Hierarchical automated clustering of cloud point set by ellipsoidal skeleton. Application to organ geometric modeling from CT-scan images.

Frédéric Banégas[*]
Dominique Michelucci
Marc Roelens
Ecole Nationale Supérieure des Mines de Saint-Etienne
Marc Jaeger
Centre de coopération Internationale
en Recherche Agronomique pour le Développement

ABSTRACT

We present a robust method for automatically constructing an *ellipsoidal skeleton (e-skeleton)* from a set of 3D points taken from NMR or TDM images. To ensure steadiness and accuracy, all points of the objects are taken into account, including the inner ones, which is different from the existing techniques. This skeleton will be essentially useful for object characterization, for comparisons between various measurements and as a basis for deformable models. It also provides good initial guess for surface reconstruction algorithms. On output of the entire process, we obtain an analytical description of the chosen entity, 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 frame rates once the e-skeleton is computed. Each e-skeleton is stored as a multiscale CSG implicit tree.

Keywords : ellipsoidal skeleton, implicit surface, inertia momentum, semantic zoom, LOD, CSG, characterization, multiscale tree.

1. PREVIOUS WORK.

Reconstructing from a set of 3D points is classical in applications ranging from medical imaging to CAD. Several techniques exists, using triangulation methods, computing a skeleton or positioning and adjusting 3D primitives inside the contour points.

Boissonat[Boi84,Boi88] uses Delaunay triangulation of the set of points. D. Attali *et al.*[ABM94] use a skeleton derived from the Voronoï graph. Generalized cylinders can also be used, as in,[SAB81,BRLS96] carried by splines[SB84] that define an axial skeleton of the point cloud. Analytical expressions are present in the *snake* model[MT95,MT97] that consists in energy-minimizing curves. Muraki[Mur91] uses implicit primitives positioned according to a triangulation of each slice and adjusts them to fit the boundary. A similar method, more focused on accuracy, has been proposed by Lim *et al.*.[LTGS95] Applying global or local deformation to primitives as proposed by Miller *et al.* in[M+91] is also widely used at that time. Even recently,[LJS97,uKY97] use superquadrics – which is similar from the method we propose in this paper – deforming them afterward in order to fit the contour.

All those techniques are focused on surface reconstruction, which is only one of the capability of our representation of solid objects.

[*]banegas@cirad.fr or fbanegas@emse.fr

2. THE *ELLIPSOIDAL SKELETON (E-SKELETON)*.

2.1. Principle.

Due to their coarseness, unorganized points sets are often difficult to analyze in terms of features extraction, especially in three dimensions.

For instance, axis of inertia are sometimes occulted by dispersion, noise or even errors during the acquisition process. The main issue consists in creating subsets — visible as single objects — that exhibits remarkable features of the whole set. Those objects should contain locally helpful informations to ensure any kind of guidance for any type of exploration. Accurate reconstruction of organs often leads to huge complexity and computation cost. The great data flow in this situation must be context-adaptive : the entirety of data is not always required. Taking into account only relevant features is better in terms of performance and understanding, leading to the concept of *semantic zoom*.

The *e-skeleton* is intended to provide synthetic geometric representation of the whole set depending on the study context while remaining independent from scale and rotation. Direct applications using such a model could be the following ones :

- quantization of the geometric dispersion in the area of observation ;
- recognition and comparison of points sets representing the same object throughout different acquisition ;
- visualization of global and then more and more local structures by increasing the e-skeleton level of detail (LOD) ;
- data compression, allowing quick transmission on low-bandwith networks.

For now we have successfully implemented all these features, which will be discussed further in this paper.

2.2. The basic element : the *e-bone*.

The simplest element — called the *e-bone* (standing for *ellipsoidal bone*) — consistuent of the e-skeleton, is a superquadric, which is defined as in[SP91] by the following equation :

$$\left[\left(\frac{x}{a}\right)^{\frac{2}{\varepsilon_2}} + \left(\frac{y}{b}\right)^{\frac{2}{\varepsilon_2}}\right]^{\frac{\varepsilon_2}{\varepsilon_1}} + \left(\frac{z}{c}\right)^{\frac{2}{\varepsilon_1}} - 1 = 0 \tag{1}$$

where a, b and c respectively define the radii along local x, y and z axis and $\varepsilon_1, \varepsilon_2$ the squareness of the ellipsoid.

What is important in 1 is the generic aspect of the formula, but for common purpose we will assume that $\varepsilon_1 = \varepsilon_2 = 1$, limiting primitives to ellipsoids. As we will see, this choice of shape has a strong physical meaning.

2.3. Positioning the *e-bone*.

In order for the *e-bone* to become a good representative element of a given subset of n points $\mathcal{S}$, a Principal Component Analysis (PCA) is performed.

A weight w_i is assigned to each of the points $p_i(x_i, y_i, z_i, w_i) \in \mathcal{S}$, including the inner ones. For example, to obtain a purely geometric representation, all w_i can be set to the same magnitude (usually to 1). It could be also valuable to weight a given p_i with its associated gray level (e.g. its density for a CT scan) so that one can easily visualize the mass distribution of the observed object.

The spreading of points of $\mathcal{S}$ represents the geometry of the measured object. This naturally leads to the use of the so-called *dispersion matrix* $D_{\mathcal{S}}$ of $\mathcal{S}$, defined as follow in three dimensions :

$$D_{\mathcal{S}} = \begin{pmatrix} var(x) & covar(y,x) & covar(z,x) \\ covar(x,y) & var(y) & covar(z,y) \\ covar(x,z) & covar(y,z) & var(z) \end{pmatrix} \tag{2}$$

where *var* and *covar* mean respectively variance and covariance along global associated axis. For the variance along the x axis we have :

$$var(x) = \sum_{i=1}^{n} (x_i - \overline{x})^2 \; w_i \tag{3}$$

and for the covariance along x and y axis :

$$covar(x, y) = \sum_{i=1}^{n} (x_i - \overline{x}) \; (y_i - \overline{y}) \; h(x_i, y_i) \tag{4}$$

where $h_i(x_i, y_i)$ is the classical *joint distribution function* for given x_i and y_i, and $\overline{x}$ is the mean along x axis. Note that since $covar(x, y) = covar(y, x)$, the matrix $\mathbf{D}_{\mathcal{S}}$ is symetrical. Another interesting feature is that $\mathbf{D}_{\mathcal{S}}$ can be computed incrementally, due to the following equation :

$$var(x) = \sum_{i=1}^{n} x_i^2 \; w_i - \frac{\left(\sum_{i=1}^{n} x_i \; w_i\right)^2}{n} \tag{5}$$

The calculus of the *eigenvectors* associated with the decreasing *eigenvalues* of $\mathbf{D}_{\mathcal{S}}$ provides three axis of inertia of decreasing significance (the greatest *eigenvalue* is attached to the main axis of inertia, the middle-ranged *eigenvalue* to the second one and so on...).

A natural idea consists in positioning an *e-bone* at the center of gravity of the subset, oriented by matching his axis with the *eigenvectors*, and with radii equal to their respective lengths. A *central inertia ellipsoid* is obtained.

2.4. Refining the e-skeleton.

Isolating local geometrical structures of $\mathcal{S}$ requires to refine the e-skeleton in a proper way. We apply the concept of scale-dependent view : for a desired accuracy, subdivisions are made on $\mathcal{S}$, and each new subset is attached to a new *e-bone*.

The purpose is to create k subclasses $\mathcal{C}_1, ..., \mathcal{C}_k$ from the main class $\mathcal{S}$. One method consists in minimizing the intra-class variance while maximizing the inter-class variance, according to the following property :

$$V_{intra} + V_{inter} = V_{init} \tag{6}$$

where V_{init} is the initial euclidian variance of the entire set of points. In other words, we want very homogeneous subclasses while being very different from one to another.

The sum of the intraclass variances of all existing classes $\mathcal{C}_1, ..., \mathcal{C}_k$ is :

$$V_{intra} = \sum_{j=1}^{k} \frac{n_j \; V_j}{n} \tag{7}$$

where n_j is the number of points contained in $\mathcal{C}_j$, V_j is the variance of the subclass $\mathcal{C}_j$ relatively to its center of gravity G_j, and n is the number of points contained in $\mathcal{S}$.

The classical formula of the global interclass variance is :

$$V_{inter} = \sum_{j=1}^{k} \frac{n_j \; \| \; G_j - G \; \| \; V_j}{n} \tag{8}$$

where G is the center of gravity of $\mathcal{S}$.

Here is the refining algorithm :

```
While $V_{intra} > V_{threshold}$
      For Each $\mathcal{C}_j \subset \mathcal{S}$
          Calculation of main axis of inertia of $\mathcal{C}_j$
          Calculation of intra-class variance $V_j$ of $\mathcal{C}_j$
          Rough splitting of $\mathcal{C}_j$ into $\mathcal{C}_{j_1}$ and $\mathcal{C}_{j_2}$
          Calculation of intra-class variance $V_{j_1}$
          Calculation of intra-class variance $V_{j_2}$
          Calculation of $r_j = \frac{V_{j_1} + V_{j_2}}{V_j}$
      EndForEach
      Real splitting of $\mathcal{C}_j$ having the smallest $r_j$
      Dynamic clustering on all existing classes
      Re-calculation of $V_{intra}$ with new set of classes
EndWhile
```

The initialization consists in setting k to 1, and calculating $V_{threshold}$ with equation 7 on $\mathcal{S}$ relatively to G. Let us explain some of the major steps of the process described here.

2.4.1. Rough splitting along the main axis of inertia

A former subdividing is done on an existing class $\mathcal{C}_j$. Let $\vec{\kappa}'_j$ be the main axis of inertia of $\mathcal{C}_j$, and $p_i^j \in \mathcal{C}_j$ with $j \in \{1, 2, ..., m\}$ – note that $\vec{\kappa}'_j$ is also the *eigenvector* attached to the greatest *eigenvalue* of $\mathbf{D}_{\mathcal{S}}$. If we calculate the mean value of the dot products :

$$\xi_i = \frac{1}{n} \sum_{i=1}^{n} \vec{\kappa}'_j \, \vec{p}\,_i^j \tag{9}$$

Then it becomes possible to split $\mathcal{C}_j$ into two subclasses $\mathcal{C}_{j_1}$ and $\mathcal{C}_{j_2}$ by selecting on one side the p_i^j verifying $\vec{\kappa}'_j \, \vec{p}\,_i^j > \xi_i$ and on the other side those for which $\vec{\kappa}'_j \, \vec{p}\,_i^j \leq \xi_i$ in 9.

Intuitively, if P_j is the plane containing G_j and the vectors $\vec{\kappa}''_j$ and $\vec{\kappa}'''_j$ — respectively the second and the third axis of inertia of $\mathcal{C}_j$ — we obtain two new classes $\mathcal{C}_{j_1}$ and $\mathcal{C}_{j_2}$ on both sides of P_j.

2.4.2. The use of r_j

Using simply the class that best lowers $V_{thresold}$ as a real-splitting criterion can work fine but we have chosen a slightly more accurate method.

Let us imagine we have a cloud of points that lies into three ellipsoids, a big one and two little ones, very close (we assume that the dispersion in space is homogeneous and uniform, which is the case in practice). We admit we have two classes at that iteration : the first is the big ellipsoid, the second is composed of the two others. If the splitting criterion is the best lowering of $V_{thresold}$, then it is very likely the algorithm will split the big ellipsoid. This may result in a significant loss of relevancy for the e-skeleton, as we would like the two small ellipsoids to be refined at first.

Calculating the ratio $r_j = \frac{V_{j_1} + V_{j_2}}{V_j}$ independent from the scale, is better as it transforms the value r_j into a normalized one metric.

2.4.3. The Dynamic clusters 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[Did71] :

```
Repeat
       For Each existing class $\mathcal{C}_j$
           Calculation of the center of gravity $G_j$ of $\mathcal{C}_j$
       EndForEach
       For Each $p_i \in \mathcal{S}$
           Assign $p_i$ to the class of which $G_j$ is closer
       EndForEach
Until neither of the $G_{\{1,\ldots\}}$ changes
```

Oscillations may happen in some case but can be easily avoided — simply by setting an upper limit to the number of iterations or by monitoring the changing points when there are small moves — and occur rarely in practice.

Note that for the dynamic clusters method to be optimal for intra-class variance minimization, the attaching criterion must be quadratic, which is fulfilled since we use an euclidian metric. Assuming this, we found the method very robust (actually we didn't encounter any oscillation phenomenon at all), and very relevant in its results.

2.5. Considering and visualizing the e-skeleton.

Once each cluster has been attached to an ellipsoid, it is useful to synthetize these new data. A tree is created, carrying geometrical and hierarchical informations.

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.

The structure of the tree is the following :

- leaves are the primitives created during the clusterization process – the ellipsoids in our case ;
- nodes carry boolean operations – usually unions – between those primitives. Combinations of ellipsoids are handled using Constructive Solid Geometry (CSG) model with implicit primitives. Information about the class that produced the two children subclasses is also encoded.

This structure was partly motivated by the *function representation in geometric modelling* paradigm, presented in.[PASS95] It combines geo-metric 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.

2.5.1. Implicit surfaces model.

Equation 1 provides a description of our primitives. It could be anything we want, like a sphere as in.[Mur91] Superquadric matches our needs as it combines the genericity – ranging from sphere to cube – with the 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 1. We then use a field formalism, as presented by J.F. Blinn in.[Bli82] A primitive described by the distance function $d(x, y, z)$ is given by :

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

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 :

$$f(x, y, z) = c \tag{11}$$

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 $f(p_{ext}) > c$.

2.5.2. Primitive positioning.

The four parameters extracted during the partionning process for each subclass are the center of gravity and the three axis of inertia. Each ellipsoid needs to be translated and oriented to fit the cloud it is attached to. In order to keep an homogeneous mathematical representation, we have chosen the model of *generalized implicit surfaces* presented in.[SP91] If $\mathbf{M}$ is the rotation matrix carrying the orientation of the three inertia axis and $\mathbf{b}$ is the position of the center of gravity, then the function $f(x, y, z)$ presented in 10 becomes :

$$g(\mathbf{x}) = f(\mathbf{M}^{-1}\,(\hat{\mathbf{x}} - \mathbf{b})) \tag{12}$$

where $\hat{\mathbf{x}}$ is the position of the point in 3D space we are checking to ensure wether or not he belongs to the isosurface. 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[SP91] and.[Bar84] 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 :

$$g(\mathbf{x}) = f(\mathcal{D}_{\mathbf{u}}\,\mathbf{M}^{-1}\,(\hat{\mathbf{x}} - \mathbf{b})) \tag{13}$$

The *modal deformation* model and the significance of the vector $\mathbf{u}$ parameterizing the *modal deformation matrix* $\mathcal{D}_{\mathbf{u}}$ are developed in[Bar84] and.[SP91]

2.5.3. Implicit union.

All the primitives constituing the e-skeleton must be unioned in order to represent the whole objet.

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 adding the equations of each involved primitive, hence the new equation of $e_1 \uplus e_2$ is :

$$f_{e_1 \uplus e_2} = f_{e_1} + f_{e_2} \tag{14}$$

With this method, a smooth union is provided, providing a good initial solution for any fitting algorithm and also a relevant simplified visualization of the whole reconstructed object. The resulting surface has the property of being at least $\mathcal{C}^2$ continuous.

Boolean union can also be performed by using the following equation :

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

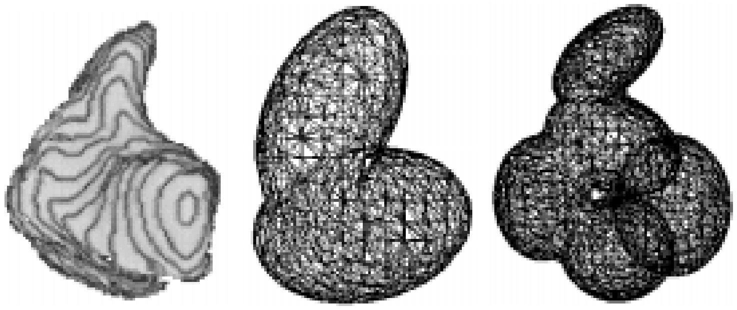
By this way it becomes possible to efficiently analyze and visualize local matter orientation and dispersion.

2.5.4. Global or local analytical description.

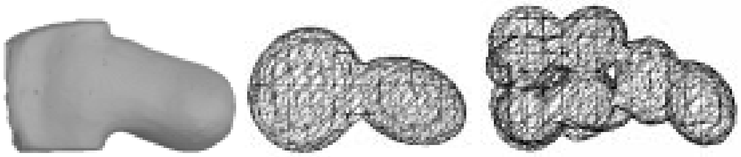
Each node of boolean union is a potential root. It is then possible to work on selected portions of the e-skeleton, roots providing the equations of each local substructure. We have added the capability for a node to be activated or not, simplifying the task of region selection. From each tree, a completely new one can be generated by suppressing the inactive sections. It provides the characterization map of only a small portion of a more complicated set of structures, while allowing the entire recovery of the whole model at any moment.

2.6. Examples, performance and applications.

2.6.1. Examples of generated e-skeletons and performance



This first example shows various e-skeletons created from a carpal bone – the *hamate*. The left view is the voxel object reconstructed from scanner slices. Contrast has been enhanced to highlight altitude lines. Note that *all* voxels are taken into account, including the inner ones. The other two pictures shows e-skeletons with respectively two and five subclasses. The number of 3D points is 38,000 ; retrieval of extreme right e-skeleton has taken about 8 seconds including slice loading on an Indy SGI with MIPS R4000 running at 100 Mhz. A Bloomenthal non-adaptive polygonization[Blo87] based on the *marching cubes* algorithm[LC87] has been made to produce the presented meshes.



Second example shows a leg stump containing 396,800 3D points, presented in voxel format to the left. E-skeletons to the right, composed respectively of two and nine subclasses, were produced in about 2 and 10 minutes on the previously described SGI workstation, including slice loading.

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 (carrying for example the mean of their densities). Prior smoothing filters can be applied for the subsampling to be valid.

No optimization have been done yet, but we think calculation time can be greatly shortened. We are now on profiling stage in order to accelerate our automated splitting process. We planned to develop a complete 3D reconstruction and interaction chain.

2.6.2. Applications of e-skeletons

They essentially involve comparisons of organs through different acquisitions and 3D reconstruction.

Comparisons

The combination of steadiness and hierarchy allows object comparison. From one medical exam to another, variations in the acquisition process are to be expected. Orientation as well as scanner resolution or even scale of the object may vary in time. The use of an orientation, scale and noise independent technique is required here.

The first and classical type of comparison we have implemented is a purely geometric one. We use class-by-class comparison – thanks to the steadiness of the splitting algorithm. What is important to notice is that only primary axis of inertia have to be analyzed. Secondary and tertiary ones are too unsteady to be relevant. They remain in the plane orthogonal to the primary axis, but tend to rotate in an unpredictable way. We have chosen to extract the orientations of the subclasses relatively to the primary axis of the entire object. Sizes are compared in terms of ratio : the three axis lengthes of each subclass (i.e. their associated eigenvalues) are divided by those of the whole cloud. Elongation factors can also be computed, by evaluating the ratio $\frac{r_{max}}{r_{min}}$ where r_{max} and r_{min} are respectively the greatest and the lowest radii.

The second method consists in comparing two trees with the above mentioned technique at equivalent subdividing steps using the multiscale capability. Starting with the two global ellipsoids and refining step by step has two major advantages :

- the splitting process can be stopped as soon as sufficient difference is found or until a maximum number of subclasses is reached ;
- area of difference can be immediately detected.

A simple metric is provided by keeping the number of splitting steps done before difference occurs.

Third method involves measuring distances between deformation values as in[MPSK98] after 3D fitting process of the various exams, which is directly applicable with e-skeletons.

Those comparison techniques are relevant as long as the number of point-per-class is sufficient. Without any control – especially on the validity of the dispersion matrix – it is obvious that inertia information extracted from the cloud of points will become completely aberrant. Our tests have been made on 5 CT-scan exams of carpal bones, but we plan to validate them on various other exams, under physician supervision.

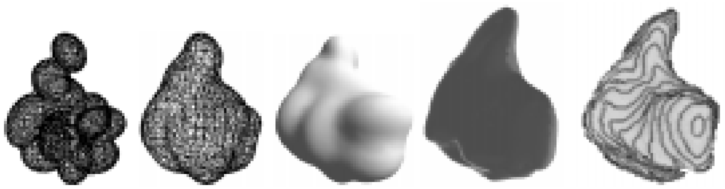
3D Reconstruction

Smooth e-skeletons can be used as initial solutions for 3D reconstruction algorithms involving gradient search. We have implemented a classical Levenberg-Marquardt method that converged in an average of 3 iterations per subclass in the tests we performed. The function to minimize in each point is the following :

$$g(x,y,z) = c - f_k^2(x,y,z)\, a_1^k\, a_2^k\, a_3^k \tag{16}$$

Where f_k is the implicit function (cf. equation 11) provided by primitive $\mathcal{E}_k$ attached to the fitted cloud, and a_1^k, a_2^k, a_3^k are the three decreasing radii of $\mathcal{E}_k$. (x,y,z) are the coordinates of the boundary point we try to fit with the implicit curve, and c is the isopotential value.

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



From left to right we show a 9 subclasses e-skeleton (boolean union performed with equation 15), a smoothed one (with equation 14) in wireframe and Gouraud shading, and finally the resulting reconstructed surface. Voxel representation is provided to the right for visual comparison purpose.

We have integrated the modal deformation model of[SP91] to every primitive in order to enhance local control withing each detect substructure. Our algorithm is the following :

```
For Each existing class $\mathcal{C}_j$
    Extract boundary points of $\mathcal{C}_j$
    Find best deformation matrix with Levenberg-Marquardt
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 using distance calculations and goodness-of-fit tests like as described in.[DS98] In our case, boundary points are recovered using 6-neighborhood topology in 3D space with slice images.

It is important to note that the whole initial object in voxel format can be recovered by applying the *marching cube*[LC87] algorithm to the implicit CSG tree. Those voxels can be used to quantize the error of reconstruction from the original organ.

2.6.3. Medical applications.

The e-skeleton is well suited for medical imaging. We are now working on its integration into the CIRAD (*Centre de coopération Internationale en Recherche Agronomique pour le Développement*) medical software *Corpus 2000* as part of the *Modeling Biological Entities* project. The following direct applications are :

- Capture of organs from segmented NMR or TDM images. An e-skeleton is permanently computed with a low $V_{threshold}$ and can be used later for diagnostic or analysis purposes. Each element will then have his own tree.
- Creation of an anatomical atlas through various e-skeletons. This organ library have many useful goals :
  - Organs comparisons, pathologies detections and following of growth behavior.
  - Generation of synthetic organs for educational purpose or recovery of another ones by halometry.
- Implementation of interactive application with deformable model. Frame rate will be adjusted with combination of semantic zoom and discretization step control.

2.7. Conclusion and future work.

We have presented a new automatic procedure to decompose a cloud of points, that produced very encouraging results. Significant substructures are well detected as we lower the threshold of variance. Problems can be easily detected and corrected. By taking into account the inner points, the construction of the e-skeleton remains very steady at reasonable threshold (i.e. not to low to ensure sufficient number of points-per-class), and noise-resistant. A wide range of representations is provided, from orientation-only to accurate surface representation, with simultaneous control of discretization step (LOD).

Further tests are due in order to adjust the behavior of the splitting process with highly concave objects and validate the characterization and comparison stage with physicians. We will also focus on the blending between primitives to ensure the accuracy of the whole union.

Applications involve medical diagnosis and surface reconstruction assistance. We are now integrating the whole chain from e-skeleton to realistic visualization into an existing application. It will provide physicians with a useful tool for pathology detection and pre-visualization of the interior of the body with semantic zoom and LOD-controlled view. We hope to extend further the concept of e-skeleton during this implantation.

REFERENCES

[ABM94] D. Attali, P. Bertolino, and A. Montanvert. Using polyballs to approximate shapes and skeletons, 1994.

[ACC+96] N. Ayache, P. Cinquin, I. Cohen, L. Cohen, F. Leitner, and O. Monga. Segmentation of complex three-dimensional medical objects: A challenge for computer-assisted surgery planning and performance. In *Computer-Integrated Surgery, Technology and Clinical Applications.*, pages 59–74. MIT Press, 1996.

[Bar84] A. Barr. Global and local deformations of solid primitives. In *Computer Graphics*, volume 18, pages 21–30. ACM Press, 1984.

[Bli82] J. F. Blinn. A generalization of algebraic surface drawing. In *ACM Trans. on Graphics*, volume 1, pages 235–256. ACM Press, 1982.

[Blo87] J. Bloomenthal. Polygonization of implicit surfaces. Technical Report CSL-87-2, XEROX PARC, May 1987.

[BM94] I. Bricault and O. Monga. From volume medical imaging to quadratic surface patches. Technical Report 2380, September 1994.

[Boi84] J. D. Boissonnat. Geometric structures for three-dimensional shape representation. In *ACM Transactions on Graphics*, volume 3, pages 266–286. ACM Press, 1984.

[Boi88] J. D. Boissonnat. Shape reconstruction from planar cross sections. In *Computer Vision, Graphics and Image Processing*, 44, 1988.

[BRLS96] V. Burdin, C. Roux, C. Lefèvre, and E. Stindel. Modeling and analysis of 3-d elongated shapes with application to long bone morphometry. In *IEEE Transactions on Medical Imaging*, volume 15, pages 79–91. IEEE, February 1996.

[Coh91] L. D. Cohen. On active contour models and balloons. In *Comput. Vision Graphics Image Proc.*, volume 53, pages 211–218, 1991.

[Did71] 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.

[DS98] G. Danuser and M. Stricker. Parametric model fitting : From inlier characterization to outlier detection. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 20, pages 263–280. IEEE, March 1998.

[Duf92] T. Duff. Interval arithmetic and recursive subdivision for implicit functions and constructive solid geometry. In *Computer Graphics*, volume 26, pages 131–138. ACM Press, July 1992.

[GMW] P. E. Gill, W. Murray, and M. H. Wright. *Practical Optimization.* Academic Press.

[H$^+$92] H. Hoppe et al. Surface reconstruction from unorganized points. In *Computer Graphics*, volume 26, pages 71–78. ACM Press, July 1992.

[KF87] P. Kamina and J-P. Francke. *Anatomie, Introduction à la Clinique.* Maloine, 1987.

[KR90] L. Kaufman and P. J. Rousseeuw. *Finding Groups in Data.* Wiley Interscience, 1990.

[LC87] 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.

[LJS97] A. Leonardis, A. Jaklič, 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.

[LTGS95] 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.

[M$^+$91] 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.

[MPSK98] J. Martin, A. Pentland, S. Sclaroff, and R. Kikinis. Characterization of neuropathological shape deformations. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 20, pages 97–112. IEEE, February 1998.

[MT95] 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.

[MT97] 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.

[Mur91] S. Muraki. Volumetric shape description of range data using "blobby model". In *Computer Graphics*, volume 25, pages 227–235. ACM Press, July 1991.

[PASS95] 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.

[Rho97] M. L. Rhodes. Computer graphics and medicine : A complex partnership. In *IEEE Computer Graphics and Applications*, pages 22–28. IEEE, January-February 1997.

[SAB81] B. Soroka, R. Andersson, and R. Bajcsy. Generalized cylinders from local aggregation of sections. In *Pattern Recognition*, volume 13, 1981.

[SB84] U. Shani and B. Ballard. Splines as embeddings for generalized cylinders. In *Computer Vision, Graphics and Image Processing*, volume 27, August 1984.

[SP91] S. Sclaroff and A. Pentland. Generalized implicit functions for computer graphics. In *Computer Graphics*, volume 25, pages 247–250. ACM Press, July 1991.

[Spi83] M. R. Spiegel. *Théorie et application de la Mécanique générale.* Mc Graw Hill, 1983.

[SPOK95] V. V. Savchenko, A.A. Pasko, O. G. Okunev, and T. L. Kunii. Function representation of solids reconstructed from scattered surface points and contours. In *Eurographics '95.* Blackwell Publishers, 1995.

[Tau91] G. Taubin. Estimation of planar curves, surfaces and nonplanar space curves defined by implicit equations with applications to edge and range segmentation. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 13, pages 1115–1138. IEEE, November 1991.

[uKY97] R. Ďurikovič, K. Kaneda, and H. Yamashita. Reconstructing a 3d structure with multiple deformable solid primitives. In *Computer & Graphics*, volume 21, pages 611–622. Pergamon, 1997.

[VL97] A. Verroust and F. Lazarus. Extracting skeletal curves from 3d scattered data. Technical Report ISSN-0249-6399, INRIA, September 1997.