Virtual Perturbation to Solve Degeneracies in Voronoi-Delone Methods

Prof.Dr.Eng. Ruiz O. Prof.Dr.Math. Cadavid C. Dipl. Eng. Vásquez E.
Student Assistant Granados M. Student Assistant PeNa S.
CAD/CAM/CAE Laboratory
EAFIT University
Medellín, Colombia
{oruiz,ccadavid,evasque3,magranad1,spenaser}@eafit.edu.co

ABSTRACT

In surface reconstruction from slice samples (typical from medical imaging, coordinate measurement machines, stereolithography, etc.) the available methods attack the geometrical and topological aspects or combinations. Topological methods classify the events occurred in the 2-manifold between two consecutive slices. Geometrical methods synthesize the surface based on local proximity of contours in consecutive slices. Many of these methods work with modifications of Voronoi - Delone (VD) techniques, applies on slices i and $i+1$. Superimposed 2D Voronoi Diagrams VD_i and VD_{i+1} (used in surface reconstruction) present topological problems if, for example, a site of VD_i lies on an site or an edge of VD_{i+1}. The usual treatment of this problem in literature is to apply a geometrical disturbance to either VD_i or VD_{i+1}, thus eliminating the degeneracy. Recent works seek to quantify the amount of the disturbance applied in relation to the probability distribution of the event "change in the topology of VD". In this article, in contrast, virtual disturbances are proposed and implemented, which allow the application of subsequent steps of the algorithm at hand (in this case, tetrahedra construction for surface reconstruction) without regard to the geometrical exception. Tetrahedra (or any other downstream constructs) can then be instantiated as per non-degenerate conditions. Although this method is applied for surface reconstruction, it gives insight as to how to circumvent degeneracies in procedures based on VD methods.

Keywords
Virtual Perturbation, Degeneracies, Voronoi, Delone, Surface Reconstruction

1. INTRODUCTION

Degenerate conditions in geometric algorithms have been attacked by different ways: (i) by stating the same problem in different spaces with better conditioning, (ii) by increasing the real computation precision, (iii) by relying on rational numbers, with no rounding errors, and (iv) by numerically disturbing the input to the geometrical algorithms, while at the same time estimating the probability of respecting the original problem topology. Strategies (i) and (ii) have been extensively applied in Numerical Analysis, for example by generating equivalent linear systems with better manipulation properties. Alternative (iii) has been investigated, for example in (Computational Geometry Algorithm Library [2]), with exact computation paradigmsi. Strategy (iv) has presented results of probablility bounds for alteration of Voronoi-Delone topology upon numerical disturbance of degenerate events (see [4]).

Virtual Perturbations have been used in other contexts (see [3] for previous reports). In the strategy presented here, the problem at hand is analyzed, and the topological structure of a correct result is created in the form of objects, without immediate instantiation. This strategy assumes

Conference proceedings ISBN 80-903100-7-9
WSCG'2005, January 31-February 4, 2005
Plzen, Czech Republic.
Copyright UNION Agency - Science Press

the possibility of detecting the degenerate condition. Beyond this point, no numerical manipulation is introduced. Instead, the generic objects are instantiated with the numerical information, and the algorithm proceeds. It should be noticed that none of the mentioned strategies solves the degeneracy problem. Each is suited for a particular domain of problems. The one presented here is of course convenient when there is a finite number of topological configurations, that allow to be enumerated and distinguished.

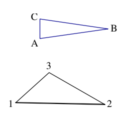
The particular context in which this strategy is presented is the general problem of surface reconstruction, from planar samples. Particular steps of the Boissonat & Geiger algorithm ([1, 5]) have been changed, to make them more robust (see [[6, 7]] besides this article). Section 2 gives the application context of the present work and reviews related literature. Section 3 describes the methodology applied and the procedures followed. Section 4 gives account of the results, while section 5 concludes the article.

2. CONTEXT AND LITERATURE REVIEW

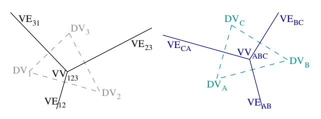
The algorithm proposed and implemented by Boissonat & Geiger in [1, 5] (called here **B+G**) builds tetrahedra filling the space between two consecutive sampling planes i and $i+1$. B+G is a fairly fast and robust algorithm, originally presenting weakenesses that have been strenghtned by complementary works. The boolean union of such tetrahedra renders the solid object cut by the sampling planes. This strategy basically uses geometrical proximity between contours to infer the existence of surface. Because of this reason, it may present over-stretched surfaces, joining local portions of contours which are close, while the contours as a whole have little to do with each other. This efect may be diminished by applying a 2D shape similarity (2DSS) algorithm (see [6, 7]). On the other hand, the tetrahedra are built by projecting the Voronoi Diagram (VD) of the point set in level i, VD_i, onto VD_{i+1}, or viceversa. A condition which is degenerate for B+G is that a Voronoi site of VD_i exactly lie on either a site or an edge of VD_{i+1}. Such a condition produces a non-manifold and self - intersection condition in the surface so build. The treatment of such exceptional situations to re-arrange the data for a smooth functioning of B+G is the goal of this work.

2.1. BRIEF REVIEW OF THE B+G METHOD

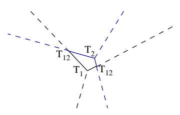
The B+G method was developed by Jean-Daniel Boissonnat ([1]) at INRIA and later improved by Bernhard Geiger ([5]).



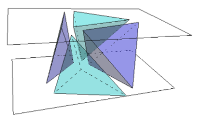
(a) Original contours



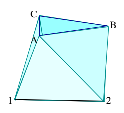
(b) Delone Triangulation and Voronoi Diagram of the contours



(c) Created Joint Voronoi Diagram



(d) Related tetrahedrons



(e) Reconstructed surface

Figure 1: A simple example of the surface reconstruction process using the B+G method

The B+G method processes each pair of adjacent levels, $level_i$ and $level_j$ and creates a flat-faced polyhedral surface that joins the contours of both levels. The process is based on geometric closeness, supported over two geometric structures: the 2D Delone Triangulation DT and the 2D Voronoi Diagram VD.

The B+G method divides the interior of the contours in triangles by creating the Delone Triangulation of the contour vertices (figure 1(b)). After some processing, the Voronoi Diagrams belonging to the levels are used to create a planar graph named the Joint Voronoi Diagram (figure 1(c)). This graph states how the the triangles in the levels are linked, by translating it to tetrahedrons (figure 1(d)). Finally the triangles of the tetrahedrons facing the exterior are taken as the reconstructed surface (figure 1(e)).

3. IMPROVEMENTS ON THE B+G METHOD

The Polyhedral Surface Method is based on the B+G method. The B+G method reconstructs incomplete surfaces and presents no-manifold situations. The improved version solves the incomplete surface problem (section 3.4.) and the no-manifold situations are minimized by taking into account special cases when creating the joint Voronoi Diagram (section 3.3.). Also minor changes, in implementation and conceptualization, were done to reach better results. In the following sections a description of the improvements did is given.

3.1. Satisfaction of conditions

Before the construction of the Joint Voronoi Diagram, some conditions must be fulfilled for the Delone Triangulation DT on each level:

Condition 0: Completeness of DT The triangulation should include all the edges which form the set of contours C_i in the level. In formal terms, condition 0 is defined in equation 1.

$$\forall \text{ edge } e \in C_i : \exists DE_{ij} \text{ s.t. } e = DE_{ij} \tag{1}$$

Condition 1: Partition of DT by contours The classification of every triangle in the Delone Triangulation as internal or external with respect to the contours must be possible. Let the union of all the external triangles be called External Region ER, and the union of all the internal triangles be the Internal Region IR, then, condition 1 is formalized in equation 2.

$$\forall DT_{ijk} \in DT : (DT_{ijk} \subset IR) \vee (DT_{ijk} \subset ER) \tag{2}$$

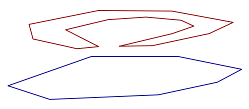
Condition 2: Confinement of circumcenters The circumcenter of every Delone triangle DT_{ijk} must lie inside the region to which DT_{ijk} belongs. In formal terms, it is defined in equation 3.

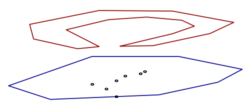
$$\forall DT_{ijk} \in DT : \begin{cases} \text{if } DT_{ijk} \subset IR \Rightarrow circumcenter(DT_{ijk}) \in IR \\ \text{if } DT_{ijk} \subset ER \Rightarrow circumcenter(DT_{ijk}) \in ER \end{cases} \tag{3}$$

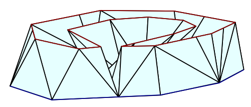
Notice that the satisfaction of condition 0 leads to the satisfaction of condition 1 and vice versa. Condition 0/1 and 2 are dependent between them; condition 0/1 must be fulfilled before condition 2 is checked. This detail is not considered on the original method, and some presented algorithms fails because of this omission.

3.2. Lifting of internal points

The B+G method adds some points in the interior of the some contours in the levels (see figure 2).


(a) Given contours


(b) Internal points added by the B+G method


(c) Surface reconstructed

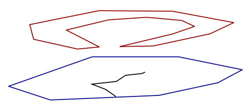

(d) Ghost line produced by re-sampling the resulting surface by the original planes

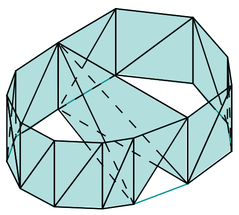
Figure 2: Internal points added by the B+G method

In the case of re-sampling the resulting surface using the original planes, ghost lines will appear inside the original contours as it is seen in figure 2(d). These lines appear because the added points create an approximated medial axis. To avoid such lines, the points that are inserted inside the contours are orthogonally projected on a level between the processed levels before the surface is finished. The distance between the created level and the original one is small, to avoid geometric and topological degeneracies.

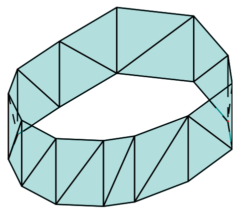
3.3. Special cases in the creation of the Joint Voronoi Diagram

The Joint Voronoi Diagram of two consecutive levels results from intersecting the orthogonal projections of their Voronoi Diagrams on a common plane. The Joint Voronoi Diagram is composed by three kinds of nodes: T_1, T_2 and T_{12}. The T_i nodes correspond to the Voronoi vertices belonging to the Voronoi Diagram on level i, with $i = 1, 2$. The T_{12} nodes correspond to the intersection of two Voronoi edges.

Every node in the graph corresponds to a tetrahedron, and the union of all these tetrahedrons form the 3D Delone Triangulation of the contour points P of both levels i and j. Because the tetrahedrons that are translated from the graph are Delone tetrahedrons, they satisfy the "empty-sphere" condition, that is, the sphere that circumscribes the tetrahedron does not contain any other point in P except its vertices. Every tetrahedron is created with four Delone vertices. See figure 1(d) where a T_1, T_2 and two T_{12} tetrahedrons are translated from the Joint Voronoi Diagram in figure 1(c).



(a) Surface reconstructed using the B+G method



(b) Surface reconstructed taking into account the special cases

Figure 3: Differences between a simple surface reconstructed using a perturbation and solving the special cases

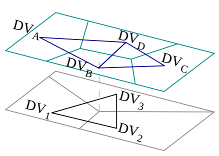
The special cases are generated when more than four Delone vertices stand on the surface of an empty sphere. In the implementation of the B+G method, when more this situation is found a perturbation is applied to vertices. This perturbation leads, some times, to non-wished surfaces, like the one shown in figure 3(a). The conceptual consideration of the special cases does not leave the election of the tetrahedrons to a random perturbation, and improves the reconstructed surfaces (figure 3(b)).

When a special case is created, the construction of the Joint Voronoi Diagram becomes ambiguous because there is more than one configuration of nodes (and therefore, tetrahedrons) that could be generated. If nodes of different configurations are kept together at the same time, the graph becomes inconsistent, because the faces of the related tetrahedrons intersect among them and the number of connections between the nodes exceed the limit. The problem is solved when a valid configuration of nodes, defined as the set of nodes whose related tetrahedrons properly share faces, is found and it is inserted into the graph. These special cases are not considered in the original method.

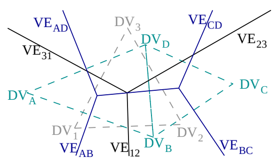
As an upper bound, at most six Delone vertices may share the same empty sphere, because on each level the limit of co-circular vertices is three and the graph generation involves just two levels.

3.3.1. Case 1: Voronoi Vertex vs. Voronoi Edge

This case is generated when five Delone vertices are co-spherical, leading to a Voronoi vertex belonging to level i be projected on a Voronoi edge belonging to level j, or vice versa. An example is shown in figure 3.3.1.



(a) Vertices DV_1, DV_2, DV_3, DV_B and DV_D are co-spherical



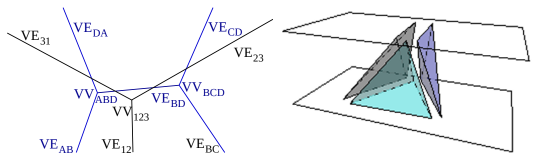
(b) Projected case

Figure 4: Special cases in the creation of the Joint Voronoi Diagram: example of Voronoi vertex vs. Voronoi edge case

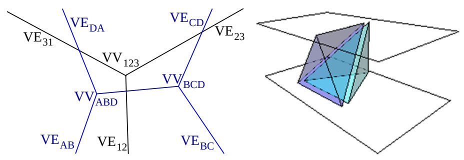
Each pair of Voronoi edges that intersect each other, generate a T_{12} tetrahedron. In this case three intersections are found, VE_{12} vs. VE_{BD}, VE_{23} vs. VE_{BD} and VE_{31} vs. VE_{BD}. The creation of all these T_{12} tetrahedrons is illegal, because their faces intersect and one face is shared by more than two tetrahedrons.

The ambiguity of the situation is essentially shown when creating the T_i tetrahedron related to VV_{123} in figure 3.3.1.. In this case, two different Delone vertices are found at the same distance to VV_{123} and the distance is the minimum among all the points, so, any of them could be used as the apex. These two found Delone vertices are the ones related to the Voronoi edge on which VV_{123} is projected, in figure 3.3.1. they are DV_B and DV_D. Because there are two alternatives to choose from, this situation allows two configurations as solutions.

The election of the apex for the T_i tetrahedron between these Delone vertices, states that the distance from the elected vertex to VV_{123} is virtually smaller than the distance from the non-elected vertex to VV_{123}. The ambiguity is eliminated as shown in figure 3.3.1..



(a) Delone vertex DV_D elected as apex



(b) Delone vertex DV_B elected as apex

Figure 5: Special cases in the creation of the Joint Voronoi Diagram: Solutions for Voronoi vertex vs. Voronoi edge case

Identification of a valid configuration The complete sequence of steps is given in detail in algorithm 1. The infinite version of the Voronoi edge on which the vertex is projected, VE_{BD} in figure 3.3.1., divides the plane into two half-planes, each of them containing one of the Delone vertices related to the edge and one or two projected Delone edges belonging to level i. Due to the virtual displacement done by electing the apex (line 1), the edges contained in the half-plane where the apex lies, do not longer intersect VE_{BD} but the edges in the second half-plane properly do it (line 4). This "intersect and no-longer-intersect" status on each found intersection leads to a proper creation of the nodes related to the T_{12} tetrahedrons that complete a valid configuration (lines 3-7).

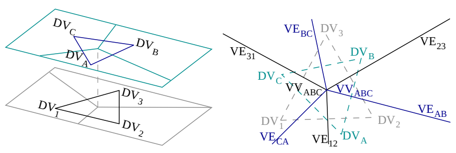
3.3.2. *Case 2: Voronoi vertex vs. Voronoi vertex*

This case is generated when six Delone vertices are co-spherical, leading to a Voronoi vertex belonging to level i be projected on a Voronoi vertex belonging to level j.

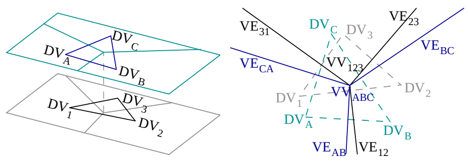
In this case nine intersections are identified, VE_{12} vs. VE_{AB}, VE_{12} vs. VE_{BC}, VE_{12} vs. VE_{CA}, VE_{23} vs. VE_{AB}, VE_{23} vs. VE_{BC}, VE_{23} vs. VE_{CA},

Algorithm 1 Solving Voronoi vertex vs. Voronoi edge Case

```
[$T_i$, $T_12$] = **solveCaseVerVsEdge**( $VV$, $VE$ )
    **Input**:            $VV$: Voronoi vertex
                          $VE$: Voronoi edge
    **Output**:           $T_i$ tetrahedron related to $VV$
                          $T_12$: set of at most two $T_{12}$ tetrahedrons
    **Precondition**:     level of $VV$ is not the same level of $VE$
                          the projection of $VV$ lies inside $VE$
    **Postcondition**:    the tetrahedrons in $T_i$ and $T_12$ form a valid configuration
1: $Apex$ = elect left or right vertex of $VE$
2: $T_i$ = new $T_i$ using the Delone triangle related to $VV$ and $Apex$
3: **for** every Voronoi edge $VE_k$ related to $VV$ **do**
4:   **if** half-plane of $VE_k$ is not the same half-plane of $Apex$ **then**
5:     $t_{12}$ = new $T_{12}$ created with the Delone edges related to $VE_k$ and $VE$
6:     add $t_{12}$ to $T_{12}$
7:   **end if**
8: **end for**
```



(a) *1a2b3c* sub-case



(b) *1ab23c* sub-case

Figure 6: Special cases in the creation of the Joint Voronoi Diagram: Voronoi Vertex vs. Voronoi Vertex case

VE_{31} vs. VE_{AB}, VE_{31} vs. VE_{BC}, VE_{31} vs. VE_{CA}. As in the previous case, the construction of these nine tetrahedrons, leads to an inconsistent graph. The "election of apex" problem is also present, with the detail that there are two T_i tetrahedrons to elect an apex, and three possible apices for each tetrahedron. When an apex for any of the T_i tetrahedrons is chosen, it restricts the election of the apex for the second T_i tetrahedron and the creation of the complementary T_{12} tetrahedrons. Because of this, this case allows three configurations as solutions.

As it happened in the previous case, the election of an apex could be translated into a virtual displacement of the levels and the elimination of the ambiguity by the assumption that the distance between the apex and the Voronoi vertex is the smallest.

Algorithm 2 Identifying Vertex vs. Vertex subcases

```
$subcase$ = idVerVsVerSubcase( $VV_i$, $VV_j$ )
    Input:           $VV_i$: Voronoi vertex on level $i$
                     $VV_j$: Voronoi vertex on level $j$
    Output:          $subcase$: Flag indicating the sub-case type,
                     its possible values are 1a2b3c or 1ab23c
    Precondition:    level of $VV_i$ is not the same level of $VV_j$
                     the projection of $VV_i$ lies on the projection of $VV_j$
    Postcondition:   A sub-case is identified
1: $Edges[\,]$ = angular order of all edges related to $VV_i$ and $VV_j$
2: $Subcase$ = 1a2b3c
3: for every Voronoi edge $VE_c$ in $Edges$ do
4:     set $VE_n$ as the edge next to $VE_c$ in $Edges$
5:     if level of $VE_c$ is level of $VE_n$ then
6:         $Subcase$ = 1ab23c
7:     end if
8: end for
```

Two different sub-cases are identified for this case, both keeping the same characteristics described above. Algorithm 2 identifies the subcases. The sub-cases are determined by the distribution of the edges on the "intersecting star" created when all the edges are projected on the same plane (see figure 3.3.2.). There are only two possible distributions, the edges are intercalated or they are not. When two consecutive edges belong to the same level, the sub-case is identified as the *1ab23c* sub-case (lines 5-7). Otherwise, if there are no two consecutive levels belonging to the same level, the sub-case is identified as the *1a2b3c* sub-case (the cycle in lines 3-9 never falls inside lines 5-7).

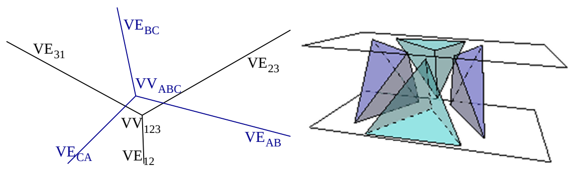


Figure 7: Special cases in the creation of the Joint Voronoi Diagram: A solution for *1a2b3c* sub-case, where DV_3 was elected as apex for the T_i tetrahedron related to VV_{123}

Identification of a valid configuration for the *1a2b3c* sub-case For this sub-case, each Voronoi region related to a Voronoi vertex contains a Voronoi edge related to the other Voronoi vertex (figure 6(a)). The three solutions for this sub-case are symmetric; the election of the apex for the first T_i tetrahedron does not change the fact that two T_{12} and two T_i tetrahedrons are created. In figure 7 the *joint Voronoi Diagram* with no ambiguity is shown, and also its physical tetrahedron representation.

Algorithm 3 implements the solution for this sub-case. The election of the apex for the first T_i is done in line 4. The vertex that lies in the region that is opposite to the first elected apex in the consecutive level is chosen as apex for the second T_i tetrahedron (line 6-8). The T_{12} tetrahedrons that complete the valid solution are created using the edges that bound the corresponding Voronoi regions of the elected apices (lines 11-13 and 15-17).

Identification of a valid configuration for the *1ab23c* sub-case For this sub-case, the solutions are not symmetric as they are in the previous sub-case. The solutions are shown in figure 8.

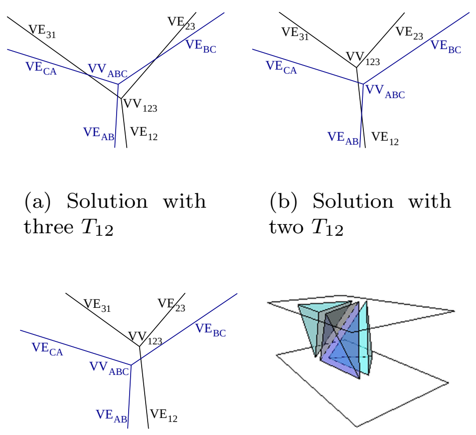
The simplest solution is shown in figure 8(c), where just one T_{12} tetrahedron is created. To construct that solution, some elements must be identified: the *Full Region* and the *Lone Edge*.

Full Region: The Voronoi region that contains two Voronoi edges belonging to the other level is named the *Full Region*. In figure 8(c), the *Full Region* for level i is the Voronoi region VR_1, bounded by VE_{12} and VE_{31}, and for

Algorithm 3 Voronoi vertex vs. Voronoi vertex $1a2b3c$ sub-case

```
[$T_i$, $T_{12}$] = **solveVerVsVer1a2b3c**( $VV_i$, $VV_j$ )
    Input:          $VV_i$: Voronoi vertex on level $i$
                    $VV_j$: Voronoi vertex on level $j$
    Output:         $T_i$: set of TWO $T_i$ tetrahedrons related to $VV_i$ and $VV_j$
                    $T_12$: set of TWO $T_{12}$ tetrahedrons
    Precondition:   level of $VV_i$ is not the same level of $VV_j$
                    the projection of $VV_i$ lies on the projection of $VV_j$
    Postcondition:  the tetrahedrons in $T_i$ and $T_12$ form a valid configuration

 1: $Edge_j$ = any Voronoi edge related to $VV_j$
 2: $Region_j$ = Voronoi region to the left of $Edge_j$
 3: $Apex_1$ = Delone vertex related to $Region_j$
 4: $t_i$ = new $T_i$ using the Delone triangle related to $VV_i$ and $Apex_1$
 5: add $t_i$ to $T_i$
 6: $Edge_i$ = Voronoi edge whose projection lies inside $Region_j$
 7: $Region_i$ = Voronoi region not bounded by $Edge_i$ on the level of $Edge_i$
 8: $Apex_2$ = Delone vertex related to $Region_i$
 9: $t_i$ = new $T_i$ using the Delone triangle related to $VV_j$ and $Apex_2$
10: add $t_i$ to $T_i$
11: $DE_i$ = Delone edge related to the Voronoi edge to the left of $Region_i$
12: $DE_j$ = Delone edge related to the Voronoi edge to the right of $Region_j$
13: $t_{12}$ = new $T_{12}$ using $DE_i$ and $DE_j$
14: add $t_{12}$ to $T_{12}$
15: $DE_i$ = Delone edge related to the Voronoi edge to the right of $Region_i$
16: $DE_j$ = Delone edge related to the Voronoi edge to the left of $Region_j$
17: $t_{12}$ = new $T_{12}$ using $DE_i$ and $DE_j$
18: add $t_{12}$ to $T_{12}$
```



(a) Solution with three T_{12}

(b) Solution with two T_{12}

(c) Solution with just one T_{12}

(d) Tetrahedrons for solution shown in (c)

Figure 8: Special cases in the creation of the Joint Voronoi Diagram: Solutions for the *1ab23c* sub-case

level j it is the Voronoi region VR_C, bounded by VE_{BC} and VE_{CA}.

Lone Edge: The Voronoi edge that is alone in a Voronoi region belonging to the other level is called the *Lone Edge*. In figure 8(c), the *Lone Edge* for level i is VE_{12}, and for level j it is VE_{BD}.

Algorithm 4 formalizes the solution for this sub-case. The valid construction is composed by the T_{12} tetrahedron related to the intersection of both Lone Edges (line 15), and the T_i tetrahedrons created by each Delone triangle related to a Voronoi vertex and the Delone vertex related to the Full Region of the other level used as the apex (lines 7-11). In figure 8(d) the tetrahedrons related to this solution are shown.

3.4. Elimination of tetrahedrons

The elimination of all the faces of the T_i tetrahedrons belonging to a non-solid connection leads to the creation of a incomplete surfaces. This defect is fixed in the version implemented as part of this project. When a T_i tetrahedron belonging to a non-solid connection is eliminated, a hole results in the place where its base stood. To avoid such holes, the horizontal triangles (bases) of the T_i tetrahedrons eliminated by non-solid connections are kept and included into the reconstructed

Algorithm 4 Solving Vertex vs. Vertex $1ab23c$ sub-case

```
[$T_i$, $T_{12}$] = **solveVerVsVer1ab23c**( $VV_i$, $VV_j$ )
    **Input**:          $VV_i$: Voronoi vertex on level $i$
                        $VV_j$: Voronoi vertex on level $j$
    **Output**:         $T_i$: set of TWO $T_i$ tetrahedrons related to $VV_i$ and $VV_j$
                        $T_12$: a $T_{12}$ tetrahedron
    **Precondition**:   level of $VV_i$ is not the same level of $VV_j$
                        the projection of $VV_i$ lies on the projection of $VV_j$
    **Postcondition**:  the tetrahedrons in $T_i$ and $T_12$ form a valid configuration
1: $Level_i$ = level of $VV_i$
2: $Edges[\ ]$ = angular order of all edges related to $VV_i$ and $VV_j$
3: [$Lone\_edge_i$, $Lone\_edge_j$] = **findLoneEdges**( $Edges$, $level_i$ )
4: [$Full\_region_i$, $Full\_region_j$] = **findFullRegions**( $Edges$, $level_i$ )
5: $base$ = Delone triangle related to $VV_i$
6: $apex$ = Delone vertex related to $Full\_region_j$
7: $t_i$ = new $T_i$ using $base$ and $apex$
8: add $t_i$ to $T_i$
9: $base$ = Delone triangle related to $VV_j$
10: $apex$ = Delone vertex related to $Full\_region_i$
11: $t_i$ = new $T_i$ using $base$ and $apex$
12: add $t_i$ to $T_i$
13: $DE_i$ = Delone edge related to $Lone\_edge_i$
14: $DE_j$ = Delone edge related to $Lone\_edge_j$
15: $T_{12}$ = new $T_{12}$ using $DE_i$ and $DE_j$
```

surface.

4. RESULTS

4.1. Skull

The Skull is a set of 258 contours, placed on 63 planes parallel to the XZ plane. The resulting surface is composed by 39.808 triangles. This set of contours present wide m-n branches specially in the levels between the nose and the eye holes. In figure 9 a detail of levels 29 and 30 is shown. In figure 10 more details may be observed.

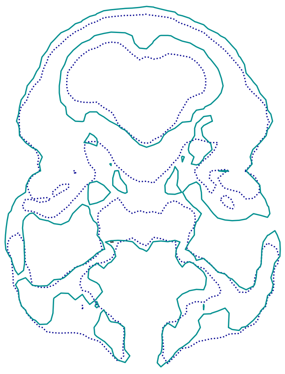


Figure 9: Detail of levels 29 and 30 of the set of contours “skull”

4.2. Brain

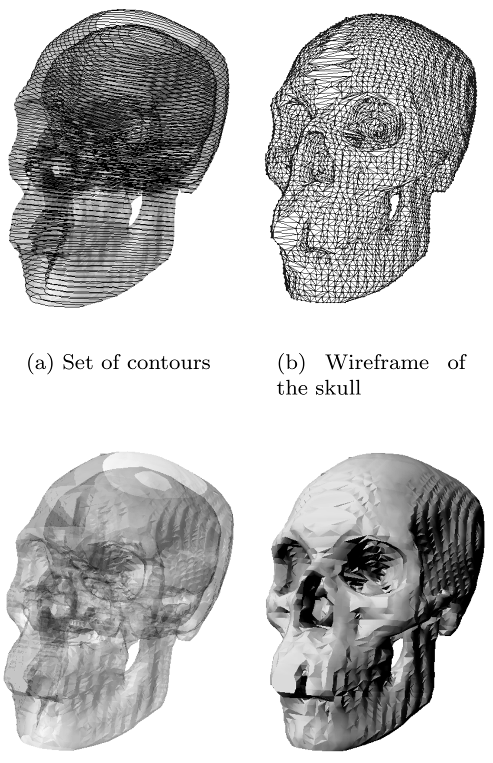
This set of contours is a complement given with the algorithm of the B+G method[1]. It is composed by 15 parallel levels, with 105 contours. The reconstructed surface has 13607 triangles. Figure 11 shows more details.

5. CONCLUSIONS

A method has been designed and implemented, to circumvent geometrical degeneracies arising of simultaneous processing of 2D superimpossed Voronoi Diagrams, in the context of Surface Reconstruction from Slice Samples. In this particular problem, for each degenerate condition an enumerable finite set of non-degenerate counterparts is programmed, and instantiated as the geometry of the degeneracy dictates. In absence of the algorithm, self intersecting and therefore non - manifold constructions follow. With the algorithm, degenerate cases are mapped to their non

[1]ftp://ftp-sop.inria.fr/prisme/NUAGES/Nuages/

- degenerate counterparts. This allows the normal downstream execution of the host algorithm (B+G, by Boissonnat & Geiger, 1988, 1993). The method presented classifies actions to be taken, based on the level of the degeneracy. The results show that the method is successful in removing the degeneracy, without further iterations, and in deterministic way. This method can be applied when the number of cases of the degeneracy is known.


(a) Set of contours (b) Wireframe of the skull

(c) Reconstructed surface in a transparent material (d) Reconstructed surface in concrete

Figure 10: Set of contours and the reconstructed surface

References

[1] J. D. Boissonnat. Shape reconstruction from planar cross-sections. *Computer Vision, Graphics and Image Processing*, pages 1–29, 1988.

[2] C. Burnikel, R. Fleischer, K. Mehlhorn, and S. Schirra. Efficient exact geometric computation made easy. In *Proceedings of the 15th Annual ACM Symposium on Computational Geometry*, 1999.

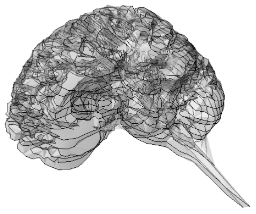
[3] H. Edelsbrunner and E. P. Mucke. Simulation of simplicity: a technique to cope with degenerate cases in geometric algorithms. *ACM Trans. Graphics*, (9):66–104, 1990.

[4] S. Funke, Ch. Klein, K. Mehlhorn, , and S. Schmitt. Controlled perturbation for delaunay triangulations. In *Proceedings of S.O.D.A*, 2005.

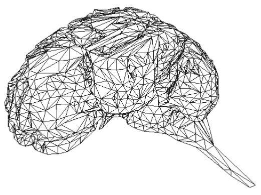
[5] B. Geiger. Three dimensional modeling of human organs and its application to diagnosis and surgical planning. Research Report 2105, INRIA, Sophia-Antipolis, Valbonne, France, 1993.

[6] O. Ruiz, C. Cadavid, and M. Granados. 2d similarity-complemented voronoi-delone methods in shape reconstruction. *Journal Anales de Ingenieria Grafica*, (15):16–24, 2002.

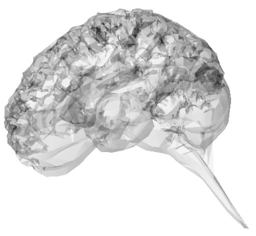
[7] O. Ruiz, C. Cadavid, M. Granados, S. Peña, and E. Vásquez. 2d shape similarity as a complement for voronoi-delone methods in shape reconstruction. *Elsevier Computer and Graphics*, 2005.



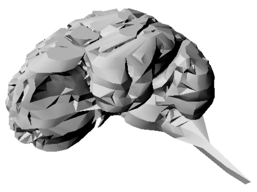
(a) Set of contours



(b) Wireframe of the brain



(c) Reconstructed surface in a transparent material



(d) Reconstructed surface in concrete

Figure 11: Set of contours and the reconstructed surface