INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

A parametric deformable model to fit unstructured 3D data

Eric Bardinet , Laurent D. Cohen , Nicholas Ayache

N˙ 2617

Juillet 1995

_____ PROGRAMME 4 _____

*R**apport de recherche*

A parametric deformable model to fit unstructured 3D data

Eric Bardinet *, Laurent D. Cohen **, Nicholas Ayache

Programme 4 — Robotique, image et vision
Projet EPIDAURE

Abstract: Recovery of unstructured 3D data with deformable models has been the subject of many studies over the last ten years. In particular, in medical image understanding, deformable models are useful to get a precise representation of anatomical structures.

However, general deformable models involve large linear systems to solve when dealing with high resolution 3D images. The advantage of parametric deformable models like superquadrics is their small number of parameters to describe a shape combined with a better robustness in the presence of noise or sparse data. Also, at the expense of a reasonable number of additional parameters, free form deformations provide a much closer fit and a volumetric deformation field.

This article introduces such a model to fit unstructured 3D points with a parametric deformable surface based on a superquadric fit followed by a free form deformation to describe the cardiac left ventricle. We present the mathematical and algorithmic details of the method, as well as three experimental studies in a various number of 3D medical images. The extension of the method to track anatomical structures in spatio-temporal images (4D data) will be presented in a companion article [9].

Key-words: Parametric models, Deformable models, Superquadrics, 3D Splines, Free-form deformations, Regularization, Surface Reconstruction.

(Résumé : tsvp)

*Eric.Bardinet@epidaure.inria.fr
**CEREMADE, U.R.A. CNRS 749, Université Paris IX - Dauphine, Place du Marechal de Lattre de Tassigny, 75775 Paris CEDEX, France.

Un modèle paramétrique déformable pour représenter des données 3D non structurées

Résumé : Représenter des données 3D non structurées par des modèles déformables a été le sujet de nombreux travaux lors de ces dix dernières années. En particulier, pour l'interprétation d'images médicales, les modèles déformables sont utiles pour disposer d'une représentation précise des structures anatomiques.

Néanmoins, les modèles déformables généraux nécessitent la résolution de grands systèmes linéaires lorsque l'on traite des images 3D de fine résolution. L'avantage des modèles paramétriques déformables comme les superquadriques est le petit nombre de paramètres pour décrire une forme ainsi qu'une meilleure robustesse en présence de bruit ou de données éparses. De plus, en ajoutant un nombre raisonnable de paramètres, les déformations de forme libre permettent d'obtenir une représentation plus précise et un champ de déformation volumique.

Cet article introduit un tel modèle pour représenter des points 3D non structurés avec une surface paramétrique déformable basée sur l'ajustement d'une superquadrique suivi par une déformation de forme libre pour décrire le ventricule gauche cardiaque. Nous présentons les détails mathématiques et algorithmiques de la méthode, ainsi que trois études expérimentales sur des images médicales de types divers. L'extension de la méthode pour suivre des structures anatomiques dans des images spatio-temporelles (données 4D) sera présenté dans un prochain article [9].

Mots-clé : Modèles paramétriques, modèles déformables, superquadriques, splines 3D, déformations de forme libre, régularisation, reconstruction de surfaces.

1 Introduction

In many computer vision and image understanding problems, it is important to fit a deformable model to unstructured 3D data. In particular, in the large domain of medical image understanding, one often needs to solve fitting problems, in order to make surface reconstruction and get a precise compact representation of anatomical structures. In this article, we introduce a parametric deformable model to fit unstructured 3D data. Although our model may be used for surface fitting and reconstruction problems involving other types of images, we are mainly concerned with applications for medical images, in particular for the heart analysis.

The extraction and representation of the left ventricular wall is an important topic, because cardiovascular diseases are still the first cause of mortality in developed countries. Medical image analysis can help the diagnosis of such diseases by assessing a number of quantitative and objective numerical parameters describing the behavior of the cardiac system. Several imaging techniques produce temporal series of three-dimensional description of the heart, i.e. 4D (3D+T) images. Among them, gated MRI, ultrasound images and nuclear medicine images; also, the DSR (Dynamic Spatial Reconstructor), a fast CT-scanner developped at the Mayo Clinic, used to produce such 4D images.

Because it is characteristic of the good health of the heart, the left ventricle motion and deformation has been extensively studied by medical image processing groups as well as hospitals. Since its creation in 1989, our group has pioneered work in the use of deformable models to extract the left ventricle [4, 21, 5, 20, 23, 6, 3, 44, 13]. Other groups as well have also made various contributions to understanding the complex deformation of the ventricle [2, 43, 1, 42, 18, 51].

Over the last ten years, many fitting problems have been formulated as the minimization of an energy functional corresponding to a model of the surface [32, 14, 38, 57, 23, 40, 27]. Although previous approaches based on general deformable surfaces give satisfactory results, they involve large linear systems to solve, in particular when dealing with high resolution 3D images. This is why we investigate in this article the introduction of adequate parametric models to describe surfaces in 3D images.

The advantage of parametric deformable templates, like superquadrics or hy-

perquadrics [60, 53, 35], is their small number of parameters to represent a shape. However, if superquadric shapes give a good global approximation to a surface, the set of shapes described by superquadrics is too limited to approximate precisely complex surfaces. In [58, 19], different approaches are used to refine the initial approximation.

In this article, we introduce a deformable superquadric model based on a superellipsoidal fit followed by a free form deformation (FFD) to describe the left ventricle. We therefore use the a priori information that the shape of the left ventricle ressembles an ellipsoid deformed by a smooth transformation, and limit strongly the number of parameters necessary for this representation. We also increase the robustness of the fit in the presence of noise or sparse data points. Other approaches for the segmentation of the left ventricle using parametric deformable models include [58, 59, 45].

After a description of the parametric model in Section 2, we explicit the details of our fitting algorithm, both for the superellipsoid model and for the FFD computation (Section 3). We then present two ways of controlling explicitly the regularity of the model (Section 4). Finally we show in Section 5 three experimental studies which demonstrate the effectiveness of this approach to capture the 3D shape of the left ventricle in a various number of 3D images, with a reduced number of parameters. The temporal tracking of these parameters will be described in a companion article [9].

2 Fitting 3D data with a superquadric

Superquadrics have been defined by A. Barr [11]. They are an extension of the quadrics family, based on the already old notion of superellipse [30]. In the field of computer graphics and CAD/CAM, this new family of parametric shapes has been widely used. In computer vision, their first use is due to A. Pentland [47], followed by many other authors [7, 16, 52, 28, 29, 48, 53, 33, 41, 34].

As for ordinary quadrics, there are four kinds of superquadrics (provided that one considers torus as a special quadric): superellipsoid, superhyperboloid with one or two sheets, supertorus. In the following, we will only consider the case of superellipsoids. Indeed, the family of surfaces described by superellipsoids is the most adapted for the representation of the anatomical structures in which

we are interested, and equations are very similar for the three other kinds of superquadrics.

We first recall the definition of superellipsoids, then we introduce a new parameterization which gives a uniform distribution of points on the model surface. We will then describe the method to fit 3D data with these surfaces and give a geomtric interpretation of the criterion of fit.

Additional details are presented in Appendix A.

2.1   Definition of superquadrics

Superquadrics form a family of implicit surfaces obtained by extension of usual quadrics. Superellipsoids are defined by the implicit equation:

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

The natural parameterization, coming from the spherical one, can be written:

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

The implicit equation of the superellipsoid permits to define an inside-outside function F:

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

Superquadrics are real mathematical solids. Therefore, the values of F define 3 distinct regions of the 3D space:

$$\begin{cases} \text{if } F(x, y, z) & = & 1 \,, \ (x, y, z) \text{ is on the surface,} \\ \text{if } F(x, y, z) & > & 1 \,, \ (x, y, z) \text{ is outside,} \\ \text{if } F(x, y, z) & < & 1 \,, \ (x, y, z) \text{ is inside.} \end{cases}$$
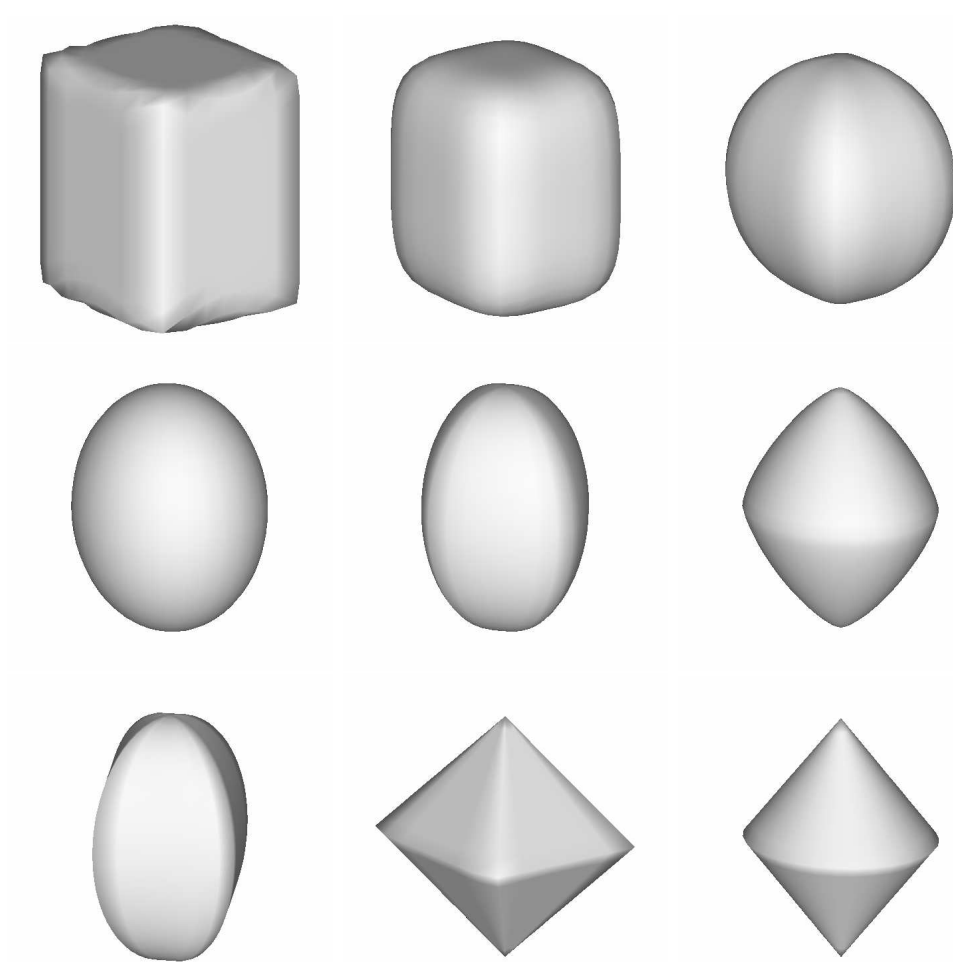
Figure 1: Examples of superellipsoids: a wide variety of shapes.

2.2   Regular parameterization

The previous parameterization (Eq. (2)) has the advantage, for CAD/CAM applications, to supply more points in large curvature areas (see Figure 2). But, to fit a set of 3D points with a superellipsoid, it is better to have a more uniform distribution of the points on the surface. Vemuri and al. [59] presented a solution to this problem by the resolution of a differential equation. We propose a less expensive solution.

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

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

By a constant step discretization grid of $[-\frac{\pi}{2}, \frac{\pi}{2}] \times [-\pi, \pi]$, the points on the sphere are regularly positioned. It is then enough to project this set of points from the sphere to the superellipsoid to get a more uniform distribution of the points on it.

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 Eq. 1, we obtain an expression of ρ as a function of η and ω:

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

We then obtain a new parameterization of the superellipsoid by the computation of ρ, 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 the computation is more costly than for the natural parameterization: 2 additions, 3 multiplications and 1 power for each point. However we now consider this parameterization of superellipsoids, to get a more uniform distribution of the points on the surface.
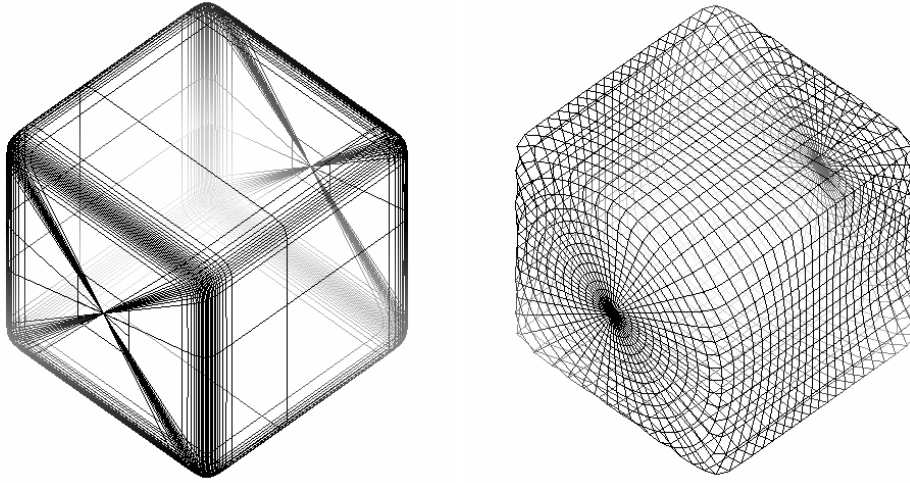
8

Figure 2: Left: curvature-dependent parameterization. Right: uniform parameterization.

2.3 Fit

The use of superquadrics for the analysis of scenes in computer vision was first introduced by Pentland [47]. He proposed a heuristic approach based on a search over the whole parameter space for the best value of the goodness-of-fit criteria. Since this approach is computationally very expensive, later authors prefered iterative algorithms, in particular, Solina and Bajcsy have used superellipsoids for the approximation of 3-D objects [7, 52, 53]. Our fitting algorithm is inspired by their work.

Least squares minimization:

Suppose that the data that we want to fit with the superellipsoid are a set of 3D points $(x_d, y_d, z_d), i = 1, \ldots, N$. The goal of the algorithm is to change the eleven parameters to find the values for which most of the 3D points are close to the surface of the model. In the general case, there will not exist a set of parameters for which the model perfectly fits to the data, because the shapes

which we are interested in are more complex than superellipsoids. Therefore, the problem has to be formulated as a least-squares problem. Now, a point on the surface of the superellipsoid satisfies $\widehat{F} = 1$ (this function corresponds to the inside-outside function F after a rigid transform, see Appendix A), therefore we can search for the minimum of the following energy:

$$E(A) = \sum_{i=1}^{N} \left[1 - \widehat{F}(x_d, y_d, z_d, a_1, a_2, a_3, \epsilon_1, \epsilon_2, \varphi, \theta, \psi, \boldsymbol{t}) \right]^2 . \qquad (3)$$

The derivatives of E with respect to the eleven parameters of the superellipsoid can be explicitly computed. The minimum of the energy is computed by using the algorithm of multidimensional conjugate gradient method [49]. This method localizes the minimum of a function only with the computation of its first derivatives, and insures a fast convergence. Figure 3 represents the result
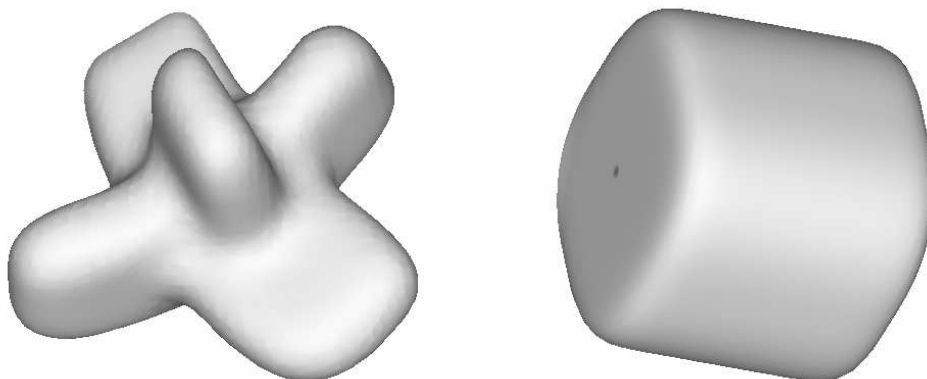

Figure 3: Left: Synthetic 3D data. Right: Fit with a superellipsoid.

of the fit of a superellipsoid on a synthetic set of 3D points (three crossing superquadrics deformed by a global tapering, see [12]).

2.4 Geometric interpretation

The exact computation of the Euclidean distance between a 3D point and a
surface is very expensive and usually an approximation is computed by use
of iterative filters [15, 26]. In this section, we detail two analytic expressions
of the approximation of the distance between a point and an implicit surface.
The first one provides a geometric interpretation of the least-squares criterion
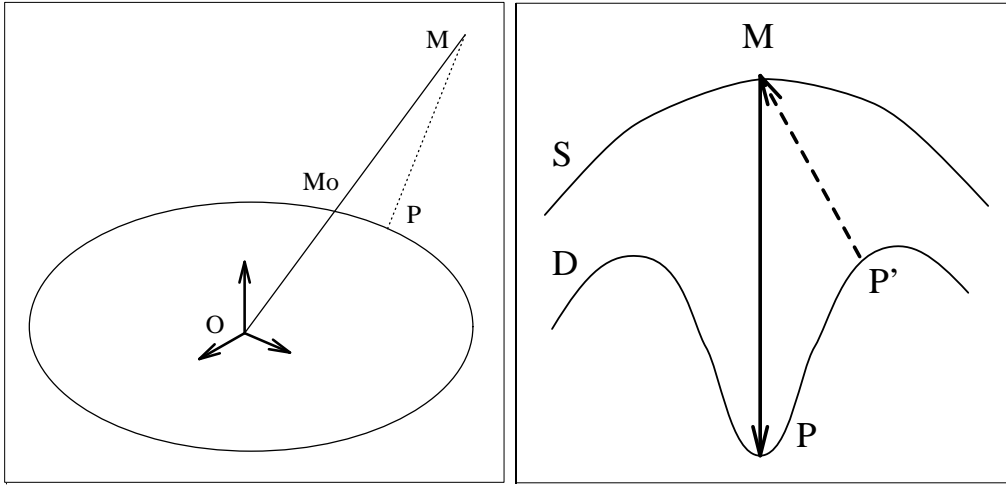which is used to fit the data with a superellipsoid.

Figure 4: Left: Approximation of the Euclidean distance between a point and
the superellipsoid. Right: Two ways to compute the closest data point to M.

- Intrinsic approximation:
  Let us consider a superellipsoid in its intrinsic local frame (Eqs. (2) and
  (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 4). We want to calculate the distance $\| M_0 M \|$.
  Since O, M_0 and M are on a same line, we can write : $\overrightarrow{OM_0} = \mu \, \overrightarrow{OM}$
  with $\mu > 0$. Therefore we have $(x_{M_0}, y_{M_0}, z_{M_0}) = (\mu \, x, \mu \, y, \mu \, z)$, and since
  $F(\mu \, x, \mu \, y, \mu \, z) = \mu \, F(x, y, z)$, with $F(M_0) = 1$, we obtain the value of

μ:

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

We then deduce the following:

$$\begin{cases} \|M_0 M\| &= |1 - \mu| \, \|OM\| \\ &= \left| \frac{F(x, y, z) - 1}{F(x, y, z)} \right| \|OM\| \\ \|M_0 M\| &= |\frac{1-\mu}{\mu}| \, \|OM_0\| \\ &= |F(x, y, z) - 1| \, \|OM_0\| \end{cases}$$

And:

$$(F - 1)^2 = \left(\frac{\| M_0 M \|}{\| OM_0 \|} \right)^2 = \left(\frac{\| OM \|}{\| OM_0 \|} - 1 \right)^2.$$

This gives a geometric interpretation of the least-squares criteria as an intrinsic homothetic characteristic. This approximation is always greater than the exact Euclidean distance $d = \| MP \|$, as one can see in Figure 4.

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

$$d \approx \frac{| F(M) - 1 |}{\| \nabla F(M) \|} \tag{4}$$

It is also possible to define a more precise approximation that is bounded as shows Taubin [56], but it is computationally more expensive.

3 Refinement of the fit with free form deformations

After the fit with a superellipsoid, we have a parametric representation of the 3D data. The model is correctly oriented with respect to axes of inertia of the data, and the 3 axes already have the good size. But it is not close enough to the data, because the set of shapes described by superellipsoids is too limited

for the description of complex structures such as the anatomical ones, e.g. the cardiac ventricle.

So we have to refine this representation. We decided to associate to the super-ellipsoid model a global volumetric deformation called free form deformation (FFD). It is a tool devoted to the deformation of solid geometric models in a free-form manner. FFDs can also deform all kinds of surfacic primitives: planes, implicit surfaces, quadrics, superquadrics.

The main interest of FFDs is that the resulting deformation of the object is just defined by a small number of points, instead of the displacement of every model point. This typical feature will allow us to represent complex 3D data sets by parametric models defined by a few parameters.

In the next three sections, 1. we explicit the definition of FFDs, 2. we explain how to use them to refine the superellipsoid representation by solving an inverse problem, 3. we finally present an iterative two-step algorithm.

3.1 Definition of free form deformations

FFDs were introduced by Segerberg and Parry [50] in computer graphics. An analogy, to understand FFDs, is to consider a rubber-like box in which the object that we want to deform is embedded. To deform the object, one just needs to deform the box, and the object follows in an way similar to an elastic parallelepiped the volumetric deformation of the box. FFDs became very popular in the field of CAD/CAM, essentially because the object deforms in a very intuitive way with respect to the deformation transmitted to the box.

More precisely, FFDs are an application from $\mathbb{R}^3$ to $\mathbb{R}^3$, defined by the tensor product of trivariate Bernstein polynomials. The use of $\mathbb{R}^3$ to $\mathbb{R}^3$ applications already appears in a paper from A. Barr [12] on specific regular deformations, namely for twisting, bending and tapering geometric solids.

The principle of FFDs is as follows: the object to be deformed is embedded in a 3D box. Inside this box, a volumetric grid of points is defined, which links the box to the object (by the previously defined polynomial which defines the deformation function). The box is then deformed by the displacement of its lattice, and the deformation is transmitted to the object with the deformation function (see Figure 5).

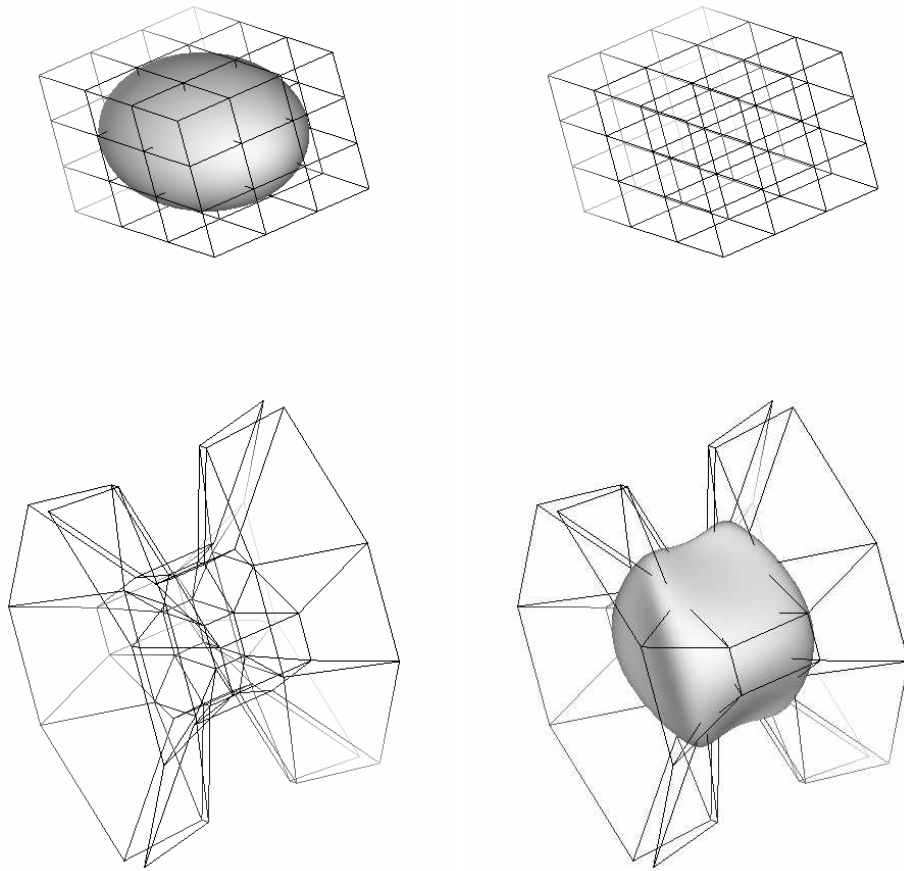Several extensions of FFDs were published. In particular, Coquillart [24, 25]

Figure 5: Free form deformations in computer graphics. Top left: object embedded in the initial box of control points. Top right: initial box. Bottom left: deformed box. Bottom right: resulting deformed object in the deformed box.

replaced the trivariate Bernstein polynomials (associated to Bezier curves) by cubic B-splines to define the deformation functions, and mainly expanded the notion of parallelepipedic box to free form grids of control points; that permits to be closer to the object to be deformed. Let us quote also Griessmair *et al.* [31], Chang *et al.* [17], who propose a generalized De Casteljau algorithm to compute the model deformed by an FFD, and Lamousin *et al.* [39] who use NURBS (non uniform rationnal B-splines) as deformation functions.

We decided to use the original definition of FFDs [50]. The motivation of this choice is essentially practical. Actually, the use of Bernstein polynomials as deformation functions, and the use of parallelepipedic boxes to define grids of control points insures a fast computation of FFDs and gives good results for the representation of medical 3D data.

The FFD formulation is divided into two steps:

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

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

The corresponding equations are presented in Appendix B. An important characteristic of FFDs is that a parametric surface remains parametric after deformation. A complete description of FFDs can be found in [50] and [46]. Figure 5 is an illustration of FFDs.

3.2 Solving the inverse problem

The superellipsoid fit provides a first parametric approximation of the set of 3D data, which is often a crude approximation of the original data points set. We want to use FFDs to refine the representation of the data we obtained with the superellipsoid. The main interest of FFDs is that the resulting deformation of the object is just defined by the position of the deformed box, instead of the displacement of every model point. This typical feature will allow us to represent complex 3D data sets by a few parameters, and we will be able to reach compression ratios of the order of 50.

Solving the inverse problem will allow us to deform the superellipsoid with

respect to the data. Hsu *et al* [37] deal with a simplified version of this problem in the field of CAD/CAM. Their problem was to be able to directly manipulate points of a model deformed by FFDs.

3.2.1 Computation of the deformation function

First, we need to define a parallelepipedic box which contains the original superellipsoid. To estimate the size and orientation of this box, we use 6 of the 11 parameters which characterize the superellipsoid. The sizes a_1, a_2, a_3 according to the 3 axes of inertia define the size of the box, and its orientation is given by the coefficients of the rigid transform $\varphi, \theta, \psi, t_x, t_y, t_z$ of the superellipsoid. We then create the volumetric grid of $(l+1)(m+1)(n+1)$ control points in the box. Eq. (17) can be rewritten:

$$\begin{cases} x(\boldsymbol{P}_{ijk}) & = & a_1(1-2\frac{i}{l}), \\ y(\boldsymbol{P}_{ijk}) & = & a_2(-1+2\frac{j}{m}), \\ z(\boldsymbol{P}_{ijk}) & = & a_3(1-2\frac{k}{n}), \end{cases}$$

if we choose as the local frame origin, the point X_0 with coordinates $(a_1, -a_2, a_3)$. The application of the rigid transform finally leads to the correct box.

Eq. (18) of Appendix B, which links control points $\boldsymbol{P}_{ijk}$ to model points $\boldsymbol{X}$, can be written in a matrix form:

$$\boldsymbol{X} \; = \; \boldsymbol{BP}, \tag{5}$$

where $\boldsymbol{B}$ is the deformation matrix $ND \times NP$ (ND: number of points on the superellipsoid, NP: number of control points $= (l+1)(m+1)(n+1)$), $\boldsymbol{P}$ is a matrix $NP \times 3$ which contains coordinates of the control points $\boldsymbol{P}_{ijk}$, and $\boldsymbol{X}$ a matrix $ND \times 3$ with coordinates of the model points.

The grid of control points being defined, it is sufficient to compute the local coordinates of the superellipsoid points to get the deformation matrix $\boldsymbol{B}$.

Actually Eq. (5) corresponds to three separate linear systems, one for each coordinate. Since the matrix of the three systems is the same, we will from now on consider that this equation stands for the first coordinate of points ($\boldsymbol{X}$ and $\boldsymbol{P}$ column vectors).

3.2.2 Evaluation of a displacement field between the superellipsoid and the data

We need to evaluate a displacement field between the superellipsoid and the data (we suppose again it is a set of 3D points). Figure 4 represents an over-simplified section of the approximation of data D by a superellipsoid S. To point M, we want to associate its closest point on D. There are two candidates, P and P'. To get the best approximation of the data D, it is better to associate M to P. This choice is still valid for other kinds of data, in the sense that data will always have a more complex shape than their approximation by a superellipsoid.

A usual technique consists in the computation of the data distance map, and then, for each superellipsoid point, search its closest point on D. In this case, all the displacement vectors will be directed towards the normal vectors to the data. Thus M will be associated to P'.

On the other hand, if we compute the distance map of the superellipsoid, and, for each data point, search its closest point on S, then the displacement vectors will be directed towards the normal vectors to S, and M associated to P. But some points of the superellipsoid may not be assigned to a closest point. In that case, a displacement vector is computed by interpolation.

The distance map is computed using an algorithm based on KD-trees (see for example [61]).

3.2.3 Deformation of the box of control points

We now want to deform the superellipsoid toward the data, by deforming the box of control points.

If we write $\delta \boldsymbol{X}$ for the displacement field previously computed, the problem we have to solve is equivalent to the resolution of the linear system $(\boldsymbol{X} + \delta \boldsymbol{X}) = \boldsymbol{B}(\boldsymbol{P} + \delta \boldsymbol{P})$, which means, using Eq. (5):

$$\delta \boldsymbol{X} = \boldsymbol{B} \delta \boldsymbol{P}, \tag{6}$$

where $\delta \boldsymbol{P}$ is the variation of the position of control points.

As we said at the beginning of section 3, our goal is to represent data of important size by a parametric deformable model that will be compactly described. This is why the linear system (6) will always be over-determined: the number
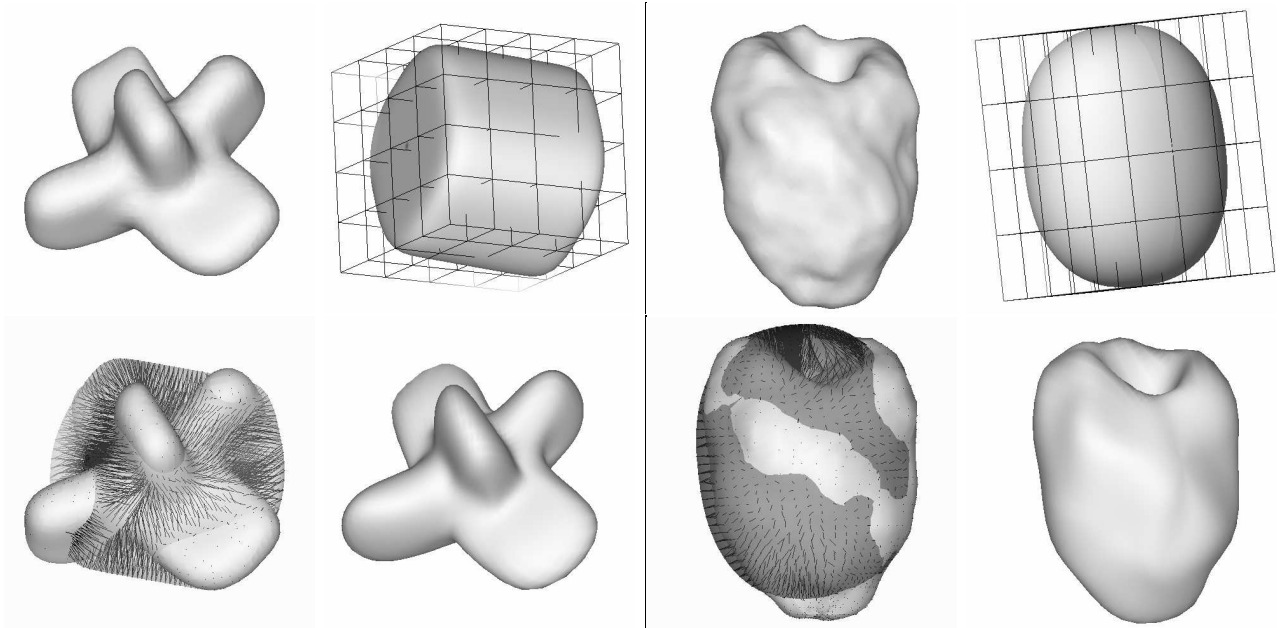
Figure 6: From the superellipsoid to the final model. Left block: synthetic data. Right block: medical data. For each block: Top left: data. Top right: superellipsoid fit and initial box of control points. Bottom left: displacement field between data and the superellipsoid. Bottom right: final model after minimization of the displacement field.

ND of superellipsoid points is always much larger than the number NP of control points. Therefore there is no exact solution for this problem. It is nevertheless possible to find the best solution in a least-squares sense. So we want to solve the problem:

$$\min_{\boldsymbol{P}} \|\boldsymbol{BP} - \boldsymbol{X}\|^2 \equiv \min_{\delta \boldsymbol{P}} \|\boldsymbol{B}\delta \boldsymbol{P} - \delta \boldsymbol{X}\|^2 \qquad (7)$$

We use a minimization algorithm based on the Singular Value Decomposition (SVD) of matrix $\boldsymbol{B}$.

3.3 Iterative two-step algorithm

The resolution of problem (7) leads to a new box of control points, and therefore, by the application of the deformation matrix (Eq. (5)), to a representation model of the data which is more accurate than the initial superellipsoid. Then one can find again a displacement field between this model and the data, and solve again problem (7). Finally, to represent 3D data with a parametric deformable model, we use an iterative two-step algorithm:

- Let $\boldsymbol{P}_0$ be the parallelepipedic box of control points; $\boldsymbol{X}_0 = \boldsymbol{BP}_0$ represents the set of points of the initial superellipsoid.

- We iterate the following two steps:

  > Step 1 : Computation of the displacement field $\delta \boldsymbol{X}_n$ such as: $\boldsymbol{X}_n^a = \boldsymbol{X}_n + \delta \boldsymbol{X}_n$
  >
  > Step 2 : Computation of the control points $\boldsymbol{P}_{n+1}$ by minimization of $\|\boldsymbol{BP} - \boldsymbol{X}_n^a\|^2$
  >
  > Computation of the deformed model: $\boldsymbol{X}_{n+1} = \boldsymbol{BP}_{n+1}$
  >
  > Stop test: computation of the least-squares error $\|\boldsymbol{X}_{n+1} - \boldsymbol{X}_n\|$

This algorithm is similar to the formulation of the B-splines snakes with auxiliary variables (see [22]). An essential feature of this approach is that we use a volumetric deformation to deform a surface shape.
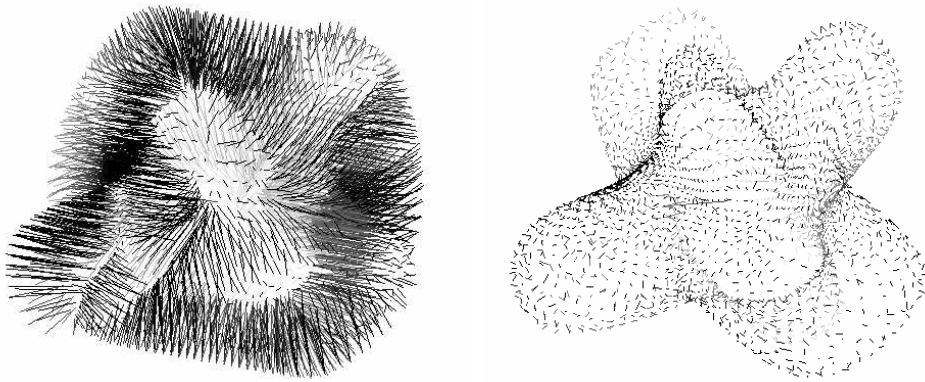
Figure 7: Displacement field of the synthetic example. Left: between the superellipsoid and the data. Right: between the final model (box of size $5 \times 5 \times 5$) and the data.

Figure 6 (left block) is a detailed example of the complete algorithm applied on synthetic data. Figure 7 represents the associated displacement fields between the model and the data, before and after FFDs. Figure 6 (right block) is a detailed example of the complete algorithm applied to medical data. Remark that the final model is totally defined by the following parameters:

- 11 parameters to describe the superellipsoid fitted to the 3D data.

- NP control points to describe the volumetric deformation applied to the initial model.

From a practical point of view, the number of data points varies between 5.000 and 20.000, and we use boxes of size $5 \times 5 \times 5$, which means 125 control points. So the representation of those data by the model allows a compression ratio between 40 and 150.

3.4 Reconstruction from sparse data

Using a parametric model to represent 3D data permits to increase the robustness of the fit in the presence of sparse data. Figures 8 and 9 show the power of the model to retrieve a shape using only sparse data. The first image was generated by intersection of the data showed in Figure 6 (right block) with three orthogonal planes (600 points vs. 6000 points). This is an image that could have been generated by three ultrasound images with different directions. The second image was generated by a uniform subsampling of the same data (900 points vs. 6000 points). This uniform grid of sparse 3D points could be issued from tagged MRI. The shape constraint of the model enables to reconstruct a complete surface which is similar to the model computed from the complete data. With a parametric model, there is a strong global regularity, contrary to general deformable models.

4 Regularization of the model

As shown in the upper left image of figure 10, the control points box may be very irregular. So it becomes difficult to infer quantitative informations on the model deformation from the configuration of those control points. It the
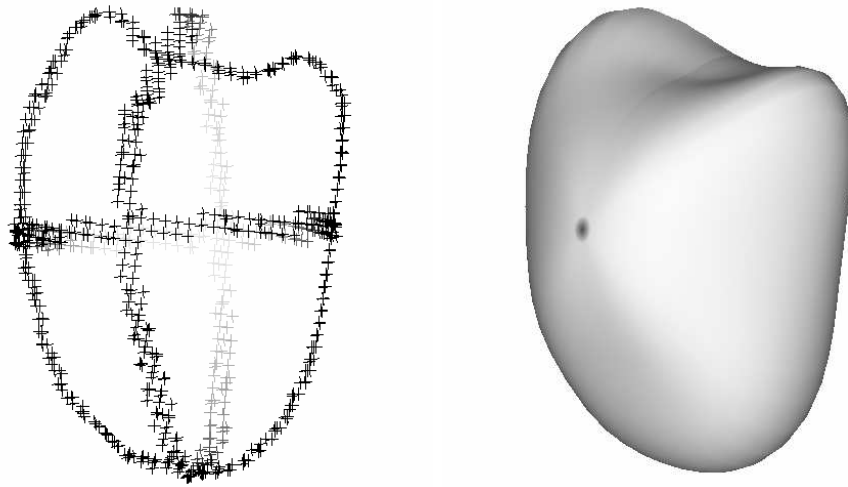
Figure 8: Reconstruction from sparse data (three orthogonal planes). Left: data. Right: final model (box of size $5 \times 5 \times 5$).
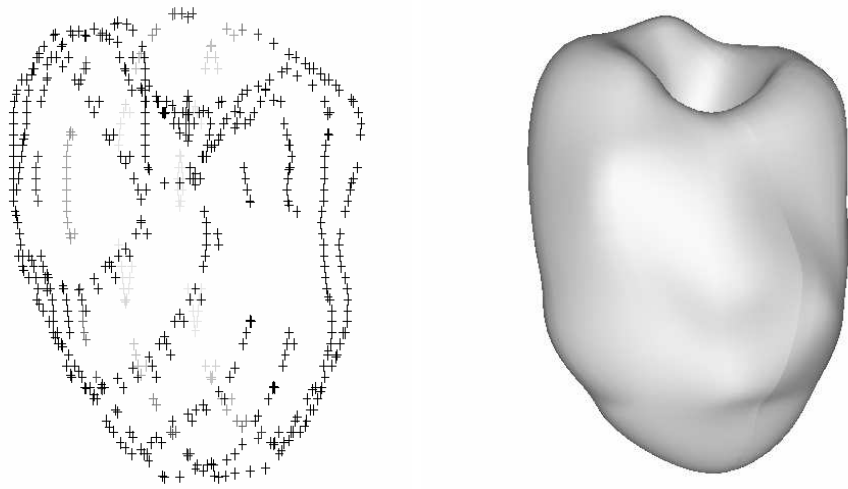
Figure 9: Reconstruction from sparse data (uniform subsampling of 6). Left: data. Right: final model (box of size $5 \times 5 \times 5$).

reason why we will try to explicitly control the regularity of the deformation. We study two possible approachs:

1. Control the norm of the control points displacement with the Singular Value Decomposition of the deformation matrix $\boldsymbol{B}$.

2. Add a regularizing term in the least-squares criterion (7).

FFDs have also been used to solve some matching problems by Szeliski and Lavallee [54, 55]. To impose *a priori* regularity constraints on the deformation parameters, they use m-order stabilizers.

4.1   Using the singular value decomposition

The linear system (Eq. (5)) is solved using Singular Value Decomposition [49]. This decomposition can be written:

$$\boldsymbol{B} = \boldsymbol{U}.\boldsymbol{W}.\boldsymbol{V}^T, \tag{8}$$

where $\boldsymbol{U}$ and $\boldsymbol{V}$ are orthogonal matrix, and $\boldsymbol{W}$ is diagonal. It can be shown that matrix $\boldsymbol{V}.\boldsymbol{W}^{-1}.\boldsymbol{U}^T$ is the pseudoinverse of $\boldsymbol{B}$. The elements of $\boldsymbol{W}$ are the nonvanishing singular values of $\boldsymbol{B}$.
By computing the pseudoinverse of $\boldsymbol{B}$ using only the largest singular values (the largest elements of $\boldsymbol{W}$), we control the norm of the solution of (5). Results of this control are shown in figure 10 with a $5 \times 5 \times 5$ box. Figure 11 shows the effect of this control on the approximation accuracy.

4.2   Using a regularizing term

By adding a regularizing term, the minimization criterion becomes:

$$\|BP - X_n^a\|^2 + \alpha \sum_{j=1}^{NP} \sum_{j'} \|P_j - P_{j'}\|^2, \tag{9}$$

where j' corresponds to the neighbours of $\boldsymbol{P}$. The second term is an internal energy corresponding to the insertion of zero-length springs between control points. This has a regularizing effect on the box to an extent controlled by the
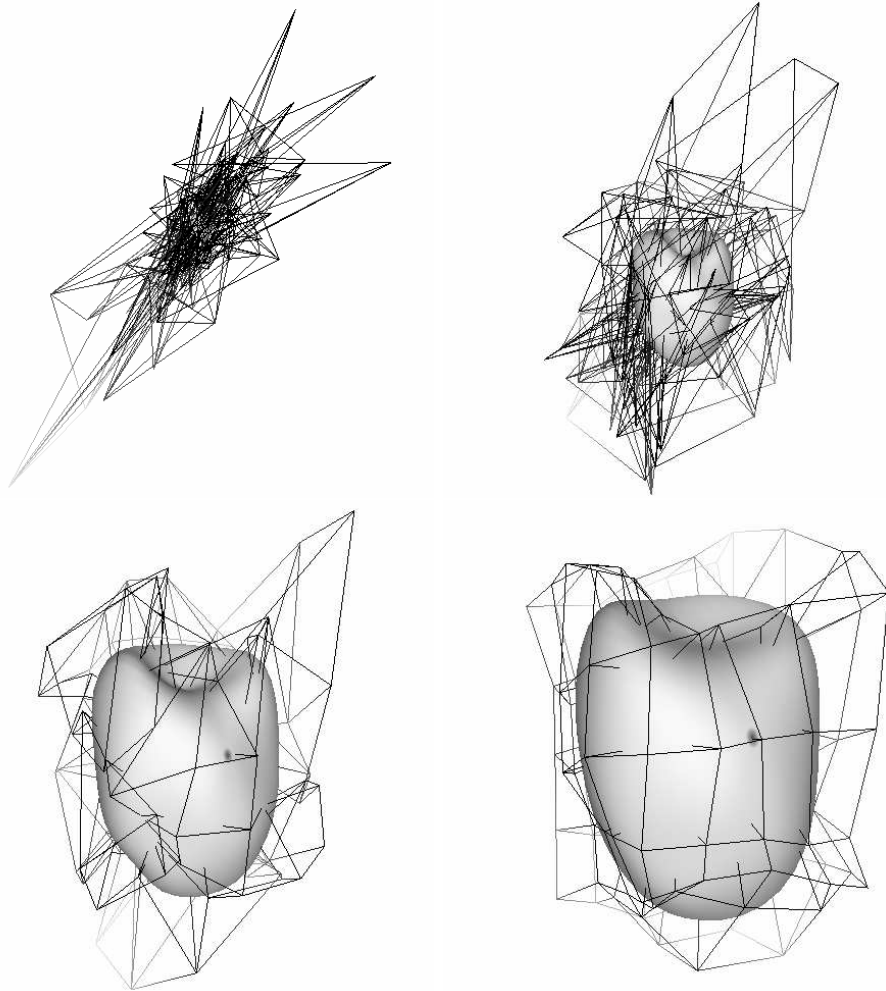
Figure 10: Regularization with SVD (from left to right and top to bottom, the number of diagonal terms is decreasing). The model size changes between the examples are due only to rescaling, the real size is the same).
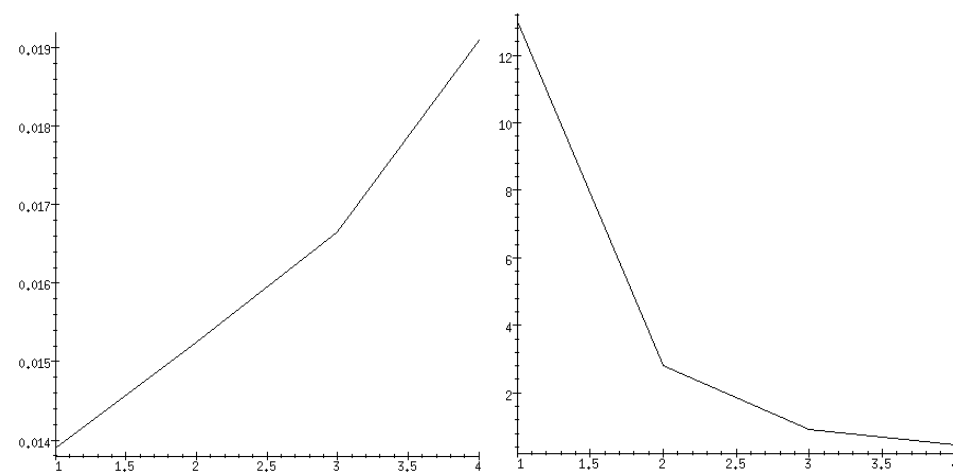
Figure 11: Left: Approximation error between the models and the data, with a decreasing number of diagonal terms. Right: Deformation of the box of control points, with a decreasing number of diagonal terms.
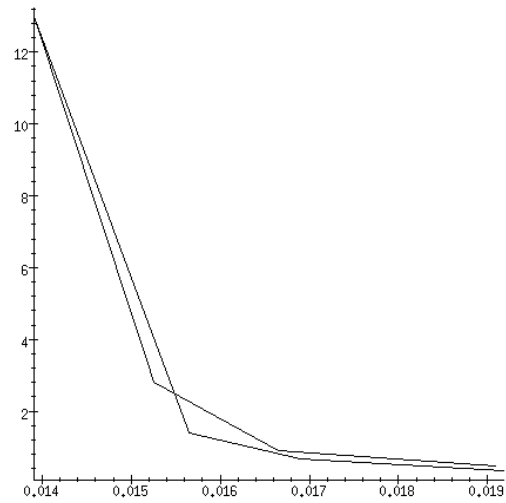
Figure 12: Norm of the control points box deformation versus least-squares approximation error between the models and the data. The 2 curves correspond to a regularisation with SVD or with a regularizing term.

weight α. This regularization term can be also written $\|DP\|^2$ where D is a matrix which represents a discretized derivative of the control points position.

Figure 12 represents the compared effects of the two regularisations. More precisely, we represent for each of the two methods the correlation between the least-squares approximation error and the norm of the control points box deformation. This figure shows that the two regularisation methods are equivalent.

5   Application to the representation of 3D medical data

We present in this section applications of the algorithm on 3D cardiac images. Those images have been obtained with nuclear medecine (SPECT images) and volumetric scanner X (Dynamic Spatial Reconstructor - Mayo Clinic).

The SPECT images (Single Photon Emission Computed Tomography) are obtained by the measure of the electromagnetic radiation coming from radioactive isotopes injected in the human body. This kind of images permits to get informations about the metabolic behavior of organs. There are functional images. The SPECT MIBI image (with corresponds to a new myocardial perfusion tracer, technium 99m MIBI) is a volume of $64 \times 64 \times 64$ voxels. The SPECT image is a volume of size $64 \times 64 \times 21$.

The DSR image has a very good resolution ($98 \times 100 \times 110$ for a voxel of 0.926 mm3). It was obtained by injection of a contrast product in the ventricle, which allows to distinguish clearly the cavity.

The 3D original data are visualized as a series of 2D cross-sections (according to the Z axis) on Figures 13 and 14.
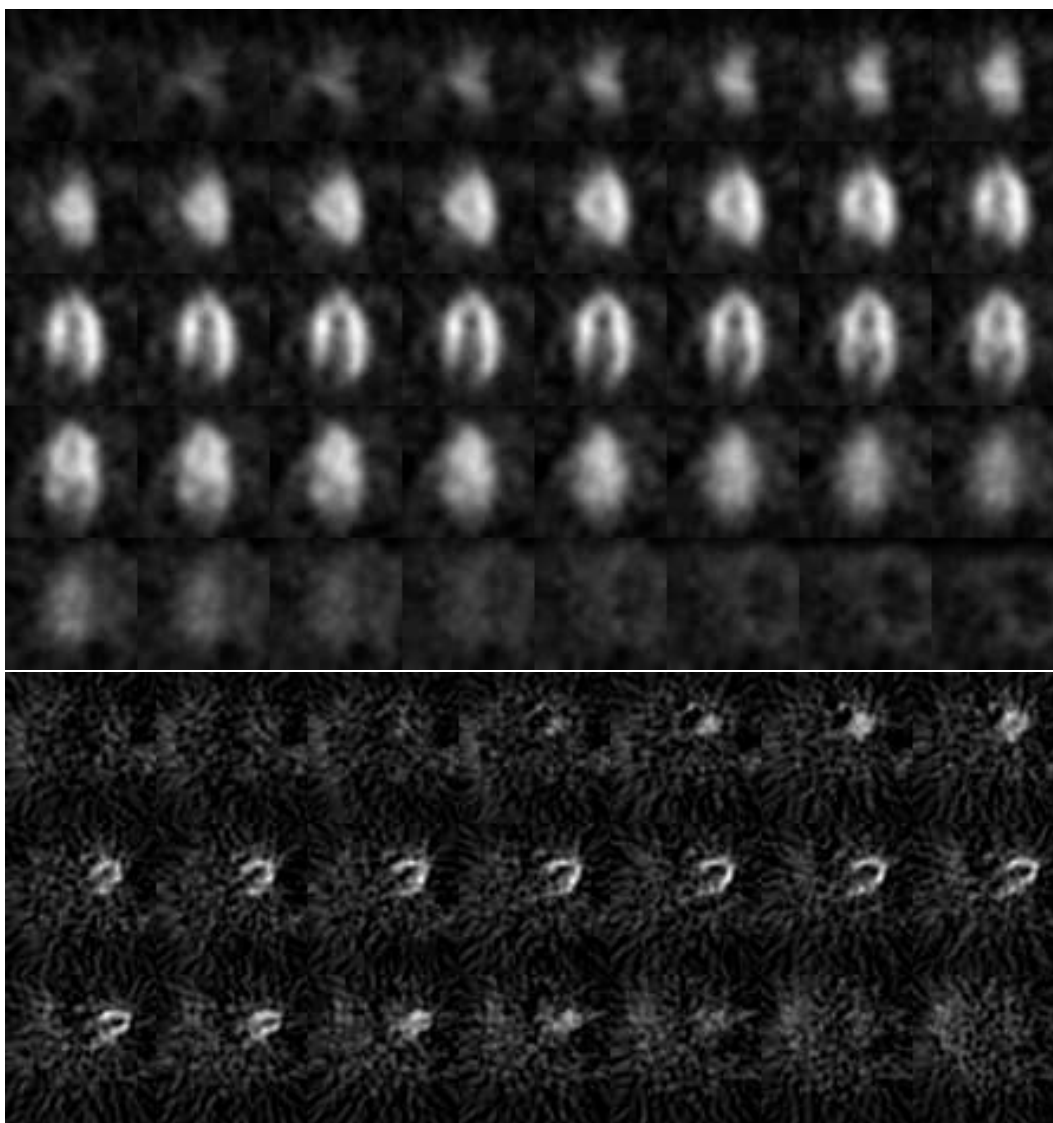
Figure 13: 3D images of the left ventricle (order of cross-sections: from left to right and from top to bottom). Top: SPECT MIBI - SOPHA Medical. Bottom: SPECT - FOCUS Medical.

Figure 14: 3D image of the left ventricle (order of cross-sections: from left to right and from top to bottom). (Scanner X - DSR).

Morphological segmentation of the data

In order to get a set of 3D points which corresponds to the anatomical structure that we want to study (epicardium or endocardium or midwall of the cardiac left ventricle), and therefore fit our model on this set of points, we have to segment the original 3D data. As one can visually remark on Figures 13 and 14, this is not an easy task. The SPECT images are very noisy, and, on the DSR image, the segmentation of the external wall of the ventricle (epicardium) cannot be done with a simple thresholding.

To obtain an accurate and robust segmentation, we must combine thresholding with mathematical morphology and connected components analysis (as in Hoehne [36]). We first choose a threshold which grossly separates the ventricle (high values in SPECT and DSR images) from the rest of the image. Then we choose the largest connected component in the resulting binary image, and perform a equal number of erosions and dilations (morphological closings). This last operation is necessary to bridge little gaps (in particular for the epicardium on the upper part of the DSR image) and smooth the overall segmentation. Finally, the extraction of an isosurface from that last image provides the set of 3D points that we need as input for the complete reconstruction algorithm. Results of those operations are presented on Figures 15, 16 and 17. For a correct estimation of the quality of those segmentations, we superimposed the segmented surface on the image (we present cross-sections along the Z axis).

Fit of the model

We applied on those data points sets a parametric model defined by a box of control points of size $5 \times 5 \times 5$, and the 11 parameters which describe the initial superellipsoid. We used the iterative two-step algorithm detailled above (see section (3.3)), with 30 iterations. The results are presented on Figures 15, 16 and 18 (cross-sections along the Z axis).

An important characteristic of FFDs is their volumetric aspect: each space point is submitted to the deformation coming from the trivariate tensor product of 3 Bezier curves. Therefore, it is possible to reconstruct several surfaces simultaneously with the deformable parametric model that we presented. Our model then gives a natural estimation of the deformation of any point included in the volume between the two surfaces.
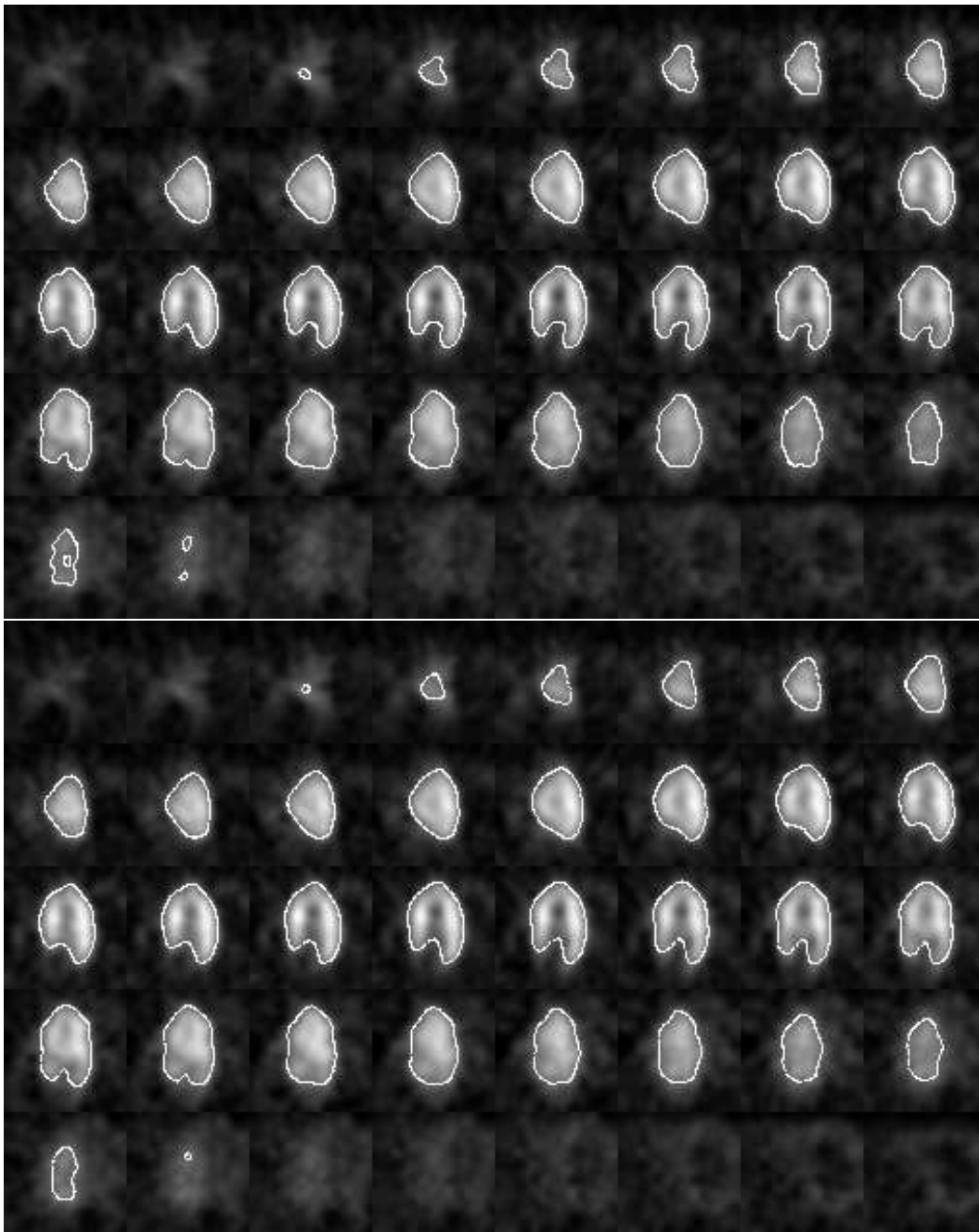
Figure 15: Segmentation and representation of the epicardium (external wall of the ventricle) for the 3D image of the left ventricle (SPECT MIBI). Top: isosurface superimposed to the image. Bottom: reconstruction by the model.
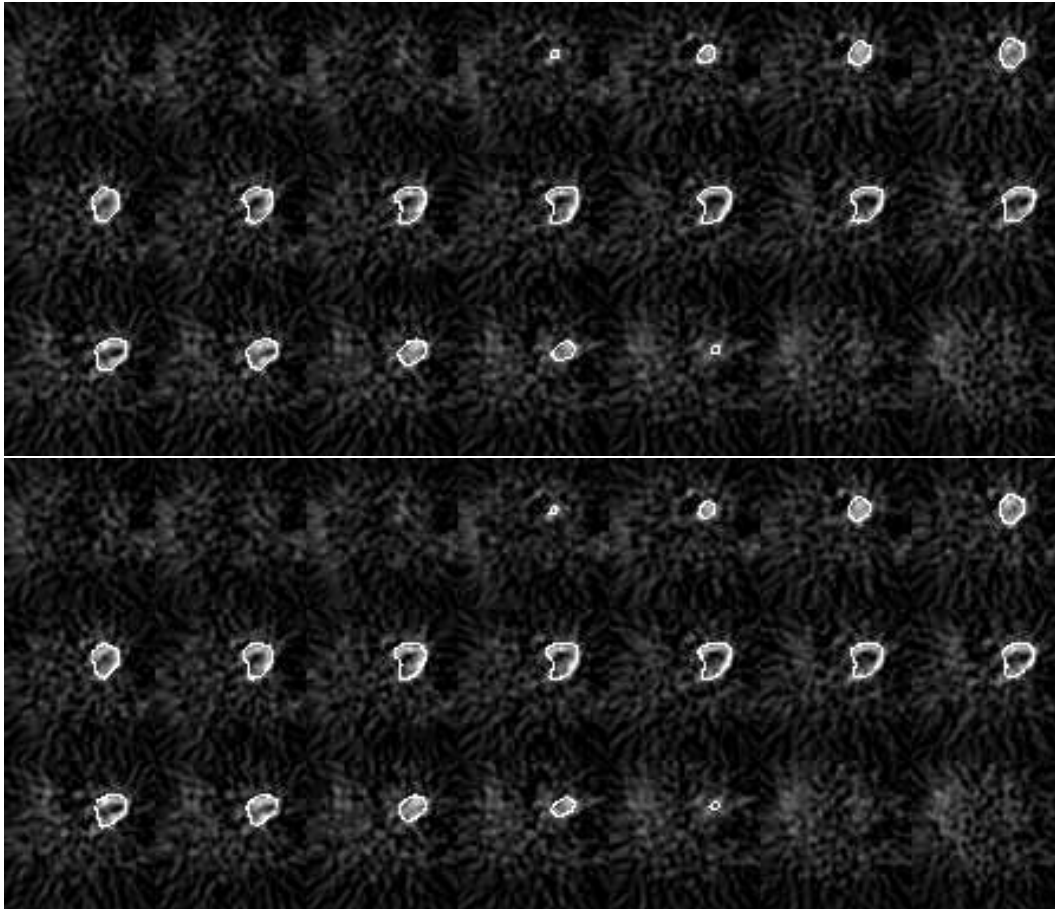
Figure 16: Segmentation and representation of the epicardium (external wall of the ventricle)for the 3D image of the left ventricle (SPECT - FOCUS Medical). Top: isosurface superimposed to the image. Bottom: reconstruction by the model.
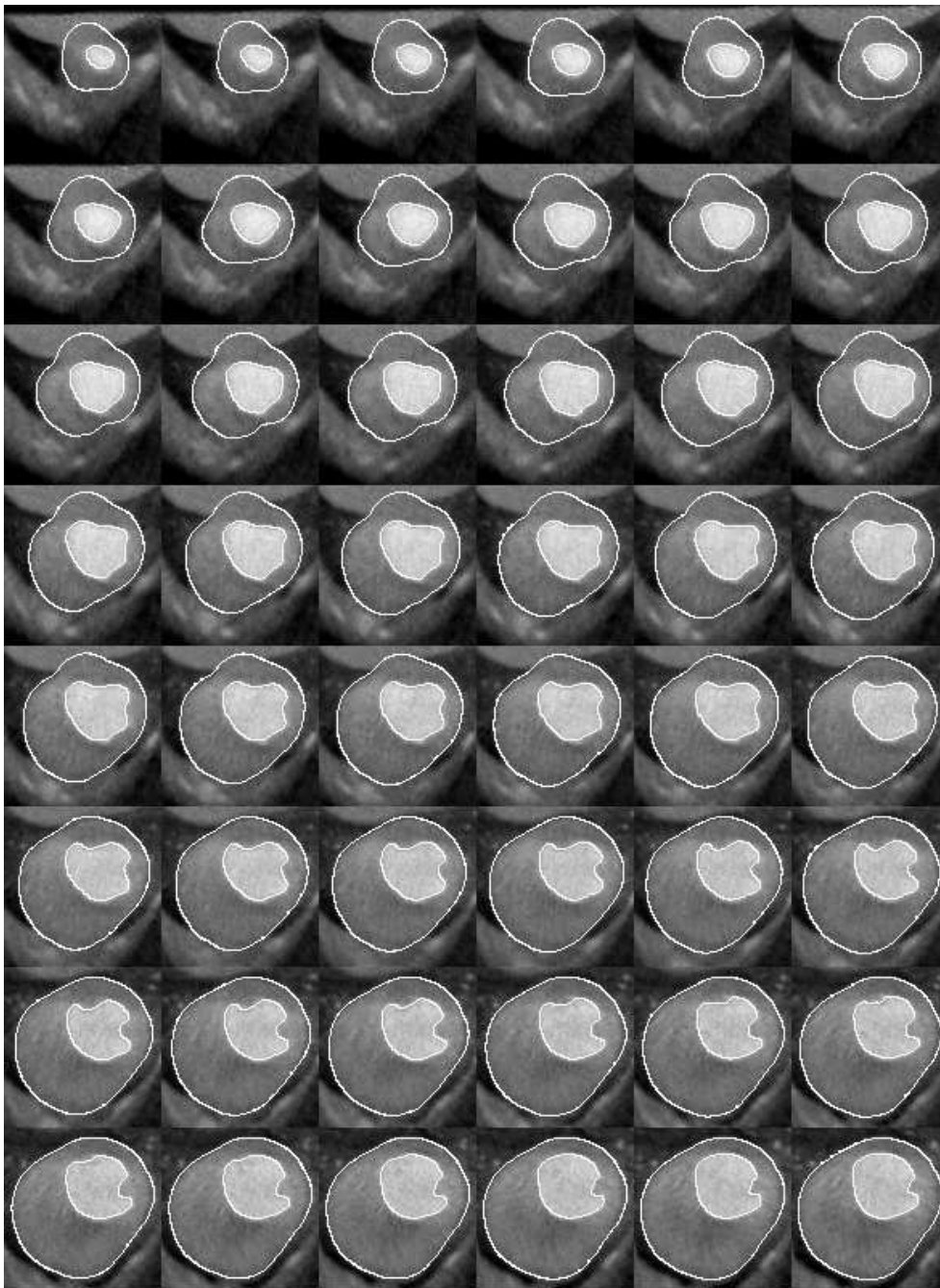
Figure 17: Segmentation of the epicardium and the endocardium (external and internal walls of the ventricle for the 3D image of the left ventricle (Scanner X - DSR).
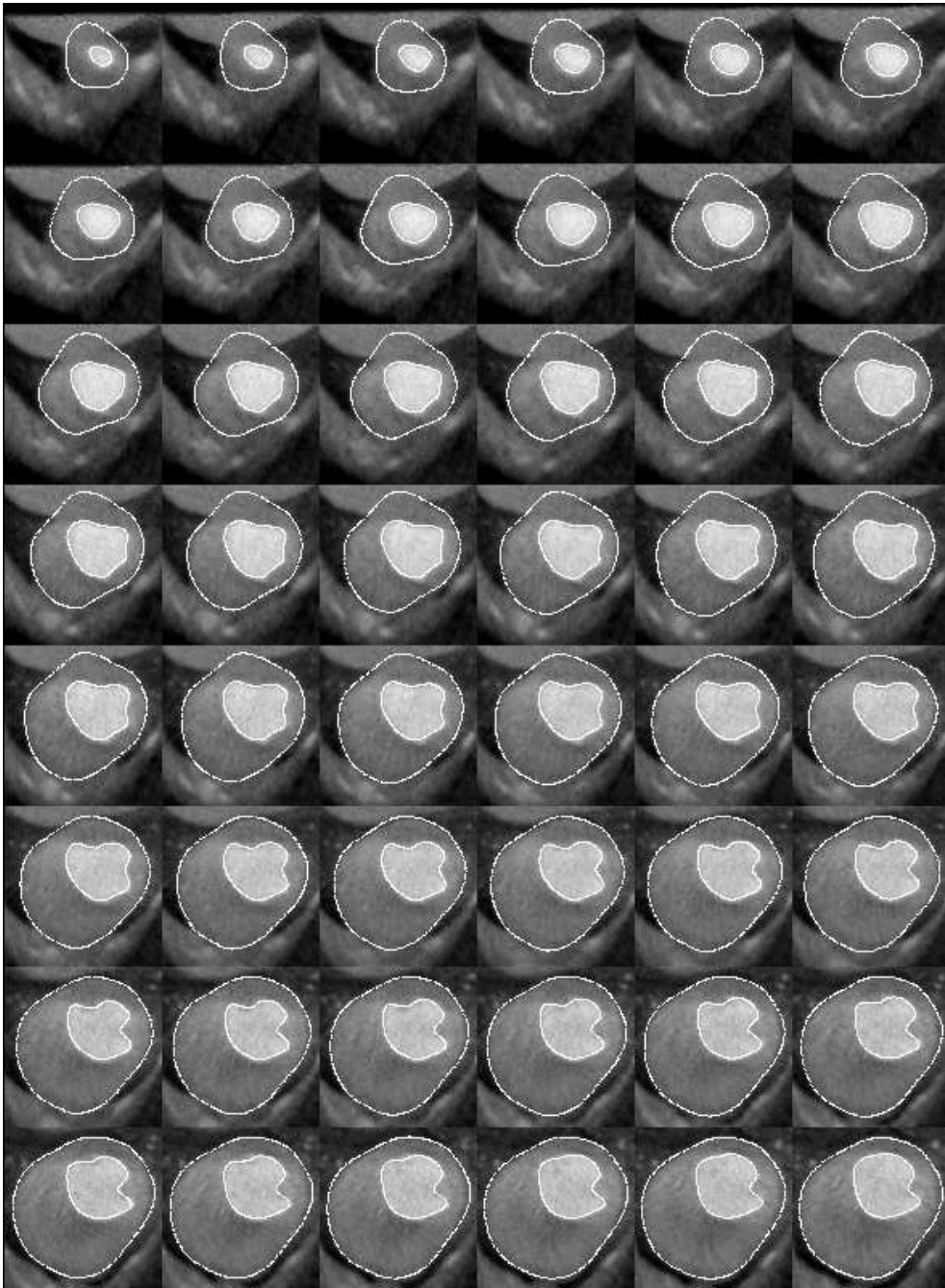
Figure 18: Reconstruction of the epicardium and the endocardium by the model. The 2 surfaces are reconstructed separately.
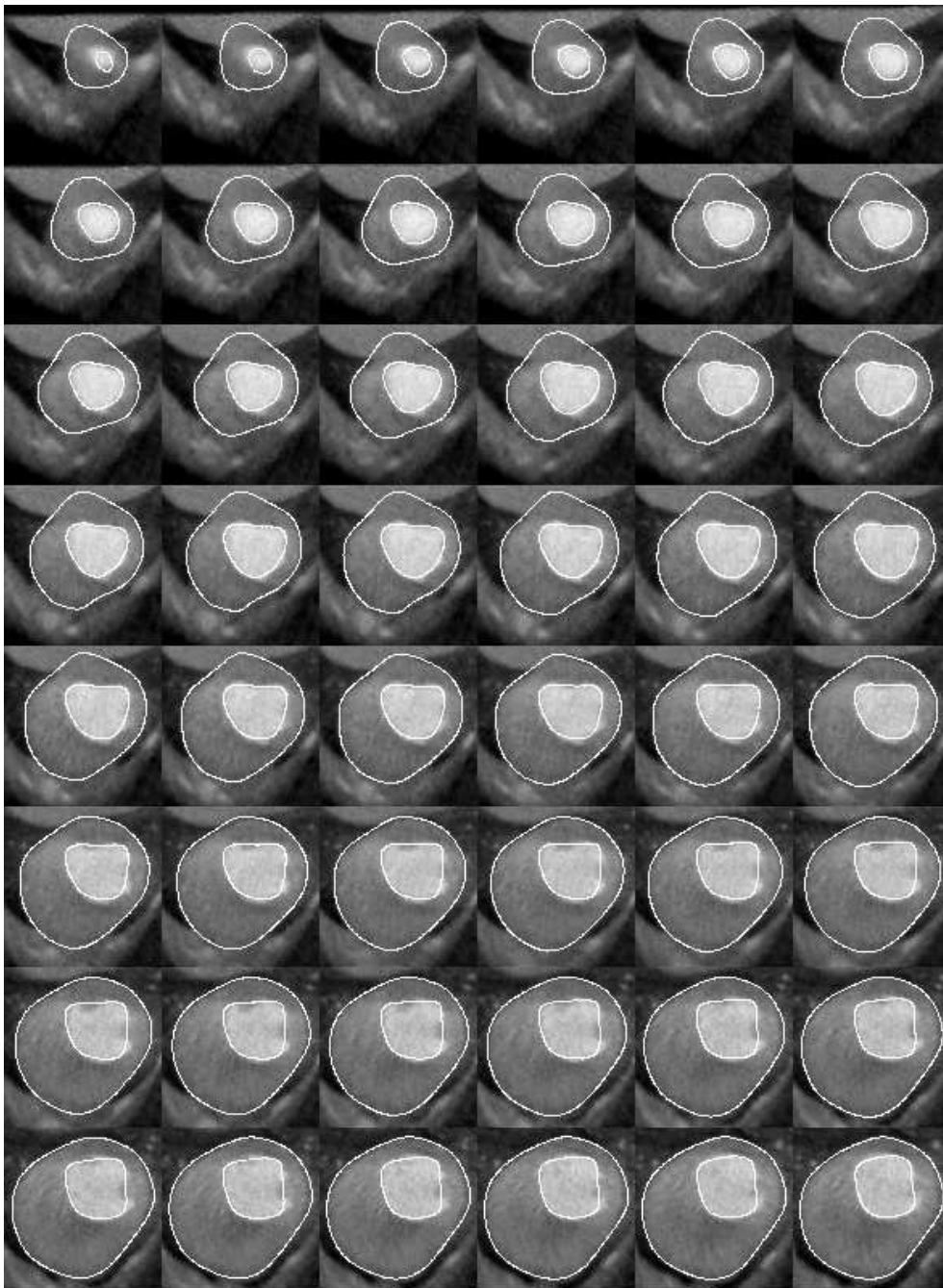
Figure 19: Simultaneous reconstruction of the epicardium and the endocardium by a single model.

From the segmented original image 17, we fitted separately 2 superellipsoids, then computed the respective displacement fields on the endocardium and the epicardium. We used a single box of control points to deform simultaneously the 2 models. The result is presented on Figure 19. One can remark that the endocardium reconstruction is not as good as on Figure 18, when the models were computed separately. This is because the amplitudes of the displacement vectors for the epicardium are much larger than for the endocardium. Therefore, the 3D deformation function tends to smooth the internal wall of the ventricle. On the other hand, we now have a single volumetric deformation to measure the global position of both the epicardium and endocardium surfaces. The use of non uniform grids of control points (see works of Szeliski and Lavallee [54, 55]) would yield to a better precision for each model.

Visualization of the data and model surfaces

The corresponding surfaces of the previous images are shown in this section. Figure 20 shows three different views of the SPECT MIBI data and of the figure 15. Figure 21 corresponds to the SPECT data and to Figure 16. Figures 22 (two different views) and 23 correspond to the DSR data and respectively to Figures 18 and 19.

Compression

The isosurfaces obtained by morphological segmentation are composed respectively of 6000 (SPECT MIBI, Figure 20), 1000 (SPECT, Figure 21) and 20.000 (DSR, Figure 22) 3D points.

With the representation of those data by a parametric model defined by 130 3D points, we reduce the information which is needed for the description of the ventricles by a ratio of 46, 7.7 and 77 respectively for SPECT MIBI, SPECT and DSR data. Note that for the simultaneous computation of one model for the epicardium and endocardium (DSR), we get a compression ratio of 154.

The two diagrams of figure 24 represent the evolution of the least-squares error $\|BP - X\|$ versus the number of iterations as well as the evolution of the deformation of the box of control points $\|P_{n+1} - P_n\|$ (each curve represents a different volume of data).
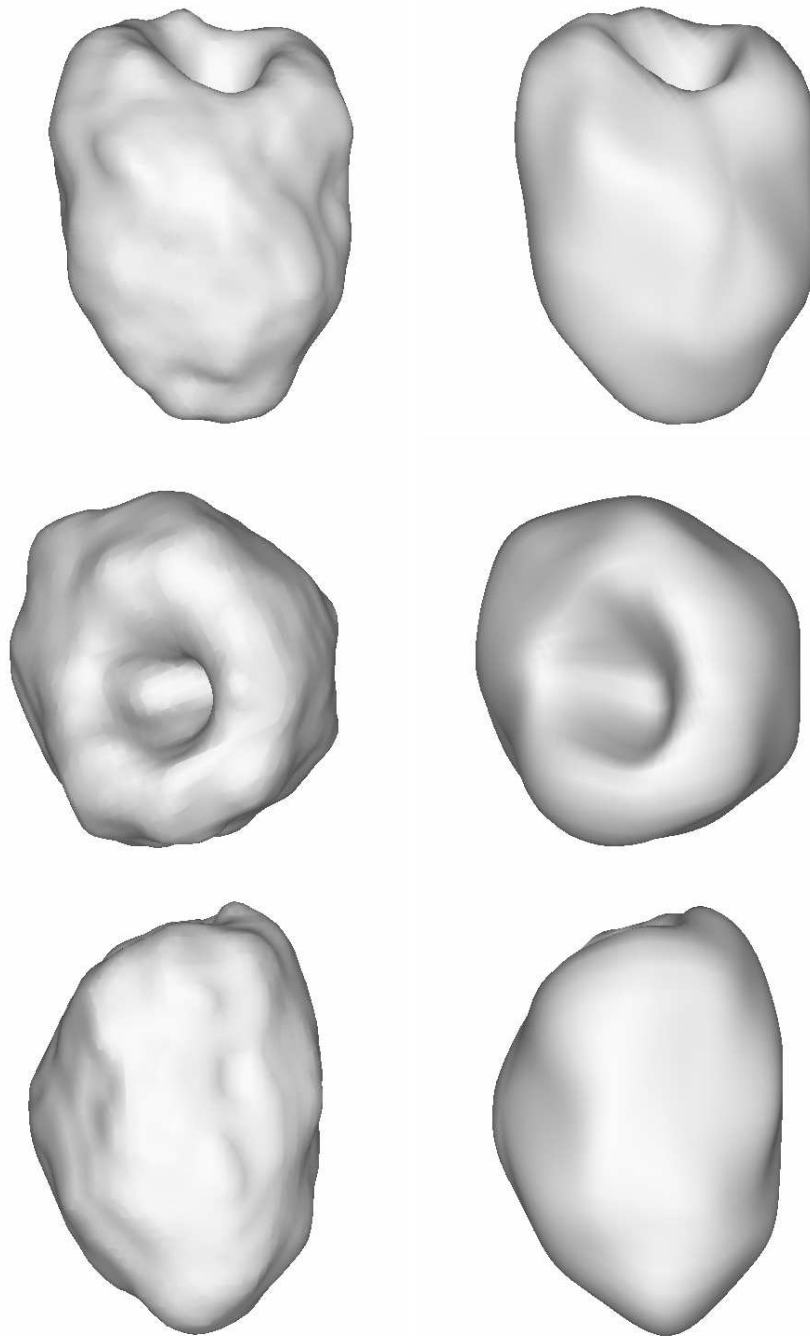
Figure 20: Left: isosurface from the segmentation of SPECT MIBI data (6000 points). Right: representation by the parametric model (defined by 130 points). Compression ratio: 46. (3 view points)
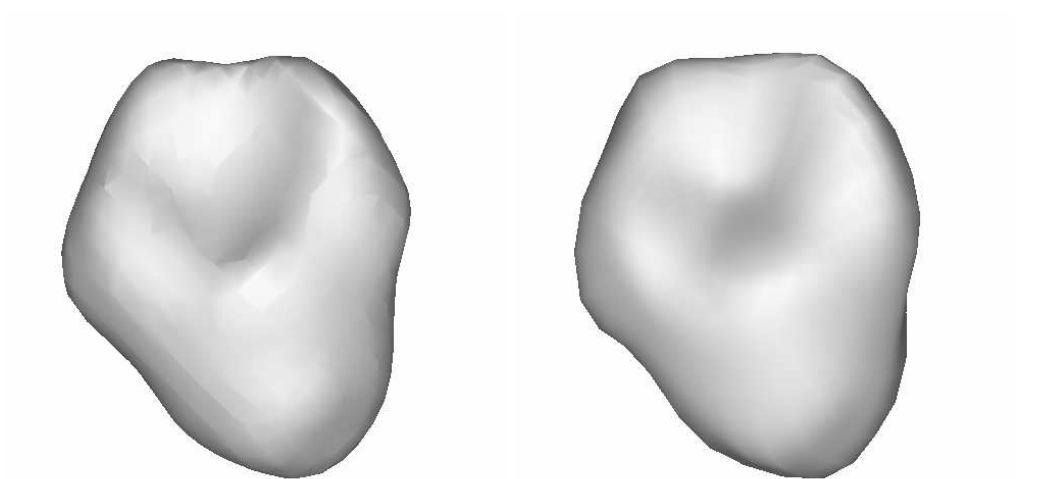
Figure 21: Left: isosurface from the segmentation of SPECT data (1000 points). Right: representation by the parametric model (defined by 130 points). Compression ratio: 7.7.
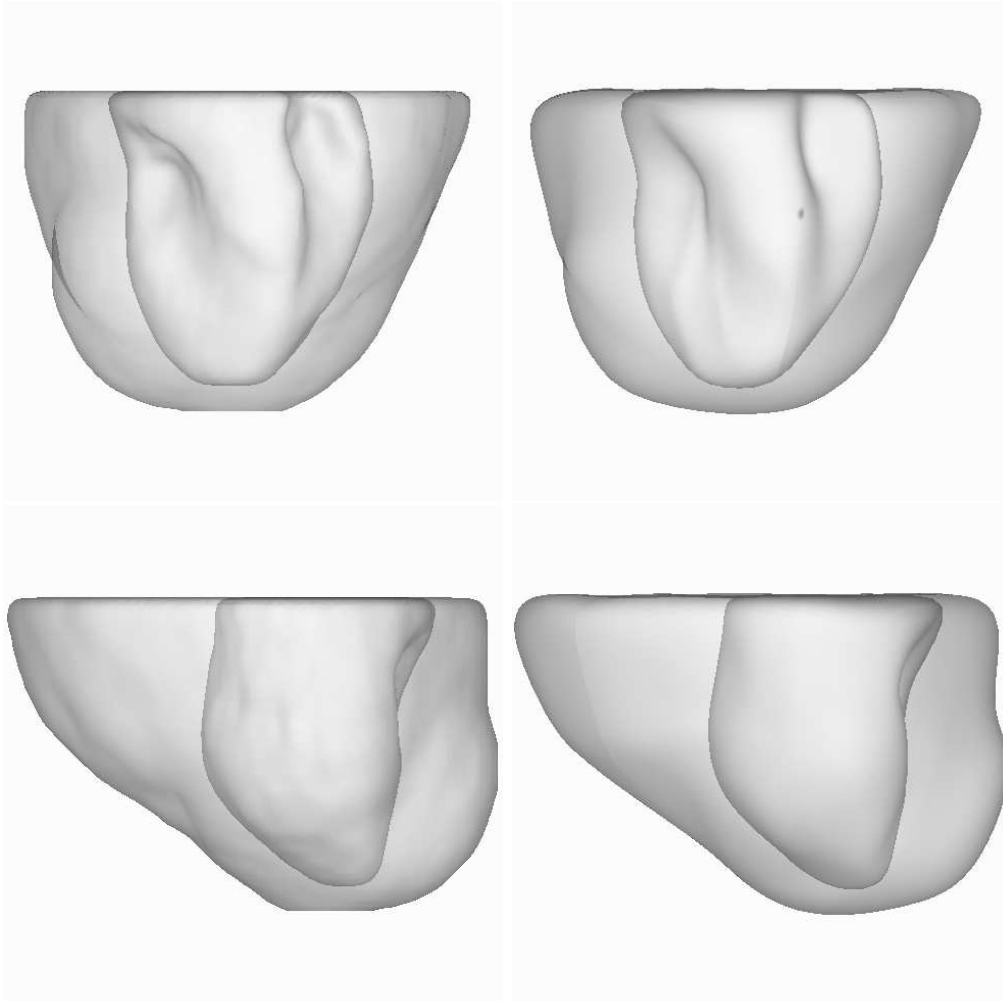
Figure 22: Left: isosurfaces (endocardium and epicardium) from the segmentation of DSR data (10.000 points for each surface). Right: representation by 2 parametric models computed separately (each model defined by 130 points). Compression ratio: 77. (2 view points)
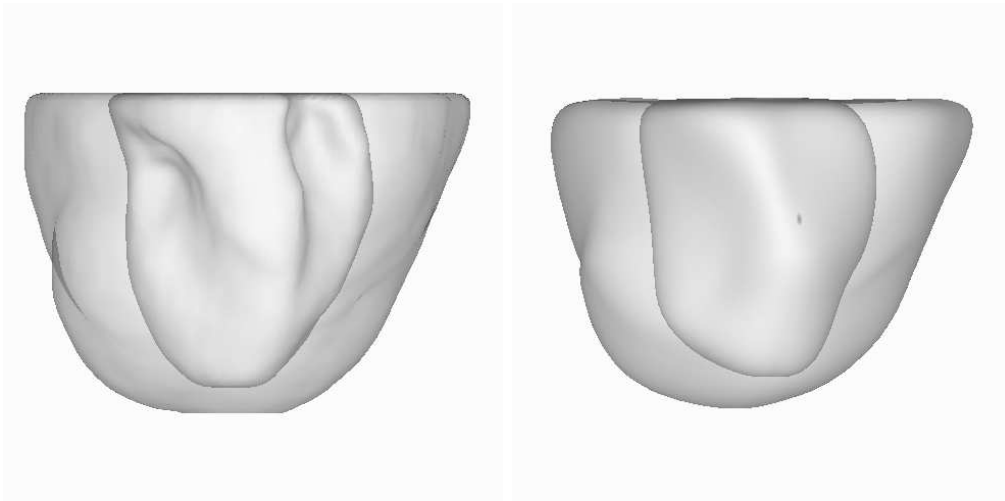
Figure 23: Left: isosurfaces (endocardium and epicardium) from the segmentation of DSR data (10.000 points for each surface). Right: simultaneous representation of the 2 surfaces by a single model (defined by 130 points). Compression ratio: 154.
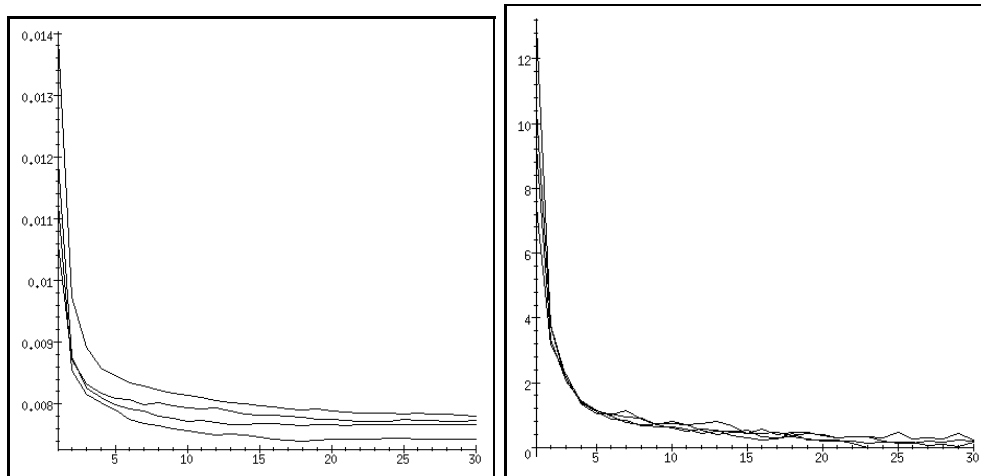
Figure 24: Left: Evolution of the least-squares approximation error between data and the models versus the number of iterations (the 4 curves correspond to 4 different models). Right: Evolution of the deformation of the box of control points versus the number of iterations (the 4 curves correspond to 4 different models).

6   Conclusion

We presented a new approach to shape reconstruction applied to unstructured 3D data. It is based on a first approximation giving the best fit with a superquadric model. This is followed by a two-step algorithm for refining the details of the previous shape by making use of free-form deformations. The data are a set of points in a 3-D image, and the closest point to data is used for the definition of a displacement field. We also presented two methods to control the regularity of the control points box defining the free-form deformations, using singular value decomposition or adding a regularizing term in the previous algorithm. The main advantage of this approach is the compactness of the description of a complex shape, and also the robustness of the fit even in the presence of outliers or sparse data..

Experimental results have been shown for a various number of 3D cardiac images to capture the 3D shape of the left ventricle.

Future work will study the use of this compact and robust model for the dynamic analysis of 4D temporal series of 3D images of the beating heart. This will be presented in a companion paper [9].

Acknowledgments

A Additional details about superquadrics

A.1 Definition

Superquadrics are obtained by the spherical product (see [11]) of two 2D curves. The superellipsoid is the spherical product of the superellipse with itself. The curve defined by the implicit equation:

$$\left(\frac{x}{a}\right)^{\frac{2}{\epsilon}} + \left(\frac{y}{b}\right)^{\frac{2}{\epsilon}} = 1 \tag{10}$$

is a superellipse. A natural parameterization is:

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

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

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

In the implicit equation, powers $\frac{2}{\epsilon}$ means:

$$u^{\frac{2}{\epsilon}} = \left(u^2\right)^{\frac{1}{\epsilon}} = \mid u \mid^{\frac{2}{\epsilon}} .$$

The ϵ coefficient controls the squareness of the curves: courbes :

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

The spherical product of these curves gives a uniform mathematical formulation of the family of superquadrics.

Remark : 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 superellipsoid along x, y and z axes; ϵ_1 and ϵ_2 define the shape of the superquadric along latitude and longitude. Some exemples are presented figure 1.

A.2 Rigid transform

All the equations of the previous section are defined in a intrinsic local frame with the center of the superellipsoid as origin. To fit the superellipsoid on the data, we must be able to position it anywhere in space, that means rewrite those equations in a general frame R_O.

A homogeneous rigid transform T is defined 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 inside-outside function is obtained by replacing $\boldsymbol{x}$ by $T^{-1}(\underline{\boldsymbol{x}})$ in 1:

$$\widehat{F}(\underline{\boldsymbol{x}}) = F(T^{-1}(\underline{\boldsymbol{x}})). \tag{11}$$

Function $\widehat{F}$ now depends on 11 parameters: $(a_1, a_2, a_3, \epsilon_1, \epsilon_2, \varphi, \theta, \psi, t_x, t_y, t_z)$. We use this function to fit the data with the superellipsoid.

A.3 Initialization of the fit

It is a two-step algorithm. The first step consists in the initialization of the parametric surface from the 3D data points set:

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, computed from the matrix of second order centered moments M:

$$M = \frac{1}{N}\sum_{i=1}^{N} \begin{bmatrix} (y_i - \overline{y})^2 + (z_i - \overline{z})^2 & -(y_i - \overline{y})(x_i - \overline{x}) & -(z_i - \overline{z})(x_i - \overline{x}) \\ -(x_i - \overline{x})(y_i - \overline{y}) & (x_i - \overline{x})^2 + (z_i - \overline{z})^2 & -(z_i - \overline{z})(y_i - \overline{y}) \\ -(x_i - \overline{x})(z_i - \overline{z}) & -(y_i - \overline{y})(z_i - \overline{z}) & (x_i - \overline{x})^2 + (y_i - \overline{y})^2 \end{bmatrix}. \tag{12}$$

The center of gravity and the 3 axes of inertia define the new frame of the object. 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:

$$J = \frac{\mu}{3} \begin{bmatrix} a_2^2 + a_3^2 & 0 & 0 \\ 0 & a_3^2 + a_1^2 & 0 \\ 0 & 0 & a_1^2 + a_2^2 \end{bmatrix}, \tag{13}$$

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 follows:

$$\begin{cases} a_1^2 & = & \frac{3}{2\mu}(\lambda_2 + \lambda_3 - \lambda_1), \\ a_2^2 & = & \frac{3}{2\mu}(\lambda_1 + \lambda_3 - \lambda_2), \\ a_3^2 & = & \frac{3}{2\mu}(\lambda_1 + \lambda_2 - \lambda_3). \end{cases} \tag{14}$$

B   Definition of FFDs

The FFD formulation is divided into two steps:

1. Computation of the local coordinates of the object points in the frame defined by the box of control points. We define a coordinate system in the local frame associated to the parallelepipedic box in which the object to be deformed is embedded. The origin of the frame is a vertex $\boldsymbol{X_0}$ of the box, and the axes of this frame are the 3 edges $(\boldsymbol{S}, \boldsymbol{T}, \boldsymbol{U})$ of the box coming from $\boldsymbol{X_0}$. Every 3D point $\boldsymbol{X}$ has (s, t, u) coordinates in this system such as:

$$\boldsymbol{X} = \boldsymbol{X_0} + s\,\boldsymbol{S} + t\,\boldsymbol{T} + u\,\boldsymbol{U}, \tag{15}$$

where s, t, u are given by:

$$s = \frac{\boldsymbol{S} \cdot (\boldsymbol{X} - \boldsymbol{X_0})}{\boldsymbol{S} \cdot \boldsymbol{S}}, \quad t = \frac{\boldsymbol{T} \cdot (\boldsymbol{X} - \boldsymbol{X_0})}{\boldsymbol{T} \cdot \boldsymbol{T}}, \quad u = \frac{\boldsymbol{U} \cdot (\boldsymbol{X} - \boldsymbol{X_0})}{\boldsymbol{U} \cdot \boldsymbol{U}}, \tag{16}$$

Remark that $\boldsymbol{X}$ is inside the box *iff* s, t and u are in $[0, 1]$.
We then generate a grid of $(l + 1)(m + 1)(n + 1)$ control points in the box by setting :

$$\boldsymbol{P_{ijk}} = \boldsymbol{X_0} + \frac{i}{l}\,\boldsymbol{S} + \frac{j}{m}\,\boldsymbol{T} + \frac{k}{n}\,\boldsymbol{U}. \tag{17}$$

The object is linked to the grid of control points by a deformation function defined as a tensor product of trivariate Bernstein polynomials (polynomials associated to Bezier curves). The position X of an arbitrary point is evaluated by the computation of its local coordinates (s, t, u), and of the trivariate vector polynomial:

$$X = \sum_{i=0}^{l} \sum_{j=0}^{m} \sum_{k=0}^{n} C_l^i C_m^j C_n^k (1-s)^{l-i} s^i (1-t)^{m-j} t^j (1-u)^{n-k} u^k \, P_{ijk}. \quad (18)$$

2. Displacement of the control points and estimation of the new position of the deformed object. The deformation of the object is specified by moving the control points P_{ijk} from their latticial position to a new position P_{ijk}^*. The position of a point X of the deformed object is then computed with the previous vector polynomial:

$$X^* = \sum_{i=0}^{l} \sum_{j=0}^{m} \sum_{k=0}^{n} C_l^i C_m^j C_n^k : (1-s)^{l-i} s^i (1-t)^{m-j} t^j (1-u)^{n-k} u^k P_{ijk}^*. \quad (19)$$

References

[1] A. Amini and J. Duncan. Bending and stretching models for lv wall motion analysis from curves and surfaces. In *Proceedings Image and Vision Computing*, volume 10, pages 418–430, August 1992.

[2] A. Amini, R. Owen, P. Anandan, and J. Duncan. Non-rigid motion models for tracking the left ventricular wall. Lecture notes in computer science: Information processing in medical images, pages 343–357, 1991. Springer-Verlag.

[3] N. Ayache. Computer vision, virtual reality and robotics in medecine. *Image and Vision Computing*, August 1995.

[4] N. Ayache, J.D. Boissonnat, E. Brunet, L. Cohen, J.P. Chièze, B. Geiger, O. Monga, J.M. Rocchisani, and P. Sander. Building highly structured volume representations in 3D medical images. In *Computer Aided Radiology*, June 1989. Berlin, West-Germany.

[5] N. Ayache, J.D. Boissonnat, L. Cohen, B. Geiger, O. Monga, J. Levy-Vehel, and P. Sander. Steps toward the automatic interpretation of 3D images. *NATO ASI Series on 3D Imaging in Medicine*, F 60:107–120, 1990.

[6] N. Ayache, I. Cohen, and I. Herlin. *Medical Image Tracking*, chapter 20. MIT Press, 1992.

[7] R. Bajcsy and F. Solina. Three dimensional object representation revisited. In *Proceedings IEEE International Conference on Computer Vision (ICCV)*, pages 231–240, London, June 1987.

[8] E. Bardinet, L. Cohen, and N. Ayache. Fitting of iso-surfaces using superquadrics and free-form deformations. In *Proceedings IEEE Workshop on Biomedical Image Analysis (WBIA)*, Seattle, Washington, June 1994.

[9] E. Bardinet, L.D. Cohen, and N. Ayache. Tracking the cardiac ventricle with a parametric deformable model. *IEEE Transactions on Medical Imaging*. Submitted.

[10] E. Bardinet, L.D. Cohen, and N. Ayache. Superquadrics and free-form deformations: a global model to fit and track 3d medical data. In *Proceedings Conference on Computer Vision, Virtual Reality and Robotics in Medecine (CVRMed)*, Nice, France, April 1995.

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

[12] A. Barr. Global and local deformations of solid primitives. *Computer Graphics (Proceedings SIGGRAPH'84)*, 18(3):21–30, July 1984.

[13] S. Benayoun, C. Nastar, and N. Ayache. Dense non-rigid motion estimation in sequences of 3d images using differential constraints. In *Proceedings Conference on Computer Vision, Virtual Reality and Robotics in Medecine (CVRMed)*, pages 309–318, Nice, France, April 1995.

[14] A. Blake and A. Zisserman. *Visual Reconstruction*. MIT Press, 1987.

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

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

[17] Y.K. Chang and A. Rockwood. A generalized de casteljau approach to 3d free-form deformation. *Computer Graphics (Proceedings SIGGRAPH'94)*, pages 257–260, July 1994.

[18] C. Chen, J. Luo, K. Parker, and T.S. Huang. CT volumetric data-based left ventricle motion estimation: an integrated approach. *International Journal of Computer Vision*, 11:85–100, October 1993.

[19] I. Cohen and L.D. Cohen. A hybrid hyperquadric model for 2-D and 3-D data fitting. In *Proceedings International Conference on Pattern Recognition (ICPR)*, pages B–403–405, Jerusalem, October 1994. Part of Inria TR 2188, to appear in *Computer Vision, Graphics, and Image Processing : Image Understanding*.

[20] I. Cohen, L.D. Cohen, and N. Ayache. Using deformable surfaces to segment 3-D images and infer differential structures. *Computer Vision, Graphics, and Image Processing: Image Understanding*, 56(2):242–263, September 1992.

[21] L.D. Cohen. On active contour models and balloons. *Computer Vision, Graphics, and Image Processing: Image Understanding*, 53(2):211–218, March 1991.

[22] L.D. Cohen. Auxiliary variables and two-step iterative algoritms in computer vision problems. Technical report, Ceremade, Février 1995. Cahiers de Mathématiques de la Décision 9511, to appear in Journal of Mathematical Imaging and Vision and Proceedings ICCV'95, Boston.

[23] 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(11), November 1993.

[24] S. Coquillart. Extended free-form deformation: A sculpturing tool for 3d geometric modeling. In *Computer Graphics (Proceedings SIGGRAPH'90)*, volume 24, pages 187–196, August 1990.

[25] S. Coquillart and P. Jancene. Animated free-form deformation: An interactive animation technique. In *Computer Graphics (Proceedings SIGGRAPH'91)*, volume 25, pages 23–26, July 1991.

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

50

[27] H. Delingette, M. Hebert, and K. Ikeuchi. Shape representation and image segmentation using deformable surfaces. In *Proceedings IEEE Computer Society Computer Vision and Pattern Recognition (CVPR)*, Maui, Hawai, June 1991.

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

[29] F. Ferrie, J. Lagarde, and P. Whaite. Recovery of volumetric object descriptions from laser rangefinder images. In *Proceedings European Conference on Computer Vision (ECCV)*, pages 387–396, Antibes, France, April 1990.

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

[31] J. Griessmair and W. Purgathofer. Deformation of solids with trivariate b-splines. In North-Holland Elsevier Science Publishers, editor, *Eurographics'89*, pages 187–193, 1989.

[32] W. E. L. Grimson. *From Images to Surfaces: A computational study of the Human Early vision system*. The MIT Press, 1981.

[33] A. Gupta and R. Bajcsy. An integrated approach for surface and volumetric segmentation of range images using biquadrics and superquadrics. *SPIE Applications of artificial intelligence X: Machine Vision and Robotics*, 1708:210–227, 1992.

[34] A. Gupta and R. Bajcsy. Volumetric segmentation of range images of 3d objects using superquadric models. *Computer Vision, Graphics, and Image Processing: Image Understanding*, 58(3):302–326, November 1993.

[35] S. Han, D. Goldgof, and K. Bowyer. Using hyperquadrics for shape recovery from range data. In *Proceedings IEEE International Conference on Computer Vision (ICCV)*, pages 492–496, Berlin, June 1993.

[36] K. Hohne and W. Hanson. Interactive 3d segmentation of mri and ct volumes using morphological operations. *Journal of Computer Assisted Tomography*, 16(2):285–294, March 1992.

[37] W. Hsu, J. Hughes, and H. Kaufman. Direct manipulation of free-form deformations. *Computer Graphics (Proceedings SIGGRAPH'92)*, 26(2):177–184, July 1992.

[38] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: active contour models. *International Journal of Computer Vision*, 1:321–331, 1987.

[39] H. Lamousin and W. Waggenspack. Nurbs-based free-form deformations. *IEEE Computer Graphics and Applications*, 14(6):59–65, November 1994.

[40] F. Leitner and P. Cinquin. Dynamic segmentation: Detecting complex topology 3D-object. In *Proceedings International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 295–296, Orlando, Florida, November 1991.

[41] D. Metaxas and S. Dickinson. Integration of quantitative and qualitative techniques for deformable model fitting from orthographic, perspective and stereo projections. In *Proceedings IEEE International Conference on Computer Vision (ICCV)*, pages 641–649, Berlin, Germany, May 1993.

[42] D. Metaxas and D. Terzopoulos. Shape and nonrigid motion estimation through physics-based synthesis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 580–591, June 1993.

[43] S.K. Mishra, D.M. Goldgof, and T.S. Huang. Motion analysis and epicardial deformation estimation from angiography data. In *Proceedings IEEE Computer Society Computer Vision and Pattern Recognition (CVPR)*, pages 331–336, Maui, Hawaii, June 1991.

[44] C. Nastar and N. Ayache. Spatio-temporal analysis of nonrigid motion from 4d data. In *Proceedings IEEE Workshop on non-rigid and articulated objects*, pages 146–151, Austin, November 1994.

[45] J. Park, Metaxas. D., and A. Young. Deformable models with parameter functions : application to heart-wall modeling. In *Proceedings IEEE Computer Society Computer Vision and Pattern Recognition (CVPR)*, pages 437–442, June 1994.

[46] S. Parry. *Free-form deformations in a constructive solid geometry modeling system*. PhD thesis, Department of Civil Engineering, Brigham Young University, April 1986.

[47] A. Pentland. Perceptual organization and the representation of natural form. *Artificial Intelligence*, 28:293–331, 1986.

52

[48] A. Pentland. Extraction of deformable part models. In *Proceedings European Conference on Computer Vision (ECCV)*, pages 397–401, Antibes, France, April 1990.

[49] 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.

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

[51] P. Shi, A. Amini, G. Robinson, A. Sinusas, C. Constable, and J. Duncan. Shape-based 4d left ventricular myocardial function analysis. In *Proceedings IEEE Workshop on Biomedical Image Analysis (WBIA)*, pages 88–97, Seattle, June 1994.

[52] F. Solina and R. Bajcsy. Range image interpretation of mail pieces with superquadrics. In *AAAI*, pages 733–737, Seattle, 1987.

[53] F. Solina and R. Bajcsy. Recovery of parametric models from range images : The case for superquadrics with global deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(2):131–147, February 1990.

[54] R. Szeliski and S. Lavallée. Matching 3d anatomical surfaces with non-rigid deformations. In *Proceedings SPIE Geometric Methods in Computer Vision II*, volume 2031, pages 306–315, San Diego, July 1993. Society of Photo-Optical Instrumentation Engineers.

[55] R. Szeliski and S. Lavallée. Matching 3-d anatomical surfaces with non-rigid deformations using octree-splines. In *Proceedings IEEE Workshop on Biomedical Image Analysis (WBIA)*, Seattle, Washington, June 1994.

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

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

[58] 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, July 1991.

[59] 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.

[60] A.L. Yuille, P.W. Hallinan, and D.S. Cohen. Feature extraction from faces using deformable templates. *International Journal of Computer Vision*, 8(2):99–111, September 1992.

[61] Z. Zhang. Iterative point matching for registration of free-form curves and surfaces. *International Journal of Computer Vision*, 13(2):119–152, 1994. Also Research Report No.1658, INRIA Sophia-Antipolis, 1992.