A Greedy Delaunay Based Surface Reconstruction Algorithm

David Cohen-Steiner, Frank Da

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

A Greedy Delaunay Based Surface Reconstruction Algorithm

David Cohen-Steiner — Frank Da

N° 4564

Septembre 2002

THÈME 2

ISSN 0249-6399 ISRN INRIA/RR--4564--FR+ENG

INRIA
SOPHIA ANTIPOLIS

A Greedy Delaunay Based Surface Reconstruction Algorithm

David Cohen-Steiner, Frank Da

Thème 2 — Génie logiciel
et calcul symbolique

Projet Prisme

Rapport de recherche n° 4564 — Septembre 2002 — 16 pages

Abstract: In this paper, we present a new greedy algorithm for surface reconstruction from unorganized point sets. Starting from a seed facet, a piecewise linear surface is grown by adding Delaunay triangles one by one. The most plausible triangles are added in the first place, in a way that prevents the appearance of topological singularities. The output is thus guaranteed to be a piecewise linear orientable manifold, possibly with boundary. Experiments show that this method is very fast, and achieves topologically correct reconstruction in most cases. Moreover, it can handle surfaces with complex topology, boundaries, and non uniform sampling.

Key-words: Delaunay triangulation, surface reconstruction, advancing front method

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles, BP 93, 06902 Sophia Antipolis Cedex (France)
Téléphone : +33 4 92 38 77 77 — Télécopie : +33 4 92 38 77 65

Reconstruction gloutonne de surfaces par la triangulation de Delaunay

Résumé : Dans ce rapport, nous présentons un nouvel algorithme glouton pour la reconstruction de surface à partir de nuages de points non structurés. A partir d'une facette germe, on fait croître une surface polyédrique en ajoutant des triangles de Delaunay un à un. Les triangles les plus plausibles sont ajoutés en premier lieu, et de façon à éviter l'apparition de singularités topologiques. La sortie de l'algorithme est donc par construction une variété orientable, éventuellement à bord. Les résultats expérimentaux montrent que cette méthode est très rapide, et fournit des reconstructions correctes dans la plupart des cas. Celle-ci peut de plus traiter des objets ayant une topologie compliquée, des bords, même si l'échantillonnage est irrégulier.

Mots-clés : Triangulation de Delaunay, reconstruction de surface, algorithme glouton

INTRODUCTION

In several applications, the only available information on a surface is a set of points lying on it. Most computations on this surface then require to find first an approximation of the surface from the samples : this is the problem of surface reconstruction. Several techniques have been applied to this problem. Among them are variational methods [1][2], tensor voting [3], implicit surfaces [4][5], and Delaunay triangulations. For Delaunay-based algorithms, the output surface usually is the union of some triangles selected in the Delaunay triangulation of the input points. Such algorithms can be classified in two sets. On the one hand, volume oriented algorithms output the boundary of some set of tetrahedra and are thus limited to closed surfaces [6][7][8] [9].On the other hand, surface oriented algorithms output a set of explicitly selected triangles.

In most surface oriented Delaunay-based algorithms, triangles are selected independently, that is in parallel [10][11][12]. Such strategies are well suited for theoretical analysis ; to our knowledge, all provably correct reconstruction algorithms fall into this class. However, performances can be enhanced by selecting triangles sequentially, that is by using previously selected triangles to select a new one. Two examples of this greedy approach are the ball pivoting algorithm, and Boyer and Petitjean's algorithm [13][14]. In both algorithms, a triangulated surface is incrementally grown starting from a well-chosen seed triangle. Because it does not compute the whole Delaunay triangulation of the samples, the ball pivoting algorithm is extremely fast. Its main drawback is that the quality of the reconstruction crucially depends on one or more user-defined parameter corresponding to the density of the sampling. Finding good values for these parameters is a delicate task, especially when the sampling is non uniform. Boyer and Petitjean approach relies on what they called regular interpolants. Their algorithm can handle non uniform sampling, but fails when nearly cocircular points are encountered. Moreover, it does not provide any guarantee on the topology of the output surface.

Our approach is inspired by these works, but we believe its efficiency relies on a new and simple idea : the most plausible triangles should be added first. Indeed, in an incremental reconstruction algorithm, an error at some stage can yield disastrous results, like in any greedy approach. It is thus very important to postpone difficult decisions. As we will see, using a confidence-based selection criterion together with a control on the topology of the reconstruction can make ambiguities vanish before they have to be treated.

The sequel of this paper is divided in five sections : we first explain how the topology of our reconstruction is controlled, then we describe the criterion used to select triangles, and give an overview of the algorithm for closed connected surfaces. The fourth section is devoted to the treatment of multiple components, boundaries, and sharp edges, and the fifth to an experimental study.

1. TOPOLOGICAL CONSTRAINTS

At each step of the algorithm, we ensure that the current reconstructed surface S is an orientable manifold with boundary. This property is desirable for itself, since one expects the correct reconstruction to be such a surface. Its main interest, though, is that enforcing it implies discarding some incorrect triangles. Before we go into the details, let us notice that the orientability condition could be removed, so that Moebius band like surfaces could be handled.

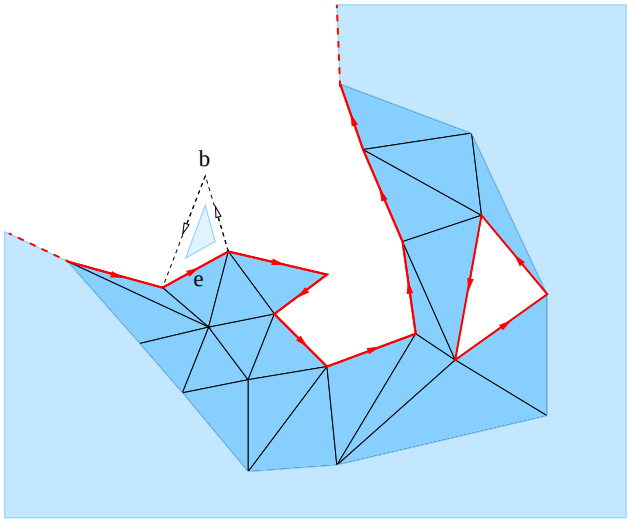
Any triangle t considered by our algorithm has to share an edge e with the boundary of the current reconstruction ∂S. Let b be the vertex of t opposite to e. According to the relative position of t and ∂S, there are four different situations where t may be added to S (see figure 1):

- *extension* : when $b \notin S$.
- *hole filling* : if $b \in \partial S$ and both neighbors of b in ∂S are endpoints of e.
- *ear filling* : if $b \in \partial S$ and only one neighbor of b in ∂S is an endpoint of e.
- *gluing* : when $b \in \partial S$ and no neighbor of b in S is an endpoint of e.

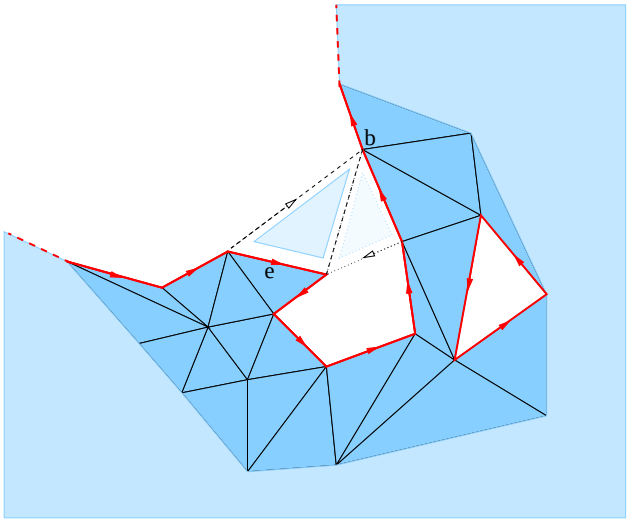
In these cases, we will say that t is a *valid triangle* for e. In any other situation, adding t to S would yield a topological singularity. t would then be called *invalid*. One can check that the set of valid triangles for an edge e can only decrease when S grows.

Extension, hole filling and ear filling do not induce any non manifold edge or vertex. In case of gluing, b will have four incident edges in ∂S after t has been added to S. In order to prevent the appearance of such singularities, one

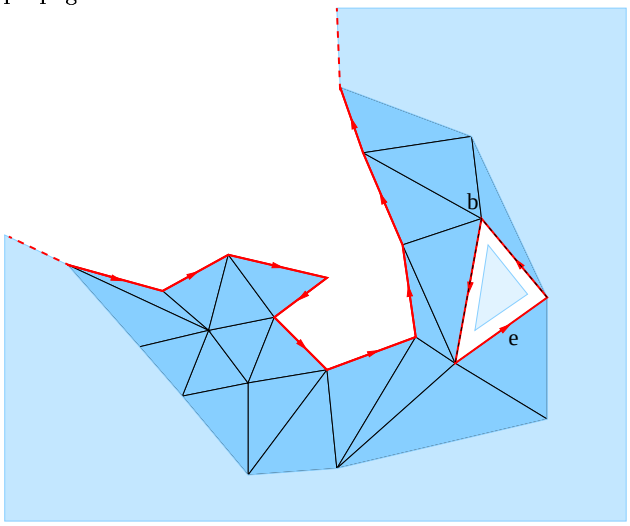
should add another facet, or *twin facet*, together with t. A twin facet for t is a facet t' satisfying the two following requirements : first, t' must have an endpoint of e as vertex, and an edge incident on b in ∂S as opposite edge. Second, adding t and t' to S should not lead to a non orientable surface, which has to be checked explicitly in this case.



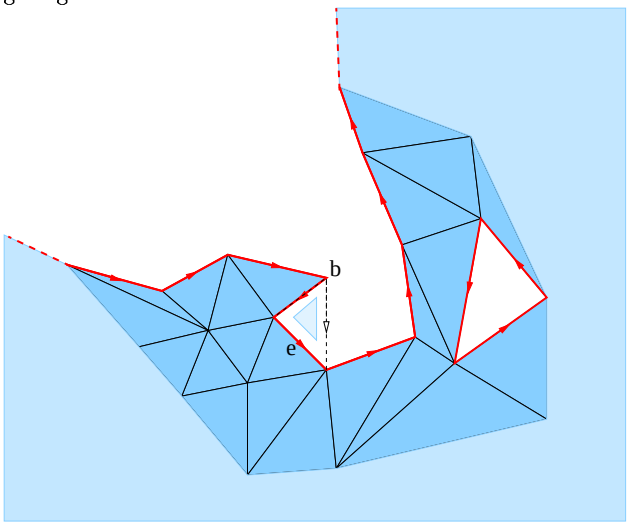
propagation.



gluing.



hole filling.



ear filling.

Figure 1: Valid candidates

2. SELECTION CRITERIA

The process of selecting a new triangle to be included in the reconstruction consists of two steps : first choose a candidate triangle among valid triangles for each edge $e \in \partial S$, then pick one triangle among all candidate triangles. Different criteria are used for both steps.

2.1 Choice of a candidate triangle for a boundary edge

Valid triangles for an edge in ∂S are compared through what we called their *radius*.

Definition 1 *The radius r_t of a triangle t is the radius of the smallest sphere passing through the vertices of t and enclosing no sample point.*

In other words, r_t is the distance from any vertex of t to the Voronoi edge dual to t. This definition of the radius is the one we use for computations.

For smooth curve reconstruction in 2D, the relevance of the corresponding criterion relies on medial axis approximation. Let us first notice that if C is a sampled smooth curve, C has a well-defined correct reconstruction, namely the union of edges connecting adjacent samples. Call these edges correct and the other incorrect. As shown in [15], the union of Voronoi edges dual to incorrect edges converge to the medial axis of $\mathbb{R}^2 \setminus C$ as the sampling density goes to infinity. Around a sample point p, incorrect edges thus have a radius close to the distance from p to the medial axis of $\mathbb{R}^2 \setminus C$ (see figure 2). Correct edges have a radius close to the spacing between adjacent samples. If C is sufficiently well sampled, correct edges will thus have a smaller radius than incorrect ones. Hence, choosing as candidate edge the one with the smallest radius would lead to a provably correct curve reconstruction algorithm.

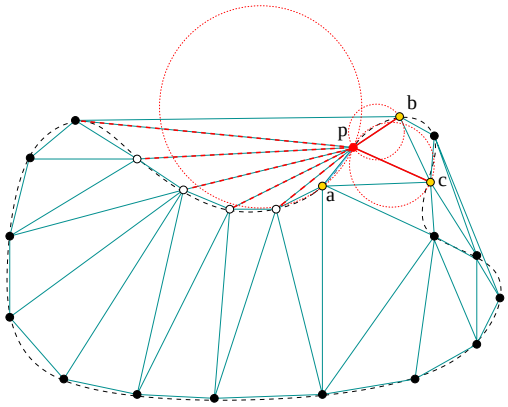


Figure 2: Curve reconstruction : edges incident on p and their radii

The situation is unfortunately more involved for surface reconstruction, as there is in general no way to define *the* good reconstruction if the surface is not known. Consider for example a *sliver* tetrahedron, that is a tetrahedron whose four vertices lie close to a great circle of its circumscribing sphere (tetrahedron e_1e_2ab in figure 3). Suppose S

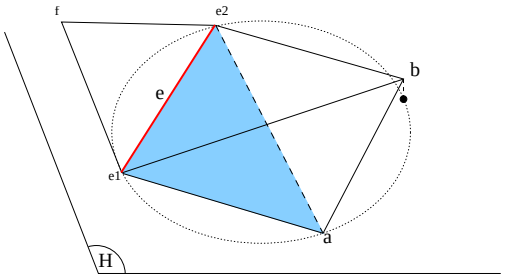


Figure 3: A sliver

is initialized to triangle fe_1e_2. Choosing triangles e_1e_2a and e_2ab or triangles e_1e_2b and e_1ab would lead to equally good reconstructions, both topologically and geometrically.

The existence of slivers has another consequence for us. If S is initialized to triangle e_1e_2a, the triangle valid for e with the smallest radius could happen to be e_1e_2b, which would obviously be a bad choice. In order to solve this problem, we discard valid triangles making a dihedral angle with S smaller than a constant α_{sliver}.

We denote by β_t the absolute value of the angle between the normal to a triangle t incident on an edge $e \in \partial S$ and the normal to the triangle incident on e and included in S.

Definition 2 *The candidate triangle c_e of an edge $e \in \partial S$ is defined by :*

$$c_e = argmin\{r_t | t \text{ is valid for } e \text{ and } \beta_t < \alpha_{sliver}\}$$

If there is no valid triangle for e satisfying the angle requirement, we set $c(e) = \emptyset$. α_{sliver} is set to $5\pi/6$ in our implementation. Experiments show that the value of α_{sliver} is not very critical. The candidate for an edge e can be easily computed by circulating around e in the Delaunay triangulation of the sample points.

The seed triangle is chosen to be the Delaunay triangle with the smallest radius.

2.2 Selection of candidates

We now have to pick one triangle among all candidates corresponding to the edges of ∂S. The heuristic selection criterion defined above works well in practice but can of course fail when the sampling is too sparse. Because we use a greedy approach, the best we can do is to choose the most plausible candidate. How to measure plausibility of a candidate triangle t? A possible criterion is the dihedral angle β_t between t and S. Indeed, if the sampling density is large with respect to the curvature of the underlying surface, the dihedrals should be small in a correct reconstruction. This heuristic statement can actually be proven rigorously in the slightly different context of ε-samples developed by Amenta and Bern [11]. Choosing the candidate with the smallest dihedral thus corresponds, in a certain sense, to choosing the triangle that maximizes a posteriori the quality of the sampling with respect to curvature.

However, this measure is not very relevant when comparing two candidates with very small dihedrals, in particular when the samples are noisy. Thus, we chose to rank candidate triangles t such that β_t is smaller than a certain constant β according to their radius. The ranking of a candidate triangle can thus be defined by the following grade :

Definition 3 *The plausibility grade $p(t)$ of a candidate t is :*

$$p(t) = \begin{cases} 1/r_t & \text{if } \beta_t < \beta \\ -\beta_t & \text{else} \end{cases}$$

if $t \neq \emptyset$, and $p(t) = -\infty$ if $t = \emptyset$.

We experimentally chose $\beta = \pi/6$.

3. OVERVIEW OF THE ALGORITHM

Our algorithm basically uses four data structures :

- the Delaunay triangulation of the sample points D.
- the set of boundary edges. Each edge has pointers to adjacent edges.
  We will also call this set ∂S, as there will be no possible confusion.
- the priority queue Q of pairs candidates/boundary edge ordered by decreasing plausibility grade.
- the set of selected triangles, also called S.

3.1 Stitching of a triangle

We first describe the function implementing the stitching of a triangle to the reconstruction S, $stitch(t, e)$. t is supposed to be a triangle of D incident on edge $e \in \partial S$. The two other edges of t will be refered as e_{cw} and e_{ccw}.

In case of extension, ear filling or hole filling, t is added to S, ∂S and Q are updated, and the return value is *true*. In case of gluing, there are two possibilities. If there is among candidates a twin facet that is more plausible than t, then t is added to S, ∂S and Q are updated, and the return value is *true*. Else, the return value is *later*. We will see in the next section that this is equivalent to ranking a gluing according to the less plausible of the two triangles involved in it.

In all other cases, that is when t is not valid for e, *stitch*(t, e) returns *false*.

Fonction : stitch(t, e)

if t satisfies the conditions for extension **then**
  insert t in S;
  remove e from ∂S;
  insert e_{cw} and e_{ccw} in ∂S;
  remove (t, e) from Q;
  insert $(c(e_{cw}), e_{cw})$ and $(c(e_{ccw}), e_{ccw})$ in Q;
  return *true*;
else if t satisfies the conditions for hole filling **then**
  insert t in S;
  remove e, e_{cw} and e_{ccw} from ∂S;
  remove the candidates for e, e_{cw} and e_{ccw} from Q;
  return *true*;
else if t satisfies the conditions for ear filling **then**
  insert t in S;
  remove from ∂S the two edges of t already in ∂S;
  remove their candidates from Q;
  insert the other edge of t, say e_c, in ∂S;
  insert $(c(e_c), e_c)$ in Q;
  return *true*;
else if t satisfies the conditions for gluing **then**
  for all edge $e' \in \partial S$ incident on b **do**
    if $(c(e')$ is a twin facet for $t)$ and $(p(c(e')) > p(t))$ **then**
      insert t in S;
      remove e from ∂S;
      remove (t, e) from Q;
      insert e_{cw} and e_{ccw} in ∂S;
      insert $(c(e_{cw}, e_{cw})$ and $(c(e_{ccw}), e_{ccw})$ in Q;
      stitch$(c(e'), e')$;
      {NB : $c(e')$ satisfies the conditions for hole filling or ear filling}
    end if
  end for
  return *later*;
end if
return *false*;

Note that during the update of Q, only candidate triangles for new boundary edges are computed. As a consequence, candidate triangles for other edges might become invalid upon execution of *stitch*, due to the change in the reconstruction S. Because we consider candidates by decreasing order of plausibility, bad candidates are more likely to be treated later than good ones. Therefore, the worse a candidate, the more likely its invalidation by the growth of S. When considered for stitching, such a candidate will be recomputed, with a better chance to be correct as there will be less valid triangles for the corresponding boundary edge.

A two dimensional example is shown in figure 4. The initial candidate (dashed) for p is incorrect, and has a poor plausibility grade (1). Thus, well-sampled parts of the curve, where candidates are more plausible, are reconstructed before ; the initial candidate for p becomes invalid (2). The new candidate for p (dotted) is correct (3).

3.2 Main algorithm

The following pseudocode describes our surface reconstruction algorithm in the case of connected surfaces without boundary.

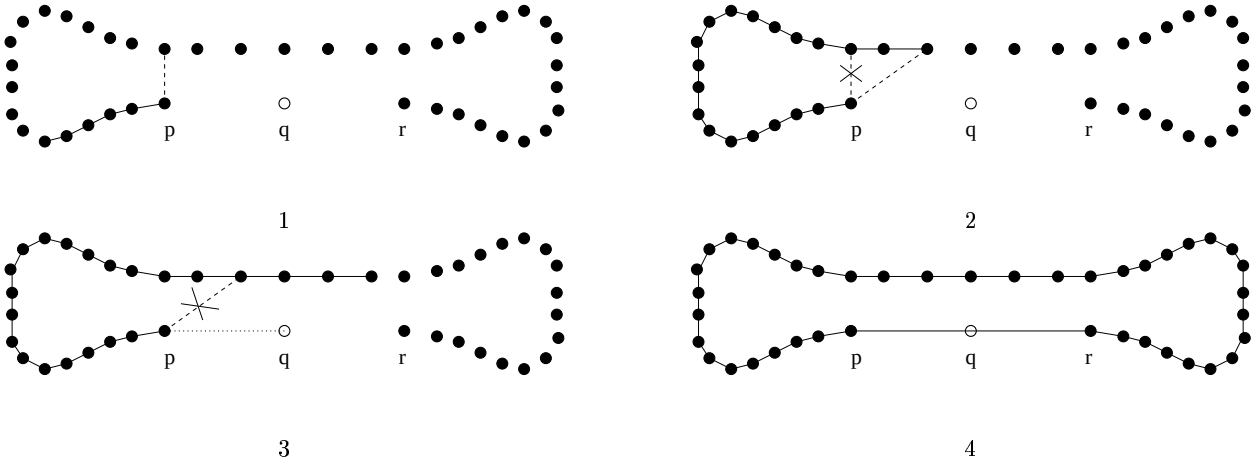


Figure 4: A difficult case at first sight

Main algorithm

compute the Delaunay triangulation D of sample points;
{ INITIALISATION : computation of radii, initialisation of the seed t_s}
$r_s \leftarrow \infty$;
for all t triangle of D **do**
  compute and store r_t;
  if $r_t < r_s$ **then**
    $t_s \leftarrow t$;
  end if
end for
insert t_s in S;
initialize ∂S with the edges of t_s;
for all $e \in \partial S$ **do**
  insert $(c(e), e)$ in Q;
end for
{ SURFACE RECONSTRUCTION }
$(t, e) \leftarrow$ the first element of Q;
while $p(t) > -\infty$ **do**
  $case = \text{stitch}(t, e)$;
  if $case =$ *true* **then**
    $(t, e) \leftarrow$ the first element of Q;
  end if
  if $case =$ *false* **then**
    remove (t, e) from Q;
    insert $(c(e), e)$ in S;
    $(t, e) \leftarrow$ the first element of Q;
  end if
  if $case =$ *later* **then**
    if (t, e) is the last element of Q **then**
      return S
    end if
  else
    $(t, e) \leftarrow$ its successor in Q;
  end if
end while
return S

In words, the surface reconstruction loop scans Q by decreasing order of plausibility until some triangles can be stitched to S. A gluing involving two triangles t and t' can be performed only if any other stitching would involve a less plausible triangle than t and t'.

Let us now prove that the surface reconstruction loop terminates. Each time *stitch* returns *true*, a new triangle is added to S. As there is a finite number of triangles in D, this can happen only a finite number of times. When *stitch* returns *false*, a new candidate is computed. Because the set of valid candidates decreases for each edge, this case also occurs only a finite number of times. Thus, if the algorithm would enter an infinite loop, *stitch* would always return *later* after some point. But Q would then remain unchanged and the algorithm would stop when all candidates $t \in Q$ such that $p(t) > -\infty$ would have been considered.

4. DEALING WITH MULTIPLE COMPONENTS, BOUNDARIES AND SHARP EDGES

4.1 Multiple components

By construction, the output of the algorithm presented in the previous section is a connected orientable manifold with or without boundary. To cope with multiple components, we merely look for a new seed facet among facets of D that are disjoint from S. If such a facet can be found, we insert it in S, insert its three edges in ∂S, and start the main algorithm again. In practice, it may happen that new components consist only of a small number of noisy samples or outliers. We thus filter out components of S having a number of vertices below a small threshold.

4.2 Boundaries

It is impossible to handle all kinds of boundaries and non uniform sampling at the same time. Consider for instance a dense sampling E of a plane P in space such that the boundary of some open disk $Dk \subset P$ is well sampled, and remove all samples contained in Dk. The void created can be viewed as a non uniformity of the sampling, as well as the witness of the existence of a boundary. Indeed, if uniformity is not taken into account, both P and $P \setminus Dk$ are well sampled by $E \setminus Dk$.

Consequently, as we do not wish to rely on any uniformity condition on the sampling, we have to give up detecting such boundaries. Our algorithm will thus fill holes cut off from 'flat' regions of the surface. However, in many cases, a boundary component cannot be closed by adding a spanning disk such that the resulting surface is well sampled. Typically, closing a boundary component due to a transversal clipping of the surface, for instance, would yield large dihedrals at boundary edges. Moreover, if the boundary is sufficiently well sampled, the radii of the two triangles incident on a boundary edge would be very different. These heuristic facts are the one we used for boundary detection :

Heuristic 1 *We discard any candidate triangle t for an edge e such that :*

- $p(t) < 0$
- $r(t) > Kr(t')$

where t' is the triangle of S incident on e and K is a parameter.

K is in some sense the minimal sampling quality for a boundary to be detected. Typical values for K range from 5 to 100. If the surface is known to be closed, then we set $K = \infty$. Note that this heuristic implicitly implies that where the sampling is too sparse with respect to curvature, it should be sufficiently uniform for our algorithm to work.

4.3 Sharp edges

Most real world objects have sharp features, especially manufactured objects considered in industrial applications. Up to now, if the case of sufficiently well sampled smooth surfaces is theoretically well studied, few papers tackle the problem of non smoothness. An algorithm able to detect such features exists [16], but a satisfying reconstruction algorithm in this case still has to be found.

Practically, the output of our algorithm might still have small holes in the vicinity of rough angles. In order to remove them, we use the following trick as a post-processing step : first, points left out by the algorithm are removed from

the Delaunay triangulation D, and the main algorithm is run one more time. If some holes remain, points on their boundaries are removed incrementally from D and the main algorithm is run again. We stop removing points from a boundary when its size becomes zero or starts increasing. After these removals, the set of vertices of D can be viewed as a sampling of a smooth surface obtained by blunting sharp edges in the vicinity of holes. This sampling, though coarse, is in many cases sufficient for the algorithm to fill the holes.

5. EXPERIMENTAL RESULTS

Results displayed in table 1 were obtained on a PC with an INTEL PENTIUM i686 1GHz processor and 1Go RAM. We implemented this algorithm using the CGAL library. For closed surfaces, the number of boundary edges indicates the topological quality of the reconstruction. Note that in this case, no user defined parameter is needed. For surfaces with boundary, the parameter K has to be tuned by the user. Concerning sharp edges, the visual quality of the output may be unpleasant because of a 'sawtooth' effect, even when the reconstruction appears to be topologically correct (see figure 10). Applying some flips to the output reconstruction should improve visual quality at low cost. When the post processing step is not needed, the total running time of the algorithm seems in practice to be less than twice the time spent computing the Delaunay triangulation. If the post processing step is required, running times can become much larger because removals in Delaunay triangulations are costly.
Models come from DASSAULT SYSTÈMES and the following web sites :
`http://www.research.microsoft.com/~hoppe/`
`http://www.cc.gatech.edu/projects/large_models/`
`http://www.swisstopo.ch/`

An implementation of our algorithm is available at : `http://cgal.inria.fr/Reconstruction/`

CONCLUSION

In this paper, we presented a surface reconstruction algorithm based on a region growing approach, following [13][14]. The main underlying idea is to grow a manifold with boundary as cautiously as possible. Results indicates that this algorithm should be competitive with respect to other surface reconstruction methods. However, we were not able to prove the topological correctness of our output under reasonable assumptions on the sampling. Understanding the behaviour of this algorithm seems to involve the study of a question of independent interest : how large can be the diameter of a sliver cluster in the Delaunay triangulation of a good sampling of a surface?

Acknowledgements

We wish to thank DASSAULT SYSTÈMES for providing us with interesting models.

References

[1] Sethian J.A. *Level Set Methods.* Cambridge University Press, 1996

[2] Zhao H.K., Osher S., Merriman B., Kang M. "Implicit Nonparametric Shape Reconstruction from Unorganized Points Using a Variational Level Set Method." *Computer Vision and Image Understanding*, vol. 80, no. 3, 2000

[3] Medioni G., Lee M., Tang C. *A Computational Framework for Segmentation and Grouping.* Elsevier Science, 2000

[4] Hoppe H., DeRose T., Duchamp T., McDonald J., Stuetzle W. "Surface reconstruction from unorganized points." *Comput. Graphics*, vol. 26, no. 2, 71–78, 1992. Proc. SIGGRAPH '92

[5] Boissonnat J.D., Cazals F. "Smooth surface reconstruction via natural neighbour interpolation of distance functions." *Proc. 16th Annu. ACM Sympos. Comput. Geom.*, pp. 223–232. 2000

[6] Edelsbrunner H. "Weighted alpha shapes." Technical Report UIUCDCS-R-92-1760, Dept. Comput. Sci., Univ. Illinois, Urbana, IL, 1992

[7] Amenta N., Bern M., Eppstein D. "The Crust and the β-Skeleton: Combinatorial Curve Reconstruction." *Graphical Models and Image Processing*, vol. 60, 125–135, 1998. URL `http://www.geom.umn.edu/~nina/papers/crust.ps.gz`

[8] Boissonnat J.D. "Geometric structures for three-dimensional shape representation." *ACM Trans. Graph.*, vol. 3, no. 4, 266–286, 1984

[9] Amenta N., Choi S., Kolluri R.K. "The power crust, unions of balls, and the medial axis transform." *Comput. Geom. Theory Appl.*, vol. 19, 127–153, 2001

[10] Adamy U., Giesen J., John M. "The λ-Complex and Surface Reconstruction." *Abstracts 16th European Workshop Comput. Geom.*, pp. 14–17. Ben-Gurion University of the Negev, 2000

[11] Amenta N., Bern M. "Surface Reconstruction by Voronoi Filtering." *Discrete Comput. Geom.*, vol. 22, no. 4, 481–504, 1999

[12] Amenta N., Choi S., Dey T.K., Leekha N. "A Simple Algorithm for Homeomorphic Surface Reconstruction." *Proc. 16th Annu. ACM Sympos. Comput. Geom.*, pp. 213–222. 2000

[13] Bernardini F., Mittleman J., Rushmeir H., Silva C., Taubin G. "The ball-pivoting algorithm for surface reconstruction." *IEEE Transactions on Visualization and Computer Graphics*, vol. 5, no. 4, 1999

[14] Petitjean S., Boyer E. "Regular and non-regular point sets: properties and reconstruction." *Comput. Geom. Theory Appl.*, vol. 19, 101–126, 2001

[15] Brandt J.W. "Convergence and Continuity Criteria for Discrete Approximations of the Continuous planar Skeleton." *CGVIP : Image Understanding*, vol. 59, no. 1, 116–124, 1994

[16] Dey T.K., Giesen J. "Detecting undersampling in surface reconstruction." *Proceedings of the seventeenth annual symposium on Computational geometry*, pp. 257–263. ACM Press, 2001

Data		Delaunay		Reconstruction					
Model	points	cells	t_1	points	triangles	boundary edges	t_2	t_3	$\sum t_i$
* Closed surfaces :									
Mechanic	12593	85158	0,99	12593	25194	0	1,03	0	2,02
blade	882954	6071275	103,76	878724	1757123	1005	105,28	20,38	229,42
bunny	35946	249345	2,75	35943	71882	0	3,19	0,01	5,95
cube	866	4172	2,78	861	1718	0	0,04	0,02	2,84
dragon	435545	314951	49,02	422441	844546	442	47,86	97,75	194,63
happy	542548	3864194	63,36	522701	1045009	567	70,02	182,27	315,65
knot108s	10000	93505	0,83	10000	20000	0	1,03	0,01	1,87
mechpart	4102	30638	0,38	4098	8204	0	0,33	0,03	0,74
papaine1	12577	92619	0,9	12562	25120	0	1,07	0,02	1,99
papaine2	18059	131943	1,39	18045	36086	0	1,54	0,02	2,95
hand	327323	2302290	38,57	327290	654596	0	33,25	0,56	72,38
holes3	3999	29352	0,25	3999	8006	0	0,33	0,01	0,59
horse	48485	342428	3,94	48473	96942	0	4,46	0,39	8,75
oilpmp	30933	200516	2,91	130919	61834	0	2,58	0,04	5,53
sphere	926	5070	0,52	926	1848	0	0,04	0	0,56
torus_random	499	3633	0,03	499	998	0	0,04	0	0,07
knuckle	6062	43401	1.22	5976	11990	4	0.64	0	1.86
* Surfaces with boundary :									
Avant_Auto	59497	394366	5,46	59491	118232	748	5,27	0,25	10,98
Culbu	71150	483383	6,12	71150	142210	88	6,52	0,03	12,67
BootSki	33711	239977	2,94	33700	66005	1393	3,04	3,29	9,27
British_Museum	51095	340275	4,15	51053	101899	251	4,42	0,18	8,75
CarDoor	489205	3217628	56,86	488278	973402	3334	49,12	13,43	119,48
DryerBody	31013	192207	6,66	31010	61093	921	2,34	0,33	9,33
DryerHandle1	49786	319549	18,77	49786	98811	759	3,94	0,02	22,73
DryerHandle2	64464	408629	15,67	64464	128260	666	5,13	0,02	20,82
HairDryer	51563	321389	14,75	51563	10248	641	3,95	0,02	18,72
Robinet	13962	101468	1,17	13884	27309	455	1,4	0,17	2,74
Rth	13894	94824	1,07	13894	27557	231	1,16	0	2,23
Tomo	47861	330959	4,72	47839	92861	2883	15,99	0,12	20,83
Voiture	407577	2646522	46,29	403629	801332	7524	37,46	44,58	128,33
bear	29648	203529	3,64	29621	58553	687	2,4	0,11	6,15
cactus	3280	22908	2,69	3277	6532	20	0,24	0	2,93
cat10	10000	66700	0,69	10000	19908	90	0,82	0	1,51
ch1000	59104	383939	5,57	59104	117707	499	5,26	0,03	10,86
distcap	12745	86887	0,9	12744	25325	161	1,04	0	1,94
engine	11360	73253	1,02	11360	22360	356	0,98	0,01	2,01
hypersheet	6752	52366	0,6	6752	12575	935	0,61	0,01	1,22
club0h	15245	103129	1,17	15232	30438	24	1,30	0,06	2,53
foundry	4101	28111	0,26	4101	7958	242	0,34	0	0,6
mannequin	12772	86663	0,91	12768	25366	168	1,03	0,03	1,97
mmal25	25921	172441	2,89	25921	51201	639	2,98	0,02	5,89
mmma25	6561	43359	0,6	6561	12833	287	0,56	0	1,16
moeller	20020	137183	1,62	20020	39964	74	1,71	0,01	3,34
monkey2	10000	68462	2,98	10000	19611	387	0,84		3,82
nascar	20621	129788	1,58	20619	40725	515	1,65	0,01	2,88
seat	14812	96922	5,16	14812	29248	374	1,29	0	6,45
skidoo_red	37973	257065	3,08	37960	75389	531	3,33	0,05	6,46

Table 1: Running times CPU **(seconds).**

t_1 = computation of the Delaunay triangulation t_2 = main algorithm

t_3 = post-processing $\sum t_i$ = total time $t_1 + t_2 + t_3$

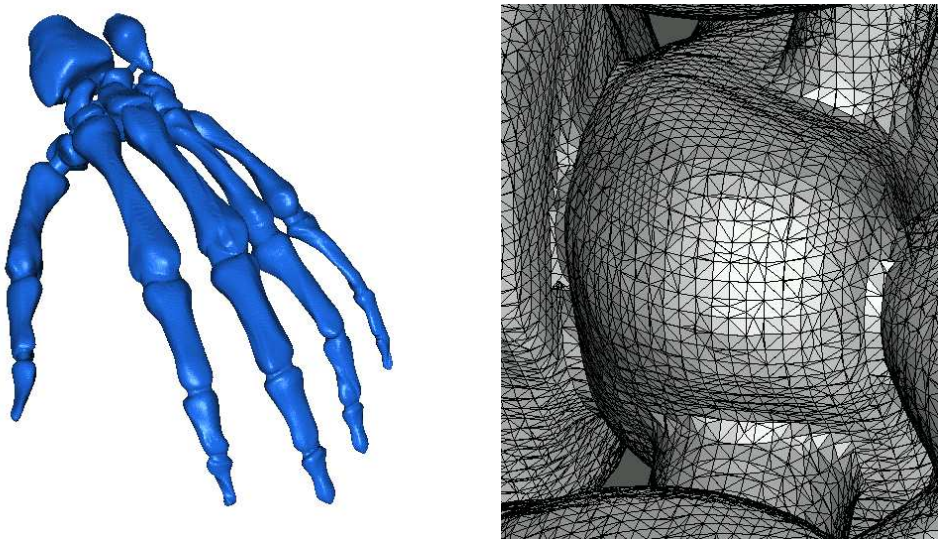


Figure 5: *Hand*

This model is perfectly reconstructed though it contains sharp edges at joints.

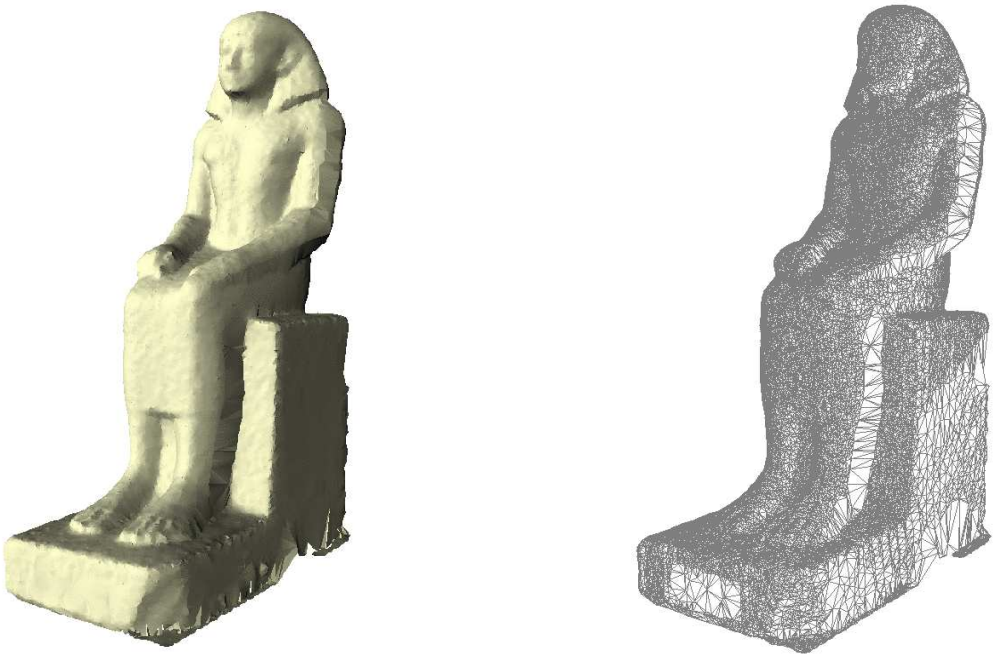
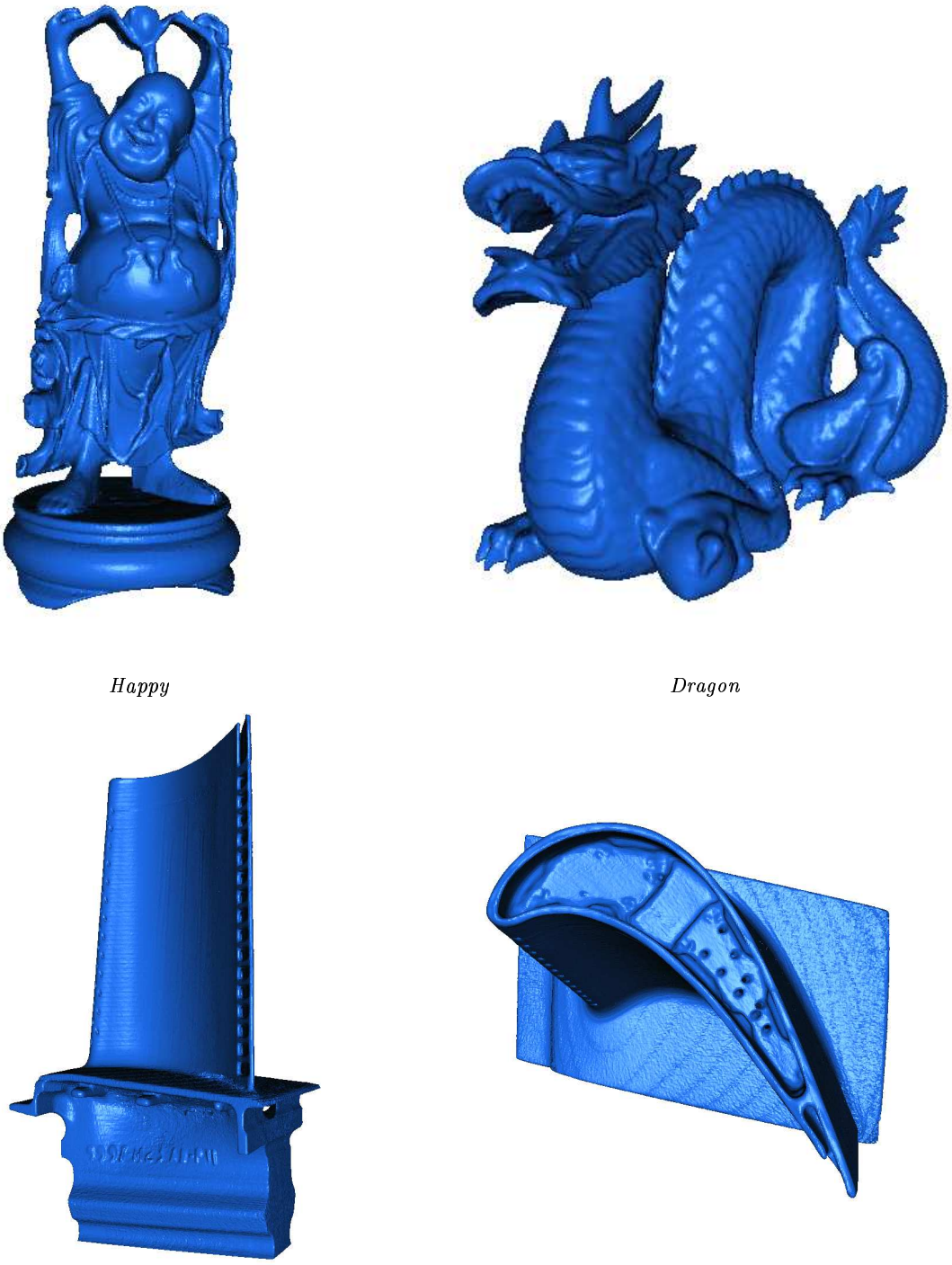


Figure 6: *British_Museum*

Except in the lower part of this statue from the British Museum where the sampling is clearly not sufficient, our reconstruction has no boundary edge. One can see that the sampling density changes a lot, especially near the legs and arms. Moreover, there are sharp edges.



Happy *Dragon*

Blade

Figure 7: Large models

Surprisingly, holes remain in the reconstruction of these very large models, although they look well sampled. We suspect that is due to the existence of very thin details which could not be sampled correctly.

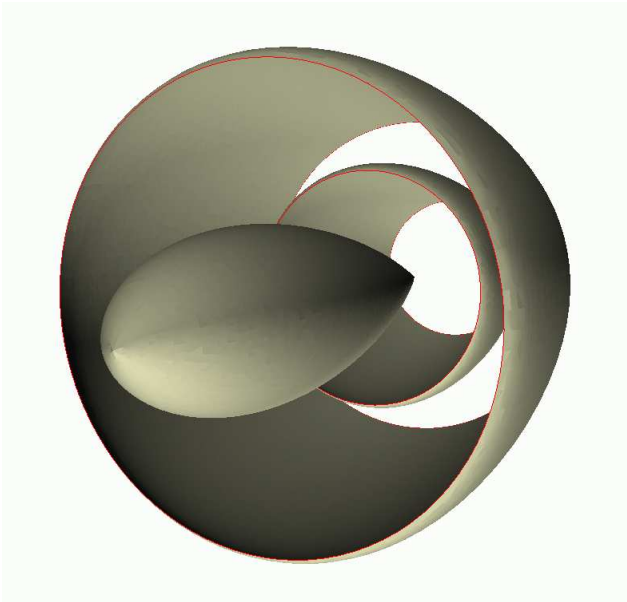


Figure 8: *Engine*

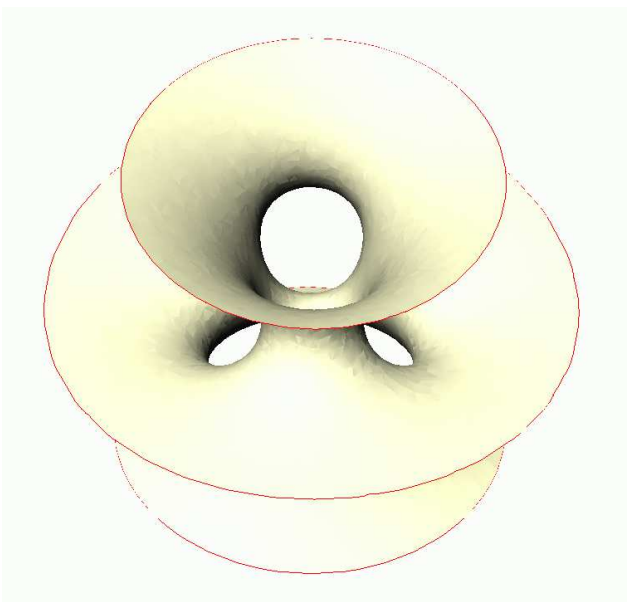


Figure 9: *Hypersheet*

The first model is an easy case of surface with boundary, because it is easy to define a volume whose boundary contains it. The second one is more difficult, as there is no obvious such volume for symmetry reasons.

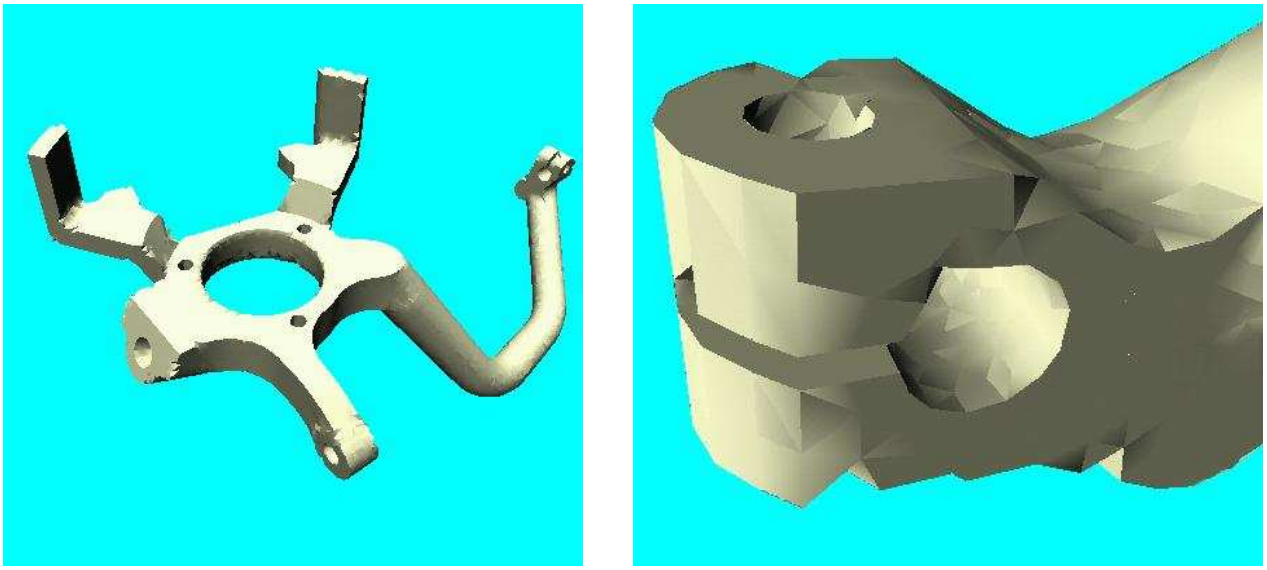


Figure 10: *Knuckle*

Our reconstruction of this difficult model only has four boundary edges.

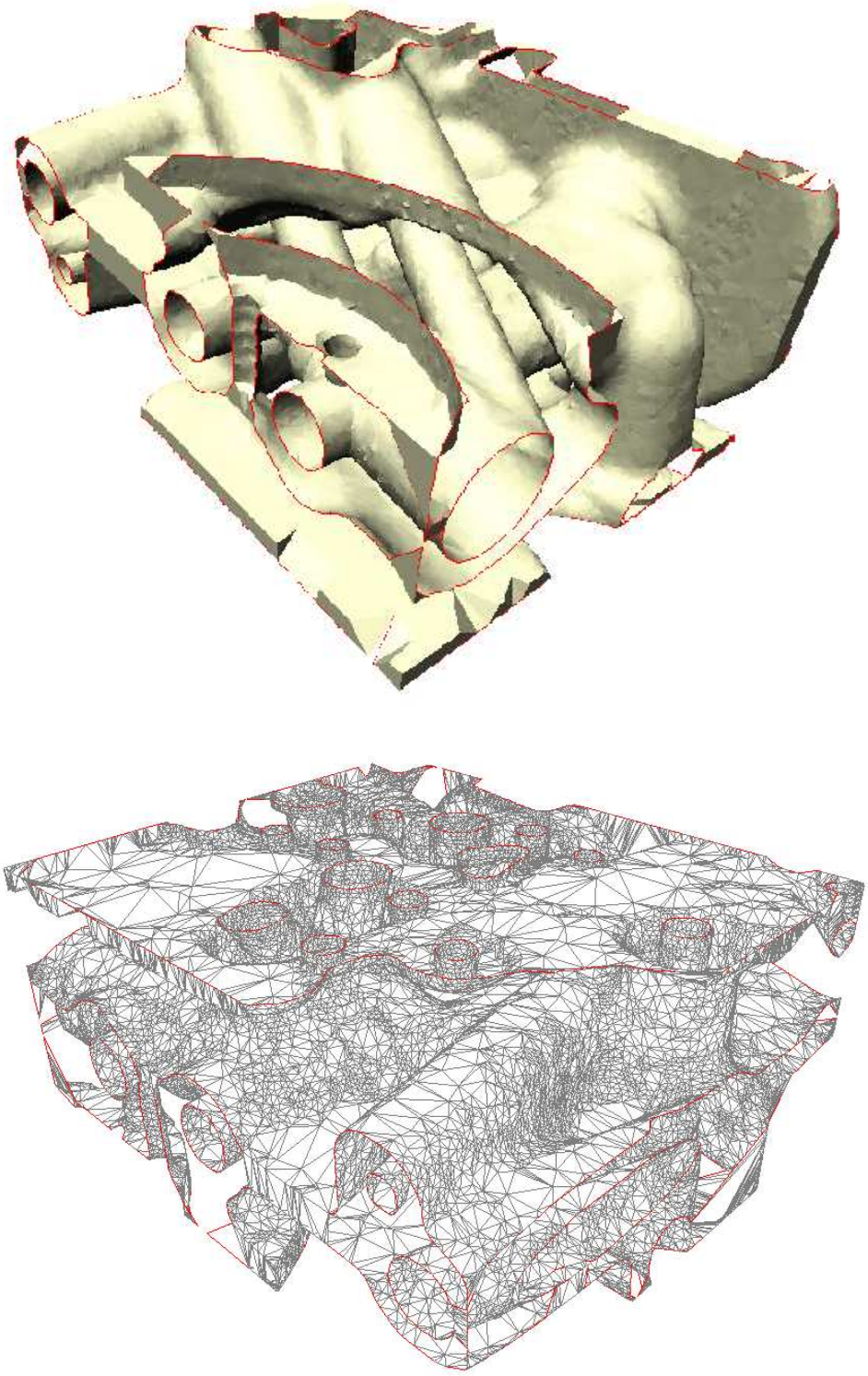


Figure 11: *Tomo*

This shape is hard to grasp, even with a 3D viewer, and its sampling is non uniform. Our reconstruction is perfect apart from some problems on the bounding box. A closer look shows that these are due to errors in the sampling.

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38330 Montbonnot-St-Martin (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399