# Segmentation and superquadric modeling of 3D objects

Laurent Chevalier
LIRIS
Université Lyon1
Bat. Nautibus,
8, av. Niels Bohr,
69622 Villeurbanne cedex
FRANCE

Fabrice Jaillet
LIRIS
Université Lyon1
Bat. Nautibus,
8, av. Niels Bohr,
69622 Villeurbanne cedex
FRANCE

Atilla Baskurt
LIRIS
Université Lyon1
Bat. Nautibus,
8, av. Niels Bohr,
69622 Villeurbanne cedex
FRANCE

lchevalier@liris.univ-lyon1.fr

## ABSTRACT

A new model for representing an unorganised 3D data points set is proposed. Based on superquadrics, this model allows to describe the points set with a union of superellipsoids. Two different segmentation and modeling methods are developed in order to determine the whole model: a region growing approach and a split and merge one. This second method leads to a low sensitive model compared to the one obtained by the region growing. The model is simple and compact: only 11 parameters are needed per superellipsoid. It seems promising for 3D object compression and 3D object indexing and retrieval. As the topological relations of the superellipsoids are known, the model can be associated to a graph. The graph theory can thus be used in order to compare and to measure the similarity between 3D objects.

**Keywords**
Superellipsoid, 3D segmentation, 3D object compression, 3D object indexing and retrieval

## 1 INTRODUCTION

This study concerns the segmentation and modeling of an unorganised 3D data points set. The constraints imposed to the model are related to the concerned applications:

- coarse visualisation of the 3D object represented by the points set;
- indexing and retrieval of similar 3D objects from dedicated databases providing a descriptor;
- compression of the data set for transmission and storage.

We need a simple descriptor which allows the representation of a 3D data set using a very compact model and the reconstruction of a coarse version with a controlled distortion rate (fuzzy approximation). For this purpose, we choose to describe 3D objects with a set of primitives which are superellipsoids.

Superellipsoids have already been used to model 3D object [3, 7, 10]. In the majority of these studies, range images of the 3D object or the 3D scene are used for modeling. Range data have regular layout and are organised in the sense that neighbouring points on the image are mostly neighbouring points in space. We want to deal with more general 3D data without any a priori knowledge. The 3D points set considered in this study is irregular and unorganised.

We propose to compare two different ways to obtain the descriptor. The first one is an extension of the region growing method proposed by Leonardis [7, 8]. The second way is an original split and merge approach that we have developed.

Section 2 defines the primitive surface descriptor: the superellipsoid. In section 3, we show how 3D data

***Journal of WSCG, Vol.11, No.1., ISSN 1213-6972***
*WSCG2003, February 3-7, 2003, Plzen, Czech Republic.*
Copyright UNION Agency - Science Press

can be approximated using only one superellipsoid. Section 4 gives details on the two segmentation algorithms. Finally, the qualitative and quantitative performances of the two methods are illustrated in section 5.

## 2 SUPERQUADRICS AND SUPERELLIPSOIDS

The superquadric model [3, 4, 6] has been introduced in computer graphics by A.H. Barr in 1981. As an extension of quadric surfaces, four kinds of model can be distinguished: supertoroid, superhyperboloid with one or two sheets, and superellipsoid. As the last one is the only one that defines a closed surface without hole, it is usually the only one used in our domain area. For the same reason, we will restrict us to describe 3D objects with superellipsoids.

A superellipsoid is defined as the solution of the general form of the implicit equation:

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

In this equation, one can recognise an ellipsoid form, enriched with two more parameters ($\epsilon_1$ and $\epsilon_2$) that allow to control the shape curvature. As for the ellipsoid case, the $a_1$, $a_2$, $a_3$ parameters are scale factors on $x$, $y$ and $z$ axis respectively.

This form provides an information on the position of a 3D point related to the superellipsoid surface, that is important for interior/exterior determination. We have:

- $f(x,y,z) = 1$ when the point lies on the surface;
- $f(x,y,z) < 1$ when the point is inside the superellipsoid;
- $f(x,y,z) > 1$ when the point is outside.

As an ellipsoid's extension, a superellipsoid is the result of the spherical product of two 2D models (two superellipses). Then, superellipsoids can be defined as a parametric model from this product:

$$S(\eta,\mu) = \begin{bmatrix} a_1 \cos^{\epsilon_1}(\eta)\cos^{\epsilon_2}(\mu) \\ a_2 \cos^{\epsilon_1}(\eta)\sin^{\epsilon_2}(\mu) \\ a_3 \sin^{\epsilon_1}(\eta) \end{bmatrix}, \quad \begin{matrix} -\frac{\pi}{2} \le \eta \le \frac{\pi}{2} \\ -\pi \le \mu \le \pi \end{matrix} \quad (2)$$

Being able to switch directly from the implicit to the parametric representation is one main point of the superellipsoid model. This is really an advantage, especially for sampling and rendering, because this is much more difficult with an implicit model.

Moreover, this is a compact model defined by only five parameters that permits to handle a large variety of shapes, including: ellipsoid ($\epsilon_1 = \epsilon_2 = 1$), parallelepiped ($\epsilon_1 \to 0$ and $\epsilon_2 \to 0$), cylinder ($\epsilon_1 = 1$ and $\epsilon_2 \to 0$)...(see figure 1). In our application, we constrain $\epsilon_1$ and $\epsilon_2$ to be less than 2, in order to have convex shape only. Obviously, the general position of the superellipsoid is obtained with the addition of three rotation parameters and three translation ones. Thus, only eleven parameters are required to describe a superellipsoid.

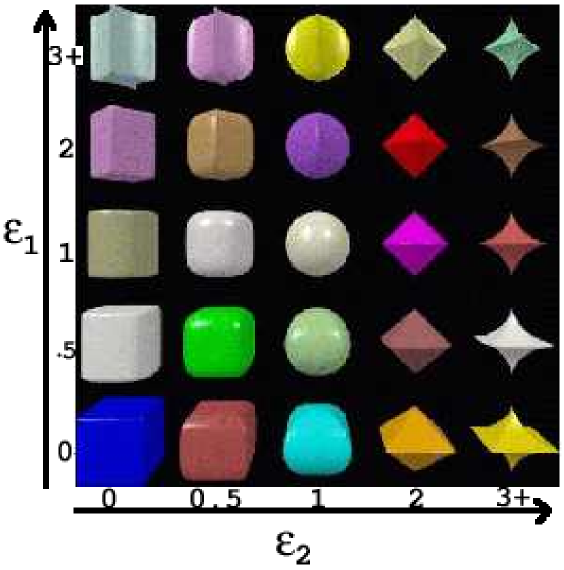


Figure 1: Examples of superellipsoids according to $\epsilon_1$ and $\epsilon_2$.

## 3 APPROXIMATION OF 3D DATA WITH ONE SUPERELLIPSOID

Given a set of N unstructured 3D data points, the first challenge is to determine the parameters of our model for fitting with a global distortion constraint. The method was formerly proposed by F. Solina in 1990 [11] and is the most popular at the moment. This is a least squares fitting method. The following global distortion is minimised:

$$\sum_{i=1}^{N} d(x_i,y_i,z_i)^2 \quad (3)$$

where $d(x,y,z)$ is the distance between a 3D data point and the superellipsoid surface (see below).

When the points set is not closed, several superellipsoids may approximate it correctly. Solina resolved this difficulty by imposing a constraint which favours the small superellipsoids. This is achieved by applying the coefficient $\sqrt{a_1a_2a_3}$ to the global distortion.

To compute this minimisation, one needs to know how to calculate the distance between a point and the surface. This computation will often be used during the process and the common Euclidean distance is far too expensive. Usually, the distance is estimated with an approximation based on the implicit form of the superellipsoid. Solina proposed this approximation for the distance:

$$d(\mathbf{a}, x, y, z) = f(\mathbf{a}, x, y, z)^{\frac{\epsilon_1}{2}} - 1 \qquad (4)$$

where **a** represents the superellipsoid parameters and $(x, y, z)$ the point coordinates.

This leads to $D$, the mean distortion per point given by:

$$D(\mathbf{a}, x, y, z) = \frac{\sqrt{a_1 a_2 a_3}}{N} \sum_{i=1}^{N} (f(\mathbf{a}, x, y, z)^{\frac{\epsilon_1}{2}} - 1)^2 \qquad (5)$$

To minimise $D$, a non-linear regression method is required. The Levenberg-Marquadt approach is a numerical method that combines a gradiant and a quadratic descent method [1]. It is most of the time performant (especially for strongly constrained system) and thus is widely used. Other approaches can also be considered. The genetic algorithm gives better results but the computation cost is not acceptable. Downhill simplex method [9] could be used too but gives results equivalent to those obtained by Levenberg-Marquadt.

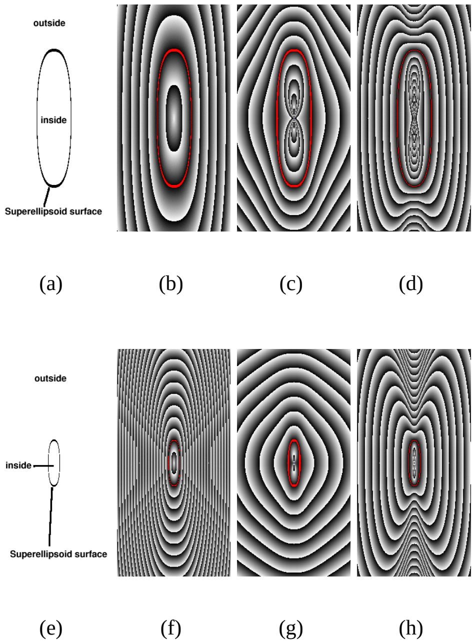
The previous distance $d$ is size dependent. That conducts to erroneous results when comparing two superellipsoids with different scales, and also in oblong cases. To improve the approximation and especially when $d$ is far from the Euclidean distance (figure 2), other distance estimations have to be chosen. The radial Euclidean distance [2] (that is the distance between the point $P$ and the intersection of the $OP$ line and the surface, where $O$ is the superellipsoid centre) is not more expensive to compute and gives better results because it is adjusted according to the scale (figure 2c and 2g).

$$d(\mathbf{a}, x, y, z) = ||\overrightarrow{OP}|| * (f(\mathbf{a}, x, y, z)^{\frac{\epsilon_1}{2}} - 1) \qquad (6)$$

Another estimation can be an approximation of the Euclidean distance[8] based on Taubin approach [12]:

$$d(\mathbf{a}, x, y, z) = \frac{|f(\mathbf{a}, x, y, z) - 1|}{||\overrightarrow{\bigtriangledown} f(\mathbf{a}, x, y, z)||} \qquad (7)$$

which is more expensive because of the gradiant computation but more accurate when the point is near the surface (figure 2d and 2h).



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 2: Equidistant lines of an oblong superellipsoid with Solina's estimation (b,f) radial Euclidean distance (c,g) the Taubin's approximation of the Euclidean distance(d,h).

# 4 DESCRIPTION OF 3D DATA WITH A SET OF SUPERELLIPSOIDS

The main motivation of this study is to approximate a set of unstructured 3D points using a set of superellipsoids as presented in section 2. The main difficulty consists in partitioning or segmenting the data in order to obtain a compact set of primitives and to provide a local approximation of good quality.

We develop two different algorithms. The first one is an extension of the approach proposed by Leonardis [7, 8], and the second is an original method that we propose [5].

Note that we use the distance from equation 6 to perform the different approximations for these two methods.

## 4.1 The region growing approach

In 1994, Leonardis has proposed a method to model range data with a set of superellipsoids [7, 8]. He uses a algorithm based on the region growing principle. The method can be divided in three steps: the seeds initialisation, the growing process and the selection process. The algorithm starts with the first point and then alternate with the last two at different occurences.

Leonardis applied this method to deal with range images. Such data are quite regular and well organised (i.e. the neighbours of a point are known and these are almost always at the same distance). We extend this method to any unorganised 3D data points.

### 4.1.1 Seeds initialisation

The method is initialised with the creation of a seeds set (figure 3). A seed is a small 3D data points set which can be modeled with a superellipsoid.

The space is partitioned following a 3D grid. We model with a superellipsoid the content of every cell of this grid. If the distortion of the approximation is less than a threshold, this new object is included in the active seeds set, else it is rejected.

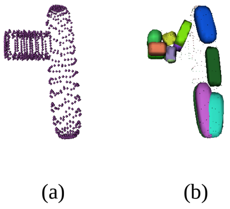


Figure 3: Seeds initialisation (a) Original data (b) Seeds.

### 4.1.2 Growing process

This procedure increases slowly the size of all the active seeds until their associated points set corresponds to full parts of the 3D object.

For each active seed, new points are included in order to make it grow. The difficulty is to determine which points belong to the same part. In order to reduce the research area, we restrict us to the neighbouring points of the seeds, which are high potential candidates.

Then, the approach consists into fitting the union of these candidate points and the ones of the growing seed with an ellipsoid. If the distortion is less than a threshold, all candidate points are merged in the growing set. Otherwise, considering that the seed can no more grow, it is removed from the active seeds list and set as inactive. The same procedure is applied for each active seed until all seeds are fully grown. Figure 4 shows these different states during the region growing and selection process, before obtaining the final descriptor.

Some problems remain with our adaptation to a set of unorganised points, especially in the choice of the neighbourhood. We use the $k$-nearest points of the seed, but they are less relevant than in the case of regularly spaced data (like in range images). Thus, our seeds may not grow enough with this method. To improve that, we have chosen to not reject the whole candidate points set if it is not appropriate. We split randomly the new points set in two smaller parts. We try to add each new subset and keep the subset that produces best results. The splitting process is repeated until no more point can be added to the seed, or until the approximation satisfies with the growing criteria and in this case the subset is added to the seed. Another delicate point is the determination of $k$ for the number of neighbours in the 3D case. When $k$ is too high, points belonging to an other part of the object can be added. At the contrary, if it goes smaller, the seeds will certainly not grow correctly and sufficiently. In compensation, we also consider the maximum of distortion to avoid including aberrant points.

### 4.1.3 Selection process

The seeds initialisation process can accept many seeds per part of the object. More, after some growing steps, some superellipsoids can overlap and model almost the same points set. The aim of this third step is to remove redondant superellipsoids from the final descriptor.

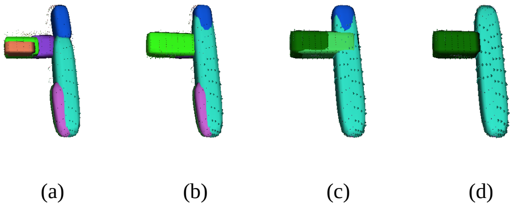


Figure 4: (a-c) Steps of the region growing and selection processes. (d) The final descriptor.

To achieve this selection, Leonardis proposed to maximize a function where the variable is a binary vector of all possible configurations of the next round ($Q$ is a $N$ vector and $M$ is a $N*N$ matrix where $N$ is the active and inactive seeds number):

$$\max_{Q}(Q.M)$$

where:

- $Q$ is the selection vector. $Q_i = 1$ means that the $i^{th}$ seed must be kept and $Q_i = 0$ means that the $i^{th}$ seed must be removed from the descriptor;
- $M$ is a matrix whose diagonal terms reflect the size and distortion of the seed, and off-diagonal ones correspond to intersecting points between two superellipsoids.

$Q$ is then computed by a greedy algorithm. Basically, when a seed is selected, its contribution to neighbouring seeds is reported on each row of the $M$ matrix.

This is well suited for range images as the coefficient range is more or less unvarying and can be fixed in advance. The same method is applied except that we produce our own matrix coefficients more adapted to irregularly spaced 3D points. A $M'$ matrix is defined with:

- $M'_{i,i} = K_1(1 - \frac{\chi_i}{T})\frac{n_i}{n}$ where $n_i$ is the size of the points set of the $i^{th}$ seed, $n$ the number of whole data points, $\chi_i$ is the mean distortion of the $i^{th}$ seed and $T$ the tolerated distortion contraint;
- $M'_{i,j} = -K_2\frac{\chi_{i,j}}{T}\frac{n_{i,j}}{n_i}$ where $i \neq j$, $n_{i,j}$ is the size of the intersection of the points set of the $i^{th}$ and the $j^{th}$ seed and $\chi_{i,j}$ is the mean distortion between the superellipsoid of the $i^{th}$ seed and the intersection points set of the $i^{th}$ and the $j^{th}$ seed.

The selection can be processed after one or more growth steps. But note that each seed rejected during the process of selection will no more be approximated. So, it is better to remove the seeds as soon as possible to accelerate the whole process, but we have to take care about not rejecting seeds with high growth capability.

### 4.1.4 Conclusion

As shown in the results presented in section 5, we succeed to model some 3D objects with this method. However, the results were not as good as expected, mostly because of the initial data.

Two problems are to be underlined with this method. The first is intrinsic to the approach. We do not know whether each part of 3D object will be in the final descriptor, because the seeds initialisation does not insure that there will be a seed occurrence in each part of the object. Moreover, we are not sure that an important seed has not been removed during the selection process.

The second problem is linked to the data type. The data points set is irregular and then not so suitable with the growth step. This is due to the difficulty to find an adequate matrix $M'$ and to set $K_1$ and $K_2$ correctly, because it depends on the scale of the object and on the varying distance between points in a same part of the object.

## 4.2 The split and merge approach

As an alternative to the previous algorithm and to deal with more general 3D data, we propose a new method based on split and merge approach [5].

Like usual split and merge algorithms, the method occurs in two sequential steps: the split and the merge procedures.

### 4.2.1 Split procedure

The aim of this first step is to split the data so that all points in a subset belong to the same part of the object (but two subsets can belong to the same part).

The split procedure must produce all the boundaries of the object because the latter step (i.e. merge step) will just remove superfluous ones but never create new regions.

This procedure is recursive and consists in the subsequent steps (figure 5):

1. The set of 3D points is fitted by one superellipsoid (as seen in section 3).
2. If the distortion $D$ is less than a threshold $T$ (the tolerated distortion constraint), the procedure is over. Otherwise, step 3 is processed.
3. The set of 3D points is splitted into two regions using the plane $P$ orthogonal to the inertia axis of this set ($P$ contains the centroid of the set).
4. Each half-subset is approximated independently using one superellipsoid. For each subset, the procedure is iterated from step 2.

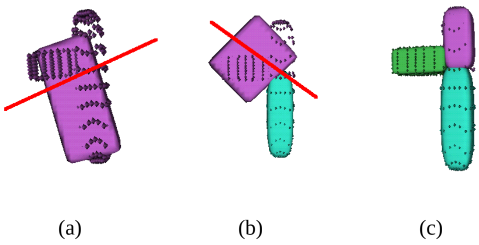

(a) (b) (c)

Figure 5: Split procedure. (a) Original data and first approximation. (b) First split step. (c) Second split step. All subsets are well approximated.

At the end of this process, we obtain a partition where each subset can be modelled with one superellipsoid with a distortion less than the threshold $T$. The threshold $T$ doesn't have to be the same that the quality $T'$ we want for the final descriptor. But note that we do not really know when the split process is over or if one subset will be merged with another in the second step. Each subset could arrive unchanged to the final descriptor and so the criteria could be no more compliant according to the global quality threshold.

The splitting plane is also of great importance (figure 5). The easiest way (and the fastest) is to split the bounding box in two equal smaller boxes, but that produces poor results. We choose to split in a much more intelligent way, using the properties of the inertia axis which is easy to process. We do not really require a thiner algorithm, because of the merge step that will balance the rough splitting.

This procedure may be compared with the seeds initialisation process of the region growing approach. A great difference is that no part of the 3D object can be omitted here. Each subset of points owns its superellipsoid and so will be in the final descriptor. Another important point is that the resulting superellipsoids do not overlap each other.

#### 4.2.2 Merge procedure

Normally, all the points in the subsets created in the previous procedure belong to the same part of the object. Now, the subsets belonging to the same part will be merged, in order to reduce the number of descriptors per part (ideally one superellipsoid per part).

This procedure minimises the number of superellipsoids without increasing the whole distortion. The topological relations between 3D points are taken into account while deciding to merge or not. The following steps are processed (figure 6):

1. For each subset of points, we determine the list of neighbouring subsets. The neighbourhood considered here is particularly large: a subset is neighbour of another subset if it is the direct (connex) neighbour (first order) or if it is the neighbour of a direct neighbour (second order).

2. We try to merge each subset with each of its neighbours. We merge the couple which minimises the approximation distortion $D$ if this one is less than a threshold $T'$ and if the sum of the size of the two superellipsoids is not smaller than the size of the new superellipsoid. The process continues from step 3, otherwise if all the distortions are greater than $T'$, the process is over.

3. A new set of superellipsoids is obtained. We go back to step 1.

Testing all couples of neighbours before to merge is tiresome, but this permits to be independant of the order in which merges are processed. Special care is taken to the size of the superellipsoids. Imagine that the points set represents a table, the procedure may try to merge two or more legs of this table. The approximation of the "two legs" will be usually a plate. As the legs are slim, the distortion will be small because all points will be near the surface: each leg at one extremity of the plate. The criteria that helps us to reject this kind of merge is the increase of the volume. The size of the plate is usually much greater than the sum of the size of the two legs (figure 7).

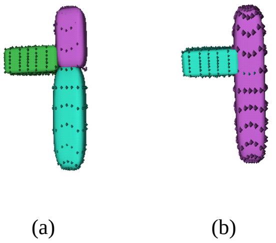

(a) (b)

Figure 6: (a) The merge procedure starts. (b) Two superellipsoids are merged. The final descriptor is the union of two superellipsoids.

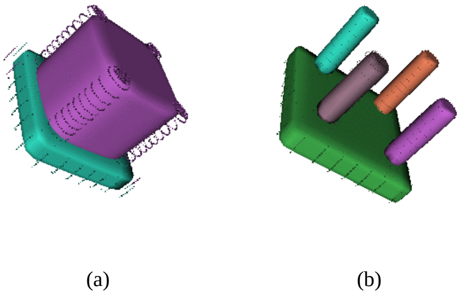

(a) (b)

Figure 7: Merging (a) without size control (b) with size control

#### 4.2.3 Conclusion

Our new segmentation method seems to be more adapted to our unorganised and irregular data than the former region growing approach. In our approach there is no reference to an a priori points relationship knowledge like neighbourhood.

Furthermore, the split and merge algorithm permits to keep topological relations between superquadrics. In fact, during the process, the improving descriptor is not only a set of unorganised superellipsoids but a graph where the node are superellipsoids and the edges label a neighbourhood relation. This is probably more difficult to arrange hierarchically the descriptors issued from the region growing process, because of the descriptor progression way, and the fact that seeds can intersect themselves.

## 5 RESULTS

We applied the two approaches on synthetic 3D objects obtained by combining superellipsoids. The data points set are sampled on the surface of these objects (figures 8a, 9a, and 10a).

For the object of figure 8, the final descriptor obtained with the two methods has the same number of primitives than the original synthetic model. The whole distortion is equivalent for these two approaches. In figure 9, the split and merge approach finds the exact number of superellipsoids (4) whereas the region growing conducts to a redundant descriptor (15 primitives). Note that the distortion is higher although the descriptor is more complex. Figure 10 gives another example with the split and merge method which converges perfectly to the synthetic data set with a quite good quality of approximation.

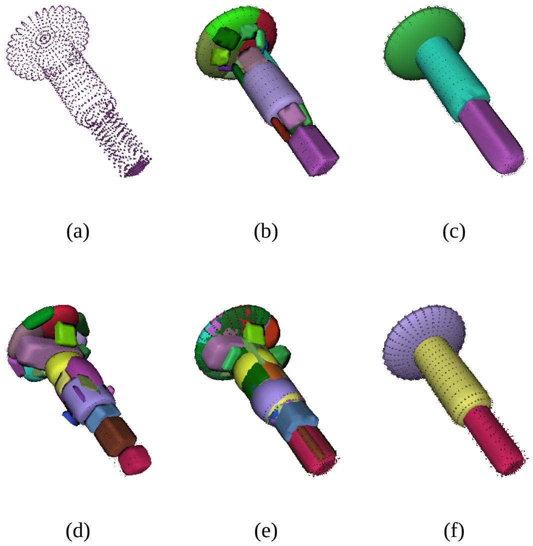


Figure 8: Model A (a) Original data (1746 points) (b) The end of the split step (c) Split and merge result ($D = 0.0020$) (d) Region growing seeds (d) An intermediate step of region growing (f) Region growing result ($D = 0.0051$).

The split and merge approach was applied on real data : the Stanford Bunny (figure 11a) and a duck from the MPEG 7 3D objects database (figure 11c). The obtained results let out a limit of the method. It works perfectly as long as the parts of the object are shaped like superquadrics. When trying to segment objects with not so well defined boundaries, it gives only a rough idea of the whole shape (figure 11 ). That may be a problem for precise reconstruction application, but is not so serious for indexation and objects comparison.

To talk about implementation, note that every growing process for each seed can be computed separately in the region growing method. In the same manner, the approximation of every couple of neighbours during the merge procedure is realised independently. Then, parallel computing can be easily used. With our implementation and without any other special optimisations, the whole process takes a few minutes to achieve using standard PC configurations.

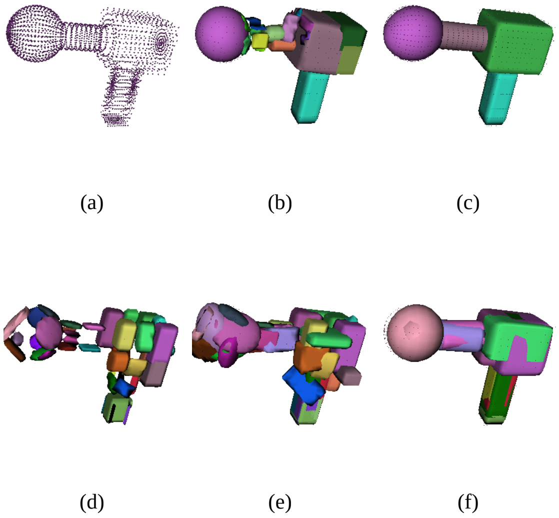


Figure 9: Model B (a) Original data (2433 points) (b) The end of the split step (c) Split and merge result ($D = 0.00005$) (d) Region growing seeds (e) An intermediate step of region growing (f) Region growing result ($D = 0.022$)

# 6 CONCLUSION

A new model for representing an unorganised 3D data points set is presented. This model is a set of superellipsoids whose union leads to an efficient representation of the 3D objects. The basic model (superellip-

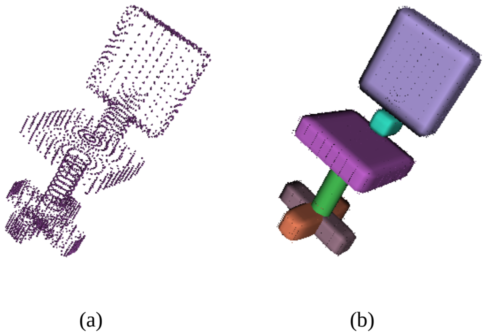


Figure 10: Model C (a) Original data (2996 points) (b) Split and merge result ($D = 0.00039$).

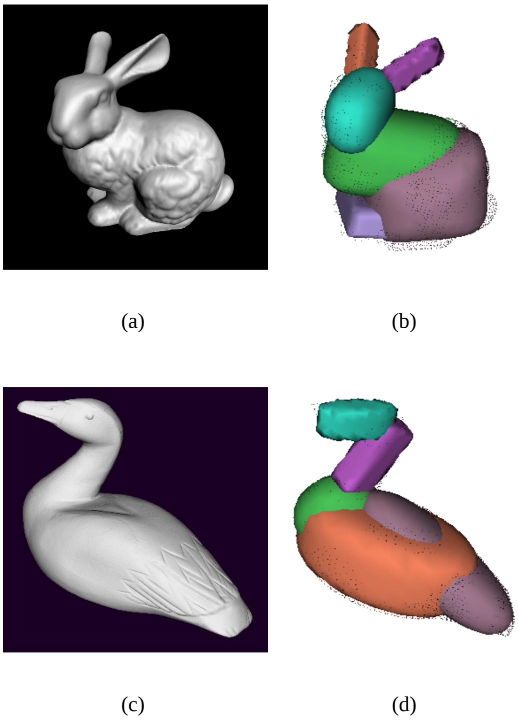

(a) (b) (c) (d)

Figure 11: (a)The Stanford Bunny (8135 points) (b) Split and merge result ($D = 0.019$) (c)The MPEG 7 Duck (3500 points) (d) Split and merge result ($D = 0.016$). .

soid) is simple and compact. Indeed, only 11 parameters are necessary to describe such a model.

Two competing segmentation and modeling methods are developed using the set of superellipsoids. The first one is a region growing approach which has not proved to be very effective. It is indeed very sensitive to the initialisation process and also to the parameters needed for region growing. The second method is a split and merge approach that leads to satisfactory results both in terms of compacity (final number of superellipsoids) and approximation error obtained.

This second method seems promising for the application domains such as 3D object compression and especially 3D object indexing and retrieval. The split and merge algorithm allows to keep the topological relations between the superellipsoids. This leads to a graph. In other words, we can use the graph theory to compare graphs and thus to measure the similarity between 3D objects. This aspect is currently under investigation.

## References

[1] William H. PRESS, Saul A. TEUKOLSKY, William T. VETTERLING, and Brian P. FLANERY. *Numerical Recipes in C.* Cambridge University Press, 1992.

[2] Eric BARDINET, Laurent D. COHEN, and Nicolas AYACHE. A parametric deformable model to fit unstructured 3D data. Research report 2617 - INRIA, 1995.

[3] Alan H. BARR. Superquadrics and angle preserving transformations. *IEEE Computer Graphics and Applications*, 1(1):11–23, January 1981.

[4] Alan H. BARR. Global and local deformations of solid primitives. *Computer Graphics*, 18(3):21–30, 1984.

[5] Laurent CHEVALIER, Fabrice JAILLET, and Atilla BASKURT. 3D shape coding with superquadrics. In *International Conference on Image Processing ICIP'2001*, pages 93–96, Thessaloniki, Greece, October 7-10 2001.

[6] Ales JAKLIC, Ales LEONARDIS, and Franc SOLINA. *Segmentation and Recovery of Superquadrics*, volume 20 of *Computational imaging and vision*. Kluwer, Dordrecht, 2000. ISBN 0-7923-6601-8.

[7] Ales LEONARDIS, Ales JAKLIC, and Franc SOLINA. Superquadrics for segmenting and modeling range data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(11):1289–1295, November 1997.

[8] Ales LEONARDIS, Franc SOLINA, and Alenka MACERL. A direct recovery of superquadric models in range images using recover-and select paradigm. *Proceedings of 3rd European Conference on Computer Vision*, pages 309–318, 1994.

[9] J.A. NEDLER and R. MEAD. A simplex method for function minimization. *Computer Journal*, 7:308–313, 1965.

[10] Alex PENTLAND. Recognition by parts. *IEEE First Int. Conf. on Computer Vision*, pages 612–620, 1987. London.

[11] Franc SOLINA and Ruzena 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.

[12] G. TAUBIN. Estimation of planar curves, surface 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, November 1991.