

**ALGORITHMIC ASPECTS OF ALTERNATING  
SUM OF VOLUMES, PART I.  
DATA STRUCTURE AND DIFFERENCE OPERATION**

**Kai Tang  
Tony Woo**

**Department of  
Industrial & Operations Engineering  
University of Michigan  
Ann Arbor, MI 48109-2117**

**Technical Report 90-29**

**October 1990**

# **ALGORITHMIC ASPECTS OF ALTERNATING SUM OF VOLUMES**

## **Part I.**

### **Data Structure and Difference Operation**

**Kai Tang  
Tony Woo**

**University of Michigan**

**\* Send correspondence to: Tony Woo  
Dept. of Industrial & Operations Engineering  
University of Michigan  
Ann Arbor, MI 48109-2117  
U.S.A.**

---

Accepted by Journal of Computer Aided Design, to appear in April 1991.

## Abstract

From the point of view of basic theory, a unique conversion from a boundary representation to a CSG representation is of importance. From the point of view of application, the extraction of features by convex decomposition is of interest. The Alternating Sum of Volumes (ASV) is a technique that offers both. However, some algorithmic issues are still unresolved. This paper is the first of a two-part paper that addresses specialized set operations and the convergence of the ASV process. In this part, a fast difference operation for the ASV process and a data structure for pseudo polyhedra are introduced.

A fast difference operation between an object and its convex hull is enabled by the crucial observation that it only takes linear time to distinguish them. However, it takes  $O(N \log N)$  time to construct a data structure with the proper tags. The data structure supporting the operation is a pseudo polyhedron, capturing the special relation between an object and its convex hull. That the data structure is linear in space is also shown.

## 1. INTRODUCTION

The idea of the Alternating Sum of Volumes (ASV) is to represent an object by a series of convex components with alternating signs (for volume addition and subtraction). It is a technique to extract "features" from the boundary representation of a three-dimensional component [12]. As an example, consider the object in Figure 1. It shows a block with a slot and a hole. The ASV series of this object is

$$H_0 - H_1 + H_2 - H_3$$

where the  $H_i$ 's are convex.

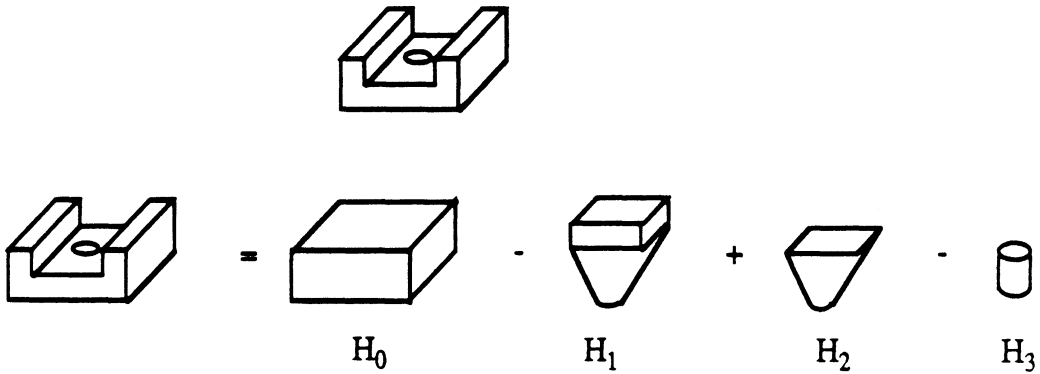


Figure 1. Alternating Sum of Volumes

Formally, the ASV series of an object  $\Omega$  is defined as

$$\Omega_0 = \sum (-1)^i H_i$$

where  $\Omega_0 = \Omega$

$H_i$  is the convex hull of  $\Omega_i$ ,  $CH(\Omega_i)$

$\Omega_i$  is the deficiency and is defined as the regularized set difference between  $H_{i-1}$  and  $\Omega_{i-1}$ .

Figure 2 shows how the terms in an ASV are derived for the object in Figure 1. The deficiency  $\Omega_i$  is obtained by subtracting  $\Omega_{i-1}$  from  $H_{i-1}$ ,  $i=1,2,\dots$ . As  $\Omega_{i+1}$  becomes the null set  $\emptyset$ , the  $H_i$ 's are collected starting with  $H_0$ . The ASV expression is formed by alternating "-" and "+" signs as in  $H_0-H_1+H_2-\dots$ .

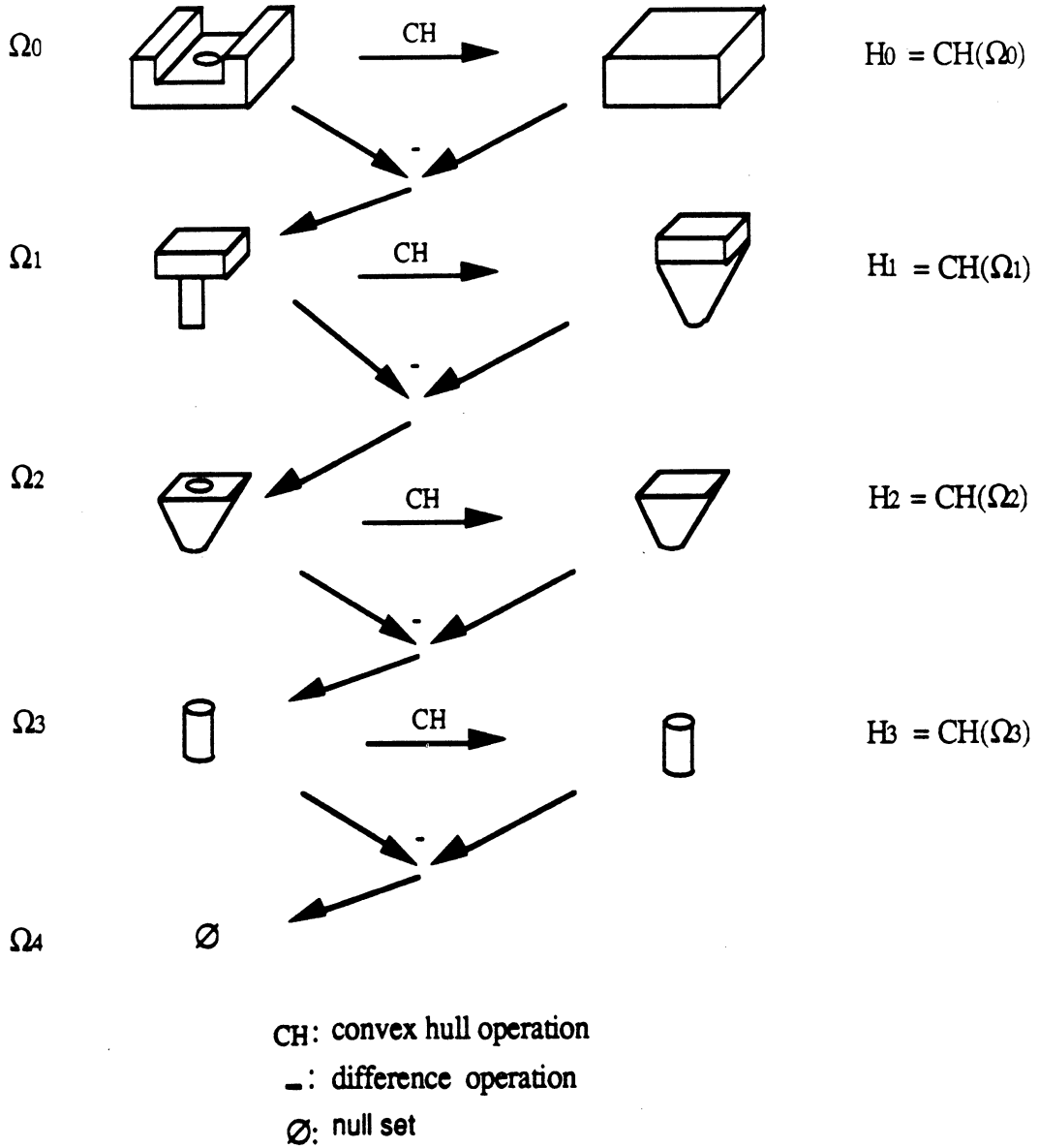


Figure 2. Derivation of ASV series

Consider the machining process of the mechanical component in the previous example. There are two features: a slot and a hole. They can be extracted by algebraic manipulation of its

ASV as illustrated in Figure 3. Parenthesizing  $H_1$  and  $H_2$  forces a change in the sign from + to -. Subtracting  $H_2$  from  $H_1$  yields a new  $H_1'$  for a disjunctive expression:

$$H_0 - (H_1 - H_2) - H_3 = H_0' - H_1' - H_2' .$$

If  $H_0'$  is the stock,  $H_1'$  and  $H_2'$  can be thought of as volumes to be removed to create the slot and the hole.

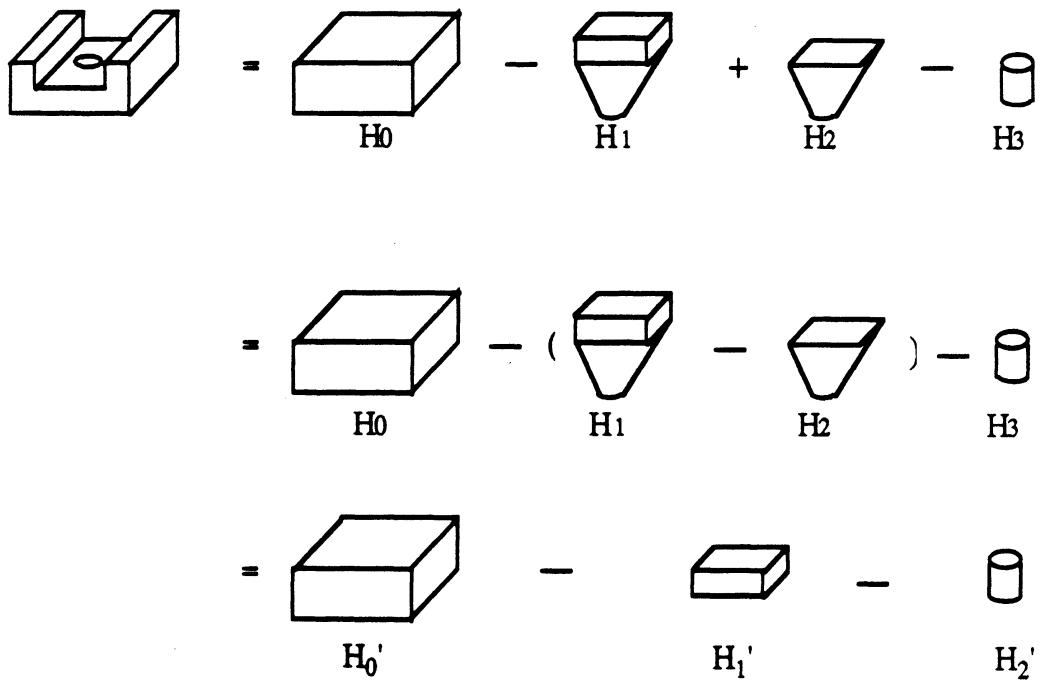


Figure 3. Algebraic Manipulation of ASV Series into Disjunctive Form

As another illustration of the material joining process, consider Figure 4. The components adjacent to the - sign are parenthesized yielding a conjunctive expression:

$$H_0 - H_1 + H_2 = (H_0 - H_1) + H_2 = H_0' + H_1' .$$

Here,  $H_0'$  is the base plate on which a protrusion  $H_1'$  is to be joined. These two examples illustrate that through the manipulation of an ASV series, features of a given object can be

extracted automatically, which in turn can help the process planners in deciding the suitable manufacturing operations such as machining or welding. The ability to "disassemble" allows conversion from boundary based solid modeling systems to those that are CSG-based [8].

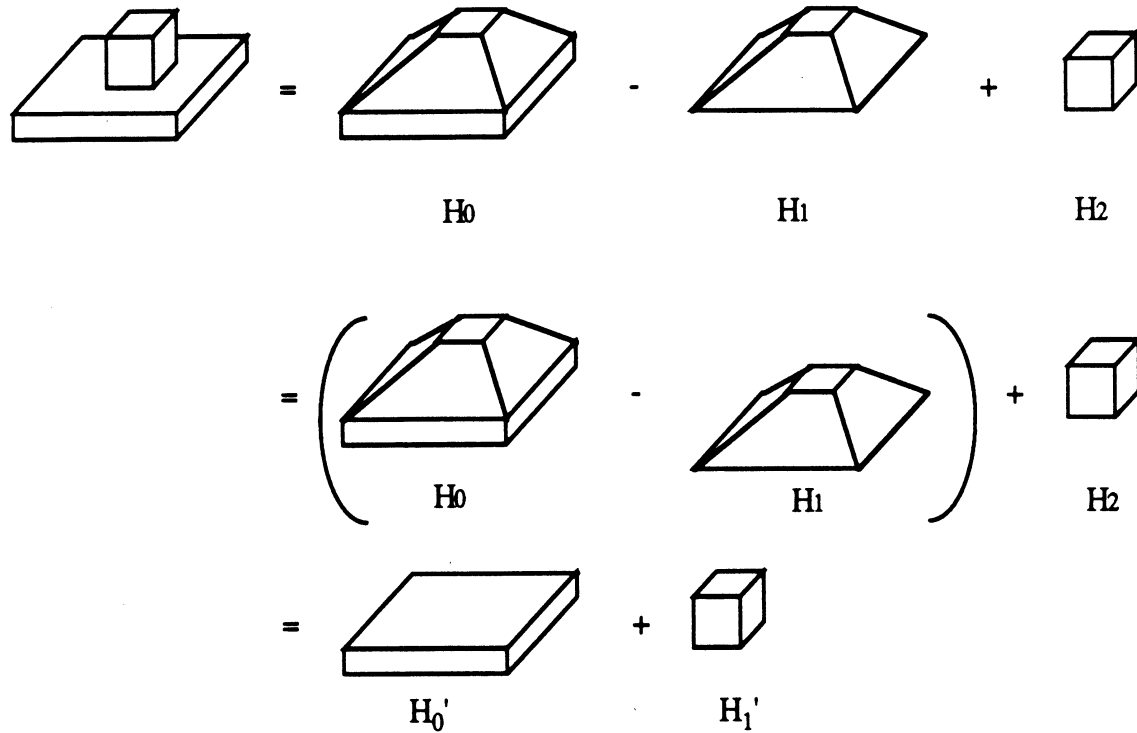


Figure 4. Algebraic Manipulation of ASV Series into Conjunctive Form

As implied in the examples, the terminating condition of an ASV series expansion is when the deficiency  $\Omega_n$  becomes convex; that is, when  $H_n$  identifies with  $\Omega_n$ . This condition, however, is not guaranteed. Figure 5 illustrates an example of an infinite ASV series. It has been shown [12] that an ASV series is non-terminating if and only if there is an integer  $i$  such that  $H_{i+1} = H_i$ . In such a case, the deficiency  $\Omega_i$  is said to be *non-convergent*. When a non-convergent deficiency  $\Omega_i$  is encountered, the ASV expansion can not continue.

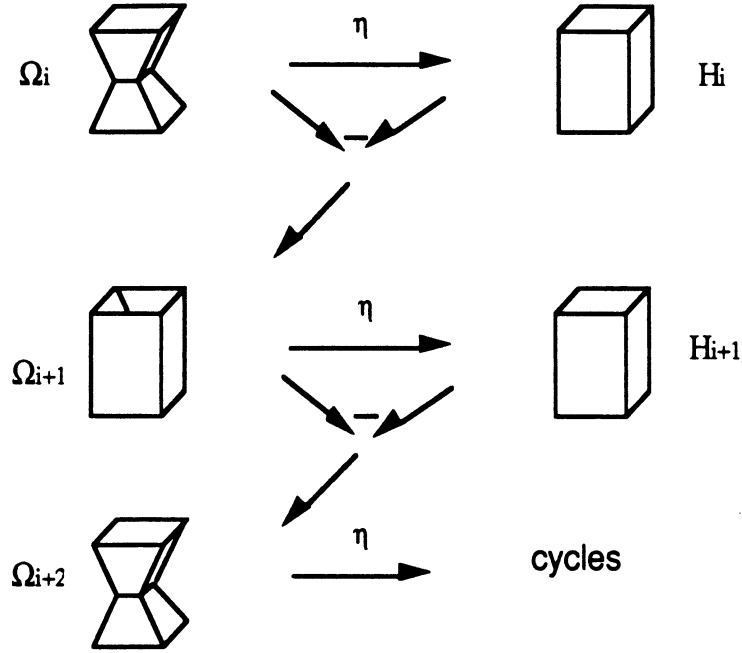


Figure 5. An example of ASV non-convergence

When a deficiency  $\Omega_n$  becomes non-convergent, one solution is to divide it into convex subsets [12]. But there is a drawback. It is known [4] that there can be  $O(n^3)$  convex subsets, where  $n$  is the number of concave edges, each subset requires further polynomial time to determine. An alternative is to decompose it into subsets which are themselves convergent so that the ASV series can expand further. For example, the object  $\Omega$  in Figure 6 is non-convergent. By separating it along the edge "e" into two parts  $P_1$  and  $P_2$ , and performing the ASV expansion on each of them, a finite ASV series of two branches, each of which is a finite ASV series, results. The observation that the type of edges such as "e" in Figure 6 may be a very small subset of the set of all concave edges encourages inquiry.



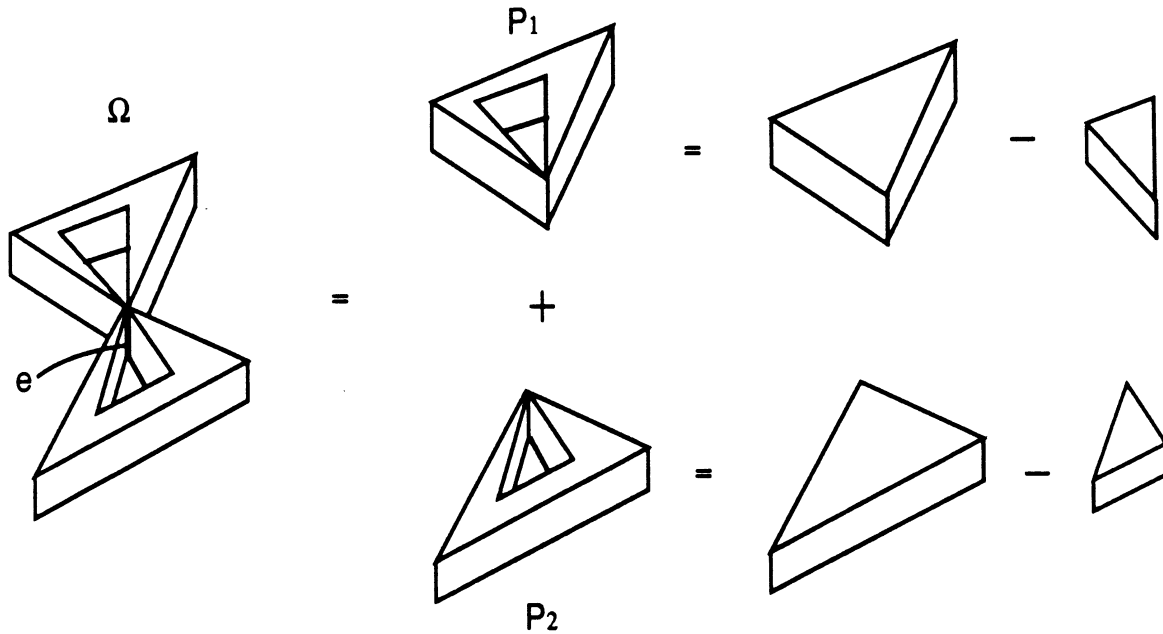


Figure 6. Remedy for non-convergence

This is the first of a two-part paper. Part I deals with a special difference operation. While general difference operation [10] may be invoked, the relation between an object and its convex hull merits investigation. This relation is made concrete by the notion of a *pseudo manifold*, as illustrated by  $\Omega$  in Figure 6. It is an entity that is between a two-manifold [2] and a non-manifold [11]. With the aid of a data structure for pseudo manifolds, an  $O(n \log n)$  time algorithm for computing a convex deficiency is made possible, where  $n$  is the number of faces of the pseudo manifold. The data structure is shown to be  $O(n)$  in space which gives an absolute upper bound to the number of edges of the type "e" in Figure 6.

The characterization of non-convergence and its remedy as illustrated by Figure 6 is given in the sequel, Part II of the paper.

## 2. DOMAIN AND DATA STRUCTURE

In this section the domain of objects and a data structure to represent them are given. An object  $\Omega$  is a set of points in three-dimensional Euclidean space  $E^3$ . It must satisfy certain restrictions. Because ASV performs operations on the boundary of volumes, each object must be a closed surface that forms the closure of an open set of finite extent in  $E^3$ . In other words, an object must be the surface of a volume and must not have "dangling" faces and edges [8]. In addition to this restriction of *homogenous three-dimensionality*, the objects must also be closed under the (regularized) difference operation, i.e., they should have *differential preservability* [8]. To define a domain of objects that will meet both the restrictions, some definitions regarding the interior and boundary points of a three-dimensional point set need to be clarified.

**Definition 1.** A point  $p$  of a set  $S$  in  $E^3$  is called an *interior* point of  $S$  if there exists an open three-dimensional neighborhood that consists of points in  $S$  only. A point  $p$  is called a *boundary* point of  $S$  if it is not an interior point. The set  $B(S)$  of all boundary points of  $S$  and the set  $I(S)$  of all the interior points of  $S$  are defined as the *boundary* and the *interior* of  $S$  respectively.

The relation between a boundary point and its neighboring points of a set  $S$  in  $E^3$  is described by three characterizations, namely, *manifold*, *pseudo manifold*, and *non-manifold*. A point  $p$  in  $B(S)$  is called a *two-manifold* point if it has a three-dimensional neighborhood such that the subset of the points of  $S$  contained in that neighborhood is topologically equivalent to a hemisphere [11]. A point  $p$  is a *pseudo manifold* point if every three-dimensional neighborhood of it contains some points in  $I(S)$ . If a point  $p$  has a three-dimensional neighborhood such that the subset of the points of  $S$  contained in that neighborhood entirely belong to  $B(S)$ , then it is called a *non-manifold point*. As an example, the boundary surface of the object in Figure 7 consists of the six faces of the cube and that dangling face "f". All the boundary points except "f" (including edge "e") are two-manifold points. The boundary points on the six faces of the cube, including edge "e", are pseudo manifold points. The non-manifold points are those on face "f" but not on edge "e".

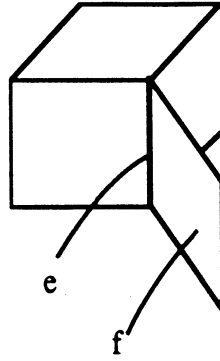


Figure 7. Points characterizing a Manifold, a Pseudo-manifold, and a Non-manifold

**Definition 2.** A point set  $S$  in  $E^3$  is a *two-manifold* if  $I(S)$  is connected and every point in  $B(S)$  is a two-manifold point.  $S$  is a *pseudo manifold* if every point in  $B(S)$  is a pseudo manifold point.  $S$  becomes a *non-manifold* if  $B(S)$  contains some non-manifold points.

A pseudo manifold point is a relaxation of two-manifold, i.e., it only requires that every neighborhood of the point contains some interior points of the set, but with no topological constraint on the neighborhoods. A pseudo manifold need not be a connected set either. The relation of these three sets can be best described by Figure 8. Because an object must have homogenous three-dimensionality, non-manifolds are immediately excluded from the consideration. Although two-manifolds satisfy the homogenous three-dimensionality, they are not closed under regularized difference operation [8]. The generality of pseudo manifolds, while still conforming to homogenous three dimensionality but also guarantees differential preservability [8], prove to be the only suitable clan of objects suitable for ASV representations. Figure 9 illustrates several examples of two-manifolds, pseudo manifolds and non-manifolds.

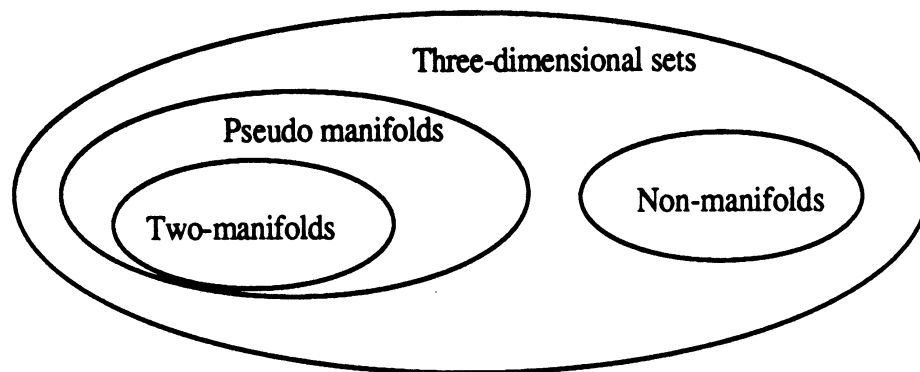


Figure 8. Set relation between Two-manifolds, Pseudo manifolds and Non-manifolds

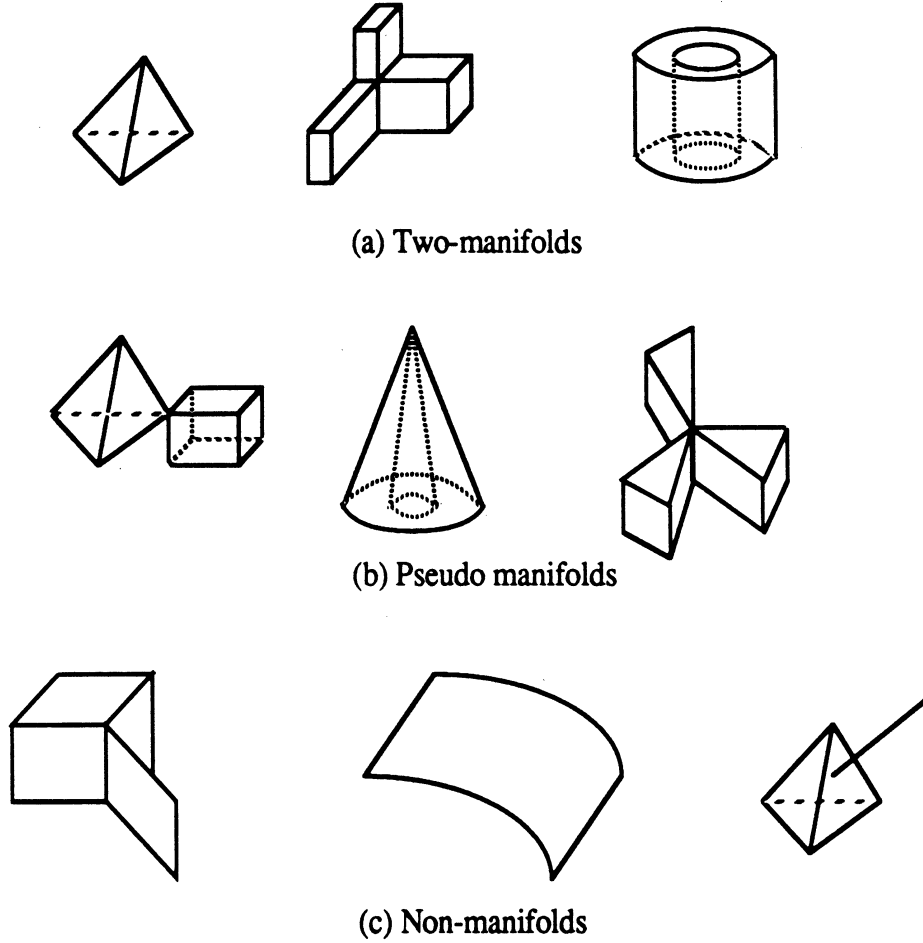


Figure 9. Illustration of Two-manifolds, Pseudo manifolds and Non-manifolds

A data structure for the pseudo manifolds is crucial to both the development and analysis of algorithms. A data structure for polyhedra [13] is not suitable because a pseudo manifold could have more than two faces meeting at an edge (see Figure 9b). The representation of general non-manifolds [11] are more than what is needed here, because of the difference preservability of pseudo manifolds. The concept of *pseudo polyhedra* is proposed. A pseudo polyhedron is "almost" a polyhedron except that it allows edges to have more than two adjacent faces.

**Definition 3.** A *pseudo polyhedron*  $P$  is a finite collection of planar faces such that (a) every edge of  $P$  has at least two adjacent faces, and (b) if any two faces meet they meet at a common edge. Specifically, a pseudo polyhedron with  $n_v$  vertices,  $n_e$  edges and  $n_f$  faces is a quintuple  $\langle V, E, F, \text{NORM}, E_f \rangle$  which is defined as:

(a)  $V = \{v_1, v_2, \dots, v_{nv}\}$  is a list storing the  $n_v$  vertices; each  $v_i$  is a coordinate triple  $(x_i, y_i, z_i)$ .

(b)  $E = \{ \langle v_{1,1}, v_{1,2} \rangle, \langle v_{2,1}, v_{2,2} \rangle, \dots, \langle v_{ne,1}, v_{ne,2} \rangle \}$  is the edge list. Each entry  $\langle v_{i,1}, v_{i,2} \rangle$  stands for an edge with  $v_{i,1}$  and  $v_{i,2}$  being indices of the two end points; e.g.,  $\langle v_{i,1}, v_{i,2} \rangle = \langle 3, 10 \rangle$  means the end points of the  $i$ th edge are  $v_3$  and  $v_{10}$ .

(c)  $F = \{F_1, F_2, \dots, F_{nf}\}$  stores face information. Each element  $F_i$  is itself an array of the form  $\{ \langle e_{1,1}, e_{1,2}, \dots, e_{1,f1} \rangle, \langle e_{2,1}, e_{2,2}, \dots, e_{2,f2} \rangle, \dots, \langle e_{k,1}, e_{k,2}, \dots, e_{k,fk} \rangle \}$ , where  $k$  is the total number of polygons in face  $F_i$ . Each  $\langle e_{j,1}, e_{j,2}, \dots, e_{j,fj} \rangle$  is a simple polygon, and each  $e_{j,l}$  is the index of an edge in  $E$ . For example, by  $F_i = \{ \langle 2, 4, 1 \rangle, \langle 5, 7, 6, 3 \rangle \}$ , we mean face  $F_i$  is bounded by two simple polygons; the indices of the edges of the outer polygon are 2, 4, 1 and are 5, 7, 6, 3 for the inner polygon. The edges are ordered clockwise (for outer) or counter-clockwise (for inner).

(d)  $NORM = \{ N_1, N_2, \dots, N_{nf} \}$  stores the outward normals of the  $n_f$  faces.

(e)  $E_f = \{ \langle f_{1,1}, f_{1,2}, \dots, f_{1,k1} \rangle, \langle f_{2,1}, f_{2,2}, \dots, f_{2,k2} \rangle, \dots, \langle f_{ne,1}, f_{ne,2}, \dots, f_{ne,kne} \rangle \}$  is a list which describes the edge-face adjacency relation. An entry  $\langle f_{i,1}, f_{i,2}, \dots, f_{i,ki} \rangle$  says that edge  $E_i$  has  $k_i$  adjacent faces and the indices for them are  $f_{i,1}, f_{i,2}, \dots, f_{i,ki}$ . For example, if  $\langle f_{i,1}, f_{i,2}, \dots, f_{i,ki} \rangle$  is  $\langle 1, 7, 6, 2 \rangle$ , then edge  $E_i$  has four adjacent faces and their indices in  $F$  are 1, 7, 6 and 2. Each  $k_i$  is defined as the *face adjacency index* of edge  $E_i$ .

The data structure given above, although quite simple, completely describes one family of pseudo manifolds: planar pseudo manifolds. (Pseudo manifolds with non-planar boundary surfaces are not considered in this paper.) In the Appendix a detailed proof that the space requirement of a pseudo polyhedron is linear in the number of its faces is given. It should be noted that though a pseudo polyhedron completely describes the boundary of a pseudo manifold, it itself carries no set theoretic information. It is the pseudo manifold, which a pseudo polyhedron *represents*, possesses the set in  $E^3$ .

### 3. DIFFERENCE OPERATION

To study the difference operation between a pseudo manifold and its convex hull, the following notations are used:  $\Omega$  denotes a pseudo manifold object,  $\Omega_h$  for its convex hull, and  $\Omega_d$  for the deficiency  $\Omega_h - \Omega$ . The same notations are used to represent their defining pseudo polyhedra unless noted otherwise.

Consider the convex hull  $\Omega_h$  of an object  $\Omega$ . The set  $\Omega_h$  can be divided into four disjoint subsets. They are,

$\xi_i$ :  $\{I(\Omega)\}$ , the interior points of  $\Omega$ ,

$\xi_h$ :  $\{B(\Omega_h)\}$ , the boundary (hull) points of  $\Omega_h$ ,

$\xi_p$ :  $\{B(\Omega) \cap I(\Omega_h)\}$ , the boundary points of  $\Omega$  excluding those which are also in  $\xi_h$ ,

$\xi_d$ :  $\{I(\Omega_h) - \Omega\}$ , interior (deficiency) points of  $\Omega_h$  excluding  $\Omega$ .

Figure 10 illustrates these subsets in two dimensions.

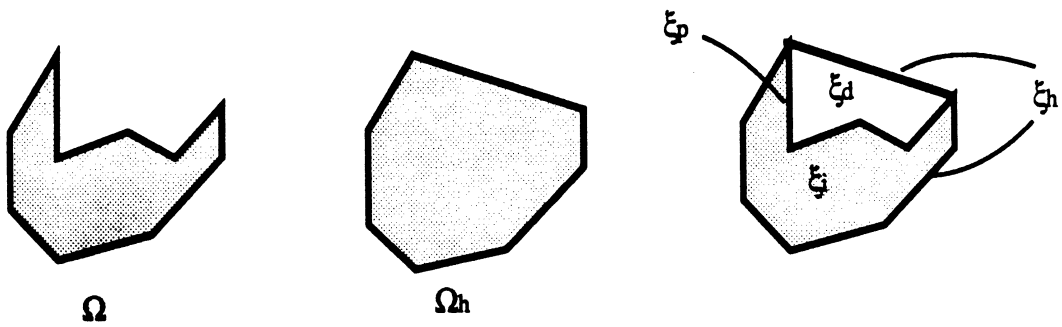


Figure 10. Illustrations of  $\xi_i$ ,  $\xi_h$ ,  $\xi_p$  and  $\xi_d$

**Definition 4.** A point  $p \in \Omega_h$  is a *preserved point* if for any real number  $\epsilon > 0$ , no matter how small it is, the open neighborhood sphere which centers at  $p$  with radius  $\epsilon$  always

has a point in  $\xi_d$ . A point  $p$  is a *lost point* if it is not a preserved point.

The difference operation between  $\Omega_h$  and  $\Omega$  is defined as follows.

**Definition 5.** The *deficiency*  $\Omega_d$  of an object  $\Omega$  is a set in  $E^3$  consisting of all the preserved points of  $\Omega_h$ .

The definition of deficiency here agrees with the intuition. The purpose of categorizing the points of  $\Omega_h$  is to relate the interior and the boundary of the deficiency  $\Omega_d$  with that of  $\Omega$  and its convex hull  $\Omega_h$ . It is easy to see that because both  $I(\Omega)$  and  $I(\Omega_h)$  are open sets, so must be the sets  $\xi_i$  and  $\xi_d$ . By definition of the deficiency, all the points of  $\xi_i$  are lost. Since  $\xi_d$  is an open set, every point in it has a neighborhood sphere of points in  $\xi_d$  only and thus is interior to  $\Omega_d$ . This means  $\xi_d$  must be a subset of the interior of deficiency  $\Omega_d$ . For any point in  $\xi_p$  or  $\xi_h$ , every neighborhood of it contains some points either in  $\xi_i$  or in  $\{E^3 - \Omega_h\}$ , that is, it contains points not belonging to  $\Omega_d$ , and hence by the definition they can not be the interior of  $\Omega_d$ . These observations are summarized in the following lemma.

**Lemma 1.** The deficiency  $\Omega_d$  of a pseudo manifold  $\Omega$  is also a pseudo manifold, whose interior set  $I(\Omega_d)$  is the  $\xi_d$ -set of  $\Omega_h$  and the boundary set  $B(\Omega_d)$  is a subset of  $\{\xi_h \cup \xi_p\}$ .

As noted by Lemma 1, the boundary surface  $B(\Omega_d)$  of the deficiency  $\Omega_d$  is a subset of  $\{\xi_h \cup \xi_p\}$  of  $\Omega_h$ . Because of the planarity of the faces,  $B(\Omega_d)$  must be a set of some faces of  $\Omega$  and  $\Omega_h$ . The key to the algorithm for finding the deficiency is to find these faces as well as the adjacency relation among them so that the result is a pseudo polyhedron representation of  $\Omega_d$ .

**Definition 6.** A face of a pseudo manifold  $\Omega$  is a *hull face* if it is in  $\xi_h$ ; otherwise it is called an *internal face*.

**Lemma 2.** A hull face  $f$  of  $\Omega$  will not exist in the boundary surface  $B(\Omega_d)$ .

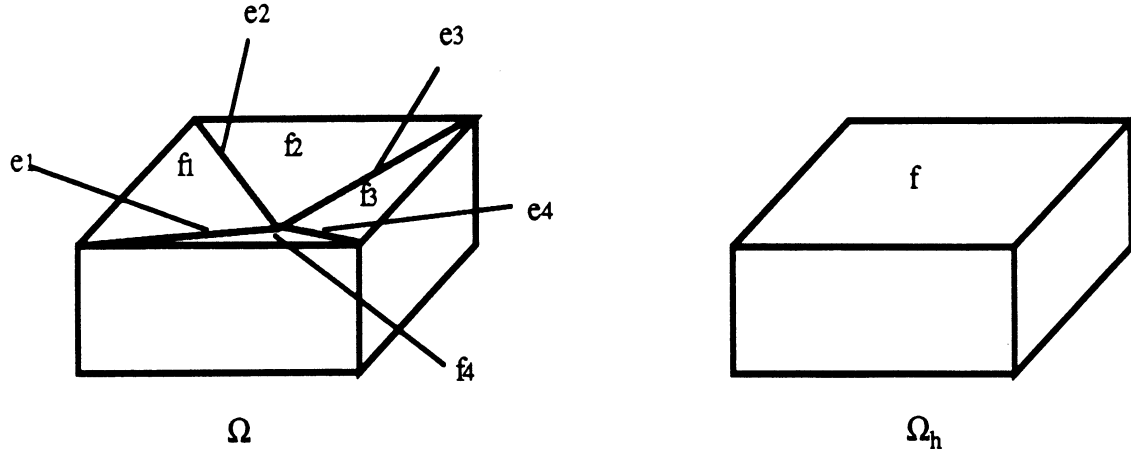
**Proof.** For any point in the interior set  $I(f)$  of  $f$ , say  $p$ , there must exist an open neighborhood  $\epsilon$  of  $p$  that belong to  $I(f)$  and hence in  $\xi_h$  only. Since  $\Omega$  is a pseudo manifold, every point in  $\epsilon$ , including  $p$  itself, must have a neighborhood sphere that consists of points in  $\{\xi_h \cup \xi_i \cup \{E^3 - \Omega_h\}\}$  only. By the way a preserved point is defined,  $p$  can only be a lost point. Therefore the entire set  $I(f)$  are lost points.

Q.E.D.

Lemma 2 asserts that the boundary surface  $B(\Omega_d)$  consists of only the internal faces of  $\Omega$  and the faces of  $\Omega_h$  but not the hull faces of  $\Omega$ . This observation leads to the development of the difference algorithm sought. It accepts as input  $(V, E, F, E_f, \text{NORM}_f)$  which is the pseudo polyhedron representation of a pseudo manifold  $\Omega$  and outputs the pseudo polyhedron representation of the deficiency  $\Omega_d$ . Suppose there is a procedure **HULLP** which takes as input the pseudo polyhedron  $(V, E, F, E_f, \text{NORM}_f)$  of a pseudo manifold  $\Omega$ . Its output are two: one is the pseudo polyhedron of the resultant convex hull  $\Omega_h$ , and the other is two arrays  $F_H$  and  $E_H$ , called *hull tag arrays* which distinguish those faces and edges of  $\Omega_h$  that do not belong to  $\Omega$ . Specifically,  $F_H(i)=1$  means face  $i$  of  $\Omega_h$  is a face of  $\Omega$ . When  $F_H(i)=0$ , the meaning is reversed.  $E_H(i)=j$  means edge  $i$  of  $\Omega_h$  is edge  $j$  of  $\Omega$ , whereas  $E_H(i)=0$  means edge  $i$  is not an edge of  $\Omega$ . Since most available three-dimensional convex hull algorithms[6,7] support data structures that embody our pseudo polyhedra, the feasibility of the output of **HULLP** is justified. For the convenience of computation it is also assumed that the vertex array  $V$  is unchanged through **HULLP**, although redundant vertices in the  $V$  array of  $\Omega_h$  are implied. Also, there are two additional arrays,  $F_I$  and  $E_I$ . They are the *Internality tag arrays* which identify internal faces and internal edges of  $\Omega$ . Specifically,  $F_I(i)=1$  means face  $i$  of  $\Omega$  is an internal face and, similarly,  $E_I(i)=1$  means edge  $i$  of  $\Omega$  is an internal edge. Likewise, when  $F_I(i)$  or  $E_I(i)$  is 0, the meaning is reversed. These two arrays are  $O(n \log n)$  derivable



from  $\Omega$  because the internality of any face  $f$  (or edge  $e$ ) can be identified by checking the internality of an arbitrary point of  $I(f)$  (or  $I(e)$ ). Figure 11 demonstrates the functionality of procedure **HULLP** on a pseudo polyhedron.



$F_H$ :  $F_H(f) = 0$ ; all the other  $F_H$ 's are 1.

$E_H$ : All the  $E_H$ 's are non-zero.

$F_I$ :  $F_I(f_1) = F_I(f_2) = F_I(f_3) = F_I(f_4) = 1$ ; all the other  $F_I$ 's are 0.

$E_I$ :  $E_I(e_1) = E_I(e_2) = E_I(e_3) = E_I(e_4) = 1$ ; all the other  $E_I$ 's are 0.

Figure 11. Functionality of procedure **HULLP**

The first algorithm **MERGE** to be given below adds those edges and faces of  $\Omega_h$  that do not belong to  $\Omega$  to the description arrays  $E$  and  $F$  of  $\Omega$  and updates  $E_f$  correspondingly. A constant time function named **INSERT\_ $E_f$** ( $E_f, i, j$ ) is used. It either sets  $E_f(i)$  to " $j$ " if  $E_f(i)$  is not previously defined, or appends " $j$ " to  $E_f(i)$ .

---

Procedure **MERGE** ( $n_v, n_e, n_f, V, E, F, E_f, NORM_f, F_I, E_I$ ,

$n'_v, n'_e, n'_f, V', E', F', E'_f, NORM'_f, F_H, E_H$ )

/\*purpose: Updates the pseudo polyhedron representation of a pseudo manifold  $\Omega$  by adding the newly generated hull faces and hull edges of its convex hull to it.

input: ( $n_v, n_e, n_f, V, E, F, E_f, NORM_f, F_I, E_I$ )-----pseudo polyhedron  $\Omega$ ,  $F_I$  and  $E_I$

are the *internality tag arrays* of its faces and edges.

$(n'_v, n'_e, n'_f, V', E', F', E'_f, \text{NORM}'_f, F_H, E_H)$ -----pseudo polyhedron representation of the convex hull  $\Omega_h$  of  $\Omega$ ,  $F_H$  and  $E_H$  are the *hull tag arrays* of its faces and edges.

output:  $(n_v, n_e, n_f, V, E, F, E_f, \text{NORM}_f, F_I, E_I)$ -----updated pseudo polyhedron of  $\Omega$  with the newly generated hull faces and hull edges of  $\Omega_h$  added,  $F_I$  and  $E_I$  are updated with the following convention:

|          |     |   |
|----------|-----|---|
| $F_I(i)$ | = 0 | face i is a hull face of $\Omega$                             |
|          | = 1 | face i is an internal face                                    |
|          | = 2 | face i is a face of $\Omega_h$ but not a face of $\Omega$     |
| $E_I(i)$ | = 0 | edge i is a hull edge of $\Omega$                             |
|          | = 1 | edge i is an internal edge                                    |
|          | = 2 | edge i is an edge of $\Omega_h$ but not an edge of $\Omega$ . |

\*/

begin

step 1.  $\text{ETOP} = n_e$

$\text{FTOP} = n_f$

for  $i=1, n'_e$  do

$\text{EMAP}(i) = 0$

end do {step 1}

step 2. for  $i=1, n'_f$  do

if  $F_H(i) = 0$  then

step 3.  $\text{FTOP} \leftarrow \text{FTOP} + 1$

step 4.  $(e_1, e_2, \dots, e_l) \leftarrow F'(i)$

step 5. for  $j=1, l$  do

if  $E_H(e_j) = 0$  then

if  $\text{EMAP}(e_j) = 0$  then

step 6.  $\text{ETOP} \leftarrow \text{ETOP} + 1$

step 7.  $\text{EMAP}(e_j) \leftarrow \text{ETOP}$

step 8.  $E(\text{ETOP}) \leftarrow E'(e_j)$

step 9.  $E_I(\text{ETOP}) \leftarrow 2$

end if

```

else
step 6'.      EMAP(ej) <-- EH(ej)
end if
step 10.      call INSERT_Ef (Ef, EMAP(ej), FTOP)
end do {step 5}
step 11.      F(FTOP) <-- (1, (EMAP(e1),EMAP(e2),...,EMAP(el)))
step 12.      NORMf (FTOP) <-- NORM'f(i)
step 13.      Fl(FTOP) <-- 2
end if
end do {step 2}

step 14.      nf = FTOP
              ne = ETOP

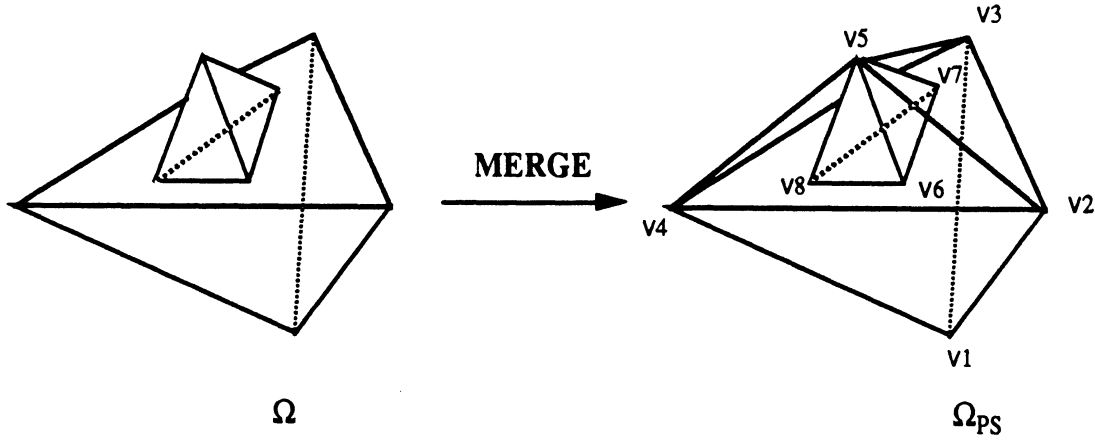
end {MERGE}

```

---

Comments on **MERGE**: Step 1 initializes two stack pointers F<sub>TOP</sub> and E<sub>TOP</sub> which stand for the numbers of current faces and edges in  $\Omega$ , respectively. Array EMAP is the index mapping between E' and E, e.g., EMAP(i) = j means edge i of  $\Omega_h$  is edge j of (current)  $\Omega$ . Steps 3 through 13 are performed once for each face of  $\Omega_h$  that is not a face of  $\Omega$  ( $F_H = 0$ ). For each edge of a selected face, whether it is also an edge of  $\Omega$  is first checked. If it is not (when its  $E_H = 0$ ) and it has not been previously added to  $\Omega$ , it is then added to E with its  $E_l$  set to 2 and its EMAP set to a unique number E<sub>TOP</sub> (see steps 6 through 9). Otherwise its EMAP is assigned with its  $E_H$  which is the index of this edge in the original  $\Omega$  (step 6'). At step 10 the face adjacency relation of this edge in  $\Omega$  is updated as reflected by the insertion of this selected face. Steps 11 and 12 append the selected face and its normal to F and NORM<sub>f</sub> of  $\Omega$ . (Note that, because  $\Omega_h$  is convex, every face of it has only one bounding polygon.) Step 13 assigns 2 to the F<sub>l</sub> of this face that indicates the added face is not a face of the original  $\Omega$ . To analyze the time requirement of **MERGE**, notice that each edge of  $\Omega_h$  has exactly two adjacent faces in  $\Omega_h$ . At most twice will an edge of  $\Omega_h$  be checked, retrieved and stored. Steps 12 and 13 take constant time. As the result, the loop from step 2 through step 13 are  $O(n'_e + n'_f)$ .

The output of **MERGE**, let it be called the *PS description* of  $\Omega$  and denoted as  $\Omega_{PS}$ , is a pseudo polyhedron. Figure 12 lists the V,E,F, $E_f$  entries of the PS description of a pseudo manifold.  $\Omega_{PS}$  itself, however, no longer represents a legitimate pseudo manifold, as it contains all the intermediate data for obtaining the deficiency  $\Omega_d$ . For the pseudo manifold  $\Omega$  in Figure 12, the boundary of its deficiency  $\Omega_d$  consists of the faces  $f_4, f_5, f_6, f_7, f_8, f_9, f_{10}$  as defined in the F entry of  $\Omega_{PS}$ ; the vertices of  $\Omega_d$  are  $v_2, v_3, v_4, v_5, v_6, v_7, v_8$  of  $\Omega_{PS}$ ; and the edges of  $\Omega_d$  are  $e_4, e_5, e_6, e_7, e_8, e_9, e_{10}, e_{11}, e_{12}, e_{13}, e_{14}, e_{15}$  of  $\Omega_{PS}$ . The procedure **DIFFBUILD** given next will generate  $\Omega_d$  utilizing  $\Omega_{PS}$ . A constant time routine called **INSERT(L,i)** will be used in the algorithm which appends an integer  $i$  into an integer list  $L$ .



$$V = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8\}$$

$$E = \{\langle v_1, v_2 \rangle, \langle v_1, v_3 \rangle, \langle v_1, v_4 \rangle, \langle v_5, v_6 \rangle, \langle v_5, v_7 \rangle, \langle v_5, v_8 \rangle, \langle v_5, v_2 \rangle, \langle v_5, v_3 \rangle, \langle v_5, v_4 \rangle, \langle v_2, v_3 \rangle, \langle v_3, v_4 \rangle, \langle v_4, v_2 \rangle, \langle v_6, v_7 \rangle, \langle v_7, v_8 \rangle, \langle v_8, v_6 \rangle\}$$

$$F = \{\langle 1, 10, 3 \rangle, \langle 1, 2, 11 \rangle, \langle 3, 12, 2 \rangle, \langle 10, 11, 12 \rangle, \langle 13, 4, 6 \rangle, \langle 5, 4, 14 \rangle, \langle 6, 5, 15 \rangle, \langle 10, 7, 9 \rangle, \langle 8, 7, 11 \rangle, \langle 9, 8, 12 \rangle\}$$

$$E_f = \{\langle 1, 2 \rangle, \langle 2, 3 \rangle, \langle 3, 1 \rangle, \langle 5, 6 \rangle, \langle 6, 7 \rangle, \langle 7, 5 \rangle, \langle 8, 9 \rangle, \langle 9, 10 \rangle, \langle 10, 8 \rangle, \langle 1, 4 \rangle, \langle 2, 4 \rangle, \langle 3, 4 \rangle\}$$

Figure 12. An example of the PS description  $\Omega_{PS}$

---

Procedure **DIFFBUILD** ( $n_v, n_e, n_f, V, E, F, E_f, \text{NORM}_f, F_I, E_I,$

$n_{dv}, n_{de}, n_{df}, V_d, E_d, F_d, E_{df}, \text{NORM}_{df}$  )

/\* purpose: Finds the deficiency  $\Omega_d$  of a pseudo manifold  $\Omega$  and outputs the pseudo polyhedron representation of  $\Omega_d$  to the external.

input: ( $n_v, n_e, n_f, V, E, F, E_f, \text{NORM}_f, F_I, E_I$  ) -----The P description of  $\Omega$ .

output: ( $n_{dv}, n_{de}, n_{df}, V_d, E_d, F_d, E_{df}, \text{NORM}_{df}$ )-----The pseudo polyhedron representation of deficiency  $\Omega_d$ .

\*/

begin

step 1.  $n_{df} \leftarrow 0$

$n_{de} \leftarrow 0$

$n_{dv} \leftarrow 0$

step 2. for  $i=1, n_f$ , do

if  $F_I(i) \neq 1$  then

step 3.1  $n_{df} \leftarrow n_{df} + 1$

step 3.2  $F_d(n_{df}) \leftarrow F(i)$

step 3.3  $\text{FMAP}(i) \leftarrow n_{df}$

if  $F_I(i) = 0$  then

step 3.4  $\text{NORM}_{df}(n_{df}) \leftarrow \text{NEG}(\text{NORM}(i))$

else

step 3.4'  $\text{NORM}_{df}(n_{df}) \leftarrow \text{NORM}(i)$

end if

end if

end do {step 2}

step 4. for  $i = 1, n_v$ , do

$\text{VMAP}(i) \leftarrow 0$

end do {step 4}

step 5. for  $i=1, n_e$ , do

step 5.1  $\langle f_1, f_2, \dots, f_k \rangle \leftarrow E_f(i)$

```

step 5.2    NewEfi <-- nil
step 5.3    for j=1, k, do
              if FI(fj) ≠ 1 then
step 5.4        call INSERT(NewEfi, FMAP(fj))
              end if
              continue {step 5.3}
              if NewEfi ≠ nil then
step 5.5        nde <-- nde + 1
step 5.6        Edf(nde) <-- NewEfi
step 5.7        EMAP(i) <-- nde
step 5.8        <v1, v2> <-- E(i)
              if VMAP(v1) = 0 then
step 5.9          ndv <-- ndv + 1
                  Vd(ndv) <-- V(v1)
                  VMAP(v1) <-- ndv
              end if
              if VMAP(v2) = 0 then
step 5.10         ndv <-- ndv + 1
                  Vd(ndv) <-- V(v2)
                  VMAP(v2) <-- ndv
              end if
step 5.11       Ed(nde) <-- (VMAP(v1), VMAP(v2))
              end if
            end do {step 5}

step 6        for i=1, ndf, do
                  <e1, e2, .., ek> <-- Fd(i)
                  Fd(i) <-- <EMAP(e1), EMAP(e2), .., EMAP(ek)>
            end do {step 6}

end {DIFFBUILD}

```

---

Comments on **DIFFBUILD**: Variables  $n_{df}$ ,  $n_{de}$  and  $n_{dv}$  are the numbers of faces, edges and vertices of  $\Omega_d$  that have been found. Three arrays **FMAP**, **EMAP** and **VMAP** are the mappings from the preserved faces, edges and vertices of  $\Omega_p$  to those of  $\Omega_d$ . For example,  $\text{FMAP}(5)=2$  means faces 5 of  $\Omega_p$  is face 2 of  $\Omega_d$ . At step 1 the total number of faces, edges and vertices in  $\Omega_d$  is 0. The loop at step 2 generates the  $\text{NORM}_f$  set  $\text{NORM}_{df}$  of  $\Omega_d$ : if a face of  $\Omega_p$  is an internal face of  $\Omega$ , it is preserved on  $\Omega_d$  and its normal should be negated (step 3.4). Otherwise it is also preserved but its normal should be the same as the original (step 3.4'). Step 3.2 retrieves the current preserved face of  $\Omega_p$  into the  $F_d$  set of  $\Omega_d$  while step 3.3 establishes the index mapping **FMAP** between them. The edge indices of the faces in  $F_d$  are still the originals from  $E$  and they will be mapped to  $E_d$  once **EMAP** are established. The mapping **VMAP** is initialized at step 4. The entire loop of step 5 generates the  $V$ ,  $E$  and  $E_f$  arrays of  $\Omega_d$ ,  $V_d$ ,  $E_d$  and  $E_{df}$ , as well as establishing the mappings **VMAP** and **EMAP**. Step 5.1 retrieves all the faces of  $\Omega_p$  that are adjacent at an edge of  $\Omega_p$ . By checking their  $F_f$ , those unpreserved faces (step 5.3 to step 5.4) are filtered out. If the remainder is not empty, this edge as well as its end points must be preserved on  $\Omega_d$ . This is done by the following: step 5.6 insert the ( $\Omega_d$ ) face adjacency relation of the edge into the  $E_f$  array  $E_{df}$  of  $\Omega_d$ ; step 5.7 assigns the mapping **EMAP** of the edge. Steps 5.9 and 5.10 establish the mapping **VMAP** of the two end points of the edge and store their coordinates from  $V$  of  $\Omega_p$  into the  $V$  array  $V_d$  of  $\Omega_d$ . At step 5.11 this edge with its new end point indices of  $V_d$  is stored into  $E$  array  $E_d$  of  $\Omega_d$ . Finally at step 6, the edge indices in  $F_d$  are replaced with their mappings in  $E_d$ .

**Theorem 1.** The deficiency  $\Omega_d$  of a pseudo manifold  $\Omega$  can be obtained in  $O(N \log N + K)$  time where  $N$  is  $\max\{n_e, n_f, n_v\}$  of  $\Omega$ , and  $K$  is the sum of the face adjacency indices of  $\Omega$ .

**Proof.** Since the  $\Omega_p$  of  $\Omega$  is  $O(N \log N)$  derivable from  $\Omega$  (see the comments on procedure **MERGE**), it is only necessary to analyze the procedure **DIFFBUILD**. The overall time from step 1 through step 4 is  $O(N)$ . The total time for the loop at step 5 plus the inner loop at step 5.3 is  $O(K)$ . Analogously, the loop at step 6 is  $O(K)$  as well.

Q.E.D.

The occurrence of  $K$  can be somewhat unpleasant because of its seemingly

non-deterministic relation with  $N$ . Fortunately,  $K$  is shown to be  $O(n_f)$  where  $n_f$  is the total number of the faces of the pseudo manifold. (Refer to the Appendix.) Therefore, the deficiency of a pseudo manifold can be obtained in  $O(N \log N)$  time.

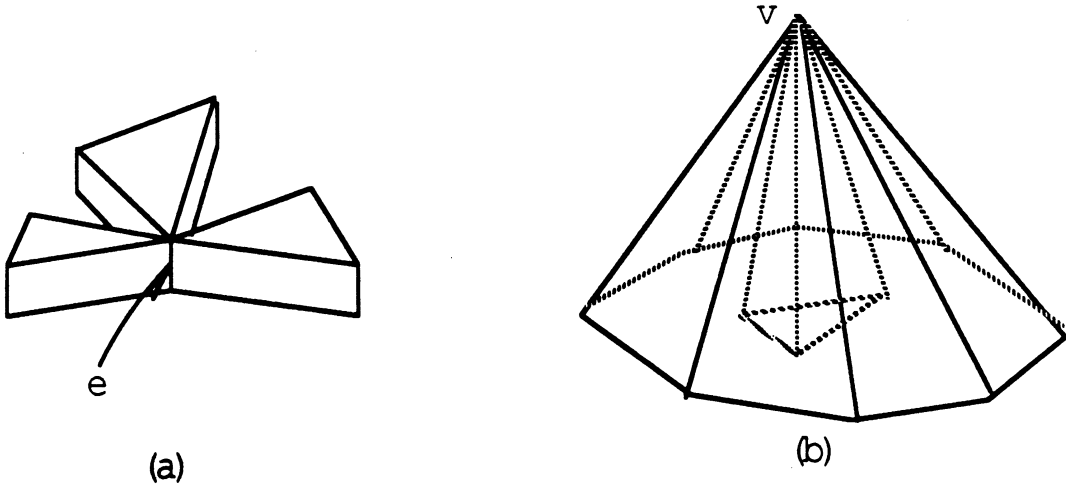


## Appendix. Space Linearity of a Pseudo Polyhedron

In this appendix, the space required by a pseudo polyhedron is shown to be linear in the number of its faces. Referring to Definition 3, let  $P = \langle V, E, F, \text{NORM}, E_f \rangle$  be a pseudo polyhedron with  $n_v$  vertices,  $n_e$  edges and  $n_f$  faces. The numbers  $n_v$  and  $n_e$  are shown to be both  $O(n_f)$  (for the items  $V$  and  $E$ ). In addition, the sum of the face adjacency indices of all the edges are shown to be linear in  $n_f$  (for  $F$  and  $E_f$ ). The face adjacency index of an edge is the number of the faces that meet at that edge. First, several definitions are introduced.

**Well-Adjacency of edges:** An edge of a pseudo polyhedron is called a *well-adjacent edge* if its face adjacency index is 2; otherwise it is an *ill-adjacent edge*.

**Well-Orientation of vertices:** A vertex  $v$  of a pseudo polyhedron is said to be *well-oriented* if the faces that are incident at  $v$  bear an order  $f_1, f_2, \dots, f_k$  such that  $f_1$  is adjacent to  $f_2$ ,  $f_2$  is adjacent to  $f_3$ ,  $\dots$ ,  $f_{k-1}$  is adjacent to  $f_k$ , and  $f_k$  is adjacent to  $f_1$ ; otherwise  $v$  is an *ill-oriented* vertex.



All the edges except  $e$  are well-adjacent

All the vertices except  $v$  are well-oriented

Figure A-1. Well-Adjacency and Well-Orientation

Figure A-1 illustrates an example of well-adjacency and well-orientation. Based on these two characterizations of edges and vertices, two operations are defined next on those ill-oriented vertices and ill-adjacent edges.

**Vertex Homogenization:** A *Vertex Homogenizing Operation* (VHO) on an ill-oriented vertex  $v$  is a replacement by a set of new vertices ( $v_1, v_2, \dots, v_m$ ) such that all the  $v_i$ 's are well-oriented and they have the same coordinates as  $v$ .

**Edge Homogenization:** An *Edge Homogenizing Operation* (EHO) on an ill-adjacent edge  $e$  is a replacement by a set of new edges ( $e_1, e_2, \dots, e_m$ ) such that all the  $e_i$ 's are well-adjacent and they have the same coordinates as  $e$ .

Figure A-2 and A-3 demonstrate VHO and EHO operations respectively.

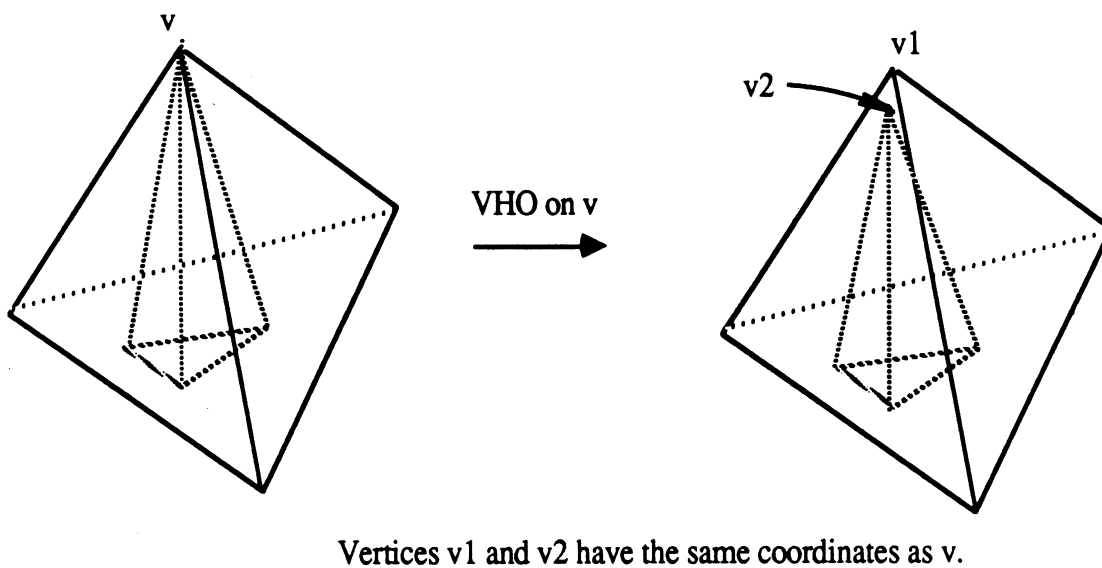


Figure A-2. Illustration of a Vertex Homogenizing Operation

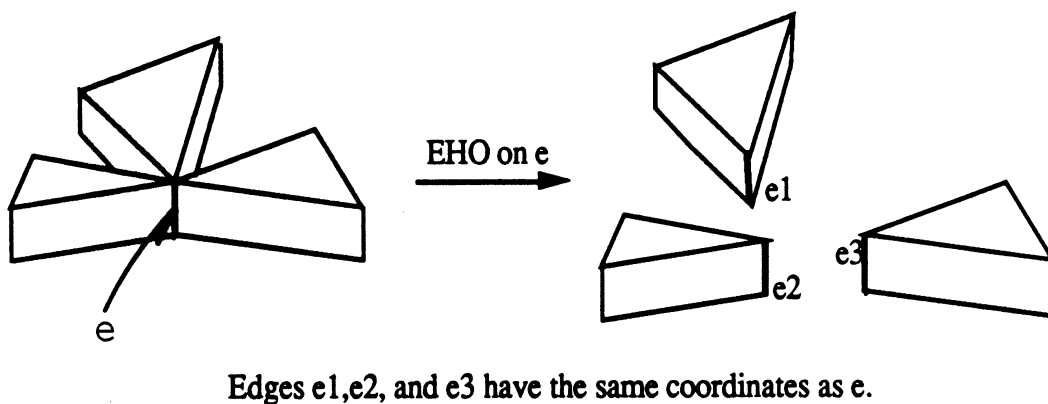


Figure A-3. Illustration of Edge Homogenizing Operation

By utilizing these two micro operations VHO and EHO, an operation on a pseudo polyhedron is defined below.

**Polyhedron Homogenization:** A *Polyhedron Homogenizing Operation* (PHO) on a pseudo polyhedron  $P$  is a series of VHO's and EHO's such that the resultant pseudo polyhedron  $P'$  has well-oriented vertices and well-adjacent edges only. See Figure A-4 for an example.

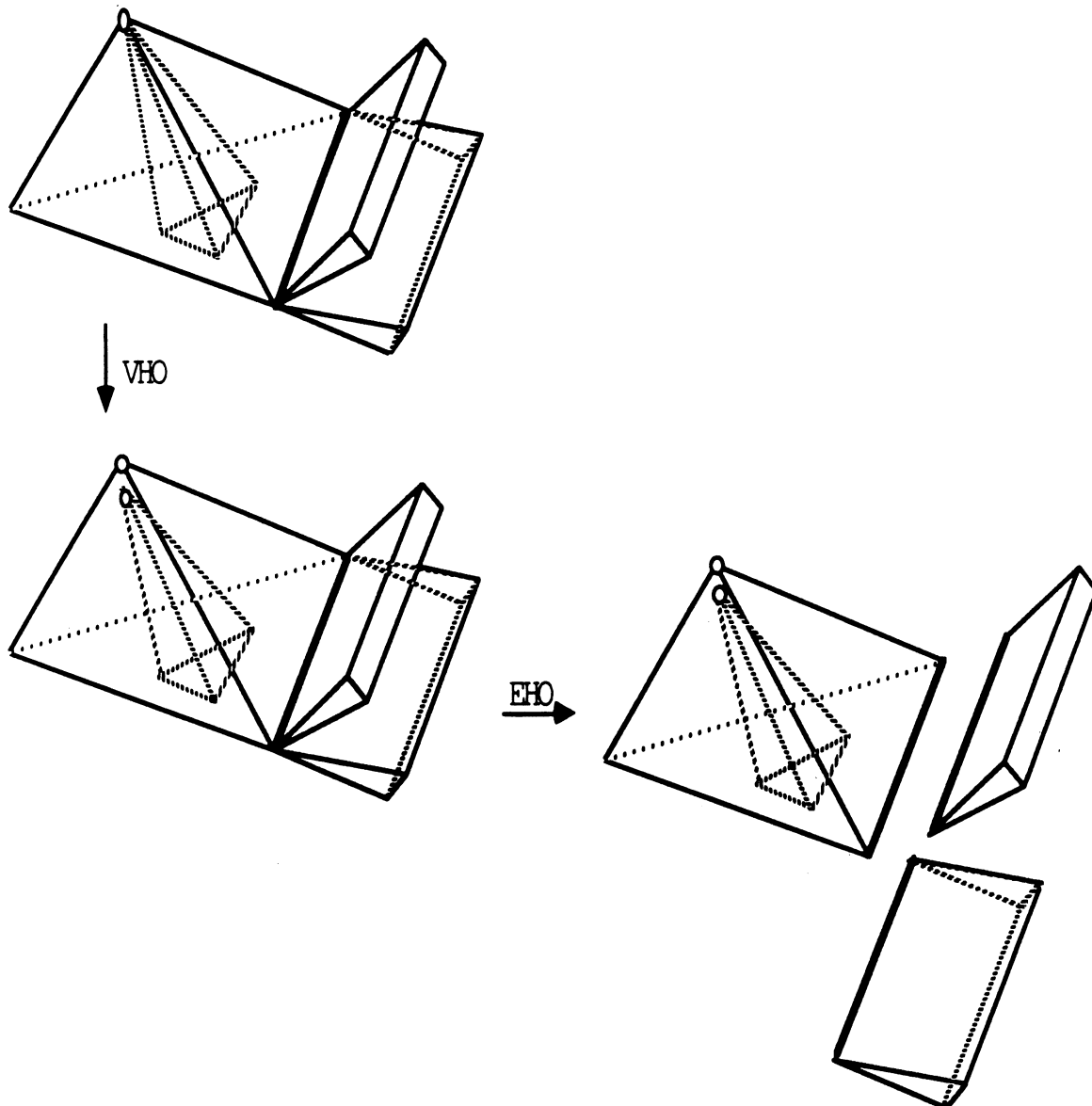


Figure A-4. Illustration of a Polyhedron Homogenizing Operation

**Lemma A.** The resultant pseudo polyhedron  $P'$  of a PHO on a pseudo polyhedron  $P$  is either a single polyhedron or a set of polyhedra.

**Proof.** Note that a polyhedron is a special case of pseudo polyhedra such that (a) all the faces of it are connected, and (b) all its vertices are well-oriented and all its edges are well-adjacent. By definition, PHO bears property (b) but not always (a).

Q.E.D.

The following theorem is induced by Lemma A.

**Theorem A.** Let  $P(n_v, n_e, n_f)$  be a pseudo polyhedron, with  $n_v$  vertices,  $n_e$  edges, and  $n_f$  faces. Then the following three claims are true:

- (a)  $n_v$  is  $O(n_f)$ ,
- (b)  $N_e$  is  $O(n_f)$ , and
- (c)  $K = \sum k_i$  is  $O(n_f)$ ;

where  $k_i$  is the face adjacency index of edge  $e_i$  ( $i=1,2,\dots,N_e$ ).

**Proof.** Let  $P_1, P_2, \dots, P_m$  be the polyhedra of  $P'(n_v', n_e', n_f)$  which is the resultant pseudo polyhedron of PHO on  $P$ , each of them has  $V_i$  vertices,  $E_i$  edges and  $F_i$  faces ( $i=1,\dots,m$ ). By Euler formula [2],  $V_i \leq 2F_i - 4$  and  $E_i \leq 3F_i - 6$  ( $i=1,\dots,m$ ). Summing both sides of the inequality yields  $n_v' = V_1 + V_2 + \dots + V_m \leq 2(F_1 + F_2 + \dots + F_m) - 4m = 2n_f - 4m$ , and  $n_e' = E_1 + E_2 + \dots + E_m \leq 3(F_1 + F_2 + \dots + F_m) - 6m = 3n_f - 6m$ . Since  $n_v \leq n_v'$  and  $n_e \leq n_e'$ ,  $n_v \leq 2n_f - 4m$  and  $n_e \leq 3n_f - 6m$ . To prove (c), let  $L$  be the total number of ill-adjacent edges of  $P$ . Each EHO operation replaces an ill-adjacent edge of  $P$  with a number of well-adjacent edges. For an ill-adjacent edge  $e_i$ , exactly  $(k_i / 2) - 1$  new edges will be generated. This implies, however, that the sum  $K' = \sum k_i$  over all the  $L$  ill-adjacent edges of  $P$  is  $2(n_e' - n_e) + 2L$ . The sum  $K'' = \sum k_i$  over the rest  $n_e - L$  well-adjacent edges of  $P$  is certainly  $2(n_e - L)$ . Therefore,  $K = K' + K'' = 2n_e' \leq 6n_f - 12m$ . Q.E.D.

## **References**

- [1] A. Baer, C. Eastman, and M. Henrion, "Geometric Modeling: A Survey" Computer -Aided Design, Vol. 11, No. 5, Sept. 1979, pp. 253-272.
- [2] C. Bollobas, Graph Theorey, Springer-Verlag, 1979
- [3] A. Cary, "BUILD USERS' GUIDE", University of Cambridge, C.A.D. Group Document, No. 102, Nov. 1979.
- [4] B.M. Chazelle, "Convex Decomposition of Polyhedra", ACM Symposium on Theory of Computing, Milwaukee 1981, pp. 70-79.
- [5] H. Chiyokura, F. Kimura, "A Method of Representing the Solid Design Process", IEEE Computer Graphics and Application, April 1985, Vol. 5, No.4, pp. 32-41.
- [6] F.P. Preparata and M. L. Shamos, Computational Geometry, Springer-Verlag, 1985
- [7] F.P. Preparata and S.J. Hong, "Convex Hull of Finite Sets of Points in Two and Three Dimensions", Comm. ACM 2(20), Feb. 1977, pp. 87-93.
- [8] A.A.G. Requicha, "Representation for Rigid Solids: Theory, Methods and Systems", ACM Computing Surveys, Vol. 12, No 4, Dec. 1980, pp. 437-464.
- [9] A.A.G. Requicha, H.B. Voelcker, "Solid Modeling: Current Status and Research Directions", IEEE Computer Graphics and Application, Oct. 1983, Vol. 3, No.7, pp. 25-37.
- [10] A.A.G. Requicha, H.B. Voelcher, "Constructive Solid Geometry", University of Rochester, Production Automation Project, TM-25, Nov. 1977.
- [11] K. Weiler, "Topologocal Structures for Geometric Modeling", Ph.D dissertation, Rensselaer Polytechnic Institute, Aug. 1986.



- [12] T.C. Woo, "Feature Extraction by Volume Decomposition", Proc. Conf. On CAD/CAM Technology in Mechanical Engineering, Cambridge, Mass. 1982
- [13] T.C. Woo, "A Combinatorial Analysis of Boundary Data Structure Schemata", IEEE CG&A, Vol. 5, No. 3, March 1985, pp. 19-27.