Computers in Industry 41 (2000) 65–81

COMPUTERS IN INDUSTRY

www.elsevier.nl/locate/compind

Tetrahedral mesh generation for solids based on alternating sum of volumes

Kai Tang [a], Shuo-Yan Chou [b,*], Lin-Lin Chen [c], Tony C. Woo [d]

[a] *Applicon, Ann Arbor, MI, USA*

[b] *Department of Industrial Management, National Taiwan University of Science and Technology, 43 Keelung Road, Section 4, Taipei, Taiwan*

[c] *Department of Industrial and Commercial Design, National Taiwan University of Science and Technology, Taipei, Taiwan*

[d] *Department of Industrial Engineering, University of Washington, Seattle, WA, USA*

Received 13 June 1997; accepted 19 February 1999

Abstract

Decomposition of a three-dimensional non-convex polyhedral object into tetrahedra using few or no 'Steiner' points assumes both theoretical and practical importance. It has been known that the determination of whether a polyhedron can be tetrahedralized is NP-complete. This prompts the investigation of the tetrahedralization of special classes of polyhedra, including convex, star-shaped, monotone, and isothetic. This paper identifies a special class of polyhedra that can be tetrahedralized without using 'Steiner' points. The proposed tetrahedralization algorithm utilizes a structure provided by the *alternating sum of volumes* process (a convex decomposition method) so that a complex solid object can first be decomposed into a set of simpler objects, namely conjuncts. The concatenation of the tetrahedralization of these conjuncts gives rise to the tetrahedralization of the original solid object. © 2000 Elsevier Science B.V. All rights reserved.

Keywords: Mesh generation; Tetrahedralization; Finite element methods; Alternating sum of volumes; Polyhedron decomposition

1. Introduction

Polyhedron tetrahedralization is the process of dividing a polyhedron into a set of non-overlapping tetrahedra whose union equals the polyhedron. Such decomposition is one of the most popular techniques used in cell/volume decomposition methods in solid modeling systems and has applications in many engineering disciplines [7,9,10,14,20,21,28,29,33,38]. The decomposition serves as a simplification scheme for dealing with complex objects or as a pre-processing process for generating finer mesh in the CAD/CAM/CAE systems.

In the computer-aided design system [28], complex objects are sometimes decomposed into a set of primitive objects—such as tetrahedra, hexahedra, convex polyhedra—so that they can be manipulated more efficiently [32,36]. Algorithms that are capable

* Corresponding author. E-mail: sychou@im.ntust.edu.tw

PII: S0166-3615(99)00008-1

of manipulating individual basic objects as well as merging the results of which can be subsequently developed. Similarly, for applications such as robot motion planning [8,29,33], the goal of decomposition is to simplify the geometry of the robot and the obstacles so that a complex path planning problem can be transformed into the aggregation of a set of path planning problems with much simpler objects. This divide-and-conquer scheme is in general more efficient and requires less effort for implementation.

For finite element mesh generation [21], the solid defined by a solid modeling system is decomposed into a set of non-overlapping tetrahedral [9,10,14, 20,38] or hexahedral [7] elements. With the decomposition, engineering analysis for properties such as stress, vibration, thermal, electromagnetic and fluid flow of an engineering design can be performed even before its prototype is made. These analysis processes significantly reduce the time required for product development and further improve the quality of products designed.

Few development of tetrahedralization algorithms were reported because the tetrahedralization of a solid in fact falls in the category of NP-completeness. Earlier development of heuristics all adopted direct decomposition approaches. Among those published methods, Delaunay tetrahedralization [12,22,24] is the most widely adopted one in which a set of nodes, namely 'Steiner' points (cf. [27]), is distributed on the boundary and/or inside the volume of the given solid, and an initial Delaunay tetrahedralization is computed. The initial tetrahedralization in general needs to be further refined by adding more nodes to the solid. Methods that guarantee to derive a good initial set of nodes, for example, methods that do not result in tetrahedra with sharp angles, remain an important research topic.

Since any tetrahedralization would serve the purpose, the number of resulting tetrahedra should be minimized. Consequently, only the vertices of the original solid are allowed to be the vertices of the resulting tetrahedra. However, it is well known that there exist polyhedra that cannot be tetrahedralized without the use of Steiner points. Fig. 1 shows a such example due to Schönhardt [27]. The P' in Fig. 1(b) is obtained by rotating the top face of the triangular prism P in Fig. 1(a) about its normal by a small angle. After the rotation, the three faces adjacent to the top face are no longer planar; each of them has to 'bend in' along the appropriate diagonals, resulting in two triangular faces. Polyhedron P' does not yield a single tetrahedron with all four of its vertices from P', and thus cannot be tetrahedralized.

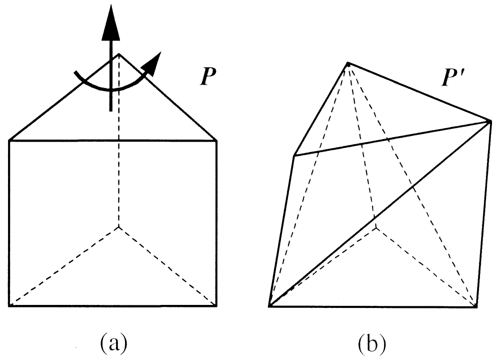


Fig. 1. A polyhedron that is not tetrahedralizable unless Steiner points are used.

How to tetrahedralize a polyhedron without using Steiner points or with a minimal number of Steiner points has long been pursued by both theoreticians and practitioners. A tetrahedralization without Steiner points usually results in fewer tetrahedra and is less prone to numerical errors. In the following sections, a polyhedron is said to be tetrahedralizable only if the tetrahedralization does not require the use of Steiner points, unless noted otherwise. But Ruppert and Seidel [26] have recently shown that the problem of deciding whether a given polyhedron can be tetrahedralized without using Steiner points is NP-complete. This prompts the investigation of the tetrahedralization of special classes of polyhedra, including those that are convex, star-shaped, monotone, rectangular, and isothetic. Ominously, except for the trivial case of convex polyhedra, all the other classes are found to have cases that cannot be tetrahedralized without using Steiner points. (The polyhedron P' in Fig. 1 is both monotone and star-shaped.)

This paper identifies a special class of polyhedra that are always tetrahedralizable. Unlike those classes of polyhedra that are characterized by intrinsic geometric properties (e.g., convexity and monotonicity), this class is by its expressibility with respect to a representation called the *Alternating Sum of Volumes* (ASV). If a polyhedron can be represented in ASV, it is said to be ASV-describable. To be more specific, a polyhedron P is ASV-describable if it can be represented by a series of convex polyhedra H_is, $i = 1, 2, \ldots, n$, satisfying the following three properties:

(i) *sign alternation*: $P = \sum_{i=1}^{n}(-1)^{i-1}H_i$

(ii) *volume monotonicity*: $H_i \supset H_{i+1}$, $i = 1, 2, \ldots, n-1$, and

(iii) *vertex inclusion*: $H_i = \mathrm{CH}(\{v | v \in P\})$, $i = 1, 2, \ldots, n$.

Fig. 2 provides a pictorial illustration of an ASV-describable polyhedron P of genus one. The symbols '+' and '−' denote *set union* and *regularized set difference* operations [25], respectively. The convex polyhedra H_is are referred to be the layers of

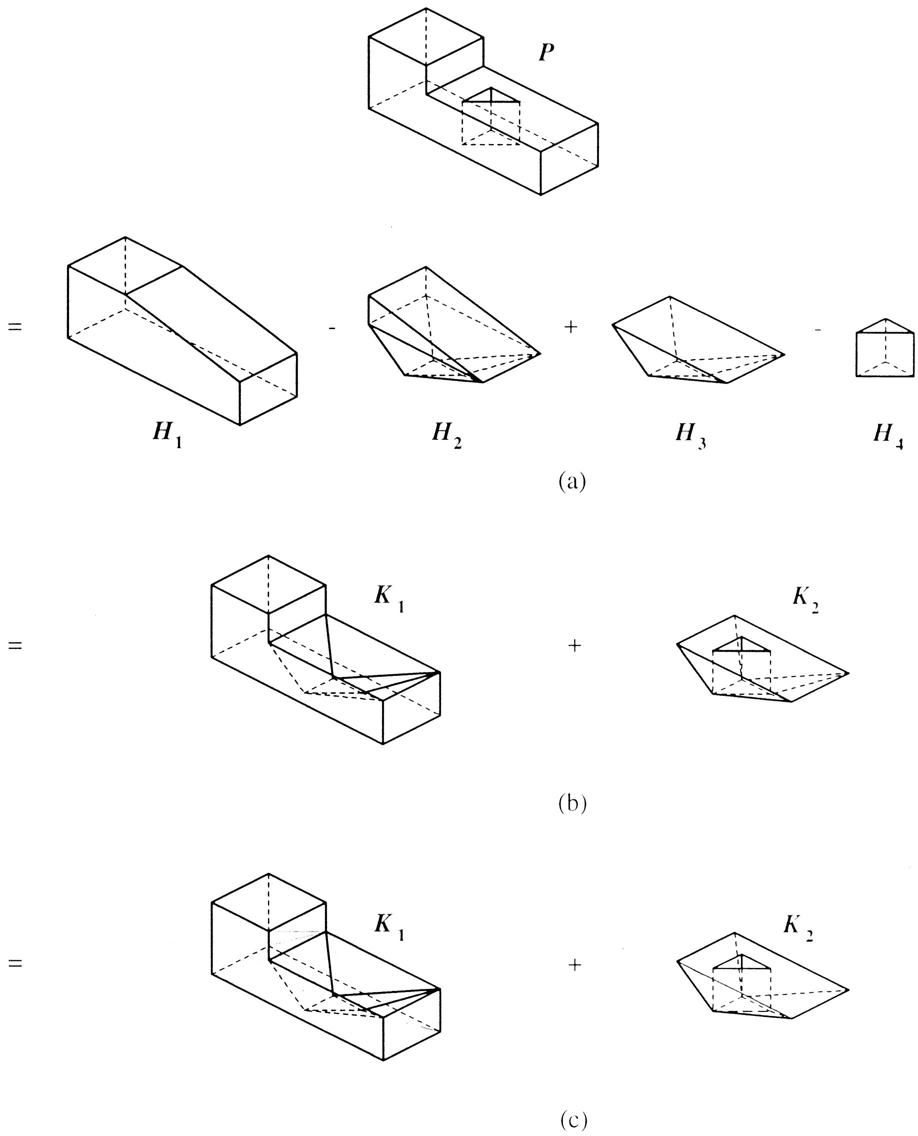


Fig. 2. An ASV-describable polyhedron P and its tetrahedralization.

the ASV series. The derivation of the equation $P = H_1 - H_2 + H_3 - H_4$ (as shown in Fig. 2(a)) can be observed from the construction procedure shown in Fig. 4.

The algorithm for tetrahedralizing an ASV-describable polyhedron relies on the notion of a conjunctive form and the tetrahedralization of a conjunct. An ASV series can be converted into a *conjunctive form* if all of its regularized set difference operations are evaluated, yielding a union of *conjuncts* K_i:

$$\begin{aligned} P &= H_1 - H_2 + H_3 - H_4 + \cdots \\ &= (H_1 - H_2) + (H_3 - H_4) + \cdots \\ &= K_1 + K_2 + \cdots \end{aligned}$$

Fig. 2(b) illustrates this *conjunction* process. Note that due to the volume monotonicity property $(H_i \supset H_{i+1})$ of the ASV-describable polyhedron, conjuncts $K_i s$ are always valid manifolds in $\mathscr{R}^3$. It will be shown in Section 3 that a conjunct can always be tetrahedralized. Again, by the volume monotonicity property, conjuncts do not overlap. As a result, the collection of the tetrahedralization of the conjuncts yields a tetrahedralization of P.

It should be remarked that the algorithm above expects an even number of terms in the underlying ASV series. If the number of terms is odd, a conjunctive form can still be achieved with no difficulty as we can always append a null object to the end of the series without affecting the resulting polyhedron.

The class of polyhedra considered in this paper can have arbitrary numbers of handles, internal voids, and two types of degeneracies (solids that meet at a vertex or an edge). Fig. 3(a) shows a polyhedron that contains a handle, an internal void, and two degeneracies. Such a class of polyhedra is referred as *pseudo-polyhedra*. However, pseudo-polyhedra cannot contain dangling faces or edges, as shown in Fig. 3(b). Data structures such as the non-manifold data structure [34], the quintuple lists [30], and the star-edge representation [3] can be used for representing pseudo-polyhedra.

Algorithm ASV_TETRA(P)

/* Tetrahedralize an ASV-describable polyhedron P */

begin

Phase 1: Check if P is ASV-describable. If it is, compute an ASV series for P and put them into a conjunctive form. $P = (H_1 - H_2) + \cdots + (H_{2i-1} - H_{2i}) + \cdots + (H_{2k-1} - H_{2k})$

Phase 2: Tetrahedralize the conjuncts $K_i = (H_{2i-1} - H_{2i})$, $i = 1, 2, \ldots, k$.

Glue the tetrahedralized conjuncts to form a tetrahedralization of P.

end.

The rest of this paper will give a detailed account on how the two phases of algorithm ASV_TETRA

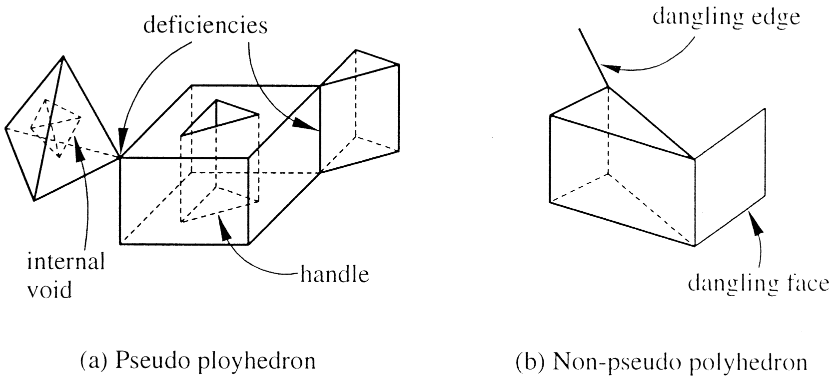


Fig. 3. Classification of polyhedra.

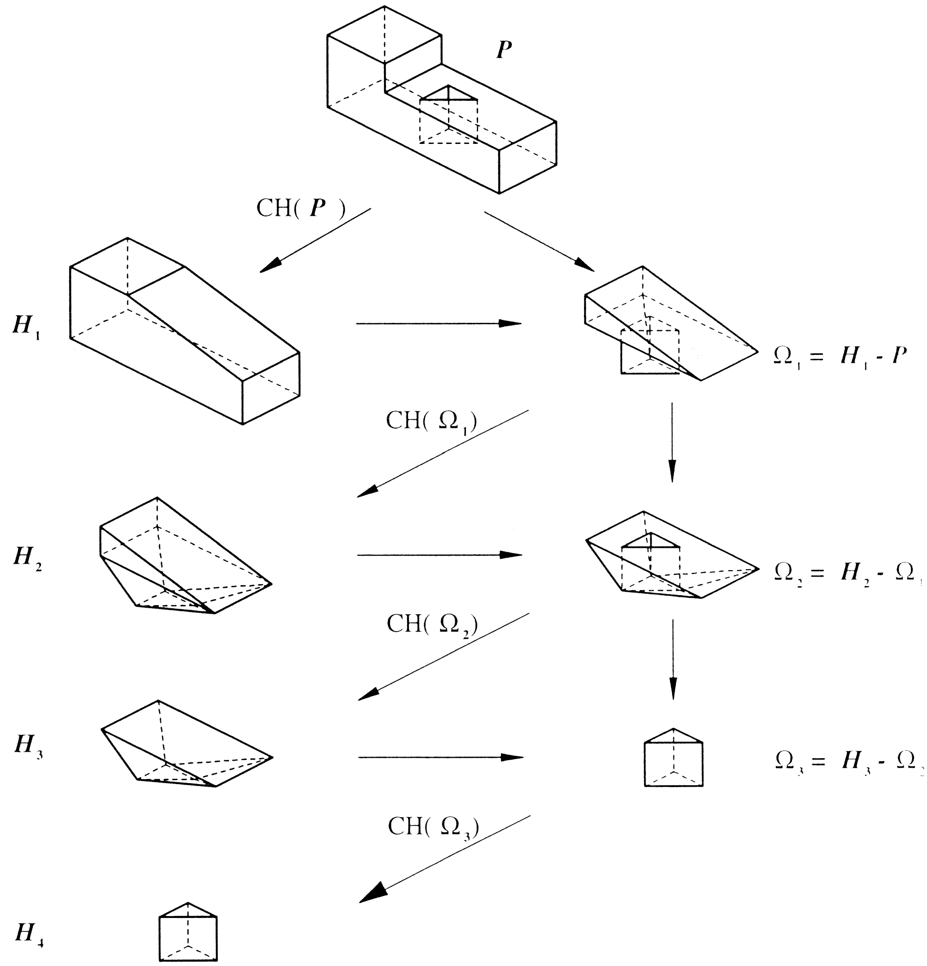


Fig. 4. Derivation of an ASV series of P.

are realized. In Section 2, a particular recursive algorithm that computes an ASV series for a polyhedron is briefly reviewed. A necessary and sufficient condition for a polyhedron to be ASV-describable is then proved, thus providing a solution for the first phase. In Section 3, the second phase is validated by introducing two algorithms for tetrahedralizing a conjunct, the beneath–beyond method and the cup-removal method, both achieving the goal without using the Steiner points. In Section 4, extensions of the algorithm to the tetrahedralization of non-ASV-describable polyhedra are discussed. Finally, the paper is concluded in Section 5. The space linearity theorem, which links the space linearity of the tetrahedralization of a pseudo-polyhedron to the space linearity of the tetrahedralization of its individual conjuncts is given in the appendix.

2. Alternating sum of volumes

Let us now briefly review the algorithm given in Ref. [35] for computing an ASV series of a polyhedron. The algorithm, to be referred to as HULL_DIFF, executes in alternation two operations

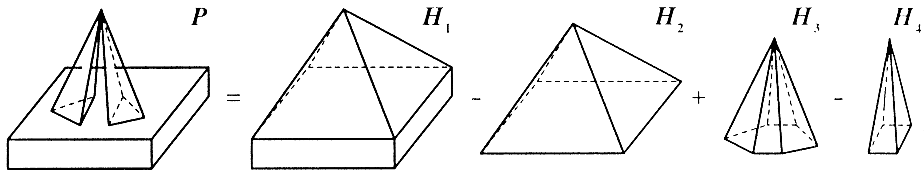


Fig. 5. An ASV-describable non-two-manifold.

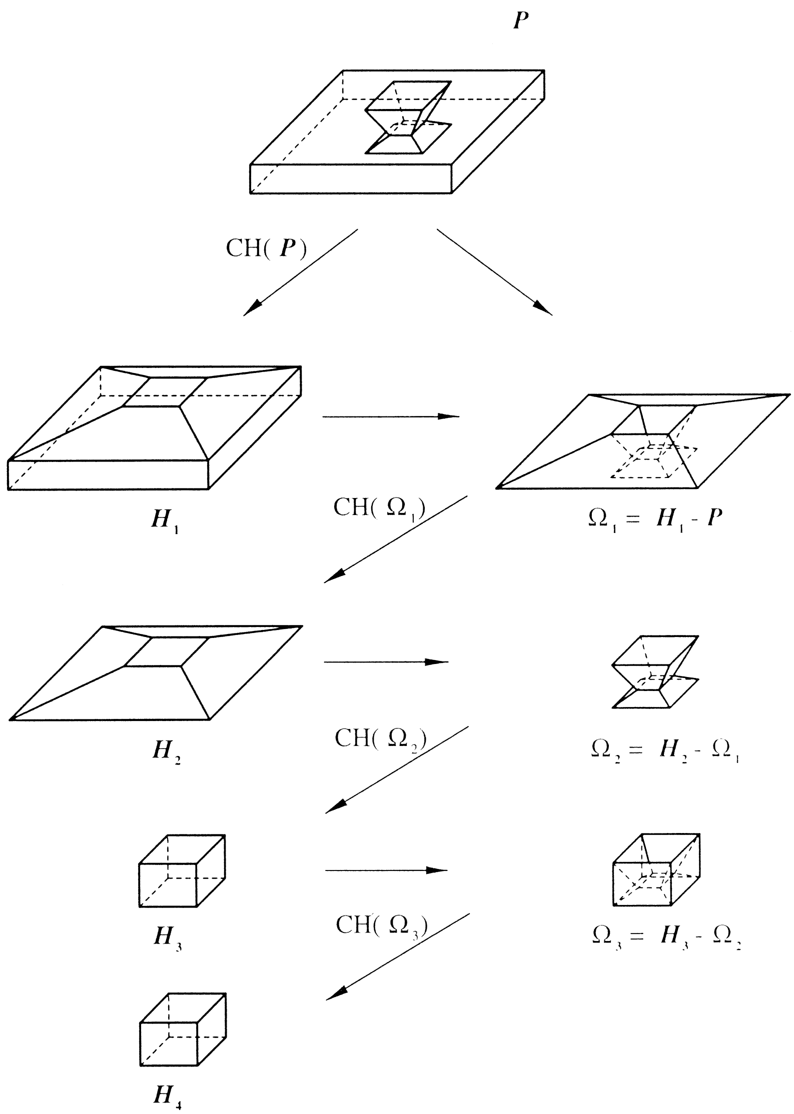


Fig. 6. A polyhedron that is tetrahedralizable but fails HULL_DIFF.

—convex hull construction and regularized set difference—and outputs convex components H_1, $H_2, \ldots, H_n$ as the ASV series of a given polyhedron P.

Algorithm HULL_DIFF(P)
/* Compute an ASV series of a polyhedron P*/
begin
  $H \leftarrow$ convex hull of P;
  Output H;
  IF $P \neq H$ then begin
    $\Omega \leftarrow H - P$;
    HULL_DIFF(Ω);
  end;
end.

Fig. 4 illustrates how the $H_i s$ in the ASV series shown in Fig. 2 are derived by using algorithm HULL_DIFF(P). While the convex polyhedron H_i is the convex hull of Ω_{i-1} (except for H_1, which is the convex hull of P), the deficiency Ω_i is obtained by subtracting Ω_{i-1} from H_i ($\Omega_1 = H_1 - P$). The derivation terminates when a resulting deficiency Ω_i is a null set. By applying to HULL_DIFF(P), the uniqueness of the resulting series $H_i s$ as well as the satisfaction of the three properties of an ASV series can be verified easily.

By using the algorithm HULL_DIFF, Tang and Woo [30,31] gave a simple $O(N \log N)$ algorithm for computing the regularized set difference between a pseudo-polyhedron and its convex hull, where N is the number of faces on the pseudo-polyhedron. They also showed that by using the algorithm HULL_DIFF an ASV series of a pseudo-polyhedron with N faces can be obtained in $O(N^2 \log N)$ time and with linear space. As degeneracies are most likely to be formed during the intermediate stages, the class of pseudo-polyhedra is therefore considered. Fig. 5 shows the ASV series of a pseudo-polyhedron P, obtained by performing the algorithm HULL_DIFF(P).

The recursive algorithm HULL_DIFF may fail to produce an ASV series in some cases, and is thus unable to facilitate the tetrahedralization. This problem occurs when there exists a deficiency Ω_i such that $\mathrm{CH}(\Omega_i) = \mathrm{CH}(\mathrm{CH}(\Omega_i) - \Omega_i)$, where CH denotes the convex hull operation (see Fig. 6). This possibility of not being able to generate a valid ASV series comes as no surprise for otherwise one would conclude that every polyhedron is tetrahedralizable, which as stated earlier is not true. For example, the polyhedron P in Fig. 1 is not tetrahedralizable, and the algorithm HULL_DIFF does not terminate on P as well. However, there exist certain polyhedra that are tetrahedralizable but may fail through HULL_DIFF, e.g., the example shown in Fig. 6. This again should be expected for otherwise one would be able to decide in polynomial time whether a given polyhedron is tetrahedralizable or not, a direct conflict to the NP-completeness of the decision problem for tetrahedralizability [26]. The ability to produce a valid ASV series is therefore a necessary but not sufficient condition for the ability to tetrahedralize a pseudo-polyhedron.

Before moving on to Section 3, it is important to answer a fundamental question. As we alluded to earlier in Section 1, the algorithm ASV_TETRA first checks if a pseudo-polyhedron is ASV-describable,

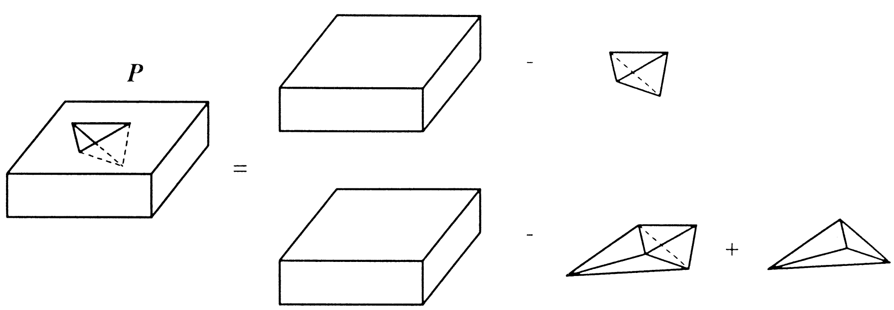


Fig. 7. Two ASV series of a same polyhedron.

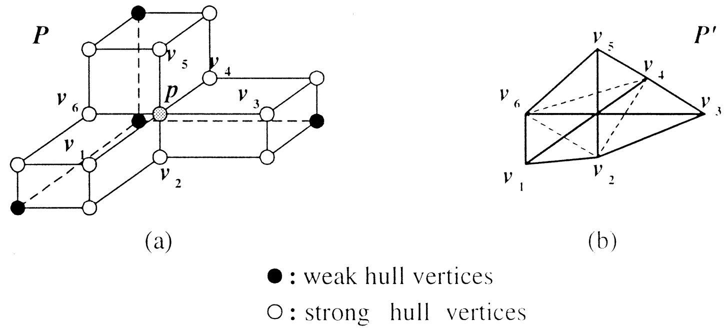


Fig. 8. Classification of hull vertices and illustration of Lemma 1.

and if it is, an ASV series is then computed. How exactly is this describability checked? It is noted that a pseudo-polyhedron may correspond to more than one ASV series, as manifested by the example shown in Fig. 7, where the one with only two layers is obtained via the algorithm HULL_DIFF. Intuitively, if the algorithm HULL_DIFF succeeds (terminates) on a pseudo-polyhedron P, the resulting ASV series has the smallest number of terms among all the possible ASV series of P. Conversely, if the algorithm HULL_DIFF does not terminate on P, it is most likely that P does not have an ASV series. It is proved next that a pseudo-polyhedron P is ASV-describable if and only if the algorithm HULL_DIFF would terminate on P.

To establish the conditions for termination, the vertices for a pseudo-polyhedron P are classified. A vertex v of P is called a *hull vertex* if v is also a vertex of CH(P). Vertex v is further called a *weak hull vertex* if there exists a neighborhood ball α centered at v such that the two regularized intersections $\alpha \cap^* P$ and $\alpha \cap^* \mathrm{CH}(P)$ are identical; otherwise, v is referred to be a *strong hull vertex*. Fig. 8(a) shows an example of this classification. Let P be said to be self-cyclic if it satisfies the criterion $\mathrm{CH}(P) = \mathrm{CH}(\mathrm{CH}(P) - P)$. It is shown in [31] that P is self-cyclic if and only if all the hull vertices of P are strong. For example, Ω_2 in Fig. 6 is self-cyclic since all its hull vertices are strong. The following lemma establishes the condition on which HULL_DIFF terminates.

Lemma 1. *The algorithm HULL_DIFF will not terminate on a pseudo-polyhedron P if and only if there exists a subset* $\{v_1, v_2, \dots, v_k\}$ *of vertices of P, called an eigen set, such that the regularized set intersection* $P \cap^* CH(v_1, v_2, \dots, v_k)$ *is self-cyclic.*

As an illustration to Lemma 1, consider the six vertices $\{v_1, v_2, \dots, v_6\}$ of P, shown in Fig. 8(a). The regularized set intersection between the convex hull $\mathrm{CH}(v_1, v_2, \dots, v_6)$ and P results in a pseudo-polyhedron P', as shown in Fig. 8(b). By Lemma 1, the algorithm HULL_DIFF does not terminate on P' since P' is self-cyclic. On the other hand, if such a P' cannot be found, the lemma asserts that HULL_DIFF must terminate, i.e., P is ASV-describable. Notice that each v_i $(i \le 6)$ is also a vertex of $\mathrm{CH}(v_1, v_2, \dots, v_6)$. Such an eigen set is said to be *irreducible*. An eigen set may also be reducible, e.g., the set $\{v_1, v_2, \dots, v_6, p\}$ in Fig. 8(a). Con-

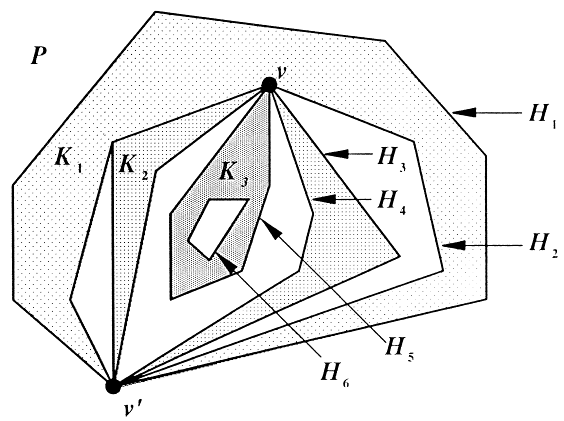


Fig. 9. Residual layer of a vertex.

ceivably, any reducible eigen set can always be reduced to an irreducible eigen set.

Suppose P is ASV-describable with an **ASV** series $H_1 - H_2 + H_3 - H_4 + \cdots + H_{n-1} - H_n$. Let K_i denote the conjunct $(H_{2i-1} - H_{2i})$, $i = 1, 2, \ldots, n/2$. Every vertex of P must also be the vertex on at least one layer in the ASV series. Of these layers, the most inner one is referred to be the *residual layer* of that vertex. Fig. 9 illustrates this definition with a two-dimensional example. Vertex v belongs to five layers H_1, H_2, H_3, H_4, and H_5; its residual layer is H_5. On the other hand, the residual layer of vertex v' is H_4. As it turns out, the introduction of the residual layer greatly expedites the proof for the ASV-describability, which is presented in the following theorem.

Theorem 1. *A pseudo-polyhedron P is ASV-describable if and only if the algorithm HULL_DIFF terminates on P*.

Proof. Suppose procedure HULL_DIFF(P) does not terminate, yet P can be represented by a (finite) ASV series $H_1 - H_2 + H_3 - H_4 + \cdots + H_{n-1} - H_n$. By Lemma 1, P has at least one irreducible eigen set, $\{v_1, v_2, \ldots, v_k\}$, such that the regularized set intersection P' between P and CH$(v_1, v_2, \ldots, v_k)$ is self-cyclic.

Consider an arbitrary vertex v_1 in the eigen set. Without loss of generality let $v_2, v_3, \ldots, v_m$ $(m \le k)$ be the vertices on CH$(v_1, v_2, \ldots, v_k)$ that are adjacent to v_1. (Two vertices are said to be adjacent to each other if they are connected by an edge.) Also, let H_r be the residual layer of v_1. We show next that at least one of the vertices $\{v_2, v_3, \ldots, v_m\}$ must be strictly outside the layer H_r. Only two cases are possible.

Case 1. When H_r is the outer boundary of a conjunct, i.e., r is an odd number. Refer to Fig. 10(a). If all the adjacent vertices $\{v_2, v_3, \ldots, v_m\}$ are inside or on H_r, v_1 always has a sufficiently small neighborhood ball α such that the open interior of the intersection $\alpha \cap^* \mathrm{CH}(v_1, v_2, \ldots, v_m)$ is strictly inside the conjunct $(H_r - H_{r+1})$. This means that v_1 cannot be a strong hull vertex of P', which contradicts the assumption that P' is self-cyclic.

Case 2. H_r is the inner boundary of a conjunct, i.e., r is an even number. As shown in Fig. 10(b), if all the adjacent vertices $\{v_2, v_3, \ldots, v_m\}$ are inside or on H_r, we can always find a sufficiently small neighborhood ball α of v_1 such that the open interior of the intersection $\alpha \cap^* \mathrm{CH}(v_1, v_2, \ldots, v_m)$ is strictly inside the boundary H_r and strictly outside the boundary H_{r+1}. In other words, the regularized intersection $\alpha \cap^* P \cap^* P'$ is empty. By the very definition of regularized set intersection, this means that v_1 cannot be a vertex of P' (since it 'floats' in the air), conflicting our supposition that the eigen set $\{v_1, v_2, \ldots, v_k\}$ is irreducible.

Now, let v_h be the vertex in $\{v_1, v_2, \ldots, v_k\}$ whose residual layer is the largest among all the residual layers. Due to the volume monotonicity of an ASV series, the residual layer of every other vertex in $\{v_1, v_2, \ldots, v_k\}$ is inside this largest layer.

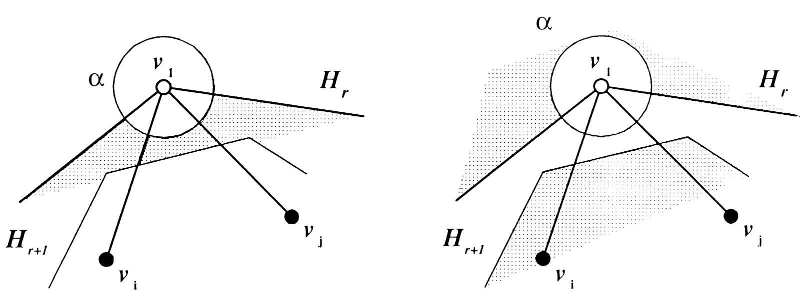


Fig. 10. Proof of Theorem 1.

Therefore, all the vertices in $\{v_1, v_2, \ldots, v_k\}$ are either strictly inside or on the residual layer of v_h. This is however contradictory to what we have just shown. Q.E.D.

3. Tetrahedralization of a conjunct

By performing HULL_DIFF, once it is verified that a pseudo-polyhedron P is ASV-describable, we will readily have an ASV series $\{H_1 - H_2 + \cdots + H_{n-1} - H_n\}$ for P. This section discusses how a conjunct $(H_i - H_{i+1})$ can be tetrahedralized. Since, by the volume monotonicity, conjuncts do not overlap (i.e., the regularized set intersection between any two conjuncts is empty), the tetrahedralization of individual conjuncts yields the tetrahedralization of P. Two algorithms are introduced for tetrahedralizing a conjunct: the beneath–beyond method and the cup-removal method.

3.1. Beneath–beyond algorithm

The first tetrahedralization algorithm is analogous, in spirit, to the beneath–beyond method given in Ref. [24] for computing the convex hull of a set of points, though differs in detail.

Let the conjunct to be tetrahedralized be represented by its two bounding convex polyhedra: the inner layer P_I and the outer layer P_O. Let v be a vertex on P_O that is strictly outside P_I. Based on the volume monotonicity, there exists at least one such vertex. Assume that the inner layer P_I has its planar faces triangulated [6,11]. The tetrahedralization starts from constructing a *supporting cone* of the polyhedron P_I with apex at v. This truncated cone consists of the line segments between v and the supporting vertices of P_I and all the edges on P_I that connect adjacent supported vertices, as demonstrated in Fig. 11. Because of the convexity of P_I, these connecting edges of P_I, darkened in Fig. 11, form a simple closed *Jordan* curve that divides the faces of P_I into two groups: those visible to v (shaded faces in Fig. 11) and those that are not. The tetrahedra determined by v and each of these visible (triangular) faces, along with P_I, constitutes a valid convex decomposition of the convex polyhedron $CH(P_I \cup v)$. Furthermore, the triangular faces of the supporting cone

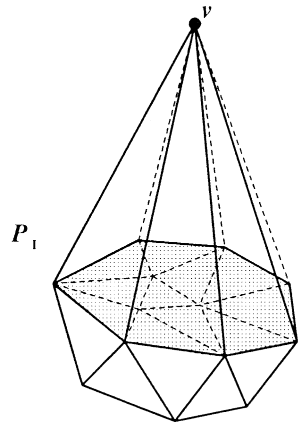


Fig. 11. Building a supporting cone.

together with those faces of P_I invisible to v are already in a triangulated representation of $CH(P_I \cup v)$. We can then construct a supporting cone of $CH(P_I \cup v)$ with respect to a vertex of P_O which is strictly outside $CH(P_I \cup v)$. This process continues until all the vertices of P_O that can be used as the apex of a supporting cone have been processed. At the end, the tetrahedra thus extracted form a tetrahedralization of the conjunct $P_O - P_I$.

Algorithm Beneath_Beyond_Tetra(P_O, P_I)
  /* Using the beneath-beyond method to tetrahedralize a conjunct bounded by two layers P_O and P_I. The inner layer P_I is assumed to have its faces triangulated and stored in a standard representation such as the quad-edge data structure [13], whereas the outer layer P_O is given as a list of vertices.*/
  begin
    $\{v_1, v_2, \ldots, v_m\}$ /* the vertices of P_O that are not any face of P_I^*/
    $P_S \leftarrow P_I$ /*P_S: the polyhedron to be computed for a supporting cone*/
    for $i = 1$ to m do begin
      Identify those faces on P_S that are visible to v_i by constructing the supporting cone of P_S with respect to v_i;
      Output the tetrahedra made of vertex v_i and the visible triangulated faces on P_S;
      $P_S \leftarrow CH(P_S \cup v_i)$
    end
  end.

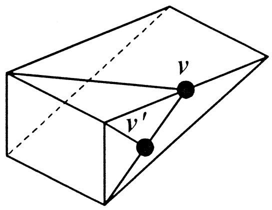


Fig. 12. Criticalness of vertices.

The above procedure realizes the beneath-beyond method for the tetrahedralization. It should be noted that the information concerning whether a vertex on P_O is strictly outside the inner layer P_I is a priori with the ASV series, because the HULL_DIFF algorithm presented in Section 2 guarantees the availability of this information once the layers of the ASV series are computed. Another subtlety worth mentioning is the *criticalness* of the vertices of P_O to P_S (the 'expanded' inner layer that is to be computed for a supporting cone). A vertex v on a polyhedron is said to be critical if there does not exist any line segment on the polyhedron that strictly contains v, e.g., for the wedge shown in Fig. 12, all the vertices except for v and v' are critical. In executing the for loop, it is assumed implicitly that all the vertices of P_O are critical to P_S. This prevents a vertex of P_O being on a face of some P_S, and thus eliminates the extraction of zero volume tetrahedra and its accompanying numerical instability problem.

The most attractive feature of the beneath-beyond tetrahedralization algorithm is its simplicity and ease of implementation. The algorithm nevertheless suffers from its negligence of the topological and geometric information of the outer layer. In the procedure Beneath_Beyond_Tetra, only the vertices from the outer layer are used, while the topological and geometric relations among them, the edges and faces, are completely ignored. As a result, the tetrahedra extracted often tend to have sharp angles when the inner layer is relatively small compared to the outer layer, which is undesirable in many applications [2,13]. Another concern with the algorithm is its inability in controlling the number of tetrahedra extracted, which typically lean to the worst case upper bound $O(m(m+m'))$, where m and m' are the numbers of vertices on the outer and inner layers, respectively. In order to avoid these two shortcomings, Section 3.2 introduces another alternative tetrahedralization algorithm, the cup-removal algorithm, in light of the algorithm in [5] for tetrahedralizing a simple polyhedron without holes or voids.

3.2. Cup-removal algorithm

To begin with, three terminologies that aid the explanation of the algorithm are introduced. Suppose v is an arbitrary critical vertex on the outer layer P_O that is strictly outside the inner layer P_I, and V is the set of vertices adjacent to v. Consider the regularized set difference between the two convex hulls $\mathrm{CH}(P_I \cup V \cup \{v\})$ and $\mathrm{CH}(P_I \cup V)$. Because v is critical, this regularized set difference can be shown to be a simple polyhedron of non-zero volume, which is referred to be the *cup* of v, or *cup*(v). The faces on *cup*(v) fall into two groups: those that belong to P_O and those that do not. The latter group of faces form a polyhedral surface, and is called the

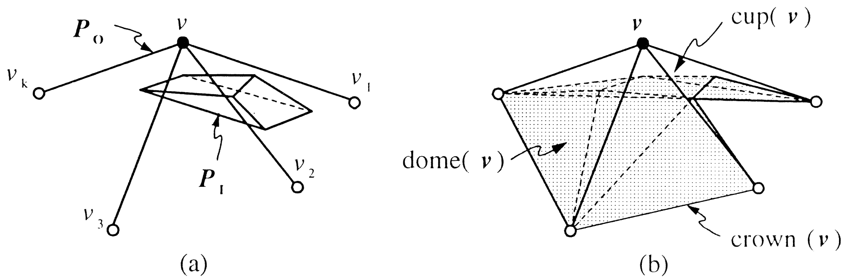


Fig. 13. Cup, Crown, and Dome.

dome of v, or *dome*(v). The set of edges on *cup*(v) that separate the two groups of faces is called the *crown*, or *crown*(v). Fig. 13 below illustrates of these three terms.

One can easily verify that all the points on *dome*(v) are visible to the vertex v. That is, the line segment connecting v and any point on *dome*(v) does not intersect *dome*(v) except at that point. Moreover, the line segment between v and an arbitrary point on *crown*(v) always lies on the boundary of *cup*(v). These two observations imply a straightforward tetrahedralization for *cup*(v). Specifically, assuming the faces on *dome*(v) are triangulated [1,2,18,19], the tetrahedra formed by vertex v and the triangulated faces of *dome*(v) yield a tetrahedralization of *cup*(v).

To correlate the tetrahedralization of a cup with that of a conjunct, a peeling procedure for decomposing a conjunct is introduced. It is noted that *cup*(v) equals the regularized set difference $\mathrm{CH}(P_O) - \mathrm{CH}(P_I \cup V')$, where V' is the set of vertices on P_O excluding v. After removing *cup*(v), the remaining convex polyhedron $\mathrm{CH}(P_I \cup V')$, which is treated as the outer layer, along with the inner layer P_I, gives rise to a new conjunct $\mathrm{CH}(P_I \cup V') - P_I$. The new conjunct is clearly a subset of the original one. This cup-removal process continues until V' becomes empty, in which case, the original conjunct $P_O - P_I$ would have been decomposed into a number of mutually exclusive cups.

The following procedure provides a recursive description of the prescribed cup-removal process. In the procedure, a ‘gluing’ operation is used to construct the new outer layer for its subsequent iteration (Step 5). This is based on the observation that *crown*(v) is always a *Jordan* curve on P_O that divides the boundary of P_O into two regions: the one on *cup*(v) and the one not.

Algorithm Cup_Removal_Tetra(P_O, P_I)
/* Tetrahedralize a conjunct defined by the outer layer P_O and the inner layer P_I */
begin
Step 1. if $P_O = P_I$ then return
Step 2. select v /* select a critical vertex on P_O that is strictly outside P_I and whose degree is the smallest */
Step 3. compute *cup*(v), *dome*(v), and *crown*(v)
Step 4. output the tetrahedra defined by v and the faces on *dome*(v)
Step 5. construct P_{new} /* construct the convex polyhedron defined by *dome*(v) and those faces on P_O that do not belong to *cup*(v) */
Step 6. Cup_Removal_Tetra(P_{new}, P_I)
end.

Although any vertex on the outer layer is eligible to be selected at Step 2 (as long as it is critical and outside the inner layer) the vertex selected is forced to have the smallest degree possible. This is to minimize the number of tetrahedra extracted. Consider the conjunct in Fig. 14 whose outer layer is due to Chazelle [5]. If the cups of the vertices $u_k, u_{k-1}, \ldots, u_2$ are removed in that order, due to the insignificance of the inner layer, the total number of tetrahedra produced can be easily seen to be $O(k^2)$. If, however, the cups of the same vertices are removed in the reverse order, only $O(k)$ tetrahedra are formed. This observation suggests that the cups of the vertices of a smaller degree should be removed first.

By means of Euler formula for planar graphs, one can deduce that the procedure Cup_Removal_Tetra always decomposes a convex polyhedron P into a number of tetrahedra bounded linearly by the size of P [4]. For the case of a conjunct, i.e., a convex polyhedron containing a convex void, it is conjectured that in general the number of tetrahedra gener-

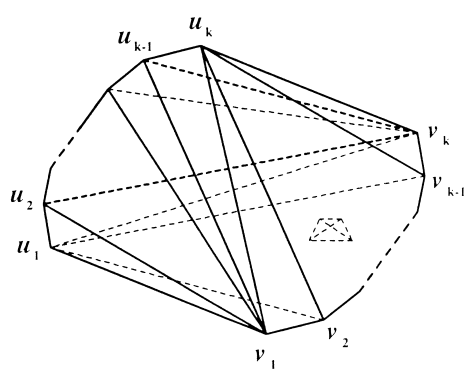


Fig. 14. A conjunct that can result in linear or quadratical numbers of tetrahedra.

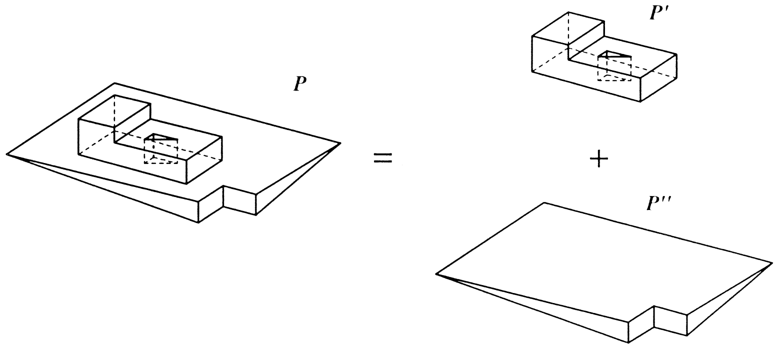


Fig. 15. Remedy of a self-cyclic polyhedron by partition.

ated by the Cup_Removal_Tetra algorithm should still be linear to the size of P.

4. Application to non-ASV-describable objects

Despite that our algorithm is designed to deal with the class of ASV-describable pseudo-polyhedra, it also shed light on the general tetrahedralization, i.e., allowing the placing of Steiner points.

One can easily show that a pseudo-polyhedron P can always be uniquely and deterministically decomposed into two disjoint pseudo-polyhedra P_A and P_B, such that P_A is ASV-describable whereas P_B is self-cyclic, i.e., $\mathrm{CH}(P_B) = \mathrm{CH}(\mathrm{CH}(P_B) - P_B)$. P_A and P_B are called the ASV and non-ASV components of P, respectively [16,17,37]. For instance, the polyhedron P shown in Fig. 6 is not ASV-describable; its ASV component is the sole conjunct $H_1 - H_2$, and its non-ASV component is the deficiency Ω_2. As the ASV component is tetrahedralizable, only the tetrahedralization of the non-ASV component may require using Steiner points, which can be accomplished by means of known decomposition algorithms (Chazelle and Palios' unhindered cup-removal method [5] and Bajaj and Dey's notch elimination approach [3]). It is fair to say that, since the ASV component brings down the number of vertices on the non-ASV component (due to the disjointedness between P_A and P_B and the fact that their vertices are also the vertices of P), the number of Steiner points introduced on P_B should be expected to be less than the number of Steiner points introduced when P is considered as a whole. In addition, because the ASV process often turns a reflex edge [1] on P into a convex edge on P_B, as illustrated in Fig. 6, it facilitates those tetrahedralization algorithms that are particularly sensitive to the number of reflex edges, e.g., the algorithm given in Ref. [5].

A more intriguing provision is to partition a self-cyclic P into a small number of parts such that each of them is by itself ASV-describable. In the example shown in Fig. 15, by partitioning the self-cyclic polyhedron P into two ASV-describable polyhedra P' and P'', it becomes possible to tetrahedralize P without using a single Steiner point. Understandably, in many cases though Steiner points have to be introduced in order for the parts from the partitioning to be ASV-describable, e.g., the polyhedron P in Fig. 1. How to design an automatic partitioning algorithm or a heuristic that keeps the number of Steiner points introduced as low as possible is yet to be investigated.

5. Conclusion

An algorithm for the tetrahedralization (without using Steiner points) of a special class of polyhedra in the three-dimensional space is presented. The algorithm is in two phases: the derivation of the alternating sum of volume series of an ASV-describable polyhedron, and the tetrahedralization of the conjuncts in the ASV series. It is in essence a

[1] An edge is said to be reflex if the (interior) dihedral angle formed by its two incident faces exceeds π.

divide-and-conquer technique where a solid is first decomposed into more easily tetrahedralizable objects, namely conjuncts, and regenerated by merging the individually tetrahedralized conjuncts. However, when implementing the algorithm, in order to ensure achieving a valid tetrahedralization after merging the conjuncts, the triangulation of the shared faces of the adjacent conjuncts need to be identical for both tetrahedralization. This can be resolved by imposing an arbitrary triangulation on the shared faces, and then performing the tetrahedralization of the two conjuncts accordingly.

The proposed algorithm can be implemented as a module in the CAD or CAE systems that provide the function for generating tetrahedral mesh. The details of the two parts of the data structure—the representation of conjuncts and the representation of the resulting tetrahedral mesh—employed in the paper can be found in the literature [5,10,11,13,15,23,30]. The only special characteristic of these representations is that they are non-manifold representations. Due to the need for applications such as automatic NC tool path generation and rapid prototyping, most of the CAD/CAM systems already have the ability to deal with non-manifold representations.

The other main contribution of this paper is the theory for detecting ASV-describability; it is shown that the ASV-describability of a pseudo-polyhedron can be uniquely determined by applying the HULL_DIFF algorithm on the pseudo-polyhedron. It is also provided in Appendix A that if the number of tetrahedra from the tetrahedralization of a conjunct is linearly bounded by the number of the vertices of the conjuncts, then an ASV-describable pseudo-polyhedron P can be tetrahedralized into a number of tetrahedra linear to the size of P. When coupled with other known decomposition techniques, such as the unhindered vertex cup-removal method by Chazelle and Palios and Bajaj and the notch elimination scheme by Dey, the proposed algorithm also provides an attractive approach for the general decomposition using a small number of Steiner points.

Appendix A. A theorem on space linearity

The total storage required for an ASV series $P = H_1 - H_2 + H_3 - H_4 + \cdots + H_{n-1} - H_n$ is linear to the simple summation of the numbers of vertices of the layers H_is. It is not surprising that this space may be quadratic to the size of P. An example of the polyhedron that results in such a worst-case behavior is shown in Fig. 16. When applying the HULL_DIFF algorithm on P, k layers H_i $(i \le k)$ are formed, each of which includes $\{u_1, u_2, \ldots, u_k\}$ as its vertices, resulting in a tight $O(k^2)$ space requirement. A question regarding to its effect on the number of tetrahedra of P arises naturally. That is, does this $O(k^2)$ bound also dictate the same upper bound for the number of tetrahedra in P? This section resolves this concern by providing a space linearity theorem. Specifically, we are to prove that if the number of tetrahedra in every conjunct K_i, is bounded by the number of vertices on its outer and inner layers, the total number of tetrahedra in P remains bounded by the number of faces on P even though the total number of the vertices on the H_is may be quadratically the size of P. Notice that the number of faces is used to designate the size of P rather than the number of vertices; this is because P is a pseudo-polyhedron whose space requirement is dominated by the number of its faces [30,34]. Before establishing the theorem, two lemmas are first introduced.

Lemma 2. *Let v be a vertex on the conjunct K_i. Among all the faces of K_i incident at v, there exists at least one face, referred to as a realizing face of v, that is also a face of P.*

Proof. Because the conjunct K_i is a subset of P, every face of K_i incident at v must be either in the

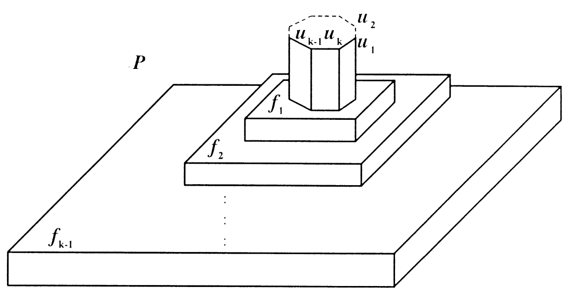


Fig. 16. A polyhedron whose ASV series requires $O(k^2)$ space.

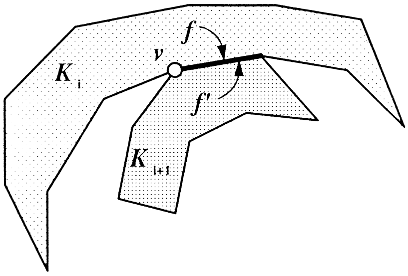


Fig. 17. Proof of Lemma 3.

interior or on the boundary of P. Because v is also a vertex of P, at least one of the faces of K_i incident at v must be on the boundary of P, for otherwise v would be an interior point of P. Q.E.D.

Lemma 3. *Let v be a vertex on both conjuncts K_i and K_{i+1}, and f and f' be the realizing faces of v on K_i and K_{i+1}, respectively. Then, f and f' are not be the same face on P.*

Proof. Because of the volume monotonicity, v can lie only on the inner layer of conjunct K_i and the outer layer of conjunct K_{i+1}. As a result, if f and f' are the same face, their outward normals must be opposite to each other, as illustrated in Fig. 17. This however means that f and f' are interior to P, a conflict to the assumption that they are both realizing faces. Q.E.D.

With the above two lemmas, it is now ready to establish the space linearity theorem.

Theorem 2. *Let the number of tetrahedra in conjunct K_i be denoted as $\sigma(K_i)$. Since $\sigma(K_i)$ is linear to the number of vertices on the outer and inner layers of K_i, for $i = 1, 2, \ldots, n/2$, the total number of tetrahedra $\sum_{i=1,2,\ldots,n/2} \sigma(K_i)$ is linear to the number of faces on P.*

Proof. By the assumption of linearity, it is clearly seen that the summation $\sum_{i=1,2,\ldots,n/2} \sigma(K_i)$ *is bounded by* $O(\sum_{v \in P} \lambda(v))$, where v is any vertex on P and $\lambda(v)$ represents the number of the conjuncts that have v as their vertex. By Lemma 2 and Lemma 3, we know $\lambda(v)$ is no greater than the number of faces on P incident at v, which is bounded by the degree of v, $d(v)$. Thus, $\sum\sigma(K_i)$ is bounded by $O(\sum d(v))$. It can be shown c20 that for a pseudo-polyhedron P, the term $O(\sum d(v))$ is linearly to the number of its faces. That is, $\sum\sigma(K_i)$ is bounded by the number of faces on P. Q.E.D.

The space linearity theory of Theorem 2 assures that the space requirement of the tetrahedralization of an ASV-describable pseudo-polyhedron P is solely determined by that of the individual conjuncts, independent to the underlying ASV series. This enables one to select from many ASV series of P without increasing the number of resulting tetrahedra significantly. In addition, the theorem implies that the running space for tetrahedralizing an ASV-describable polyhedron might be significantly smaller than the space requirement for the ASV series. This is particularly true if the ASV series is derived by the HULL_DIFF algorithm. During the recursion of the procedure HULL_DIFF(P), a conjunct K_i is immediately tetrahedralized once its two layers H_{2i-1} and H_{2i} are at hand, and thus releases the space for the next pair of layers H_{2i+1} and H_{2i+2} to use.

References

[1] A.V. Aho, J.E. Hopcroft, J.D. Ulman, The Design and Analysis of Computer Algorithms, Addison-Wesley, Reading, MA, 1974.

[2] T. Asano, T. Asano, R.Y. Pinter, Polygon triangulation: efficiency and minimality, Journal of Algorithms 7 (2) (1986) 221–231.

[3] C.L. Bajaj, T.K. Dey, Convex decomposition of polyhedral and robustness, SIAM J. Comput. 21 (2) (1992) 339–364.

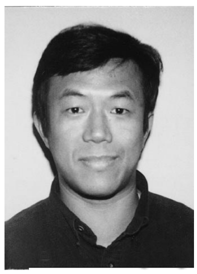
[4] B. Bollobas, Graph Theory, An Introductory Course, Springer, New York, 1979.

[5] B. Chazelle, Triangulating a nonconvex polytope, Discrete and Comput. Geom. 5 (1990) 505–526.

[6] B. Chazelle, Triangulating a simple polygon in linear time, Discrete and Comput. Geom. 6 (5) (1991) 485–524.

[7] J.L. Coulomb, F.X. Zgainski, Y. Marechal, Paramidal element to link hexahedral, prismatic and tetrahedral edge finite elements, IEEE Transactions on Magnetics 33 (2) (1997) 1362–1365, Pt. 2.

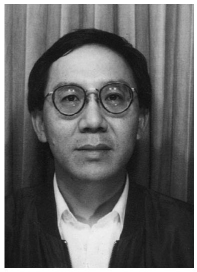
[8] D.P. Dobkin, M.J. Laszlo, Primitives for the manipulation of three-dimensional subdivisions, Algorithmica 4 (1) (1989) 3–32.
[9] K. Forsman, L. Kettunen, Tetrahedral mesh generation in convex primitives by maximizing solid angles, IEEE Transactions on Magnetics 30 (5) (1994) 3535–3538, Pt. 2.
[10] L.A. Freitag, C. Ollivier-Gooch, Tetrahedral mesh improvement using swapping and smoothing, Int. J. Numer. Methods Engineering 40 (21) (1997) 3979–4002.
[11] M.R. Garey, D.S. Johnson, F.P. Preparata, R.E. Tarjan, Triangulating a simple polygon, Inf. Proc. Letter 7 (4) (1978) 175–179.
[12] N.A. Golias, R.W. Dutton, Delaunay triangulation and 3D adaptive mesh generation, Finite Elements in Analysis and Design 25 (3–4) (1997) Pt. 2.
[13] L.J. Guibas, J. Stolfi, Primitives for the manipulating of general subdivisions and the computation of Voronoi diagrams, ACM Trans. Graphics 4 (2) (1985) 74–123.
[14] H.N. Gursoy, Tetrahedral finite element mesh generation from NURBS solid models, Engineering with Computers 12 (3–4) (1996) 211–223.
[15] Y.S. Kim, D.J. Wilde, A convergent decomposition using convex hulls and local cause of its non-convergence, ASME Journal of Mechanical Design 114 (1992) 459–467.
[16] Y.S. Kim, D.J. Wilde, A convergent convex decomposition of polyhedral objects, ASME Journal of Mechanical Design 114 (1992) 468–476.
[17] Y.S. Kim, Recognition of form features using convex decomposition, Computer-Aided Design 24 (9) (1992) 461–476.
[18] D.G. Kirkpatrick, M.M. Klawer, R.E. Tarjan, Polygon triangulation in O($n$log log$n$) time with simple data structures, Discrete and Comput. Geom. 7 (4) (1992) 329–346.
[19] D.G. Kirkpatrick, Optimal search in planar subdivisions, SIAM J. Comput. 12 (1) (1983) 28–35.
[20] C.K. Lee, S.H. Lo, Automatic adaptive 3-D finite element refinement using different-order tetrahedral elements, Int. J. Numer. Methods Engineering 40 (12) (1997) 2195–2226.
[21] S. Mohan, J.F. Lee, Three-dimensional finite element mesh generation using Delaunay triangulation, IEEE 1993 International Symposium on Digest Antennas and Propagation 1 (1993) 312–315.
[22] O.R. Musin, Properties of the Delaunay triangulation, Proc. 1997 Annual Symposium on Computational Geometry (1997) 424–426.
[23] J. O'Rourke, Art Gallery Theorems and Algorithms, Oxford University Press, New York, 1987.
[24] F.P. Preparata, M.L. Shamos, Computational Geometry: An Introduction, Springer, New York, 1985.
[25] A.A.G. Requicha, Representation for rigid solids: theory, methods and systems, ACM Comput. Survey 12 (4) (1980) 437–464.
[26] J. Ruppert, R. Seidel, On the difficulty of tetrahedralizing 3-dimensional nonconvex polyhedra, Proc. 5th Annual ACM Symp. on Comput. Geom. (1989) 380–392.
[27] E. Schönhardt, Über die zerlegung von dreieckspolyedern in tetraeder, Math. Ann. 98 (1928) .
[28] L.L. Schumaker, Triangulations in CAGD, IEEE Computer Graphics and Applications 13 (1) (1993) 47–52.
[29] L.D. Seneviratne, W.S. Ko, S.W.E. Earles, Triangulation-based path planning for a mobile robot, Journal of Mechanical Engineering Science 211 (5) (1997) 365–371.
[30] K. Tang, T. Woo, Algorithmic aspects of alternating sum of volumes: Part 1. Data structure and difference operation, Computer-Aided Design 23 (5) (1991) 357–366.
[31] K. Tang, T. Woo, Algorithmic aspects of alternating sum of volumes: Part 2. Nonconvergence and its remedy, Computer-Aided Design 23 (6) (1991) 435–443.
[32] S. Tor, A. Middleditch, Convex decomposition of simple polygons, ACM Transactions on Graphics 3 (4) (1984) 244–265.
[33] G.M. Turkiyyah, D.W. Storti, M. Ganter, H. Chen, Accelerated triangulation method for computing the skeletons of free-form solid models, Computer Aided Design 29 (1) (1997) 5–19.
[34] K. Weiler, Topological Structures for Geometric Modeling, PhD thesis, Rensselaer Polytechnic Institute, USA, 1986.
[35] T. Woo, Feature extraction by volume decomposition, Proc. Conf CAD/CAM Technology in Mechanical Engineering, Cambridge, MA, 1982.
[36] T. Woo, A Combinatorial analysis of boundary data structure schemata, IEEE Comput. Graph. Applic. 5 (3) (1985) 19–27.
[37] J.R. Woodwark, Some speculation on feature recognition, in: D. Kapur, J. Mundy (Eds.), Geometry Reasoning, The MIT Press, Cambridge, MA, 1989.
[38] H. Yamashita, A. Yamaji, V. Cingoski, K. Kaneda, A novel tetrahedral mesh generation method for rotating machines including end-coil region, IEEE Transactions on Magnetics 32 (3) (1996) 1353–1356, pt. 1.



Dr. Kai Tang is a senior programmer at Applicon Inc., Ann Arbor, Michigan, U. S. A. He received his Ph.D. from the Department of Electrical Engineering and Computer Science at the University of Michigan, in 1991. His research interests are CAD/CAM and computational geometry.

Shuo-Yan Chou is an associate professor of industrial management. His research interests include the application of geometric algorithms in CAD/CAM, data visualization, image processing, and product data management. He received his BBA in industrial management from National Cheng-Kung University, Taiwan, in 1983, and his M.S. and Ph.D. in industrial and operations engineering from the University of Michigan, USA, in 1987 and 1992.

Lin-Lin Chen is a professor of industrial and commercial design. Her research interests are in design for manufacturing, solid modeling, human-computer interface design, and WWW applications. She received her BS in industrial design from National Cheng-Kung University, Taiwan, in 1984, and her Ph.D. in industrial and operations engineering from the University of Michigan, USA, in 1992. She worked as a design engineer at Sampo Corporation, Taiwan, from 1984 to 1986, and as a part-time programmer in the computer-aided engineering department of the Ford Motor Company from 1987 to 1989.



Tony C. Woo is a Professor of Industrial Engineering, Adjunct Professor of Mechanical Engineering, and holder of the John M. Fluke Distinguished Chair in Manufacturing Engineering at the University of Washington. He previously held positions at the University of Michigan and the National Science Foundation. His research and teaching interests are in computational geometry for design and manufacturing. He received his education in Electrical Engineering at the University of Illinois.