ORTHOGONAL SIMPLIFICATION OF OBJECTS REPRESENTED BY THE EXTREME VERTEX MODEL

Irving Cruz-Matías[1] and Dolors Ayala[1,2]
[1]*Universitat Politècnica de Catalunya, Barcelona, Spain*
[2]*Institut de Bioenginyeria de Catalunya, Barcelona, Spain*

Keywords: Simplification, LOD, Bounding Volumes, Orthogonal Polyhedra, Binary Volumes.

Abstract: This work presents a new approach to simplify objects. It is restricted to orthogonal pseudo-polyhedra (OPP) and binary volumes. The method produces a level-of-detail (LOD) sequence of OPP and it is incremental. Any object in this sequence is a bounding OPP of the previous objects. The sequence finishes with the minimum axis-aligned bounding box. OPP are represented by the Extreme Vertices Model. Simplification is achieved using a new approach called *merging faces* that relies on the application of 2D Boolean operations.

1 INTRODUCTION

The large size and complexity of models often affects the computation speed-up of their characteristics. Simplification techniques can diminish this problem. Moreover, in some cases it is advantageous to exchange an exact geometric representation of an object for an approximated one, which can be processed more efficiently. Bounding structures are a widely used specific typology of model simplification to accelerate tasks as collision detection, distance computation or any basic set membership operation.

We present a new approach to simplify objects. It is restricted to OPP and therefore it can be applied to binary volumes or 3D images represented by OPP. From an initial OPP, the method computes a LOD sequence of bounding volumes that are also OPP. The sequence ends with a convex OPP that is the axis-aligned minimum bounding box (AABB) of all the sequence. We will denote as BOPP (bounding OPP) such bounding structures. All the BOPP of sequence satisfy two important characteristics required for bounding structures: (1) any object contains the previous one and (2) all of them, as well as the original object, have the same AABB. In this work, OPP are represented by the Extreme Vertices Model (EVM), a complete model that stores a subset of their vertices (called extreme vertices) and performs fast Boolean operations. The simplification is achieved using a new approach called *merging faces* that relies on the application of 2D Boolean operations over the OPP faces, which are orthogonal pseudo-polygons.

2 RELATED WORK

There has been an extensive research on methods for model simplification, recent works are based on quadrilateral meshes (Tarini et al., 2010), as well as methods for LOD sequences of triangular and tetrahedral meshes (Ripolles et al., 2009). In contrast to these methods that rely on geometric operations, simplification can follow other strategies like morphological operators as filleting and rounding (Williams and Rossignac, 2007). Alternative representations can be used as octrees (Vanderhyde and Szymczak, 2008) in such a way that the geometry as well as the topology can be simplified, or BSP (Huang and Wang, 2010) obtaining a LOD sequence with a decreasing number of nodes. Simplification strategies have also been developed for B-Rep models (Sun et al., 2010), and for point clouds (Pauly et al., 2002).

In some cases it is desirable to compute an approximation that is also a bounding volume. Bounding volumes are used in many applications as collision detection, ray tracing (Wald et al., 2007), graphics interaction (Jiménez et al., 2011) and volume of interest computation (Fuchs et al., 2010). Works concerning bounding volumes use the most classical ones, as spheres (Gagvani and Silver, 2000) and AABB (Zachmann, 2002), or hybrid bounding volumes (Chang et al., 2010).

The so-called vertex representation (Esperança and Samet, 1998) is used to represent orthogonal polyhedra that are used as bounding structures and a 2D simplification method based on this model has

Cruz-Matías I. and Ayala D..
ORTHOGONAL SIMPLIFICATION OF OBJECTS REPRESENTED BY THE EXTREME VERTEX MODEL.
DOI: 10.5220/0003861301930196
In *Proceedings of the International Conference on Computer Graphics Theory and Applications* (GRAPP-2012), pages 193-196
ISBN: 978-989-8565-02-0

also been devised (Esperança and Samet, 1997). Our method has the similarity with this one that orthogonal faces are displaced, but our strategy and operations are completely new. We merge the faces geometry by applying Boolean operations. Moreover our approach deals with general 3D orthogonal objects with any number of shells, cavities and through holes.

3 EXTREME VERTICES MODEL

The EVM is a very concise representation scheme in which any OPP can be described using only a subset of its vertices: the extreme vertices (EV). The EVM can be created easily from the voxel model. Let Q be a finite set of points in E^3. The *ABC*-sorted set of Q is the set resulting from sorting Q according to A, then to B, and then to C-coordinate. Let P be an OPP: A *brink* is the maximal uninterrupted segment built out of a sequence of collinear and contiguous two-manifold edges of P and its ending vertices are the EV. A *cut* (C) of P is the set of EV lying on a plane perpendicular to a main axis of P. A *slice* is the prismatic region between two consecutive cuts, and a *section* (S) is its base polygon. Sections can be computed from cuts and vice versa. For $i = 1 \dots n$:

$$\overline{S_0(P)} = \overline{S_n(P)} = \emptyset, \quad \overline{S_i(P)} = \overline{S_{i-1}(P)} \otimes^* \overline{C_i(P)} \quad (1)$$

$$\overline{C_i(P)} = \overline{S_{i-1}(P)} \otimes^* \overline{S_i(P)} \quad (2)$$

where n is the number of cuts and $\otimes^*$ denotes the regularized xor operation. The overline symbolizes the project operator, that from a d-dimensional set of vertices lying on a plane produces a (d-1)-dimensional set of vertices. As the $\otimes^*$ operation can be expressed as union of differences, Equation 2 can be written as:

$$\overline{C_i(P)} = (\overline{S_{i-1}(P)} -^* \overline{S_i(P)}) \cup (\overline{S_i(P)} -^* \overline{S_{i-1}(P)}) \quad (3)$$

and any cut can be decomposed into its *forward difference (FD)* and *backward difference (BD)*:

$$\overline{FD_i(P)} = \overline{S_{i-1}(P)} -^* \overline{S_i(P)} \quad (4)$$
$$\overline{BD_i(P)} = \overline{S_i(P)} -^* \overline{S_{i-1}(P)}$$

$FD_i(P)$ is the set of $C_i(P)$ faces whose normal vector points to the positive side of the coordinate axis perpendicular to $C_i(P)$ and $BD_i(P)$ the opposite. Figure 2(b) shows an object with all its EV, cuts (C_1 to C_4) and sections (in yellow). Arrows on cuts indicate whether they are FD or BD.

There is a platform that uses the EVM for most volume operations, as it is efficient for Boolean operations. All Boolean operations are carried out by applying recursively the same operation over the 2D OPP sections. The base case performs this operation in 1D. The xor operation is the faster one, as it is a simple point-wise xor without section computation. For details concerning EVM see (Aguilera, 1998).

4 ALGORITHM OVERVIEW

Our method takes as input an EVM represented OPP. Let $B(P)$ be the BOPP of an OPP P, and o_0 be the initial object. A finite sequence $o_1, o_2, \dots, o_p$ of OPP is generated that follows the following properties:

1. *Subset property:* $B(o_i) = o_{i+1}$, $i = 0 \dots p-1$, and, therefore, $o_i \subseteq o_{i+1}$, $i = 0 \dots p-1$
2. $o_p = AABB(o_i)$, $i = 0 \dots p$

It is an incremental approach, i.e., we compute o_{i+1} from $o_i, i = 0 \dots p-1$. The main idea follows a *merging faces* strategy. Cuts in an *ABC*-sorted EVM are perpendicular to A-coordinate axis. If cut faces are displaced towards the contiguous cut and both sets of faces are merged, the total number of vertices (EV and non-EV) will never increase. So, the *simplification algorithm* receives an EVM P, which corresponds to the initial object o_0 and the desired maximum number of EV n in the simplified model, and returns an EVM Q corresponding to the BOPP o_k, which has no more than n EV. The method computes all the o_i between 1 and k, where the *merging faces* strategy applies a displacement of $d = i$ voxels to compute o_i. Therefore, to compute o_k the algorithm performs k iterations.

The *merging faces* process is applied to the current *ABC*-sorting of P that only coarsens the OPP in the A-coordinate, then it is necessary to repeat the process for the other two main directions. So, we can take any three sortings with different A-coordinate, as shown in the following pseudocode:

```
EVM Simplification(EVM P, int n){
  EVM Q = P;
  for(int d=1; Q->Get_nev() >= n; d++){
    Q->SetSorting(XYZ); Q=mergingFaces(Q,d);
    Q->SetSorting(YXZ); Q=mergingFaces(Q,d);
    Q->SetSorting(ZXY); Q=mergingFaces(Q,d);}
  return Q; }
```

Get_nev returns the number of EV of the corresponding object. The *mergingFaces* function receives an EVM object and a distance d, and returns another EVM object, where consecutive cuts with distance $<= d$ between them have been processed. When $n = 8$ the algorithm finishes with the AABB.

5 MERGING FACES APPROACH

Given the ABC-sorted EVM representation of an OPP, pairs of consecutive cuts along the A-coordinate axis are extracted and processed with a sequential scan. Let C_A and C_B be two consecutive cuts and FD_A, BD_A, FD_B and BD_B their corresponding forward and backward differences. A coarser OPP is achieved by displacing BD_B to the position of BD_A, and, similarly, displacing FD_A to the position of FD_B. Since the displacement is done in the direction of the respective normal vectors, and it is restricted by the cuts of the initial object, the model will never extend beyond these cuts. Thus, this process satisfies the *subset* property. So, the merging process performs the union of those faces with normal vector in the same direction that belong to two consecutive cuts, i.e.:

$$\overline{newC_A} = \overline{BD_A} \cup^* \overline{BD_B}, \quad \overline{newC_B} = \overline{FD_A} \cup^* \overline{FD_B} \tag{5}$$

where $newC_A$ and $newC_B$ are the pair of cuts that replace C_A and C_B respectively. Now, for objects with cavities, Equation 5 is not enough. For example, applying this equation to the object in Figure 1, the same OPP without any coarsening is obtained. In order to deal with this case, the void space must be detected with the expression:

$$vSpace(C_A, C_B) = \overline{FD_A} \cap^* \overline{BD_B} \tag{6}$$

as BD_A and BD_B are sets of faces whose normal vector points to the opposite direction of FD_A and FD_B, but only FD_A and BD_B contribute to form a void space between the cuts. If a void space is detected, it must be removed from $newC_A$ and $newC_B$ in order to close it. Then, combining Equation 5 with Equation 6:

$$\begin{aligned} \overline{newC_A} &= (\overline{BD_A} \cup^* \overline{BD_B}) -^* (\overline{FD_A} \cap^* \overline{BD_B}) \\ \overline{newC_B} &= (\overline{FD_A} \cup^* \overline{FD_B}) -^* (\overline{FD_A} \cap^* \overline{BD_B}) \end{aligned} \tag{7}$$

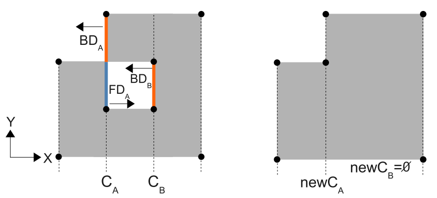

Figure 1: 2D example with a single hole. Left: faces to be displaced in order to close the hole. Right: resulting BOPP.

Figure 2 shows a 3D example where applying Equation 8, $newC_A = \emptyset$ and $newC_B$ gets a new value (see Figure 2(c)), which partially closes the hole. The result of this process is an object with a concavity (see Figure 2(d)), which will be closed in the next sweep.

As this last operation contributes just to close void spaces, it guaranties the subset property. Now, there are two EVM properties that make possible to perform unions and differences as simple point-wise xor operations (Aguilera, 1998).

- *P1:* Let P and Q be two OPP such that $P \cap Q = \emptyset$, having $evm(P)$ and $evm(Q)$ as their respective models, then $evm(P \cup Q) = evm(P) \otimes evm(Q)$.
- *P2:* Let P and Q be two OPP such that $P \supseteq Q$, with $evm(P)$ and $evm(Q)$ as their respective models, then $evm(P - Q) = evm(P) \otimes evm(Q)$.

According to these properties and as $\overline{FD_A} \cap^* \overline{FD_B} = \emptyset$, $\overline{BD_A} \cap^* \overline{BD_B} = \emptyset$, $vSpace \subseteq \overline{newC_A}$ and $vSpace \subseteq \overline{newC_B}$, Equation 7 can be rewritten in order to be faster to compute as:

$$\begin{aligned} \overline{newC_A} &= \overline{BD_A} \otimes^* \overline{BD_B} \otimes^* (\overline{FD_A} \cap \overline{BD_B}) \\ \overline{newC_B} &= \overline{FD_A} \otimes^* \overline{FD_B} \otimes^* (\overline{FD_A} \cap \overline{BD_B}) \end{aligned} \tag{8}$$

In Figure 1, $\overline{FD_A} = \overline{BD_B}$, $vSpace(C_A, C_B) = \overline{FD_A} = \overline{BD_B}$. Then, applying Equation 8, $newC_A = BD_A$, $newC_B = \emptyset$ and the hole is closed.

Observe the Figure 2(b), where if $C_A = C_1$ and $C_B = C_2$, $newC_A$ and $newC_B$ will be exactly the same as C_A and C_B respectively. This happens when $FD_A = BD_B = \emptyset$, which is the condition to establish that there are no faces to be displaced to get a coarser model. In these cases, C_1 is omitted and copied to the resulting model. On the other hand, the *merging faces* function process only those pairs of cuts with distance $<= d$. Formally, given the pair $(C_A = C_i, C_B = C_{i+1})$, if $FD_A = BD_B = \emptyset$, or $distance(C_A, C_B) > d$, then C_i is copied to the resulting BOPP, and the method continues with the pair $(C_A = C_{i+1}, C_B = C_{i+2})$. In the other case, the pair (C_A, C_B) is processed with Equation 8.

The *merging faces* process takes cuts in pairs: $(C_A = C_i, C_B = C_{i+1}), i = 1, i < n-1, i = i+2$, i.e. first the pair (C_1, C_2), then (C_3, C_4), and so on.

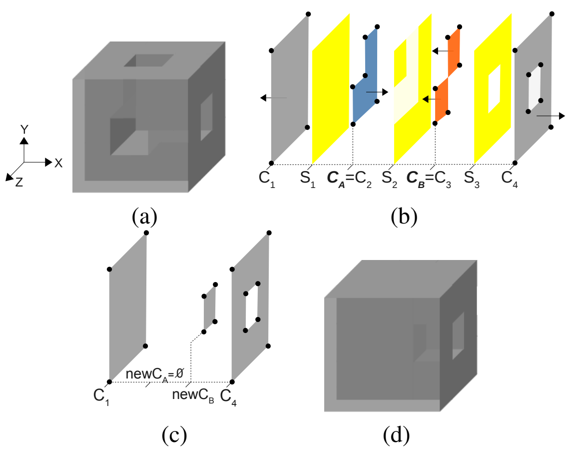

Figure 2: A 3D OPP with holes. (a) Original object. (b) EVM encoding (EV, cuts and sections), where cuts C_2 and C_3 are the sets of faces to be displaced in a first step. (c) Resulting cuts and (d) object after applying Equation 8.

6 EXPERIMENTAL RESULTS

We have tested the *merging faces* method in several cases. Table 1 shows a compilation of some models with their corresponding size, the maximum d in order to get the AABB and time to get it. Figure 3 depicts some results of simplifying two models, both depicted BOPP have less than 10% of EV of the original model. The algorithm was executed on a PC Intel(R) Pentium(R) CPU 3.20GHz with 3.2Gb of RAM.

Table 1: Information of some models used in the experiments. Size in voxels; d: max value to get AABB

Model	Size	d	Time (sec)
Binzilla	83x35x58	35	0.41
Camel	39x125x127	34	1.20
Venus	148x148x512	33	16.24
Athene	350x195x512	64	17.69

Figure 3: Some results. From left to right: Original (12,602 EV) and BOOP (1,148 EV) of Camel model, original (179,400 EV) and BOPP (15,100 EV) of Athene model.

7 CONCLUSIONS

We have described an approach for orthogonal simplification of models represented by the EVM, which is based on the application of efficient Boolean operations. Our approach deals with OPP with any number of holes and connected components, and computes a LOD sequence of BOPP. This sequence satisfies common properties of bounding structures. Directions for future work include the design of alternatives where individual faces of FD and BD are taken into account. We also plan to study a lossless simplification approach based on the presented one.

ACKNOWLEDGEMENTS

This work was partially supported by the national project TIN2008-02903 of the Spanish government. I. Cruz-Matías acknowledges the MAEC-AECID grant from the Spanish government.

REFERENCES

Aguilera, A. (1998). *Orthogonal Polyhedra: Study and Application*. PhD thesis, Univ. Politècnica de Catalunya.

Chang, J.-W., Wang, W., and Kim, M.-S. (2010). Efficient collision detection using a dual obb-sphere bounding volume hierarchy. *Com.-Aided Design*, 42(1):50 – 57.

Esperança, C. and Samet, H. (1997). Orthogonal polygons as bounding structures in filter-refine query processing strategies. In *Advances in Spatial Databases Fifth Int. Symposium*, pages 197–220. Springer Verlag.

Esperança, C. and Samet, H. (1998). Vertex representations and their applications in computer graphics. *The Visual Computer*, 14:240–256.

Fuchs, R., Welker, V., and Hornegger, J. (2010). Non-convex polyhedral volume of interest selection. *Journal of Computerized Medical Imaging and Graphics*, 34(2):105–113.

Gagvani, N. and Silver, D. (2000). Shape-based volumetric collision detection. In *VVS'00: of the 2000 IEEE symposium on Volume visualization*, pages 57–61, New York, NY, USA. ACM Press.

Huang, P. and Wang, C. (2010). Volume and complexity bounded simplification of solid model represented by binary space partition. In *Proc. - 14th ACM Symp. on Solid and Physical Modeling*, pages 177–182.

Jiménez, J. J., Feito, F. R., and Segura, R. J. (2011). Tetra-trees properties in graphic interaction. *Graphical Models*, 73(5):182 – 201.

Pauly, M., Gross, M., and Kobbelt, L. P. (2002). Efficient simplification of point-sampled surfaces. In *Proc. of the IEEE Visualization Conference*, pages 163–170.

Ripolles, O., Chover, M., Gumbau, J., Ramos, F., and Puig-Centelles, A. (2009). Rendering continuous level-of-detail meshes by masking strips. *Graphical Models*, 71(5):184–195.

Sun, R., Gao, S., and Zhao, W. (2010). An approach to b-rep model simplification based on region suppression. *Computers & Graphics*, 34(5):556 – 564.

Tarini, M., Pietroni, N., Cignoni, P., Panozzo, D., and Puppo, E. (2010). Practical quad mesh simplification. *Computer Graphics Forum (Special Issue of Eurographics 2010 Conference)*, 29(2):407–418.

Vanderhyde, J. and Szymczak, A. (2008). Topological simplification of isosurfaces in volume data using octrees. *Graphical Models*, 70:16 – 31.

Wald, I., Boulos, S., and Shirley, P. (2007). Ray tracing deformable scenes using dynamic bounding volume hierarchies. *ACM Trans. Graph.*, 26.

Williams, J. and Rossignac, J. (2007). Tightening: Morphological simplification. *International Journal of Computational Geometry & Applications*, 17(5):487–503.

Zachmann, G. (2002). Minimal hierarchical collision detection. In *Proceedings of the ACM symposium on Virtual reality software and technology*, VRST '02, pages 121–128, New York, NY, USA. ACM.