Supplementary Material of Robust and Accurate Superquadric Recovery: a Probabilistic Approach

Weixiao Liu[1,2] Yuwei Wu[1] Sipu Ruan[1] Gregory S. Chirikjian[1*]
[1]National University of Singapore [2]Johns Hopkins University
{mpewxl, yw.wu, ruansp, mpegre}@nus.edu.sg

Abstract

In this supplementary material, we provide the detailed derivations, discussions and experiment settings. In Sec. 1, the approximation of the superquadric surface area is detailed. In Sec. 2, we further discuss the stability problem about the optimization. In Sec. 3, we present the mathematical formulation of the S-step, which includes the candidate generation and the switching strategy. In Sec. 4, we introduce how to conduct equal-distance sampling on a superquadric surface. Finally, in Sec. 5, detailed experiment settings are presented.

1. Interpolation of Superquadric Area

Generally, the surface area A_θ of a superquadric cannot be expressed in closed-form with $\{\epsilon_1, \epsilon_2, a_x, a_y, a_z\}$ (note that the surface area is independent of the pose $\{\mathbf{R}, \mathbf{t}\}$). However, closed-form solution is available when the shape parameter $\{\epsilon_1, \epsilon_2\}$ is a combination of 0 and 2.

(1) When $\{\epsilon_1, \epsilon_2\} = \{0, 0\}$, the superquadric is a cuboid, whose surface area is

$$A_{\{0,0\}} = 8(a_x a_y + a_y a_z + a_x a_z) \tag{1}$$

(2) When $\{\epsilon_1, \epsilon_2\} = \{0, 2\}$, the superquadric is a hexahedron, whose surface area is

$$A_{\{0,2\}} = 8a_z(a_x^2 + a_y^2)^{\frac{1}{2}} + 4a_x a_y \tag{2}$$

(3) When $\{\epsilon_1, \epsilon_2\} = \{0, 2\}$, the superquadric is a octahedron, whose surface area is

$$A_{\{2,0\}} = 4(a_x(a_y^2 + a_z^2)^{\frac{1}{2}} + a_y(a_x^2 + a_z^2)^{\frac{1}{2}}) \tag{3}$$

(4) Lastly, when $\{\epsilon_1, \epsilon_2\} = \{2, 2\}$, the superquadric is a octahedron, whose surface area is

$$A_{\{2,2\}} = 8\left(a_0(a_0 - a_{xy})(a_0 - a_{yz})(a_0 - a_{xz})\right)^{\frac{1}{2}} \tag{4}$$

where

$$\begin{aligned} a_{xy} &= (a_x^2 + a_y^2)^{\frac{1}{2}} \\ a_{yz} &= (a_y^2 + a_z^2)^{\frac{1}{2}} \\ a_{xz} &= (a_x^2 + a_z^2)^{\frac{1}{2}} \\ a_0 &= (a_{xy} + a_{yz} + a_{xz})/2 \end{aligned}$$

We approximate the area with a bi-linear interpolation

$$A_\theta = \begin{bmatrix} 1 - \epsilon_1/2 \\ \epsilon_1/2 \end{bmatrix}^T \begin{bmatrix} A_{\{0,0\}} & A_{\{0,2\}} \\ A_{\{2,0\}} & A_{\{2,2\}} \end{bmatrix} \begin{bmatrix} 1 - \epsilon_2/2 \\ \epsilon_2/2 \end{bmatrix} \tag{5}$$

We test the interpolation method exhaustively throughout the convex region of superquadrics, and it shows an average relative error of less than 10% compared with the the area calculated through the triangular mesh. More importantly, the bi-linear interpolation can fully capture the property that the surface area of a superquadric grows monotonically with ϵ_1 and ϵ_2.

2. Discussion on Optimization

The Levenberg–Marquardt (LM) algorithm is widely used in superquadrics fitting [2, 3, 6, 7]. However, it has been confirmed that the optimization suffers from numerical instability as either one of the shape parameters ϵ_1 or ϵ_2 approaches 0. As a consequence, most of the methods compromise by constraining the lower bounds of the shape parameters to 0.1, resulting in less accuracy when representing shapes with sharp edges, *e.g.*, cuboids and cylinders. In [7], the authors claim that the problem is caused by the inherent instability of the implicit function and its gradient. They solve this problem by approximating the implicit function with linear functions in the unstable region. We find out that the instability can also be explained as being introduced by the way that the LM deals with the bound. When the optimization steps outside of the bound, it will be simply projected back to the nearest point on the bound. Therefore, if we set the lower bounds of ϵ_1 and ϵ_2 to 0, the LM algorithm is likely to be forced to visit the points on the hyperplane of the lower bounds, where the implicit function is not well defined. In contrast, the trust-region-reflective [1] tackles the bounding condition differently by conducting a line search along the reflective path, and thus avoid directly checking the value of the implicit function on the bound. Therefore, by utilizing the trust-region-reflective, our method can maintain numerical stability without approximating the implicit function, as shown in Fig. 1. It can be observed that both of the methods are stable; however, [7] shows less accuracy in recovering the superquadric parameter within the unstable region, which is caused by their approximation of the implicit function.

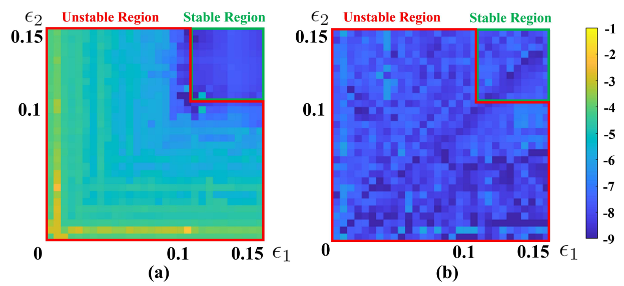

Figure 1. Performance in the unstable region. The error is evaluated by $\log_{10}(\|[\epsilon_1, \epsilon_2, a_x, a_y, a_z] - [\epsilon_1, \epsilon_2, a_x, a_y, a_z]_{gt}\|_2)$, where $[\epsilon_1, \epsilon_2, a_x, a_y, a_z]$ is the estimated shape and scale parameters and $[\epsilon_1, \epsilon_2, a_x, a_y, a_z]_{gt}$ is the ground-truth. (a) The recovery error of NS. (b) The recovery error of the proposed method.

3. Similarity and Switching

In this section, we show the detailed mathematics formulation of the S-step (geometry-guided local optimum avoidance) in Sec. 3.5 of the paper. First, we formulate how to search for candidate parameters $\{\boldsymbol{\theta}_i^c\}$ encoding similar superquadrics from the current estimation $\boldsymbol{\theta}$, which is, presumably, a local optimum. Recall that the superquadric parameter $\boldsymbol{\theta} = \{\epsilon_1, \epsilon_2, a_x, a_y, a_z, \mathbf{R}, \mathbf{t}\}$ (Sec. 2.1 in the papaer). We denote $\mathbf{R} = [\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3]$, where $\mathbf{r}_1$, $\mathbf{r}_2$ and $\mathbf{r}_3$ are orthonormal column vectors, corresponding to the directions of the x, y and z-axis (principal axis) of the superquadric frame, respectively.

Axis-mismatch similarity: In this category, we can obtain 2 candidates by re-assigning the principal axis to the x-axis and the y-axis, and re-arranging the corresponding shape, scale and rotation parameters:

$$\begin{aligned} \boldsymbol{\theta}_1^c &= \{\epsilon_2, \epsilon_1, a_y, a_z, a_x, [\mathbf{r}_2, \mathbf{r}_3, \mathbf{r}_1], \mathbf{t}\} \\ \boldsymbol{\theta}_2^c &= \{\epsilon_2, \epsilon_1, a_z, a_x, a_y, [\mathbf{r}_3, \mathbf{r}_1, \mathbf{r}_2], \mathbf{t}\} \end{aligned} \tag{6}$$

It can be easily verified that both $[\mathbf{r}_2, \mathbf{r}_3, \mathbf{r}_1]$ and $[\mathbf{r}_3, \mathbf{r}_1, \mathbf{r}_2]$ are proper rotation matrices (orthogonal with determinant equals to 1). When $\epsilon_1 = \epsilon_2$, the shapes encoded by the candidate parameters are identical to the current estimation. This can be proved by substituting the candidate parameters into the implicit function of superquadrics. In more general cases, $\boldsymbol{\theta}_1^c$ and $\boldsymbol{\theta}_2^c$ provides two superquadrics similar to $\boldsymbol{\theta}$, and thus hold similar likelihoods when evaluated by the probabilistic model. Examples are visualized in Fig. 2.

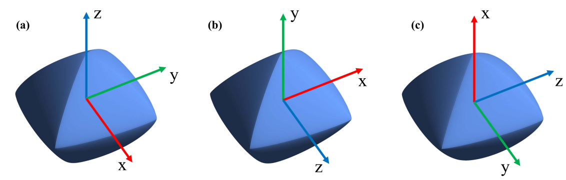

Figure 2. Similar superquadrics encoded by candidate parameters generated via axis-mismatch similarity. (a) The underlying shape of the current estimation $\boldsymbol{\theta} = [1.5, 1.7, 1, 1.2, 0.8, \mathbf{I}, [0, 0, 0]^T]$. (b) The underlying shape of the candidate $\boldsymbol{\theta}_1^c$. (c) The underlying shape of the candidate $\boldsymbol{\theta}_2^c$.

Duality Similarity: The duality similarity is more complex compared with the axis-mismatch similarity. To well illustrate the idea, we first review a property of superquadrics. When viewed from the superquadric frame (that is, not considering the spatial pose of the superquadric), a superquadric can be interpreted as the spherical product of two superellipses. This property can be shown by expressing the superquadric surface with its parametric equation

$$\mathbf{p}(\eta, \omega) = \begin{bmatrix} a_x \cos^{\epsilon_1} \eta \cos^{\epsilon_2} \omega \\ a_y \cos^{\epsilon_1} \eta \sin^{\epsilon_2} \omega \\ a_z \sin^{\epsilon_1} \eta \end{bmatrix} \tag{7}$$

where $\mathbf{p}(\eta, \omega)$ is a bijective function which maps a point on a unit sphere $[\cos \eta, \sin \omega] \in \mathbb{S}^2$ to a point on the superquadric surface (parameterized by $\{\epsilon_1, \epsilon_2, a_x, a_y, a_z\}$). It can be observed that

$$\mathbf{p}(\eta, \omega) = \begin{bmatrix} \cos^{\epsilon_1} \eta \\ a_z \sin^{\epsilon_1} \eta \end{bmatrix} \otimes \begin{bmatrix} a_x \cos^{\epsilon_2} \omega \\ a_y \sin^{\epsilon_2} \omega \end{bmatrix} \tag{8}$$

where $\otimes$ denotes the spherical product. The right-hand side of the equation indicates a spherical product of two superellipses (paramaterized by $\{\epsilon_1, 1, a_z\}$ and $\{\epsilon_2, a_x, a_y\}$, respectively). We call the former one the superellipse along the principal axis (z-axis), and the latter one the superellipse orthogonal to the principal axis.

Now that we have decoupled a superquadric into a pair of superellipses, it is natural to ask: can we find a similar superquadric by exploiting similar superellipses? Actually, this goal can be achieved by utilizing the parametric ambiguity of the superellipse orthogonal to the principal axis ($\{\epsilon_2, a_x, a_y\}$). When $a_x = a_y = \bar{a}$, we can always obtain a similar or even identical superellipse by (1) generating a 'dual' superellipse $\{2 - \epsilon_2, \bar{a}, \bar{a}\}$; (2) rotating $\pi/4$ about the centroid; and (3) scaling $\bar{a}$ with a scaler s. Then, we are able to obtain a similar superquadric by conducting the spherical product

$$\begin{bmatrix} \cos^{\epsilon_1} \eta \\ a_z \sin^{\epsilon_1} \eta \end{bmatrix} \otimes \left(\begin{bmatrix} \cos \frac{\pi}{4} & -\sin \frac{\pi}{4} \\ \sin \frac{\pi}{4} & \cos \frac{\pi}{4} \end{bmatrix} \begin{bmatrix} s \cdot \bar{a} \cos^{2-\epsilon_2} \omega \\ s \cdot \bar{a} \sin^{2-\epsilon_2} \omega \end{bmatrix} \right)$$

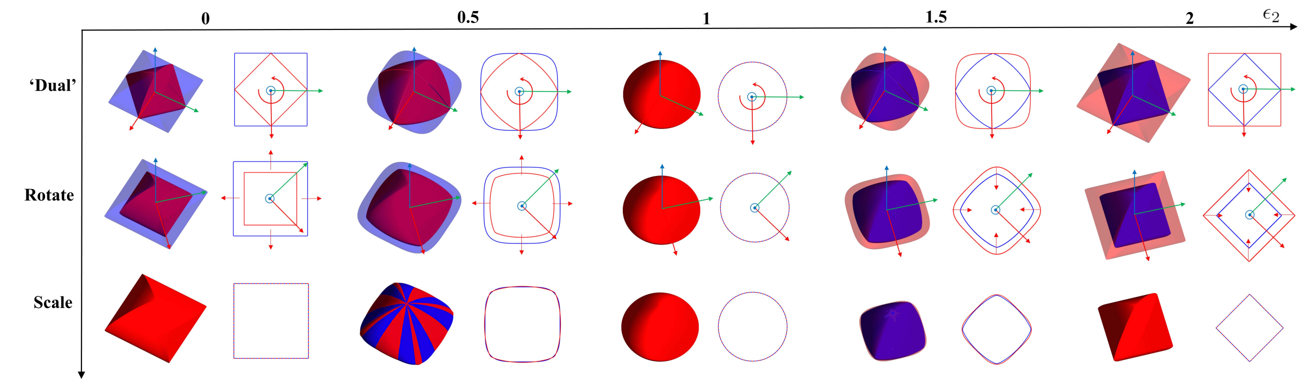

Figure 3. Generate candidate superquadrics with duality similarity. The blue superquadrics are the current estimation. The red superquadrics demonstrate how similar superquadrics are generated by a sequence of transformations.

Examples are shown in Fig. 3. When $\epsilon_2 = 0$, $\epsilon_2 = 1$ or $\epsilon_2 = 2$, the generated superquadric is identical to the original one (Eq. (8)). In more general cases, the generated superquadric provides a candidate similar to the original one. By taking the pose of the superquadric into consideration and relaxing $a_x = a_y$ with $a_x \approx a_y$ (in this paper, we assume $a_x \approx a_y$ when $0.8 < |a_x/a_y| < 1.2$), we can obtain a general formulation of the duality similarity as follows

$$\boldsymbol{\theta}_3^c = \{\epsilon_1, 2-\epsilon_2, s\cdot\bar{a}, s\cdot\bar{a}, a_z, \mathbf{R}\cdot\mathbf{R}_z(\pi/4), \mathbf{t}\} \tag{9}$$

where

$$s = \begin{cases} \left((1-\sqrt{2})\,\epsilon_2 + \sqrt{2}\right) & if\ \epsilon_2 \le 1 \\ \left(\sqrt{2}/2 - 1\right)\epsilon_2 + 2 - \sqrt{2}/2 & otherwise \end{cases}$$
$$\bar{a} = (a_x + a_y)/2$$

s is the scale which compensates the expansion (when $\epsilon_2 > 1$) or shrinkage (when $\epsilon_2 < 1$) induced by the duality transformation ($2-\epsilon_2 \to \epsilon_2$). $\mathbf{R}_z(\pi/4)$ denotes a rotation of $\pi/4$ about the z-axis.

Combinations of Similarities: Similar superquadrics can also be obtained by combining the axis-mismatch similarity with the duality similarity. That is:

(1) when $a_y \approx a_z$, we re-assign the principal axis to the y-axis and then look for its duality similarity

$$\boldsymbol{\theta}_4^c = \{\epsilon_2, 2-\epsilon_1, s\cdot\bar{a}, s\cdot\bar{a}, a_x, [\mathbf{r}_2, \mathbf{r}_3, \mathbf{r}_1]\cdot\mathbf{R}_z(\pi/4), \mathbf{t}\} \tag{10}$$

where

$$s = \begin{cases} \left((1-\sqrt{2})\,\epsilon_1 + \sqrt{2}\right) & if\ \epsilon_1 \le 1 \\ \left(\sqrt{2}/2 - 1\right)\epsilon_1 + 2 - \sqrt{2}/2 & otherwise \end{cases}$$
$$\bar{a} = (a_y + a_z)/2$$

(2) similarly, when $a_x \approx a_z$, we re-assign the principal axis to the y-axis and then look for its duality similarity

$$\boldsymbol{\theta}_5^c = \{\epsilon_2, 2-\epsilon_1, s\cdot\bar{a}, s\cdot\bar{a}, a_y, [\mathbf{r}_3, \mathbf{r}_1, \mathbf{r}_2]\cdot\mathbf{R}_z(\pi/4), \mathbf{t}\} \tag{11}$$

where

$$s = \begin{cases} \left((1-\sqrt{2})\,\epsilon_1 + \sqrt{2}\right) & if\ \epsilon_1 \le 1 \\ \left(\sqrt{2}/2 - 1\right)\epsilon_1 + 2 - \sqrt{2}/2 & otherwise \end{cases}$$
$$\bar{a} = (a_x + a_z)/2$$

Switching Strategy: Utilizing the similarities, 'highways' are built within the parameter space, connecting distant parameters encoding superquadrics with similar geometric shapes. The S-step is triggered when the relative decrease of the negative log-likelihood is less than a threshold δ. Utilizing the similarities, a set of candidate parameters $\{\boldsymbol{\theta}_i^c\}$ are generated based on the current estimation $\boldsymbol{\theta}$. Then, we check if the negative log-likelihood can be further decreased from a candidate. If a valid candidate is found, we switch to it and continue the optimization; Otherwise, we declare the current estimation $\boldsymbol{\theta}$ as optimal. The detailed process of the S-step is summarized in Algorithm 1.

Algorithm 1 S-step: geometric local optimum avoidance

Input: $\boldsymbol{\theta}, \sigma^2, \mathbf{X}$ ▷ current estimation and point cloud
Output: $\boldsymbol{\theta}_s, \sigma_s^2, success$ ▷ switched parameter
$\boldsymbol{\theta}_s \leftarrow \boldsymbol{\theta}$; $\sigma_s^2 \leftarrow \sigma^2$
$success \leftarrow 0$
$l \leftarrow Likelihood(\boldsymbol{\theta}, \sigma^2, \mathbf{X})$ ▷ negative log-likelihood
$\{\boldsymbol{\theta}_i^c\} \leftarrow Similarities(\boldsymbol{\theta})$ ▷ generate candidates
for $i = 1, ..., |\{\boldsymbol{\theta}_i^c\}|$ **do**
  $\hat{\boldsymbol{\theta}}_i^c, \hat{\sigma}_i^2 \leftarrow EM(\boldsymbol{\theta}_i^c, \sigma^2, \mathbf{X})$ ▷ update candidate
  $l_i \leftarrow Likelihood(\hat{\boldsymbol{\theta}}_i^c, \hat{\sigma}_i^2, \mathbf{X})$
  if $l_i < l$ **then**
    $\boldsymbol{\theta}_s \leftarrow \hat{\boldsymbol{\theta}}_i^c$; $\sigma_s^2 \leftarrow \hat{\sigma}_i^2$
    $success \leftarrow 1$
    Break
  end if
end for

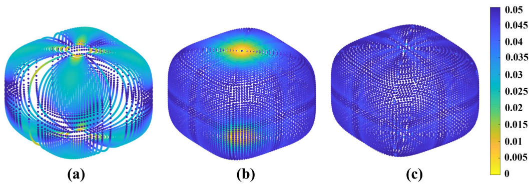

Figure 4. Examples of different sampling strategies on a superquadric with shape parameters $\epsilon_1 = 0.5$ and $\epsilon_2 = 0.5$. The target sampling interval is $\Delta = 0.05$. We use a heatmap to demonstrate the uniformity of the sampling results. The value of each point denotes the average Euclidean distances between the point and its four closest neighbors. (a-c) Results of the vanilla mapping, the method proposed by [7] and ours, respectively.

4. Equal-distance Sampling on Superquadric Surface

Eq. (7) defines a bijective mapping between the unit sphere and a superquadric surface. The most straightforward way of sampling points on a superquadric surface is to sample η and ω within their range and then map to the target surface utilizing Eq. (7). However, due to the non-linearity of the mapping, points evenly distributed in the spherical coordinates result in uneven samples on the superquadric surface. In [7], the authors propose a method to obtain a more homogeneous coverage of the surface. As shown in Eq. (8), a superquadric is the spherical product of two superellipses. Taking advantage of this property, two sets of 2D points are sampled independently on the periphery of each superellipse with a fixed interval [5]. Then, samples on the superquadric surface are constructed by taking the spherical product of the points between the sets. As shown in Fig. 4, the samples distribute more evenly compared with the vanilla mapping approach. However, the density gets significantly higher when the samples approach the poles along the z-axis. As $\eta \to \frac{\pi}{2}/\frac{-\pi}{2}$, $|\cos^{\epsilon_1}\eta| \to 0$ and the perimeter of the sliced superellipse at higher latitude shrinks accordingly. However, the superellipses at different latitudes contain the same number of points, and thus resulting in unevenly distributed samples after the spherical product.

To solve this problem, we improve the sampling strategy as follows. Suppose the superquadric is parameterized by $\{\epsilon_1, \epsilon_2, a_x, a_y, a_z\}$, which can be decoupled into two superellipses $\{\epsilon_1, 1, a_z\}$ and $\{\epsilon_2, a_x, a_y\}$. First, we sample the superellipse $\{\epsilon_1, 1, a_z\}$ with a fixed interval Δ using the algorithm in [5], and record the corresponding latitudes $\eta_i \in [-\pi/2, \pi/2], (i = 1, 2, ...)$. Then, instead of sampling directly on the superellipse $\{\epsilon_2, a_x, a_y\}$, we sample points evenly on re-scaled superellipses parameterized by $\{\epsilon_2, a_x \cos^{\epsilon_1}\eta_i, a_y \sin^{\epsilon_1}\eta_i\}$. In other words, we adaptively adjust the scale of the superellipse $\{\epsilon_2, a_x, a_y\}$ according to the current sampling latitude η_i. And finally, by conducting the spherical product at each latitude, we obtain an overall equal-distance sampling of the superquadric surface.

5. Implementation Details

All the baseline methods are implemented with the official MATLAB Optimization Toolbox [4]. The step-tolerance (TolX) and optimality-tolerance (TolFun) are all set to 10^{-6}. Due to the numerical instability of Implicit-LSQ [6], Radial-LSQ [2] and Robust-fitting [3], the lower bounds of the shape parameters ϵ_1 and ϵ_2 are set to 0.1, as recommended by the authors. Following the settings in [7], the lower bounds of ϵ_1 and ϵ_2 are set to 0 for the NS method.

For the proposed method, the lower bounds of ϵ_1 and ϵ_2 are set to 0. In Sec. 4.1 of the paper, the prior outlier probability $\omega_o = 0$ for the partial data experiments, $\omega_o = 0.2$ for the outlier experiments and $\omega_o = 0.01$ for the noise experiments. ω_o is set relatively low for the partial data and noise experiments, since no artificial outliers are added. In Sec. 4.2, $\omega_o = 0.01$ for the KIT dataset and $\omega_o = 0.05$ for the BigBIRD dataset. We set a higher outlier probability for the latter, since the point clouds are captured by a RGB-D camera, resulting in more outliers and noise. ω_o is set to 0.8 in the Sec. 4.3, since we need to identify a large number of points as outliers in order to capture the major superquadric-like shape from a complex object. Also in Sec. 4.3 of the paper, the maximum layer of the multi-superquadrics recovery is 3, and the cluster-pruning threshold is set to 60 points. In all the experiments, the switching threshold $\delta = 0.1$.

References

[1] T. F. Coleman and Y. Li. An interior trust region approach for nonlinear minimization subject to bounds. *SIAM Journal on Optimization*, 6(2):418–445, 1996. 1

[2] A. D. Gross and T. E. Boult. Error of fit measures for recovering parametric solids. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 690–694, 1988. 1, 4

[3] Y. Hu and W. G. Wee. Robust 3D part extraction from range images with deformable superquadric models. In *Signal Processing, Sensor Fusion, and Target Recognition IV*, volume 2484, pages 524 – 535. International Society for Optics and Photonics, SPIE, 1995. 1, 4

[4] Matlab optimization toolbox, 2021a. The MathWorks, Natick, MA, USA. 4

[5] M. Pilu and R. B. Fisher. Equal-distance sampling of superellipse models. In *Proceedings of the 1995 British Conference on Machine Vision (Vol. 1)*, BMVC '95, page 257–266, GBR, 1995. BMVA Press. 4

[6] F. Solina and R. Bajcsy. 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. 1, 4

[7] N. Vaskevicius and A. Birk. Revisiting superquadric fitting: A numerically stable formulation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(1):220–233, 2019. 1, 2, 4