Converting Orthogonal Polyhedra from Extreme Vertices Model to B-Rep and to Alternating Sum of Volumes.

A. Aguilera * D. Ayala †

Abstract

In recent published papers we presented the Extreme Vertices Model (EVM), a concise and complete model for representing orthogonal polyhedra and pseudo-polyhedra (OPP). This model exploits the simplicity of its domain by allowing robust and simple algorithms for set-membership classification and Boolean operations that do not need to perform floating-point operations.

Several applications of this model have also been published, including the suitability of OPP as geometric bounds in Constructive Solid Geometry (CSG).

In this paper, we present an algorithm which converts from this model into a B-Rep model. We also develop the application of the Alternating Sum of Volumes decomposition to this particular type of polyhedra by taking advantage of the simplicity of the EVM. Finally we outline our future work, which deals with the suitability of the EVM in the field of digital images processing.

1 Introduction

In previous papers we presented a specific model for OPP, the Extreme Vertices Model, (EVM). This model is very concise, and although it only needs to store some of the OPP vertices, it has been proved to be complete. In [2] we presented the EVM for OP, a Boolean operations algorithm and an application consisting in using OP as geometric bounds in CSG. In [3] the domain was extended to OPP and we proved the completeness of the model and all the remaining formal properties. We also analyzed set-membership classification algorithms in the EVM. The problems of point and plane classification were extensively detailed in [4].

In this paper we present two contributions related with the model. We first present an algorithm which converts from EVM into a B-Rep. Then, we develop the application of the Alternating Sum of Volumes decomposition to the particular type of OPP by taking advantage of the simplicity of the EVM.

The paper is arranged as follows. The section below includes a brief review on the EVM, focusing particularly on those concepts and properties which are needed in the following sections. Section 3 explains the EVM to B-Rep conversion algorithm. Section 4 introduces the ASV decomposition and section 5 develops the application of this technique to OPP. Finally, the last section outlines future work which is oriented to the study of the suitability of the EVM in the field of digital images processing.

*Universidad de las Américas-Puebla, Puebla (México), aguilera@mail.udlap.mx

†Universitat Politècnica de Catalunya, Barcelona (Spain), dolorsa@lsi.upc.es

2 The Extreme Vertices Model

2.1 Orthogonal polyhedra (OP) and pseudo-polyhedra (OPP)

An OP is a polyhedron with all of its edges and faces oriented in three orthogonal directions. A pseudo-polyhedron [26] is a regular polyhedron with a non-manifold boundary. An OPP is an orthogonal pseudo-polyhedron. The class of OPP involves a drastic restriction with respect to general polyhedra, concerning geometry, as follows from its definition. However, with respect to topology, OPP do not involve any restriction at all, and they can have any genus and any number of shells.

In order to classify vertices for our purposes, we have done an exhaustive analysis of the neighborhood of an orthogonal vertex [1]. Vertices in an OPP follow the same pattern as do nodes in the marching cubes algorithm [20]. There are $2^8 = 256$ combinations which are grouped, by applying rotational symmetries, into 22 cases [25] and, by grouping complementary cases into the 14 basic patterns [20]. Figure 1 shows the 22 cases (from a to v). The 14 basic patterns are those from a to n.

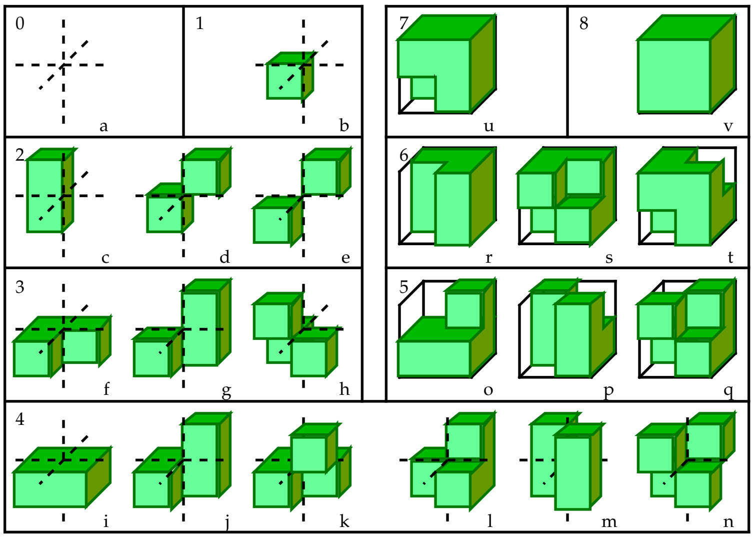

Figure 1: The 22 possible cases.

The 14 basic patterns have finally been grouped into 8 classes depending on the number of manifold and non-manifold incident edges. The name of the vertex indicates the total number of incident edges, if it is a non-manifold vertex (N) and, in this case, the number of non-manifold incident edges. See Figure 2 and the following table.

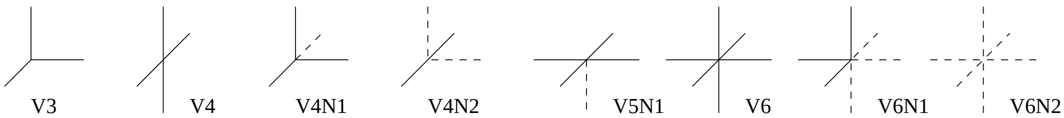

Figure 2: Vertex classification.

Vertex name	V3	V4	V4N1	V4N2	V5	V6	V6N1	V6N2
patterns	b, f	j	g	k	d	e, l	h	n

2.2 EVM definitions

Let P be an orthogonal pseudo-polyhedron (OPP).

A *brink* is the longest uninterrupted segment built out of a sequence of collinear and contiguous two-manifold edges of P. See Figure 3.

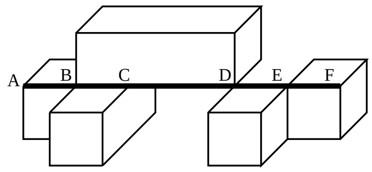

Figure 3: An OPP with a brink having five edges and six vertices.

In a brink each ending vertex is V3, V4N1 or V6N3 and the remaining (interior) vertices are V4, V4N2, V5N1 or V6. Vertices V6N6 do not belong to any brink. Vertices V3, V4N1, and V6N3 share the property of having exactly three incident two-manifold and linearly independent edges, regardless of the number of incident non-manifold edges, and are called *extreme vertices, EV*.

The *Extreme Vertices Model*, *EVM*, is a representation scheme for OPP in which any OPP is represented by its set of EV. From a theoretical point of view, vertices in the EVM can be without any order. However, for implementation purposes, they will be considered in an ordered way so that brinks parallel to axis X, Y or Z will appear consecutively. There are six different possible sortings: XYZ, XZY, YXZ, YZX, ZXY, and ZYX. For instance, when extreme vertices are sorted in the XZY or ZXY way, brinks parallel to the Y axis appear directly as pairs of consecutive EV (see Figure 4).

A *plane of vertices* is the set of vertices lying on a plane perpendicular to a main axis of P. A *slice* is the region between two consecutive planes of vertices. A *section* is the resulting polygon from the intersection between P and an orthogonal plane. If P has n planes of vertices, $plv_i(P), i = 1 \ldots n$, it will have $n-1$ slices and it can be expressed as $P = \bigcup_{k=1}^{k=n-1} slice_k(P)$. Each slice $slice_k(P)$ has its representing section $S_k(P)$ and there are two more empty sections, the initial and final sections, $S_0(P)$ and $S_n(P)$ (see Figure 4). Planes of vertices and sections are 2D-OPP embedded in 3D space, and we will sometimes need to work with their 2D projection, so we will denote by $\overline{P}$ the projection of a (d-1)-dimensional OPP, P, which is embedded in E^d, onto a main plane parallel to P. In order to obtain such a projection we only need not to consider the first coordinate of all P vertices.

All these definitions can be extended to any dimension [10]. In this paper we are concerned with dimension ≤ 3.

2.3 Properties of the EVM

All the following properties are formally demonstrated in [1].

The first property concerning EVM is that coordinate values of non-extreme vertices may be obtained from EV coordinates. Some non-extreme vertices correspond to the intersection of two or three perpendicular brinks and, therefore, their coordinates come directly from the EV coordinates of these brinks. Coordinates of the remaining non-extreme vertices are obtained from EV and from the non-extreme vertices first obtained.

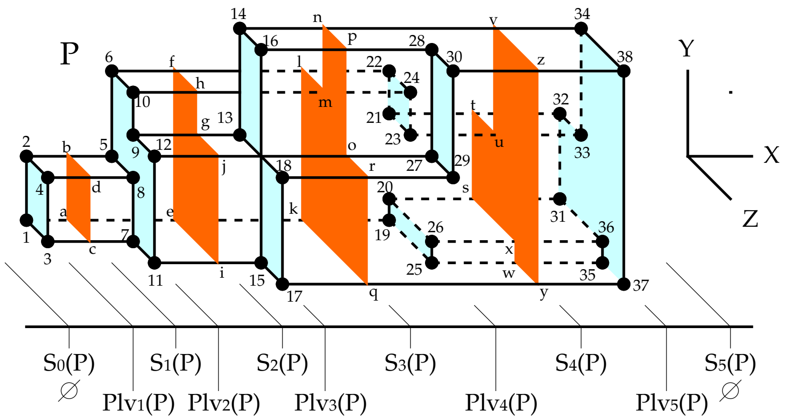

Figure 4: An OP, P, with five planes of vertices (light regions) and four slices with their corresponding sections (dark regions). Extreme vertices are numbered in the same way as they appear in a XZY-sorted EVM.

The two next properties shown below relate to sections and planes of vertices of P. We can compute sections from planes of vertices:

$$S_0(P) = \emptyset$$
$$\overline{S_i(P)} = \overline{S_{i-1}(P)} \otimes^* \overline{plv_i(P)}, \forall i \in [1, np]$$

And conversely, we can compute planes of vertices from sections:

$$\overline{plv_i(P)} = \overline{S_{i-1}(P)} \otimes^* \overline{S_i(P)}, \forall i \in [1, np]$$

where $\otimes^*$ is the regularized XOR operation.

Applying the definition of the $\otimes$ operation, this last equation can be expressed as: $\overline{plv_i(P)} = \overline{S_{i-1}(P)} \otimes^* \overline{S_i(P)} = (\overline{S_{i-1}(P)} -^* \overline{S_i(P)}) \cup (\overline{S_i(P)} -^* \overline{S_{i-1}(P)}), \forall i \in [1, np]$, and, thus, we can decompose any plane of vertices into two terms that we will call *forward difference* ($\overline{FD_i(P)} = (\overline{S_{i-1}(P)} -^* \overline{S_i(P)})$ and *backward difference* ($\overline{BD_i(P)} = (\overline{S_i(P)} -^* \overline{S_{i-1}(P)})$).

The following property guarantees that the correct orientation of all faces of P can be obtained from its EVM. $FD_i(P)$ is the set of faces on $plv_i(P)$ whose normal vector points to one side of the main axis perpendicular to $plv_i(P)$, while $BD_i(P)$ is the set of faces whose normal vector points to the other side (see Figure 5).

Concerning the so-called formal properties [23] we have proved that EVM is complete (non-ambiguous) and unique. The domain is limited to OPP and there is a validity condition for a finite point set to be a valid EVM for some OPP [1].

2.4 Boolean operations

The EVM behaves nicely with Boolean operations. The XOR operation applied to EVM represented OPP fulfills the following nice property:

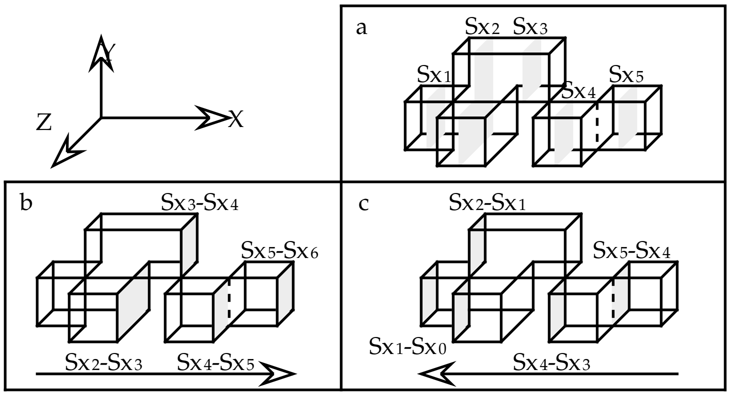

Figure 5: An OPP with its a) sections, b) forward differences and c) backward differences, perpendicular to X.

Theorem 1. *Let P and Q be two d-D ($d \leq 3$) OPP, having $EVM(P)$ and $EVM(Q)$ as their respective models, then $EVM(P \otimes^* Q) = EVM(P) \otimes EVM(Q)$*

This theorem is formally proved in [1]. It is proved by induction over the dimension and the basis of the induction (case 1D) is proved exhaustively. The property means that XOR between two OPP, which are infinite sets of points, can be carried out by applying the operation XOR to their EVM models, which are finite sets of EV.

The following two properties are corollaries of the previous one and are used in the application presented in section 5.

Corollary 1. *If $Q \subseteq P$ then $EVM(P -^* Q) = EVM(P) \otimes EVM(Q)$*

Corollary 2. *If P and Q are quasi-disjoint, then $EVM(P \cup^* Q) = EVM(P) \otimes EVM(Q)$*

General Boolean operations between two OPP can be carried out by applying the same operation over the corresponding OPP sections. The corresponding algorithm is presented in [2] and consists in a geometric merger between the EVM of both operands.

2.5 The splitting operation

Set-membership classification tests were given a general analysis in [3] and the particular cases of point and plane classification were detailed in [4]. However, here we review the splitting operation as it is needed in the application presented in section 5.

The classification of an OPP, P, against a splitting plane, SP, perpendicular to a main axis produces two polyhedra, Q and R, in the IN and OUT half-spaces of SP respectively. EVM (Q) and EVM(R) will be subsets of EVM(P) except for some possible new vertices that will be created and that will lie on SP. If SP is perpendicular to the C axis, only brinks parallel to this axis, C-brinks, have to be considered and these brinks appear as consecutive pairs of vertices in a conveniently ordered EVM (see Figure 6).

Let $V_b = V_{2k-1}$ and $V_e = V_{2k}$ be the beginning and ending vertices of the kth C-brink. The classification of this brink with respect to SP gives the following two cases:

- V_b and V_e lie in the same half-space of SP or one of them is ON. Then both of them will be assigned either to Q or to R.
- Each vertex, V_b and V_e, belongs to a different half-space. A new vertex V_i is computed as the intersection between the C-brink and SP. The brink is split into two brinks: the brink from V_b to V_i goes to Q and the brink from V_i to V_e goes to R. V_i is obtained without any floating-point computation. Consider an ZYX-sorted EVM and let $x = x_p$ be the SP equation, $V_b = (x_1, y, z)$ and $V_e = (x_2, y, z)$, then $V_i = (x_p, y, z)$.

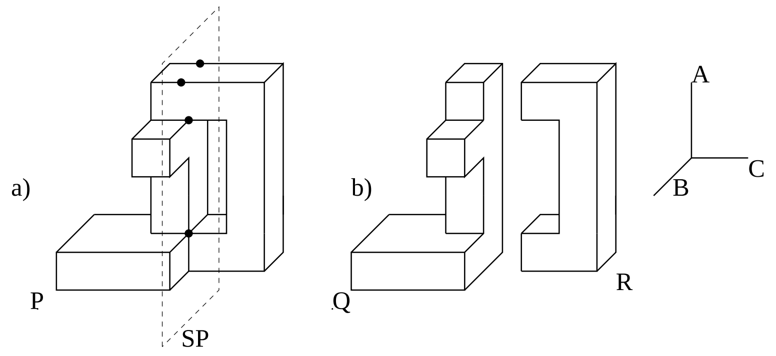

Figure 6: Splitting operation. a) Object P and splitting plane SP. Dots show new vertices created. b) Objects Q and R.

2.6 Applications

There are a number of published papers dealing with OP. In [13], [14] the problem of converting a B-Rep into a Peterson-style CSG is studied for OP.

In [7] a method is presented for simplifying OP. This method has been extended to general polyhedra but it uses OP in its process [6], [5].

In [10] a representation scheme for OPP in any dimension is presented and operations such as face detection and Boolean operations are studied. This representation is very similar to ours but it includes all the vertices with assigned colors. The authors work in the field of dynamical systems and restrict the state-space to being OPP [11].

Concerning EVM-represented OPP, in [2] the suitability of OPP as geometric bounds in CSG is discussed and the use of OPP as geometric approximations of general polyhedra is presented in [1].

The restricted class of convex and orthogonal polyhedra, i.e., orthogonal boxes, has been widely used in many applications [22], [12], [24].

3 EVM to B-Rep Conversion Algorithm

The EVM is complete, which means that all the geometry and topology can be obtained from it. In this section we present a conversion algorithm from EVM to a B-Rep.

The input of the algorithm is the set of points constituting the EVM of an OPP and the output is the set of faces of a B-Rep model each one with its corresponding normal vector and with its associated set of edges. Edges are oriented according to the normal vector of the face to which they belong. Therefore the output consists in the geometrical information corresponding to the normal vectors of each face and the coordinates of each vertex and the topological relations $f : \{e\}$ and $e : \{v\}$.

The algorithm does not provide edges ordered in the traveling order around faces and does not make distinction between edges belonging to the external boundary and to the possible internal boundaries (holes) of a face. If such order and distinction are required then a well-known postprocess is needed [12] which applies a domino-like procedure to obtain contours, and several point polygon containment tests in order to classify contours as external or as holes. An outline of the algorithm is showed below:

```
procedure EVMtoBRep(I p : EVM, O q : BRep)
  var
    dim : integer   {current dimension}
    dir : boolean   {orientation for faces and edges}
  endvar
  q := IniBRep(); dim := 3; dir := true
  Sort(p, XYZ); ProcessDifferences(p, dim, dir, q)
  Sort(p, YZX); ProcessDifferences(p, dim, dir, q)
  Sort(p, ZXY); ProcessDifferences(p, dim, dir, q)
endprocedure
procedure ProcessDifferences(I p : EVM, I dim : integer,
  I dir : boolean, I/O q : BRep)
  var
    Si, Sj : EVM  {two consecutive sections}
    plv : EVM     {the current plane or line of vertices}
    ForwardDif, BackwardDif : EVM
  endvar
  dim := dim − 1; Si := ∅; plv := GetPlv(p, dim)
  while ¬EndEVM(p) do
        Sj := ComputeSeccion(Si, plv, dim)
        ForwardDif := OpBool(Si, Sj, dim, '−')
        BackwardDif := OpBool(Sj, Si, dim, '−')
        if dim = 2 → ProcessDifferences(ForwardDif, dim, dir, q)
                     Sort(ForwardDif, ACB)
                     ProcessDifferences(ForwardDif, dim, dir, q)
                     ProcessDifferences(BackwardDif, dim, ¬dir, q)
                     Sort(BackwardDif, ACB)
                     ProcessDifferences(BackwardDif, dim, ¬dir, q)
        □ dim = 1 → AddEdgeBRep(ForwardDif, dir, q)
                     AddEdgeBRep(BackwardDif, ¬dir, q)
        endif
        Si := Sj; plv := GetPlv(p, dim)
  endwhile
endprocedure
```

The algorithm works first for dimension 3 (3D) and then for dimension 2 (2D). In 3D, the set of EV of the EVM is sorted in three ways thus making it possible to obtain the faces parallel to each coordinate plane. Moreover the property concerning forward and backward differences (FD, BD), showed in section 2.3, allows to determine which of these faces have the normal vector pointing to the interior of the solid and which of them have it pointing out the solid. Then, in 2D, the sets of EV corresponding to FD and to BD are sorted in two orderings which enables us to obtain the edges parallel to each coordinate axis, also correctly oriented thanks to the mentioned property. FD and BD are initially already sorted in one way (the sorting which comes from the algorithm when it works in 3D, say ABC) and so we only need to sort them in the other possible way (ACB).

Planes and lines of vertices, sections, and FD and BD are EVM-represented 2D or 1D orthogonal objects. Planes of vertices come directly from the EVM. Sections are computed by means of XOR operations and FD and BD computation involve Boolean differences. The variable dir is used to assign the correct orientation to each face and edge: $dir = TRUE$ indicates that the FD normal vector points say to the solid interior while $dir = FALSE$ indicates that the BD normal vector points to the solid exterior.

When computing FD and BD (2D and 1D) not only the correct orientation of faces and edges is obtained but also vertices that did not appear in the EVM come up.

Figure 7 shows how this algorithm works in two examples corresponding to the planes of vertices plv_2 and plv_4 of Figure 5. In both cases a V6 vertex, V, appears which was not in the EVM. For plv_2 when the algorithm works in 3D (sorting XBC), the whole plane of vertices belongs to BD, then this BD is processed in 2D (sorting XZY). When the 1D BD $S_{xz2} - S_{xz1}$ is computed, both vertex V and edge $(V, 4)$ appear and when the 1D FD $S_{xz1} - S_{xz2}$ is computed, vertex V and edge $(3, V)$ both appear. For plv_4 when the algorithm works in 3D this plane of vertices is split into two faces which correspond to the 2D FD and BD and the vertex V is then obtained.

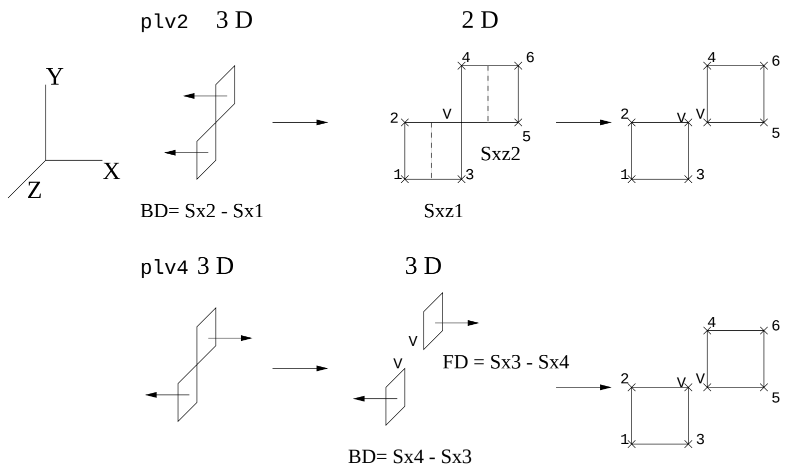

Figure 7: Working example for the EVM to B-Rep conversion algorithm.

In [1] the worst case and experimental complexities of this algorithm and of all the processes in which it is based (computing sections from planes of vertices and Boolean operations) are widely analyzed.

The first issue to remark is that the basic operation of all the processes involved in this algorithm is the XOR operation between finite sets of points. Therefore the algorithm is robust because does not perform any floating point operation.

As in most algorithms concerning EVM the bottle neck process is the computation of all the sections of the object from the EVM (i. e. from its planes of vertices) and it is this process that gives the worst case complexity to the conversion algorithm. The worst case complexity of computing all sections is $O(n \times np)$, n being the number of extreme vertices and np being the number of planes of vertices. As np ranges from 2 to n, the worst case complexity is quadratic. However, experimental results show that the average experimental complexity is far less than quadratic but slightly greater than linear. Performing a numerical regression to the data used in these experimental results of the form $y = ax^b$ the coefficient b obtained was $b = 1.221$.

Finally, it has to be noted that also as in most algorithms concerning EVM a preprocess is needed to sort extreme vertices and so there is a preprocess of $O(nlgn)$ complexity.

4 The Alternating Sum of Volumes Decomposition

A great amount of work has been done in the field of form feature recognition and there are several approaches in the literature. A survey of feature recognition can be found in [21].

As this section focuses on the Alternating Sum of Volumes decomposition, only this method will be reviewed. The earliest representations of form feature recognition [17], [29] propose a convex decomposition method which uses convex hulls and Boolean differences. Reflecting the nature of alternating volume contribution, this technique was called *Alternating Sum of Volumes* (ASV) [29] and form features can be automatically obtained by a manipulation of the resulting expression [26], [28].

4.1 Alternating Sum of Volumes

Let $CH(P)$ be the convex hull of a polyhedron P and $CHD^*(P)$ the regularized convex hull difference, also called the deficiency, of P, $CHD^*(P) = CH(P) -^* P$. The ASV decomposition of a polyhedron P, $ASV(P)$, is defined by means of the following recursive expression [15]:

$$ASV(D_k) = \begin{cases} D_k & \text{if } D_k \text{ is convex} \\ H_{k+1} -^* ASV(D_{k+1}) & \text{otherwise} \end{cases}$$

where $D_0 = P$, $H_k = CH(D_{k-1})$ and $D_k = H_k -^* D_{k-1}$ (D stands for deficiency).

The ASV decomposition allows P to be expressed as

$P = (H_1 -^* (H_2 -^* (H_3 -^* (H_4 -^* (\dots)))))$ or

$P = H_1 -^* H_2 +^* H_3 -^* H_4 +^* \dots$

$+^*$ being the quasi-disjoint union operator. The name ASV for this decomposition comes from this last expression.

In an ASV decomposition any convex hull H_k completely encloses all the subsequent convex hulls, i.e., $H_j \subseteq H_k, j > k$. This makes it possible to distinguish between terminating and non-terminating ASV series. ASV terminates when a deficiency D_n is found to be convex for some n. In this case $H_{n+1} = D_n$, $D_{n+1} = \emptyset$ and the relation $H_k \subset H_{k-1}$ holds for all $k \leq n$. Conversely, when two consecutive convex hulls coincide, $H_k = H_{k-1}$, the corresponding deficiency D_k becomes equal to a previous one D_{k-2} and the process becomes cyclic. This deficiency is said to be *ASV-irreducible* or *non-convergent* [15] [26].

Through the manipulation of an ASV series, form features of a given object can be extracted automatically. The following expression

$$P = H_1 -^* H_2 +^* H_3 -^* H_4 +^* \ldots = H_1 -^* (H_2 -^* H_3) -^* (H_4 -^* H_5) -^* \ldots$$

expresses P as the result of a series of volumes to be removed from an initial convex raw material H_1.

4.2 Non-Convergence of ASV and its remedy

When a deficiency becomes ASV-irreducible, the ASV decomposition is non-convergent. This problem can be solved by decomposing the non-convergent deficiency into subsets that are themselves convergent and finding the ASV series of each subset. This method is called Alternating Sum of Volumes with Partitioning (ASVP) [16] and consists in splitting the irreducible deficiency by a plane passing through vertices with two or more non-collinear concave incident edges.

4.3 Non-extremal faces reduction

A face of a polyhedron P is an *extremal face* if the corresponding plane is a supporting plane of P, i.e., if P is on one side of the closed half-space determined by the plane. Otherwise it is a *non-extremal face* [16].

In the ASV decomposition, the boundary of H_k consists of the set of extremal faces of D_{k-1} plus a new set of fictitious hull faces, while the boundary of D_k consists of the set of non-extremal faces of D_{k-1} plus the same set of fictitious hull faces. Therefore, the set of non-extremal faces of D_k is a subset of the set of non-extremal faces of D_{k-1}.

The convergence condition may be expressed in terms of non-extremal faces reducibility. A D_k with any non-extremal face is a convex deficiency, and then the ASV decomposition converges. So the remedy for non-convergence is to partition non-extremal faces of the irreducible deficiency in such a way that the resulting sets of non-extremal faces can be reduced.

5 Extracting Orthogonal Form Features using the ASV Technique

In this section we apply the ASV method to EVM-represented OPP by taking advantage of its simplicity. We will call this derived method ASOV (O stands for orthogonal).

Theorem 2. *Let P be an OPP, CH(P) its convex hull and OH(P) its orthogonal hull (minimum bounding box). Let A be the set of faces of P lying on the boundary of OH(P) and B the set of faces of P lying on the boundary of CH(P). Then A = B.*

This theorem is proved in [14]. It follows from it that computing deficiency sets with respect to $OH(P)$ is equivalent to computing them with respect to $CH(P)$. Therefore, we can use orthogonal hulls instead of convex hulls and, as the initial polyhedron is an OPP, we are guaranteed that all the objects in the ASOV decomposition will be OPP. And the EVM will be used to handle all the necessary operations.

Let P be an OPP and $D_0 = P$, $H_k = OH(D_{k-1})$, $D_k = H_k -^* D_{k-1}$, the same recursive expression as in the general case, holds:

$$ASOV(D_k) = \begin{cases} D_k & \text{if } D_k \text{ is a box} \\ H_{k+1} -^* ASOV(D_{k+1}) & \text{otherwise} \end{cases}$$

Figure 8 shows an example.

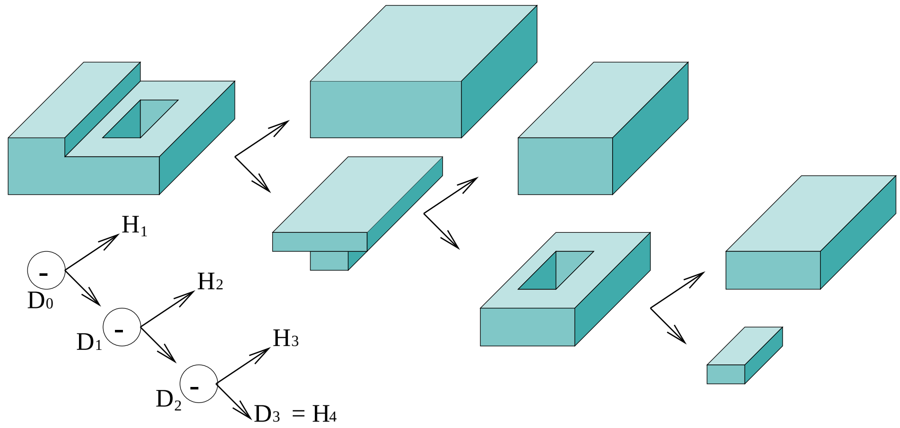

Figure 8: Example of Alternating Sum of Volumes.

In order to compute the ASV decomposition of P we need to compute the EVM of both the bounding box and the deficiency of an OPP.

Theorem 3. *$EVM(H_k) = MaxMin\ (EVM(D_{k-1}))$.*

Proof: $H_k = OH(D_{k-1})$. As H_k is a box, $EVM(H_k)$ contains all the vertices of H_k. Moreover, MaxMin (Vertices (D_{k-1})) = MaxMin $(EVM(D_{k-1}))$ because as the first property of the EVM stands, coordinate values of non-extreme vertices can be obtained from coordinate values of EV. □

Theorem 4. *$EVM(D_k) = EVM(H_k) \otimes EVM(D_{k-1})$.*

Proof: $EVM(D_k) = EVM(H_k -^* D_{k-1})$. Since $D_{k-1} \subseteq H_k$, by corollary 1 $EVM(H_k -^* D_{k-1}) = EVM(H_k) \otimes EVM(D_{k-1})$. □

Conversely, we can obtain P from its ASOV decomposition:

Theorem 5. *Let $H_i, \forall i \in [1, n]$ be the resulting boxes of the ASOV decomposition of an OPP, P, then $EVM(P) = (\otimes^*)_{i=1}^{i=n} EVM(H_i)$.*

Proof: The proof follows from the fact that $D_{k-1} \subseteq H_k, \forall k \in [1, n]$ and from corollary 1.□

Detecting and solving non-convergence is also derived from the general method.

Definition 1. *A full extremal face (FEF) of an OPP, P, is a face of P which coincides completely with a face of OH(P).*

Now we will demonstrate that the ASOV decomposition convergence is related with the existence of full extremal faces.

Lemma 1. *$H_k \subset H_{k-1} \Leftrightarrow D_{k-2}$ has at least one FEF.*

Proof: $\Leftarrow$ Let F be a FEF of D_{k-2}. Since $H_{k-1} = OH(D_{k-2})$ then F is also a FEF (i.e., a face) of H_{k-1}. Thus, neither D_{k-1} nor $H_k = OH(D_{k-1})$ contain F (remember that $D_{k-1} = H_{k-1} -^* D_{k-2} = H_{k-1} \otimes D_{k-2}$). This implies that $H_{k-1} \neq H_k$. Moreover, as in the general case $H_k \subseteq H_{k-1}$. Then, $H_k \subset H_{k-1}$.

$\Rightarrow$ $H_k = OH(D_{k-1})$ and $H_{k-1} = OH(D_{k-2})$. $H_k \subset H_{k-1}$ means that H_k and H_{k-1} differ at least in one face (for instance, faces on planes $y = y^k_{MAX}$ of H_k and $y = y^{k-1}_{MAX}$ of H_{k-1} had to be such that $y^k_{MAX} < y^{k-1}_{MAX}$). And this implies that a part of H_{k-1} has to be removed (the part from y^{k-1}_{MAX} to y^k_{MAX}) and this is only possible if the corresponding face of H_{k-1} fully coincides with a face of D_{k-2}, i.e., D_{k-2} has at least this FEF. $\square$

Figure 9 shows two 2D examples of an ASOV decomposition by the deficiency series. FEF are marked with a cross in their middle. Dashed lines in D_k correspond to the orthogonal hull H_{k+2}. The example in Figure 9 a) converges: all the deficiencies have at least one FEF and we can see how the deficiencies shrink in a direction perpendicular to the FEF. The example in Figure 9 b) do not converge: there is no FEF in deficiency D_1 and, therefore, $H_2 = H_1$, which is the non-convergence condition.

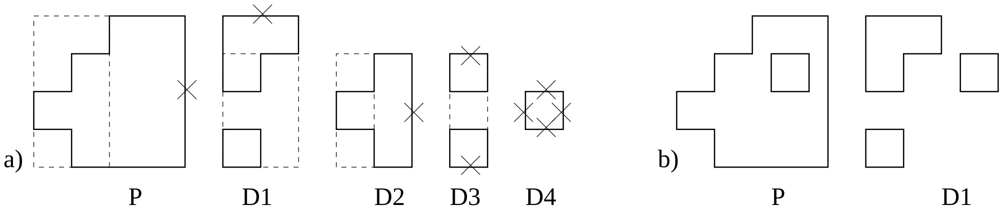

Figure 9: 2D examples of an ASOV decomposition.

Theorem 6. *ASOV converges if there is at least one FEF.*

Proof: As in the general case, ASOV converges when $H_k \subset H_{k-1}$. Then the proof follows from this fact and from lemma 1 $\square$

In order to remedy the problem of non-convergence, as in the general case, the irreducible deficiency is split by a plane (see section 2.5).

Definition 2. *The splitting vertex, SV, is the first extreme vertex of an OPP, P, which does not coincide with a corner of OH(P).*

SV belongs to the first plane of vertices of P and therefore to an extremal face. Our approach chooses the splitting plane to be a plane through SV. There are three orthogonal planes passing through SV. One of these planes is a supporting plane (the plane that contains the first plane of vertices) and cannot be a splitting plane, so in this case we have to choose one of the remaining two planes. Moreover, if SV belongs to the first line of vertices, then the two planes intersecting in this line are supporting planes and so, in this case, there is only one possible splitting plane. Generally, it is appropriate to select as the splitting plane the plane through SV which is perpendicular to the lines of vertices of the OPP.

As SV belongs to an extremal face, it will coincide with a corner of the orthogonal hull of at least one of the two split parts and this leads to the conversion of the split extremal face into fully extremal faces, thus enabling convergence.

Then the ASOV with partitioning (ASOVP) is defined by the following recursive expression:

$$ASOVP(D_k^j) = \begin{cases} D_k^j & \text{if } D_k^j \text{ is convex, i.e., a box} \\ H_{k+1}^j -^* ASOV(D_{k+1}^j) & \text{if } D_k^j \text{ has at least one FEF} \\ ASOVP(D_0^{2j+1}) +^* ASOVP(D_0^{2j+2}) & \text{otherwise} \end{cases}$$

where $D_0^0 = P$, $H_k^j = OH(D_{k-1}^j)$, $D_k^j = H_k^j -^* D_{k-1}^j$.
D_0^{2j+1} and D_0^{2j+2} are the (disjoint) split parts of D_k^j, i.e., $D_k^j = D_0^{2j+1} +^* D_0^{2j+2}$.
Figure 10 shows an example of ASOV with partitioning.

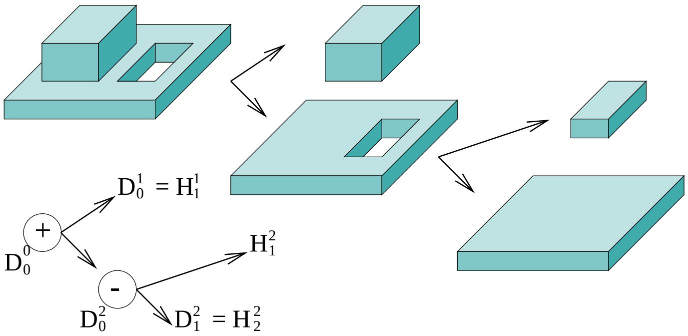

Figure 10: Example of ASOV with partitioning.

We can obtain P from its ASOVP decomposition:

Theorem 7. *Let H_i^j be the resulting boxes of the ASOVP decomposition of an OPP, P, then $EVM(P)$ can be expressed as the regularized XOR between all the H_i^j.*

Proof: An ASOVP is a tree in which the operations are Boolean differences or quasi-disjoint unions.

1. If P has at least one FEF then $P = H -^* D$ and $D \subset H$. Then, by corollary 1, $EVM(P) = EVM(H) \otimes EVM(D)$.

2. If P has no FEF then $P = Q +^* R$ and Q and R are quasi-disjoint. Then, by corollary 2, $EVM(P) = EVM(Q) \otimes EVM(R)$.

□

In the example in Figure 10, $D_0^0 = H_1^1 +^* H_1^2 -^* H_2^2$ and $EVM(D_0^0) = EVM(H_1^1) \otimes EVM(H_1^2) \otimes EVM(H_2^2)$.

6 Conclusions and Future Work

In this paper we have presented two contributions related with the EVM model.

The first one is a conversion algorithm from EVM to a B-Rep. The description of this algorithm can be understood as an informal proof of the completeness (non-ambiguity) of the EVM. This algorithm performs XOR operations between vertices and Boolean differences. The algorithm is robust because does not perform any floating-point operation: all the information that is not explicitly represented in the EVM is obtained by merging the extreme vertices coordinates.

The second contribution is the application of the Alternating Sum of Volumes decomposition to the particular type of OPP by taking advantage of the simplicity of the EVM. The method uses orthogonal hulls instead of convex hulls, replaces both Boolean difference and quasi-disjoint union by the XOR operation and uses a remedial process for non-convergence based on the potential of the EVM.

The restricted domain of the EVM reduces its applicability in the field of CAD. Applications of it are always directed to its use as an approximation of more complex solids. In Computer Aided Architectural Design (CADD) its applicability is bigger because a great number of modern apartment blocs are 3D objects belonging to the OP domain. We think that the applicability of our model can be more exploited in the field of digital images processing.

So, our future work is oriented to the study of the suitability of the EVM in the field of digital images processing. We have a few initial conclusions. A digital image is an OPP and thus it can be represented by the EVM. Recently a definition of a well-composed picture has appeared in the literature [19], [18], and pictures with this property behave better than those without it when performing the most common operations. In fact, well-composed pictures are manifold sets, i.e., OP. Therefore an open problem is to determine whether a picture is well-composed and, if it is not, to process it in order for it to become well-composed. Among the problems that appear in this field are how to improve operations such as thinning and boundary extraction. Several approaches for representing a digital picture have been published that take these operations into account. We only cite two papers among those that we have begun to study. Bieri has proposed bintrees in [8] and hyperimages in [9] and Udupa has proposed a model called a shell in [27]. In the immediate future we will compare the EVM with these and other representations and study the suitability of the EVM for the most demanding image operations.

7 Acknowledgments

This work has been partially supported by a CICYT grant TIC99-1230-C02-02. The authors are very grateful to the referees whose comments and suggestions have aid to greatly improve the paper.

References

[1] A. Aguilera. *Orthogonal Polyhedra: Study and Application.* PhD thesis, LSI-

Universitat Politècnica de Catalunya, 1998.

[2] A. Aguilera and D. Ayala. Orthogonal Polyhedra as Geometric Bounds in Constructive Solid Geometry. In C. Hoffmann and W. Bronsvort, editors, *ACM SM'97. Atlanta (USA)*, pages 56 – 67, 1997.

[3] A. Aguilera and D. Ayala. Domain extension for the extreme vertices model (EVM) and set-membership classification. In *CSG'98. Ammerdown (UK)*, pages 33 – 47. Information Geometers Ltd., 1998.

[4] A. Aguilera and D. Ayala. Solving point and plane vs. orthogonal polyhedra using the extreme vertices model (EVM). In V. Skala, editor, *WSCG'98. The Sixth Int. Conf. in Central Europe on Computer Graphics and Visualization'98. University of West Bohemia. Plzen (Czech Republic)*, pages 11 – 18, 1998.

[5] C. Andújar, D. Ayala, and P. Brunet. Validity-Preserving Simplification of Very Complex Polyhedral Solids. In M. Gervautz, A. Hildebrand, and D. Schmalstieg, editors, *Virtual Environments'99*. Eurographics, Springer, 1999.

[6] C. Andújar, D. Ayala, P. Brunet, R. Joan-Arinyo, and J. Solé. Automatic generation of multiresolution boundary representations. *Computer Graphics Forum*, 15(3), 1996.

[7] D. Ayala, C. Andújar, and P. Brunet. Automatic simplification of orthogonal polyhedra. In D. Fellner, editor, *Modeling, Virtual Worlds, Distributed Graphics: Proceedings of the International MVD'96 Workshop*. Infix, 1995.

[8] H. Bieri. Computing the Euler Characteristic and Related Additive Functionals of Digital Objects from Their Bintree Representation. *Computer Vision, Graphics and Image Processing*, 40:115 – 126, 1987.

[9] H. Bieri. Hyperimages - an alternative to the conventional digital images. In C. E. Vandoni and D. A. Duce, editors, *EUROGRAPHICS'90*, pages 341 – 352. Amsterdam: North-Holland, 1990.

[10] O. Bournez, O. Maler, and A. Pnueli. Orthogonal Polyhedra: Representation and Computation. In *Hybrid Systems: Computation and Control*, LNCS 1569, pages 46 – 60. Springer, 1999.

[11] T. Dang and O. Maler. Reachability Analysis via Face Lifting. In T. A. Henzinger and S. Sastry, editors, *Hybrid Systems: Computation and Control*, LNCS 1386, pages 96 – 109. Springer, 1998.

[12] C. M. Hoffmann. *Geometric and Solid Modeling*. Morgan Kauffmann Publishers, Inc., 1989.

[13] R. Juan-Arinyo. On Boundary to CSG and Extended Octrees to CSG conversions. In W. Strasser, editor, *Theory and Practice of Geometric Modeling*, pages 349–367. Springer-Verlag, 1989.

[14] R. Juan-Arinyo. Domain extension of isothetic polyhedra with minimal CSG representation. *Computer Graphics Forum*, 5:281 – 293, 1995.

[15] Y. S. Kim. Recognition of Form Features Using Convex Decomposition. *Computer-Aided Design*, 24(9):461 – 476, 1992.

[16] Y. S. Kim and D. Wilde. A Convergent Convex Decomposition of Polyhedral Objects. In *SIAM Conf. Geometric Design*, 1989.

[17] L.K. Kyprianou. *Shape Classification in Computer-Aided Design.* PhD thesis, Christ's College. University of Cambridge, 1980.

[18] L. Latecki. 3D Well-Composed Pictures. *Graphical Models and Image Processing*, 59(3):164 – 172, 1997.

[19] L. Latecki, U. Eckhardt, and A. Rosenfeld. Well-Composed Sets. *Computer Vision and Image Understanding*, 61(1):70 – 83, 1995.

[20] W. Lorensen and H. Cline. Marching cubes: A high resolution 3D surfaces construction algorithm. *Computer Graphics*, 21(4):163 – 169, 1987.

[21] M. J. Pratt. Towards Optimality in Automated Feature Recognition. In *Computing Suppl. 10*, pages 253 – 274. Springer-Verlag, 1995.

[22] F. P. Preparata and M. I. Shamos. *Computational Geometry: an Introduction.* Springer-Verlag, 1985.

[23] A. Requicha. Representations for rigid solids: Theory, methods, and systems. *Computing Surveys of the ACM*, 12:437–464, 1980.

[24] H. Samet. *The Design and Analysis of Spatial Data Structures.* Addison Wesley Publ., Reading, MA, 1989.

[25] S. N. Srihari. Representation of three-dimensional digital images. *ACM Computing Surveys*, 13(4):399 – 424, 1981.

[26] K. Tang and T. Woo. Algorithmic aspects of alternating sum of volumes. Part 1: Data structure and difference operation. *CAD*, 23(5):357 – 366, 1991.

[27] J. K. Udupa and D. Odhner. Shell Rendering. *IEEE Computer Graphics and Applications*, 13(6):58 – 67, 1993.

[28] D. L. Waco and Y. S. Kim. Geometric reasoning for machining features using convex decomposition. *CAD*, 26(6):477 – 489, 1994.

[29] T. Woo. Feature Extraction by Volume Decomposition. In *CAD/CAM Technology in Mechanical Engineering*, 1982.