Chapter 3

Superquadrics for Object Representation

3.1 Introduction

There is a multitude of geometric parametric entities commonly employed in the computer vision literature for three dimensional object modeling. Among them generalized cylinders [19], implicit (fourth degree) polynomials [89], geons [18], spherical harmonic and Fourier surfaces [150], [149], symmetry seeking models [155], blob models [111], hyperqadrics [69], and others.

We decided to use *Superquadrics* [11], [148], [99], [77] for modeling our target objects for the following reasons: Firstly, by constraining their shape parameters these models can be made to resemble our symmetric box-like objects in a remarkable way. Secondly, all their parameters have an intuitive meaning, which makes their handling straightforward. Thirdly, the superquadric parameters have a large expressive power for natural shapes with rounded edges and corners, as well as for standard geometric solids with sharp edges. This implies that our system is given the potential to be used for recovery of a wide variety of objects. Besides, the expressiveness of superquadrics can be easily further enhanced by the addition of global and local shape deformations. Fourthly, closed form expressions exist for the models as well as for special curves on their surface, which makes both rendering and usage fast and direct. Finally, superquadric recovery via fitting to three dimensional data is a problem which has been thoroughly investigated. Robust and fast methods have been developed for this purpose.

Target of this chapter is to give the reader all the necessary information about superquadrics and their properties needed for comprehending the contens of the subsequent chapters. More specifically, section 3.2, introduces the models. Section 3.3 focuses on the way in which uniform model rendering can be performed. Section 3.4, presents the box-like superquadric models. Section 3.5 presents the global deformations we introduced to the models so that they are made to more closely resemble our target objects. Section 3.6 discusses the additional parameters that have to be introduced in the superquadric parameter set so that model representation in a general position of the sensor coordinate system is enabled. Section 3.7 focuses on the issue of fitting superquadric models to range data. Section 3.8 introduces the way in which we can determine whether a superquadric model corresponds to a graspable

object in an piled object configuration. Finally, section 3.9 concludes this chapter.

3.2 Definition

The term Superquadrics, was firstly used in [11] to define a family of shapes that includes *superellipsoids*, *superhyperboloids* of one sheet and *superhyperboloids* of two sheets, as well as *supertoroids*. In the computer vision literature however, the term superquadrics is used to refer to superellipsoid objects. In this thesis, we will also use the term superquadrics as a synonym for superellipsoids.

Superquadrics are defined by means of their analogous two-dimensional entities, the *superellipses*. A superellipse, is a two dimensional curve, the explicit form of which is defined as follows:

$$\mathbf{s}_{a,b,\epsilon}(\alpha) = \begin{bmatrix} x(a,b,\epsilon;\alpha) \\ y(a,b,\epsilon;\alpha) \end{bmatrix} = \begin{bmatrix} a\cos(\alpha)^{\epsilon} \\ b\sin(\alpha)^{\epsilon} \end{bmatrix} \quad -\pi \leq \alpha \leq \pi,\ 0 < \epsilon \tag{3.1}$$

In the preceding expression (a, b, ϵ) are the parameters of the superellipse. More specifically, a, b, express the size along its $\mathbf{X}$ and $\mathbf{Y}$ axes respectively, and ϵ determines its shape. Note, in the preceding expression as well as in the expressions that follow, exponentiation with ϵ is a signed power function such that $\cos(\alpha)^{\epsilon} = \text{sign}(\cos(a))|\cos(\alpha)|^{\epsilon}$. The explicit form of the superellipse is employed to generate two dimensional points on it, by assigning to α values obtained by uniformly sampling the range $[-\pi, \pi]$. These points are usually used for both model rendering and fitting purposes. Fig. 3.1 illustrates points on four superellipses for values of ϵ in $[0.1, 2]$ using (3.1).

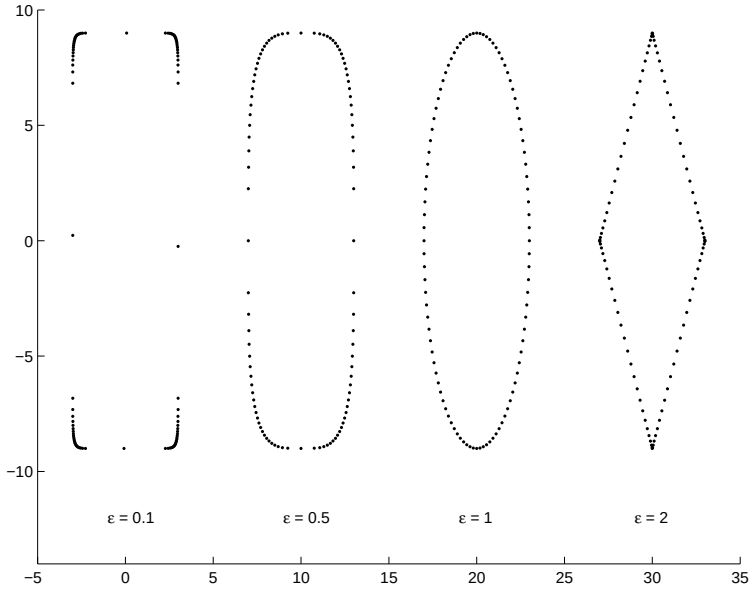


Figure 3.1: Superellipses of various shapes

The implicit superellipse equation, can be derived from the preceding explicit form using simple algebraic manipulations and amounts to:

$$\left(\frac{x}{a}\right)^{\frac{2}{\epsilon}} + \left(\frac{y}{b}\right)^{\frac{2}{\epsilon}} = 1 \tag{3.2}$$

A superquadric can be obtained as a *spherical product* of a pair of superellipses (see [26], [77] p.19) for details). More specifically, the superquadric model $\mathbf{S}_{a_1,a_2,a_3,\epsilon_1,\epsilon_2}(\eta,\omega)$ is defined as the spherical product of the superellipses $\mathbf{s_1}_{1,a_2,\epsilon_1}(\eta)$, and $\mathbf{s_2}_{a_1,a_2,\epsilon_2}(\omega)$, and its explicit form amounts to:

$$\mathbf{S_p}(\eta,\omega) = \begin{bmatrix} x(\mathbf{p};\eta,\omega) \\ y(\mathbf{p};\eta,\omega) \\ z(\mathbf{p};\eta,\omega) \end{bmatrix} = \begin{bmatrix} a_1\cos(\eta)^{\epsilon_1}\cos(\omega)^{\epsilon_2} \\ a_2\cos(\eta)^{\epsilon_1}\sin(\omega)^{\epsilon_2} \\ a_3\sin(\eta)^{\epsilon_1} \end{bmatrix} \tag{3.3}$$

In the preceding equation, η and ω lie in the range $[-\pi/2, \pi/2]$ and $[-\pi, \pi]$ respectively. The parameter vector $\mathbf{p} = (a_1, a_2, a_3, \epsilon_1, \epsilon_2)$, incorporates the parameters of the superquadric. More specifically, ϵ_1, ϵ_2 control the shape of the model, and a_1, a_2, a_3 express the size of the model along its $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$ axes respectively. Superquadric model rendering is performed using the explicit form (3.3). Fig. 3.2 shows various superquadric models rendered in this way. Each of these models corresponds to a different value of the shape parameters, within the range $[0.1, 2]$.
The implicit superquadric equation can be derived from the explicit form of (3.3), by simple manipulations of the latter:

$$\left(\left(\frac{x}{a_1}\right)^{\frac{2}{\epsilon_2}} + \left(\frac{y}{a_2}\right)^{\frac{2}{\epsilon_2}}\right)^{\frac{\epsilon_2}{\epsilon_1}} + \left(\frac{z}{a_3}\right)^{\frac{2}{\epsilon_1}} = 1 \tag{3.4}$$

All points with coordinates (x, y, z) for which the preceding equation is satisfied, lie by definition on the surface of the model. The left part of the equation equals to the so called *inside- outside* function F, shown in (3.5). If $F(\mathbf{p}, x, y, z) = 1$, the point (x, y, z) lies on the model with parameter vector $\mathbf{p}$. If $F(\mathbf{p}, x, y, z) < 1$ the point lies inside the superquadric, and if $F(\mathbf{p}, x, y, z) > 1$ it lies outside the model. In other words, the function F can be used as a tool for determining the relative location of a point with respect to the model.

$$F(\mathbf{p}; x, y, z) = \left(\left(\frac{x}{a_1}\right)^{\frac{2}{\epsilon_2}} + \left(\frac{y}{a_2}\right)^{\frac{2}{\epsilon_2}}\right)^{\frac{\epsilon_2}{\epsilon_1}} + \left(\frac{z}{a_3}\right)^{\frac{2}{\epsilon_1}} \tag{3.5}$$

3.3 Superquadric sampling

In figure 3.2, we can observe that the points on the superquadric surface generated using the implicit superquadric form (3.4), are not uniformly sampled. More specifically, (3.4), generates a considerably bigger number of points on areas of large curvature. This problem is actually inherited from the corresponding problem of the superellipse, from which as we saw superquadrics are generated. Actually, the problem can be observed in Fig. 3.1 as well.

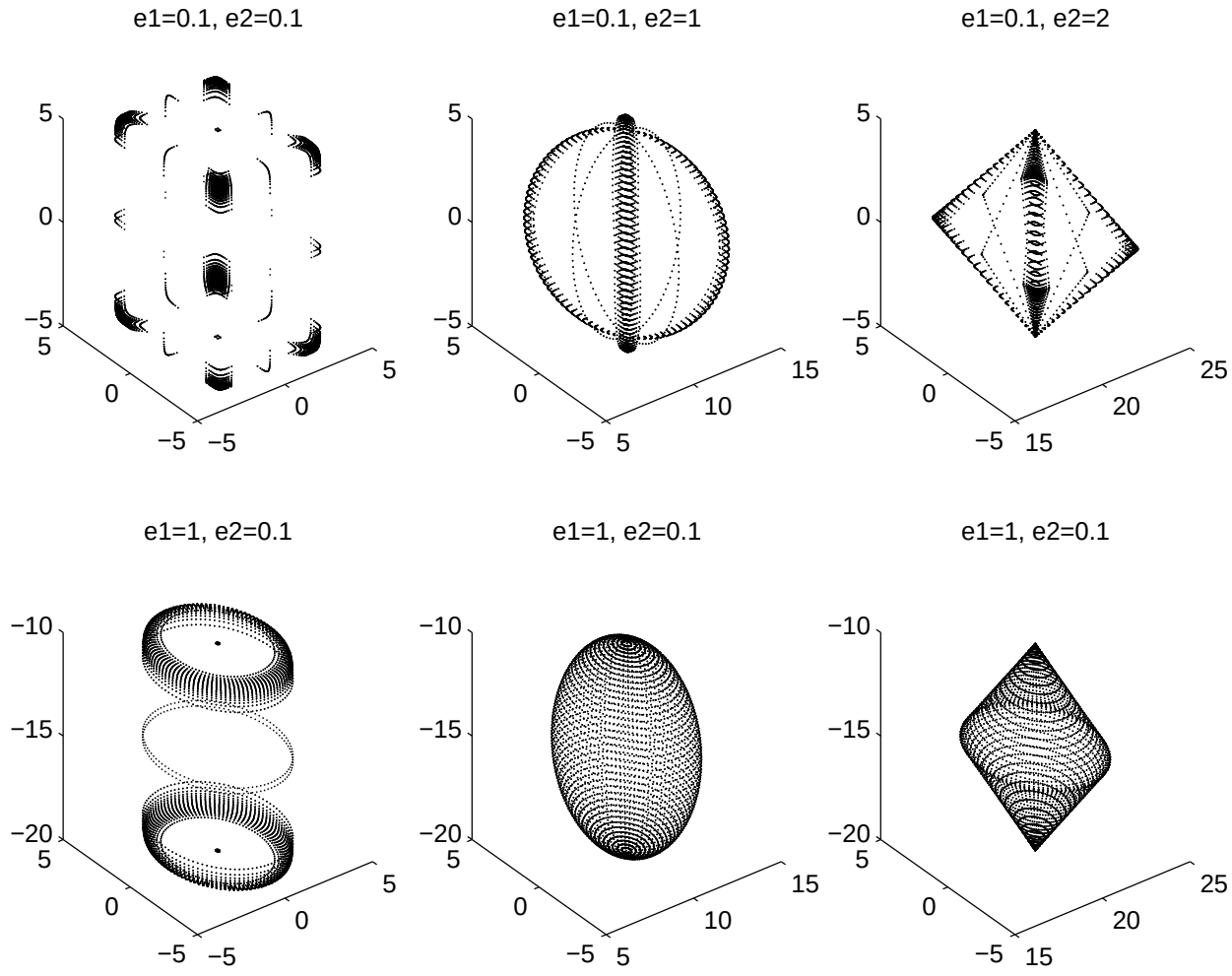


Figure 3.2: Superquadrics with various values for the shape parameters

Uniform superquadric sampling is often required for model fitting of high accuracy, and as we will see in 5, this is the case in our application as well. A variety of approaches has been proposed for addressing this problem [10], [122], [54], [103]. A comparative study is conducted in [26], where it is proved that the sampling proposed in [122] produces superior results. Hence, we used this approach for superquadric sampling. Figures 3.3 and 3.4 demonstrate the obtained results for superellipses and superquadrics of various shapes respectively: The points on the models in both figures are (almost) uniformly sampled.

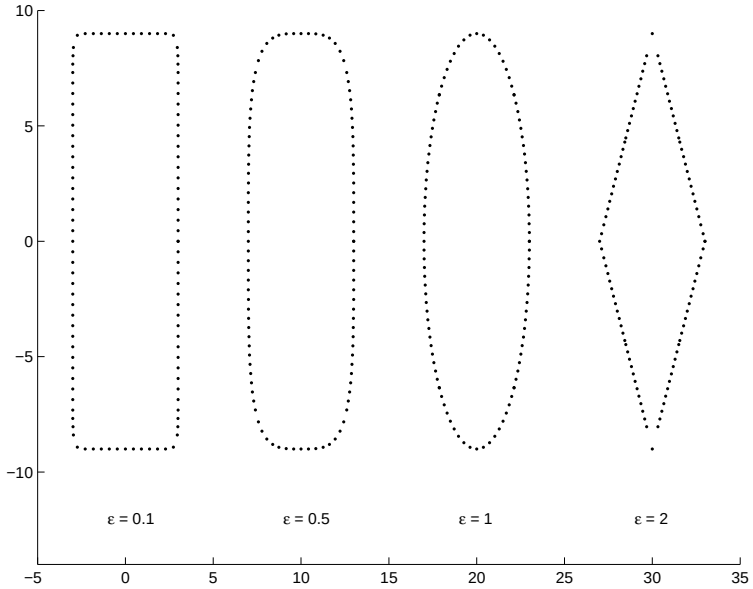


Figure 3.3: Uniformly sampled superellipses

3.4 Box-like superquadrics

By constraining the shape parameters of the models in the range $[0.1, 0.3]$ the models take the form of our box-like objects. This is illustrated on the leftmost images of figures 3.2, 3.4. A box-like superquadric is also shown in figure 3.5 (a). The models are placed at the origin of the model coordinate system, the axes of which will be hereinafter denoted as $\mathbf{X_m}, \mathbf{Y_m}, \mathbf{Z_m}$. In all these figures the values of the shape parameters have been set to $\epsilon_1 = \epsilon_2 = 0.1$.

3.5 Global deformations

The expressive ability of superquadrics can be considerably increased by incorporating global deformations to the models. In [148], [77], model *bending*, and *tapering* is defined. More specifically, global bending along a circular section by introducing two additional deformation parameters, has been used in an earlier version of our system [87]. However, this kind of bending does not resemble the deformations occurring to our objects. Parabolic deformations produce objects with greater similarities to the real deformations happening to the piled target objects on one hand (as illustrated in Fig. 3.5 (b)), and results into considerably smaller fitting error residuals than both those obtained from undeformed or from circularly

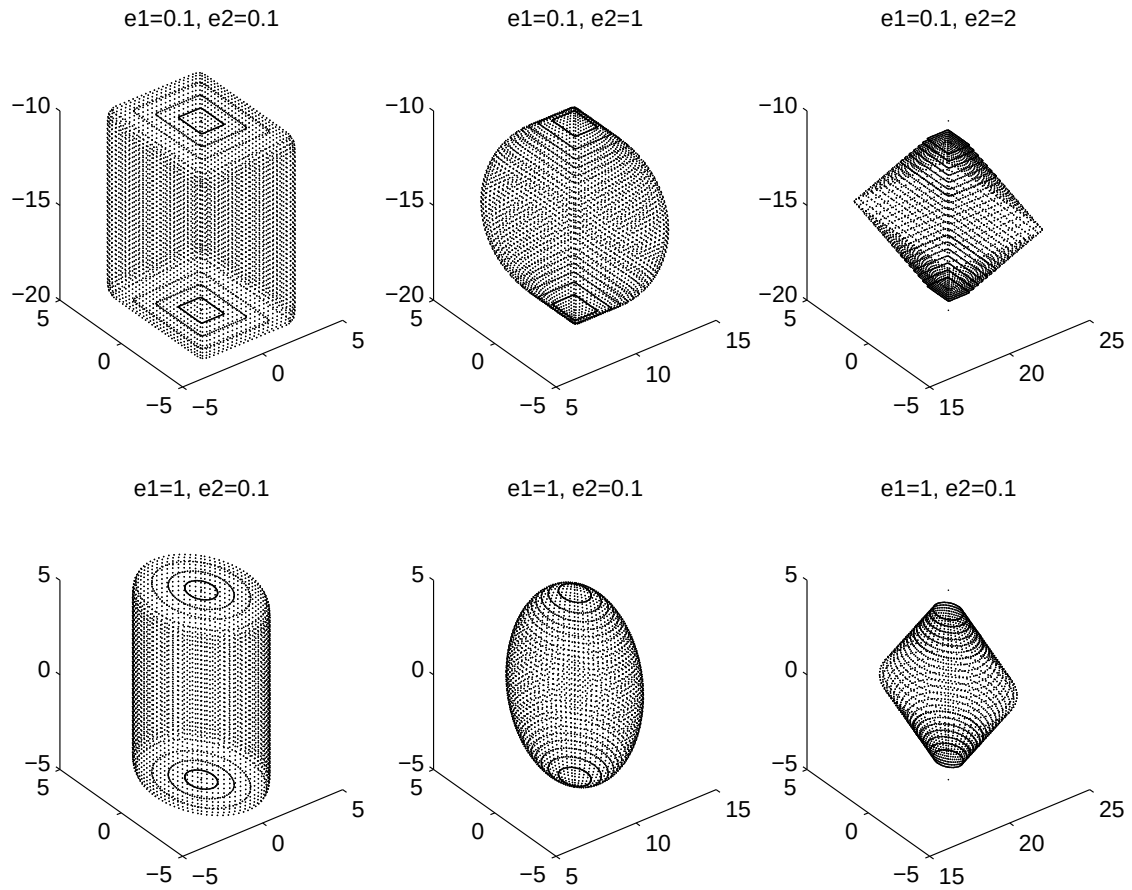
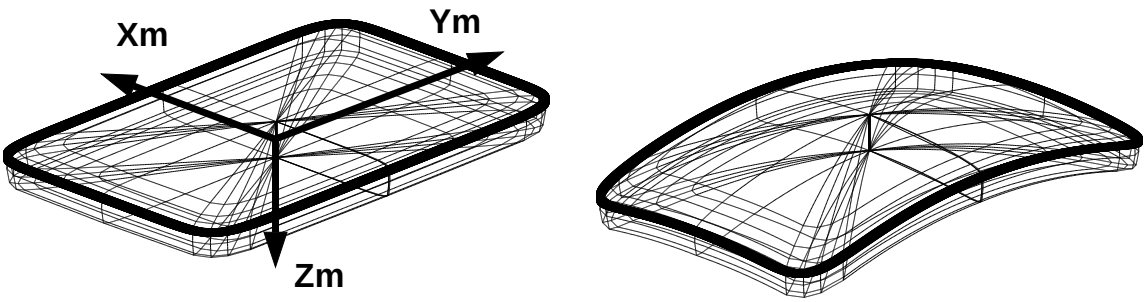


Figure 3.4: Uniformly sampled superquadrics



(a) Undeformed Superquadric (b) Parabolically deformed Superquadric

Figure 3.5: Box-like Superquadrics

bended models on the other.

The implementation of parabolic deformations is analogous to the technique presented in [148]. We define a plane perpendicular to the $\mathbf{Z_m} = 0$ plane of the model coordinate system crossing its origin, the bending plane, which forms an angle α with the $\mathbf{X_m}$ axis (see Fig. 3.6 (a)). The intersection of the bending plane with the $\mathbf{Z_m} = 0$ plane defines the bending axis $\mathbf{R}$. Three steps comprise the deformation process which transforms a point $\mathbf{v_m}(x_m, y_m, z_m)$ on the surface of a non-deformed superquadric to the corresponding point $\mathbf{v_d}(x_d, y_d, z_d)$ of the deformed superquadric: Firstly $\mathbf{v_m}(x_m, y_m, z_m)$ is projected to the bending plane so that point $\mathbf{v_i}(r(x_m, y_m), z_m)$ is acquired (Fig. 3.6 (a)), the coordinates of which are now expressed in the bending plane coordinate system. The quantity $r(x_m, y_m)$ is computed in (3.6)-(3.7).

$$\begin{aligned} \beta &= atan2(y_m, x_m) && (3.6)\\ r &= \cos(\alpha - \beta)\sqrt{x_m^2 + y_m^2} && (3.7) \end{aligned}$$

Secondly the deformation on the bending plane is performed: The superquadric bends in such a way that the point $\mathbf{v_i}$ transforms to $\mathbf{v_j}(r_d, z_d)$. This is done via a user defined deformation function $f(R)$, a parabola in our case, so that $z_d = z_m + f(r_d)$ (Fig. 3.6 (b)). Finally, the point $\mathbf{v_j}$ is back projected to a plane parallel to the bending plane passing from the original point $\mathbf{v_m}$ to obtain the final point $\mathbf{v_d}$ (see Fig. 3.6 (a)).

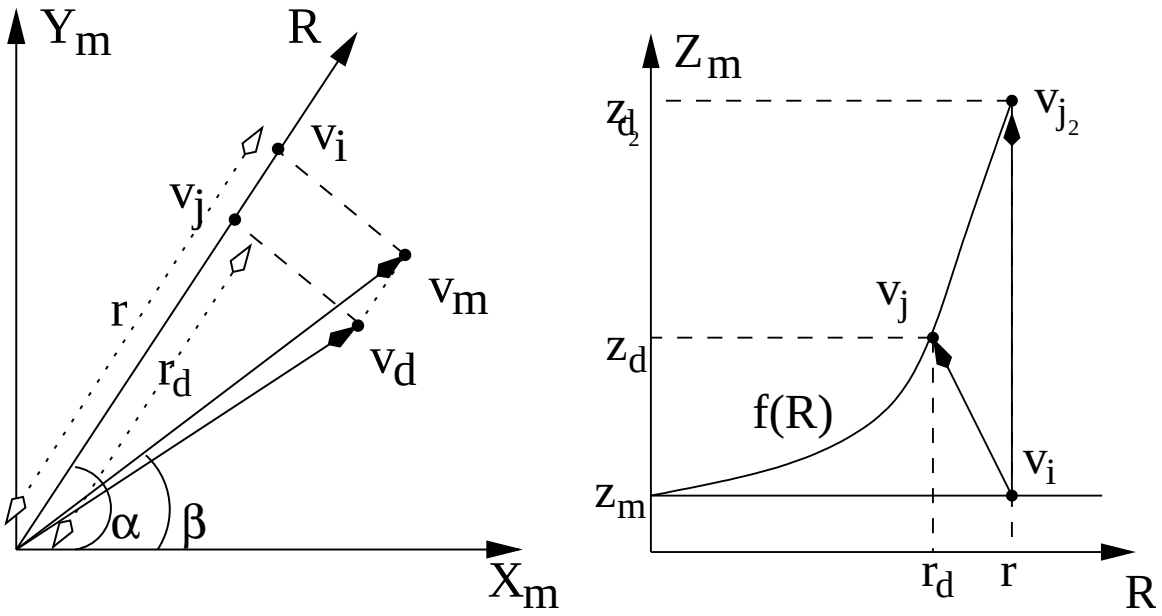
We introduce two deformation parameters: The parameter b, which expresses the curvature of the parabola at its vertex c, as well as the position of the vertex. These two along with the angle of the bending plane α comprise the set of deformation parameters. The overall deformation mechanism for recovering the coordinates of the final point $\mathbf{v_d}(x_d, y_d, z_d)$ given the initial point $\mathbf{v}(x_m, y_m, z_m)$ is expressed by the set of equations (3.8).

$$\begin{aligned} x_d &= x_m + (r_d - r)cos(\alpha)\\ y_d &= y_m + (r_d - r)sin(\alpha) && (3.8)\\ z_d &= z_m + f(r_d) = z_m + b(r_d - c)^2 \end{aligned}$$

Ideally, the deformation process should be defined so that the superquadric spine to which the point $\mathbf{v_m}$ belongs bends to a parabolic segment of the same length (see Fig. 3.6 (b). This implies that the abscissa r_d of $\mathbf{v_j}$ should be such that the length of the parabolic arc ending at $\mathbf{v_j}$ is equal to r. Therefore, the desired quantity r_d should satisfy (3.9), which cannot be analytically solved, thus introduces inefficiency in the parameter recovery process.

$$\int_0^{r_d} \sqrt{1 + f'(R)^2}\, dR = \int_0^{r_d} \sqrt{1 + \left(2b(R - c)\right)^2}\, dR = r \qquad (3.9)$$

We can avoid this problem by considering that since our objects are full of material, the parameter b is expected to be close to zero. Actually, the experimentally observed value of b is in the order of 10^{-3}, which allows us to set $r = r_d$. Consequently, the deformation process brings the point $\mathbf{v_i}$ to $\mathbf{v_{j2}}$ (see Fig. 3.6 (b)).



(a) Projection on the bending plane (b) Parabolic deformation

Figure 3.6: Superquadric global deformations

Finally, we introduce an additional parabolic deformation along the axis $\mathbf{X_m}$ controlled by the parameter d, with its vertex lying at the origin of the axis, to express slight bending occurring simultaneously with the bending along the arbitrary axis $\mathbf{R}$. Following the same procedure, we end up with a simple set of equations (3.10)-(3.12) describing the overall global bending deformations.

$$x_d = x_m \tag{3.10}$$
$$y_d = y_m \tag{3.11}$$
$$z_d = z_m + b(r_d - c)^2 + dx_m^2 \tag{3.12}$$

3.6 Superquadrics in general position

Up to know we assumed that the center of gravity of the models coincides with the origin of the *model coordinate system*, as in fig. 3.5 (a). However, the object recovery task requires model representation in a general position of the *sensor* (or world) coordinate system. Six additional parameters are considered to allow for this purpose. Three of them express the translation of the model relative to the origin of the sensor coordinate system, hereinafter referred to as p_x, p_y, p_z. The remaining three express the orientation of the model, and are

expressed by means of the Euler angles ϕ, θ, ψ. More specifically, ϕ denotes the rotation of the model about the $\mathbf{Z_m}$ axis of the model coordinate system followed by the rotation θ about the new $\mathbf{Y_m}$ axis, and finally a rotation ψ about the new $\mathbf{Z_m}$ axis. Thereby, the model parameter vector $\mathbf{p}$ incorporates 15 parameters in our system: 5 expressing the scale and shape $a_1, a_2, a_3, \epsilon_1, \epsilon_2$, 6 the pose $p_x, p_y, p_z, \phi, \theta, \psi$, and 4 the parabolic deformations α, b, c, d. In conclusion, the model parameter vector $\mathbf{p}$ in the general position, has as follows:

$$\mathbf{p} = (a_1, a_2, a_3, \epsilon_1, \epsilon_2, p_x, p_y, p_z, \phi, \theta, \psi, \alpha, b, c, d) \tag{3.13}$$

3.7 Superquadric fitting in range data

Fitting superquadric models in range data is a problem that has been studied for many years. The most widespread method for superquadric parameter recovery via model fitting, is initially presented in [148], and extensively used in [77] for recovering multiple models.

According to this approach superquadric fitting to a set of three dimensional points x_{si}, y_{si}, z_{si}, $i = 1..n$ is based on the minimization of the sum of squares of *algebraic distances* of the points to the model. According to (3.4), (3.5), this sum amounts to:

$$L_a(\mathbf{p}) = \sum_{i=1}^{n} \left((F(\mathbf{p}; x_{si}, y_{si}, z_{si}) - 1) \right)^2 \tag{3.14}$$

Note, that in order to calculate the algebraic distance of a point in the sensor coordinate system to the model, the point is firstly back-transformed in the deformed model coordinate system, and then to the undeformed model coordinate system (see [77]).

Superquadric fitting can be performed by minimizing L_a with respect to $\mathbf{p}$. However, in [148], [77] it is observed that this approach suffers from two major problems: Firstly, minimization of L_a favors models with larger volumes, given they fit the data equally well as models with smaller volumes. This is a problem in cases where the data do not sufficiently constrain all three scale parameters of the model, which happen when not all sides of an object are visible at the same time. Secondly, the function L_a favors models of high curvature.

In [148], [77], an attempt to resolve the former problem is made, by multiplying the inside-outside function with the term $\sqrt{a_1 a_2 a_3}$, which is proportional to the volume of the model, so that the volume is as well minimized. The latter problem is addressed by raising F to the power ϵ_1. This makes the fitting function independent from the shape of the superquadric regulated by ϵ_1. In conclusion, in [148], [77], fitting is performed my minimizing the function L_r shown in (3.15), with respect to the model parameter vector $\mathbf{p}$.

$$L_r(\mathbf{p}) = \sum_{i=1}^{n} \left(\sqrt{a_1 a_2 a_3} (F^{\epsilon_1}(\mathbf{p}; x_{si}, y_{si}, z_{si}) - 1) \right)^2 \tag{3.15}$$

The major advantage of using (3.15) for model fitting is the computational efficiency of the fitting process, since L_r has a closed form. Minimization of L_r is performed using a variant of the Levenberg- Marquardt method [64]. However, despite the fact that the robustness

of the method is increased with respect to parameter recovery via minimization of (3.14), the problems related to the usage of algebraic distanced for fitting have not been completely eliminated: As observed in [77], despite the introduction of the exponent ϵ_1, L_r is not completely uniform in all directions and depends on the ratio of the size parameters: In the direction of the longer dimensions of the models, the function grows slower. Besides, minimizing the quantity $F^{\epsilon_1} - 1$ still favors models of large volumes. The reason why this happens is clarified by the following equation, which was first shown in [163]:

$$F^{\epsilon_1}(\mathbf{p}; x_s, y_s, z_s) - 1 = \frac{d}{|\mathbf{r_S}|}\left(\frac{d}{|\mathbf{r_S}|} + 2\right). \tag{3.16}$$

In this equation, given $P(x_s, y_s, z_s)$ a three dimensional point, considering $\mathbf{OP}$ the vector connecting the center of gravity of the model to the point, as in fig. 3.7, and given $\mathbf{P_0}$ the intersection of $\mathbf{OP}$ with the model, $\mathbf{r_S} = \mathbf{PP_0}$. Hence, what (3.16) implies is that since $|\mathbf{r_S}|$ increases when the volume of the model increases, the fitting function L_r still favors models with larger volumes.

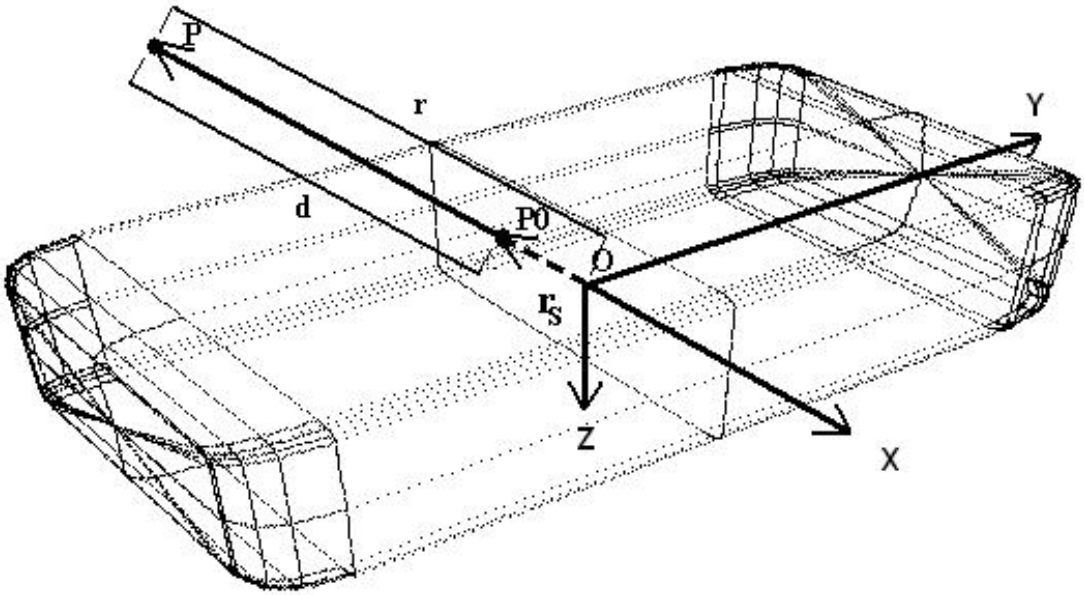


Figure 3.7: Radial Euclidean Distance of a point to a model

In order to resolve the problems associated with minimization of the expression (3.15), we tried minimizing the *Radial Euclidean Distances* of all points to the model, the value of which is illustrated in the subsequent equation:

$$\begin{aligned} |d| &= |\mathbf{r} - \mathbf{r_S}| \\ &= |\mathbf{r}||1 - F^{-\frac{\epsilon_1}{2}}(x, y, z)| \\ &= |\mathbf{r_S}||F^{\frac{\epsilon_1}{2}}(x, y, z) - 1| \end{aligned} \tag{3.17}$$

The quality of the results obtained using the radial euclidean distance, did not compensate the additional computational costs introduced to the fitting process. Various other distance

metrics have been recently proposed in the literature for superquadric fitting [134], [172], [171]. The superquadric model fitting based on orthogonal distances [4], [57] seems promising, but with considerably higher computational costs.

In this work, we performed superquadric fitting using minimization of (3.15), as in [77]. The main reason for using this approach is its computational efficiency, since it is able to perform model fitting to about 1000 range point in less then 5 seconds in a Pentium 4, 2.8GHz PC. The inaccuracies in the model recovery introduced by this approach, do not affect our recovery process to a big extend. The reason why this happens is that, as we will see in chapter 5, *boundary information* is as well employed for model fitting. A quantitative assessment of our model fitting method combining range and boundary information, will be presented in chapter 6 of the thesis.

3.8 Graspable superquadrics

In this thesis, we focus on superquadric recovery for robotic grasping purposes. In particular, it is assumed that the target objects are piled on platforms from which they have to be automatically removed by the robotic hand. Given an input range image corresponding to a specific object configuration on the platform, the object recovery approach in many cases localizes more than one object. Hence, it must be ensured that when one of these objects is removed by the robotic hand, the position of the others do not change. This does not happen, (or rather it happens very rarely), when the object being removed is not occluded. For this reason non occluded objects are referred to as *graspable objects* in this work.

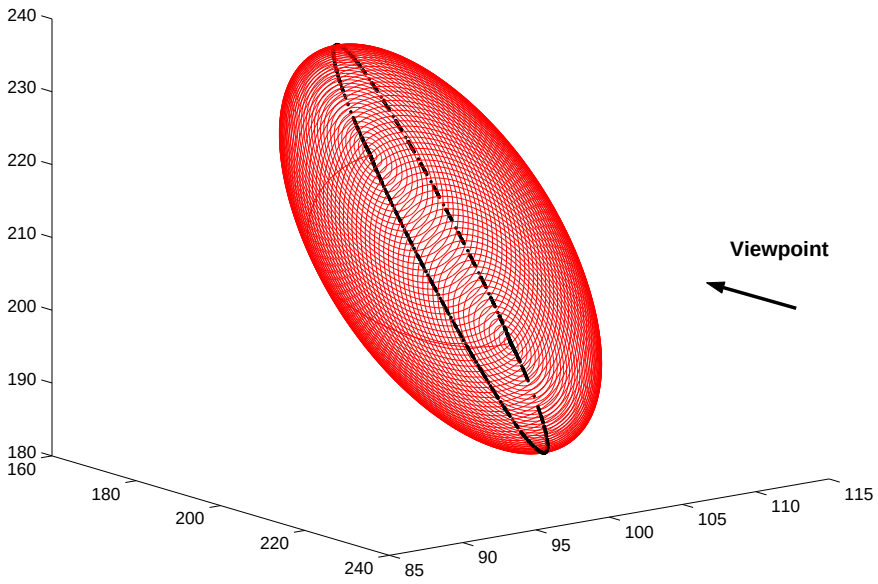


Figure 3.8: Superquadric rim

More specifically, a superquadric object cannot be occluded by other objects, and thereby is

graspable, if no objects lie on its visible part, that is the part of its surface exposed to the sensor. Hence, given a set of three dimensional points assumed to correspond to a unique superquadric object, in order to determine whether the object is graspable we should check if the points span the entire visible part of the model's surface. This can be accomplished by means of the superquadric *RIM*. The superquadric rim is by definition a closed three dimensional curve which partitions the object surface into a visible and an invisible part. Assuming that the *boundary* or convex hull of the set of points is extracted, in order to determine if the object is graspable we can fit the curve to the boundary of the point set. Low fitting error residual indicates that the object is graspable.

Assuming that the origin of the sensor coordinate frame is an infinitely distant point, and the depth axis of the sensor coincides with the $\mathbf{Y}$ axis of the frame, a point on the surface of the superquadric lies on the rim only if the viewing unit vector $\mathbf{Y} = \mathbf{v_s} = (0, 1, 0)$ is perpendicular to the surface normal vector at the point, that is, when:

$$\mathbf{v_s}\mathbf{n}(\eta, \omega) = 0 \tag{3.18}$$

Given n_x, o_y, a_y the second row of the rotation matrix $\mathbf{R}$ relating the model to the sensor coordinate system (defined through the elements ϕ, θ, ψ of the model parameter vector), the preceding equation results in the following relationship between η, and ω for points on the rim:

$$\eta(\omega) = \arctan\left\{-\frac{a_3}{a_y}\left(\frac{n_y}{a_1}\cos(\omega)^{2-\epsilon_2} + \frac{o_z}{a_2}\sin(\omega)^{2-\epsilon_2}\right)^{\frac{1}{2-\epsilon_1}}\right\} \tag{3.19}$$

In other words, points $\mathbf{R_p}$ on the rim of a superquadric with parameter vector $\mathbf{p}$ are obtained as follows:

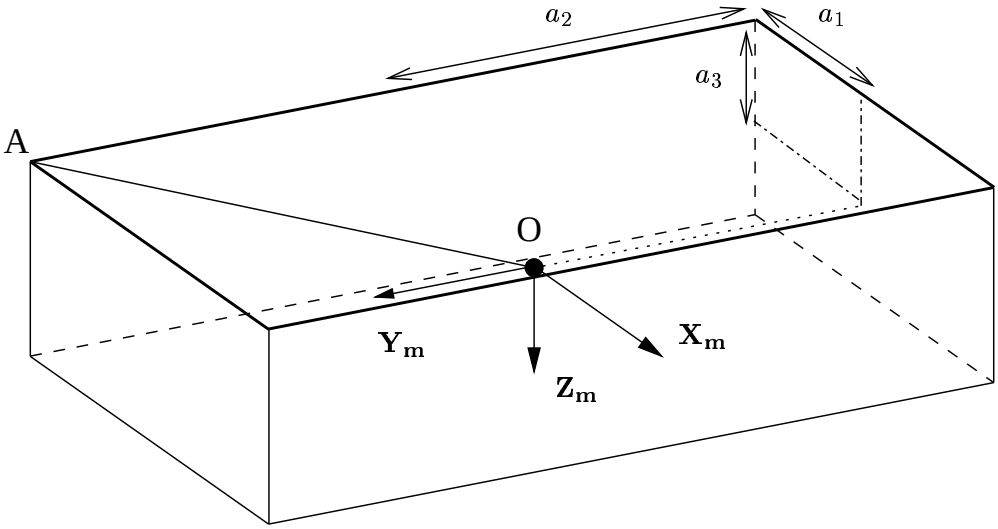
$$\mathbf{R_p}(\omega) = \mathbf{S_p}(\eta(\omega), \omega), \tag{3.20}$$

where $\mathbf{S_p}$ are points on the superquadric surface obtained via (3.4), ω ranges in $[-\pi, \pi]$, and $\eta(\omega)$ is defined in (3.19). The reader is referred to [26], [77] p. 29 for a more formal definition of the rim. The rim of a superquadric related to a particular viewpoint is illustrated in fig. 3.8.

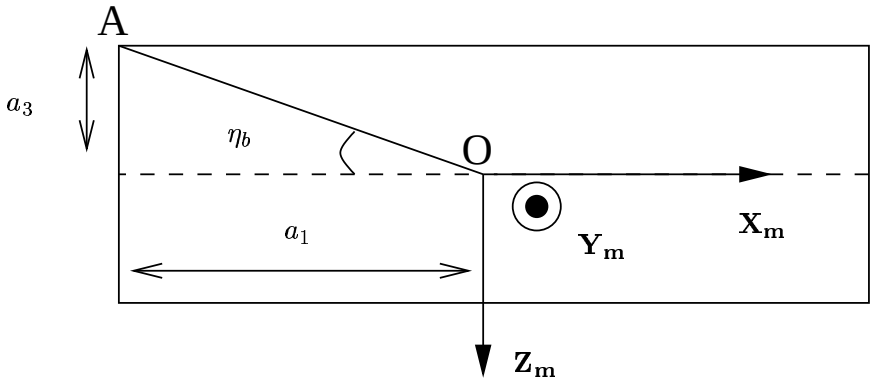
Our application concerns box-like superquadrics in particular, as those illustrated in figures 3.5, and 3.7. These objects usually have a biggest side, which is assumed to be parallel to the plane defined by the $\mathbf{X_m} - \mathbf{Y_m}$ axes. When the objects are placed on platforms, they expose this side to the laser source. Besides, given that a vacuum gripper is used to grasp the objects from the center of gravity of their exposed surface (see chapter 2 for details), an object is considered graspable in our case if its exposed surface is not occluded. Thus, in order to determine if a box-like object is graspable, it is enough to consider the *boundary of the exposed surface* of the object, instead of its rim.

The boundary of the exposed surface of an object, which will be simply referred to as *model boundary* in the chapters that follow, is illustrated as the bold continuous curve in figure 3.5. This curve is a three dimensional superellipse, and therefore is much easier to be dealt with than the rim, which can take more complicated forms. Three dimensional points on the curve can be obtained using the explicit superquadric equation as in (3.21):

$$\mathbf{E_p}(\omega) = \mathbf{S_p}(\eta_b, \omega), \omega \in [-\pi, \pi], \tag{3.21}$$



(a) Model in $3D$



(b) Frontal side view

Figure 3.9: Boundary of exposed surface of a box-like superquadric

where η_b is a constant, to which the following value is assigned:

$$\eta_b = \arccos(\frac{a_3}{\sqrt{a_1^2 + a_2^2 + a_3^2}}) - \frac{\pi}{2} \tag{3.22}$$

The reason why this is so, is shown in fig. 3.9: Fig. 3.9 (a), shows a box-like model in space, the boundary of the exposed surface of which is illustrated in bold. Fig. 3.9 (b), shows a side view of the model. The angle η for which points on the boundary of the exposed surface can be obtained by means of the explicit equation for the model, equals to the angle η_b in this figure. The value for η_b in eq. (3.22) is acquired by calculating the dot product of the vector $\mathbf{OA} = (-a_1, a_2, -a_3)$ with the vector $\mathbf{Z_m} = (0, 0, 1)$.

In chapter 5, we will demonstrate the way in which the boundary of the exposed surface of the models can be employed for recovering graspable objects in range images. Besides, we will show some initial experiments toward the recovery of generic superquadrics, using the superquadric rim. As discussed earlier, graspable object recovery is related to fitting of the boundary of the exposed surface or the rim of the models. The accuracy of the fitting operation, depends on the sampling of points on these curves: The more uniform the sampling is the more accurate the fitting result will be. Uniformity in model sampling discussed in section 3.3 is important to our work exactly for this reason.

3.9 Conclusions

The goal of this chapter was the introduction of the superquadrics, the entities used for employing the target objects in our application. More specifically, after defining the models, we discussed issues like uniform sampling, global model deformations and model fitting to range data, which are important parts of our strategy for recovering multiple superquadrics from images. Finally, we touched upon the issue of recovering graspable objects, and showed in which way special curves on the surface of our models, i.e. the boundary of the exposed surfaces of the models and the superquadric rim, can be employed for this purpose.