Domain Extension
for the Extreme Vertices Model (EVM)
and Set Membership Classification

A. Aguilera
D. Ayala

Report LSI-97-63-R
Domain Extension for the Extreme Vertices Model (EVM) and Set Membership Classification

A. Aguilera
Universidad de las Américas-Puebla
Puebla, México
antonio@udlapvms.pue.udlap.mx
aguilera@goliat.upc.es

D. Ayala
Universitat Politècnica de Catalunya
Barcelona, Spain
ayala@lsi.upc.es

Abstract

In a previous work, orthogonal polyhedra were proposed as geometric bounds in CSG [2]. CSG primitives were approximated by orthogonal polyhedra, and the orthogonal bound of the object was obtained by applying the corresponding boolean algebra. Also, a specific model for orthogonal polyhedra was presented, the Extreme Vertices Model (EVM).

EVM allows simple and robust algorithms for performing the most usual demanding tasks such as closed and regularized boolean operations as presented in [2], and the remaining set membership classification algorithms as will be shown in this paper.

In this work we continue with this proposal in three directions. First, we extend the EVM domain in order to represent pseudo-manifold orthogonal polyhedra; then we discuss the formal properties of EVM, and finally we present and analyze set membership classification algorithms on EVM.

1 Introduction

Set membership classification [16] and, specifically, the boundary evaluation of a CSG tree, are problems of a certain complexity. Up to now, several accelerating techniques have been proposed to speed up geometric computations in CSG. These techniques include Active Zones [13], [4] the Extended Convex Differences Tree [10] and Approximating Shapes or Geometric Bounds [6], [5].
The most extensively used geometric bounds are the well-known bounding boxes, although other shapes such as spheres and convex hulls have also been proposed and studied [5].

In a previous work we have proposed orthogonal polyhedra (OP) as geometric bounds in CSG [2]. We have presented a specific model for OP, the Extreme Vertices Model (EVM) that only needs to store some of the OP vertices. We have also developed a simple and robust algorithm for closed and regularized boolean operations on EVM. Comparing OP with boxes as geometric bounds we have concluded that OP are more time consuming than boxes, not only when computing the geometric bound but also when performing classification tests on them; however, OP are tighter than boxes and so many tests will be avoided.

In this work we continue with this proposal in three directions. First, we extend the EVM domain in order to represent pseudo-manifold orthogonal polyhedra (OPP) in EVM. Then, we show that the previous results obtained for OP also apply for OPP. We prove that EVM is complete and study the remaining formal properties. Finally we present and analyze set membership classification algorithms on this model.

The paper is arranged as follows. The next section introduces some necessary concepts on orthogonal and pseudo-polyhedra and then analyzes the vertex classes on OPP. Section 3 introduces EVM and discusses its formal and informal properties as defined in [11]. Next section presents set membership classification algorithms. Section 5 brings more results about the suitability of OPP as geometric bounds, comparing them with simple boxes. Finally, conclusions and trends for future work are given in the last section.

2 Orthogonal Polyhedra and Pseudo-Polyhedra

2.1 Background

A pseudo-polyhedron $P$ can be defined as 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 [15]. A two-manifold edge is adjacent to exactly two faces and a two-manifold vertex is the apex of only one cone of faces. Conversely, a non-manifold edge is adjacent to more than two faces, and a non-manifold vertex is the apex of more than one cone of faces [12]. Polyhedra are two-manifold r-sets. Pseudo-polyhedra are r-sets with non-manifold boundary.

A polyhedron is said to be orthogonal (OP) if all of its edges and faces
are oriented in three orthogonal directions. An orthogonal pseudo-polyhedron (OPP) is a regular and orthogonal polyhedron with non-manifold boundary. This kind of objects can be obtained converting from an orthogonal spatial enumeration model (voxelization, octree, bintree,...) into a B-Rep model. In an OPP, a non-manifold edge is adjacent to exactly four faces and a non-manifold vertex is the apex of two cones of faces (see figure 1).

![Figure 1: a) an OP. b) An OPP. c) a non-regular OP.](image)

### 2.2 Vertex analysis for OPP

In an OP the number of incident edges for any vertex can be only three, four and six [7]. In this subsection we characterize vertices of OPP.

As an OPP can be understood as the resulting B-Rep of an orthogonal spatial enumeration model as a voxelization, we can characterize vertices of an OPP by studying vertices in a voxelization. Moreover, the classification of these vertices follows the same pattern as the classification of nodes in the marching cubes algorithm [8]. Considering the common vertex of eight octants, which can be full or void, there are $2^8 = 256$ combinations which, by applying rotational symmetries, may be grouped into 22 cases (configurations) [14] and, grouping complementaries, lead to the 14 basic patterns [8].

Figure 2 shows the 22 configurations (from $a$ to $v$). The 14 basic patterns are those from $a$ to $n$. Configurations from $o$ to $v$ are the complementaries of configurations $a$ to $h$. From the analysis of figure 2, vertices can be classified into eight types depending on the number of two-manifold and non-manifold edges incident to them. Figure 3 shows these eight classes of vertices and the results of an exhaustive analysis over them [3]. The last row of the table indicates whether this vertex is extreme or not (this concept will be defined in the next section).
3 The Extreme Vertices Model (EVM) for OPP

In this section, the theoretical basis for EVM is set up. We begin with several definitions and then we discuss the properties of EVM. All proofs had not been included due to the obvious space limitations; the reader can find them in [3].

3.1 Extreme Vertices

A brink (or extended-edge) is the longest uninterrupted segment, built out of a sequence of collinear and contiguous two-manifold edges of an OPP. Similarly, an extended-face is the maximal set of coplanar and contiguous faces, joined by vertices or non-manifold edges of an OPP.

Non-manifold edges do not belong to brinks. Every two-manifold edge belongs to a brink, whereas every brink consists of one or more edges and contains as many vertices as the number of edges plus one. Similarly, every
Figure 3: Vertex classification according to the number of incident edges (dashed lines represent non-manifold edges).

face belongs to an extended-face, and every extended-face consists of one or more faces (see figure 4).

Figure 4: a) An OPP with a brink having five edges and six vertices. b) One extended-face of the same OPP composed by five faces joined by vertices. Arrows represent normal vectors.

In a brink each ending vertex is V3, V4N1, or V6N1 and the remaining (interior) are V4, V4N2, V5N or V6. Vertices V6N2 do not belong to any brink. According to the above analysis, Vertices V3, V4N1, and V6N1 are the only ones that have exactly three incident two-manifold and linearly independent edges, regardless of the number of incident non-manifold edges; therefore these vertices mark the end of brinks in all three orthogonal directions. Vertices V4, V4N2, V5N or V6 correspond to the common point of two neighbor edges of a same brink, so they cannot be ending vertices of a brink. Finally all six incident edges of a V6N2 are non-manifold edges, so none of them belongs to a brink, and therefore this vertex does not belong to any brink, either.
We will call *Extreme Vertices (EV)* of an OPP to the ending vertices of all the OPP bricks, i.e., vertices V3, V4N1, and V6N1 of the OPP. Then, we define the *Extreme Vertices Model (EVM)* for OPP as a model that only stores all their EV. Finally, An *ABC-sorted EVM* is an EVM where its EV are sorted first by coordinate A, then by B, and then by C. An EVM can be sorted on six different ways: XYZ, XZY, YXZ, YZX, ZXY, and ZYX. From now on, \( EVM(P) \) will denote the ABC-sorted EVM of an OPP, \( P \), since most of the definitions and results (although not all of them) require of this ABC-ordering.

Although the EVM has been defined for 3D-OPP, it can also be defined for 2D-OPP and 1D-OPP [3].

**Theorem 3.1** Coordinate values of non-extreme vertices may be obtained from coordinates of EV.

This theorem can be informally proved as follows. Non-extreme vertices are in the intersection of two or three perpendicular bricks, so their coordinates are a subset of the coordinates of the extreme vertices defining these bricks. For a more formal proof see [3].

**Theorem 3.2** The number of EV of any OPP, \( nEV(P) \) is even.

The proof comes from the fact that \( EVM(P) \) is a sequence of bricks and a brink is defined by a couple of EV.

### 3.2 Planes (Lines) of Vertices and Sections

A *plane of vertices* of an OPP is the set of extended-faces lying on a plane perpendicular to a main axis. We will also refer as *line of vertices* (within a plane of vertices) to the set of bricks lying on a line parallel to a main axis. Both planes and lines of vertices for polyhedra (polygons) will be referred as *plv*. Moreover, \( plv_k(P) \) will refer to the \( k \)-th plane (line) of vertices of a polyhedron (polygon) \( P \).

A *Slice* is the OPP region between two consecutive planes (lines) of vertices. \( Slice_k(P) \) will denote the \( k \)-th slice of polyhedron (polygon) \( P \). Hence, \( P = \bigcup_k slice_k(P) \). A *section* is the 2D(1D)-OPP resulting from the intersection between a 3D(2D)-OPP and an orthogonal plane (line) perpendicular to a main axis which must not coincide with any plane (line) of vertices. \( S_k(P) \) will refer to the \( k \)-th section of \( P \) corresponding to \( Slice_k(P) \), between \( plv_k(P) \) and \( plv_{k+1}(P) \).
An OPP with \( np \) planes of vertices has \( np - 1 \) slices with their representing sections and two more empty sections, the initial and final sections. See figure 5.

![Diagram of OPP with planes and slices](image)

Figure 5: An OP, \( P \), with five planes of vertices (light regions) and four slices with their corresponding sections (dark regions). Initial and final empty sections \( S_0(P) \) and \( S_5(P) \) are also shown. Extreme vertices are numbered in the same way as they appear in the model EVM.

Planes of vertices and sections are 2D objects immersed in 3D space and we will need sometimes to refer to their projections. So, from now on, \( \overline{P} \) will denote the projection of a \((d-1)\)dimensional OPP, \( P \), which is immersed in \( E^d \), in a plane parallel to \( P \). In order to obtain such a projection we only need not to consider the first coordinate of all \( P \) vertices.

3.3 Relating sections and \( \text{plv} \)

This section shows how to compute OPP sections from their planes (lines) of vertices, as well as computing OPP planes (lines) of vertices from sections.

**Theorem 3.3** The projection of any section \( S_i(P) \) of a 3D (2D) OPP is computed by doing a regularized XOR \((\otimes^*)\) between the projection of its previous section \( S_{i-1}(P) \) and the projection of its previous plane (line) of vertices \( \text{plv}_i(P) \). Or, which is the same, by doing a regularized XOR among the projection of all the previous \( \text{plv} \).

\[
S_0(P) = \emptyset \text{ and } S_i(P) = S_{i-1}(P) \otimes^* \text{plv}_i(P), \forall i \in [1, np]
\]
or, which is the same,
\( S_0(P) = \emptyset \) and \( \overline{S_i}(P) = (\otimes^*)_{k=1}^{i} \overline{plv_k}(P), \forall i \in [1, np] \)

Corollary 3.1 The projection of a plane (line) of vertices, \( \overline{plv_i}(P) \), of an OPP \( P \) is computed by doing a regularized XOR between the projections of its previous \( \overline{S_{i-1}}(P) \) and next \( \overline{S_i}(P) \) sections.

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

It must be noted that \( \overline{S_{i-1}}(P) \) and \( \overline{S_i}(P) \) must be two consecutive sections of \( P \), but they (and thus, their projections) must not necessarily be different. Therefore, if \( \overline{S_{i-1}}(P) = \overline{S_i}(P) \) then, by corollary 3.1 \( \overline{plv_i}(P) = \emptyset \).
This means that any number of virtual planes (lines) of vertices may be considered wherever they are needed, without altering \( P \).

3.4 Forward and Backward Differences

From corollary 3.1,

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

Moreover, according to the definition of the \( \otimes \) operation,

\[
\overline{plv_i}(P) = \overline{S_{i-1}}(P) \otimes^* \overline{S_i}(P) = (\overline{S_{i-1}}(P) -^* \overline{S_i}(P)) \cup (\overline{S_i}(P) -^* \overline{S_{i-1}}(P))
\]

Then \((\overline{S_{i-1}}(P) -^* \overline{S_i}(P))\) and \((\overline{S_i}(P) -^* \overline{S_{i-1}}(P))\) will be called the Forward Difference and Backward Difference and, abbreviated, \( FD_i(P) \) and \( BD_i(P) \), respectively.

Theorem 3.4 In a 3D-OPP, forward differences \( FD_i(P) \) are the sets of faces on \( \overline{plv_i}(P) \) whose normal vectors point to the positive side of the main axis perpendicular to \( plv_i(P) \), while backward differences \( BD_i(P) \) are the sets of faces whose normal vectors point to the negative side.

Figure 6 shows the use of forward and backward differences in order to obtain the orientation of all the faces of an OPP. Also note that non extreme vertices (which do not appear in EVM) will show up as a result of using this differences.
3.5 Regularized boolean operations on EVM

Concerning the regularized XOR operation ($\otimes^*$), the following theorem provides an easy method to compute it.

**Theorem 3.5** Let $P$ and $Q$ be two $d$-dimensional OPP having $EVM(P)$ and $EVM(Q)$ as their respective models, then $EVM(P \otimes^* Q) = EVM(P \otimes EVM(Q))$.

where $\otimes^*$ denotes the regularized XOR.

This theorem is proved in [3] for 1D OPP and then, by induction, for 2D OPP and beyond.

Concerning the remaining operations, in [2] an algorithm which performs all regularized boolean operations on EVM is presented. This algorithm is recursive in the dimension. The trivial case performs 1D boolean operations. The recursive case merges the slices of both operands considering the plv of an operand as virtual plv for the other, and performs the boolean operation with the corresponding sections. The algorithm is based on theorems 3.3 and 3.5 and on corollary 3.1. Although input data (i.e., vertices coordinates) can be floating-point values, no time-consuming floating-point arithmetic is ever performed in this boolean operations algorithm because all results are obtained by simply classifying vertices coordinates of the initial data.

3.6 EVM properties

EVM has the following formal properties
Domain: The set of representable objects in EVM is the set of Orthogonal Pseudo-Polyhedra.

Validity: The following theorem states the validity condition:

**Theorem 3.6** Let \( Q \) be a finite point set in \( E^d \), then \( Q \) is a valid EVM for some pseudo-polyhedron \( P \), i.e., \( Q = EVM(P) \), if all points in \( Q \) are arranged in orthogonal lines and each of such lines holds an even number of points of \( Q \).

**Proof:** As EVM is a model for OP, points must be arranged in orthogonal lines. Furthermore, from theorems 3.3 and 3.5,

\[
EVM(S_{np}(P)) = \bigotimes_{k=1}^{k_{np}} EVM(plv_k(P))
\]

(\( np \) is the last section) and, as \( S_{np}(P) \) must be \( \emptyset \), then the number of points belonging to any line of \( Q \) must be even. ●

Completeness: EVM is complete as enunciate this theorem:

**Theorem 3.7** EVM is complete.

**Proof:** To prove this theorem we must prove that a B-Rep can be unambiguously obtained from EVM and so all the geometry, topology and correct orientation of the OPP boundary. Concerning with geometry, from theorem 3.1, all coordinates of non-extreme vertices appear as coordinates of EV and so they can be inferred from them. Concerning with orientation and topology, according to theorem 3.4 edges and faces can be extracted from brinks with their correct orientation by using respectively the 2D and 3D forward and backward differences.●

A conversion algorithm from EVM into a hierarchical B-Rep is presented in [1].

Uniqueness: The set of EV is unique for each OPP.

Concerning with informal properties, EVM is highly concise because only a subset of the OPP vertices needs to be stored. Creating an EVM is very easy following theorem 3.6. Finally, EVM is efficient in the context of applications: boolean operations [2] are simple and robust and so are other set membership classification operations as will be seen in the following section.
4 Set Membership classification in EVM

As the application in which we are interested is to use OPP as geometric bounds for CSG we are more interested in algorithms that test points, lines and planes against OPP than in a complete set membership classification of them. So we first present an algorithm for performing this tests and then we discuss some methods for complete set membership classification.

Before performing geometric tests, a preprocess must be executed which converts an EVM into an ordered sequence of disjoint boxes. This preprocess has linear complexity, and only needs to be executed once for any OPP. The ordered sequence of disjoint boxes is obtained with a two steps method (see figure 7):

1. splitting the OPP, $P$, at every internal plane of vertices perpendicular to an axis, thus obtaining an ordered sequence of slices (prisms), $P = \bigcup_{k=1}^{n_p-1} \text{slice}_k(P)$.

2. splitting each slice at every of its internal planes of vertices perpendicular to a different axis, thus obtaining an ordered sequence of disjoint boxes, $\text{slice}_k(P) = \bigcup_{j=1}^{j=n_p k-1} \text{Box}_{k,j}(P)$.

Therefore, the OPP, $P$, can be expressed as:

$$P = \bigcup_{k=1}^{n_p-1} \bigcup_{j=1}^{j=n_p k-1} \text{Box}_{k,j}(P)$$

Figure 7: a) Slices (axis X) and b) Boxes (axis Y) for the OPP in figure 5.

Every sorting of an EVM leads to the corresponding sorting of boxes; so there are six ordered sequences of disjoint boxes.
The boxes sequence is stored in an ordered and balanced binary tree. Each node of the tree represents a box (Box); nodes on its left correspond to boxes which appear before, and nodes on its right correspond to boxes which appear after in the sequence. Moreover, each node also keeps the minimal bounding box of all the boxes of the corresponding subtree ($MBB$).

The following algorithm performs any geometric test (point, line or plane) against an OPP represented by its ordered sequence of disjoint boxes. The boolean function $ElementBox$ must be tailored for the specific element (point, general line, general plane). The function returns TRUE if the element and the box interfere (cases IN and ON) and FALSE otherwise (case OUT).

```plaintext
FUNCTION ElementClass (x: Element, n: BoxesTreeNode) RETURN BOOLEAN

IF TerminalNode (n) THEN
    Return (ElementBox (x, n.Box))
ELSE
    IF ElementBox (x, n.MBB) THEN
        IF ElementBox (x, n.Box) THEN
            RETURN (TRUE)
        ELSE
            IF ElementClass (x, n.left) THEN
                RETURN (TRUE)
            ELSE
                RETURN (ElementClass (x, n.right))
            ENDIF
        ENDIF
    ELSE
        RETURN (false)
    ENDIF
ELSE
    RETURN (false)
ENDIF
ENDFUNCTION
```

This algorithm has a worst case complexity $O(log(n))$ if the element is a point. For general lines and planes is $O(n)$, because in the worst case a complete tree traversal must be performed. However, simulation results indicate that the average complexity is nearer to a logarithmic function than to a linear one.

A similar algorithm has been developed that performs a complete membership classification (IN, ON and OUT) for a point and a general line [1].
The algorithm for a point has to deal with several ON cases, but is still $O(log(n))$. For a general line, the algorithm is always $O(n)$ because all the boxes must be tested. Finally, we have not studied the classification of a general plane against an OPP because we cannot use the simplicity of OPP to devise an algorithm simpler than the general plane against polyhedron algorithm.

5 Suitability of OP as geometric bounds in CSG

In this section we compare OPP with simple boxes as geometric bounds. As said in the introduction, our approach is more time-consuming than boxes, not only when computing the geometric bound but also when performing geometric tests on them. However OPP are tighter than boxes and so many tests will be avoided.

The minimal containing box of the bounding OPP for a CSG is the smallest box for this CSG. We can also obtain this smallest box using boxes as geometric bounds but then the CSG must be expressed in its disjunctive form [9]. This result is due to the fact that boolean operations over boxes are a lattice. Boolean operations over OPP are, instead, a boolean algebra.

It is possible to choose either the OPP or the simple box before doing the classification tests. A good OPP will be one with a surface not much larger than the box surface and with a volume lesser enough than the box volume.

The volume of an OPP can be computed from the OPP expressed as a sequence of disjoint boxes and is the sum of the volume of all boxes. The volume can also be computed as the sum of the volumes of the slices using the following expression:

$$Volume = \sum_{k=1}^{np-1} Area(S_k) \cdot dist(plv_k, plv_{k+1})$$

The surface of an OPP can be computed from the EVM of the OPP using the following expression:

$$Area = \sum_{k=1}^{np} Area(plv_k) + \sum_{k=1}^{np-1} Perimeter(S_k) \cdot dist(plv_k, plv_{k+1})$$

Furthermore, the volume and surface of the OPP can be used as an approximation of the volume and surface of the CSG object.
6 Conclusions and Future Work

In this work we have extended the EVM domain in order to represent OPP. We have formalized the model discussing its formal properties. Finally we have presented a set membership classification algorithm on EVM and compared its suitability as geometric bounds for CSG with simple boxes.

In related works, we have devised a robust boolean operations algorithm [2] and conversion algorithms from EVM to B-Rep and to octrees and vice-versa [1]. So, as a future work we will intend to use OPP in some other applications as collision detection or feature recognition.

7 Acknowledgments

The present work has been partially supported by CICYT grants TIC-95-630-C03.

References


