Published in
the proceedings of the IEEE Workshop on Biomedical Image Analysis (WBIA'94)
June 24–25, 1994, Seattle, Washington

Fitting of Iso-Surfaces Using Superquadrics and Free-Form Deformations

E. Bardinet *

L. D. Cohen **

N. Ayache *

* INRIA - Sophia Antipolis
2004, Route des Lucioles BP 93
06902 Sophia Antipolis CEDEX, France

** Université Paris IX - Dauphine
Place du Marechal de Lattre de Tassigny
75775 Paris CEDEX 16, France

Abstract

Recovery of 3-D data with simple parametric models has been the subject of many studies over the last ten years. Many have used the notion of superquadrics introduced for graphics in [Bar81]. Different improvements were introduced to make the model a better representation of the data [BG87, FLW89, SB90, TM91].

This paper describes a two-steps method to fit a parametric deformable surface to 3-D points. We suppose that a 3-D image has been segmented to get a set of 3-D points. The first step consists in our version of a superquadric fit with global tapering, similar to the method proposed in [BG87]. We then make use of the technique of free-form deformations, as introduced by [SP86] in computer graphics. We present experimental results with synthetic and real 3-D medical images.

1 Introduction

Over the last ten years many surface reconstruction problems have been formulated by the minimization of an energy corresponding to a model of the surface. Using deformable models and templates, the extraction of a shape is obtained through an energy composed of an internal regularization term and an external attraction potential (data term), illustrated for example in [Ter88, TWK88, CC93, YHC93, SB90, TM91]. Since the relevant surfaces in medical images are usually smooth, the use of such models is often very efficient for locating surface boundaries of organs and structures, and then tracking of these shapes in a time sequence.

The advantage of deformable templates like superquadrics is their small number of parameters to represent a shape. However, if superquadric shapes give a good approximation of a shape, they are not sufficient to describe more complex surfaces. Therefore they were coupled with a deformable model in [TM91] to take into account local deformations.

In our work, starting with superquadric models, we present an improvement to refine the shape making use of the technique of free-form deformations, as introduced by [SP86] in computer graphics. Since this deformation is also defined by a small number of parameters, the previous advantage of representing a shape with few parameters remains.

The original contribution of this work is twofold. First we present our algorithm for fitting data with a superquadric, based on [SB90] with some variations.

We then improve the shape extraction by introducing the use of free-form deformations (FFD). The idea is to put our previous surface, here a superquadric, in a rubber-like box. On this box are disposed nodes, called control points, on a regular 3D grid. We then deform the object just by moving these points and deform the whole box using interpolation by trivariate Bernstein polynomials. In graphics and CADs, FFD are used to design shapes by moving a point of the object to some place and retrieving a global deformation of the object. By successive applications of this deformation, a complex shape can be obtained easily. Here, we use this approach for deformation of some points on the superquadric that are to far from data. Since we limit the number of control points to be small, we have to solve an inverse problem [HHK92], that is to recover a box of control points corresponding to a displacement field. This field is computed by comparison between the iso-surface and the superquadric obtained in the first step.

We show results for examples in synthetic images and 3-D medical images of the myocardium.

The paper is organized as follows. We begin by a mathematical description of superquadrics (Section 2.1) and then review fitting methods with superquadrics (Section 2.2) including our algorithm

(Section 2.3). In Section 3, we give the main ideas of free-form deformations followed by illustrations of our method to medical data.

2 Data Fitting Using Superquadrics

This class of objects have been introduced by A. Barr [Bar81] for computer graphics and is an extension to 3-D of the superellipse [Gar65]. Their first use in computer graphics and in vision is due to Pentland [Pen87], followed by Bajcsy [BS87] and later by Terzopoulos and Metaxas [TM91].

There are four kinds of superquadrics but we'll study only the case for superellipsoids. After giving their definition we will see how they can be parameterized and deformed.

2.1 Definition of superquadric surfaces

Superquadrics form a family of implicit surfaces obtained by extension of usual quadrics. They are obtained by spherical product [Bar81] of two 2-D curves. The superellipsoid is the spherical product of the superellipse with itself, we give now his implicit equation followed by a parameterization:

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

$$\begin{cases} x = a \cos^{\epsilon}\theta \ , & -\pi \leq \theta < \pi \\ y = b \sin^{\epsilon}\theta \end{cases}$$

For simplification in the formulation in the above equation and the following, the powers of the cos, sin and tan functions are to take in the following sense:

$$u^{\epsilon} = sign(u) \mid u \mid^{\epsilon} = u \mid u \mid^{\epsilon-1}$$

However in the implicit equation where we have powers $2/\epsilon$, this means

$$u^{2/\epsilon} = (u^2)^{1/\epsilon} = \mid u \mid^{2/\epsilon},$$

the two coefficients control the squareness of curves:

- ϵ is small, the shape is close to a square;
- $\epsilon = 1$ we have a circle;
- $\epsilon = 2$ the shape has flat bevel;
- ϵ is larger the shape is pinched.

We now give the equations and parameterization of superellipsoids. The implicit equation is:

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

Parameterization as a spherical product of two superellipses:

$$\boldsymbol{x}(\eta,\omega) = \begin{bmatrix} \cos^{\epsilon_1}\eta \\ a_3 \sin^{\epsilon_1}\eta \end{bmatrix} \otimes \begin{bmatrix} a_1 \cos^{\epsilon_2}\omega \\ a_2 \sin^{\epsilon_2}\omega \end{bmatrix}$$
$$= \begin{bmatrix} a_1 \cos^{\epsilon_1}\eta \cos^{\epsilon_2}\omega \\ a_2 \cos^{\epsilon_1}\eta \sin^{\epsilon_2}\omega \\ a_3 \sin^{\epsilon_1}\eta \end{bmatrix}, \begin{cases} -\frac{\pi}{2} \leq \eta \leq \frac{\pi}{2} \\ -\pi \leq \omega < \pi \end{cases} \quad (2)$$

2.1.1 Remarks

1. Parameters η and ω correspond to the latitude and longitude of vector $\boldsymbol{x}$ in spherical components; parameters a_1, a_2 and a_3 define the size of the superquadric along x, y and z axes; ϵ_1 and ϵ_2 define the shape of the superquadric along latitude and longitude.
2. The implicit equation of the superellipsoid permits to define an inside-outside function F:

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

The values of F define 3 regions of the space:

$$\begin{cases} \text{if } F(x,y,z) = 1, & (x,y,z) \text{ is on the surface,} \\ \text{if } F(x,y,z) > 1, & (x,y,z) \text{ is outside,} \\ \text{if } F(x,y,z) < 1, & (x,y,z) \text{ is inside.} \end{cases}$$

2.1.2 Regular Parameterization

The parameterizations of the previous sections supply more points in large curvature areas (see figure 2). To fit a 3-D object, we may wish a more regular and uniform parameterization. We now give such a parameterization.

Let us consider the parameterization of a sphere in spherical coordinates:

$$\begin{cases} x = \cos\eta\cos\omega & -\frac{\pi}{2} \leq \eta \leq \frac{\pi}{2} \\ y = \cos\eta\sin\omega & -\pi \leq \omega < \pi \\ z = \sin\eta \end{cases}$$

By a constant step discretization grid of $[-\frac{\pi}{2}, \frac{\pi}{2}] \times [-\pi, \pi]$, the points on the sphere are regularly positioned. We then use this set of points on the sphere to define a new discretization of the superellipsoid by projection. This is done in two steps:

- From the sphere to the ellipsoid:

$$x_e = a_1\, x \quad ; \quad y_e = a_2\, y \quad ; \quad z_e = a_3\, z$$

- From the ellipsoid to the superellipsoid:

$$x_s = \rho\, x_e \quad ; \quad y_s = \rho\, y_e \quad ; \quad z_s = \rho\, z_e$$

Replacing x, y and z by x_s, y_s and z_s , in equation 1, we obtain an expression of ρ as a function of η and ω:

$$\rho = \left[\left(\mid \cos\omega\, \cos\eta \mid^{\frac{2}{\epsilon_2}} + \mid \sin\omega\, \cos\eta \mid^{\frac{2}{\epsilon_2}} \right)^{\frac{\epsilon_2}{\epsilon_1}} + \mid \sin\eta \mid^{\frac{2}{\epsilon_1}} \right]^{-\frac{\epsilon_1}{2}}$$

We obtain our parameterization of the superellipsoid by calculation of ρ and then x_s, y_s and z_s for each value of the grid on the domain $[-\frac{\pi}{2}\,,\frac{\pi}{2}]\times[-\pi\,,\pi]$. The points are then regularly spaced on the surface (see figure 2). Remark that computation is at more cost than the previous parameterization (2): 2 additions, 3 multiplications and 1 power for each point. However we now consider this parameterization of superellipsoids, to have a more regular grid on the surface.

2.1.3 Frame change

Equations of the previous section are defined in a local frame with the center of the superellipsoid as origin. We will need the formulation in a general frame R_O in the following. We define a homogeneous rigid transform T by its matrix:

$$T = \begin{bmatrix} R & \boldsymbol{t} \\ 0 & 1 \end{bmatrix},$$

where R is a 3×3 rotation matrix and $\boldsymbol{t}$ a 3×1 translation vector. We note $\boldsymbol{x}$ and $\underline{\boldsymbol{x}}$ the position vectors respectively in the object frame and in frame R_O. We have then:

$$\begin{aligned} \underline{\boldsymbol{x}} &= R\,\boldsymbol{x} + \boldsymbol{t}, \\ \boldsymbol{x} &= R^t\,(\underline{\boldsymbol{x}} - \boldsymbol{t}). \end{aligned}$$

since the inverse R^{-1} is equal to R^t.

We choose the 3 Euler angles φ, θ and ψ to represent a rotation in $\mathbb{R}^3$. The new function of position is obtained by replacing transform T in 1:

$$\widehat{F}\,(\underline{\boldsymbol{x}}) = F(T^{-1}\,(\underline{\boldsymbol{x}})). \tag{3}$$

Function $\widehat{F}$, which will be used in the minimization algorithms depends on 11 parameters: $(a_1, a_2, a_3, \epsilon_1, \epsilon_2, \varphi, \theta, \psi, t_x, t_y, t_z)$.

2.2 Previous work

2.2.1 Pentland's Approach

The use of superquadrics for analysis of scenes in computer vision was first introduced by Pentland [Pen87]. He proposed a heuristic approach by search over the whole parameter space for the best value of the goodness-of-fit criteria. Since this approach is computationally too expensive, later authors preferred iterative algorithms.

2.2.2 Solina and Bajcsy's Model

F. Solina and R. Bajcsy [SB90] have used superellipsoids for approximation of 3-D objects. They used the parameterization defined in equation 2. They minimized a least square energy defined by the inside-outside function. The data is a set of points which correspond to the description of a 3-D object.

Here are the main steps of their approach:

- Initialization of the surface:
  The initial superquadric is an ellipsoid (parameters ϵ_1 and ϵ_2 equal to 1). It is centered at the center of gravity of the data set. Its orientation is defined by the moments of inertia of the data. This define the new frame of the object. The initial size of the three axes is also roughly estimated from data.

- Least squares minimization:
  Since we want the data to position on the superquadric surface and this surface is defined by its inside-outside function $\widehat{F} = 1$ (see equstion 3), the criteria used for minimization with respect to the 11 parameters of $\widehat{F}$ is:

  $$E(A) = \sum_{i=1}^{N} \left[1 - \widehat{F}(a_1, a_2, a_3, \epsilon_1, \epsilon_2, \varphi, \theta, \psi, \boldsymbol{t}) \right]^2. \tag{4}$$

  Since derivatives of E can be computed, the minimization is done using the Levenberg-Marquardt [PFTV89] method for non-linear least squares. To overcome some problem due to self occlusion, they modified the criteria to be:

  $$\widehat{E}(A) = \sum_{i=1}^{N} \left[\sqrt{a_1\,a_2\,a_3} \left(1 - \widehat{F}(a_i, \epsilon_j, \varphi, \theta, \psi, \boldsymbol{t}) \right) \right]^2.$$

  This adds that among the possible minimizing surface we choose the one with smaller volume.

- Global deformations:
  A. Barr [Bar84] studied the deformation of superquadrics. F. Solina and R. Bajcsy introduced some global deformations (pinching, torsion, modelization of cavity) in their model to extend the variety of possible shapes. These deformations are defined by only 4 parameters which are added to the previous set of parameters. Function $\widehat{F}$ depends then on 15 parameters. These deformations has to be composed in a precise order.

2.2.3 Terzopoulos and Metaxas' Model

Terzopoulos and Metaxas [TM91] add to the previous global deformation the possibility of local deformation of the superquadric model. The local deformation is done on a given parametric representation of the superquadric and satisfies the minimization of a physically-based deformable model. They minimize a local potential at the same time with respect to global parameters of superquadrics and local deformation. These deformable superquadrics are no more simple parametric models but are in fact deformable models with some kind of memory of shape.

Since we wish to keep a small number of parameters we will not make use of this kind of ideas and that is why we give only a short description.

Instead we will introduce (in section 3) the use of Free Form Deformations [SP86] used in Computer Graphics.

2.3 Description of our Method

In our applications, the original data is a 3-D medical image which represents for example the myocard or the head area. Interesting features can be either edges extracted from this data which form a set of surfaces, or an iso-surface. We want to approximate one of these surfaces by a superellipsoid.

To fit a superquadric on a set of data points, we use the algorithm developed in [SB90], with some technical improvements. Before we detail these improvements, we will focus on the notion of distance between a point and a surface (or a set of data points).

2.3.1 Distance between a point and a surface

The exact computation of the Euclidean distance at each point is very expensive and usually an approximation is used computed by use of iterative filters [Bor84, Dan80]. We now give two analytic expressions of an approximation of the distance between a point and an implicit surface.

- Intrinsic approximation:
  Let us consider a superellipsoid in its local intrinsic frame (equations 2, 1). Let $M = (x, y, z)$ be a data point. The line joining the center O of the local frame and M intersects the superellipsoid surface at $M_0 = (x_{M_0}, y_{M_0}, z_{M_0})$ (see Figure 3).

  We want to calculate the distance $\| M_0M \|$. Since O, M_0 and M are on a same line, we can write $\overrightarrow{OM_0} = \mu \overrightarrow{OM}$ with $\mu > 0$. So we have $(x_{M_0}, y_{M_0}, z_{M_0}) = (\mu x, \mu y, \mu z)$. Since we have: $F(\mu x, \mu y, \mu z) = \mu\, F(x, y, z)$, with $F(M_0) = 1$, we obtain the value of μ:

  $$\mu = \frac{1}{F(x,y,z)}.$$

  We then deduce the following:

  $$\begin{cases} \|M_0M\| &= |1-\mu|\ \|OM\| \\ &= \left|\dfrac{F(x,y,z)-1}{F(x,y,z)}\right| \|OM\| \\ \|M_0M\| &= |\frac{1-\mu}{\mu}|\ \|OM_0\| \\ &= |F(x,y,z)-1|\ \|OM_0\| \end{cases}$$

  And:

  $$(F-1)^2 = \left(\frac{\| M_0M \|}{\| OM_0 \|}\right)^2 = \left(\frac{\| OM \|}{\| OM_0 \|} - 1\right)^2.$$

  This gives a geometric interpretation of the criteria used in [SB90] as a intrinsic homothetic characteristic, but it is not a distance. On figure 3, we easily see that $\| M_0M \|$ is greater that the Euclidean distance: $d = \| MP \|$.

- First order approximation:
  We also have an approximation of d when the data point M is close to its projection on the superquadric P.

  $$d \approx \frac{| F(M) - 1 |}{\| \nabla F(M) \|} \tag{5}$$

  It is possible to define a more precise approximation that is bounded as shows Taubin [Tau93], but it is computationally expensive.

2.3.2 Solina and Bajcsy's Model revisited

As the initialization of the superquadric, we still use the matrix of second order centered moments M to evaluate the orientation (see section 2.2.2). But to compute the size of the three axes, we compare the matrix of an ellipsoid surface J and M after diagonalization. The form of J is:

$$J = \frac{\mu}{3}\begin{bmatrix} b^2+c^2 & 0 & 0 \\ 0 & c^2+a^2 & 0 \\ 0 & 0 & a^2+b^2 \end{bmatrix}, \tag{6}$$

where a, b and c are the sizes of the axes and μ is the mass (here 1). Hence, if λ_1, λ_2 and λ_3 are the eigen values of M, the sizes of the axes are computed as follow:

$$\begin{cases} a^2 &= \frac{3}{2\mu}(\lambda_2+\lambda_3-\lambda_1), \\ b^2 &= \frac{3}{2\mu}(\lambda_1+\lambda_3-\lambda_2), \\ c^2 &= \frac{3}{2\mu}(\lambda_1+\lambda_2-\lambda_3); \end{cases} \tag{7}$$

The criteria used for minimization is the same as equation 4, but minimization is done using the Conjugate Gradient Method in Multidimensions [PFTV89], which is an astute algorithm to find the minimum of a function only with the computation of first derivates. This method ensures a fast convergence. We also take into account one type of global deformation: a tapering among the three axes. This can be expressed in a matrix form:

$$\boldsymbol{X_{tap}} \ = \ \boldsymbol{T}.\boldsymbol{X},$$

with:

$$T = \begin{pmatrix} f(y)f(z) & 0 & 0 \\ 0 & f(x)f(z) & 0 \\ 0 & 0 & f(x)f(y) \end{pmatrix}.$$

Figure 5 represents the result of this fit to a myocardium.

3 Free-Form Deformations to Improve the Fit

We now have a first approximation of the isosurface. The superquadric is correctly oriented and has already the good size for the three axes of inertia. The problem is that the range of shapes described by superquadrics is really too limited to describe complex medical objects, like myocardium or brain. We need to make the model more complex. This idea is not new, it has raised different solutions, like adding a spline energy or the decomposition of the displacement in a wavelets basis [TM91, VR92].

We decided to use a tool developed in computer graphics [SP86, Coq90] called Free-Form Deformations (FFD). Our choice was guided by the fact that we wanted to dispose of a simple global model at the end of the process. In [TM91] for example, the model is made of global and local information.

In this section, we first present the FFD as introduced by [SP86], then we explain how we use this tool in the context of computer vision. Our method is based on a work presented in [HHK92].

3.1 Definition

The FFD is a tool devoted to the deformation of solid geometric models in a free-form manner. It is independent of the nature of the object to be deformed. An analogy for FFD is to consider a box in which the object is captured. On each face of the box are disposed points on a regular grid. To deform the object, you just need to move the points, called control points, on the box, and the object follows in an intuitively way the deformation of the box.

The algorithm is based on trivariate Bernstein polynomials, and divided in two steps:

1. Computation of the local coordinates of the object points in the system defined by the box of control points.

2. Displacement of the control points and estimation of the new position of the object.

Let's explain in detail those two steps:

1. First we need a box in which the object is embedded. We define a local coordinate system by choosing a vertex on the box and his three edges. Any point $\boldsymbol{X}$ has (s, t, u) coordinates in this system such that:

   $$\boldsymbol{X} \ = \ \boldsymbol{X_0} + s\,\boldsymbol{S} + t\,\boldsymbol{T} + u\,\boldsymbol{U},$$

   where $\boldsymbol{X_0}$ is the vertex and s, t, u are given by:

   $$s \ = \ \frac{\boldsymbol{T} \times \boldsymbol{U}(\boldsymbol{X} - \boldsymbol{X_0})}{\boldsymbol{T} \times \boldsymbol{U} \cdot \boldsymbol{S}} \ , \ \ t \ = \ \frac{\boldsymbol{S} \times \boldsymbol{U}(\boldsymbol{X} - \boldsymbol{X_0})}{\boldsymbol{S} \times \boldsymbol{U} \cdot \boldsymbol{T}} \ ,$$

   $$u \ = \ \frac{\boldsymbol{S} \times \boldsymbol{T}(\boldsymbol{X} - \boldsymbol{X_0})}{\boldsymbol{S} \times \boldsymbol{T} \cdot \boldsymbol{U}},$$

   with $\boldsymbol{S}$, $\boldsymbol{T}$ and $\boldsymbol{U}$ be the three other vertex joined to $\boldsymbol{X_0}$.

   We also generate a grid of $(l + 1)(m + 1)(n + 1)$ control points on the box by setting:

   $$\boldsymbol{P_{ijk}} \ = \ \boldsymbol{X_0} + \frac{i}{l}\,\boldsymbol{S} + \frac{j}{m}\,\boldsymbol{T} + \frac{k}{n}\,\boldsymbol{U}.$$

2. The deformation of the object is specified by moving the control points from their latticial position and evaluating the new position for any point $\boldsymbol{X}$ with the vector-valued trivariate Bernstein polynomial:

   $$\boldsymbol{X_{new}} \ = \ \sum_{i=0}^{l}\sum_{j=0}^{m}\sum_{k=0}^{n} \big(C_l^i C_m^j C_n^k \quad (8)$$
   $$(1-s)^{l-i} s^i (1-t)^{m-j} t^j (1-u)^{n-k} u^k \boldsymbol{P_{ijk}}\big) \, .$$

   $\boldsymbol{X_{new}}$ is the vector of the displaced point expressed in Cartesian coordinates.

An important characteristic of FFD is that a parametric surface is still parametric after deformation, simply obtained by composition. For more details about the definition of FFD, see [SP86]. As we said, FFD is a tool developed for computer graphics. In the next section, we show how to use it to improve the result of the fit of complex 3D shapes with superquadrics.

3.2 The Inverse Problem

The algorithm presented in the previous section can be summed up like this: displacement of the control points permits to compute a displacement map for any space point. Here, the problem is reversed. What we have are the data points and the superquadric resulting from the fit presented in section 2. To improve the precision of the fit, the superquadric is embedded in a parallelepiped box and a displacement field between the data and the model points is computed. Then we want to find the displacement of the control points that corresponds to this field, that is to resolve the inverse problem. This question has been tackled in [HHK92], still in computer graphics, to get the FFD easier to manipulate. Contrary to them, we will still use Bernstein polynomials because we are looking for a global model with few parameters, that means in concrete terms that the size of the box will ever be less than or equal to 5x5x5.

3.2.1 Computation of the displacement field

We need for each point on the superquadric a vector whose norm is the distance to the data. If we consider the figure 6 and suppose the isosurface above corresponds to the superellipsoid and the other to the data, it is easy to see that we want to find the vector $\boldsymbol{MP_1}$ which is orthogonal to the superellipsoid, instead of $\boldsymbol{MP}$, thus avoiding the case: data points never reached.

The computation breaks up in 2 parts:

1. The distance map of the superellipsoid is computed using the algorithm of Danielsson [Dan80]. Then, for each data point, we find his closest point on the superellipsoid.

2. After this step, some points on the ellipsoid haven't been reached. A displacement value is attributed by interpolation; that means, for each non-valued point, we search his 4 valued neighbors on the superellipsoid, and compute the average of the displacement values.

3.2.2 Displacement of the control points

We want to find the new position of control points that recover the displacement field we obtained in the previous section. That means we have to resolve equation 9, where the unknown are $\boldsymbol{P_{ijk}}$. This equation can be written in a matrix form:

$$\boldsymbol{X} = \boldsymbol{B}\boldsymbol{P}, \tag{9}$$

where $\boldsymbol{B}$ is $NP \times ND$ (NP: number of control points, ND: number of displaced points from the superellipsoid). Actually equation 9 is separable in three independent linear systems according to points coordinates. We will now consider the case for the first coordinate x, that means $\boldsymbol{X}$ and $\boldsymbol{P}$ will represent the x-coordinate of the considered points.

In our case, because we are looking for a global model with relatively few parameters, NP is always less than ND, that means the problem is overdetermined and there is no solution. However one can find the best "compromise" solution in the least squares sense, using different algorithms. We will present results for two of them [PFTV89]:

- Direct solution of the normal equations associated to the linear least-squares problem.
- Direct solution of the linear least-squares problem with the Singular Value Decomposition.

Normal equations: The normal equations associated to equation 9 are: $\boldsymbol{B}^T\boldsymbol{X} = (\boldsymbol{B}^T\boldsymbol{B})\,\boldsymbol{P}$. This square linear system is solved by the Conjugate Gradient Method, and the solution can be written:

$$\boldsymbol{P} = (\boldsymbol{B}^T\boldsymbol{B})^{-1}\,\boldsymbol{B}^T\,\boldsymbol{X}. \tag{10}$$

Singular Value Decomposition: SVD is based on the following theorem of linear algebra: any $M \times N$ matrix $\boldsymbol{B}$ whose number of rows M is greater than or equal to its number of columns N, can be written as the product of an $M \times N$ column-orthogonal matrix $\boldsymbol{U}$, an $N \times N$ diagonal matrix $\boldsymbol{W}$ with positive or zero elements, and the transpose of an $N \times N$ orthogonal matrix $\boldsymbol{V}$. This can be written as: $\boldsymbol{B} = \boldsymbol{U}.\boldsymbol{W}.\boldsymbol{V}$. It follows immediately that the inverse of $\boldsymbol{B}$ is:

$$\boldsymbol{B}^{-1} = \boldsymbol{V}.[diag(1/w_j)].\boldsymbol{U}^T.$$

A problem can appear if one of the w_j's is zero. But, actually, SVD explicitly constructs orthonormal bases for the nullspace and range of a matrix, thus it's enough to replace $1/w_j$ by zero if $w_j = 0$ to pick the solution with the smallest length [PFTV89]. Application of this theorem to equation 9 induces to this solution for $\boldsymbol{P}$:

$$\boldsymbol{P} = \boldsymbol{V}.[diag(1/w_j)].\boldsymbol{U}^T.\boldsymbol{X}. \tag{11}$$

4 Results

We present in this section some results obtained by application of the two-steps algorithm described in the previous sections. The original isosurfaces are extracted either from synthetic or medical data.

Synthetic Data: This example shows clearly that superquadric fit is not sufficient to recover complex shapes. It also demonstrate the powerful capabilities of FFD analysis. Another important fact is that, as we already mentionned, a parametric surface being deformed by FFD remains parametric, even after some iterations, like in this example.

Medical Data: Myocardium We now present three medical cases. All myocardia are extracted from the same time sequence. Figure 10 was obtained with a 4x4x4 box and resolution by Conjugate Gradient Method. The two others results were obtained with a 5x5x5 box and resolution by Singular Value Decomposition. In that case, the quality of fit is much better, however the resulting box is much larger than the one in 10. This comes from the global nature of Bernstein polynomials, and it can be improved by using splines or other piecewise polynomials. Those data are each composed of 6000 points, and the model is defined each time by 11 parameters for the superellipsoid and 5x5x5 3D points for the control points box, that is only less than 130 3D points.

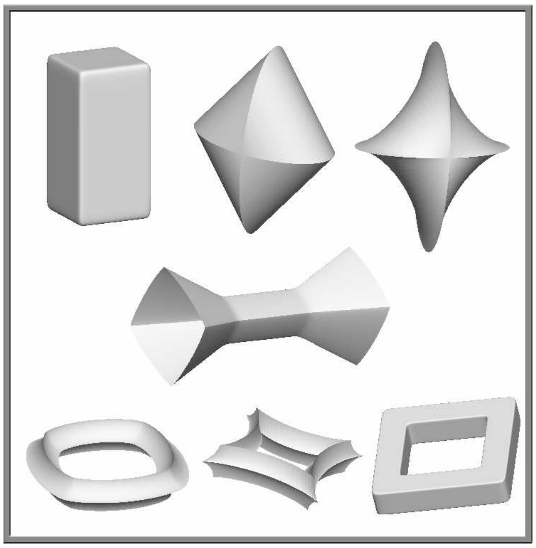

Figure 1: Different types of superquadrics.

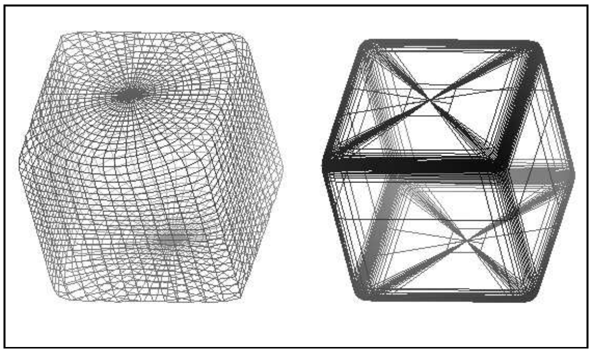

Figure 2: Two types of parameterization: regular and curvature-dependent

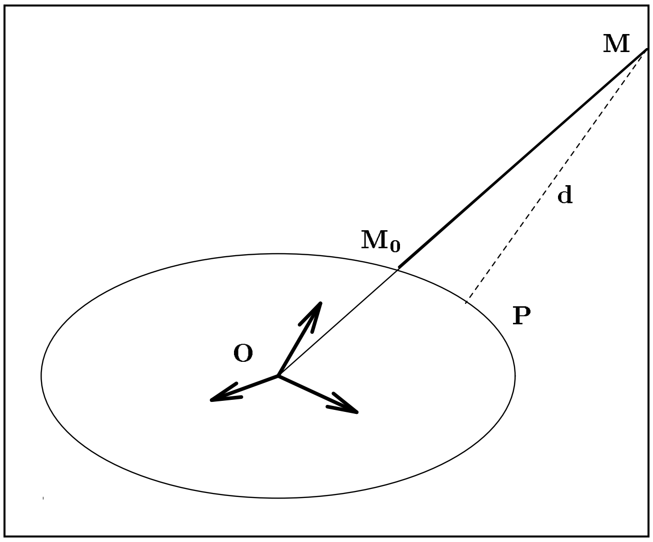

Figure 3: Intersection of line OM and surface

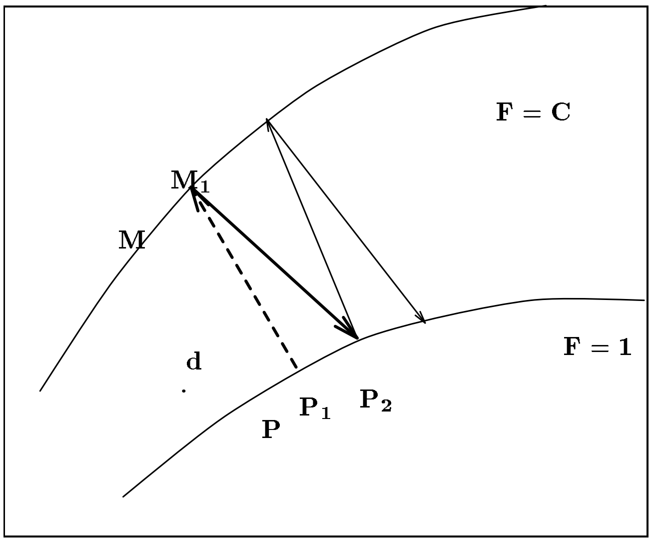

Figure 4: Distances between 2 isosurfaces

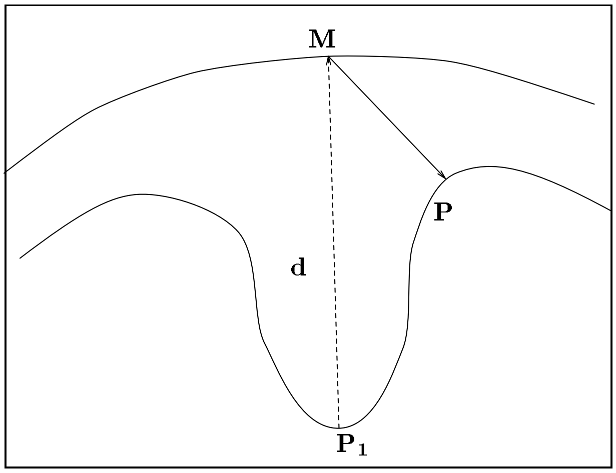

Figure 6: Displacement vectors between the model and the data

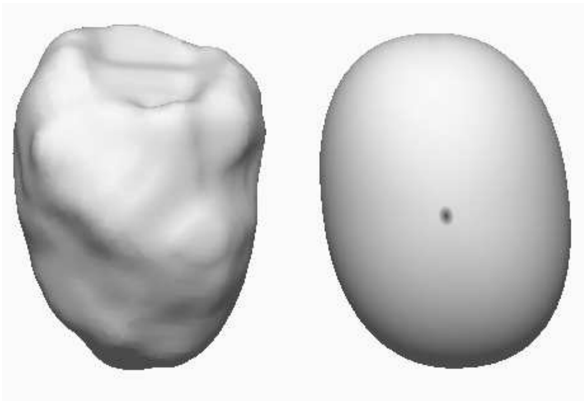

Figure 5: A myocardium fitted by a tapered superquadric.

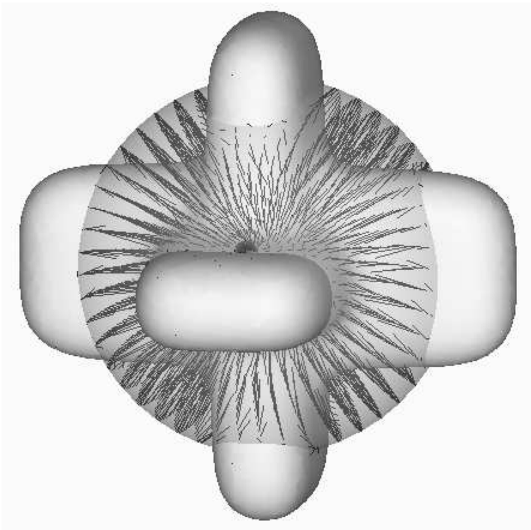

Figure 7: An example of displacement field.

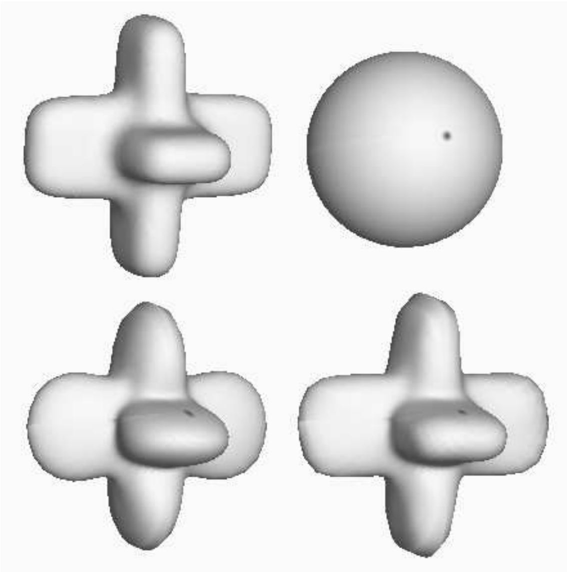

Figure 8: A cross, the initial sphere, and two iterations with a 5x5x5 box.

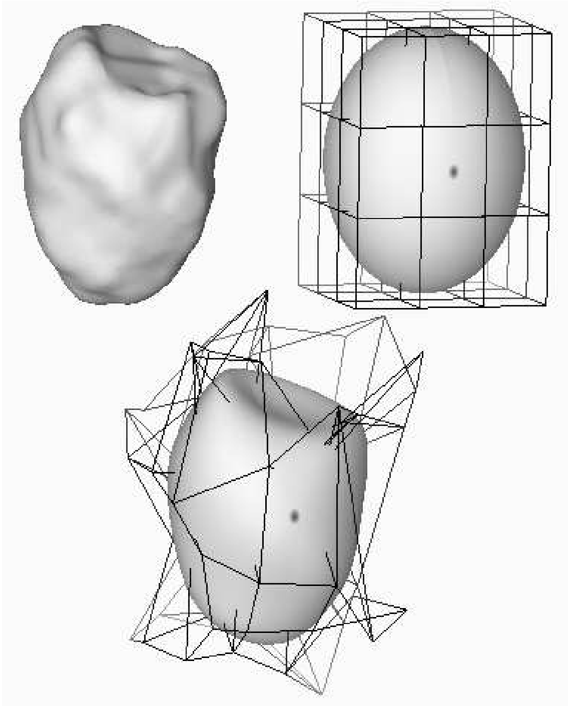

Figure 10: A myocardium, the fitted superellipsoid embedded in the control points box (4x4x4), and the final result.

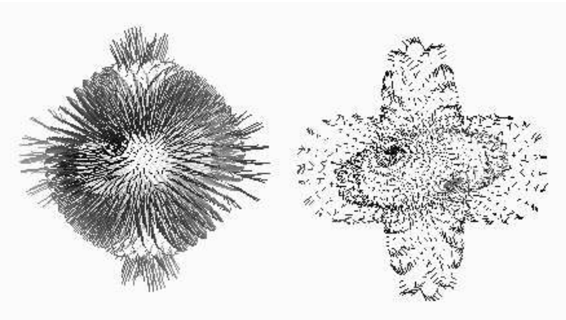

Figure 9: The displacement field before and after FFD analysis .

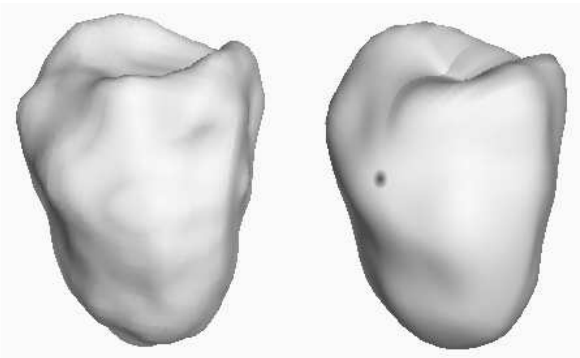

Figure 11: Another myocardium fitted with a 5x5x5 box.

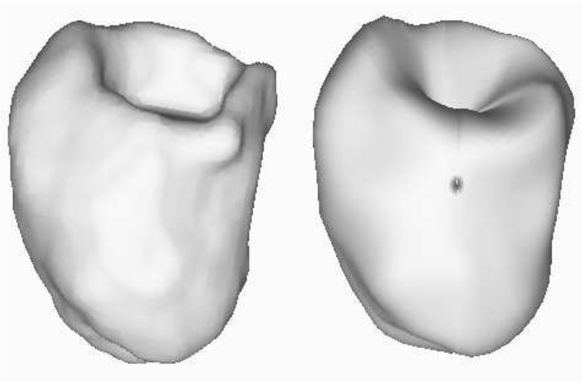

Figure 12: A third one, still fitted with a 5x5x5 box.

5 Conclusion

We have presented a new approach for shape reconstruction applied to 3-D medical data. It is based on a first step giving the best fit with a superquadric model. This is followed by a second step for refining the details of the previous shape by making use of free-form deformations. In both steps the data is an iso-surface obtained from a 3-D image, and an attraction potential defined by the distance to the closest data point is used for energy minimization in the first step and definition of the displacement field in the second step.

The interesting aspect of this approach is that it gives a description of a complex shape with only a small number of parameters.

We gave examples of results for synthetic images and 3-D medical images of the myocardium. Our current goal is to use this approach for automatic shape tracking in a time sequence of images.

Acknowledgments

We would like to thank Grégoire Malandain and Alexis Gourdon who provided us a substantial help at the time of enriching discussions. We thank also Serge Benayoun for his judicious remarks. This work was partially supported by Digital Equipment Corporation.

References

[Bar81] A.H. Barr. Superquadrics and angle-preserving deformations. *IEEE Computer Graphics Applications*, 1:11–23, 1981.

[Bar84] A.H. Barr. Global and local deformations of solid primitives. *ACM Computer Graphics*, 18(3):21–30, 1984.

[BG87] T.E. Boult and A.D. Gross. Recovery of superquadrics from depth information. In *Spatial Reasoning and Multi-Sensor Fusion Workshop*, pages 128–137, Saint Charles, Illinois, October 1987.

[Bor84] G. Borgefors. Distance transformations in arbitrary dimensions. *Computer Vision, Graphics and Image Processing*, 27:321–345, 1984.

[BS87] R. Bajcsy and F. Solina. Three dimensional object representation revisited. In *Proceedings of the Fisrt International Conference oc Computer Vision*, pages 231–240, London, June 1987.

[CC93] L.D. Cohen and I. Cohen. Finite element methods for active contour models and balloons for 2-D and 3-D images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15, 1993.

[Coq90] S. Coquillart. Extended free-form deformation: A sculturing tool for 3d geometric modeling. In *SIGGRAPH'90*, volume 24, pages 187–196, Dallas, 1990.

[Dan80] P.E. Danielsson. Euclidean distance mapping. *Computer Graphics And Image Processing*, 14:227–248, 1980.

[FLW89] F.P. Ferrie, J. Lagarde, and P. Whaite. Darboux frames, snakes and superquadrics: geometry from the bottom-up. In *Proceedings of the IEEE Workshop on Interpretation of 3D Scenes*, pages 170–176, Austin, November 1989.

[Gar65] M. Gardiner. The superellipse: a curve that lies between the ellipse and the rectangle. *Scientific American*, 213:222–234, 1965.

[HHK92] W.M. Hsu, J.F. Hughes, and H. Kaufman. Direct manipulation of free-form deformations. In *SIGGRAPH'92*, volume 26, pages 177–184, Chicago, 1992.

[Pen87] A.P. Pentland. Recognition by parts. In *IEEE Proceedings of the first International Conference on Computer Vision*, pages 612–620, 1987.

[PFTV89] W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling. *Numerical recipies in C, the art of scientific computing.* Cambridge University Press, 1989.

[SB90] 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*, 12:131–147, 1990.

[SP86] T.W. Sederberg and S.R. Parry. Free-form deformation of solid geometric models. In *SIGGRAPH'86*, volume 20, pages 151–160, Dallas, 1986.

[Tau93] G. Taubin. An improved algorithm for algebraic curve and surface fitting. In *International Conference on Computer Vision (ICCV)*, pages 658–665, Berlin, 1993.

[Ter88] Demetri Terzopoulos. The computation of visible-surface representations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-10(4):417–438, July 1988.

[TM91] D. Terzopoulos and D. Metaxas. Dynamic 3d models with local and global deformations: deformable superquadrics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(7):703–714, 1991.

[TWK88] Demetri Terzopoulos, Andrew Witkin, and Michael Kass. Constraints on deformable models: recovering 3D shape and nonrigid motion. *AI Journal*, 36:91–123, 1988.

[VR92] B.C. Vemuri and A. Radisavljevic. From global to local, a continuum of shape models with fractal priors. Technical Report TR-92-036, Computer and Information Sciences Department - University of Florida, November 1992.

[YHC93] A.L. Yuille, P.W. Hallinan, and D.S. Cohen. Feature extraction from faces using deformable templates. *IJCV*, 8(3), September 1993.