Jamadan

On Good Triangulations in Three Dimensions*

Tamal K. Dey *Chanderjit L. Bajaj* *Kokichi Sugihara*

Department of Computer Science
Purdue University
West Lafayette, IN 47907

Abstract

In this paper, we give an algorithm that triangulates the convex hull of a three dimensional point set with guaranteed quality tetrahedra. Good triangulations of convex polyhedra are a special case of this problem. We also give a bound on the number of additional points used to achieve these guarantees and report on the techniques we use to produce a robust implementation of this algorithm under finite precision arithmetic.

1 Introduction

Triangulation of a point set or a polytope is an important problem with applications for finite element simulations in CAD/CAM. Though a number of algorithms exist for triangulating a point set or a polytope in two and three dimensions [1, 6, 11, 13], few of them address the problem of guaranteeing the shape of the triangular elements. To reduce ill-conditioning as well as discretization error, finite element methods require triangular meshes of bounded aspect ratio [2, 12]. By aspect ratio of triangles or tetrahedra, one may consider the ratio of the radii of the circumscribing circle to that of inscribing circle (spheres in case of tetrahedra).

In 2D, there are basically two approaches known so far to produce guaranteed quality triangulations. The first approach, based on *Constrained Delaunay Triangulations*, was first suggested by Chew [7]. He guarantees that all triangles produced in the final triangulation have angles between 30° and 120°. In [8] we improved this algorithm with minor modifications to guarantee the boundary triangles to have better angle bounds (between 38.9° and 97.2°). There is another approach based on *Grid Overlaying* which was first used by Baker, Grosse, and Raferty in [3] to produce a non-obtuse triangulation of a polygon. In [8], we proposed a simpler method based on this grid approach to triangulate a polygon with good angles. Recently, in [5], Bern, Eppstein, and Gilbert give algorithms for producing good triangulations which uses a special type of a grid that simulates the planar subdivision with the quadtree.

It is suggestive to see whether the two approaches in 2D generalize in 3D. Bern, Eppstein, and Gilebert [5] generalize their quadtree approach in 2D to produce a good triangulation of a 3D point set. Their method, however, introduces points outside the convex hull of the input point set. In a recent paper, Mitchell and Vavasis [14] use the same approach in attempt to produce guaranteed quality triangulations of polytopes. Their method, however, has not yet been proved to be correct. In this paper we build on the Constrained Delaunay approach of Chew. This approach has certain advantages over the grid overlaying approach. The algorithm with the Constrained Delaunay approach is simple and generalizes easily in 3D.

*Supported in part by ARO Contract DAAG29-85-C0018 under Cornell MSI, NSF grant DMS 88-16286 and ONR contract N00014-88-K-0402.

On the other hand grid overlaying approach seems to lead to many special cases when generalized to 3D. In this approach most of the new points are added along certain preferred directions since they lie on orthogonal grid planes. This may lead to undesirable artifacts in the numerical solutions obtained through these triangular meshes.

Although generalization of Chew's 2D algorithm is straightforward, the fact that it produces good tetrahedra is not obvious. We show that Chew's 2D algorithm as extended in this paper for triangulating a 3D point set indeed produces good tetrahedra. In particular we prove that four out of five possible types of bad tetrahedra are never produced if we assume lower and upper bounds on the dihedral angles of the adjacent facets on the convex hull boundary. Unfortunately the assumption of an upper bound on the dihedral angles leads to a constant number of vertices on the convex hull boundary. Thus, this assumption prohibits us to consider the triangulation of general convex polytopes as special case. The algorithm, however, guarantees that all tetrahedra having at least one vertex not on the boundary are good irrespective of the large dihedral angles. Thus this algorithm can still be applied to general convex polytopes where "most" of the tetrahedra are guaranteed to be good. The bad tetrahedra, in this case, are produced due to the badness in the input.

Our main results are as follows: (i) We show that the generalization of Chew's algorithm produces a good triangulation of a 3D point set. In particular, it never produces four out of five possible bad tetrahedra if we assume lower and upper bounds on the dihedral angles of the boundary of the convex hull. If we do not assume any upper bound on the dihedral angles, all tetrahedra except the ones with all vertices on the boundary satisfy the above property. (ii) We give a bound on the number of additional points used to achieve this guarantee. We also report on the techniques we use to produce a robust implementation of this 3D triangulation algorithm in the presence of numerical errors under finite precision arithmetic.

2 Preliminaries

2.1 Characterizing Bad Tetrahedra

In three dimensions, a tetrahedron that is not of bounded aspect ratio can be degenerate or bad in three possible ways as described in [4]. The following two parameters ω, κ characterize bad tetrahedra as follows. Let $\omega = \frac{R}{L}$ and $\kappa = \frac{L}{l}$, where R is the radius of the circumscribing sphere of a tetrahedron, L and l are the lengths of its longest and shortest edges respectively.

Category(i): $\omega = O(1), \kappa \gg 1$.
Category(ii): $\omega \gg 1$.
Category(iii): $\omega = O(1)$, $\kappa = O(1)$.
Definition: A sliver is a tetrahedron that is formed by four almost coplanar points and all of whose solid angles are very close to zero.

Category(i) corresponds to tetrahedra that have a very short edge relative to other edges and have circumscribing spheres that do not have an arbitrarily large radius compared to the length of the longest edge. Specifically, category(i) consists of type(i) and type(ii) tetrahedra. Type(i) tetrahedra are needle-like tetrahedra in which one of the solid angles is highly acute and the face opposite to it has a negligible area (Figure 1(a)). Type(ii) tetrahedra are slivers with a very short edge (Figure 1(b)).

Category(ii) corresponds to tetrahedra that have a circumscribing sphere with arbitrarily large radius compared to the longest edge. Specifically, category(ii) consists of type(iii) and type(iv) tetrahedra. Type(iii) tetrahedra are flat tetrahedra which have one of the solid angles highly obtuse (Figure 2(a)). Type(iv) tetrahedra are slivers which lie very close to the surface of their large circumscribing spheres (Figure 2(b)). Category(iii) consists of type(v) tetrahedra. Type(v) tetrahedra are slivers whose edges have lengths within a constant factor of each other and which do not have a close incidence with the

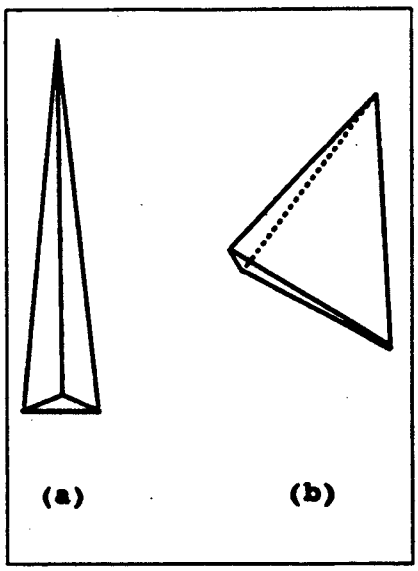

Figure 1: Category(i) tetrahedra

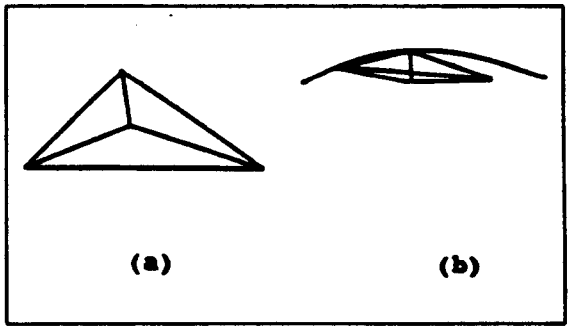

Figure 2: Category(ii) tetrahedra

surface of the circumscribing sphere (Figure 3). We present an algorithm that triangulates the convex

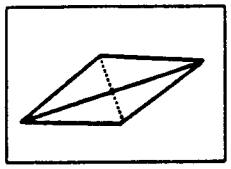

Figure 3: Category(iii) tetrahedra

hull of a three dimensional point set with the guarantee that type(i) through type(iv) tetrahedra are not generated.

2.2 2D Algorithm

The core of the algorithm presented in this paper consists of the Delaunay triangulation which is the straight line dual of the Voronoi diagram. In two dimensions, the circumscribing circle of a triangle in the Delaunay triangulation of a point set does not contain any other point inside it. Similarly, in three dimensions, the circumscribing sphere of a tetrahedron in the Delaunay triangulation does not contain any other points inside it. This property of the Delaunay triangulation is utilized by Chew in two dimensions to produce good triangulations. He adds the centers of the circumscribing circles of large radius. Of course, the edges of the boundary (boundary of the convex hull in case of point set; given explicit polygonal boundary otherwise) have to satisfy certain length criteria. In his algorithm, Chew used edge lengths in between d and $\sqrt{3}d$ where any pair of input points is at least d units away from each other. In the modified algorithm of [8], we require edge lengths in between d and $1.5d$. This gives two distinct advantages.

1. It is easier to divide edges between d and $1.5d$ in practice.
2. The triangles that have circumcenters outside the boundary have better bounds on their angles.

We present below this modified algorithm for good triangulation of a point set in two dimensions.

Algorithm 2D-TRI:
Input: Finite number of points in plane.

Input Conditions: There exists a quantity d, such that all boundary edges (convex hull edges) are breakable into segments of lengths in between d and $1.5d$ and no two points are closer than d.

Step 1. Break the edges of the convex hull into segments of lengths in between d and $1.5d$.

Step 2. Construct the Delaunay triangulation of the resulting point set.

Step 3. Find a triangle g satisfying the following properties: (i) the radius of the circumscribing circle c of g is greater than d, (ii) the center of c is inside the boundary.

Step 4. If no triangle g is found in Step 3, return the current triangulation. Otherwise, add the circumcenter of g to the current point set and go to Step 2.

Of course, to satisfy the input conditions of the algorithm 2D-TRI, we have to show that such a d exists. Let d_1 be the minimum distance between any two input points. Let d_2 be the minimum distance between a point and a boundary edge and d_3 be the minimum length of any boundary edge. A simple

minded choice of d would be $d = min(d_1, d_2, \frac{d_3}{3})$. With this choice of d, all boundary edges have lengths greater than or equal to $3d$. Such edges can be easily divided into segments which have lengths in between d and $1.5d$. However, if the boundary has a very sharp internal angle θ at a vertex v, the new points introduced on the edges e_1, e_2 incident on v may be closer than d. We divide the edges e_1, e_2 in such a way that the segments incident on v have lengths $1.5d$. The other two endpoints of these segments are at a distance of $3d \sin\frac{\theta}{2}$. For this distance to be greater than d, we need $\theta > 2\sin^{-1}\frac{1}{3}$. With this scheme, each edge is divided in such a way that two segments incident on the two endpoints have lengths of $1.5d$. Thus each edge must have a length of at least $6d$ ($3d$ units for two extreme segments and another $3d$ units for middle segments). We define d as $d = min(d_1, d_2, \frac{d_3}{6})$. With the assumption that $\theta > 2\sin^{-1}\frac{1}{3}$, this choice of d satisfies the input conditions of the algorithm 2D-TRI.

Note that in the algorithm 2D-TRI, we check whether the new points to be added are within the boundary or not. Any triangle with the circumcenter lying outside the boundary must be obtuse and has edges of lengths in between d and $1.5d$. The radius of circumscribing circle of such triangles can not be greater than d. Thus the check whether the circumcenter lies inside the boundary or not is redundant.

Algorithm *2D-TRI* produces a planar triangulation T that has the following properties.

Property 1: All edges in T have lengths in between d and $2d$ and in particular all boundary edges have lengths in between d and $1.5d$.

Property 2: The circumscribing circle of all triangles in T has radius less than or equal to d.

2.3 Geometric Lemmas

We use the following geometric lemmas in the next section.

Lemma 2.1: Let T be a Delaunay triangulation of a point set in two dimensions. Let R be the maximum radius of all circumscribing circles of Delaunay triangles in T. The radius of any empty circle whose center lies inside T is less than or equal to R.

Proof: See Theorem 6.15[15]. ♣

In the rest of the paper, we use $cl(c)$ to denote the "disk" enclosed by a circle c and $cl(s)$ to denote the "ball" enclosed by a sphere s.

Definition: Let c be a circle drawn on a sphere s. Let p_1p_2 be the axis which is perpendicular to the supporting plane of c and which passes through the center of c. This axis intersects s at p_1 and p_2. The points p_1, p_2 are called the *poles* corresponding to the circle c.

Lemma 2.2: Let c be a circle with radius less than or equal to r drawn on a sphere s. Let the distance between $cl(c)$ and its nearest pole be greater than or equal to d. The radius R of s must satisfy the condition $R \leq \frac{r^2+d^2}{2d}$.

Proof: Consider the circle c as shown in Figure 4 with the nearest pole p_1. Let a, b be the centers of s and c respectively. Obviously, $|ab| \leq (R - d)$. Consider the right angled triangle $\triangle abt$ where t is a point on the circle c. Since the radius of c is less than or equal to r, we have $|bt| \leq r$. Hence, $|at|^2 = R^2 = |ab|^2 + |bt|^2 \leq (R-d)^2 + r^2$ giving $R \leq \frac{r^2+d^2}{2d}$. ♣

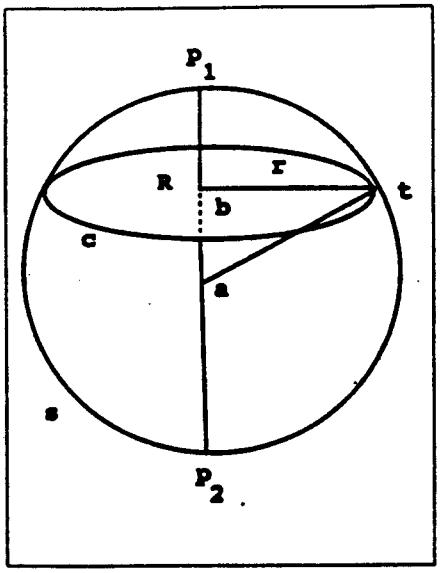

Figure 4: Lemma 2.2

3 3D Algorithm

We assume that a finite number of points is given in three dimensional space. We call the boundary of the convex hull of these points the *boundary*. In what follows, by the convex hull of a point set, we mean its interior along with its boundary. A point is called an *internal point* if it is not on the boundary and is called a *boundary point* otherwise. The facets of the boundary are referred to as *boundary facets* and the edges on the boundary facets are called *boundary edges*. The dihedral angles between adjacent boundary facets are referred to as *input dihedral angles*.

Algorithm 3D-TRI:
Input: Finite number of points in three dimensional space.

Step 1. Let d_1 be the minimum distance between any two points; d_2 be the minimum distance from an internal point to a boundary facet; d_3 be the minimum distance between a vertex and an edge on a facet; d_4 be the minimum distance between two nonadjacent boundary facets. Choose $r = \frac{1}{6}\{d_1, d_2, d_3, d_4\}$.

Step 2. Break all edges so that each segment has length in between r and $1.5r$.

Step 3. Triangulate each facet of the boundary using the algorithm 2D-TRI with $d = r$. The Step 1 of the algorithm 2D-TRI is not executed since the edges are already broken accordingly.

Step 4. Let P be the current point set. Construct a 3D Delaunay triangulation $T(P)$ of the point set P.

Step 5. Search for a tetrahedron t in $T(P)$ satisfying the following properties: (i) the circumscribing sphere s of t has a radius greater than $2r$, (ii) the center v of s lies inside the boundary.

Step 6. If no tetrahedron is found in Step 5, return the current triangulation. Otherwise, add v to current point set and go back to Step 4.

With the above choice of r and with the assumption that all the face-angles of the facets on the

boundary satisfy the minimum angle criterion, it is possible to triangulate them by *2D-TRI* maintaining the edge lengths as stated. In the following Lemma, we prove that the above procedure terminates.

Lemma 3.1: Algorithm *3D-TRI* terminates.

Proof: Algorithm *2D-TRI* terminates since the points added by it are always at a certain distance from all other points. There can be only finitely many such points inside the given polygonal boundary. Extending this argument to Algorithm 3D-TRI, we can observe that all the circumcenters of tetrahedra that are added as new points are at a distance of at least $2r$ from all other points. There can be only finitely many such points inside the convex hull of the input points, which assures the termination of the Algorithm 3D-TRI. ♣

Lemma 3.2: Any point on a boundary facet that does not lie on a boundary edge must be at a distance of at least $\frac{\sqrt{7}}{4}r$ from all edges of that facet.

Proof: Consider a point p on a facet f. Let e be any edge of f. Note that the edge e is divided into smaller edges $e_1, e_2, ..., e_n$ through the triangulation of the boundary facets adjacent to e. Drop a perpendicular from p on the line supporting e. If the perpendicular intersects the edge e, let e_l be the edge of the triangulation on e which is intersected by it. According to property 1, all boundary edges of the triangulation of f must have lengths in between r and $1.5r$. Further, the point p is at least r units away from the end points of e_l. Thus, the minimum distance between p and e_l is at least $\frac{\sqrt{7}}{4}r$. In case the perpendicular dropped from p does not intersect e, it must intersect some other edge e' of f. In that case, the distance between p and e must be greater than the distance between p and e'. We can estimate the minimum distance between p and e by estimating the same between p and e'. While estimating the distance between p and e', if it occurs that the perpendicular dropped from p does not intersect e', we will have another edge to estimate the minimum distance between p and e'. Since there are finite number of edges and since each time we go to a next edge, its distance from p gets smaller than the previous one, there must be an edge of f which is intersected by the perpendicular dropped from p. Let e'' be the first such edge encountered in the above process. As argued above, the distance between p and e'' is at least $\frac{\sqrt{7}}{4}r$. Hence, the distance between p and e is at least $\frac{\sqrt{7}}{4}r$. Thus, any point on a boundary facet that does not lie on a boundary edge must be at a distance of at least $\frac{\sqrt{7}}{4}r$ from all edges of that facet.♣

Lemma 3.3: All edges in the triangulation produced by the algorithm *3D-TRI* have lengths greater than l_{min} where $l_{min} = \min(r, \frac{\sqrt{7}}{2}r \sin\frac{\theta_m}{2})$. Here θ_m is the minimum input dihedral angle.

Proof: Initially, all internal points are at a distance of at least $6r$ units from every other point. Two boundary points, lying on non adjacent facets, are at least $6r$ units away from each other. These conditions are ensured by the particular choice of r. A boundary point is at a distance of at least r from every other point on the same facet which is ensured by the algorithm *2D-TRI*. The points added by the algorithm *3D-TRI* are always at a distance of at least $2r$ from every other point. Thus, all points except the points on the adjacent facets are at a distance of at least r from each other. To estimate the minimum distance between any two points on the adjacent boundary facets, consider two points p_1, p_2 lying on the adjacent facets f_1, f_2 respectively. Let e be the edge shared by f_1 and f_2. Let T be the plane that passes through p_1 and is perpendicular to e. Let T intersect e at p_3. The normal dropped from p_2 on T lies on the plane supporting f_2 and is parallel to e. Let it meet T at p_2'. In the right-angled triangle $p_1p_2p_2'$, $|p_1p_2| > |p_1p_2'|$ since p_1p_2 is the hypotenuse. Consider the triangle $p_1p_2'p_3$. Let the minimum dihedral angle between any two adjacent facets be θ_m. From the above discussion, it follows that $|p_1p_3| > \frac{\sqrt{7}}{4}r$ and $|p_2'p_3| > \frac{\sqrt{7}}{4}r$. Thus,

the distance between p_1, p_2' and hence between p_1, p_2 is at least $\frac{\sqrt{7}}{2} r \sin\frac{\theta_m}{2}$. Hence, all edges in the final triangulation produced by the algorithm *3D-TRI* have lengths greater than $l_{\min} = \min(r, \frac{\sqrt{7}}{2} r \sin\frac{\theta_m}{2})$. ♣

Lemma 3.4: Let d be the distance of a point p present as a vertex in the triangulation produced by the algorithm *3D-TRI* from any boundary facet on which p does not lie. $d \geq r$ if p is an internal point and $d \geq \frac{\sqrt{7}}{4} r \sin\theta_m$ if p is a boundary point. Here θ_m is the angle such that all input dihedral angles are within θ_m and $180° - \theta_m$.

Proof: If p is an internal input point, we already know p is at least r units away from every boundary facet. We can show that if p is an added internal point, it is also r units away from every boundary facets. p is at least $2r$ units away from every other point. Let f be the closest boundary facet to p. The foot of the perpendicular dropped from p on the supporting plane of f lies inside a triangle, say Δqst on f. Consider the tetrahedron formed by p, q, s, t. Since all edges of the triangle Δqst have lengths of at most $2r$ and the point p is at least $2r$ away from q, s, t, the minimum height of p from Δqst is achieved when $pqst$ is a regular tetrahedron with all edge lengths equal to $2r$. This height is greater than r.

Consider the case when p is a boundary point. By the choice of r, any point on a boundary facet is at least r units away from any other nonadjacent facet. We prove that if p lies on a boundary facet but not on a boundary edge, it is at a distance of at least $\frac{\sqrt{7}}{4} r \sin\theta_m$ from all adjacent facets. Let p lie on f_1 and let f_2 be any facet adjacent to f_2. In Lemma 3.2, we proved that the distance of p from any line supporting an edge of the facet f_1 is at least $\frac{\sqrt{7}}{4} r$. Let l be the distance of p from the line where f_1 and f_2 meet. The distance d of p from f_2 is given by $d = l \sin\theta$ where θ is the dihedral angle between f_1 and f_2. Putting the minimum value of l and θ gives the lower bound on d. Thus, the distance of a point from any facet that does not contain it is at least $d_{\min} = \min(r, \frac{\sqrt{7}}{4} r \sin\theta_m) = \frac{\sqrt{7}}{4} r \sin\theta_m$. ♣

4 Qualities of Tetrahedra

Definition: A tetrahedron in the final triangulation is said to have a *good circumcenter* if the center of its circumscribing sphere lies inside or on the boundary (convex hull boundary). Conversely, a tetrahedron is said to have a *bad circumcenter* if the center of its circumscribing sphere lies outside the boundary.
We classify the tetrahedra with bad circumcenters into two classes, namely class A and class B.

Definition: A tetrahedron t with a bad circumcenter is called a class A tetrahedron if it satisfies the following property. There exists a facet f intersected by the circumscribing sphere s of t in such a way that the foot of the perpendicular dropped from the center of s on the supporting plane of f lies inside f. Any other tetrahedron with a bad circumcenter is called a class B tetrahedron. See figure 5 and figure 6.

We further divide the class A tetrahedra into class A1 and class A2 tetrahedra and the class B tetrahedra into class B1 and class B2 tetrahedra.

Definition: A class A tetrahedron is called a class A1 tetrahedron if it has an internal point as its vertex. Any other class A tetrahedron is called class A2 tetrahedron. Similarly, we define class B1 and class B2 tetrahedra.

Assuming a lower bound on the input dihedral angles, we can prove that "most" of the tetrahedra produced by *3D-TRI* cannot be in category(i) or category(ii). Assuming both lower and upper bounds on the input dihedral angles, we can prove that all tetrahedra produced by *3D-TRI* cannot be in category(i) or

category(ii). We show that all edges in the final triangulation has a lower bound on their lengths if we assume a lower bound on the input dihedral angles. Intuitively, this is true because all points added by 2D-TRI and 3D-TRI cannot be too close to any other existing point. Next we show that all tetrahedra have a circumscribing sphere that is not too large. Intuitively, a tetrahedron with good circumcenter cannot have a large circumscribing sphere because our algorithm could add its circumcenter. A class A tetrahedron cannot have a large circumscribing sphere because our algorithm would add a point to a boundary facet between the tetrahedron and its circumcenter. Finally, the circumscribing sphere of a class B tetrahedron is broken up by a point added to a boundary edge. Although we cannot avoid category(iii) tetrahedra, occurrences of them in practice are rare, as stated in [4]. Finally, in most of the cases these category(iii) tetrahedra can often be avoided by introducing a suitable point inside the circumscribing sphere. See [4].

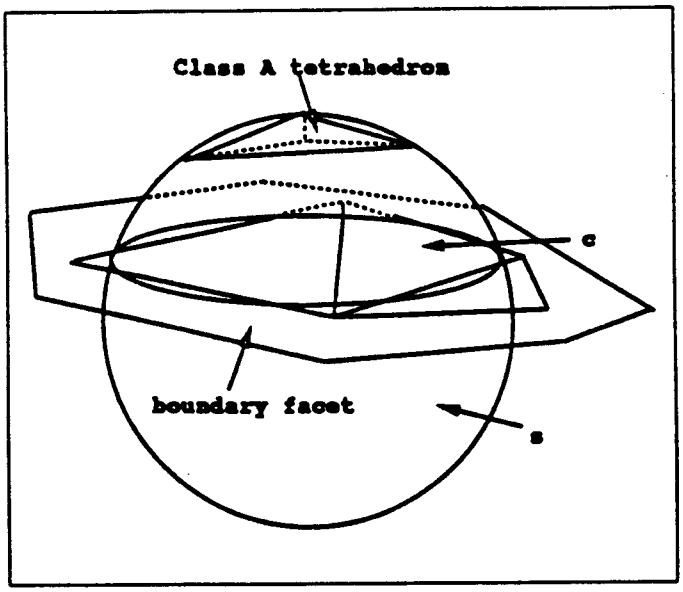

Figure 5: class A tetrahedron

Lemma 4.1: Assuming a lower bound on the input dihedral angles, no tetrahedron with good circumcenter can be in category(i) or category(ii).

Proof: All tetrahedra in the final triangulation having good circumcenters must have circumscribing spheres with radii less than or equal to $2r$, because otherwise these circumcenters would have been introduced as new points. Hence, all these tetrahedra have edges of length less than or equal to $4r$. By Lemma 3.3, all edges have lengths greater than $\min(r, \frac{\sqrt{7}}{2} r \sin\frac{\theta_m}{2})$ where θ_m is the minimum input dihedral angle. Thus, κ for these tetrahedra can be at most $\max(4, \frac{8}{\sqrt{7}\sin\frac{\theta_m}{2}})$. Assuming a lower bound on the input dihedral angles, we get κ for these tetrahedra to be of $O(1)$ which violates the condition for category(i) tetrahedra. Further, ω for these tetrahedra can be at most $\max(2, \frac{4}{\sqrt{7}\sin\frac{\theta_m}{2}}) = O(1)$ which prohibits them to be in category(ii). ♣

Lemma 4.2: Assuming a lower bound on the input dihedral angles, no class A1 tetrahedron can be in category(i) or category(ii); assuming both lower and upper bounds on the input dihedral angles, no class A2 tetrahedron can be in category(i) or category(ii).

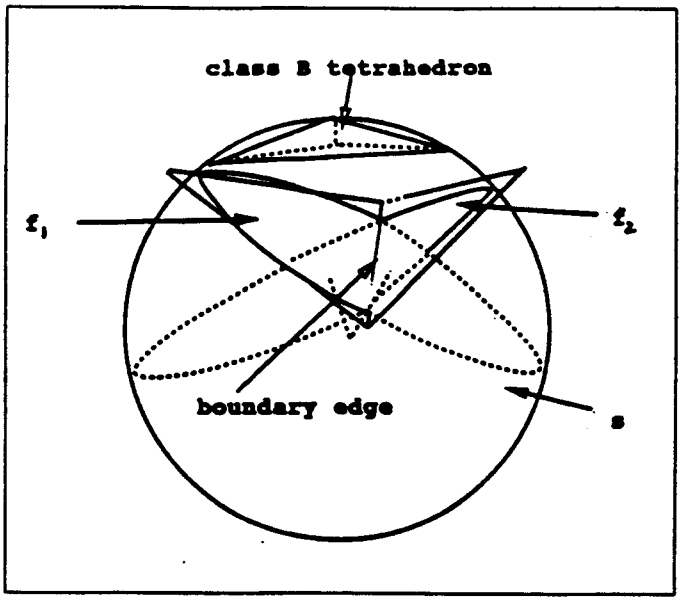

Figure 6: class B tetrahedron

Proof: Let t be a class A tetrahedron with the circumscribing sphere s. By the definition of class A tetrahedron, there exists a boundary facet f such that the foot of the perpendicular dropped from the center of s on the supporting plane of f lies inside f. Let c be the circle of intersection of s with the supporting plane of f. Let p be the vertex of t that is farthest from f and has a distance of d from it. If t is a class A1 tetrahedron, $d \geq r$. If t is a class A2 tetrahedron, $d \geq \frac{\sqrt{7}}{4} r \sin\theta_m$ where all input dihedral angles are in between θ_m and $180° - \theta_m$. The center of the circle c lies inside f. Thus, the center must lie inside the triangulation T of f produced by the algorithm *2D-TRI*. Further, c must be an empty circle since s does not include any point of f inside it. See figure 5. By property 2, all triangles of T have circumscribing circles of radii less than or equal to r. Hence, according to Lemma 2.1, c must have a radius less than or equal to r. The vertex p lying on s must be at a distance of at least d from $cl(c)$. Further, the vertex p and the center of s lie on the opposite sides of $cl(c)$. This implies $cl(c)$ is at a distance of at least d from its nearest pole. Thus, according to Lemma 2.2, s must have a radius less than or equal to $k_1 r$ where $k_1 = 1$ if t is a class A1 tetrahedron and $k_1 = (\frac{\sqrt{7}\sin\theta_m}{8} + \frac{2}{\sqrt{7}\sin\theta_m})$ if t is a class A2 tetrahedron. This puts an upper bound of $2k_1 r$ on the lengths of the edges of t. By Lemma 3.3, all edges of t are greater than $k_2 r$ where $k_2 = O(1)$ if we assume a lower bound on the input dihedral angles. Hence, if t is a class A1 tetrahedron, ω, κ for t are $O(1)$ assuming a lower bound on the input dihedral angles. If t is a class A2 tetrahedron, ω, κ for t are $O(1)$ assuming both lower and upper bounds on the input dihedral angles (A lower bound on θ_m puts lower and upper bounds on the dihedral angles between adjacent boundary facets). This prohibits it to be in category(i) or category(ii). ♣

Lemma 4.3: Let t be a class B tetrahedron with the circumscribing sphere s. There must exist two boundary facets f_1, f_2 intersected by s with the following criterion:
Let c be any circle drawn on s where $cl(c)$ is normal to the line where f_1, f_2 meet. The feet of the perpendiculars dropped from the center of c on the supporting planes P_1 and P_2 of f_1 and f_2 lie outside the line segments $cl(c) \cap f_1$, $cl(c) \cap f_2$.

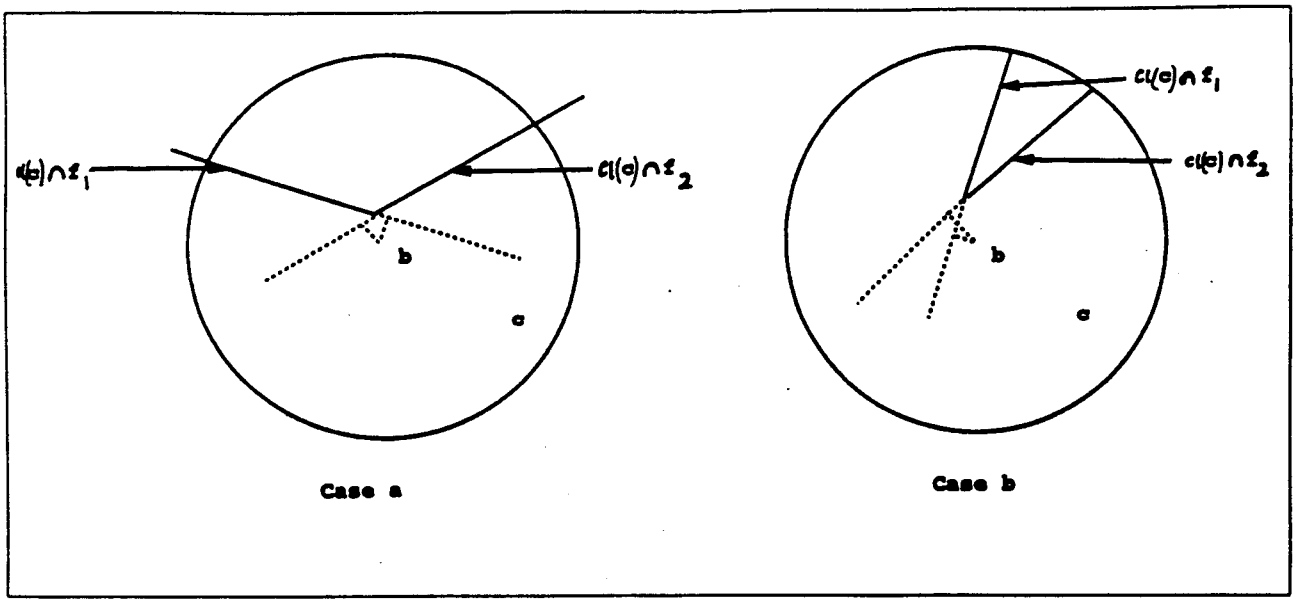

Figure 7: Lemma 4.3.

Proof: Consider a boundary facet f_1 that has the convex hull and the center of s on opposite sides. Since t has a bad circumcenter, such a facet always exists. Consider any other facet f_2 sharing an edge with f_1 that has been intersected by s. Drop perpendiculars from the center of s on the supporting planes of f_1 and f_2. The feet of these perpendiculars lie outside f_1, f_2 since t is a class B tetrahedron. Consider the great circle c' of s whose supporting plane is normal to the edge shared by f_1 and f_2. The feet of the perpendiculars dropped from the center of s on the supporting planes P_1 and P_2 of f_1 and f_2 cannot lie on the line segments $cl(c') \cap f_1$ and $cl(c') \cap f_2$. Two different cases are shown in figure 7. This immediately implies that the condition stated in Lemma 4.3 is true for any circle c on s that has a supporting plane parallel to that of c'.♣

Lemma 4.4: Assuming a lower bound on the input dihedral angles, no class B1 tetrahedron can be in category(i) or category(ii); assuming both lower and upper bounds on the input dihedral angles, no class B2 tetrahedron can be in category(i) or category(ii).

Proof: Let t be a class B tetrahedron. Let the circumscribing sphere s of t intersect the boundary edge e shared by the facets f_1 and f_2 which satisfy the criterion as stated in Lemma 4.3. The endpoints of the edge segment e_n on e which is intersected by s cannot be inside s. Let w, y be the points where s intersects e_n. Further, let a and R denote the center and radius of s respectively.

Case(i): The tetrahedron t has a vertex p that lies neither on the facet f_1 nor on the facet f_2. Note that a class B1 tetrahedron always satisfies this condition. Consider the circle c on s with $cl(c)$ perpendicular to e_n and passing through p. Let R' be the radius of c. Join the center b of c with the point u where $cl(c)$ meets e_n. Extend the line bu beyond u until it intersects c at v as shown in figure 8. Let $|bu| = x$. Certainly, $|uv| = R' - x$. Let d denote the minimum distance of p from the two facets f_1 and f_2. There are two subcases as shown in figure 8. In subcase $i(a)$, the center of c lies in the sides of the planes containing f_1, f_2 which are opposite to those containing the convex hull. It is not difficult to see that in this subcase $d \leq |uv| = R' - x$. Since, $R \geq R'$, we have $d \leq R - x$. To estimate a lower bound on x, drop a

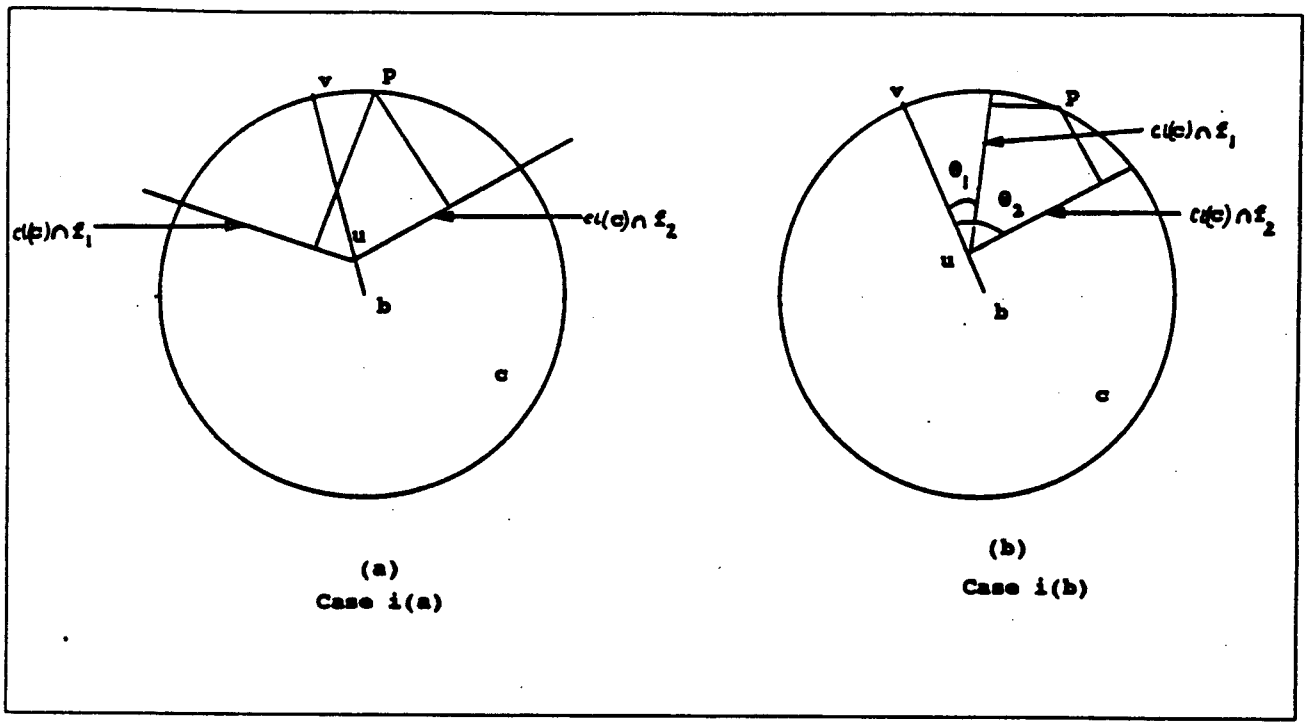

Figure 8: Lemma 4.4, case (i).

perpendicular az from the center a of s on e_n. This perpendicular has the same length as bu. Consider the triangle $\triangle awy$. We observe that $|az| = \sqrt{R^2 - \frac{|wy|^2}{4}}$. Since e_n can have a length of at most $1.5r$, we have $x = |az| \geq \sqrt{R^2 - \frac{9r^2}{16}}$. Thus, $d \leq R - \sqrt{R^2 - \frac{9r^2}{16}}$. We already know $d \geq r$ if t is a class B1 tetrahedron and $d \geq \frac{\sqrt{7}}{4} r \sin\theta_m$ if t is a class B2 tetrahedron (follows from Lemma 3.4). Hence for class B1 tetrahedra,

$$\begin{aligned} r &\leq R - \sqrt{R^2 - \frac{9r^2}{16}}, \\ R &\leq \frac{25r}{32}, \end{aligned}$$

and for class B2 tetrahedra,

$$\begin{aligned} \frac{\sqrt{7}r}{4} \sin\theta_m &\leq R - \sqrt{R^2 - \frac{9r^2}{16}}, \\ R &\leq \frac{7\sin^2\theta_m + 9}{8\sqrt{7}\sin\theta_m} r. \end{aligned}$$

Now, consider the subcase $i(b)$. In this subcase, one of the supporting planes of f_1 and f_2 has the center of c and the convex hull on its opposite sides and the other one has them on same side. Without loss of generality, assume that the supporting plane of f_1 has them on same side as shown in figure 8(b). The line segments $cl(c) \cap f_1$ and $cl(c) \cap f_2$ make angles less than equal to $90°$ with uv. Otherwise, f_1, f_2 do not satisfy the criterion as stated in Lemma 4.3. In this subcase, we have $d \leq R - x$ since the distance of

v from the supporting plane of f_2 is greater than that of p from the same plane. Thus, in both subcases $i(a)$ and $i(b)$, we have $R \leq \frac{25r}{32}$ if t is a class B1 tetrahedron, and

$$R \leq \frac{7\sin^2\theta_m + 9}{8\sqrt{7}\sin\theta_m} r$$

if t is a class B2 tetrahedron.

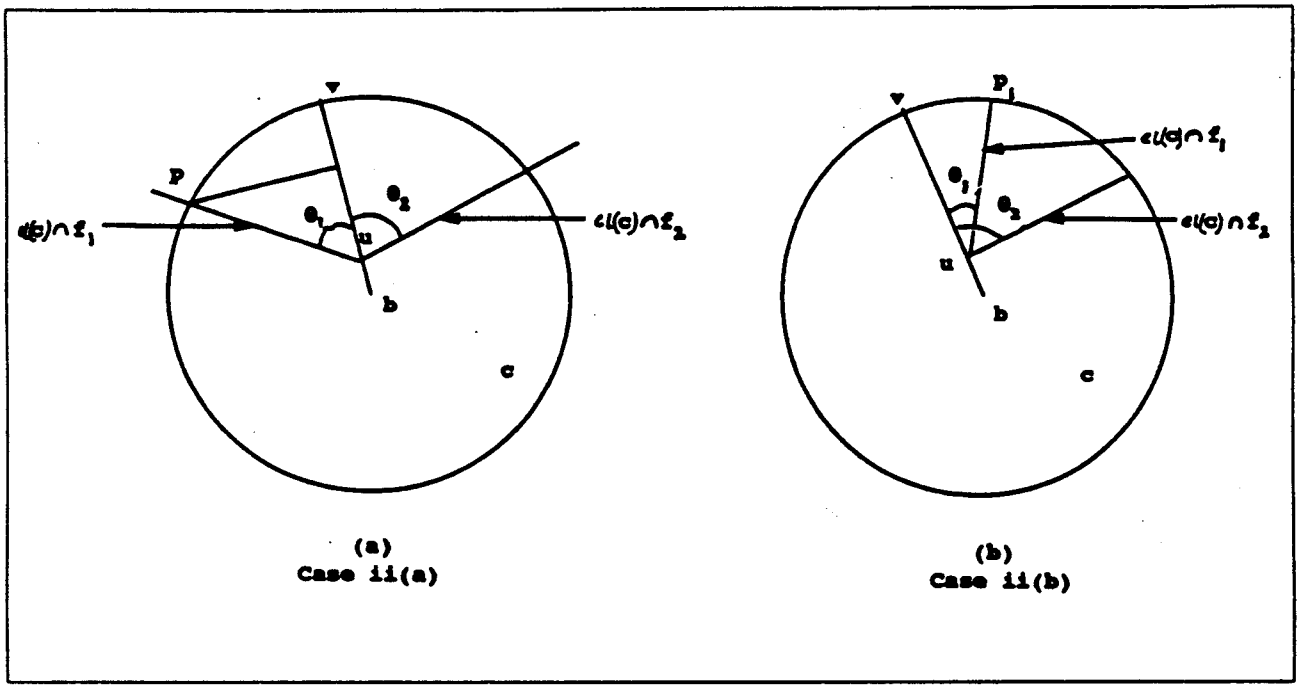

Figure 9: Lemma 4.4, case (ii).

Case(ii): All vertices of the tetrahedron t lie either on f_1 or on f_2. This immediately implies that one of the vertices of t lies on f_1 but not on f_2 and another on f_2 but not on f_1. Note that t cannot be a class B1 tetrahedron in this case. Consider the vertex p_1 lying on f_1 but not on f_2. Let c be the circle passing through p_1 with $cl(c)$ being perpendicular to e_n. As in the previous case, let b be the center of c, u be the foot of the perpendicular dropped from b to e_n, and v be the point of intersection of the line bu and the circle c such that u is in between b and v. Again, we have two subcases as shown in Figure 9. Consider the subcase $ii(a)$. We have $|p_1u| \leq \frac{|uv|}{\cos\theta_1}$, where θ_1 is the angle between p_1u and uv. We proved in lemma 3.2 that the distance of any point on a boundary facet that does not lie on any of its edges is at least $\frac{\sqrt{7}}{4}r$ away from any of its edges. Thus, $|p_1u| \geq \frac{\sqrt{7}}{4}r$. Hence, $\frac{\sqrt{7}}{4}r \leq \frac{R'-x}{\cos\theta_1} \leq \frac{R-x}{\cos\theta_1}$, where $x = |bu|$. Similarly, considering the vertex p_2 of t lying on f_2 but not on f_1, we can prove that $\frac{\sqrt{7}}{4}r \leq \frac{R-x}{\cos\theta_2}$, where θ_2 is the angle between $cl(c) \cap f_2$ and uv. The angle $\theta = \theta_1 + \theta_2$ is the dihedral angle between f_1 and f_2. Since one of θ_1, θ_2 is less than or equal to 90° and the cos function decreases monotonically from 0° to 90°, we have $\frac{\sqrt{7}}{4}r \leq \frac{R-x}{\cos\frac{\theta}{2}}$. By the same argument as in case(i), we get $x \geq \sqrt{R^2 - \frac{9}{16}r^2}$. Hence,

$$\frac{\sqrt{7}}{4}r \leq \frac{R - \sqrt{R^2 - \frac{9}{16}r^2}}{\cos\frac{\theta}{2}}$$

$$R \leq \frac{\frac{9}{16}r^2 + \frac{7}{16}r^2\cos^2\frac{\theta}{2}}{\frac{\sqrt{7}}{2}r\cos\frac{\theta}{2}}.$$

Assuming an upper bound on $\theta \leq (180° - \theta_m)$ we have,

$$R \leq \frac{7\sin^2\frac{\theta_m}{2} + 9}{8\sqrt{7}\sin\frac{\theta_m}{2}}r.$$

Now, consider the subcase $ii(b)$. The angles between uv and the line segments $cl(c) \cap f_1$ and $cl(c) \cap f_2$ are less than 90° since otherwise f_1, f_2 violate the condition of Lemma 4.3. Without loss of generality assume that $\theta_1 < \theta_2$. The distance between v and $cl(c) \cap f_2$ is greater than that between p_1 and $cl(c) \cap f_2$. This implies $d \leq R - x$ giving the same upper bound on R as we derived in case(i).

Thus, all class B1 tetrahedra have a circumscribing sphere of radius $k_1 r$ where $k_1 = O(1)$. Assuming a lower bound on the input dihedral angles, all edges have lengths $k_2 r$ where $k_2 = O(1)$ (recall Lemma 3.3). This makes ω and κ of these tetrahedra to be $O(1)$ and thus prohibits them to be in category(i) or category(ii). Class B2 tetrahedra have a circumscribing sphere of radius $k_1 r$ where $k_1 = O(1)$ if we assume both lower and upper bounds on the input dihedral angles. The fact that they cannot be in category(i) or category(ii) with this assumption is immediate.♣

The following Theorem is immediated from Lemmas 4.1, 4.2, and 4.4.

Theorem 4.1: Algorithm *3D-TRI* triangulates the convex hull of a three dimensional point set with the following properties: (i) Tetrahedra with at least one internal vertex can not be of type(i) through type(iv) if we assume a lower bound on the input dihedral angles, (ii) rest of the tetrahedra cannot be of type(i) through type(iv) if we assume both lower and upper bounds on the input dihedral angles.

5 Complexity

Algorithm *3D-TRI* produces tetrahedra whose edges are greater than l_{min} as defined in Lemma 3.3. The circumscribing sphere of each such tetrahedron must have a volume of $\Omega(l_{min}^3)$. Let V be the volume of the convex hull of the given point set. Let n and n_o be the number of points present in the input and output respectively. Certainly, $n_o = O(\frac{V}{l_{min}^3})$. Consider a triangulation T of the input point set where $|T| = O(n)$. Such a triangulation is always possible as follows. Compute the convex hull of the input point set. A linear triangulation of the convex hull is always possible. Insert the rest of the point one at a time by joining it to the four corners of the tetrahedron containing it. This guarantees a linear triangulation of the input point set. See [11] for details. Let L be the largest edge length in T. All tetrahedra in T have a volume less than L^3. Thus, $V = O(nL^3)$. This gives an upper bound of $O(n\frac{L^3}{l_{min}^3})$ on n_o. Putting $A = \frac{L}{l_{min}}$, we have $n_o = O(nA^3)$. The quantity A captures the notion of how badly distributed the input point set is.

The basis of *3D-TRI* is the incremental Delaunay triangulation algorithm. We use Watson's algorithm [18] for this purpose. In this algorithm, all tetrahedra whose circumscribing spheres contain the inserted point inside are removed. The new point is connected to the triangles present in the boundary of the union of all removed tetrahedra to produce new triangulation. In *3D-TRI* we introduce the circumcenters of tetrahedra that satisfy specific properties as new points. We maintain a linked list of all such tetrahedra throughout the algorithm. In the internal structure of each tetrahedron, we maintain a field to indicate whether it is still present in the current triangulation or not. We can pick a tetrahedron t whose circumcenter is to be added by scanning the linked list of tetrahedra. During this scanning, the tetrahedra that are not present in the current triangulation are also encountered. They are, however, scanned only once

and thus contribute $O(n_0)$ to the overall complexity. We can determine all tetrahedra to be removed and to be added in $O(n_o)$ time once we have chosen t. This is because there are at most $O(n_o)$ tetrahedra to be removed and added for each insertion and they form a connected component together. marking the removed tetrahedra and updating the list for added tetrahedra takes $O(n_o)$ time. Thus, inserting all valid circumcenters takes $O(n_o^2)$ time. Algorithm *2D-TRI* cannot take more than $O(n_o^2)$ time [8]. Hence, *3D-TRI* takes $O(n_o^2) = O(n^2A^6)$ time and $O(n_o) = O(nA^3)$ space.

6 Implementation Issues

We consider the problem of numerical errors under finite precision arithmetic while implementing the algorithm *3D-TRI*. The basic numerical computation in the incremental Delaunay triangulation is the insphere test. This test tells us whether a point is inside the circumscribing sphere of a tetrahedron or not. In presence of numerical error this test may provide inaccurate answers. This, in turn, may cause the program to fail. To overcome this problem, we use certain topological properties of the 3D triangulations to guide the answers of the insphere tests. The four necessary properties are (i) for each vertex v, the subgraph spanned by the set V of vertices adjacent to v ($v \notin V$) has a planar embedding with all triangular faces except possibly the outer face, (ii) each triangular face is incident on at most two tetrahedra, (iii) Orientations of the faces in tetrahedra are such that a face incident on two tetrahedra has opposite orientations on them, (iii) the boundary of the triangulation is connected and is homeomorphic to a sphere. Of course, these properties are not sufficient to guarantee a valid triangulation, however, they are useful to reduce the inconsistency in numerical tests. In [17] we give an algorithm for 3D Delaunay triangulation that never fails and the output always satisfies the above four conditions.

Another difficulty that arises under finite precision arithmetic is the following. With numerical errors, the computed points on the boundary facets may not be exactly coplanar, and without proper care they may form very thin tetrahedra. While constructing the triangulation of the point set obtained by triangulating all boundary facets, we take into account the topological constraint that the points generated on a boundary facet are coplanar. We have implemented our good triangulation algorithm on SUN workstations in AKCL. An example where a convex polytope is triangulated is shown in figure 10. For clarity we show only the triangulations on the facets. The original polytope had 8 factes. The final number of facets produced is 170.

7 Conclusion

The good triangulation algorithm of convex polyhedra together with the convex decomposition algorithm of nonconvex polyhedra [9] gives a method for good triangulations of nonconvex polyhedra as well. However, this method has the limitation that the convex polyhedra produced by the convex decomposition algorithm may be very bad in shape. An algorithm that achieves good triangulations directly of nonconvex polyhedra is more practical.

Though, in our algorithm we avoided type(i) through type(iv) tetrahedra, we could not avoid some special type of slivers i.e., type(v) tetrahedra. Our immediate goal is to find a new method or modify this algorithm so that we can avoid these slivers too. The difficulty with the avoidance of these slivers comes from the fact that an upper bound on the radius of circumscribing sphere and a lower bound on lengths of the edges of a tetrahedron do not prohibit it to be a type(v) tetrahedron. A lower bound on the radius of the inscribing sphere together with an upper bound on the radius of the circumscribing sphere of a tetrahedron avoids such tetrahedra. But, currently we are unable to achieve both these bounds simultaneously.

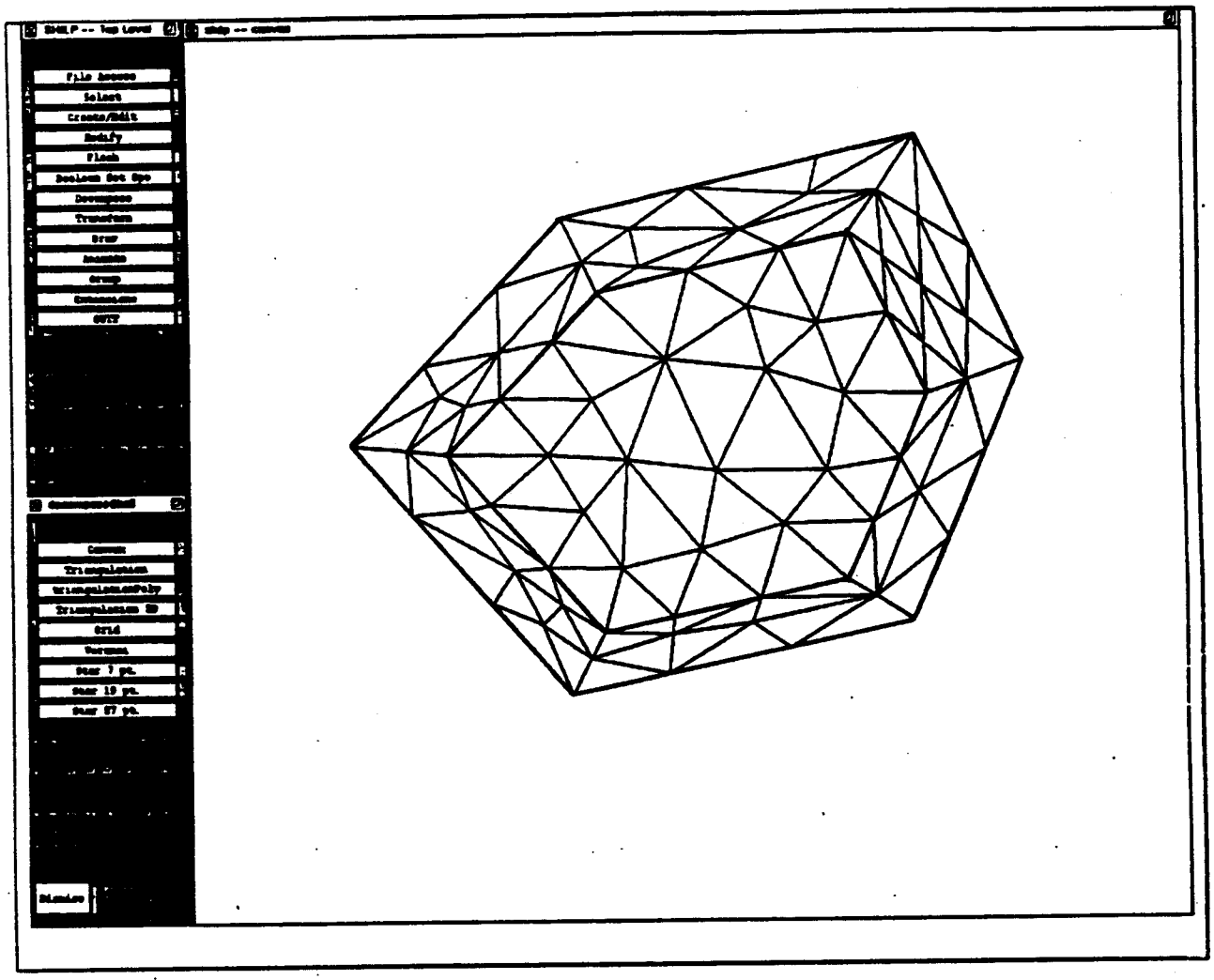

Figure 10: Good triangulation of a convex polytope

Acknowledgements: We are thankful to two referees for their valuable comments. A preliminary version of this paper appeared in the *Proc. of the Symposium on Solid Modeling Foundations and CAD/CAM Applications, 1991, Texas, pp. 431-441.*

References

[1] D. Avis and H. ELGindy, (1986), "Triangulating Simplicial Point Sets in Space", *Proc. 2nd. Ann. ACM Symposium on Computational Geometry*, pp. 133-141.

[2] Babuska and A.K.Aziz, (1976), "On the Angle Condition in the Finite Element Method", *SIAM Numerical Analysis*, 13, 214-226.

[3] B. S. Baker, E. Grosse, and C.S. Rafferty, (1988), "Nonobtuse Triangulation of Polygons", *Discrete and Computational Geometry*, 3, 147-168.

[4] T.J. Baker, (1989), "Automatic Mesh Generation for Complex Three-Dimensional Regions Using a Constrained Delaunay Triangulation", *Engineering with Computers*, 5, 161-175.

[5] M. Bern, D. Eppstein, and J. Gilbert, (1990), "Provably Good Mesh Generation", *Proc. 31st Annual IEEE Symposium on Foundations of Computer Science*, pp. 231-241.

[6] B. Chazelle and L. Palios, (1990), "Triangulating a Non-convex Polytope", *Discrete and Computational Geometry*, 5, pp. 505-526.

[7] L. P. Chew, (1989), "Guaranteed-Quality Triangular Meshes", Technical Report TR-89-983, Cornell University.

[8] T. Dey, (1990), "Good Triangulations in Plane", *Proc. of Second Canadian Conference in Computational Geometry*, 102-106.

[9] T. Dey, (1991), "Triangulation and CSG Representation of Polyhedra with Arbitrary Genus", to appear in the *Proc. of 7th Annual Symposium on Computational Geometry*, to be held at North Conway, New Hampshire, 10-12 June, 1991.

[10] H. Edelsbrunner,(1989), "Spatial Triangulations with Dihedral Angle Conditions", *Proc. of Intl. Workshop on Discrete Algorithms and Complexity, Fukuoka, Japan*, 83-89.

[11] H. Edelsbrunner, F.P. Preparata and D.B. West, (1986), "Tetrahedrizing Point Sets in Three Dimensions", Tech. Report UIUCDCS-R-86-1310.

[12] I. Fried, (1972), "Condition of Finite Element Matrices Generated from Nonuniform Meshes", *AIAA J.*, 10, pp. 219-221.

[13] B. Joe., (1989), "Three-dimensional Triangulations from Local Transformations", *SIAM J. Sci. Stat. Comput.*, 10, pp. 718-741.

[14] S.A. Mitchell, and S.A. Vavasis, (1990), "Quality Mesh Generation in Three Dimensions", unpublished manuscript.

[15] F.P. Preparata, and M.I. Shamos, (1986), "Computational Geometry, An Introduction", *Springer-Verlag.*

[16] D.T. Lee, and A.K. Lin,(1986), "Generalized Delaunay triangulation for planar graphs", *Discrete and Computational Geometry*, 1, 201-217.

[17] T. Dey, K. Sugihara, and C. Bajaj, (1991), "Triangulations in Three Dimensions with Finite Precision Arithmetic", in preparation.

[18] D. F. Watson, (1981), "Computing the n-Dimensional Tesselation with Applications to Voronoi Polytopes", *The Computer Journal*, 24, pp. 167-172.