Output sensitive construction of the 3D Delaunay triangulation of constrained sets of points*

J.-D. Boissonnat† **A. Cérézo**‡ **O. Devillers**† **M. Teillaud**†

Abstract

In this paper, two algorithms are presented to compute the Delaunay triangulation of a set S of n points in 3-dimensional space when the points lie on a set $\mathcal{P}$ of k planes. If $k = 2$ the algorithm runs in time $O(n \log n + t)$ where t is the size of the output : if $k \geq 3$ the time bound is $O(tk \log n)$. In both cases, the storage is $O(n)$.

1 Introduction

The Delaunay triangulation and its dual, the Voronoï diagram, are fundamental structures in computational geometry. In two dimensions, there exist several optimal algorithms to compute such structures. In higher dimensions, the size of the output depends on the input distribution, the complexity of the algorithms may depend on the input size n and the output size t, which may vary from $O(n)$ to $O(n^{\lfloor \frac{d+1}{2} \rfloor})$. Seidel gives an $O(n^{\lceil \frac{d+1}{2} \rceil})$ algorithm which is worst-case optimal in odd dimensions ; [8] describes an output sensitive algorithm running in time $O(n^2 + t \log n)$. Unfortunately, in three dimensions the Delaunay triangulation of a set S of n sites may have $t = \Omega(n^2)$ tetrahedra, so a worst case optimal algorithm gives a quadratic complexity $O(n^2)$ which is not really interesting if t is significantly less than $O(n^2)$. One of the main open questions related to the Delaunay triangulation asks for the existence of an output sensitive algorithm computing the Delaunay triangulation in optimal time $O(n \log n + t)$. At this time there exist incremental algorithms whose complexity is randomized and sensitive to the size of the successive triangulations [5,3]. There exists also a deterministic algorithm which reaches the optimal time bound in the special case of two parallel planes [1].

This paper presents two algorithms to compute the Delaunay triangulation of a set S of n points in 3-dimensional space when the points lie on a set $\mathcal{P}$ of k planes. The algorithm for the case of two planes runs in time $O(n \log n + t)$. If $k \geq 3$ the time complexity is $O(tk \log n)$. In both cases, the storage is $O(n)$.

This algorithm is especially attractive when the number of planes is small with respect to the number of points. This is in particular true when dealing with tomographic images and 3-dimensional shape reconstruction problems [1].

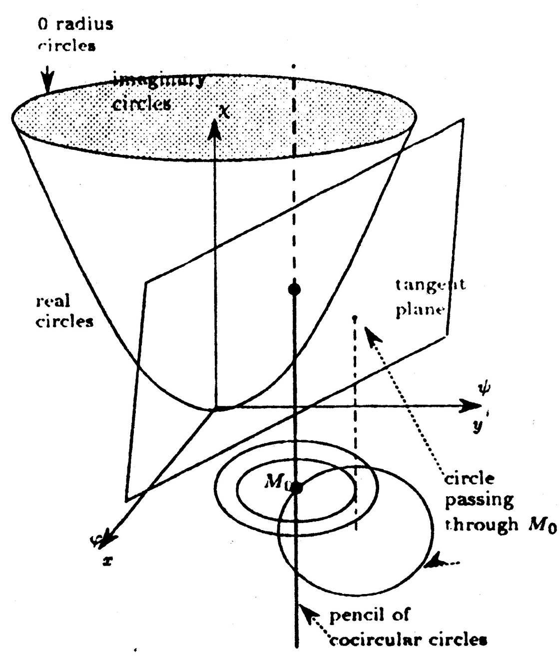

Figure 1: The space of circles

2 Preliminary results

2.1 Space of circles and pencils of circles

We take interest in the set of circles drawn in plane P. This set of circles is represented by a three dimensional set $\mathcal{C}_P$: the space of circles. Let $x^2 + y^2 - 2\varphi x - 2\psi y + \chi = 0$ be the equation of a circle in P ; this circle is represented by the point (φ, ψ, χ) in $\mathcal{C}_P$.

We present in this section the properties of $\mathcal{C}_P$.

- The set of circles of radius 0 is a paraboloid of equation : $\varphi^2 + \psi^2 = \chi$. A point in P is often identified with the 0 radius circle centered at this point. If plane P is identified to the plane $\chi = 0$ then the 0 radius circle is obtained by raising the point on the paraboloid. The interior of the paraboloid is the set of "imaginary" circles (with negative square radius) (see Figure 1). Notice that the horizontal projection

---

†INRIA, B.P.109, 06561 Valbonne cedex (France),
‡Université de Nice, Parc Valrose, 06034 Nice cedex (France)
*This work has been supported in part by the ESPRIT Basic Research Action Nr. 3075 (ALCOM).

of a point in C_P is the center of the circle, and the χ-coordinate is the power of the origin with respect to the circle.

- Two circles $(\varphi_1, \psi_1, \chi_1)$ and $(\varphi_2, \psi_2, \chi_2)$ are perpendicular if the corresponding points are conjugate with respect to the paraboloid or equivalently if $\chi_1 + \chi_2 = 2\varphi_1\varphi_2 + 2\psi_1\psi_2$.

- As a particular case, the set of circles passing through $M_0 \in P$ is also the set of circles perpendicular to the 0 radius circle M_0, which is in C_P the polar plane of M_0. But, as M_0 belongs to the paraboloid, the polar plane is nothing else than the tangent plane to the paraboloid at M_0. For a circle in the half space (limited by this plane) which does not contain the paraboloid, M_0 is inside the circle, For a circle in the other half space, M_0 is outside.

- A pencil of circles, that is the set of linear combinations of two circles, transforms in C_P into the line through the two points. A pencil of circles with limit points is a line hitting the paraboloid in the two limit points. A pencil of circles with base points is a line which does not hit the paraboloid, this line is the intersection of the two planes tangent to the paraboloid at the base points. A concentric pencil is a line parallel to the χ axis (see Figure 1).

- The points at infinity of the projective closure of C_P correspond to the straight lines in P, namely the point at infinity in the direction of (φ, ψ, χ) is the straight line of equation $-2\varphi x - 2\psi y + \chi = 0$, and the point at infinity of a line in C_P is the radical axis of the corresponding pencil.

  Thus the pencils whose radical axis is a given line in P form in C_P the set of lines parallel to a given direction (orthogonal to the radical axis).

  In particular, a pencil of circles having the x-axis as its radical axis corresponds in C_P to a line parallel to the ψ-axis.

2.2 Empty circles and Voronoï diagrams

As noticed in the preceding section, the set of circles which do not contain a given point maps in C_P into an half space, limited by the polar plane of the point. Thus, if $\mathcal{S}_P$ is a set of sites in plane P, then the set of empty circles (i.e. the set of circles which do not surround any site of $\mathcal{S}_P$) of plane P is in C_P the intersection of the corresponding half spaces. It is a convex polyhedron $U_{\mathcal{S}_P}$, whose facets are tangent to the paraboloid. The intersection of this convex set with a line, representing a pencil of circles is made up of the extremal empty circles of this pencil (see Figure 2). In general there are zero or two such extremal circles. These circles are the empty circles passing through a point of $\mathcal{S}_P$. Because there is exactly one circle of a pencil passing through a given point, the problem of the determination of the extremal empty circles of a given pencil can be viewed equivalently, as the determination of the points of $\mathcal{S}_P$ on these circles.

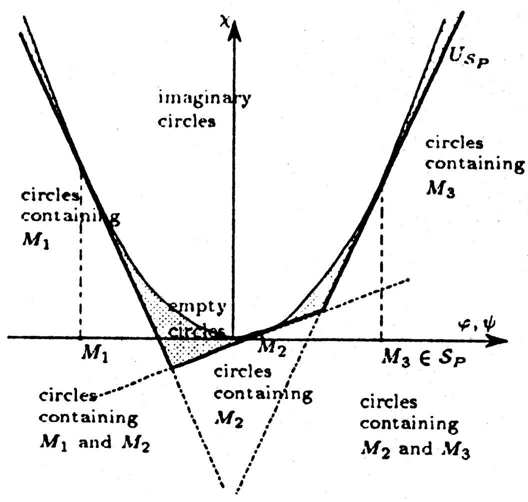

Figure 2: Set of empty circles

In particular, for a concentric pencil, the intersection with a vertical line in C_P gives the largest empty circle having a given center. This circle passes through a site of $\mathcal{S}_P$, namely the site of $\mathcal{S}_P$ which is the closest to the center. In other words the projection of $U_{\mathcal{S}_P}$ on the horizontal plane is the Voronoï diagram of $\mathcal{S}_P$. This can be viewed as a new interpretation of the well known correspondance between intersection of half spaces tangent to the paraboloid and Voronoï diagram [6].

2.3 A query problem

Lemma 1 *Let $\mathcal{S}_P$ be a set of sites in plane P, and F be a pencil of circles, the extremal empty circles of F can be found in $O(\log n)$ time using $O(n)$ space and $O(n \log n)$ preprocessing time.*

Proof. The algorithm first constructs $U_{\mathcal{S}_P}$ and then computes in C_P the intersection of the line representing F with the convex $U_{\mathcal{S}_P}$, which can be done within the given bounds [7]. If there is no intersection between the line and $U_{\mathcal{S}_P}$, then all the circles of the pencil contain some points of $\mathcal{S}_P$. □

3 Voronoï diagram of n points in k planes

This section deals with the main problem of this paper : the construction of the Delaunay triangulation of a set $\mathcal{S}$ of n points belonging to a set $\mathcal{P}$ of k planes.

The algorithm starts from a first tetrahedron and enumerates all the Delaunay tetrahedra in a shelling order. Lemma 1 is used as described in Section 3.1 as a hint to determine the Delaunay neighbor of an already constructed tetrahedron through one of its facets.

3.1 Searching for a neighbor

Lemma 1 which solves a 2-dimensional problem, can be used to solve 3-dimensional queries in a special case. Let P be a plane in the euclidean 3-dimensional space, and Q be the radical plane of a pencil of spheres ; then the intersection of the spheres with P is a pencil of circles whose radical axis is $P \cap Q$. Let S_P be a set of sites in plane P. The extremal spheres (if there exist any) which do not contain any point of S_P are obviously determined by the extremal empty circles of the pencil of circles.

If all the planes P in $\mathcal{P}$ are considered now, then the extremal spheres which do not contain any point of $\mathcal{S} = \cup_{P \in \mathcal{P}} S_P$ can be determined in two steps. Firstly, for each plane P, the extremal spheres which do not contain any point of S_P are determined. Then, we select from the at most $2k$ candidate spheres the possible solutions.

In this way, it is possible to find the neighbors of a given Delaunay tetrahedron. If $abcd$ is a Delaunay tetrahedron, then there exist two extremal empty spheres in the pencil of spheres with base points abc. One is the sphere circumscribing $abcd$, and the other is determined by a point of $\mathcal{S}$, denoted p_{abc}. $abcp_{abc}$ is the tetrahedron adjacent to $abcd$ through triangle abc in the Delaunay triangulation.

Lemma 2 *A set $\mathcal{S}$ of n points lying in k planes can be preprocessed in $O(n \log n)$ time and $O(n)$ space, so that the neighbor of a Delaunay tetrahedron can be determined in $O(k \log n)$ time.*

3.2 The main algorithm

These remarks yield a simple algorithm to construct the Delaunay triangulation of a set of n points lying in k planes. The algorithm starting from one initial tetrahedron constructs the whole set of Delaunay tetrahedra incrementally. In order to achieve a linear space complexity, we process the tetrahedra in an order that ensures that the set of already constructed tetrahedra remains simply connected at each step of the incremental construction.

This can be ascertained by making use of the mechanism of shelling introduced in [4] for proving that the boundary complex of any convex polytope is shellable in any dimension [4, Proposition 2]. This idea has already been exploited in [8] and the reader can also refer to [8, Section 3].

The 3-dimensional Delaunay triangulation is interpreted as a 4-dimensional simplicial lower convex hull $\mathcal{L}$ of points on a paraboloid. Let D be a line parallel to the axis of the paraboloid, and imagine an observer moving down along D from the intersection point of D and $\mathcal{L}$. This observer reports first the facet F_1 of $\mathcal{L}$ hit by D and, as he moves down along D, he reports the facets $F_2, \ldots, F_t$ in the order they become visible (see Figure 3). This order called the shelling order is given by the attribution of a priority to each facet of $\mathcal{L}$. The priority of a facet is the altitude of the intersection of its supporting hyperplane with D and the shelling order of the facets corresponds to the decreasing order of these priorities. [4] ensures that this order is a good shelling of every convex polytope, i.e. that for all i, $\cup_{j=1}^{i} F_j$ is a simply connected 3-dimensional topological ball in 4-dimensional space.

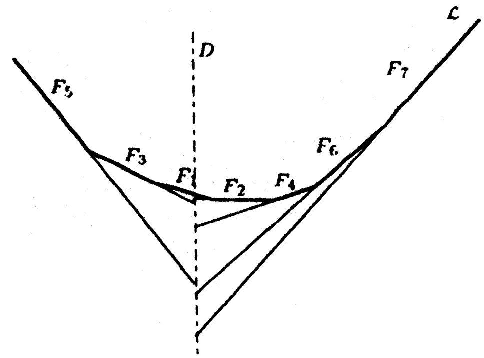

Figure 3: The shelling order

Going back to the original problem of constructing 3-dimensional Delaunay triangulation, we are able to compute the priority of each tetrahedron ; and, if the tetrahedra are constructed by decreasing priority, the set of constructed tetrahedra remains simply connected at each step. We assume that line D is in general position, and there are no two distinct tetrahedra with the same priority (degeneracies can be solved by perturbing D).

Let $\mathcal{T}$ be the set of already constructed tetrahedra. We maintain in a priority queue the set of tetrahedra not in $\mathcal{T}$, but sharing a face with the tetrahedra of $\mathcal{T}$. Notice that, by the result of [4] a tetrahedron of the queue shares one, two or three faces with $\mathcal{T}$; thus it appears one two or three times in the priority queue (with the same priority). A tetrahedron is present only once if and only if one of its vertices has never been considered so far.

As all the Delaunay tetrahedra adjacent to $\mathcal{T}$ are present in the priority queue, we can ensure that the next tetrahedron in the shelling is at the beginning of the queue.

Thus our algorithm will run as follows :

1. Execute the preprocessing step of Lemma 2.
2. Initialization :
   (a) Compute a triangle $a_1b_1c_1$ of the convex hull (for example the first face produced by the gift wrapping algorithm).
   (b) Compute $d_1 = p_{a_1b_1c_1}$ using Lemma 2.

(c) Compute the "vertical" line D in $\mathbb{R}^4$ hitting facet $a_1b_1c_1d_1$ lifted on the 4-dimensional paraboloid.

(d) Insert $a_1b_1c_1d_1$ in the priority queue.

3. Repeat

   (a) Find the maximal tetrahedron $abcd$ in the priority queue.

   (b) If $abcd$ appears only once, without loss of generality $d = p_{abc}$. Add tetrahedron $abcd$ in $\mathcal{T}$, compute p_{abd}, p_{acd} and p_{bcd}, add $abdp_{abd}$, $acdp_{acd}$ and $bcdp_{bcd}$ in the priority queue.

   (c) If $abcd$ appears twice, without loss of generality $d = p_{abc}$ and $c = p_{abd}$. Add tetrahedron $abcd$ in $\mathcal{T}$, compute p_{acd} and p_{bcd}, add $acdp_{acd}$ and $bcdp_{bcd}$ in the priority queue.

   (d) If $abcd$ appears three times, without loss of generality $d = p_{abc}$, $c = p_{abd}$ and $b = p_{acd}$. Add tetrahedron $abcd$ in $\mathcal{T}$, compute p_{bcd}, add $bcdp_{bcd}$ in the priority queue.

   Until the queue is empty.

3.3 Complexity

The complexity of this algorithm is the following.

Step 1 is completed in $O(n \log n)$ time and uses $O(n)$ space by Lemma 2.

Step 2a is done in $O(n)$ time.

Step 2b takes $O(k \log n)$ time by Lemma 2.

Step 2c and 2d are done in constant time.

At each iteration of Step 3 a new tetrahedron is added to the Delaunay triangulation thus this step is executed t times. Operations on the priority queue are done in $O(\log n)$ time, the determination of a point p_{xyz} is done in $O(k \log n)$ time by Lemma 2 and the computation of the priority of a tetrahedron takes constant time. Thus the overall cost of Step 3 is $O(tk \log n)$.

Each item in the priority queue corresponds to a triangle on the boundary of the already constructed tetrahedra. As the tetrahedra are reported in a shelling order, these triangles form a topological sphere, and by Euler's relation there is at most a linear number of such triangles. So the size of the priority queue is $O(n)$.

Theorem 3 *The three dimensional Delaunay triangulation of n points lying in k planes can be computed using $O(tk \log n)$ time and $O(n)$ extra space where t is the size of the output.*

In the special case of only two planes, it is possible to speed up the algorithm [2]. This is in fact a generalization of the algorithm for two parallel planes [1].

In this abstract, we just stated the following theorem :

Theorem 4 *The three dimensional Delaunay triangulation of n points lying in 2 planes can be computed in $O(t + n \log n)$ time and $O(n)$ space where t is the size of the output.*

4 Conclusion

We have presented output sensitive algorithms for constructing the Delaunay triangulation of special sets of points in 3-dimensional space. Specifically, when the input consists of n points scattered through k planes, we have shown that their Delaunay triangulation can be computed in $O(tk \log n)$ time and $O(n)$ space. This time bound can be reduced to $O(n \log n + t)$ (which is optimal with respect to the input and the output) when $k = 2$.

This result makes a further step (after [1]) towards the efficient construction of the Delaunay triangulation in an output sensitive fashion in 3-dimensional space. We believe that these results are of interest in several practical applications, especially computer vision and computerized tomography where data are naturally distributed in planes. We let as an open question whether our construction can be improved with respect to k ; and we recall the main (and probably difficult) unsolved question in that area : *is $O(n \log n + t)$ an upper bound for constructing the Delaunay triangulation of n points in 3-dimensional space if the triangulation consists of t tetrahedra ?*

References

[1] J. Boissonnat. Shape reconstruction from planar cross-sections. *Computer Vision, Graphics, and Image Processing*, 44:1–29, 1988.

[2] J. Boissonnat, A. Cérézo, O. Devillers, and M. Teillaud. *Output sensitive algorithm for constructing the 3D Delaunay triangulation of constrained sets of points.* Technical Report, INRIA, Apr. 1991.

[3] J. Boissonnat, O. Devillers, R. Schott, M. Teillaud, and M. Yvinec. On-line geometric algorithms with good expected behaviours. In *IMACS in Dublin*, Jul. 1991.

[4] H. Bruggesser and P. Mani. Shellable decompositions of cells and spheres. *Math. Scand.*, 29:197–205, 1971.

[5] K. Clarkson and P. Shor. Applications of random sampling in computational geometry, II. *Discr. and Comp. Geom.*, 4(5), 1989.

[6] H. Edelsbrunner. *Algorithms on Combinatorial Geometry.* Springer-Verlag, 1987.

[7] H. Edelsbrunner and H. Maurer. Finding extreme points in three dimensions and solving the post office problem in the plane. *Inf. Proc. Letters*, 21:39–47, Jul. 1985.

[8] R. Seidel. Constructing higher-dimensional convex hulls at logarithmic cost per face. In *ACM STOC*, pages 404–413, 1986.