Algorithmic aspects of alternating sum of volumes.
Part 1: Data structure and difference operation

K Tang and T Woo

In terms of basic theory, a unique conversion from a boundary representation to a CSG representation is of importance. In terms of application, the extraction of features by convex decomposition is of interest. The alternating sum of volumes (ASV) technique offers both. However, some algorithmic issues are still unresolved. The paper is the first section of a 2-part paper that addresses specialized set operations and the convergence of the ASV process. In the first part, a fast difference operation for the ASV process and a data structure for pseudopolyhedra are introduced.

A fast difference operation between an object and its convex hull is made possible by the crucial observation that it takes only 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 pseudopolyhedron, capturing the special relationship between an object and its convex hull. That the data structure is linear in space is also shown.

The idea of the alternating sum of volumes (ASV) technique 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 3D component. As an example, consider the object shown in Figure 1: a block with a slot and a hole. The ASV series of this object is

$$
\Omega_0 = \Omega
$$

where

- $\Omega_i$ 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 $H_{i-1}$ from $H_i$, where $i = 1, 2, \ldots$. As $\Omega_{i+1}$ becomes the null set $\emptyset$, the $H_i$s are collected.
starting with $H_i$. The ASV expression is formed by alternating '－' and '＋' signs, as in $H_0 - H_1 + H_2 - \ldots$.

Consider the machining process of the mechanical component in the above example. There are two features: a slot and a hole. They can be extracted by algebraic manipulation of the object’s ASV as illustrated in Figure 3. Parenthesizing $H_i$ and $H_j$ forces a change in the sign from + to −. Subtracting $H_2$ from $H_1$ yields a new $H'$ for a disjunctive expression:

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

If $H'_0$ is the stock, $H'$ and $H_1'$ can be thought of as volumes to be removed to create the slot and the hole.

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 can in turn help the process planners in deciding on suitable manufacturing operations, such as machining or welding.

The ability to 'disassemble' allows conversion from boundary-based solid-modelling systems to those that are CSG-based.

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

When a deficiency $\Omega_n$ becomes nonconvergent, one solution is to divide it into convex subsets. However, there is a drawback. It is known that there can be $O(n^2)$ convex subsets, where $n$ is the number of concave edges, and each subset requires further polynomial time to determine. An alternative is to decompose the deficiency into subsets that are themselves convergent, so that the ASV series can expand further. For example, the object $\Omega$ in Figure 6 is nonconvergent. 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 edges of the type of $e$ in Figure 6 may be a very small subset of the set of all concave edges encourages inquiry.

![Figure 3. Algebraic manipulation of ASV series into disjunctive form](image)

![Figure 4. Algebraic manipulation of ASV series into conjunctive form](image)

![Figure 5. Example of ASV nonconvergence](image)

![Figure 6. Remedy for nonconvergence](image)
This paper is the first part of a 2-part paper. Part 1 deals with a special difference operation. While a general difference operation may be invoked, the relationship between an object and its convex hull merits investigation. This relationship is made concrete by the notion of a pseudomanifold object, shown as \( \Omega \) in Figure 6. This is an entity with characteristics between those of a 2-manifold and a nonmanifold object. With the aid of a data structure for pseudomanifold objects, the use of an \( O(n \log n) \) time algorithm for computing a convex deficiency is made possible, where \( n \) is the number of faces of the pseudomanifold. 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 nonconvergence and its remedy as illustrated by Figure 6 is given in Part 2 of this paper.

**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 3D Euclidean space, \( E^3 \). It must satisfy certain restrictions. Because the ASV process 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. In addition to this restriction of homogeneous three-dimensionality, the objects must also be closed under the (regularized) difference operation, i.e. they should have differential preservability. To define a domain of objects that will meet both the restrictions, some definitions of the interior and boundary points of a 3-dimensional point set must 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 3-dimensional neighbourhood 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 relationship between a boundary point and its neighbouring points of a set \( S \) in \( E^3 \) is described by one of three characterizations, namely manifold, pseudomanifold, and nonmanifold. A point \( p \) in \( B(S) \) is called a 2-manifold point if it has a 3D neighbourhood such that the subset of the points of \( S \) contained in that neighbourhood is topologically equivalent to a hemisphere. A point \( p \) is a pseudomanifold point if every 3D neighbourhood of it contains some points in \( I(S) \). If a point \( p \) has a 3D neighbourhood such that the subset of the points of \( S \) contained in that neighbourhood entirely belong to \( B(S) \), then it is called a nonmanifold point. As an example, the boundary surface of the object in Figure 7 consists of the six faces of the cube and a ‘dangling’ face \( f \). All the boundary points except those on \( f \) (including edge \( e \)) are 2-manifold points. The boundary points on the six faces of the cube, including edge \( e \), are pseudomanifold points. The nonmanifold points are those on face \( f \) but not on edge \( e \).

**Definition 2:** A point set \( S \) in \( E^3 \) is a 2-manifold set if \( I(S) \) is connected, and every point in \( B(S) \) is a 2-manifold point. \( S \) is a pseudomanifold set if every point in \( B(S) \) is a pseudomanifold point. \( S \) becomes a nonmanifold set if \( B(S) \) contains some nonmanifold points.

A pseudomanifold point is a relaxation of a 2-manifold point, i.e. it only requires that every neighbourhood of the point contains some interior points of the set, but with no topological constraint on the neighbourhoods. A pseudomanifold set need not be a connected set either. The relationship of these three sets can best be described in Figure 8. Because an object must have homogeneous three dimensionality, nonmanifold sets are immediately excluded from consideration. Although 2-manifold sets satisfy the homogenous 3-dimensionality condition, they are not closed under regularized difference operation. Pseudomanifold sets, while still conforming to homogeneous three dimensionality but also guaranteeing differential preservability, prove to be the only clan of objects suitable for ASV representations. Figure 9 shows several examples of 2-manifold, pseudomanifold and nonmanifold objects.

A data structure for the pseudomanifolds is crucial to both the development and analysis of algorithms. A
A pseudopolyhedron is 'almost' a polyhedron, except that it allows edges to have more than two adjacent faces.

**Definition 3:** A pseudopolyhedron \( P \) is a finite collection of planar faces such that:

1. Every edge of \( P \) has at least two adjacent faces.
2. If any two faces meet, they meet at a common edge.

Specifically, a pseudopolyhedron with \( n_v \) vertices, \( n_e \) edges and \( n_f \) faces is a quintuple \(<V, E, F, \text{NORM}, \text{El}>>\), which is defined as:

- \( V = \{v_1, v_2, \ldots, v_{n_v}\} \) is a list storing the \( n_v \) vertices; each \( v_i \) is a coordinate triple \((x_i, y_i, z_i)\).
- \( E = \{\langle v_{i_1}, v_{i_2}\rangle, \langle v_{i_3}, v_{i_4}\rangle, \ldots, \langle v_{i_k}, v_{i_{k+1}}\rangle\} \) 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. \( \langle v_5, v_2\rangle = \langle 3, 10\rangle \) means that the end points of the \( i \)th edge are \( v_3 \) and \( v_{10} \).
- \( F = \{F_1, F_2, \ldots, F_{n_f}\} \) stores face information. Each element \( F_i \) is itself an array of the form \( \langle e_{i_1}, e_{i_2}, \ldots, e_{i_k}\rangle \), where \( k \) is the total number of polygons in face \( F_i \); each \( e_{i_j} \) is the index of an edge in \( E \). For example, \( F = \{\langle 2, 4, 1\rangle, \langle 5, 7, 6, 3\rangle\} \) means that face \( F_1 \) is bounded by two simple polygons; the indices of the edges of the outer polygon are \( 2, 4, 1 \), and they are \( 5, 7, 6, 3 \) for the inner polygon. The edges are ordered clockwise (for the outer polygon) or counterclockwise (for the inner polygon).
- \( \text{NORM} = \{N_1, N_2, \ldots, N_{n_f}\} \) stores the outward normals of the \( n_f \) faces.

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

**DIFFERENCE OPERATION**

To study the difference operation between a pseudomanifold and its convex hull, the following notations are used: \( \Omega \) denotes a pseudomanifold object, \( \Omega_h \) its convex hull, and \( \Omega_d \) the deficiency \( \Omega_h \setminus \Omega \). The same notations are used to represent their defining pseudopolyhedra 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 that are also in \( \xi_h \).
- \( \xi_d : \{I(\Omega_h) \setminus \Omega\} \), interior (deficiency) points of \( \Omega_h \) excluding \( \Omega \).

Figure 10 shows these subsets in two dimensions.

**Definition 4:** A point \( p \in \Omega_h \) is a preserved point if, for any real number \( \varepsilon > 0 \), no matter how small it is, the open neighbourhood sphere that centres at \( p \) with radius \( \varepsilon \) always has a point in \( \Omega \). 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 $\mathbb{E}^3$ consisting of all the preserved points of $\Omega_h$.

The definition of deficiency here correlates with the intuitive view. The purpose of categorizing the points of $\Omega_h$ is to relate the interior and the boundary of the deficiency $\Omega_d$ to those 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 the sets $\xi_d$ and $\zeta_d$ be. By definition (of the deficiency), all the points of $\zeta_d$ are lost. As $\zeta_d$ is an open set, every point in it has a neighborhood sphere of points in $\zeta_d$ only and is thus interior to $\Omega_d$. This means that $\zeta_d$ must be a subset of the interior of deficiency $\Omega_d$. The neighborhood of any point in $\xi_d$ or $\zeta_d$ contains some points in either $\xi_d$ or $\{\mathbb{E}^3 - \Omega_h\}$, i.e., it contains points not belonging to $\Omega_d$, and hence, by definition, they cannot be in the interior of $\Omega_d$. These observations are summarized in the following lemma.

Lemma 1: The deficiency $\Omega_d$ of a pseudomanifold $\Omega$ is also a pseudomanifold, whose interior set $I(\Omega_d)$ is the $\xi_d$ set of $\Omega_h$ and whose boundary set $B(\Omega_d)$ is a subset of $\{\xi_d \cup \zeta_d\}$.

As noted in Lemma 1, the boundary surface $B(\Omega_d)$ of the deficiency $\Omega_d$ is a subset of $\{\xi_d \cup \zeta_d\}$ of $\Omega_d$. Because of the planarity of the faces, $B(\Omega_d)$ must be a set of some faces of $\Omega$ and $\Omega_d$. The key to the algorithm for finding the deficiency is to find these faces and the adjacency relationship between them, so that the result is a pseudopolyhedron representation of $\Omega_d$.

Definition 6: A face of a pseudomanifold $\Omega$ is a hull face if it is in $\xi_d$; 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 $\xi$ of $p$ that belongs to $I(f)$ and hence in $\xi_d$. As $\Omega$ is a pseudomanifold, every point in $\xi$, including $p$ itself, must have a neighborhood sphere that consists of points in $\{\xi_d \cup \zeta_d \cup \{\mathbb{E}^3 - \Omega_h\}\}$ only. Because of the way that a preserved face is defined, $p$ can only be a lost point. Therefore, every point in set $I(f)$ is a lost point.

QED

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 desired difference algorithm. It accepts as input $(V, E, F, \text{NORM}_f)$, which is the pseudopolyhedron representation of a pseudomanifold $\Omega$, and it outputs the pseudopolyhedron representation of the deficiency $\Omega_d$. Suppose that there is a procedure HULLP that takes as input the pseudopolyhedron $(V, E, F, \text{NORM}_f)$ of a pseudomanifold $\Omega$. Its outputs are two: one is the pseudopolyhedron of the resultant convex hull $\Omega_h$, and the other is two arrays $F_h$ and $E_h$, called hull tag arrays, that distinguish those faces and edges of $\Omega_h$ that do not belong to $\Omega$. Specifically, $F_h(i) = 1$ means that face $i$ of $\Omega_h$ is a face of $\Omega$. When $F_h(i) = 0$, the meaning is reversed. $E_h(i) = j$ means that edge $i$ of $\Omega_h$ is edge $j$ of $\Omega$, whereas $E_h(i) = 0$ means that edge $i$ is not an edge of $\Omega$. As most available 3D convex-hull algorithms support data structures that embody our pseudopolyhedra, the feasibility of the output of HULLP is justified. For 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 that face $i$ of $\Omega$ is an internal face, and, similarly, $E_I(i) = 1$ means that edge $i$ of $\Omega$ is an internal edge. Similarly, when $F_I(i)$ or $E_I(i)$ is equal to 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 pseudopolyhedron.

![Figure 11. Functionality of procedure HULLP](image)

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

MERGE algorithm

```
Procedure MERGE (nv, ne, nf, V, E, F, E_I, NORM, F_I, E_I)
    n_v', n_e', n_f', V', E', F', E_I', NORM', F_I', E_I')
    /* purpose: updates the pseudopolyhedron representation of a pseudomanifold $\Omega$ by adding the newly generated hull faces and hull edges of its convex hull to it.
```

volume 23 number 5 june 1991 361
**input:** \((n, n', n, V, E, F, E, NORM_f, F, E)\) \(\cdots\) pseudopolyhedron \(\Omega\); \(F\) and \(E\) are the internality tag arrays of its faces and edges.

**output:** \((n, n', n, V, E, F, E, NORM_f, F, E)\) \(\cdots\) pseudopolyhedron representation of the convex hull \(\Omega_h\) of \(\Omega\); \(F\) and \(E\) are the hull tag arrays of its faces and edges.

\(\cdots\) updated pseudopolyhedron of \(\Omega\) with the newly generated hull faces and hull edges of \(\Omega_h\) added; \(F\) and \(E\) are updated with the following convention:

\(F(i) = 0\) 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 \(\Omega\)

\(E(i) = 0\) 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 \(\Omega\)


```plaintext
/*
begin
  FTOP = n,
  FTOP = nf,
  for i = 1, n' do
    EMAP(i) = 0
  end do {step 1}
  for i = 1, n' do
    if FH(i) = 0 then
      FTOP = FTOP + 1
      (e1, e2, \ldots, ej) = F'(i)
      for j = 1, l do
        if EH(ej) = 0 then
          if EMAP(ej) = 0 then
            ETOP = ETOP + 1
            EMAP(ej) = ETOP
            E(ETOP) = E'(ej)
            EI(ETOP) = 2
          else
            EMAP(ej) = EH(ej)
          end if
        end if
      end do {step 5}
      call INSERT_Ef(Ef, EMAP(ej), FTOP)
    end if
  end do {step 2}
  FTOP = FTOP
  n' = ETOP
end {MERGE}
```

**Comments on MERGE**

Step 1 initializes two stack pointers FTOP and ETOP, 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. \(\text{EMAP}(i) = j\) means that edge \(i\) of \(\Omega\) is edge \(j\) of (current) \(\Omega\). Steps 3–13 are performed once for each face of \(\Omega\) that is not a face of \(\Omega\) (\(F_\Omega = 0\)). For each edge of a selected face, whether it is also an edge of \(\Omega\) is first checked. If it is not (when its \(E_\Omega = 0\)), and it has not been previously added to \(\Omega\), it is then added to \(E\) with its \(E_\Omega\) set to 2 and its EMAP set to a unique number ETOP (see steps 6–9). Otherwise, its EMAP is assigned with its \(E_\Omega\), which is the index of this edge in the original \(\Omega\) (step 6'). At step 10, the face-adjacency relationship 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 of \(\Omega\). (Note that, because \(\Omega_h\) is convex, every face of it has only one bounding polygon.) Step 13 assigns 2 to the \(F_i\) of this face that indicates that the added face is not a face of the original \(\Omega\). To analyse the time requirement of MERGE, note that each edge of \(\Omega_h\) has exactly two adjacent faces in \(\Omega_h\). At most, an edge of \(\Omega_h\) will be checked, retrieved
Figure 12. Example of PS description $\Omega_{ps}$

$V = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9\}$

$E = \{(v_1, v_2), (v_1, v_3), (v_2, v_3), (v_2, v_4), (v_2, v_5), (v_2, v_6), (v_3, v_7), (v_4, v_7), (v_5, v_8), (v_6, v_7)\}$

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

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

and stored twice. Steps 12 and 13 take constant time.

As a result, the loop from step 2 to step 13 is $O(n_e + n_f)$.

The output of MERGE, called the PS description of $\Omega$, and denoted as $\Omega_{ps}$, is a pseudopolyhedron. Figure 11 lists the $V$, $E$, $F$, $E_f$ entries of the $P$ description of a pseudomanifold $\Omega_{ps}$ itself, however, no longer represents a legitimate pseudomanifold, as it contains all the intermediate data for obtaining the deficiency $\Omega_{d}$. For the pseudomanifold $\Omega$ in Figure 12, the boundary of its deficiency $\Omega_{d}$ consists of the faces $f_{e}, f_{e}, f_{e}, f_{e}, f_{e}, f_{e}, f_{e}, f_{e}, f_{e}$, as defined in the $F$ entry of $\Omega_{ps}$. The vertices of $\Omega_{d}$ are $v_{e}, v_{e}, v_{e}, v_{e}, v_{e}, v_{e}, v_{e}, v_{e}$ of $\Omega_{ps}$, and the edges of $\Omega_{d}$ are $e_{e}, e_{e}, e_{e}, e_{e}, e_{e}, e_{e}, e_{e}, e_{e}, e_{e}, e_{e}, e_{e}, e_{e}, e_{e}$ of $\Omega_{ps}$.

The procedure DIFFBUILD given below will generate $\Omega_{d}$ using $\Omega_{ps}$. A constant time routine called INSERT($L$, $i$) will be used in the algorithm which appends an integer $i$ into an integer list $L$.

**DIFFBUILD procedure**

```
Procedure DIFFBUILD (nv, ne, nd, V, E, F, Ef, NORM, Fd, Ed, Fdf, NORMdf)

/* purpose: finds the deficiency $\Omega_d$ of a pseudomanifold $\Omega$, and outputs the pseudopolyhedron representation of $\Omega_d$ to the external. */

input: (nv, ne, nd, V, E, F, Ef, NORM, Fd, Ed, Fdf, NORMdf) ----the PS description of $\Omega$.

output: (ndv, nde, ndf, Vd, Ed, Fd, Fdf, NORMdf) ----the pseudopolyhedron representation of deficiency $\Omega_d$.

begin

(1) ndv <- 0
nde <- 0
nde <- 0

(2) for i = 1, nv, do

if Fi(i) $\neq$ 1 then

(3.1) ndv <- ndv + 1

(3.2) Fd(ndv) <- Fi(i)

(3.3) FMAP(i) <- ndv

if Fi(i) $=$ 0 then

(3.3') NORMdf(ndv) <- NEG(NORM(i))

else

end if

end if

end do {step 2}

(4) for i = 1, ne, do

VMAP(i) <- 0

end do {step 4}

(5) for i = 1, ne, do

(5.1) $\langle f_1, f_2, ..., f_k \rangle$ <- Ef(i)

NewEf <- nil

(5.3) for j = 1, k, do

if Ff(j) $\neq$ 1 then

(5.4) call INSERT(NewEf, FMAP(j))

end if

continue {step 5.3}

if NewEf $\neq$ nil then

(5.5) ndf <- ndf + 1

(5.6) Ed(ndf) <- NewEf

(5.7) EMAP(i) <- ndf

end if

end do {step 5}

end begin
```
Variables \( n_{fa}, n_{de}, \) and \( n_{v} \) are the numbers of faces, edges and vertices of \( \Omega \) that have been found. Three arrays \( FMAP, EMAP \) and \( VMAP \) are the mappings from the preserved faces, edges and vertices of \( \Omega_{ps} \) to those of \( \Omega \). For example, \( FMAP(5) = 2 \) means face 5 of \( \Omega_{ps} \) is face 2 of \( \Omega \). At step 1, the total number of faces, edges and vertices in \( \Omega_{ps} \) is 0. The loop at step 2 generates the NORMa set \( \text{NORMal}_{a} \) of \( \Omega_{ps} \); if a face of \( \Omega_{ps} \) is an internal face of \( \Omega \), it is preserved on \( \Omega_{ps} \), 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 \( \Omega_{ps} \) into the \( F_{d} \) set of \( \Omega \), 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 \( EMAP \) is established. The mapping \( VMAP \) is initialized at step 4. The entire loop of step 5 generates the \( V, E \) and \( E_{d} \) arrays of \( \Omega_{ps} \), as well as establishing the mappings \( VMAP \) and \( EMAP \). Step 5.1 retrieves all the faces of \( \Omega_{ps} \) that are adjacent at an edge of \( \Omega_{ps} \). By means of checking their \( F_{e} \), those unpreserved faces (steps 5.3-5.4) are filtered out. If the remainder is not empty, this edge as well as its end points must be preserved on \( \Omega_{ps} \). This is done as follows: step 5.6 inserts the \( \Omega_{ps} \) face-adjacency relationship of the edge into the \( E_{d} \) array \( E_{d} \) of \( \Omega_{ps} \); 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 \( \Omega_{ps} \) into the \( V_{d} \) array \( V_{d} \) of \( \Omega_{ps} \). At step 5.11, this edge, with its new end points indices of \( V_{d} \), is stored into the \( E_{d} \) array \( E_{d} \) of \( \Omega_{ps} \) and its coordinates from \( V \) of \( \Omega_{ps} \) into the \( V_{d} \) array \( V_{d} \) of \( \Omega_{ps} \). Finally, at step 6, the edge indices in \( F_{d} \) are replaced with their mappings in \( E_{d} \).

**Theorem 1:** The deficiency \( \Omega_{ps} \) of a pseudomanifold \( \Omega \) can be obtained in \( O(N \log N + k) \) time, where \( N \) is \( \max\{n_{fa}, n_{de}, n_{v}\} \) of \( \Omega \), and \( k \) is the sum of the face-adjacency indices in \( \Omega_{ps} \).

**Proof:** As the \( \Omega_{ps} \) of \( \Omega \) is \( O(N \log N) \)-derivable from \( \Omega \) (see the comments on the MERGE procedure), it is only necessary to analyse the procedure DIFFBUILD. The overall time taken from step 1 to 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.

QED

The occurrence of \( k \) can be somewhat unpleasant, because of its seemingly nondeterministic relationship with \( N \). Fortunately, \( k \) is shown to be \( O(n_{v}) \) where \( n_{v} \) is the total number of the faces of the pseudomanifold (see Appendix). Therefore, the deficiency of a pseudomanifold can be obtained in \( O(N \log N) \) time.

**SUMMARY**

An ASV expression of an object \( \Omega \) is based on two operations, convex-hull and difference. It is known that the convex-hull operations take \( O(N \log N) \) time, where \( N \) is the number of vertices in \( \Omega \). In this paper, it is shown that the difference between an object \( \Omega \) and its convex hull \( CH(\Omega) \) is also \( O(N \log N) \) in time.

Although the time to find the deficiency of a given \( \Omega \) is only \( O(N) \), it takes \( O(N \log N) \) time to construct its pseudopolyhedral representation \( \Omega_{ps} \). To support the data structure and the computation, manifolds, pseudomanifolds, and nonmanifolds are distinguished.

**REFERENCES**

1. Woo, T C 'Feature extraction by volume decompo-

sition' Proc. Conf. CAD/CAM Technology in computer-aided design

364
Well adjacency of edges: An edge of a pseudopolyhedron 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 pseudopolyhedron is said to be well oriented if the faces that are incident at \( v \) have an order \( f_1, f_2, \ldots, f_k \) such that \( f_1 \) is adjacent to \( f_2 \), \( f_2 \) is adjacent to \( f_3 \), \ldots, \( f_{k-1} \) is adjacent to \( f_k \), and \( f_k \) is adjacent to \( f_1 \); otherwise, \( v \) is an ill oriented vertex.

Figure 13 shows an example of well adjacency and well orientation. Based on these two characterizations of edges and vertices, two operations are defined on the 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 \( (v_1, v_2, \ldots, v_m) \) such that all the \( v_i \)s are well oriented, and 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 \( (e_1, e_2, \ldots, e_n) \) such that all the \( e_i \)s are well adjacent, and have the same coordinates as \( e \).

Figures 14 and 15 show VHO and EHO operations, respectively.

An operation on a pseudopolyhedron is defined below using these two microoperations VHO and EHO.

**Polyhedron homogenization**: A polyhedron-homogenizing operation (PHO) on a pseudopolyhedron \( P \) is a series of VHOs and EHOs such that the resultant pseudopolyhedron \( P' \) has well oriented vertices and well adjacent edges only (see Figure 16).

**Lemma A**: The resultant pseudopolyhedron \( P' \) of a PHO on a pseudopolyhedron \( P \) is either a single polyhedron or a set of polyhedra.

**Proof**: Note that a polyhedron is a special case of pseudopolyhedra such that (a) all the faces of it are connected, and (b) all its vertices are well oriented.
Figure 14. Vertex-homogenizing operation  
|Vertices v₁ and v₂ have the same coordinates as v.|

Figure 15. Edge-homogenizing operation  
|Edges e₁, e₂, and e₃ have the same coordinates as e.|  

Figure 16. Polyhedron homogenizing operation  

and all its edges are well adjacent. By definition, a PHO has the property (b) but not always (a). QED

The following theorem is induced by Lemma A.

Lemma B: Let P(nᵥ, nₑ, nᵦ) be a pseudopolyhedron with nᵥ vertices, nₑ edges and nᵦ faces. The following are then true:

nᵥ is O(nᵦ),
nₑ is O(nᵦ),
K = ∑kᵢ is O(nᵦ),

where kᵢ is the face-adjacency index of edge eᵢ (i = 1, 2, ..., nₑ).

Proof: Let P₁, P₂, ..., Pₘ be the polyhedra of P'(n'ᵥ, n'ₑ, n'ᵦ), which is the resultant pseudopolyhedron of a PHO on P; each of them has Vᵢ vertices, Eᵢ edges and Fᵢ faces (i = 1, ..., m). By the Euler formula, Vᵢ ≤ 2Fᵢ - 4 and Eᵢ ≤ 3Fᵢ - 6 (i = 1, ..., m). Summing both sides of the inequality yields n'ᵥ = V₁ + V₂ + ... + Vₘ ≤ 2F₁ + F₂ + ... + Fₘ - 4m = 2nᵦ - 4m, and n'ₑ = E₁ + E₂ + ... + Eₘ ≤ 3F₁ + F₂ + ... + Fₘ - 6m = 3nᵦ - 6m. As nᵥ ≤ n'ᵥ and nₑ ≤ n'ₑ, nᵦ ≤ 2nᵦ - 4m and nᵦ ≤ 3nᵦ - 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 eᵢ, exactly (kᵢ/2) - 1 new edges will be generated. This implies, however, that the sum K' = ∑kᵢ over all the L ill adjacent edges of P is 2(n'ₑ - nₑ) + 2L. The sum K'' = ∑kᵢ over the rest of the nₑ - L well adjacent edges of P is certainly 2(nᵦ - L). Therefore, K = K' + K'' = 2n'ₑ ≤ 6nᵦ - 12m. QED

366 computer-aided design