while nut and wheel contains only 40 points, the Column contains 362. Also, the wheel and nut are flatter objects, as their dimensions along the x and y axes are much smaller than along z axis, whereas the Column is elongated, with a z dimension far greater than the remaining axes. Table 1 also shows that the objects differ greatly in scale. For instance, while the maximum dimension of the Column is 760 mm, the maximum dimension of the nut is 100 mm. Each object exhibits unique characteristics, which justify their use in the numerical experiments on superquadric fitting.

To standardize object dimensions and enable more meaningful comparisons of results across different objects, normalization was applied. Specifically, the 3D points were first separated into their respective X, Y and Z coordinates, and each coordinate was normalized independently to $[-1, 1]$, as:

$$x_i = \frac{X_i}{\max(|X|)}, \quad y_i = \frac{Y_i}{\max(|Y|)}, \quad z_i = \frac{Z_i}{\max(|Z|)},$$

where X_i, Y_i and Z_i denote the original coordinates, and x_i, y_i and z_i are the normalized ones. The dimensions of the cloud point along each axis, R_x, R_y and R_z, were computed as the difference between the maximum and minimum values of each coordinate. This procedure also ensured that all objects were centered at $(0, 0, 0)$ with orientation $(0, 0, 0)$.

Given the highly nonlinear nature of the optimization problems presented in Sect. 3, the existence of multiple optima is expected. Thus, considering that SLSQP is a local solver, preliminary tests were carried out showing that the optimal solution obtained for the superquadrics parameters depended on the bounds defined for the decision variables of the optimization problems (i.e. the parameters of the superquadrics $\boldsymbol{\lambda}$).

Varying the lower bounds of λ_3 and λ_4 did not lead to significant changes in the results. Therefore, in line with previous studies, the lower and upper limits of λ_3 and λ_4, were set fix to 0.1 and 1.4, respectively.

On the other hand, changes in the upper bounds of λ_0, λ_1, and λ_2 affected of the results, justifying their detailed analysis. Based on this, it was defined to fix the lower limits of λ_0, λ_1, λ_2 to $0.5R_x$, $0.5R_y$ e $0.5R_z$, respectively, while the upper limits would vary from $0.6R_x$ to $0.8R_x$, $0.6R_y$ to $0.8R_y$, and $0.6R_z$ to $0.8R_z$, respectively. Specifically, a total of 27 combinations for the upper bounds of λ_0, λ_1 and λ_2 were considered (see Table 2).

Table 2. Lower and upper limits for $\boldsymbol{\lambda}$.

Var.	λ^L	λ^U
λ_0	$0.5R_x$	$\{0.6R_x, 0.7R_x, 0.8R_x\}$
λ_1	$0.5\,R_y$	$\{0.6R_x, 0.7R_x, 0.8R_x\}$
λ_2	$0.5\,R_z$	$\{0.6R_x, 0.7R_x, 0.8R_x\}$
λ_3	0.1	1.4
λ_4	0.1	1.4

4.2 Results Using Model 1

For the all the objects and for the 27 combinations of the bounds of $\boldsymbol{\lambda}$ (see Table 2), an optimal solution was obtained for fitting the best superellipsoid. To improve clarity, in Table 3 and Fig. 1, only the results corresponding to the combinations of $\boldsymbol{\lambda}$ bounds that yield the smallest optimal value (referred to as the best case) and the largest optimal value (the worst case) are presented. The results show that, for all objects except the Column, the optimal objective values are close to zero ($<5 \times 10^{-6}$). This indicates that, for these objects, the clouds' point lie very close to the surface of the fitted superquadric. In contrast, for the Column, some points lie inside the superellipsoid, reflecting the greater geometric complexity of this object.

Table 3. Results of $\boldsymbol{\lambda}$ for the worst (w) and best (b) optimal objective function value F^* using Model 1 for superellipsoid fitting.

Object	Case	Optimal values					Objective	Upper limits		
		λ_0^*	λ_1^*	λ_2^*	λ_3^*	λ_4^*	F^*	$\lambda_0^{(U)}$	$\lambda_1^{(U)}$	$\lambda_2^{(U)}$
column	w	40.1872	43.640	383.677	0.1	1.1143	4.19659e-3	0.7 R_x	0.8 R_y	0.7 R_z
	b	40.1869	43.619	383.717	0.1	1.1139	4.19655e-3	0.7R_x	0.7R_y	0.6R_z
nut	w	**70.00**	**51.96**	20.048	0.114	**0.100**	4e-7	0.7 R_x	0.6 R_y	0.6 R_z
	b	**78.84**	**69.28**	20.000	0.100	**0.604**	4.72e-11	0.8 R_x	0.8 R_y	0.7R_z
wheel	w	**83.15**	**83.997**	20.03	0.100	0.740	2.189e-6	0.7 R_x	0.6 R_y	0.7 R_z
	b	**110.34**	**110.30**	20.00	0.100	0.761	3.270e-11	0.8R_x	0.8R_y	0.7R_z

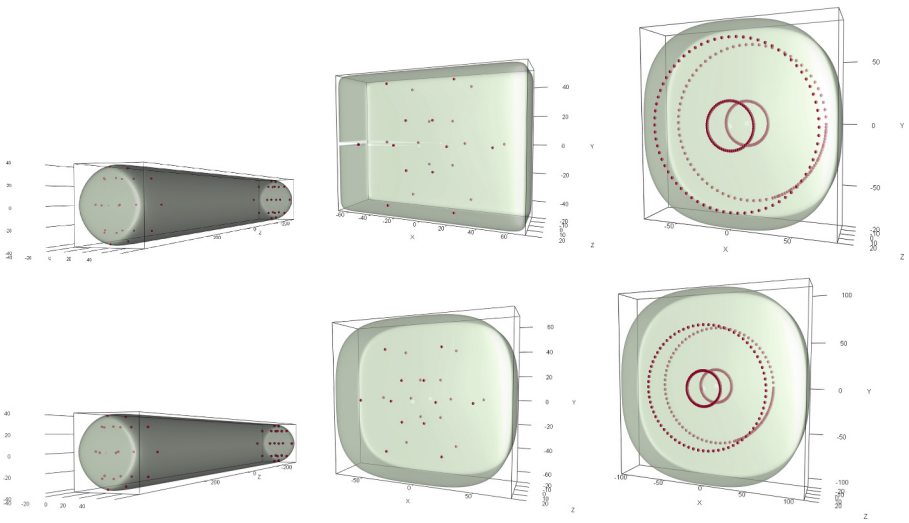


Fig. 1. Comparison between the worst (top) and best (bottom) objective-function results for the column (left), nut (middle) and wheel (right), using Model 1.

For the Column, the solutions obtained in the worst and best cases are very similar. The difference in the objective function between the two cases is only 4.8×10^{-8}. The parameters, λ_0, λ_1 and λ_2 coincide up to the second decimal place, while λ_3 is the same and λ_4 varies by 4×10^{-4}.

For the nut, the worst and best cases exhibit similar fitting behavior, with a difference in the objective function of 4×10^{-7}. The optimal values of λ_0, λ_1 and λ_2 differ by 8.84, 17.32 and 0.048, respectively, while the differences for λ_3 and λ_4 are 0.014 and 0.504.

For the wheel, both the worst and best cases result in fittings that closely match the real geometry of the object. The difference in the objective function between the two cases is 2.19×10^{-6}. In terms of parameters, λ_0 and λ_1 differ

by 27.19 and 26.30, respectively, while λ_2 varies by only 0.03. The values of λ_3 do not differ and the values of λ_4 differ by 0.021.

The results in Table 3 suggest that there is an association between the upper limits that is set for λ_0, λ_1 and λ_2 and the optimal solution. To evaluate this, Spearman correlations were computed. The Spearman correlation coefficient was selected for the association analysis because it is appropriate for data that do not follow a normal distribution. This coefficient does not require parametric assumptions and has the advantage of capturing both linear and nonlinear monotonic relationships between variables, thus providing more robust results. Table 4 presents the correlation coefficients and their corresponding p-values.

Table 4. Spearman correlation between the upper bounds of the optimization model parameters and the optimal solution value.

Object	Correlation Coefficient (p-value) of F^* *vs.*					
	$\lambda_0^{(U)}$		$\lambda_1^{(U)}$		$\lambda_2^{(U)}$	
column	$\simeq 0$	(0.99)	$\simeq 0$	(0.25)	$\simeq 0$	(0.99)
nut	-0.27	(0.17)	$\mathbf{-0.42}$	(0.03)	-0.14	(0.47)
wheel	$\mathbf{-0.53}$	(<0.01)	$\mathbf{-0.53}$	(<0.01)	0.01	(0.97)

For a significant level of 10%, no statistically significant correlation exists between the choice of the upper limit of λ_2 and the optimal solution found, as all the p-values are greater than 10%.

For the wheel, both λ_0 and λ_1 upper limits are significantly negatively associated ($r \simeq -0.53$) to the optimal solution. Thus, as the upper limit of the λ_0 and λ_1 parameters increases, the optimal solution decreases, which indicates a tighter fitting of the superellipsoid surface to the cloud points.

For the nut, the only significant correlation is between the upper limit of the λ_1 parameter of the superellipsoid, which presents a negative correlation coefficient of -0.42. This indicates that as the upper limit of λ_1 increases, the optimal objective value decreases, thus a tighter fitting of the superellipsoid surface to the cloud points is found.

These results highlight the sensitivity of the optimal solution to the upper bounds imposed on the superellipsoid parameters, but only for certain objects. This reinforces the importance of conducting an object-specific analysis to ensure an optimal fit in each case.

4.3 Model 2

Model 2 uses the volume of the superquadric as the objective function. The best and worst results of the 27 combinations of lower and upper bounds of the parameters of each object are presented Table 5 and Fig. 2.

For the Column, the difference of the solutions obtained in the worst and best cases is of 2×10^{-7}. The parameters λ_0, λ_1 and λ_2 present differences in the

Table 5. Results of $\boldsymbol{\lambda}$ for the worst (w) and best (b) optimal objective function value F^* using Model 2 for superellipsoid fitting.

Object	Case	Optimal values					Objective	Upper limits		
		λ_0^*	λ_1^*	λ_2^*	λ_3^*	λ_4^*	F^*	$\lambda_0^{(U)}$	$\lambda_1^{(U)}$	$\lambda_2^{(U)}$
column	w	40.1486153	42.224166	387.94293	0.1	1.071	6.5607275	0.8 R_x	0.6 R_y	0.8 R_z
	b	40.1486163	42.2241964	387.94278	0.1	1.071	6.5607273	0.8R_x	0.8R_y	0.7R_z
nut	w	51.0240076	51.9372869	21.129346	0.1	1.046	7.9209705	0.6 R_x	0.6 R_y	0.7 R_z
	b	51.0240077	51.936918	21.1293461	0.1	1.046	7.9209704	0.8 R_x	0.8 R_y	0.8R_z
wheel	w	71.4336643	71.4336637	21.1293462	0.1	1	6.8615689	0.7 R_x	0.6 R_y	0.7 R_z
	b	71.4336672	71.4336641	21.1293413	0.1	1	6.8615685	0.7 R_x	0.6 R_y	0.8 R_z

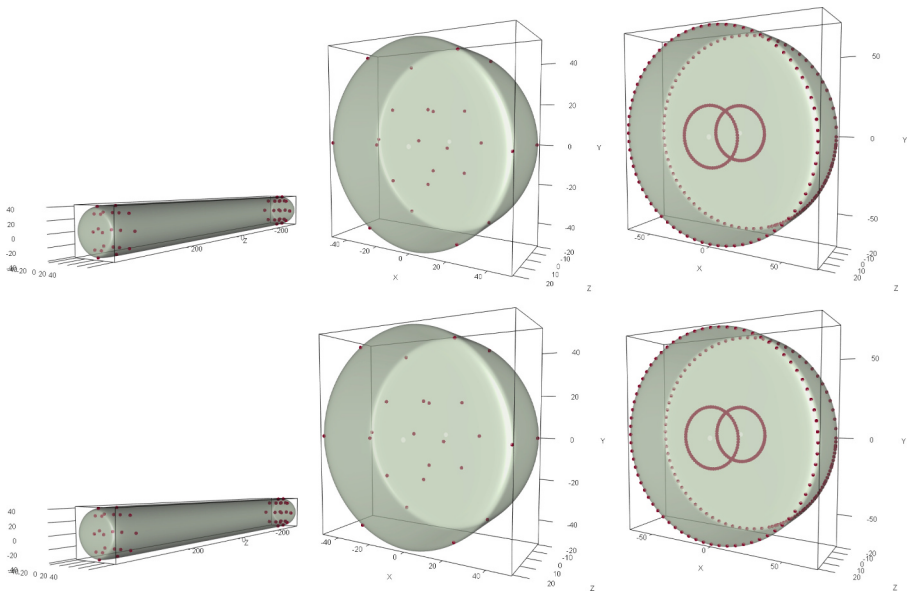


Fig. 2. Comparison between the worst (top) and best (bottom) objective function results for the column (left), nut (middle) and wheel (right), using Model 2.

order of 10^{-5}. Similar to the Column, the results for the worst and best cases for nut are almost the same. In fact, the difference in the objective function values is only 6.7×10^{-10}. In terms of the λ_0, λ_1 and λ_2 parameters, the differences occur only on the 7th decimal place. Finally, for the wheel, the difference is of 3.6×10^{-7}. The parameters λ_0, λ_1 and λ_2 present differences in the order of 10^{-6}. For all objects, λ_3 and λ_4 are the same for the best and worst cases.

These results suggest that there is no association between the upper limits that are set for λ_0, λ_1 and λ_2 and the optimal solution found. In fact, for a level of significance of 10%, the only statistically significant correlation was found

between the upper bound of λ_1 and the optimal value of the objective function, for the column. This was a moderate negative correlation of -0.32 ($p - value \simeq 0.099$). These results are not in line with the results obtained in Model 1, for which no significant correlations were found for this object.

4.4 Comparison of the Results for Models 1 and 2

Overall, the analysis of the correlations between the upper bounds of the parameters λ_0, λ_1, and λ_2 and the objective function value revealed differences in how the two models behave with respect to parameter sensitivity across the various objects. In Model 1, only a subset of the parameters significantly influenced the quality of the fit for certain objects. In Model 2, the impact of changing the upper bounds of the parameters was almost none.

To determine which of the two models provides a better representation of the objects, the estimated volume for each fitted superquadric was computed. Since all points of each object are contained within the corresponding superquadric, the resulting volume serves as an appropriate comparative indicator of fitting quality. In fact, the smaller the estimated volume for the same set of points, the better the superquadric fits the object when comparing the two models quantitatively.

Table 6 presents the volume of each superquadric obtained from the optimal fit for both models applied to the objects. Overall, it is observed that Model 2 produced significantly smaller volumes.

Table 6. Volume of the superquadric fitting each object for the worst (w) cases and best (b) for the two proposed models.

Object	Case	Model 1	Model 2	diff.
column	w	4010806.876	3988922.314	−2.19e+04
	b	4009976.797	3988922.219	−2.11e+04
	diff.	−8.30e+02	−9.45e−02	
nut	w	575621.197	342988.321	−2.33e+05
	b	781018.363	342988.317	−4.38e+05
	diff.	2.05e+05	−3.78e+05	
wheel	w	958297.984	672433.751	−2.86e+05
	b	1656186.185	672433.715	−9.84e+05
	diff.	6.98e+05	−3.57e−02	

For the column, the best solution obtained using model 1 presents a smaller volume that the one obtained for the worst case (it presents a difference of -8.30×10^2). For the other objects, the volume is larger for the best case. This can be explained by the fact that the nut and wheel have a hole. For model 2, as expected, the volume is always better for the best case.

Comparing the volumes using model 1 and 2, for all objects, as expected, the volume of the superquadric fitting the object is smaller using Model 2 than using Model 1(see last column in Table 6).

In summary, the results show that Model 2 performs better for all objects, as it produces more accurately fitted superquadrics with smaller volumes.

5 Conclusions and Future Work

This work provided a detailed examination of superellipsoids and their suitability for modeling three-dimensional objects from point cloud data. The study focuses on object modeling from point cloud data, using three different objects, previously used in human-robot collaboration tasks.

Two optimization models were developed—one based on implicit surface fitting and the other on volume minimization—and evaluated through graphical inspection, correlation analysis under varying upper-bound constraints, and a volume-based comparison of the resulting fits.

The results consistently showed that the volume-based formulation yields more compact and accurate representations, while also demonstrating reduced sensitivity to parameter upper limits when compared to the implicit surface fitting model. Overall, the findings indicate that the volume-minimization approach offers a more robust and reliable framework for superellipsoid fitting across the studied objects.

As future work, it would be valuable to extend this analysis to other families of superquadrics, enabling a broader assessment of their modeling capabilities and potential advantages in different geometric contexts.

Acknowledgments. E. Costa e Silva was supported by national funds through FCT - Fundação para a Ciência e Tecnologia through project UIDB/04728/2025 (https://doi.org/10.54499/UID/04728/2025).

References

1. Costa e Silva, E., et al.: Towards human-like bimanual movements in anthropomorphic robots: a nonlinear optimization approach (2015). https://doi.org/10.12785/amis/090210
2. Jaklic, A., Leonardis, A., Solina, F.: Segmentation and Recovery of Superquadrics, vol. 20. Springer, Dordrecht (2000). https://doi.org/10.1007/978-94-015-9456-1
3. Joshy, A.J., Hwang, J.T.: PySLSQP: a transparent python package for the SLSQP optimization algorithm modernized with utilities for visualization and post-processing. arXiv preprint arXiv:2408.13420 (2024)
4. Kraft, D.: A software package for sequential quadratic programming. Forschungsbericht- Deutsche Forschungs- und Versuchsanstalt fur. Luft- und Raumfahrt (1988)
5. Kumar, S., Hanson, R.: Robust methods for estimating pose and recovering shape from range data. Comput. Vis. Image Underst. **61**(3), 313–337 (1995). https://doi.org/10.1006/cviu.1995.1024

6. Makhal, A., Thomas, F., Gracia, A.P.: Grasping unknown objects in clutter by superquadric representation. In: Second IEEE International Conference on Robotic Computing (IRC), pp. 292–299. IEEE (2018)
7. Portal, R., Ferreira, A., Macho, A., Lima, P.U.: Using superquadrics for efficient collision detection in robotic manipulation. Robotica **28**(2), 253–265 (2010). https://doi.org/10.1017/S0263574709990519
8. Costa e Silva, E., Costa, M., Erlhagen, W., Bicho, E.: Superquadrics objects representation for robot manipulation. AIP Conf. Proc. **1738**, 300004 (2016)
9. Šircelj, J., et al.: Segmentation and recovery of superquadric models using convolutional neural networks. arXiv preprint arXiv:2001.10504 (2020)
10. Solina, F., Bajcsy, R.: Recovery of parametric models from range images: the case for superquadrics with global deformations. IEEE Trans. Pattern Anal. Mach. Intell. **12**(2), 131–147 (1990). https://doi.org/10.1109/34.45763
11. Torres, J.B., Carmona, E., Makhal, A., Heidari, O., Gracia, A.P.: Supertoroid fitting of objects with holes for robotic grasping and scene generation. arXiv preprint arXiv:2412.04174 (2024)