# Automatic Reconstruction of Unstructured 3D Data: Combining a Medial Axis and Implicit Surfaces

Eric Bittar †, Nicolas Tsingos ‡, Marie-Paule Gascuel ‡,

† TIMB - TIMC / IMAG
Institut Albert Bonniot, Université Joseph Fourier,
Faculté de Médecine de Grenoble, 38 706 La Tronche, France
Eric.Bittar@imag.fr
‡ iMAGIS / IMAG
BP 53, F-38041 Grenoble cedex 09, France
Nicolas.Tsingos@imag.fr, Marie-Paule.Gascuel@imag.fr
(iMAGIS is a joint project of CNRS, INRIA,
Institut National Polytechnique de Grenoble and Université Joseph Fourier)

**Abstract**
*This paper presents a new method that combines a medial axis and implicit surfaces in order to reconstruct a 3D solid from an unstructured set of points scattered on the object's surface. The representation produced is based on iso-surfaces generated by skeletons, and is a particularly compact way of defining a smooth free-form solid. The method is based on the minimisation of an energy representing a "distance" between the set of data points and the iso-surface, resembling previous reserach [19]. Initialisation, however, is more robust and efficient since there is computation of the medial axis of the set of points. Instead of subdividing existing skeletons in order to refine the object's surface, a new reconstruction algorithm progressively selects skeleton-points from the pre-computed medial axis using an heuristic principle based on a "local energy" criterion. This drastically speeds up the reconstruction process. Moreover, using the medial axis allows reconstruction of objects with complex topology and geometry, like objects that have holes and branches or that are composed of several connected components. This process is fully automatic. The method has been successfully applied to both synthetic and real data.*
***Keywords:*** *Shape reconstruction, implicit surfaces, medial axis.*

## 1. Introduction

With the recent development of advanced range-imaging sensors, automatic object reconstruction from sets of sample points has became a very important issue in Computer Graphics. It can be used for mixing real and synthetic objects in production applications, or for speeding up the modelling of complex shapes like character faces. Reconstruction techniques are also needed in many application fields such as inverse engineering, simulation, and medical imaging. Although many different approaches have already been proposed, robust methods lack for reconstructing unstructured sets of points. Methods must be applicable to complex geometrical and topological objects, including those which contain holes, branches, and possibly several connected components,even without any a-priori information. This paper presents a fully automatic reconstruction algorithm based on both medial axis and implicit surfaces that solves this problem.

### 1.1. Previous approaches

One of the most common approaches for reconstruction consists in deforming a surface or a volume in order to fit the set of data points. Energy-minimising curves known as "snakes", were introduced by Kass, Witkin and Terzopoulos [15]. The idea has been widely used and developed [26], and the active contour has turned to an active surface deformed by internal forces such as pressure and by external forces based on the data set [8]. Most of the time implementation relies on splines, but Miller proposes a geometrically deformable model based on triangles [18]. Sclaroff and Pentland [23] fit a super-quadric using modal deformations and displacement maps. Terzopoulos and Metaxas use a physically-based approach for local and global deformations of super-quadrics [25]. All these methods reconstruct fixed topological surfaces.

A few recent papers describe more general models, that can handle objects of various topologies. Delingette [9] introduces "Simplex Meshes" which enable the user to adapt topology interactively. Other approaches change the topology in a purely automatic way, such as Leitner who uses an adaptative spline surface [17], and Lachaud and Bainville who derive their model from triangulation [16].

Another way to by-pass the limitation of snakes to fixed topological type is to use local techniques. Szeliski reconstructs objects using physically-based interacting particles [24]. Hoppe introduces a graph traversal technique in [13] to approximate tangent planes at each data point. For every point in space, he calculates a signed distance to the nearest tangent plane which acts as a local linear approximation of the surface to be reconstructed. The reconstructed surface is then defined as the zero-set of this distance function, and a polygonal representation is computed from a spatial partitionning technique. In [12], the author completes the method by a piecewise smooth surface optimisation in order to obtain more accurate representation and better recovery of sharp features. Boissonnat [5, 6] derives his reconstruction methods from the Delaunay triangulation of the set of data-points. Edelsbrunner and Mücke [10] introduce a parameter $\alpha$ that tunes the degree of detail while deriving a shape from a Delaunay triangulation. Attali [1] computes the Voronoï graph of the points to build the "skeleton" of the object and reconstruct its shape. The main limitation of these techniques is that they do not work well with unstructured sets of points, and cannot handle very noisy databases.

Muraki [19] is the first to use implicit surfaces generated by skeletons for shape reconstruction. These surfaces have the advantage of modelling smooth free-form shapes in a very compact way, and permit an easy definition of a "distance" to the set of data-points that can be used for energy minimisation. However, the method developped in [19] is very slow, and does not fully exploit the potential of implicit surfaces.

### 1.2. Overview

We propose a new method for reconstructing a closed object of complex unknown topology with an implicit surface generated by skeleton-points. The method uses only an unstructured set of data-points scattered on the object's surface; no further information, such as normals to the surface at each data-point, is needed. In the first step, we calculate a medial axis from the set of points. Then, we use smart implicit surfaces providing local shape control to initialise and then progressively refine surface reconstruction. Skeleton-points generating the iso-surface are automatically selected from the medial axis elements during an iterative process. Energy minimisation steps are performed at each iteration in order to adjust the skeleton's parameters. This reduces the final number of skeletons used.

In the rest of this paper Section 2 introduces implicit surfaces generated by skeletons, discusses the previous techniques for reconstruction with implicit surfaces, and presents the basis of the approach. Section 3 defines the medial axis of a set of points and shows how it can be used for initialising the positions and parameters of the skeleton-points that generate the iso-surface. Section 4 presents the iterative algorithm that progressively adds skeletons in order to perform precise reconstruction while limiting the number of skeletons used. We present results and conclusion in Section 5.

## 2. Principles of shape reconstruction with implicit surfaces

### 2.1. Iso-surfaces generated by skeletons

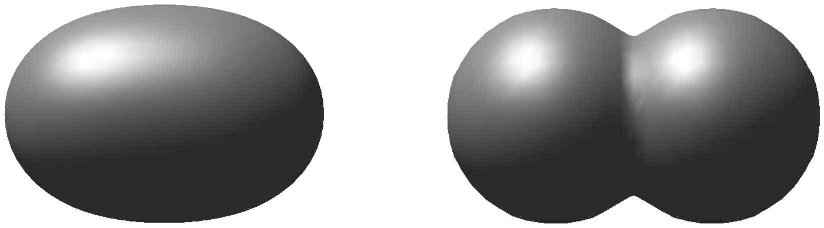
Developed up to now as a tool for free form modeling [30, 29, 4, 3], iso-surfaces generated by skeletons constitute a good alternative to traditional implicit surfaces directly defined by an analytical equation. Using skeletons that are simple geometric primitives facilitates the design and offers compact storage of complex shapes.

An implicit surface $S$ generated by a set of skeletons $S_i (i = 1..n)$ with associated "field functions" $f_i$ is defined by:

$$S = \{P \in \mathbb{R}^3 \ / \ f(P) = iso\} \text{ where } f(P) = \sum_{i=1}^{n} f_i(P)$$

- $iso$ is a given scalar iso-value.
- The skeletons $S_i$ can be any geometric primitive admitting a well defined distance function.
- The field functions $f_i$ are monotonically decreasing functions of the distance to the associated skeleton. They can be defined, for instance, by exponential functions [2], by pieces of polynomials [20, 30], or by parametrized procedural functions [14].

The surface surrounds a volume defined by $f(P) \geq iso$, and normal vectors are directed along the gradient of $f$. The examples of implicit surfaces given in Figure 1 show how the field function can be used for tuning the way implicit primitives blend together.



(a) "soft" field function (b) "sharp" field function

**Figure 1:** *The implicit surface defined by two skeleton-points that generate the same field function. The slope used for the field function on the right side is ten time greater than on the left side.*

### 2.2. Previous approaches for reconstruction with implicit surfaces

Muraki [19] presents an automatic method for generating an implicit shape description from range data that include both scattered points and normal vectors to the surface at those points. The surface used for reconstruction is defined by Blinn's "Blobby Model" [2]. Only skeleton-points $S_i = (x_i, y_i, z_i)$ are used, and fields are expressed as exponential functions of the distance $d(P, S_i)$:

$$f_i(P) = b_i\, e^{-a_i d(P,S_i)}$$

where $b_i$ can be either positive or negative. A "negative skeleton" ,i.e., a skeleton with a negative field, reduces the implicit volume, while positive skeletons extend it. The principle of the reconstruction algorithm consists in minimising an energy representing the "distance" between the current implicit surface and the data. The reconstructed shape is iteratively refined by subdividing some skeleton-points into two new ones, and optimising their parameters according to the energy function.

However, no satisfying way is proposed to select which skeleton points must be divided, leading to a very intensive computation process which forbids the use of this method for detailled objects. Moreover, the choice of an exponential field function doesn't localise the reconstruction process. Every part of the surface is influenced by all the skeleton-points, including those that are distant. Adding new skeletons to the representation modifies the shape everywhere, even in areas that are already reconstructed well. Finally, the initialisation proposed with a single skeleton-point positionned at the center of mass of the data does not seem appropriate for objects of complex topology, such as objects with holes.

Tsingos et al. [28] present a semi-automatic method for reconstruction with implicit surfaces. As in [19], skeletons are progressively subdivided to refine reconstruction, but efficiency is improved by use of a selection criterion based on "local skeleton energy". In order to hold objects of complex topology and geometry, initialisation is interactively performed by the user who has to define an initial set of skeletons and a number of slightly overlapping "reconstruction windows". Results are good, but the experiments we did with this method show that a lot of time has to be spent trying some positions for the initial skeletons and for the reconstruction windows. This can be a problem since, despite optimisation, reconstruction does not run at interactive rates.

For most applications, a fully automatic method seems more appropriate. This is the issue addressed in this paper by introduction of medial axis to provide an automatic and robust way of initialising and adding skeletons.

### 2.3. Combining a medial axis and implicit surfaces

Iso-surfaces generated by skeletons are very convenient for reconstruction, since they provide an easy way to define an energy which gives a current "distance" to the data, they define a smooth surface, and they offer a very compact representation of the object for only positions of skeleton-points and associated field parameters have to be stored.

As shown in the previous section, neither initialising the set of skeletons with a single primitive nor asking the user to perform the initialisation is a good solution. Moreover, splitting existing primitives in two is inefficient since the new skeletons are introduced at the same site, so the optimisation process has to "drag" them over a non negligible distance.

Consequently, we propose a new method that exploits medial axis information computed from the data to both initialise and progressively refine the set of skeleton-points. The idea is to iteratively select these points from a pre-computed set of candidates. This is done in two steps:

- We compute the medial axis of the data-points and tune its resolution in order to obtain a representation of the object by an appropriate number of spheres. This step takes the object's topology into account, and provides a set of candidate primitives to be used for reconstruction. The center of each sphere can be used for positionning a skeleton-point, while its radius gives information on the field function to be chosen.
- In order to perform smooth surface reconstruction while using only a few of the medial axis primitives, we run an iterative process that progressively selects useful primitives from the medial axis data, uses them as new skeletons for the implicit surface, and tunes their parameters in order to obtain a better data fit. The heuristic we use for quickly selecting which primitive to add next is based on the choice of a function with local support for expressing the field functions.

The two next sections detail the two steps of this algorithm.

## 3. Medial axis

The "medial axis" [21] is a filling-out of an object with balls. It can be intuitively considered as the set of points that are the farthest "inside" the object, forming a "skeleton" that gives information on the object's topology and geometry. Thus, these points seem very good candidates for becoming skeletons of an iso-surface that represents the object.

### 3.1. Definitions

Let us consider a set S of spheres that are included inside the object.

**Maximal sphere:** a maximal shpere is a sphere of S that is not included in any other sphere of S.
**Medial axis:** the medial axis is the locality of the centers of the maximal spheres of the object, associated with their radii.

### 3.2. Medial axis extraction

We use an approach based on discrete geometry, for which an approximation of the object as a set of voxels is required. To obtain this representation, we first divide a bounding box for the set of data-points into a uniform 3D lattice. Then, we have to identify voxels to be labelled as "inside voxels".

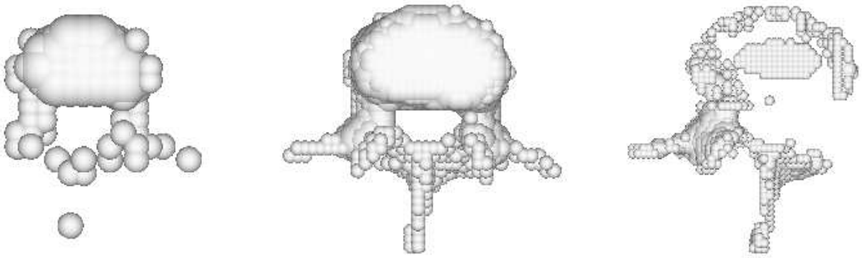
1. **Space partitionning:** Voxels containing data points are labelled as "border voxels" which means that they will be crossed by the surface to be reconstructed.
2. **Labelling the inside:** Finding where the inside is located is not easy for an object of any topology that may have several connected components. Thus, the outside is labelled first. We start at a voxel located at a corner of the 3D lattice, and propagate the information "outside voxel" to all its unlabelled neighbors. The inner voxels are those that have not been reached at the end of this process. The border voxels are not entirely included in the object, so we do not consider them as inner voxels.
   This method works for most applications, but would not hold the case of a closed object containing a closed hole. In other words, the outside is supposed to have a single connected component.

3. **Distance map computation and Medial-axis extraction:** Once the border and the inside of the object are defined, a distance-map is computed based on the Chamfer distance giving a distance between each innner voxel of the object and the outside. Then, we use a local criterion for extracting the medial axis. A detailled description of the algorithm we use is given in Appendix A.

### 3.3. Tuning the medial axis resolution according to the data

Low resolution medial axis may miss the object's details (see Figure 2 (a)), but resolution cannot be increased to any value. If we represent the inner volume of the object as a function of the resolution (defined as the number of voxels in one edge of the cubic lattice), we see that this function grows to a maximum, and then decreases (see Figure 3). After the peak, the resolution is so high that the voxels labelled as "border voxel" do not define a closed surface anymore. In this case, the inside/outside separation cannot be correctly established (see Figure 2 (c)).

In practice, the resolution parameter that is chosen (see Figure 2 (b)) must represent a good compromise between the quality of the final reconstruction and the number of medial axis elements, which increases with resolution and influences the processing time. We set it at the middle of the area where volume increases.



(a) 124 spheres (b) 1050 spheres (c) 1191 spheres

**Figure 2:** *medial axis of the vertebra, resolution = (a) 24, (b) 42, (c) 58*

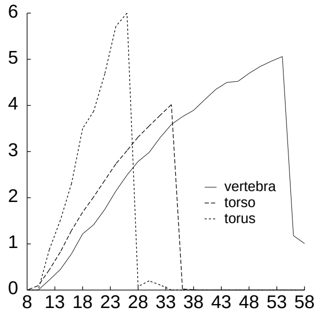



**Figure 3:** *Estimation of the resolution parameter.*

### 3.4. Turning medial axes into implicit surfaces

Since an isolated skeleton-point generates an implicit sphere, a simple solution would be to directly use the medial axis points for generating blended iso-surfaces. The field function characteristics could be computed from the radii of medial axis spheres, with sharp tangents in order to limit blending between neighboring spheres.

However, this approach would have major limitations.

- The surface obtained would not be smooth, since sharp field functions are needed to generate a surface that is close to the set of maximal spheres.

- Some artifact in the reconstructed shape would be produced, since the medial axis is not minimal, in the sense that a medial axis sphere may be included in the union of some other spheres. Using all these overlapping spheres for implicit reconstruction would locally move the surface away from the data points.

Thus, our idea is to further exploit the blending properties of implicit surfaces in order to generate an iso-surface from only a small subset of the medial axis points. This leads to both a more precise and a more compact representation of the reconstructed shape.

The next section explains how we use the medial axis for selecting and adding new skeletons in an iterative way, progressively refining the object's representation. This new method avoids the intensive computational process of splitting skeletons used in previous approaches.

## 4. Creation and refinement of the implicit surface

The basic idea is to progressively select some of the medial axis elements, turn them into skeletons with associated field functions, and adjust their parameters in order to improve the quality of reconstruction. To achieve this, the definition of a robust and efficient selection heuristic is a very important point. The use of local field functions permits definition of such an heuristic.

This section describes the choice of field functions, and then gives details about the selection heuristic and the general reconstruction algorithm.

### 4.1. Definition of local field functions

Local field functions that become zero with their derivatives at a certain radius of influence $R$ have been introduced for optimising field computations during the design of complex objects [30]. Another benefit is that such fields offer local control on the implicit surface, which is particularly important for a reconstruction process since optimising skeletons in an area should not modify an adequate reconstruction of the data already obtained elsewhere. As shown in the next section, the use of local field will also simplify the definition of a local selection criterion for turning a medial axis into an implicit surface.

The chosen field function will be used in a computer-intensive minimisation process. So we need a model that is controlled by very few parameters (in order to limit the dimension of the search space), and that provides efficient computation time as well. Consequently, field contributions $f_i$ that are composed of both a linear and a quadratic curve segment are demonstrated. They have only two parameters (see Figure 4):

- The radius $e_i$ of the sphere created by a skeleton-point $S_i$ alone ($e_i$ is characterized by $f_i(e_i) = iso$) ;
- The stiffness $k_i$ in $e_i$, that defines the blending properties of the surface. $k_i$ can be used to tune the smoothness of the resulting surface (see Figure 1).

Contrary to Muraki who uses both positive and negative field functions, we use positive fields only. This seems more appropriate for future applications that may involve a physically-based simulation of the deformations of reconstructed objects, for instance with the method presented in [11].

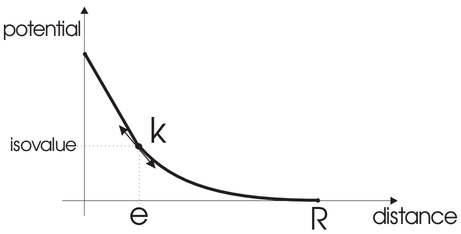



**Figure 4:** *Local field function with two parameters.*

The function is expressed by:

$$f_i(P) = -k_i r + k_i e_i + 1 \qquad \text{if } r \in [0, e_i]$$

$$
\begin{aligned}
f_i(P) &= \frac{k_i e_i (e_i - R_i) + 3e_i - R_i}{(e_i - R_i)^3}(r - R_i)^2 \quad \text{if } r \in [e_i, R_i] \\
f_i(P) &= 0 \qquad\qquad\qquad\qquad\qquad \text{elsewhere}
\end{aligned}
$$

Where $r = d(P, S_i)$ and $R_i = (e_i - \frac{2}{k_i})$ is the radius of influence.

This leads to only 5 parameters which have to be optimised for each skeleton-point: $e_i$, $k_i$, $x_i$, $y_i$, $z_i$.

### 4.2. A local criterion to efficiently select new primitives

Before the reconstruction process, an implicit primitive (e.g., a skeleton-point provided with a field function) is associated with each element of the medial axis: the parameters $x_i$, $y_i$, $z_i$ are defined by the medial axis points, $e_i$ is given by the radius of the associated maximal sphere, and $k_i$ is set to the inverse of the size of voxels used in the medial axis construction in order to give a significant to the size of the area of influence around each primitive.

Then, at each step of the reconstruction process, some of these primitives are added to the iso-surface in order to improve its fit with the data. Indeed, the best candidates for insertion are those of the medial axis primitives $S_i$ that were not already selected, and which area of influence correspond to the zone of the surface where the reconstruction is the worst, as depicted in Figure 5.

So, we define the selection criterion as:

$$C_i = \left( \sum_{j=1}^{m_i} (f(P_{j,i}) - iso)^2 \right)$$

where the sum is made only on the $m_i$ points $P_{j,i}$ that are inside $S_i$'s sphere of influence. The primitives with greatest $C_i$ will be those selected next.

In practice, we use this criterion for selecting several primitives at each iteration of the algorithm, taking care that they stay well distributed within the object. Then we optimise the parameters of these primitives before performing further selections. Details of this process follow.

### 4.3. The reconstruction algorithm

We associate a flag "chosen primitive" to each medial axis element in order to indicate if it has been inserted to the implicit object or not. We also attach a "chosen point" flag to each data point, in order to indicate wether the point is or is not within the area of influence of a sphere that has been selected at the current step.

At each reconstruction step:

1. The "chosen point" flag of each data point is reset.
2. As long as some data points have not been selected:
   - The local criterion $C_i$ associated with each of the non-selected primitives, based on the non-selected data points is calculated, and
   - The primitive with highest $C_i$ is marked as "chosen primitive", and the data points of its area of influence are marked as "chosen points".
3. The $e_i$ and $k_i$ parameters of the primitives that have just been added are optimised, keeping them fixed in space (indeed, the medial axis point gives a good estimate of their position).
4. All the parameters of all the skeletons defining the implicit surface are optimised. This permits reconstruction with fewer skeletons.

As in previous approaches, the energy that is minimised during the optimisation steps is defined from the field values at the data points $P$. The number of the data points is $n_{data}$.

$$E = \frac{1}{n_{data}} \left( \sum_P (f(P) - iso)^2 \right)$$

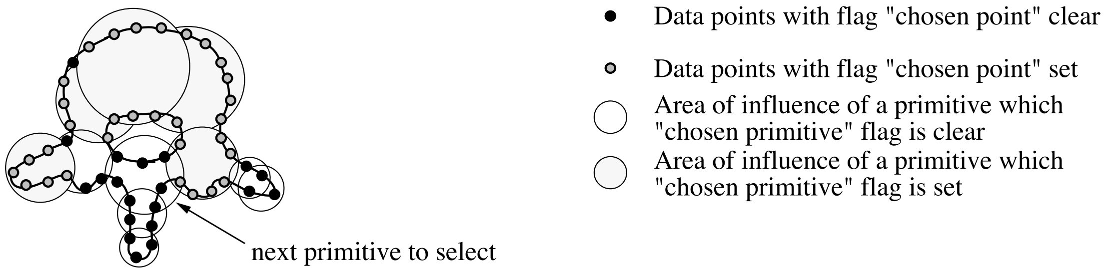


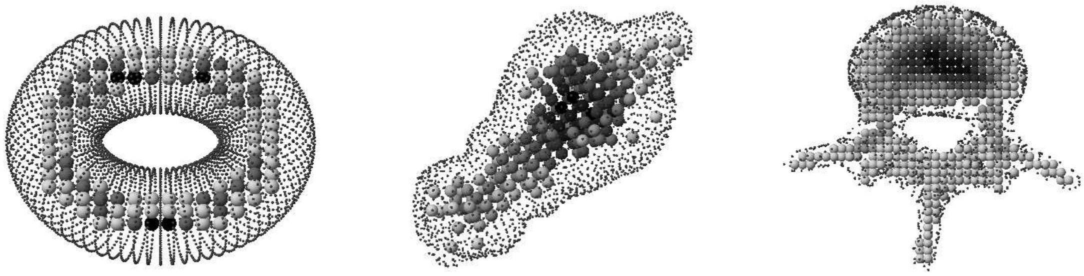

**Figure 5:** *A step of the algorithm*

## 5. Experimental results

The automatic reconstruction process presented is implemented within the visualisation software AVS on a DEC AXP3000 workstation. The numerical details are given in Table 1.

### 5.1. Medial axes examples

We show the results of medial axes calculations on a synthetic torus, on the torso of a dummy which data points come from a range sensor, and on a vertebra (see Figure 6). The data points of the vertebra have been obtained by segmentation of scanner images. A medial axis computation takes about 10 seconds for these examples.



(a) (b) (c)

**Figure 6:** *The data points and the spheres of the medial axis.*

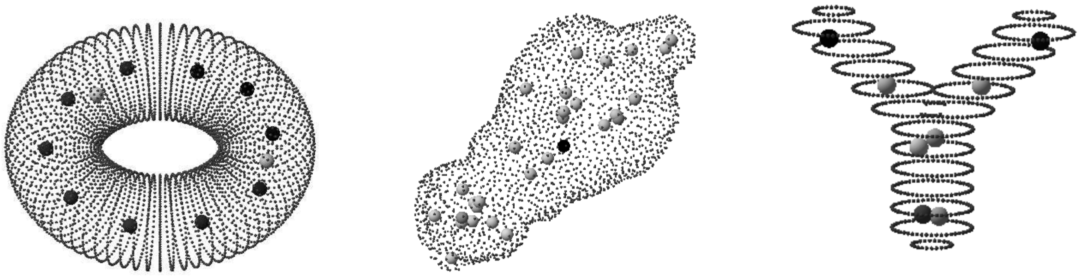
The radii are represented in grey-level: the darkest are the largest.

### 5.2. Samples of reconstruction

This section shows results of reconstruction with the progressive selection algorithm we have defined (Figures 7, 8 and 9).

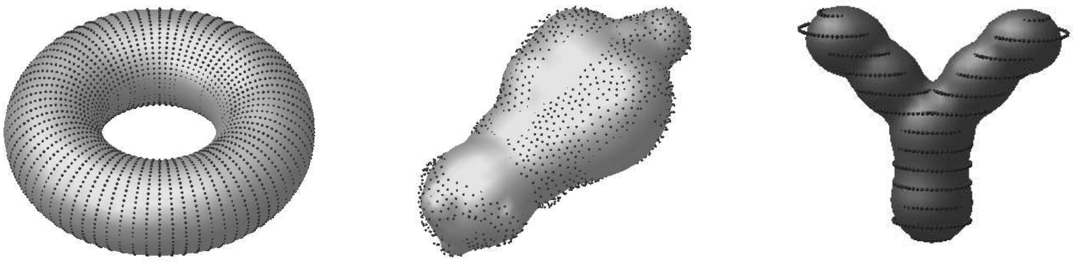
The example of the synthetic "Y" (Figures 7 and 8 (c)) demonstrates that the reconstruction works, even if the distance between adjacent slicing planes is large in comparison to the distance between points on the same contour, which is a limitation for other methods, like Hoppe's [12].

The set of data points of the vertebra which contained 19837 points has been reduced to 2037 points to be used in the reconstruction step.



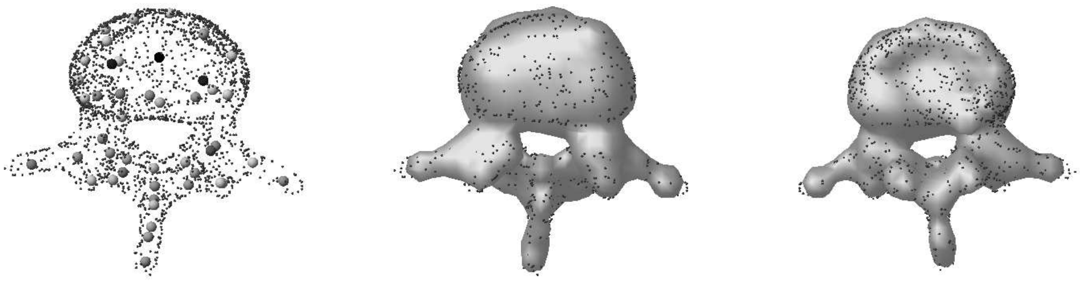
(a) (b) (c)

**Figure 7:** *Data points and skeleton-points.*



(a) (b) (c)

**Figure 8:** *Data points and reconstructed surfaces.*



(a) (b) (c)

**Figure 9:** *Reconstruction of the vertebra with 46 skeleton-points.*

(a) Front view of the data points and skeleton-points
(b) and (c) front and back views of the reconstructed surface.

**Table 1:** *Reconstruction statistics*

| object | number of data points | medial axis resolution | n. of medial axis spheres | number of skeletons | final Energy $E$ | calculation time (sec) | number of passes |
|---|---|---|---|---|---|---|---|
| torus | 4176 | 22 | 172 | 12 | $5.46e^{-4}$ | 163 | 1 |
| torso | 2027 | 26 | 181 | 42 | $3.78e^{-3}$ | 843 | 3 |
| Y | 871 | 14 | 16 | 10 | $2.74e^{-3}$ | 194 | 1 |
| vertebra | 19837 | 42 | 1050 | | | | |
| | 2037 | | | 46 | $3.14e^{-2}$ | 3135 | 1 |

## 6. Conclusion and Future work

A fully automatic surface reconstruction method is presented that can be used on an unstructured set of data points and that can handle objects of complex geometry and topology. The method is based on iso-surfaces generated by skeletons providing a smooth compact representation of complex shapes. The essential principle is to exploit the information given by the discrete medial axis of the set of points to facilitate and accelerate reconstruction. Rather than splitting implicit primitives as in [19] before each optimisation step, a fast and robust algorithm is presented that progressively selects new primitives within the discrete medial axis set. This results in a compact representation of smooth free form shapes, since the number of skeletons selected for the implicit surface generation is much smaller than the discrete medial axis. Reconstruction with still fewer implicit primitives could be performed by constructing more complicated skeletons such as curve segments from the medial axis. Non isotropic field functions depending on the curve parameter could then be computed for defining the isosurface.

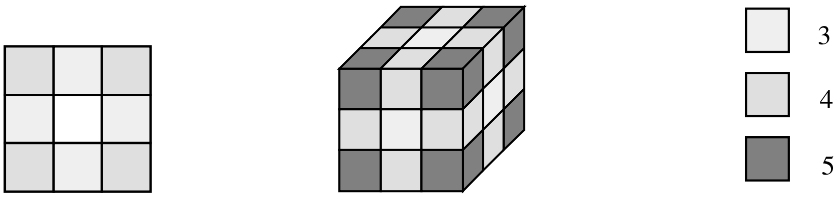
## Appendix A: Medial axis computation

### Distance map

The distance transform operation uses an image as input where the voxels that are inside the object have the value 1 and the others have the value 0, and replaces the value of the object voxels by the distance from the voxel to the background, i.e., to the nearest background voxel. This is called a distance map.

### The Chamfer distance

We use a discrete distance called Chamfer distance $d_{3,4,5}$ [7, 27] to approximate the euclidian distance in $3D$. The principle is to give a weighted cost of 3, 4, or 5 to the move from a voxel $v_0$ to each of its 26 neighbours (see Figure 10). The restriction of $d_{3,4,5}$ in $2D$, $d_{3,4}$, is used as an example (see Figures 10 and 12).




(a) $d_{3,4}$ (b) $d_{3,4,5}$ (c) the values

**Figure 10:** *Local weights of the Chamfer distance $d_{3,4,5}$ and $d_{3,4}$.*

### Building a distance map

The Chamfer distance allows a rapid distance map computation. Instead of exhaustively testing the whole volume for each inner voxel in order to locate the nearest background voxel, the Rosenfeld method [22] is prefered. The distance information is propagated to all the voxels using two sequential convolution operations on the volume. The convolution filters are the two halves of a $3 \times 3 \times 3$ cube, centered on current voxel $v_0$. They are symmetric around $v_0$, the first half $FF$ contains the voxels $v_i$ before $v_0$ in the scanning direction, their local coordinates are $(x_i, y_i, z_i) \in \{-1, 0, 1\}^3$ and their weights are $w_i$ (3, 4 or 5). The backward half-filter $BF$ contains the voxels $v_i(-x_i, -y_i, -z_i)$ of weights $w_i$. None of the two filters contains 0 (see Figure 11).

Each voxel of the object must have an initial value of 1, the background is 0. The two passes successively replace each voxel value $V$ by a value depending on the voxel neighbours values.

**Forward Pass:** top to bottom, back to front, and left to right

$$V[x, y, z] = min_{v_i \in FF}\{V[x + x_i, y + y_i, z + z_i] + w_i\}$$

**Backward Pass:** bottom to top,front to back, and right to left

$$V[x, y, z] = min_{v_i \in BF}\{V[x, y, z], V[x - x_i, y - y_i, z - z_i] + w_i\}$$

Once we have computed the distance map on the object, we need to select the points that constitute the medial axis.

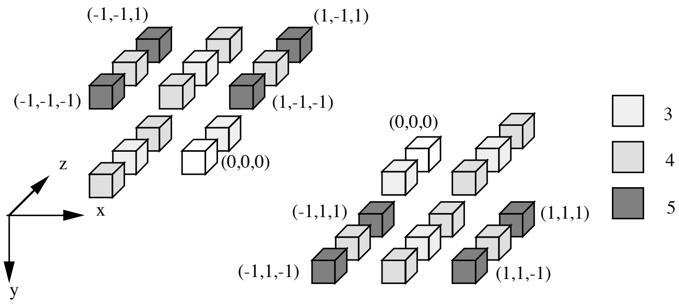



**Figure 11:** *Forward and backward half-filter, and their associated weights.*

### Medial axis extraction

We use a simple local criterion. Each voxel that verifies the local maximum criterion is included to the medial axis.

**Local maximum:** A voxel $v$ that does not propagate the distance information to any of its neighbours $n_i(v)$ of the $3 \times 3 \times 3$ filter is called a local maximum:

$$n_i(v) < v + w_i \qquad \forall i$$

For a technical reason explained in [27], all the values 3 in the distance map must be replaced by the value 1 before using the local maximum criterion.

| | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | 1 | 1 | 1 | 1 | 1 | 1 |
| | 1 | 1 | 1 | 1 | 1 | |

(a) initial image

| | | | | | | |
|---|---|---|---|---|---|---|
| 3 | 3 | 3 | 3 | 3 | 3 | |
| 3 | 5 | 5 | 5 | 5 | 4 | 3 |
| 3 | 4 | 7 | 9 | 8 | 5 | 3 |
| | 3 | 5 | 5 | 5 | 4 | 3 |
| | 3 | 3 | 3 | 3 | 3 | |

(b) distance map

| | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | |
| 1 | 5 | 5 | 5 | 5 | 4 | 1 |
| 1 | 4 | 7 | 9 | 8 | 5 | 1 |
| | 1 | 5 | 5 | 5 | 4 | 1 |
| | 1 | 1 | 1 | 1 | 1 | |

(c) medial axis (circled)

**Figure 12:** *Example of the steps of the medial axis calculation in 2D with $d_{3,4}$.*

### References

1. D. Attali, P. Bertolino, and A. Montanvert. Using polyballs to approximate shape and skeletons. In $12^{th}$ *ICPR*, pages 626–628, October 1994.

2. J. Blinn. A generalization of algebraic surface drawing. *ACM Transactions on Graphics*, pages 235–256, July 1982.

3. Jules Bloomenthal and Ken Shoemake. Convolution surfaces. *Computer Graphics*, 25(4):251–256, July 1991. Proceedings of SIGGRAPH'91 (Las Vegas, Nevada, July 1991).

4. Jules Bloomenthal and Brian Wyvill. Interactive techniques for implicit modeling. *Computer Graphics*, 24(2):109–116, March 1990.

5. J.-D. Boissonnat. Geometric structures for three-dimensional shape representation. *ACM Trans. Graph.*, 3(4), 1984.

6. J.-D. Boissonnat and B. Geiger. Three-dimensional reconstruction of complex shapes based on the Delaunay triangulation. Report 1697, INRIA Sophia-Antipolis, Valbonne, France, 1992.

7. G. Borgefors. Distance transformations in digital images. *Computer Vision, Graphics, and Image Processing*, 34:344–371, 1986.

8. L.D. Cohen and I. Cohen. Deformable Models for 3D Medical Images using Finite Element and Balloons. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'92)*, pages 592–598, June 1992.

9. H. Delingette. *Modélisation et reconnaissance d'objets tridimensionnels 'a l'aide de maillages simplexes.* PhD thesis, Ecole Centrale Paris, France, July 1994.

10. Herbert Edelsbrunner and Ernst P. Mucke. Three-dimensional alpha shapes. *1992 Workshop on Volume Visualization*, pages 75–82, 1992.

11. Marie-Paule Gascuel. An implicit formulation for precise contact modeling between flexible solids. *Computer Graphics*, pages 313–320, August 1993. Proceedings of SIGGRAPH'93.

12. H. Hoppe, T. DeRose, T. Duchamp, and M. Halstead. Piecewise Smooth Surface Reconstruction. In *Computer Graphics (SIGGRAPH'94)*, pages 295–302, July 1994.

13. H. Hoppe, T. DeRose, T.and McDonald J. Duchamp, and Stuetzle W. Surface reconstruction from unorganized points. In Catmull E. E., editor, *Computer Graphics (SIGGRAPH '92 Proceedings)*, pages 71–78, July 1992.

14. Zoran Kacic-Alesic and Brian Wyvill. Controlled blending of procedural implicit surfaces. In *Graphics Interface'91*, pages 236–245, Calgary, Canada, June 1991.

15. M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1:321–331, 1988.

16. J.O. Lachaud and E. Bainville. A discrete adaptative model following topological modifications of volumes. In *Discrete Geometry for Computer Imagery, Grenoble*, September 1994.

17. F. Leitner. *Segmentation dynamique d'images tridimensionnelles.* PhD thesis, Institut National Polytechnique de Grenoble, France, September 1993.

18. J. V. Miller, D. E. Breen, Lorensen W. E., O'Bara R. M., and Wozny M. J. Geometrically deformed models: A method for extracting closed geometric models from volume data. In Sederberg T. W., editor, *Computer Graphics (SIGGRAPH '91 Proceedings)*, pages 217–226, July 1991.

19. Shigeru Muraki. Volumetric shape description of range data using blobby model. *Computer Graphics*, 25(4):227–235, July 1991.

20. H. Nishimura, M. Hirai, T. Kawai, T. Kawata, I. Shirakawa, and K. Omura. Objects modeling by distribution function and a method of image generation (in japanese). *Trans. IEICE Japan*, J68-D(4):718–725, 1985.

21. J.L. Pfaltz and A. Rosenfeld. Computer representation of planar regions by their skeletons. In *Comm. of ACM*, volume 10, pages 119–125, February 1967.

22. A. Rosenfeld and Pfaltz J.L. Sequential operations in digital picture processing. *Journal of the ACM*, 13(4):471–494, October 1966.

23. Pentland A. Sclaroff S. Generalized implicit functions for computer graphics. In Sederberg T. W., editor, *Computer Graphics (SIGGRAPH '91 Proceedings)*, pages 247–250, July 1991.

24. R. Szeliski and D. Tonnesen. Surface Modeling with Oriented Particle Systems. In *SIGGRAPH'92*, volume 26, pages 185–194, July 1992.

25. 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.

26. D. Terzopoulos, A. Witkin, and M. Kass. Constraints on deformable models: Recovering 3D shape and nonrigid motion. *Artificial Intelligence*, 36:91–123, 1988.

27. E. Thiel. *Les Distances de Chanfrein en Analyse d'Images: Fondements et Applications.* PhD thesis, Université Joseph Fourier-GRENOBLE I, Sept. 1994.

28. Nicolas Tsingos, Eric Bittar, and Marie-Paule Gascuel. Semi-automatic reconstruction of implicit surfaces for medical applications. In *Computer Graphics International (to be published)*, June 1995.

29. Brian Wyvill and Geoff Wyvill. Field functions for implicit surfaces. *The Visual Computer*, 5:75–82, December 1989.

30. Geoff Wyvill, Craig McPheeters, and Brian Wyvill. Data structure for soft objects. *The Visual Computer*, pages 227–234, August 1986.