Layer-based Decompositions of Polyhedra

A. J. Rueda, F. Martinez and F. R. Feito
Escuela Politécnica Superior
Campus Universitario Las Lagunillas
23071 Jaén, Spain
{ajrueda|ffeito}@ujaen.es

ABSTRACT

This work describes a decomposition scheme for polyhedra called Layer-based decomposition. This decomposition can be computed in a efficient way for any kind of polyhedron, and has interesting applications in several geometric problems, like Boolean operation computation, point-in-polyhedron inclusion test, 3D location and ray-scene intersection computation.

Keywords

3D decompositions, Geometric Algorithms.

1. INTRODUCTION

Decomposition techniques are extensively used in the areas of Geometric Modelling, Computational Geometry and Computer Graphics as a useful tool in the description of objects and the design of simple algorithms for non-trivial problems. Triangulations, decomposition into trapezoids or convex subpolygons, and triangle or quad mesh generation are well-known decompositions in 2D [Ber00, Ber95, Goo97]. Tetrahedra decomposition and tetrahedra or hexahedra mesh generation are similar examples in 3D [Ber00, Ber95].

The layer-based decomposition of polygons was already presented in previous works [Fei99], describing several of its applications: Boolean operations computation, point-in-polygon inclusion test or location test [Riv00, Rue02]. Recently, this decomposition has been generalized to polyhedra [Rue02, Rue04], showing interesting properties and applications like 3D Boolean operations computation, point-in-polyhedron inclusion test, ray-scene intersection test or hidden surface removal.

Most part of this work describes the computation of the layer-based decomposition of polyhedra, illustrating this decomposition with a full example. In the last part we give some ideas beyond the decomposition of polyhedra into extended layers, which is currently an active area of research.

WSCG 2005 POSTERS proceedings, ISBN 80-903100-8-7
WSCG'2005, January 31-February 4, 2005
Plzen, Czech Republic.

2. DECOMPOSITION PROCESS

For the sake of simplicity we will only consider polyhedra consisted of triangular faces, although this is not a strong restriction as any polyhedron can be converted to this type just by triangulating its faces. We start by choosing an arbitrary point called *origin*. This point will determine the size and number of layers of the decomposition. From this point we make a full covering of the polyhedron by building tetrahedra between the origin and every triangular face of the polyhedron.

The layer-based decomposition of a polyhedron is based on the study of the *subordination relations* [Fei99, Riv00] between the tetrahedra from this covering:

Definition 1 *Given two tetrahedra* $s = \triangle OP_0P_1P_2$ *and* $t = \triangle OQ_0Q_1Q_2$*, we say* s is subordinated to t*, or* t dominates s *(*$s \lhd t$*), if* $s = t$ *or in the case* $s \neq t$ *there exists a point inside the face of* s *opposite to* O *which belongs to the interior of* t.

It is important to highlight that the subordination relation is only defined for tetrahedra determined by a common point origin and two non-intersecting triangular faces.

We use a primary data structure to store all the tetrahedra of the covering, and for each tetrahedron a secondary data structure storing references to every tetrahedra that dominates it. Building this data structure requires testing subordination between every two tetrahedra, with a $O(n^2 \log n)$ time cost if a binary search tree is used for the secondary data structure. The layer decomposition algorithm works as follows:

1. *Generate all the tetrahedra between the point origin O and the triangular faces of the polyhedron.*

2. *Build primary and secondary data structures computing subordination between the tetrahedra. Initialize layer counter i to 1.*

3. *Check the primary data structure for tetrahedra with no dominating tetrahedra (secondary data structure with no elements).*

4. *Add these tetrahedra to layer L_i.*

5. *Delete these tetrahedra from the primary data structure and all their occurrences in the second data structures of the rest of tetrahedra.*

6. *Increment i and return to step 3 until there are no tetrahedra in the primary data structure.*

The second phase of the decomposition algorithm also has a $O(n^2 \log n)$ time cost with an efficient implementation of the secondary data structure. The result is a set of layers $L(P,O) = \{L_1, L_2, ..., L_k\}$ which constitutes the layer-based decomposition of the polyhedron P respect to the point origin O. See [Fei99, Rue02, Rue04] for a thorough and more formal introduction to the concept of layer, its properties and decomposition algorithms.

3. APPLICATIONS

The layer-based decomposition has been successfully applied to several geometric problems:

- *Boolean operations computation*. Starting with the layer-based decomposition of two objects, it is possible to compute the result of a Boolean operation combining the tetrahedra of both decompositions following a different criterion depending on the operation [Riv00].

- *Point-in-polyhedron inclusion test*. This test is based on testing the point against each layer in descending order. Once the test succeeds for a given layer, the rest of tetrahedra can be discarded without testing [Rue02, Rue04].

- *Location test*. The point-in-polyhedron inclusion test can be generalized to a point location test if the layer-based decomposition is applied to a scene with several polyhedra [Rue02, Rue04].

- *Ray-scene intersection test*. A common layer-based decomposition of the polyhedra from a scene taking the observer as origin can be useful for determining the polyhedron intersected by a ray starting from the observer [Rue04].

- *Hidden surface removal*. The previous layer structure can also be used for the implementation of the Painter's algorithm. Rendering the tetrahedra of each layer following an ascending order gives a correct final visual result of the scene from the point of view of an observer placed at the origin of the layer decomposition [Rue04].

4. ACKNOWLEDGEMENTS

This work has been partially granted by the Ministry of Science and Technology of Spain and the European Union by means of the ERDF funds, under the research projects TIC-2001-2099-C03-03 and TIN-2004-06326-C03-03.

6. REFERENCES

[Ber95] Bern, M., Eppstein, D. Mesh generation and optimal triangulation. *Computing in Euclidean Geometry 2nd ed.*, pp. 189-196, World Scientific, 1995.

[Ber00] Bern, M., Plassmann, P. Mesh generation. *Handbook of Computational Geometry*, pp. 291-332, North Holland, 2000.

[Fei99] Feito, F. R., Rivero, M. L., Rueda, A. J. Boolean representation of general planar polygons. *Proc. WSCG'99*, pp. 87-92, 1999.

[Goo97] Goodman, J. E., O'Rourke, J. (eds.) *Handbook of Discrete and Computational Geometry*, CRC Press, 1997.

[Riv00] Rivero, M. L., Feito, F. R. Boolean operations on general planar polygons. *Computer & Graphics*, 24, pp. 881-896, 2000.

[Rue02] Rueda, A. J., Feito, F. R., Rivero, M. L. A triangle-based representation for polygons and its applications, *Computer & Graphics*, 26, pp. 805-814, 2002.

[Rue04] Rueda, A. J. *Representación de Objetos Gráficos mediante Capas y sus Aplicaciones*. PhD Thesis, Departamento de Lenguajes y Ciencias de la Computación, Universidad de Málaga, 2004 (in spanish).