
http://researchrepository.murdoch.edu.au/5629/

Copyright © 2011 IEEE

Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.
An Algorithm for Splitting an Orthogonal Polyhedron with an Orthogonal Polyplane

Jefri Marzal, Hong Xie, and Chun Che Fung
School of Information Technology
Murdoch University, Murdoch, WA 6150, Australia
30860445@student.murdoch.edu.au; H.Xie@murdoch.edu.au; l.fung@murdoch.edu.au

Abstract — Splitting of an orthogonal polyhedron is an essential operation in many orthogonal polyhedron related problems. This paper proposes an algorithm for splitting an orthogonal polyhedron with an orthogonal polyplane. There are four major steps of this algorithm: 1) sort the vertices of the orthogonal polyhedron in ABC-sorted order; 2) calculate the vertices at the intersection between the orthogonal polyhedron and the polyplane; 3) find the polyplane vertices, and 4) group the updated vertices into two groups of orthogonal polyhedra. Overall, the algorithm performs with O(n log n) time where n is the number of vertices of a given orthogonal polyhedron.

Keywords: orthogonal polyhedron, polyplane, ABC-sorted, intersection vertex.

I. INTRODUCTION

Orthogonal polyhedra are the 3D analogue of 2D orthogonal polygons. They are used in computational geometry as a well-known model to represent many real 3-dimensional objects.

Many different operations can be defined for an orthogonal polyhedron, for example: partitioning [1], splitting [2], and Boolean operations on arbitrary orthogonal polyhedra of any dimension [3] etc.

Splitting orthogonal polyhedron is an operation to split an orthogonal polyhedron into two halves. An orthogonal polyplane is a set of connected rectangles that fulfilled certain conditions. Splitting an orthogonal polyhedron with an orthogonal polyplane is a special of splitting operation aiming at separate an orthogonal polyhedron into two orthogonal polyhedra. This kind of operation is essential for many problems based on orthogonal polyhedra.

A rectangular can be used to split a given orthogonal polyhedron, and Ayala has shown a linear time algorithm to split an orthogonal polyhedron with a rectangular plane [2]. However, if a polyplane is used instead of a single rectangular plane, the splitting task is no longer simple anymore, and Ayala’s algorithm is not applicable. Therefore, a suitable algorithm for splitting operation that uses a polyplane is needed.

Let OP be an orthogonal polyhedron having n vertices and PP be an orthogonal polyplane that intersects OP, and we will consider those vertices and edges of PP lie in the interior of OP. We propose an algorithm for splitting OP with PP in four steps, namely sorting vertices in three different axis orders, calculating the intersecting vertices on the boundary between the orthogonal polyhedron and the polyplane, and also determining the vertices in polyplane, and finally grouping vertices into two sub orthogonal polyhedra.

II. DEFINITION AND TERMINOLOGY

A solid object can be modelled as a three dimensional polyhedron. Coxeter defined a polyhedron as a finite, connected set of plane polygons, such that every edge of each polygon belongs to just one other polygon, with the proviso that the polygons surrounding each vertex form a single circuit (to exclude anomalies such as two pyramids with a common apex). These polygons are called faces. The faces do not cross one another. Thus the polyhedron forms a single closed surface [4].

The most practical and important class of polyhedra are orthogonal polyhedra (sometimes also called isothetic polyhedra) whose boundary of all their faces oriented in one of the three orthogonal directions [5]. Each face has two basic types of components: vertices and edges. A vertex is a point on its boundary that is shared by at least three faces. An edge is a line segment on polyhedron’s boundary that connects two vertices. Each vertex may have different vertex degree which is a number of incident edges. Juan-Ariyio stated that vertex degree of a vertex on an orthogonal polyhedron can be three, four or six [6].

An orthogonal polyplane is a set of rectangles connected at their edges satisfying the following conditions: 1) each edge is shared by at most by two rectangles; 2) if an edge is shared by two rectangles, the two rectangles form a facial angle of either 0° or 90°; 3) if two rectangles share an edge and the angle between the two rectangles is 90° then the lengths of the shared edges on the two rectangles must be equal. This definition implies that each vertex has at most degree three. Each vertex on an orthogonal polyplane is called corner unless it is in the following situation. If two rectangles shared an edge and the angle between rectangles is 0° then the coincided vertices not considered as corner. A shared vertex is a corner of rectangle that belongs to more than one rectangle. For the rest of this paper, the term polyplane is used as a shorthand for the orthogonal polyplane defined above.

As an example, Fig. 1(a),1(b) and 1(c) satisfy the definition of polyplane, but Fig. 1(d) is not considered as a polyplane because the length of each rectangle at the shared edge is not equal. Fig. 1(e) is not a polyplane because it has four edges meet at a vertex v, as it has the potential to divide an orthogonal polyhedron into more than two orthogonal polyhedra. The vertex labelled c in the diagrams is a corner while the vertex labelled c’ is not a corner.
Fig. 2 shows an orthogonal polyhedron having a polyplane. Numbers 1, 2, ..., 28 in the figure are labels for intersection vertices in a split only exist on the edges of \( Pp \). The following figures show some valid and not valid polyplanes on orthogonal polyhedra.

From the above conditions, it can be concluded that the intersection vertices in a split only exist on the edges of orthogonal polyhedron. The following figures show some valid and not valid polyplanes on orthogonal polyhedra.

(i). The polyplane in fig. 3(a) consists of two rectangles, and all the corners of polyplane intersect (coincide) with the edges of \( OP \).

(ii) Fig. 3(b) has one rectangle as a splitting polyplane that intersects with four edges of the orthogonal polyhedron and creates four intersection vertices.

(iii) The polyplane in fig. 3(c) has two contiguous rectangles that intersect four edges, and the polyplane has two shared vertices.

(iv) The splitting plane in fig. 3(d) has three contiguous rectangle that intersect \( OP \) not on the edges; hence, it does not satisfy the rule of a polyplane intersect an orthogonal polyhedron.

### III. DESIGN OF ALGORITHM

The algorithm for splitting an orthogonal polyhedron by polyplane works in four steps:

1. Sort the vertices in XYZ-order, YZX-order, and ZXY-order, and put each of the sorted vertices in a file respectively.
2. Find intersection vertices that are corners of \( Pp \) that lie on the boundary of \( OP \).
3. Find shared vertices on the polyplane.
4. Group the vertices into two orthogonal polyhedra.

### A. Data structure for Orthogonal Polyhedron

Generally, there are two type of data structures in polyhedra representation: edge-based data structure and vertex based data structure [7]. Aquilera and Ayala represented an orthogonal polyhedra by using extreme vertices only [2]. This method requires less number of vertices compare with other methods that involved all vertices. A brief review of some common features of this method is given below.

As stated by Juan-Arinyo that the number of incident edges at any vertex on an orthogonal polyhedron can be three, four, and six. They are labelled as \( V3, V4 \) and \( V6 \) to represent the degree of the vertex. \( V3 \) means three edges meet at vertex the \( V \), and similar meanings exist for \( V4 \) and \( V6 \) vertices.

A brink is the longest uninterrupted line segment, built out of a sequence of collinear and contiguous two-manifold edges of an orthogonal polyhedron \( OP \). Every edge belongs to a brink and each brink has at least one edge. A brink may be \( V3, V4 \) and \( V6 \), but the two ending vertices of a brink is \( V3 \). A \( V4 \) and \( V6 \) may only be an intermediate vertex of a brink. The ending vertices of all brinks in \( OP \) are called extreme vertices. Extreme Vertices Model (EVM) is defined as a model that only stores all extreme vertices of an orthogonal polyhedron. The vertices of \( OP \) is accessed by ABC-Sorted EVM in which its extreme vertices are sorted first by coordinate \( A \), then by \( B \), and then by \( C \) [2]. The labels of vertices in Fig.2 reflect XYZ-sorted order.

In ABC-sorted EV model, vertices are arranged in a set of brinks. The \( k \)th C-brinks has \( v_{i} = v_{2k-1} \) as the beginning vertices and \( v_{e} = v_{2k} \) as the ending vertices, for \( k = 1, 2, ..., n \) be two consecutive vertices (C-brink refers to those brinks parallel to the C-axis).

### B. Intersection Vertices

A rectangle in a polyplane is formed by at least three points, and has general equation \( Ax + By + Cz + D = 0 \) [8]. Given three points \( s_0(x_1,y_1,z_1) \), \( s_2(x_2,y_2,z_2) \) and \( s_3(x_3,y_3,z_3) \) then the coefficients of the rectangle equation are formulated as follows:
Since each rectangle in an orthogonal polyplane is perpendicular to one of the three orthogonal axes, every rectangle equation has only one non-zero scalar. For example, \( x = d \) or \( y = d \) or \( z = d \).

A polyplane intersects the boundary of an orthogonal polyhedron at intersection vertices. The intersection vertices are found as following main steps; 1) read each rectangle of the polyplane, and 2) determine the intersection vertices between the rectangle and the orthogonal polyhedron. If an intersection point lies between two ending points of a brink then add two new vertices to \( OP \), and if the intersection point lies on one end of a brink then update the vertex on the point and the next vertex that lie on the extended splitting plane.

To develop the procedure of intersection vertices, the following primitive operations will be needed:

\[
\begin{align*}
A &= y_1(z_2-z_3) + y_2(z_1-z_3) + y_3(z_1-z_2) \\
B &= z_1(x_2-x_3) + z_2(x_1-x_3) + z_3(x_1-x_2) \\
C &= x_1(y_2-y_3) + x_2(y_3-y_1) + x_3(y_1-y_2) \\
D &= x_1(z_2-y_3) + x_2(z_3-y_1) + x_3(z_1-y_2)
\end{align*}
\]  

(1)

To get a shared vertex coordinate on a polyplane is simple. The \( x \)-coordinate is obtained from the rectangle that perpendicular to \( X \)-axis, and the other coordinates are obtained from the below boundary or the upper boundary of the rectangle of the polyplane. Example: a rectangle equation is \( x = s \), where \( y_1 \leq y < y_2 \) and \( z_1 \leq z \leq z_2 \); then the vertices on rectangle are \( p_1(s, y_1, z_1) \), \( p_2(s, y_1, z_2) \), \( p_3(s, y_2, z_1) \) and \( p_4(s, y_2, z_2) \). Exclude the intersection vertices from the polyplane, and if the intersection vertices between \( OP \) and \( Pp \). This procedure run with \( O(mn) \) time where \( m \) is the number of rectangles in a polyplane, and \( n \) is the number of extreme vertices of \( OP \).

**C. Shared Vertices on Polyplane**

To get a shared vertex coordinate on a polyplane is simple. The \( x \)-coordinate is obtained from the rectangle that perpendicular to \( X \)-axis, and the other coordinates are obtained from the below boundary or the upper boundary of the rectangle of the polyplane. Example: a rectangle equation is \( x = s \), where \( y_1 \leq y < y_2 \) and \( z_1 \leq z \leq z_2 \); then the vertices on rectangle are \( p_1(s, y_1, z_1) \), \( p_2(s, y_1, z_2) \), \( p_3(s, y_2, z_1) \) and \( p_4(s, y_2, z_2) \). Exclude the intersection vertices from the vertices of polyplane to get the shared vertices.

**D. Grouping Vertices**

After getting intersection vertices and shared vertices, the next task is grouping all vertices into two polyhedra. The rules are: first, except the intersection vertex, each vertex relate to three other vertices. Hence, each vertex has three moving possibilities, two moving forward, and one backward. An intersection vertex has only related vertex. Second, a vertex is visited in priority order, pair in a brink, pair in same plane, and pair in the next plane. The grouping OP is started from a brink that contains one or two intersection vertices.

The following is the steps for grouping vertices, and implement them in **GROUPING**:

1. Make a path starting from one of the first intersection vertex, \( v_0 \).
2. Decide what direction of the first edge is. If it is parallel to \( X \)-axis then go to \( YZX \)-sorted file to find the pair of \( v_0 \), or if it is parallel to \( Y \)-axis then go to \( YZX \)-sorted file to find the pair of \( v_0 \), otherwise go to \( XYZ \)-sorted file.
3. Let \( v_r \) be a pair of \( v_i \) and \( v_{j_1} \), has \( X \)-direction, then the next brink is obtained by moving to the vertex that has same rectangle with \( v_r \).
4. Move to the next rectangle if there is no more brink in the same rectangle.
5. Back to the previous vertex, and visit another brink that has not visited yet.
6. Stop moving if all intersection vertices have been visited.

IV. THE ALGORITHM IMPLEMENTATION

The following example illustrates the implementation of the algorithm. Fig. 6 is an orthogonal polyhedron having twelve vertices. A polyplane \( Pp \) has rectangular equations as follows:

\[
\begin{align*}
x &= 6, \quad 1 \leq y \leq 4, \quad 1 \leq z \leq 4 \quad \ldots \ldots \quad (1) \\
z &= 4, \quad 4 \leq x \leq 6, \quad 1 \leq y \leq 4 \quad \ldots \ldots \quad (2)
\end{align*}
\]

The following example illustrates the implementation of the algorithm. Fig. 6 is an orthogonal polyhedron having twelve vertices. A polyplane \( Pp \) has rectangular equations as follows:

\[
\begin{align*}
x &= 6, \quad 1 \leq y \leq 4, \quad 1 \leq z \leq 4 \quad \ldots \ldots \quad (1) \\
z &= 4, \quad 4 \leq x \leq 6, \quad 1 \leq y \leq 4 \quad \ldots \ldots \quad (2)
\end{align*}
\]

![Figure 6: The vertices of orthogonal polyhedron after splitting](image)

(1) By using Quick-sort method, sort all vertices in XYZ sorted, it means the vertices are sorted according to coordinate first, then by y-coordinate, and then by z-coordinate. So the point coordinates after XYZ sorted are \( v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9, v_{10}, v_{11}, v_{12} \), and save them in a file \( f_1 \).

With the similar method, sort all vertices in ZXY-sorted, so the point coordinates after ZXY sorted are \( v_1, v_4, v_6, v_7, v_8, v_{10}, v_{12}, v_1, v_{11}, v_9, v_2, v_3 \), and save them in a file \( f_2 \).

By using procedure \( \text{IntersectionOnEdgeParallelToZAxis} \) to get the intersection vertices, the time complexity is \( O(n) \), where \( n \) is less than \( m \).

Practically, \( m \) is less than \( n \).

V. DISCUSSION AND CONCLUSION

Let \( n \) be the number of vertices of a given orthogonal polyhedron, \( a \) is the number of rectangles in a polyplane, then the time cost is calculated for each of the following activities:

- ABC-sorted vertices: the time complexity is \( O(n \log n) \), it applies Quick-sort method [9].
- Calculating the intersection vertices: the time complexity is \( O(an) \), where \( a \) is smaller than \( n \).
- Calculating the shared vertices: the time complexity is \( O(mn) \), \( m \) is the number of planes that intersect one of any axis. Practically, \( m \) is less than \( n \).
- Each vertex is visited at most 3 times, then grouping vertices into two orthogonal polyhedra has \( O(3n) \) time.

For any sufficiently large orthogonal polyhedron, it is reasonable to assume that the number of rectangles in the polyplane, \( a \), is smaller than \( \log n \). For these orthogonal polyhedra, the time complexity for splitting them using an orthogonal polyplane consisting of \( a \) rectangles is \( O(n \log n) \). Using this algorithm, a smaller orthogonal polyhedron can be removed from a large orthogonal polyhedron. This procedure can be applied successively to an orthogonal polyhedron generate the required partitioning of the orthogonal polyhedron.

REFERENCES
