

# Fast Surface Reconstruction from Contours using Implicit Surfaces

Eric Galin & Samir Akkouché  
Laboratoire d'Informatique Graphique Image et Modélisation  
Ecole Centrale de Lyon  
B.P. 163, 69131 Ecully Cedex

e-mail : [galin-samir]@cc.ec-lyon.fr

## Abstract

*This paper presents a fast and efficient surface reconstruction method from contour data sets. The reconstructed surface is defined as an implicit surface. We create a stratification of the polygonal contours in each cross-section to speed-up the point membership classification involved in the field function computation and avoid the creation of a geometric skeleton. Tests carried out with medical scanner data-sets show that the reconstruction may be performed at interactive rates.*

**Keywords :** implicit surfaces ; polygonization ; stratification ; surface reconstruction.

## 1. Introduction

In this paper, we focus on surface reconstruction from contour data sets that are produced by many medical applications such as tomography and scanners.

Reconstruction techniques may be classified according to the type of the input data. Surface reconstruction from scattered or unorganized points proves to be very difficult issue. Although several methods have been proposed, all of them suffer from high computational complexity because of the lack of knowledge of samples.

In this paper, we address the reconstruction of a three-dimensional object from cross-sectional contours. Each cross section is a non intersecting polygonal shape consisting in one or several closed polygonal contours. We present a new technique based on a *stratification* of polygons and *anisotropic* distance functions that fully exploit the partial structure of the data. We create a potential field function for each cross-section and combine them to create an implicit surface that contains the contours.

The paper is organized as follows. In section 2, we present a short overview of existing reconstruction techniques. In section 3, we describe our reconstruction algorithm and fo-

cus on both the stratification of contours in parallel cross-sections and the definition of an accelerated field function. In section 4, we present polygonized objects and show that reconstruction may be performed at interactive rates.

## 2. Related work

Algorithms may be split into two categories : purely *geometric* algorithms that *directly* create an external triangulated surface, and implicit surface based algorithms.

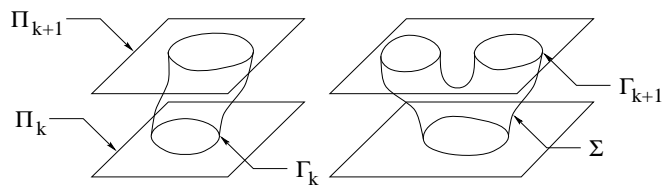


Figure 1. Tiling and branching problems that occur in geometric algorithms

Those algorithms face the following main two problems related to surface reconstruction :

- the *correspondence problem* that consists in determining adjacency relationships between contours from consecutive cross-sections,
- the *tiling and branching* problems that occur when constructing a triangular mesh between vertices of different contours (figure 1).

Several techniques have been proposed to deal with the *correspondence* and the *branching* problems, A good overview and an extended bibliography may be found in [10].

Parametric reconstruction techniques [8] approximate the surface by smoothly deforming a fixed topology polyhedral

model. Constraint based deformations enable the polyhedral model to grow and fit to the final shape. [12] has described a hybrid implicit–parametric method for fitting a deformed sphere to a set of points using deformations of a superquadric.

On comparison to geometric algorithms, implicit surface based reconstruction techniques do not suffer from topological considerations. Existing methods attempt to create a smooth implicit function  $f : \mathbb{R}^3 \rightarrow \mathbb{R}$  such that the sample data points is close or in the zero–set of  $f$ . [9] has presented an incremental technique to fit an implicit surface built from point skeletons at a given approximation level ; however, the reconstruction process is very slow.

Another approach, introduced in [5] and completed [6] consists in using graph traversal techniques to calculate a signed distance function from the data points, approximate tangent planes at the data points and build an implicit surface from this function.

[1] has developed a medial axis generation technique to locate point skeleton based implicit surfaces for scattered data points. [11] has proposed to use a set of R-Functions based on splines, however this technique is computationally very expensive. In [7], a potential field function is built from a signed distance for each contour. The overall implicit function is defined by interpolating the field function of two consecutive slices. Accelerations are performed by using a voxel decomposition of space.

An implicit surface reconstruction algorithm for branching shapes has been addressed in [4], however this technique relies on the creation of a complex skeleton (which involves Bézier triangles as extra skeletal elements to perform bulge–free blends between branches) and its associated field function is computationally expensive to evaluate.

### 3. Description of the algorithm

Our method creates a procedurally defined implicit function whose zero–set includes the set of parallel contours  $\Gamma_k, k \in \{1, \dots, n\}$ . The whole surface itself is defined as an implicit surface  $\Sigma = \{M(x, y, z) \in \mathbb{R}^3, f(x, y, z) = 0\}$ .

In this section, we will assume that the cross–sections are parallel to the plane  $(O, \vec{i}, \vec{j})$  located at the corresponding height  $z_k, k \in \{1, \dots, n\}$ .

We recall that we aim at computing a field function value  $f(x, y, z)$  at every point of space  $M(x, y, z)$ . For each cross–section  $\Gamma_k$ , we create a field function  $f_k$  whose region of influence is bounded by lower and upper planes  $z = z_{k-1}$  and  $z = z_{k+1}$  (figure 2). Thus, given a point  $M(x, y, z)$  in space, we sort it against the parallel cross–sections : let  $z = z_k$  and  $z = z_{k+1}$  the planes bounding  $M(x, y, z)$ .  $f$  will be defined as a combination of  $f_k$  and  $f_{k+1}$ . As we will

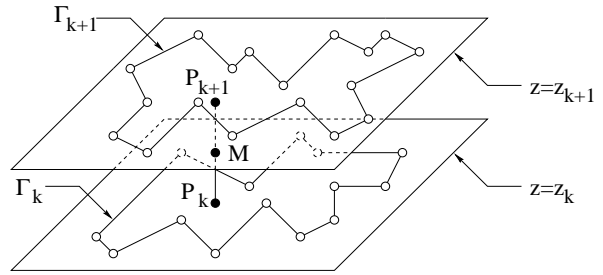


Figure 2. Polygonal contours  $\Gamma_k$  and  $\Gamma_{k+1}$  in consecutive cross–sections, and projected vertices  $P_k$  and  $P_{k+1}$  corresponding to  $M$

see later, the field function associated to a contour  $\Gamma_k$  will be defined by classifying the projection  $P_k$  of  $M$  against  $\Gamma_k$ .

### 3.1. Overview

The class of problems we are interested in can be stated as follows : given a polygonal contour  $\Gamma_k$  in a cross–section  $k$ , define a field function  $f_k$  whose corresponding implicit surface  $\Sigma_k$  satisfies  $\Gamma_k \subset \Sigma_k$ . Then, given  $n$  parallel polygonal contours  $\Gamma_k, k \in \{1, \dots, n\}$ , create the reconstructed surface  $\Sigma$  by blending the implicit surfaces  $\Sigma_k$ .

As proposed in [7],  $f$  may be defined as  $f_k = g_k \circ d_k$  where  $d_k$  is a signed distance function to the polygonal skeleton  $\Gamma_k$  and  $g_k$  a field function. Still, evaluating  $d_k$  is computationally expensive as  $d_k$  is defined as the minimum distance between a point  $M$  and every segment of  $\Gamma_k$ . Although a Voronoï diagram would classify  $M$  against cells and speed up the distance computation, pre–processing each cross–section is both memory exhausting and computationally expensive.

Thus, we propose to use a different partitioning technique and rely on a stratification of the polygonal contours  $\Gamma_k$  to characterize  $f_k$ . Our method may be split into three steps :

- first, we create a stratification of the polygonal contours into trapezoids ; this decomposition is later used to create an implicit function  $f_k$  corresponding to the polygons  $\Gamma_k$  in each cross–section.
- once the stratification of the cross–sections is performed, we define field function  $f_k$  such that its associated implicit front surface  $\Sigma_k$  rests on the contour  $\Gamma_k$ , *i.e.* satisfies  $\Gamma_k \subset \Sigma_k$ .
- eventually, the overall implicit function  $f$  is defined as a combination of the functions  $f_k$  from two consecutive cross–sections.

### 3.2. Trapezoïds classification

In this section, we assume that  $\Gamma_k$  is a polygonal contour in the cross-section  $k$  embedded by a bounding rectangle  $\mathcal{B}_k$ .  $\Gamma_k$  may have holes, however multiple polygons in a cross-section are handled separately.

The stratification of  $\Gamma_k$  is performed as follows. The domain  $\mathcal{B}_k$  is split into vertical strips such that each vertical segment of  $\Gamma_k$  the stratification<sup>1</sup> passes through a vertex  $V_j$  of  $\Gamma_k$ .

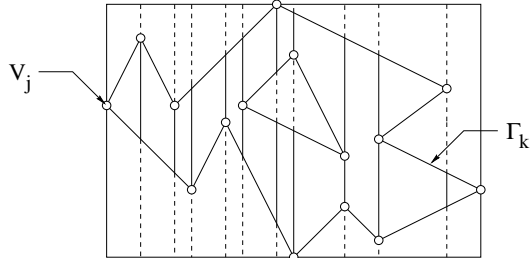


Figure 3. Vertical stratification of a polygon  $\Gamma_k$ , dashed segments are parallel segments of trapezoïds outside  $\Gamma_k$

Thus, the stratification creates a set of trapezoïds with parallel vertical segments (figure 3). Each trapezoïd is marked as *in* or *out*. As we will see later, trapezoïds will provide a fast classification of points that will speed up the overall performance dramatically.

### 3.3. Front surfaces

Let  $\mathcal{T}$  a trapezoïd of the stratification of the contour  $\Gamma_k$ , we create a front surface  $\Sigma_k$  as follows. Let  $\Pi$  a plane orthogonal to the cross section and parallel to the stratification direction (figure 4).  $\Pi$  intersects  $\mathcal{T}$  in two points, referred to as  $I_1$  and  $I_2$ , and  $C$  is set as the midpoint of the segment  $[I_1, I_2]$ . We create a curve  $\Gamma_\Pi$  in the plane  $\Pi$  that passes through vertices  $\Pi \cap \mathcal{T}$ . Therefore,  $\Sigma_k$  is defined as a swept surface by moving  $\Pi$ , and  $\Gamma_k \subset \Sigma_k$ . If  $\Gamma_k$  is convex, the swept surface  $\Sigma_k$  is in general unambiguously defined, however the following two cases may generate creases in the swept surface :

- if a segment of  $\Gamma_k$  is parallel to the direction of the stratification (*i.e.* parallel to the sweeping plane  $\Pi$ ),  $\Sigma_k$  does not contain that segment (figure 5),

<sup>1</sup> Strips may be either vertical or horizontal ; as shown in section 3.4, we will rely on two orthogonal stratifications to define the front surface  $\Sigma_k$  associated to the polygon  $\Gamma_k$

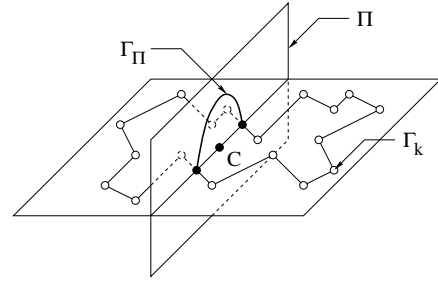


Figure 4. Front surface creation with a vertical stratification of a polygon  $\Gamma_k$

- If  $\Gamma_k$  is non-convex, the swept curve may not be uniquely defined in planes  $\Pi_j$  that cross non-convex vertices  $V_j$  (as shown in figure ,  $\Gamma_\Pi^- \neq \Gamma_\Pi^+$ ).

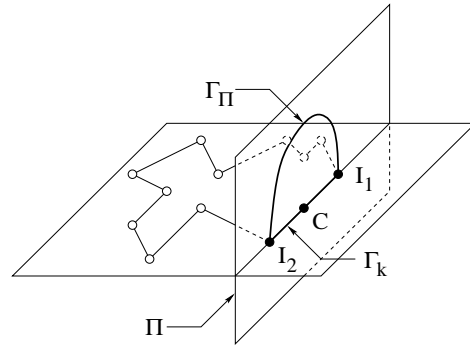


Figure 5. Discontinuities in the definition of  $\Sigma_k$  at edges parallel to the sweeping plane  $\Pi$

Whenever such cases are met, the swept surface may be closed by adding the piece of surface contained in the plane  $\Pi$  (*e.g.* between  $\Gamma_\Pi^-$  and  $\Gamma_\Pi^+$ ). As we will see later, this problem may be skipped by using two orthogonal stratifications.

### 3.4. Potential field

In general, the potential field function  $f$  may be defined as a combination of a distance function  $d : \mathbb{R}^3 \rightarrow \mathbb{R}_+$  and a field function  $g : \mathbb{R}_+ \rightarrow \mathbb{R}$  ; therefore we will refer to the following notation :  $f = g \circ h$ .

We have chosen to use accelerated *anisotropic* distance functions presented in [2]. Given a center point  $C$ , a direction  $\vec{u}$  and an analytic profile curve  $\Gamma$ , a point of space  $M(x, y, z)$  may be defined by a *pseudo-polar* tuple  $(\alpha, r)$  where  $\alpha$  is the *pseudo-polar* angle  $\theta$  with the direction  $\vec{u}$  and  $r$  is the distance  $\|CM\|$  (figure 7).

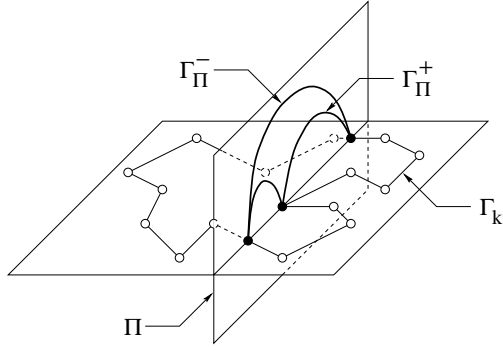


Figure 6. Discontinuities in the definition of  $\Gamma_{\Pi}$  at non-convex vertices of  $\Gamma_k$  with a vertical stratification

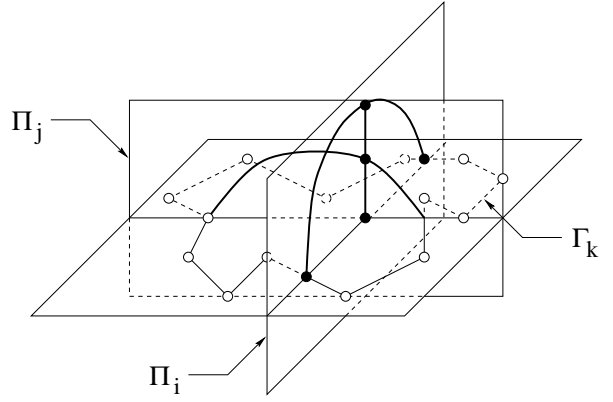


Figure 8. Creation of two analytic front curves in orthogonal planes  $\Gamma_{\Pi_i}$  and  $\Gamma_{\Pi_j}$  corresponding to orthogonal stratifications

$\Gamma$  is used to compute the front surface distance  $\rho(\alpha)$  and the distance  $h_k$  is defined as :

$$h_k(x, y, z) = \frac{r}{\rho(\alpha)}$$

We shall use this technique to compute our potential field  $f_k = g_k \circ h_k$ .

In this section, we assume that the stratification of the contour  $\Gamma_k$  has been performed, therefore the front surface  $\Sigma_k$  is known. The potential field  $f_k$  is defined as follows. Given a point  $M(x, y, z)$  in space,  $\Pi$  is set as the vertical plane that contains  $M$ , and we denote  $P$  the projection of  $M$  on the cross-section.

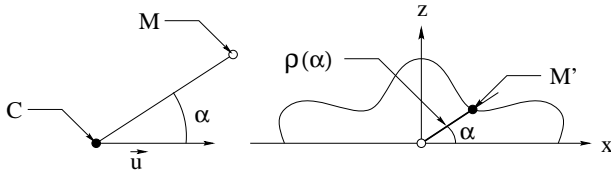


Figure 7. Anisotropic distance function defined by a center point  $C$ , a direction  $\vec{u}$  and an analytic profile curve  $\Gamma$

If  $P$  is classified in an *outside* trapezoid, then the field function  $f_k(x, y, z)$  should be negative, otherwise  $P$  is classified in an *inside* trapezoid and  $f_k(x, y, z)$  should be positive.

Unfortunately, the front surface  $\Sigma_k$  may have vertical sheets when built from non-convex vertices of  $\Gamma_k$  (figure 6). This leads to discontinuities in the definition of the potential field  $f_k$ , which will eventually result in a non-continuous definition of  $f$ .

To avoid those discontinuities, we rely on a two orthogonal stratifications of  $\Gamma_k$  instead of only one (figure 8).  $f$  is defined as a combination of the resulting two potential fields  $f_i$  and  $f_j$  defined in each direction  $(O, \vec{i})$  and  $(O, \vec{j})$  of a given cross-section. The combination of  $f_i$  and  $f_j$  should be *positive* (respectively *negative*) when both  $f_i$  and  $f_j$  are *positive* (respectively *negative*) and should drop to 0 if any of the two field functions converges to 0. This criterion ensures that the whole contour  $\Gamma_k$  is embedded by the front surface  $\Sigma_k$ . Although  $f$  may be any appropriate combination of  $f_i$  and  $f_j$ , we propose to use the R-Function intersection [13] :

$$f_k = f_i \wedge f_j = f_i + f_j - \sqrt{f_i^2 + f_j^2}$$

In practice,  $f_k = \min(f_i, f_j)$  works well and is preferable out of efficiency. Strange as it may seem, the blending combination  $f_k = f_i + f_j$  does produce excellent results and the resulting reconstructed surface appears to be smoother.

We recall that as a point  $M(x, y, z)$  in space is influenced by two consecutive cross-sections  $\Gamma_k$  and  $\Gamma_{k+1}$ ,  $f$  is defined as a combination of  $f_k$  and  $f_{k+1}$ . The following R-Function union definition works well :

$$f = f_k \vee f_{k+1} = f_k + f_{k+1} + \sqrt{f_k^2 + f_{k+1}^2}$$

Needless to say that  $f = \max(f_k, f_{k+1})$  is more efficient and provides good results as well. Visually speaking, our experiments have shown that  $f = f_k + f_{k+1}$  produces the best reconstructed surfaces.

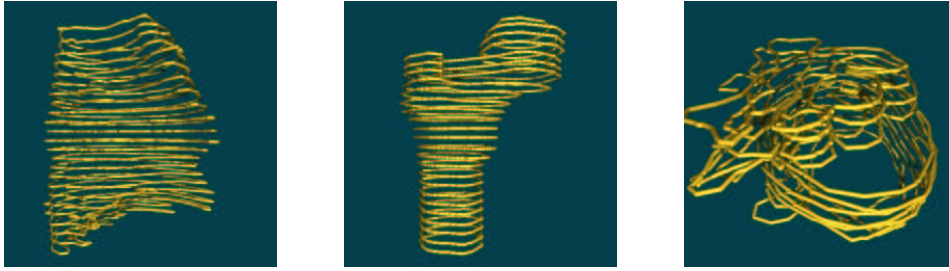


Figure 9. Contour data-sets provided by tomography (from left to right, a lung, a femur and a heart)

### 3.5. Complexity

In this section, we evaluate the complexity of the computation of  $f_k$ . Let  $n$  the number of vertices of the polygon  $\Gamma_k$ . With a *divide and conquer* approach, the stratification of  $\Gamma_k$  enables us to find the trapezoid containing the projected point in  $O(\ln(n))$  time. In fact, only floating point comparisons are to be performed at each step, which is more efficient than any point membership classification over a Voronoï decomposition of the cross-section whose query time is  $O(n \ln(n))$  [3].

It is worth noticing that polygonization algorithms based on a voxel decomposition of space may take advantage of the coherence between adjacent queries with a view to speeding-up point membership classification.

Eventually, we rely on accelerated anisotropic field functions to evaluate  $f_i$ . [2] has presented a set of analytic profile curves  $\Gamma$  that speed-up the computation of the pseudo-distance. A mere dozen floating points operations are involved in the computation of anisotropic field functions.

### 4. Results

The reconstruction algorithm has been implemented in C++ on a Pentium-II processor work-station with a 266MHz Clock and 64 megabytes of main memory. Reported cpu time is in seconds.

Our algorithm has been first tested with an artificial data-set described as 3 slices with holes (figure 10). We have also tested the reconstruction process on different data sets produced by medical tomography. The heart (figure 9,11) is characterized by 15 parallel cross-sections (1729 vertices) which is a rather simple case, whereas the femur (figure 9,11) has been constructed from 26 cross-sections (4435 vertices). The lung is built from 24 cross-sections (2574 vertices).

To evaluate the efficiency of the field function computation, we have voxelized space to create  $n^3$  sample vertices (*i.e.*



Figure 10. Polygonized reconstructed surface from a contour data-set with holes

( $n - 1$ )<sup>3</sup> cells). The meshing was created with a look-up table, combined with a ambiguous configuration detection scheme to guarantee a topologically valid polygonization. Timings show that even such a brute force polygonization performs very fast (table 1).

| Samples | Femur |           | Heart |           |
|---------|-------|-----------|-------|-----------|
|         | Time  | Triangles | Time  | Triangles |
| $50^3$  | 0.44  | 2556      | 0.50  | 5910      |
| $75^3$  | 1.45  | 5783      | 1.68  | 7038      |
| $100^3$ | 3.38  | 10470     | 3.99  | 12734     |

Table 1. Timings for  $n^3$  evaluation of  $f$  for the femur and the heart data-sets, the number of generated triangles is also reported

In practice, a marching cube algorithm would perform better by analyzing fewer cells, *i.e.* only  $O(n^2)$ , would be visited. Since we guarantee that the function  $f$  is null over the contours  $\Gamma_k$ , the implicit surface  $\Sigma$  also contains  $\Gamma_k$  and the marching cube initialization is straight forward.

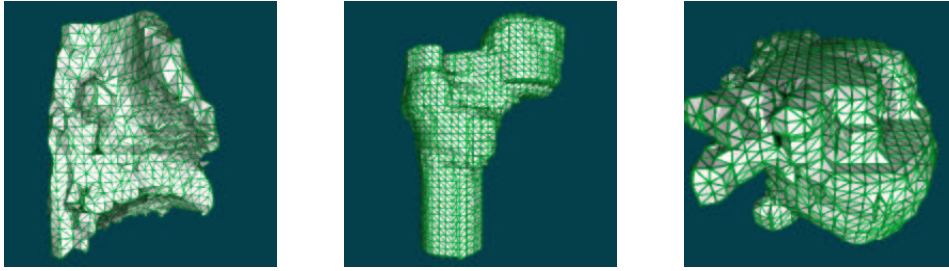


Figure 11. Polygonized reconstructed surfaces from contour data-sets (from left to right, a lung, a femur and a heart)

It is worth noticing that as expected, the stratification of cross-sections  $\Gamma_k$  is almost instant (0.08, 0.05 and 0.06 seconds for the femur and the heart and lung data-sets respectively).

## 5. Conclusion

We have proposed a fast reconstruction technique for contour data-sets. The algorithm takes advantage of the partial structure of the data to create a two-dimensional stratification of contours in each cross-section to speed-up the potential field computation. It is worth mentioning that our algorithm should deal with multiply nested contours successfully.

Although this method is fully automatic, control over the resulting implicit surface may be achieved in several ways, either by changing the analytic profile curves, or choosing different combinations of field functions involved in vertical and horizontal stratifications.

## References

- [1] E. Bittar, N. Tsingos and M.P. Gascuel. Automatic reconstruction of unstructured data : combining medial axis with implicit surfaces. *Proceedings of Eurographics'95*, Vol. 14(3) : pages 457–468, August 1995.
- [2] C. Blanc and C. Schlick. Implicit Sweep Objects. *Proceedings of Eurographics'96*, Vol 15(3) : pages 165–174, August 1996.
- [3] J.D. Boissonat and M. Yvinec. *Géométrie Algorithmique*. Ediscience International, 1995.
- [4] E. Ferley, M.P. Gascuel and D. Attali. Skeletal reconstruction of branching shapes. *Computer Graphics Forum*, Vol 16(5) : pages 283–293, 1997.
- [5] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald and W. Stuetzle. Surface reconstruction from unorganized points. *Computer Graphics (Siggraph'92 Proceedings)*, pages 71–78, July 1992.
- [6] H. Hoppe, T. DeRose, T. Duchamp, M. Halstead, H. Jin, J. McDonald, J. Schweitzer and W. Stuetzle. Piecewise Smooth Surface Reconstruction. *Computer Graphics (Siggraph'94 Proceedings)*, pages 295–302, July 1994.
- [7] M.W. Jones and M. Chen. A new approach to the construction of surfaces from contour data. *Proceedings of Eurographics'94*, Vol 13(3) : pages 75–84, August 1994.
- [8] J.V. Miller, D.E. Breen, W.E. Lorensen, R.M. O'Bara and M.J. Wozny. Geometrically deformed models : a method for extracting closed geometric models from volume data. *Computer Graphics (Siggraph'91 Proceedings)*, Vol 25 : pages 217–225, July 1991.
- [9] S. Muraki. Volumetric shape description of range data using Blobby Model. *Computer Graphics (Siggraph'91 Proceedings)*, Vol 25 : pages 227–235, July 1991.
- [10] J.M. Oliva, M. Perrin and S. Coquillart. 3D Reconstruction of complex polyhedral shapes from contours using a simplified generalized Voronoï diagram. *Eurographics'96*, Vol. 15 (3), pages 397–408, August 1996.
- [11] V. Savchenko and A. Pasko. Function representation of solids reconstructed from scattered surface points and contours. *Computer Graphics Forum*, Vol. 14(4) : pages 97–106, 1995.
- [12] S. Sclaroff and A. Pentland. Generalized implicit functions for computer graphics. *Computer Graphics (Siggraph'91 Proceedings)*, Vol 25 : pages 247–250, July 1991.
- [13] V. Shapiro. Real functions for representation of rigid solids. *Computer Aided Geometric Design*, Vol 12(2) : pages 153–175, 1994.