Ellipsoidal Skeleton for Multi-scaled Solid Reconstruction.

Frédéric Banégas Dominique Michelucci Marc Roelens Marc Jaeger

Ecole Nationale Supérieure des Mines de Saint Etienne
Centre de coopération Internationale en Recherche Agronomique pour le Développement
fbanegas@emse.fr or banegas@cirad.fr

January 5, 1999

Abstract

We present a robust method for automatically constructing an ellipsoidal skeleton (e-skeleton) from an unorganized set of 3D points. This skeleton will be essentially useful for dynamic visualization, manipulation and deformation. It also provides a good initial guess for surface reconstruction algorithms. On output of the entire process, we obtain an analytical description of a solid, semantically zoomable (local features only or reconstructed surfaces), with any level of detail (LOD) by discretization step control in voxel or polygon format. This capability allows us to handle objects at interactive framerates once the e-skeleton is computed. To ensure robustness and accuracy, all points sampled from a solid are taken into account, including the inner ones. Each e-skeleton is stored as a multi-scale CSG implicit tree. We propose a data structure to store along not only extracted features and geometry but also the hierarchy of shape refinement. Applications cover a wide range in computer graphics, from CAD to medical imaging.

Keywords: ellipsoidal skeleton, implicit surface, semantic zoom, level of detail, multi-scale tree.

1 Introduction.

3D digitized solids become more and more complex as acquisition devices increase in accuracy. Handling and analyzing such objects is computationally expensive. A potential approach for this issue would consist in filtering the huge amount of data by extracting only significant features for every semantic level.

Our investigations remained focused on these following aspects :

- The method has to be sufficiently robust in order to be used on a wide range of object.
- Feature Extraction must be progressive, so that relevant semantic zoom can be performed afterward.
- Ellipsoidal Skeleton must be steady and independant from rotation, translation or scaling for pattern matching purposes.
- The original object can be recovered by an error controlled fitting process, for data compression purposes.

We propose in this paper the model of E-skeleton, that could fulfill these requirements :

- No particular assumptions are made on the nature of the data and the order of the sample points.
- At first, the main shape aspect is highlighted, sharper substructures being progressively detected as the scale of vision is refined.
- Steadiness is guaranteed by taking into account the inner points of the solid. Locally extracted parameters (elongation, substructures orientation relatively to the global axis of inertia...) enrich the geometrical representation.
- Surface fitting procedures can be performed on the ellipsoidal skeleton, with intuitive error measurement.

While being scale dependent, E-skeleton is intented as a generic approach, aimed to describe global shapes.

2 Previous work.

Positioning primitives within a discretized object boundary is usually seen as a pre-processing stage that will eventually lead to surface fitting. Most of the existing methods do not exploit the information possibly extracted during this stage.

The first attempt to surface reconstruction, proposed by Boissonat [5, 6], consists in combining a Delaunay triangulation with its associated Voronoï diagram between each slice of the object in order to tetrahedrize its volume. However, this approach does not provide a synthetic structure of the reconstructed solid. Muraki [14] also uses Delaunay triangles to adequatly position implicit primitives that are smoothly combined analytically. A similar method, more focused on accuracy, has been proposed by Lim *et al.* [9]. Energy minimizing curves, like snakes [11, 12], provide a complete analytical description of the boundary, which is efficient in terms of data compression. The idea of a skeleton virtually carrying the topological structure of an object appears in [19] and [18] : it consists in axial lines (or splines) that will represent the backbone of more complex geometrical entities like generalized cylinders in this case. Applying global or local deformation to primitives as proposed by Miller et al. in [13] is also widely used at that time. Even recently, [8, 21] use superquadrics - which is similar from the method we propose in this paper - deforming them afterward in order to fit the contour, but do not offer incremental refining, smooth union of primitives nor structural information.

We found relevant to combine those various techniques while adding shape recognition features, as originally presented in [4], but instead of using a Medial Axis Transform Skeleton, which in practice is very sensitive to noise, we prefered to use the whole volume and to consider the point cloud as a material system in the mechanical sense. As proposed in [15], we integrated the concept of hierarchical skeleton.

3 The *ellipsoidal skeleton (e-skeleton)*.

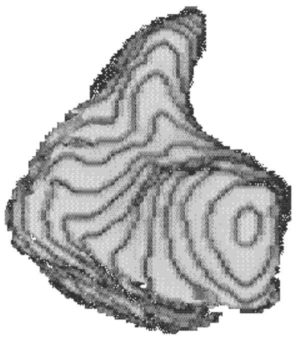
3.1 Principle.

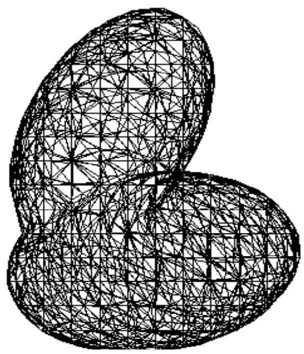
E-skeleton is aimed to highlight geometrical substructures of a solid. An automated splitting process subdivides the set of points into subsets until it reaches a given maximum number of classes, continuously storing hierarchical information and local measurements. Classes are detected according to their visual importance, using two intuitive criterions :

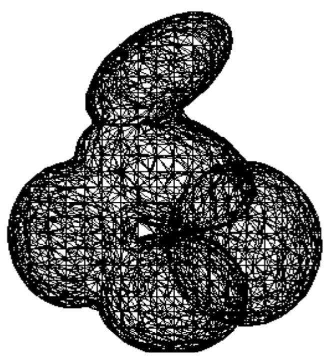
- A well differenciated shape from the overall structure (for a given vision scale) indicates that a class is a potential candidate.
- The volume or mass determines the order of detection in the splitting process.

Each primitive of the E-skeleton is a superellipsoid matching physical criterions extracted from the subset of points. It is attached to the subcloud and will be used as a basis for visualization, surface fitting and geometrical comparison.

Here are two levels of E-skeletons performed on a wrist bone, the *hamate*, with respectively 2 and 3 superellipsoids, the original voxelized object (inner points included) being shown first :







3.2 Describing a class.

Let C_j be a detected subset of the whole point set C at a given splitting stage. Each point of C being weighted to 1, a Principal Component Analysis (PCA) is performed, and D_{C_j}, the dispersion matrix (also called matrix of inertia), is obtained. Other parameters, such as volume or mass, can be computed as well.

3.3 Refining the set of classes.

Detecting relevant geometrical substructures requires to refine the e skeleton in a proper way. The purpose is to create k subclasses $C_1, ..., C_k$ from the main class C, assuming that :

$$\begin{cases} \sum_{j=1}^{k} n_j = n \\ \cup_{j=1}^{k} C_j = C \\ \cap_{j=1}^{k} C_j = \emptyset \end{cases} \quad (1)$$

where n_j is the number of points in C_j and n is the number of points contained in the global set C.

A possible approach consists in minimizing the intraclass variance, given the following property :

$$V_{intra} + V_{inter} = V_{init} \quad (2)$$

where V_{init} is :

$$V_{init} = var(X) + var(Y) + var(Z) \quad (3)$$

V_{init} is the global variance of the entire point set C (please refer to equation 6 for generic calculation of $var(X)$, $var(Y)$ and $var(Z)$).

As it is a constant value for a given solid, maximizing V_{intra} will minimize V_{inter}. In more intuitive words, we will obtain very homogeneous subsets while being very different from one to another.

The global intraclass variance is :

$$V_{intra} = \sum_{j=1}^{k} \frac{n_j V_j}{n} \quad (4)$$

where V_j is the variance (euclidean variance) of the subclass C_j, its expression being :

$$V_j = var(X_j) + var(Y_j) + var(Z_j) \quad (5)$$

with :

$$var(X_j) = \frac{\sum_{i=1}^{n_j} x_i^2}{n_j - 1} - \frac{\left(\sum_{i=1}^{n_j} x_i\right)^2}{n_j (n_j - 1)} \quad (6)$$

$var(Y_j)$ and $var(Z_j)$ are given by the same expression for respectively y and z coordinates of points belonging to C_j.

The classical formula of the global interclass variance is :

$$V_{inter} = \sum_{j=1}^{k} \frac{n_j \| G_j - G \| V_j}{n} \quad (7)$$

where G is the center of gravity of C.

The algorithm we implemented simulates the splittings of each existing class, and selects the one for which the lowering of V_{intra} is maximal. Some enhancements have been made in order to improve the efficiency of the method and will be discussed further.

Here is the refining algorithm :

```
While V_intra > V_threshold
  For Each C_j ⊂ S
      Calculation of main axis of inertia of C_j
      Calculation of intraclass variance V_j of C_j
      Rough splitting of C_j into C_j1 and C_j2
      Calculation of intraclass variance V_j1
      Calculation of intraclass variance V_j2
  EndForEach
  Real splitting of C_j with smallest (V_j1 + V_j2)/V_j
  Dynamic clustering on all existing classes
  Re calculation of V_intra with new set of classes
EndWhile
```

It starts with the whole set C ($k = 1$) and calculates $V_{threshold}$ with equation 4. Let us explain some of the major steps of the process described here.

3.3.1 Rough splitting along the main axis of inertia.

A former subdividing is done on an existing class C_j. Let κ'_j be the main axis of inertia of C_j, n_j the number of points of C_j and $p_i^j \in C_j$ with $j \in \{1, 2, ..., m\}$ (note that κ'_j is also the eigenvector attached to the greatest eigenvalue of D_{C_j}). If we calculate the mean value of the dot products :

$$\xi_j = \frac{1}{n_j} \sum_{i=1}^{n_j} \kappa'_j \cdot p_i^j \quad (8)$$

Then it becomes possible to split C_j into two subclasses C_{j_1} and C_{j_2} by selecting on one side the p_i^j verifying $\kappa'_j \cdot p_i^j > \xi_j$ and on the other side those for which $\kappa'_j \cdot p_i^j \leq \xi_j$ in 8.

Intuitively, if P_j is the plane containing G_j and the vectors κ''_j and κ'''_j (respectively the second and the third axis of inertia of C_j), we obtain two new classes C_{j_1} and C_{j_2} on both sides of P_j.

3.3.2 The Dynamic clustering method (DC).

Existing classes must be balanced after a splitting operation in order to optimally lower intraclass variance. This step is critical since it drastically improves not only the class differenciation but also their inner homogeneities. The DC algorithm is classically the following, as presented by E. Diday in [7] :

Repeat
  For Each existing class C_j
    Calculation of the center of gravity G_j of C_j
  EndForEach
  For Each $p_i \in C$
    Assign p_i to the class of the closer G_j
  EndForEach
Until neither of the $G_{\{j\}}$ changes

Note that for the dynamic clustering method to be optimal for intraclass variance minimization, the attaching criterion must be quadratic, which is fulfilled since we use an euclidean distance between every $p_i \in C$ and G_j. Assuming this, we found the method very robust (actually we didn't encounter any oscillation phenomenon at all), and very relevant in its results.

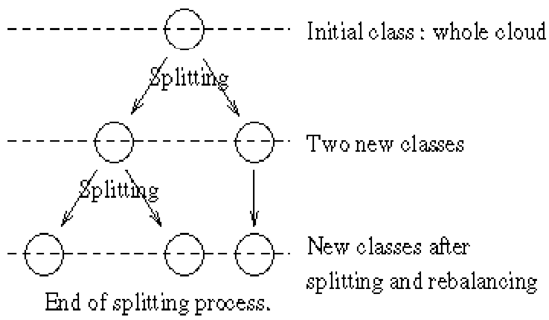
3.4 Hierarchical and geometrical data structure.

Data structure must combine the following aspects :

- Hierarchy of the whole subdividing process is kept.
- Each class is described at each splitting step, as parameters evolve during the subdividing process.

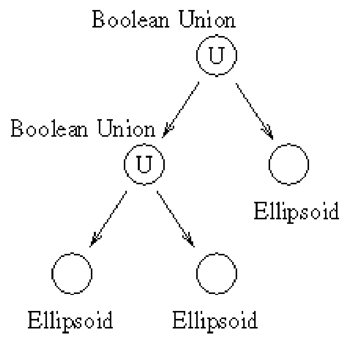
Keeping track of the subdividing steps allows not only to easily compare two distincts exams with multiscale capabilities, but also to rapidly refine or regroup a chosen area for semantic zoom purpose. We introduce the concept of *hierarchical tree*, aimed to store all information gathered during the splitting process independently from any representation format :

- Each node describes a class of points : locally extracted information are stored and can be retrieved instantaneously.
- The vertical structure, viewed as a binary tree, stores the subdivision hierarchy.
- Each horizontal level, viewed as a semantic level, provides a list of all existing classes at that step.



As we can see, each node has got zero, one or two sons, depending on the operation that has been performed.

From this tree and for a given semantic level, a geometrical instanciation can be immediately produced : an implicit CSG (Constructive Solid Geometry) tree is created, where each class is represented by a super ellipsoid. Here is a CSG tree corresponding to a particular semantic level :



- Leaves are the primitives created during the clusterization process (the ellipsoids in our case).
- Nodes carry boolean operations (usually unions) between those primitives.

This structure was partly motivated by the *function representation in geometric modelling* paradigm presented in [16]. It combines geometric construction with multiscale functionality as primitives can be refined as desired. Let us explain further the mathematical tools involved in the representation of the e-skeleton.

3.4.1 Primitive positioning.

The main parameters extracted during the partionning process for each subclass are the center of gravity and the three axis of inertia. As geometrical representative, we have chosen the superellipsoid, which has a strong physical meaning [20]. Its analytical expression is :

$$\left(\frac{x}{a}\right)^2 + \left(\frac{y}{b}\right)^2 + \left(\frac{z}{c}\right)^2 - 1 = 0 \quad (9)$$

where a, b and c respectively define the radii along local x, y and z axis of the superellipsoid.

Each superellipsoid is translated and orientated to fit the cloud it is attached to. A natural idea consists in positioning such a primitive at the center of gravity of every subset C_j, matching its three axis with the *eigenvectors* of D_{C_j}, and setting each radii to the corresponding eigenvector length. Such an ellipsoid is best known as *central inertia ellipsoid*.

3.4.2 The *implicit surface* model.

Equation 9 provides a description of our primitives. It could be anything we want, like a sphere as in [14]. Superellipsoids matches our needs as it allows anisotropy in three directions — matching the three inertia axis.

For any point $p(x,y,z)$ in 3D space, we obtain a distance $d(x,y,z)$ through 9. We then use a field formalism, as presented by J.F. Blinn in [2]. The potential emitted in space by a primitive defined with the distance function $d(x,y,z)$ is :

$$f(x,y,z) = \alpha\, e^{-\beta\, d(x,y,z)} \quad (10)$$

where α adjusts the strength of the field and β affects its speed decay. By seeking the points in space where $f(x,y,z)$ equals a constant c, we find what is called an *isosurface*, so the implicit equation of such an object becomes :

$$f(x,y,z) = c \quad (11)$$

The solid geometry feature is easily provided as inner points p_{int} of the objects verify $f(p_{int}) \leq c$ and outer points p_{ext} verify $f(p_{ext}) > c$.

In order to keep an homogeneous mathematical representation, we have chosen the model of *generalized implicit surfaces* presented in [17]. If M is the rotation matrix carrying the orientation of the three inertia axis and b is the position of the center of gravity, then the function $f(x,y,z)$ presented in 10 becomes :

$$g(X) = f(M^{-1}(X - b)) \quad (12)$$

where X is the position of the point in 3D space we are checking to determine wether he belongs to the isosurface or not. g is the representative function of the primitive once it has been properly positioned. We have also integrated the whole model by including the Barr deformation model described in [17] and [1]. It will allow us to deform each ellipsoid in order to match the points of each subcloud, as explained further in this paper. Adding this model to equation 12 leads to :

$$g(X) = f(\mathcal{T}_u\, M^{-1}(X - b)) \quad (13)$$

The *modal deformation* model and the significance of the vector u parameterizing the *modal deformation matrix* $\mathcal{T}_u$ are developed in [1] and [17].

3.4.3 Implicit union.

All the primitives within the e-skeleton must be unioned in order to reconstruct the whole object.

Let f_{e_1} and f_{e_2} be the field equations (cf. 10) of the two respective primitives e_1 and e_2. Modelling capabilities of implicit surfaces allow smooth union (expressed with the $\uplus$ symbol) by simply summing the equations of each involved primitive, hence the new equation of $e_1 \uplus e_2$ is :

$$f_{e_1 \uplus e_2} = f_{e_1} + f_{e_2} \quad (14)$$

With this method, a continuous surface (at least C^2) is provided, useful as an initial guess for any fitting algorithm and also as a relevant simplified geometrical instance of the object.

Exact union — also called boolean union — can also be performed by using the following equation :

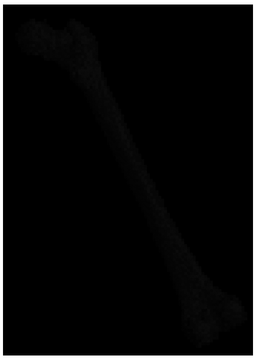
$$f_{e_1 \cup e_2} = max(f_{e_1}, f_{e_2}) \quad (15)$$

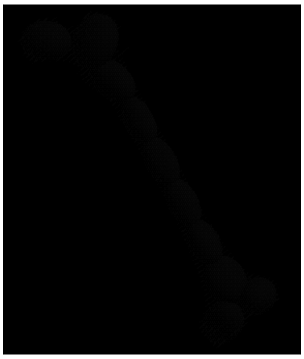
3.5 Examples, performance and applications.

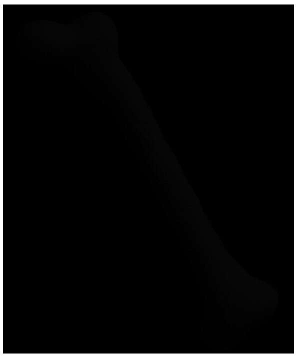
3.5.1 Examples of generated e-skeletons and performance.

The e-skeletons previously shown in this paper were created from a carpal bone — the *hamate*. Note that *all* voxels are taken into account, including the inner ones. The number of 3D points is 38,000 ; retrieval of most complex e-skeleton (9 subclasses) takes about 8 seconds including slice loading time on an Indy SGI with MIPS R4000 running at 100 Mhz. A Bloomenthal non adaptive polygonization [3] based on the *marching cubes* algorithm [10] has been performed to produce the presented meshes.

Second example shows a femur bone, composed of 150,000 3D points. Complete construction of the hierarchical tree took about 2 minutes on the previsously described Indy workstation (the maximum number of subclasses was arbitrarily set to 15). Mesh polygonization took less than 1 second. The original object is presented, followed by an e-skeleton composed of 10 subclasses, and then the same e-skeleton with smooth union, with a maximum error of 3 millimeters (without any surface fitting procedure) :







For sufficient number of points, *subsampling* revealed to be useful. While not affecting the construction of the e-skeleton, it allows faster computation times. The technique we have developed consists in lowering the spatial resolution, i.e. grouping centers of adjacent voxels into new ones.

On these examples, no specific optimization has been performed. We are now on profiling stage in order to accelerate our automated splitting process.

3.5.2 Applications of e-skeletons.

They essentially involve 3D reconstruction, data compression and medical imaging.

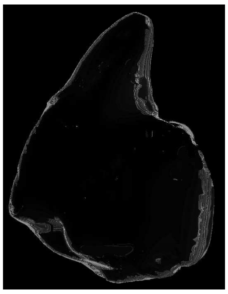
3D Reconstruction.

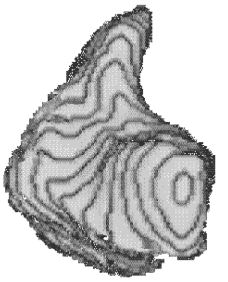
Smooth e-skeletons can be used as initial solutions for 3D reconstruction algorithms involving gradient search. We have implemented a classical Levenberg-Marquardt technique to fit the surface points extracted from scanner slices. The function to minimize for each point is the following :

$$h(x,y,z) = c - f_k(x,y,z)\, a_1^k a_2^k a_3^k \qquad (16)$$

Where f_k is the implicit function (cf. equation 11) provided by primitive e_k attached to the fitted cloud, and a_1^k, a_2^k, a_3^k are the three decreasing radii of e_k. (x,y,z) are the coordinates of the boundary point we try to fit with the implicit curve, and c is the isopotential value. The use of the radii in equation 16 helped in avoiding divergences. The main issue resides in the lack of robustness of the Levenberg-Marquardt method and problems occured with numerous solids. However, the goal was to show the potential answer of the e-skeleton to this type of problem, and at that time we did not focus on the efficiency of the gradient search method.

Here is an example of a reconstruction with the *hamate*, one of the carpal bones (wrist) :





We first show a 9 subclasses smoothed e-skeleton (with the blending equation (4)) in Gouraud shading that represents our inital guess and then the resulting reconstructed surface. Voxel representation is provided for visual comparison purpose. The fitting error, measured as the maximum distance between boundary points and the implicit surface, was less than 2 millimeters in this case.

With the *modal deformation* model from [17] integrated to every primitive the algorithm is the following :

> For Each existing class C_j
>   Extract boundary points of C_j
>   Find best deformation matrix with any
>   gradient search technique
> EndForEach

Once each primitive has been fitted to each local boundary, the implicit tree naturally combines them into a whole object. An error distance is computed as previously described, using geodetic distance from a point to the closer ellipsoid or simple euclidian distance from the polygonal reconstructed surface. Error measurement is not trivial in our case, as implicit surfaces provides only a potential value in space rather than intuitive distance, which is needed in our application.

It is important to note that the original object in voxel format can be recovered, when surface search succeeds. We are now investigating further in fitting techniques in order to find a more robust way to recover the surface. Others issues are :

- Objects with several non connected parts cannot be rendered easily as smoothed implicit surfaces do not provide topological information. A possible solution we implemented consists in monitoring the potential values returned during the discretization step, for each primitive in the CSG tree. Once this step is done, each primitive which has never returned a potential close to the iso-value c is discretized independently.
- Very fine details are difficult to obtain, as we decided not to use finite elements technique in order to keep a global and analytical representation. In our integration of the e-skeleton in medical software *Corpus 2000* (presented later in this paper), we switch to voxel representation when accurate surface rendering is required.
- Holes can appear in an e-skeleton wheras they do not really exist. Such holes are usually filled when smooth union is performed, as primitives are very close, but they can give wrong visual information on the object. This issue is currently under investigation.

3.5.3 Data Compression.

The hierarchical multiscaled CSG tree model offers a great amount of data compression. Typical ratio is in the area of about 1:1000 relatively to compressed slice images or medium quality polygonal representation. Deformation matrices, hierarchy and textual information are totally integrated to the data structure, allowing the whole recalculation of the reconstructed solid from a single ASCII file.

3.5.4 Medical applications.

The e-skeleton is well suited for medical imaging. It is now integrated into the CIRAD image processing and reconstruction software *Corpus 2000* as part of the *Modeling Biological Entities* project. The following direct applications include :

- Capture of organs from segmented NMR or TDM images. An e-skeleton is permanently computed with a low $V_{threshold}$ and stored on disk for later diagnosis or analysis purpose.
- Creation of anatomical atlas through various e-skeletons.

- Interactive deformable model (not done yet). Framerate is adjusted with combination of semantic zoom and discretization step control.
- Tooling of prosthesis using computer driven milling machine, based on a dense polygon mesh recovered from a leg stump.

4 Conclusion and future work.

We have presented a new automatic procedure to decompose a cloud of points, that produced encouraging results. Significant substructures are well detected as we lower the threshold of variance. By taking into account the inner points, the construction of the e-skeleton remains very steady and noise proof. A wide range of representations is provided, from orientation only to accurate surface representation, with simultaneous control of discretization step (LOD) and semantic level.

Further tests are due in order to adjust the behavior of the splitting process, especially for tubular sections of bones. We will also focus on the blending between primitives to ensure the accuracy of the whole union, and the implementation of a realtime deformable model. The generic aspects of the method must be kept to maintain performance and large application spectrum.

References

[1] A. Barr. Global and local deformations of solid primitives. In *Computer Graphics*, volume 18, pages 21–30. ACM Press, 1984.

[2] J. F. Blinn. A generalization of algebraic surface drawing. In *ACM Trans. on Graphics*, volume 1, pages 235–256. ACM Press, 1982.

[3] J. Bloomenthal. Polygonization of implicit surfaces. Technical Report CSL-87-2, XEROX PARC, May 1987.

[4] H. Blum. A transformation for extracting new descriptions of shape. In *MPSVF*, pages 362–380, 1967.

[5] J. D. Boissonnat. Geometric structures for three dimensional shape representation. In *ACM Transactions on Graphics*, volume 3, pages 266–286. ACM Press, 1984.

[6] J. D. Boissonnat. Shape reconstruction from planar cross sections. In *Computer Vision, Graphics and Image Processing*, 44, 1988.

[7] E. Diday. Une nouvelle méthode en classification automatique et reconnaissance des formes : la méthode des nuées dynamiques. In *Rev. Statist. Appl.*, volume 19, pages 19–33, 1971.

[8] A. Leonardis, A. Jaklic, and F. Solina. Superquadrics for segmenting and modeling range data. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 19, pages 1289–1295. IEEE, November 1997.

[9] Chek T. Lim, George M. Turkiyyah, Mark A. Ganter, and Duane W. Storti. Implicit reconstruction of solids from cloud point sets. In *Solid Modelling '95*, pages 393–402. ACM Press, 1995.

[10] W. E. Lorensen and H. E. Cline. Marching cubes : A high resolution 3d surface algorithm. In *Computer Graphics*, volume 21, pages 163–169. ACM Press, July 1987.

[11] T. McInerney and D. Terzopoulos. Topologically adaptable snakes. In *Proc. of the Fifth Int. Conf. on Computer Vision (ICCV'95)*, pages 840–845. IEEE Computer Society Press, June 1995.

[12] T. McInerney and D. Terzopoulos. Medical image segmentation using topologically adaptable surfaces. In *First Joint Conference on Computer Vision, Virtual Reality and Robotics in Medicine (CVRMed-MRCAS'97)*, pages 23–32. J. Troccaz, E. Grimson, R. Mosges Editor, March 1997.

[13] J. V. Miller et al. Geometrically deformed models : a method for extracting closed geometric models from volume data. In *Computer Graphics*, volume 25, pages 217–225. ACM Press, 1991.

[14] S. Muraki. Volumetric shape description of range data using "blobby model". In *Computer Graphics*, volume 25, pages 227–235. ACM Press, July 1991.

[15] M. Näf, O. Kübler, R. Kikinis, M.E. Shenton, and G. Székely. Characterization and recognition of 3d organ shape in medical image analysis using skeletonization. In *IEEE Workshop on Mathematical Methods in Biomedical Image Analysis*. IEEE, June 1996.

[16] A. Pasko, V. Adzhiev, A. Sourin, and V. Savchenko. Function representation in geometric modelling : concepts, implementation and applications. In *The Visual Computer*, pages 429–446. Springer Verlag, 1995.

[17] S. Sclaroff and A. Pentland. Generalized implicit functions for computer graphics. In *Computer Graphics*, volume 25, pages 247–250. ACM Press, July 1991.

[18] U. Shani and B. Ballard. Splines as embeddings for generalized cylinders. In *Computer Vision, Graphics and Image Processing*, volume 27, August 1984.

[19] B. Soroka, R. Andersson, and R. Bajcsy. Generalized cylinders from local aggregation of sections. In *Pattern Recognition*, volume 13, 1981.

[20] M. R. Spiegel. *Théorie et application de la Mécanique générale*. Mc Graw Hill, 1983.

[21] R. Durikovic, K. Kaneda, and H. Yamashita. Reconstructing a 3d structure with multiple deformable solid primitives. In *Computer & Graphics*, volume 21, pages 611–622. Pergamon, 1997.