Perturbations and Vertex Removal in Delaunay and Regular 3D Triangulations

Olivier Devillers, Monique Teillaud

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Perturbations and Vertex Removal in Delaunay and Regular 3D Triangulations

Olivier Devillers — Monique Teillaud

N° 5968

Septembre 2006

Thème SYM

Rapport de recherche

ISSN 0249-6399 ISRN INRIA/RR--5968--FR+ENG

Perturbations and Vertex Removal in Delaunay and Regular 3D Triangulations

Olivier Devillers, Monique Teillaud

Thème SYM — Systèmes symboliques
Projet Geometrica

Rapport de recherche n° 5968 — Septembre 2006 — 11 pages

Abstract: Though Delaunay and regular triangulations are very well known geometric data structures, the problem of the robust removal of a vertex in a three-dimensional triangulation is actually a problem in practice.

We propose a simple method that allows to remove any vertex even when the points are in very degenerate configurations. The solution is available in CGAL*.

Key-words: Delaunay, regular triangulation, degeneracies

INRIA 4624 is a preliminary version of the current report, this new version generalizes from Delaunay to regular triangualtion and gives precisions on convex hull management.

* `http://www.cgal.org`

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

Perturbation et suppression de sommet dans une triangulation régulière 3D

Résumé : Bien que la triangulation de Delaunay et la triangulation régulière soient des structures bien connues, supprimer un point de manière robuste reste un problème délicat en pratique.

Nous proposons une méthode simple pour la suppression d'un point dans une triangulation 3D régulière ou de Delaunay qui fonctionne dans tous les cas, même très dégénérés. La solution est disponible dans la bibliothèque CGAL[†].

Mots-clés : Delaunay, triangulation régulière, dégénérescences

[†] http://www.cgal.org

1 Introduction

Let us first briefly recall that the Delaunay triangulation of a set $\mathcal{S}$ of points in general position in $\mathbf{R}^3$ is the partition of the convex hull of the points consisting of the tetrahedra whose circumscribing ball does not contain any point of $\mathcal{S}$ in its interior.

The regular triangulation is a generalization of the Delaunay triangulation when the sites in $\mathcal{S}$ are spheres (also called *weighted points*). Given four sites (spheres) in $\mathcal{S}$, we consider their orthogonal sphere B. B is said to be *"empty"* if any site s in $\mathcal{S}$ is either disjoint of B or the angle in which s and B intersect is smaller or equal to $\frac{\pi}{2}$ (we say that s and B are *sub-orthogonal*; similarly, we define the term *over-orthogonal* for angles greater or equal to $\frac{\pi}{2}$). Given a tetrahedron T whose vertices are the centers of four sites of $\mathcal{S}$, we define the *sphere of* T as the sphere orthogonal to these four sites. A tetrahedron whose sphere is empty is said to be *regular*. The regular triangulation $\mathcal{RT}(\mathcal{S})$ is formed by all regular tetrahedra.

The general position assumption states that no four sites have coplanar centers and no five sites admit a common orthogonal sphere.

When $\mathcal{S}$ is not in general position, the sphere of a regular tetrahedron T can admit other orthogonal sites than the four sites defining T (which is a degeneracy), then it is called *weakly regular*. A tetrahedron T whose sphere is strictly sub-orthogonal to all other sites of $\mathcal{S}$ is called *strongly regular*. The facets of a regular (resp. weakly regular, strongly regular) tetrahedron are called regular (resp. weakly regular, strongly regular). The set of strongly regular tetrahedra is not a triangulation, it can be completed in several ways by a subset of the weakly regular tetrahedra.

When a vertex v is removed from the regular triangulation $\mathcal{RT}(\mathcal{S})$ of $\mathcal{S}$, the tetrahedra incident to v are removed, which creates a polyhedral hole, and the interior of this polyhedron H must be triangulated with regular tetrahedra. Triangulating H is exactly the inverse operation of inserting v in $\mathcal{RT}(\mathcal{S} \setminus \{v\})$. After the removal, the regular triangulation is the triangulation that would have been obtained if v had never been inserted.

When degeneracies occur, any of the possible regular triangulations of the set of cospherical points can be returned.

The difficulty arises when there are at least four sites s_1, s_2, s_3, s_4 whose centers p_1, p_2, p_3, p_4 lie on the boundary of H, that have a common orthogonal sphere and whose centers are coplanar. Indeed, in this case, for any site s_5 whose center is on H, the five sites s_1, s_2, s_3, s_4, s_5 have a common orthogonal sphere, and there are two possible triangulations of this set of sites, corresponding to the two different choices for the diagonal of the convex polygon p_1, p_2, p_3, p_4 which is a facet of H (or a sub-facet in the case when there are more than four sites whose centers are coplanar and having a common orthogonal sphere). Depending on this choice, we will get a different triangulation of this facet. In fact, there is no choice: since the outside of H is already triangulated, the triangulations of all the facets of H are given and must be respected.

Note that a similar problem cannot occur in the planar case, since the hole created by the removal of a vertex is a polygon, so, even in the case when this polygon has collinear vertices, there is no choice to be made: the edges of the polygon are uniquely determined by the order of the collinear vertices on the line that contains them. Any triangulation of the interior of the polygon will have the same edges on the polygon.

The problem can be seen as a special instance of the following question:

> *Is it always possible to compute a regular triangulation of a given polyhedron H, whose facets are weakly regular?* (1)

In this paper, we show that, though this general question has a negative answer, the more restrictive instance of the problem posed by the vertex removal from a 3D regular triangulation can be solved in a simple way, even in degenerate situations.

Paper Overview

After some remarks about previous work (Section 2), in Section 3 we discuss algorithmic issues and motivate our choices achieving robustness to degeneracies and genericity of the code while preserving a good efficiency. Section 5 focuses on symbolic perturbations, dealing with all cases including the points on the convex hull.

2 Previous Work

2.1 Algorithms.

The deletion of vertices in two dimensional triangulations has been widely studied from both theoretical and practical points of view. A degree k vertex v can be removed in optimal $\Theta(k)$ time [1, 5], but in practice people often use sub-optimal solutions in $O(k^2)$ or $O(k \log k)$ time [6].

In higher dimensions, the problem has received less attention since the complexity of the result can reach $\Omega(k^2)$.

The so-called *ear algorithm* [12, 6] can be used to remove a vertex from the Delaunay triangulation. The hole H is a simple polyhedron; the general idea of the algorithm is to add new tetrahedra one by one in a way that ensures that the hole stays simple; such a tetrahedron is called an *ear*. There are two kinds of ears: ears formed by two triangles on the boundary of the hole, that share an edge, and ears formed by three triangles incident to a vertex that has degree 3 on the boundary of the hole.

At each step, there may be several ears satisfying the Delaunay criterion; Devillers proves that the ear for which the removed point has the smallest power with respect to the circumscribing sphere belongs to the Delaunay triangulation [6], and he proposes to choose this ear. By maintaining a priority queue of the candidate ears, this yields a theoretical complexity of $O(f \log k)$, where f is the number of tetrahedra to be created.

This approach can be generalized to regular triangulations, since the proofs actually rely on the transformation between 3D Delaunay triangulations and convex hulls in 4D, which also holds for regular triangulations.

2.2 Perturbation Techniques.

A general approach to solve the degenerate cases is the use of symbolic perturbations [17, 10, 16]. The general idea is to make the problem dependent of a parameter ε such that:

- there exists $\varepsilon_0 > 0$ such that the parameterized problem is in general position for $\varepsilon \in (0, \varepsilon_0]$,
- if the original problem is in general position, the solution of the parameterized problem tends to the solution of the original problem when ε goes to zero,
- if the original problem is not in general position, the solution of the parameterized problem tends to something relevant when ε goes to zero by positive values.

More precisely, if $\mathcal{S}$ is not in general position, the regular triangulation is not uniquely defined and the aim of a perturbation technique is to select one of the possible regular triangulations.

Different perturbation methods can be used in that case: either the input of the problem can be perturbed or the definition of the problem can be slightly changed.

When using a perturbation, some properties can be lost in the result: to check whether they are satisfied, we must
- check that the parameterized solution satisfies the properties
- check that the properties are still true at the limit.

Perturbing the input can have very serious drawbacks: if the points move with ε [2], then a non flat tetrahedron can become flat at the limit. By *flat* tetrahedron, we mean a tetrahedron whose four vertices are coplanar. As explained in Section 4, we want to avoid this.

Edelsbrunner and Mücke write that using their perturbation technique for Delaunay triangulations is a "real pain" and suggest to use the transformation in convex hull in one dimension higher and perturb the computation of the convex hull [10].

We propose a variant that consists in changing only the radii of the 3D sites, which perturbs the points in 4D in the vertical direction only.

One important advantage is that the 3D points do not move, thus if a tetrahedron belongs to the limit solution, the same tetrahedron is regular for a set of non degenerate sites with same centers but different weights, thus this tetrahedron is non flat. Four sites with coplanar centers on the 3D convex hull will still be in degenerate position after the perturbation, since they are associated with a vertical facet in 4D that will stay vertical, but we will show in Section 5.3 that this case can still be handled.

For a Delaunay triangulation, the perturbation reduces to computing the regular triangulation for a set of sites with weight going to zero when ε goes to zero. This is an analogy with the "sliver exudation" method [4, 3] that consists in associating weights to points, chosen so that the *almost* flat tetrahedra that are unavoidable in a Delaunay triangulation disappear in the regular triangulation. In our case, the tetrahedra are not almost flat, but really flat, the weights are symbolic, and the triangulation we compute is a Delaunay triangulation without any flat tetrahedron. A drawback of the method is that property that is satisfied by Delaunay triangulations of non-degenerate sets of sites, but not by regular triangulations, might not be satisfied by the limit solution (e.g. [9]).

Another feature of the technique is that to assign to each site a function of ε as radius, we need a one-to-one map between sites and functions, which is usually achieved by indexing the sites. We will show that in fact we can replace the indexing by an ordering of the sites. Relying on an indexing or ordering of sites is both a drawback and an advantage: it is a drawback because the result depends of that ordering; it is an advantage because the ordering can be chosen freely to fit other purposes.

3 Algorithmic issues

3.1 Ear algorithms

The algorithm was quickly summarized in Section 2.1.

In order to cope with degeneracies, Devillers' method would need a perturbation of the comparison of the powers of a given point with respect to different ears. These power comparisons have higher algebraic degree than the predicates that are really needed by the computation of the regular triangulation, namely the *orient* and the *orthogonality* predicates (see Section 5). A perturbation of these power comparisons would have to be compatible with the perturbation of the *orthogonality* predicate, which would be quite involved.

Moreover, Devillers' algorithm is not really suitable for a general purpose library such as CGAL: The algorithms proposed by CGAL are parameterized by a *traits class* defining the basic geometric operations needed by the algorithms. The user can provides the algorithms with his/her own traits class. Therefore we need to keep the requirements for the traits class minimal. Devillers' algorithm would require an additional predicate for comparing the powers of a point with respect to two spheres, that is not necessary for the construction of the regular triangulation.Thus, though this algorithm could be coded very efficiently as a stand-alone program, its integration in CGAL or in any library providing genericity and robustness is somehow problematic. It seems to be better to use only the predicates that are intrinsic to the regular triangulation, namely the *orthogonality* and the *orient* predicates, and to avoid any construction.

3.2 Variant

Rather than using the previous algorithm, we could use a non-optimal variant that consists in maintaining dynamically a list of candidate ears and testing if their orthogonal ball is empty by applying the *orthogonality* predicate to *all the sites* of the hole.

When an ear is found to belong to $\mathcal{RT}(\mathcal{S} \setminus \{v\})$, then the hole and the list of candidate ears are updated.

If the regular tetrahedra are added by increasing powers, then they are ears on the hole at the moment they are added. In the variant, the tetrahedra are added in a different order. Let us denote H_* the hole at some stage of the algorithm and T the regular tetrahedron of smallest power inside H_*. If we consider H_T the hole before the addition of T in the original algorithm then T is an ear of H_T, thus T is also an ear of H_* since $H_* \subset H_T$, which proves that there is at least one regular tetrahedron T which is an ear. So, in all cases, there will exist regular ears in the list of candidate ears at each stage.

The number of candidates during the whole algorithm is $O(f)$, the test takes $O(k)$ for each ear, thus the whole complexity is $O(fk)$, where, as in Section 2.1, k is the degree of the removed vertex and f is the number of created tetrahedra.

Since both f and k are small constants in general, this method will be quite efficient in practice. For example, the expected value of k when the input points follow a Poisson distribution is $\frac{96}{70}\pi^2 + 2 \simeq 15.5$ and the expected value of f is $\frac{72}{35}\pi^2 \simeq 20$ [15].

3.3 Sewing algorithm

A naive algorithm consists in simply computing the regular triangulation of the sites of the hole H, then removing from this small triangulation the tetrahedra that are outside H and finally *"sewing"* this triangulation in the hole. Though this approach leads to a worst-case asymptotic optimal algorithm, it must be noticed that not only the hole is triangulated, but the whole convex hull of this hole, so, many tetrahedra are constructed and then deleted, which is useless memory management work. Still, this method was proved to perform better than the variant of the ear algorithm presented in section 3.2, so, it is the method that is currently implemented in CGAL.

In the sequel, we focus on the perturbation used for solving degeneracies rather than on the algorithmic issues.

- The perturbation method presented in Section 5 consists in adding a symbolic value to the radius of each site. It does not create any flat tetrahedron. Allowing flat tetrahedra would have allowed us to use more trivial algorithms, but, as mentioned in Section 4, we consider this as unacceptable both for theoretical and practical reasons.
- We remark that the perturbation defines the regular triangulation uniquely even for degenerate configurations.

The method is implemented in CGAL. As far as we know, CGAL is the only publicly available software proposing a fully dynamic 3D Delaunay and regular triangulations.

4 Observations

4.1 Repairing the triangulation outside the hole is difficult.

Let us consider a straight prism H with triangular basis such that its six vertices are cospherical. Assume that its rectangular facets are triangulated as shown in Figure 1. Let us now try to triangulate the interior of H. The six vertices of H are exactly in the same configuration regarding their incidences on H. Take one of them, say p *wlog*, then it can easily be seen that any possible tetrahedron having this vertex and any other three vertices of H will have an edge that crosses an existing edge on H.

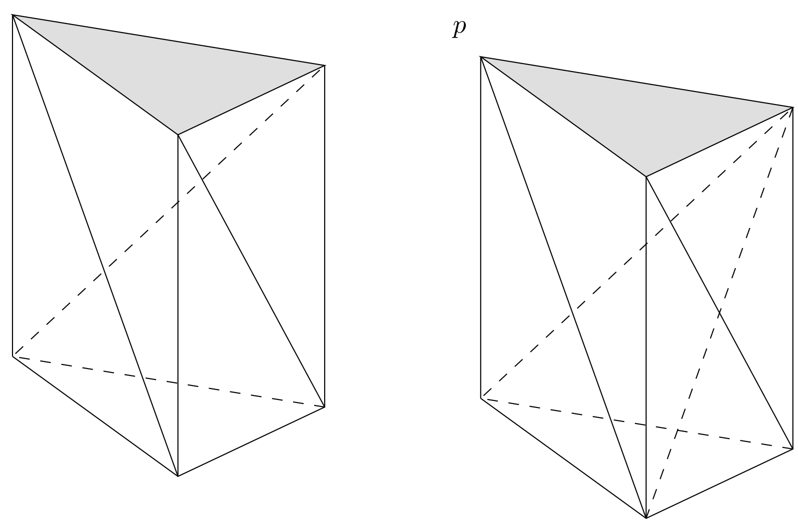

Figure 1: (1) has negative answer

Going back to the vertex removal in a 3D regular triangulation, if the set of tetrahedra incident to a vertex v form the polyhedron H, then the previous remarks show that, when v is removed, whatever choice will be made, retriangulating the hole H creates edges that cross edges of the tetrahedra outside the hole.

A strategy could consist in trying to flip tetrahedra outside H as suggested by Ledoux et al. [14]. However, the flips could propagate far away and the whole triangulation could possibly be recomputed, for instance on input data that would consist of gluing together polyhedra like the polyhedron of Figure 1.

4.2 Incremental Construction.

At first sight, the example of Figure 1 seems to prove that avoiding such edge crossings is impossible. In fact, this case cannot be built by an incremental algorithm coded as follows: As mentioned, the only problematic case is when there are more than 4 sites having a ommon orthogonal sphere (and whose centers are coplanar at the same time). When a site s is inserted, the set of tetrahedra ***conflicting*** with s (defined as the tetrahedra whose sphere is strictly over-orthogonal to s) is determined, and these tetrahedra are deleted from the triangulation. If the algorithm considers as non-conflicting the tetrahedra whose sphere is orthogonal to s, and thus if it does not delete these tetrahedra, then we get a unique construction of a regular triangulation. This triangulation is unique for a given order of the sites, but it depends on the order of insertion of the sites.

Assuming that the configuration shown in Figure 1 has been constructed by this incremental algorithm leads to the conclusion that the order of insertion of the points is cyclic, which is of course impossible. So, such a configuration cannot occur in our case.

Moreover, this incremental construction does not create any flat tetrahedron: Let us consider the case of a Delaunay triangulation. If the point p to be inserted is coplanar with a triangle t that is a facet of the Delaunay triangulation, then the two tetrahedra incident to t will have the same conflict status with respect to p, which means that either they will both stay, and t will still be their common facet in the updated triangulation, or they will both be deleted and t will disappear. Thus, p will not form a flat tetrahedron together with t. The proof generalizes to the case of a regular triangulation: the spheres that are orthogonal to the two spheres B_1 and B_2 of the two tetrahedra incident to t form a pencil $\mathcal{P}$ of spheres, which is also the pencil of spheres generated by the three sites centered at the vertices of t; the spheres of $\mathcal{P}$ are centered on the plane of t, and each sphere centered in this plane and orthogonal to B_1 is also orthogonal to B_2.

Our problem can then be more precisely rephrased as: assuming that the regular triangulation is constructed with the incremental algorithm considering each last inserted site s as not conflicting with tetrahedra whose sphere is orthogonal to s, find the regular triangulation of the hole that would have been computed if the removed point had never been inserted and if the other points had been inserted in the same order.

4.3 Allowing Flat Tetrahedra.

Going back to the example in Figure 1, the new dashed edge that would be created to triangulate the prism would cross an edge of a tetrahedron that is exterior to the hole. The four vertices of these two crossing edges can be seen as a flat tetrahedron that could be added to the triangulation in order to make it combinatorially valid.

In the same way, if there are $k \geq 4$ sites with a commun orthogonal sphere and whose centers lie on some facet of the hole created when removing a vertex of the regular triangulation, the regular triangulation of the sites of the hole is not uniquely defined, and retriangulating it may lead to two different triangulations of the same facet: one given by the interior tetrahedra, and one given by the exterior tetrahedra. It is well known that any triangulation in the plane can be obtained from any other triangulation having the same vertices by a sequence of $O(k^2)$ flips of diagonals of convex quadrilaterals [13] (in the case of a Delaunay triangulation, the k vertices are in convex position and the number of flips goes down to $O(k)$). Each such edge flip in 2D can be seen in 3D as a flat tetrahedron formed by the four *(coplanar)* vertices of the quadrilateral. Adding the $O(k^2)$ flat tetrahedra would yield a combinatorially valid triangulation.

Thus, allowing flat tetrahedra would allow us to get a simple solution to the problem. However, we do not consider this as an acceptable solution for the CGAL users. Indeed, these flat tetrahedra correspond neither with the definition of a regular triangulation, since the sphere of a flat tetrahedron is not uniquely defined, nor to the usual intuition. Moreover, in practice, this would lead to a heavier user code: before applying geometric operations to a tetrahedron, such as computing its circumcenter, the user would have to check that the tetrahedron is not flat.

Since we know that a regular triangulation without any flat tetrahedron always exists (it can be computed for instance using the incremental construction seen in Section 4.2), our goal is to find a method that does not create any flat tetrahedron.

5 Perturbing the *orthogonality* Predicate

5.1 The Perturbation.

The *in_ sphere* predicate is the basic predicate used when inserting or removing a point in a Delaunay triangulation.

Let p_0, p_1, p_2, p_3 be four non-coplanar points in $\mathbf{R}^3$.

$$in\_sphere\ (p_0, p_1, p_2, p_3, p_4) \quad \left.\begin{array}{ll} > 0 & \text{if } p_4 \text{ is outside} \\ = 0 & \text{if } p_4 \text{ is on the boundary of} \\ < 0 & \text{if } p_4 \text{ is inside} \end{array}\right\} \text{ the ball circumscribing } p_0, p_1, p_2, p_3.$$

For a regular triangulation, and s_0, s_1, s_2, s_3, s_4 four spheres with non-coplanar centers, the *orthogonality* predicate generalizes *in_ sphere*

$$orthogonality\ (s_0, s_1, s_2, s_3, s_4) \quad \left.\begin{array}{ll} > 0 & \text{if } s_4 \text{ is strictly sub-orthogonal} \\ = 0 & \text{if } s_4 \text{ is orthogonal} \\ < 0 & \text{if } s_4 \text{ is strictly over-orthogonal} \end{array}\right\} \text{ to the sphere orthogonal to } s_0, s_1, s_2, s_3.$$

Let us notice straightaway that, if the triangulation admits no flat tetrahedron, then this predicate will be needed only when the points s_0, s_1, s_2, s_3 have actually non coplanar centers, and that no degenerate case has to be considered here: when inserting a new site s_4, it is only tested against tetrahedra that are already existing in the triangulation; this is also the case when filling the hole formed by the removal of another site, because we only consider ears of the hole formed by non coplanar facets, since the goal triangulation does not contain any flat tetrahedron.

The decision whether to create or delete a tetrahedron (s_0, s_1, s_2, s_3) in the regular triangulation is easy when all the *orthogonality* tests against all sites s_4 give > 0 and < 0 results. When 0 results appear, decisions must be made using other criteria.

Notice that coding the insertion as mentioned in Section 4.2, by considering that the last inserted site is not conflicting with tetrahedra such that the result of the *orthogonality* test is 0 (i.e. considering that the site is over-orthogonal to the sphere of the tetrahedra), can be seen as using an implicit symbolic perturbation. It will be clear later that this perturbation corresponds to the perturbation proposed in the sequel.

It is well known that the *orthogonality* test can be computed in the following way:

orthogonality $(s_0, s_1, s_2, s_3, s_4) =$

$$sign\ \frac{Det(s_0, s_1, s_2, s_3, s_4)}{orient(p_{0,1}, p_2, p_3)}$$

where, if site s_i has center $p_i = (x_i, y_i, z_i)$ and radius r_i for each i,

$$orient(p_0, p_1, p_2, p_3) = \begin{vmatrix} 1 & 1 & 1 & 1 \\ x_0 & x_1 & x_2 & x_3 \\ y_0 & y_1 & y_2 & y_3 \\ z_0 & z_1 & z_2 & z_3 \end{vmatrix}$$

and

$$Det(s_0, s_1, s_2, s_3, s_4) = \begin{vmatrix} 1 & 1 & 1 & 1 & 1 \\ x_0 & x_1 & x_2 & x_3 & x_4 \\ y_0 & y_1 & y_2 & y_3 & y_4 \\ z_0 & z_1 & z_2 & z_3 & z_4 \\ x_0^2+y_0^2+z_0^2-r_0^2 & x_1^2+y_1^2+z_1^2-r_1^2 & x_2^2+y_2^2+z_2^2-r_2^2 & x_3^2+y_3^2+z_3^2-r_3^2 & x_4^2+y_4^2+z_4^2-r_4^2 \end{vmatrix}$$

The *sign* $Det(s_0, s_1, s_2, s_3, s_4)$ predicate in $\mathbf{R}^3$ can be seen as an orientation predicate in $\mathbf{R}^4$, if each site $s = ((x, y, z), r)$ of $\mathbf{R}^3$ is mapped onto a point $\pi(s) = (x, y, z, t)$ of $\mathbf{R}^4$ with $t = x^2 + y^2 + z^2 - r^2$ [11, 7].

Five sites in $\mathbf{R}^3$ have a common orthogonal sphere if and only if their images by π lie in the same hyperplane of $\mathbf{R}^4$. The symbolic perturbation of the ***orthogonality*** test consists in adding respectively some value to the fourth coordinate of $\pi(s_0), \pi(s_1), \pi(s_2), \pi(s_3), \pi(s_4)$ so that these points are not in the same hyperplane any more in $\mathbf{R}^4$. Then the predicate answers positive or negative instead of zero.

Let $\mathcal{S}$ be the set of n sites $\{s_0, s_1, \ldots, s_{n-1}\}$ in $\mathbf{R}^3$. We add here ε^{n-i} to the fourth coordinate of the point $\pi(s_i)$ of $\mathbf{R}^4$. The quantity each point is perturbed with depends on its index. The way the sites are indexed will be discussed in Section 5.2.

$Det(s_i, s_j, s_k, s_l, s_m)$ is perturbed into

$$Det_\varepsilon(s_i, s_j, s_k, s_l, s_m) = \begin{vmatrix} 1 & 1 & 1 & 1 & 1 \\ x_i & x_j & x_k & x_l & x_m \\ y_i & y_j & y_k & y_l & y_m \\ z_i & z_j & z_k & z_l & z_m \\ t_i+\varepsilon^{n-i} & t_j+\varepsilon^{n-j} & t_k+\varepsilon^{n-k} & t_l+\varepsilon^{n-l} & t_m+\varepsilon^{n-m} \end{vmatrix}$$

where

$$t_\star = x_\star^2 + y_\star^2 + z_\star^2 - r_\star^2 \text{ for } \star = i, j, k, l, m$$

Developing with respect to the last raw yields a polynomial in ε

$$\begin{aligned} Det_\varepsilon(s_i, s_j, s_k, s_l, s_m) = \quad & Det(s_i, s_j, s_k, s_l, s_m) + \textbf{\textit{orient}}(p_i, p_j, p_k, p_l)\varepsilon^{n-m} - \textbf{\textit{orient}}(p_i, p_j, p_k, p_m)\varepsilon^{n-l} \\ & + \textbf{\textit{orient}}(p_i, p_j, p_l, p_m)\varepsilon^{n-k} - \textbf{\textit{orient}}(p_i, p_k, p_l, p_m)\varepsilon^{n-j} + \textbf{\textit{orient}}(p_j, p_k, p_l, p_m)\varepsilon^{n-i} \end{aligned}$$

From now on, the five points p_i, p_j, p_k, p_l, p_m will be assumed to be non coplanar. As noticed at the beginning of this section, this assumption makes no restriction in practice since this is always the case when ***orthogonality*** is used.

When s_i, s_j, s_k, s_l, s_m have a common orthogonal sphere, the constant term $Det(s_i, s_j, s_k, s_l, s_m)$ of this polynomial vanishes to zero. The tested tetrahedron of the regular triangulation is non flat, and one of the coefficients of the polynomial is the ***orient***() determinant on its vertices, so, the polynomial Det_ε is not identically zero. In the case of a Delaunay triangulation, at most one of the coefficients ***orient***() can be zero, otherwise two subsets with four points would consist of four coplanar points: either the five points would be coplanar, which contradicts our hypothesis, or the three common points of the two subsets would be collinear, which is impossible since these three points are cospherical.

The coefficients of Det_ε are examined in order of the exponents of ε, until the first non null coefficient is found, which determines the sign of $Det_\varepsilon(s_i, s_j, s_k, s_l, s_m)$.

5.2 Indexing the Sites.

The indexing of sites mentioned in Section 4.2 is the order of insertion in the triangulation. This choice corresponds to the implicit perturbation used for the insertion: at each step, the last point is more perturbed than all the previous ones, so, when it has a common orthogonal sphere with the four sites defining an already existing tetrahedron, it is considered as being sub-orthogonal to the sphere of the tetrahedron.

This was first the choice made in CGAL for the Delaunay triangulation. Some users reported this choice as being annoying for their application [8]. In fact, any other way of indexing the sites can be used. We changed the indexing in the following release and chose the lexicographical order on the points, or any other (such as the comparison of memory addresses). This change was straightforward in the vertex removal since it was already coded using the perturbation scheme, but it required a change in the insertion: the perturbation had to be made explicit. The same choice was made recently when the vertex removal in the regular triangulation was implemented.

Choosing the lexicographical ordering of points has the advantage of giving a unique definition of the regular triangulation, even in degenerate cases. On the other hand, it may lead to a slower construction of the triangulation for some very degenerate input. We may think of leaving the choice of the order to the user.

5.3 Convex Hull Managment.

The problems of sites whose centers lie on the convex hull were only briefly mentioned in Section literature-perturb.

In CGAL the unbounded cell is subdivided into tetrahedra by considering that each convex hull facet is incident to an infinite cell having as fourth vertex an auxiliary vertex called the *infinite vertex*. In that way, each facet is incident to exactly two cells and special cases at the boundary of the convex hull are simple to deal with. The triangulations that are manipulated are triangulations of the combinatorial sphere $\mathcal{S}^3$.

The definition of *in_ sphere* used for the Delaunay triangulation is then extended to infinite cells. For four points p_0, p_1, p_2, p_3 with positive orientation, and a fifth point p_4,

$$in\_sphere\ (p_0, p_1, p_2, p_3, p_4) = sign\ Det(p_0, p_1, p_2, p_3, p_4).$$

We define *in_ sphere* $(p_0, p_1, p_2, \infty, p_4)$ as the limit of the sign of $Det(p_0, p_1, p_2, p_3, p_4)$ when p_3 goes to infinity (staying in the same half-space defined by p_0, p_1, p_2). Geometrically, the "ball" circumscribing p_0, p_1, p_2 and ∞ is the open half-space defined by p_0, p_1, p_2 together with the open disk circumscribing them. The "sphere" circumscribing p_0, p_1, p_2 is thus reduced to the circumscribing circle of p_0, p_1, p_2. The actual implementation of *in_ sphere* uses this geometric interpretation, looking at the side of p_4 with respect to the plane of p_0, p_1, p_2, and its side with respect to the circle in the case of coplanarity.

Note that if *in_ sphere* $(p_0, p_1, p_2, \infty, p_4)$ was defined as the sign of the limit (instead of the limit of the sign) of $Det(p_0, p_1, p_2, p_3, p_4)$, the "sphere" circumscribing p_0, p_1, p_2 would also contain all points of the plane of p_0, p_1, p_2, even the points in the circle. So, in the incremental algorithm, the implicit perturbation based on the order of insertion of the points would consider a coplanar point as non conflicting and would thus create flat tetrahedra on the boundary of the convex hull.

The perturbation scheme explained in Section 5 is then applied only on the predicate of comparison of a point with a circle defined by three points of the convex hull (and not on the test of coplanarity).

Similarly, for a regular triangulation, *orthogonality* $(s_0, s_1, s_2, \infty, s_4)$ is the limit of *sign* $Det(s_0, s_1, s_2, s_3, s_4)$ when the center of s_3 goes to infinity, while its radius stays finite. The geometric interpretation generalizes the interpretation above for Delaunay triangulations.

6 Conclusion

We proposed a method for removing a vertex from a regular triangulation, that works even in degenerate situations.

This method uses a symbolic perturbation technique that does not produce any flat tetrahedron. The code for perturbing the *orthogonality* predicate is quite simple.

The solution is implemented in CGAL. The Delaunay triangulation is dynamic since release 2.3, the vertex removal in the regular triangulation was integrated in release 3.2.

Acknowledgments.

The authors wish to thank Mariette Yvinec for the original idea of the sewing algorithm, Jeff Erickson for informal discussions at the very beginning of this research, Andreas Fabri, Sylvain Pion and Christophe Delage for coding various aspects of the work presented in this paper in the CGAL 3D triangulation package, CGAL users, especially Tamal Dey and his students, for encouraging us in improving CGAL.

References

[1] A. Aggarwal, L. J. Guibas, J. Saxe, and P. W. Shor. A linear-time algorithm for computing the Voronoi diagram of a convex polygon. *Discrete Comput. Geom.*, 4(6):591–604, 1989.

[2] Pierre Alliez, Olivier Devillers, and Jack Snoeyink. Removing degeneracies by perturbing the problem or the world. *Reliable Computing*, 6:61–79, 2000.

[3] Siu-Wing Cheng and Tamal K. Dey. Quality meshing with weighted Delaunay refinement. In *Proc. 13th ACM-SIAM Sympos. Discrete Algorithms*, pages 137–146, 2002.

[4] Siu-Wing Cheng, Tamal K. Dey, Herbert Edelsbrunner, Michael A. Facello, and Shang-Hua Teng. Sliver exudation. In *Proc. 15th Annu. ACM Sympos. Comput. Geom.*, pages 1–13, 1999.

[5] L. P. Chew. Building Voronoi diagrams for convex polygons in linear expected time. Technical Report PCS-TR90-147, Dept. Math. Comput. Sci., Dartmouth College, Hanover, NH, 1986.

[6] Olivier Devillers. On deletion in Delaunay triangulation. *Internat. J. Comput. Geom. Appl.*, 12:193–205, 2002.

[7] Olivier Devillers, Stefan Meiser, and Monique Teillaud. The space of spheres, a geometric tool to unify duality results on Voronoi diagrams. In *Proc. 4th Canad. Conf. Comput. Geom.*, pages 263–268, 1992.

[8] Tamal K. Dey and Joachim Giesen. Detecting undersampling in surface reconstruction. In *Proc. 17th Annu. ACM Sympos. Comput. Geom.*, pages 257–263, 2001.

[9] M. B. Dillencourt. Realizability of Delaunay triangulations. *Inform. Process. Lett.*, 33(6):283–287, February 1990.

[10] H. Edelsbrunner and E. P. Mücke. Simulation of simplicity: A technique to cope with degenerate cases in geometric algorithms. *ACM Trans. Graph.*, 9(1):66–104, 1990.

[11] H. Edelsbrunner and R. Seidel. Voronoi diagrams and arrangements. *Discrete Comput. Geom.*, 1:25–44, 1986.

[12] M. Heller. Triangulation algorithms for adaptive terrain modeling. In *Proc. 4th Internat. Sympos. Spatial Data Handling*, pages 163–174, 1990.

[13] F. Hurtado, M. Noy, and J. Urrutia. Flipping edges in triangulations. *Discrete Comput. Geom.*, 22(3):333–346, 1999.

[14] Hugo Ledoux, Christopher M. Gold, and George Baciu. Flipping to robustly delete a vertex in a delaunay tetrahedralization. In Osvaldo Gervasi, Marina L. Gavrilova, Vipin Kumar, Antonio Laganà, Heow Pueh Lee, Youngsong Mun, David Taniar, and Chih Jeng Kenneth Tan, editors, *Computational Geometry and Applications Workshop*, volume 3480 of *Lecture Notes Comput. Sci.*, pages 737–747. Springer-Verlag, 2005.

[15] Atsuyuki Okabe, Barry Boots, and Kokichi Sugihara. *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*. John Wiley & Sons, Chichester, UK, 1992.

[16] R. Seidel. The nature and meaning of perturbations in geometric computing. *Discrete Comput. Geom.*, 19:1–17, 1998.

[17] C. K. Yap. A geometric consistency theorem for a symbolic perturbation scheme. *J. Comput. Syst. Sci.*, 40(1):2–18, 1990.

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

Unité de recherche INRIA Futurs : Parc Club Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 ORSAY 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 - 38334 Montbonnot Saint-Ismier (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