An exact hierarchical geometric model. Combining remeshing and spatial decomposition

A. Martinez [1], J. Jimenez[1], F. Paulano[1], R. Pulido[1] and F. Feito[1]
[1]Departamento de Informatica. Universidad de Jaen

ABSTRACT

The use of hierarchical spatial decomposition in 3D scenes in order to manage the complexity of the objects is a well known approach. The main problem with this technique is the updating process when the mesh is modified. Any deformation or rotation means a new complete reconstruction of the structure. By other way remeshing techniques modify the structure of a mesh in order to achieve a given quality requirement. In this study a combination of remeshing techniques and hierarchical spatial decomposition is presented. Our goal is to develop an new model applying a remeshing process based on the hierarchical structure elements. This new model allows to extend one deformation in the spatial decomposition to the mesh. The tetra-tree is chosen as the spatial decomposition because of its advantages in relation to the remeshing algorithm. Tests with medium meshes with the new model were performed with good results.

Keywords: Normal orientation, mesh repair, visibility, patch connectivity, CAD tools.

1 INTRODUCTION

Nowadays the visualization of complex scenes is common in interactive environments. The complexity of the objects inside the 3D interactive-scene and the time requirements (real-time) force us to develop simplification techniques to fulfill the requirements without decreasing the detail. Spatial decomposition is one of these approaches that solves this problem [15]. One advantage of these structures is that can be implemented hierarchically, so its complexity can be adapted to different environments. The main disadvantage is the updating process that is forced by any modification (deformation) of the original mesh. This problem appears because the relation between triangles and nodes of the spatial decomposition is not unique, so one triangle could belong to more than one node.

On the other hand remeshing techniques are common in many areas of computer graphics areas such as: surface sampling [3], surface parametrization [14], remeshing irregular geometry [1], improving mesh quality [16] and mesh approximation [6]. These techniques modify the triangles of the mesh in order to fulfill some quality requirement where the complexity is often constrained. In our paper a combination between remeshing techniques and spatial decomposition is proposed. We call *exact model* if each triangle of the mesh belongs to only one node, on otherwise is an *non-exact model*. Our goal is adapting the mesh to the nodes of the spatial decomposition in order to develop an exact geometric modeling. Previous work [5, 10, 13] uses spatial decomposition to define the way to generate new meshes. This remeshing transforms the triangles mesh inside the nodes into new nodes of the structure. In this paper a novel hierarchical spatial decomposition based on the work by Jimenez [7] is proposed, but with some modifications in order to overcome the limitations of the classical model and for developing an exact geometric model.

2 REMESHING

In many applications, a remeshing procedure is necessary to increase the level of detail, generate non-homogeneous triangulation or improve triangulated meshes [5, 10, 13]. Besides needing to reduce the complexity of the meshes, the mesh quality must be improved. Other remeshing techniques focus on compatible refinement, hierarchical simplicial trees and the definition of the maximum factor of mesh growth [4]. Alliez et al. [2] performed a complete survey of remeshing techniques, defining remeshing as: *"Given a 3D mesh, compute another mesh, whose elements satisfy some quality requirements, while approximating the input acceptably"*. In this paper, a remeshing procedure is applied to adjust the original mesh to a hierarchical spatial decomposition, so our quality requirements are based on the nodes of the hierarchical spatial decomposition. The remeshing is localized because the procedure is not applied to the whole mesh but only to a subset of triangles which do not belong completely to a node of the hierarchical decomposition. This remeshing procedure could be considered as a high quality remeshing, taking into consideration the fact that the mesh is subdivided in order to obtain a new mesh with more triangles than the original one. If in previous applications this approach is focused on adapting the mesh to some properties of the mesh such as curvature [1], in our work the mesh is adapted to the hierarchical spatial decomposition.

3 COMBINING HIERARCHICAL SPATIAL STRUCTURES AND REMESHING TECHNIQUES

The nodes of the tetra-tree are tetra-cones that are tetrahedrons with their base at infinity. So the problem of intersecting nodes and shared triangles is reduced to a triple intersection triangle and lateral tetra-cone face, that is a triangle - triangle 3D intersection. The triangle - triangle intersection is easily reduced to a segmented/ray - triangle intersection where each ray is an edge of the triangle and the triangle is one of the lateral faces of the tetra-cone. The ray - triangle intersection is an important problem in many computer graphic areas. The classical algorithm used for the ray - triangle intersection is the one proposed by Möller [11]. The main problem of this approach is how it deals with limit cases which forces us to trace new rays. A review of ray-triangle intersection algorithms is shown in [9]. To overcome the problem of the limit cases we use a robust point in polygon test based on barycentric coordinates [8] and the intersection algorithm proposed by the same author [9] where the limit cases are found directly during the study of the value of barycentric coordinates. In this scheme the use of tetra-tree is a major advantage because, the classification method for the hierarchical structure, the point in polygon test and the ray-intersection algorithm are all based on barycentric coordinates so some calculations may be reused. After the combined inclusion and intersection procedure, all the possible cases of triangle - triangle intersection are studied and one of the splitting patterns is chosen (see figure 2). The splitting patterns are selected to be as simple and efficient as possible, attempting to reduce the number of triangles generated and being invariant to the edges intersected. In this paper this scheme has been applied in a general form without any precalculation during the building of the spatial structure, so this approach could be applied to any spatial structure that is composed or reduced using tetrahedrons or tetra-cones.

The special case are divided into rejected special cases, which are directly removed from the remeshing procedure, and degenerated special cases where the remeshing is applied but a test to remove degenerated triangles is performed as well. In general geometric problems the special cases are not usual or are very rare [12], but in our case a previous splitting process often generates many special cases. If these cases are not explicitly controlled the splitting process could have no ending. The Jimenez algorithm deals with the limit cases naturally using the barycentric coordinates, so the special cases are handled directly with no additional operations. If the mesh is required to be compatible some post-processing algorithm could be applied [4].

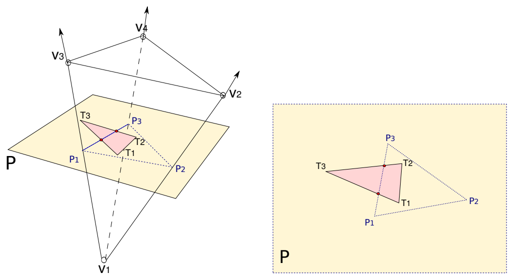


Figure 1: The triangle - tetra-cone intersection can be easily expressed as a triangle-triangle intersection using the plane that contains the triangle.

3.1 Cases of intersection between two triangles and subdivision patterns

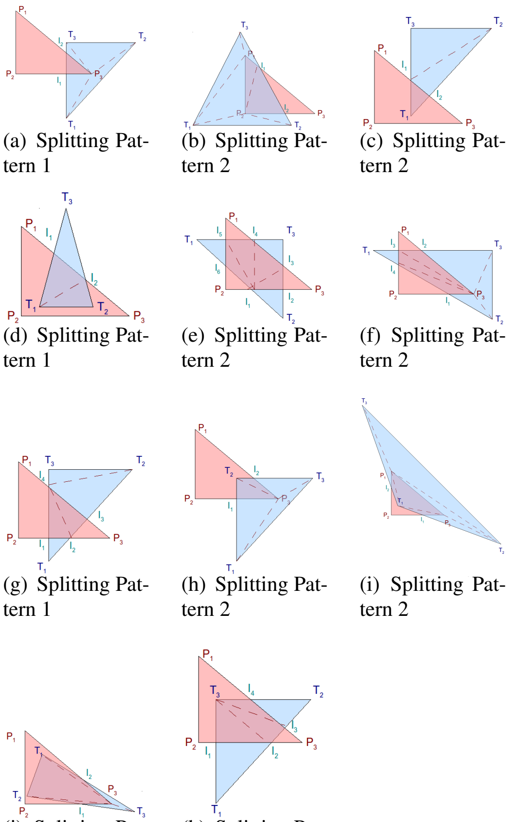
There are 11 cases of intersection between two triangles (see figure 2). In these images we consider the blue triangle as the triangle of the mesh to split and the red one as the triangle of the intersection of the plane that contains the triangle and the tetra-cone (see figure 1). Taking into consideration the assumption that the tetra-cone is larger than the triangle to be classified, the most probable subdivision cases are (sorted by decreasing probability): case 3, case 4, case 1, case 8, case 7 and case 11. Cases 2, 5, 6 and 9 are plausible theoretically but did not appear on our tests [see section 4]. So most of the triangles intersect only one face of the tetra-cones and have two intersection points. The tests performed confirm this probability distribution. Once the intersection case is defined, a subdivision pattern is applied. These patterns have been defined attempting to reduce the number of degenerated triangles. An additional test to reject degenerated triangles is also used.

4 EXPERIMENTAL RESULTS

The new model has been tested over medium-size meshes, similar to those most usually used in computer graphics and common applications. The first mesh has 40000 triangles and 20002 vertices. Table (4) shows how the size of the mesh increases according to tetra-tree levels. The amount of new primitives increases geometrically in relation to the number of tetra-cones which increases exponentially. The distribution of the cases where the subdivision is applied is shown in table (4) too.

The mesh before the remeshing procedure and after is shown on figure [5], and the green triangles are the shared triangles found on the mesh.

More tests are summarized on tables 6, 7 and 8.



(a) Splitting Pattern 1 (b) Splitting Pattern 2 (c) Splitting Pattern 2

(d) Splitting Pattern 1 (e) Splitting Pattern 2 (f) Splitting Pattern 2

(g) Splitting Pattern 1 (h) Splitting Pattern 2 (i) Splitting Pattern 2

(j) Splitting Pattern 1 (k) Splitting Pattern 2

Figure 2: Intersection cases and splitting patterns.

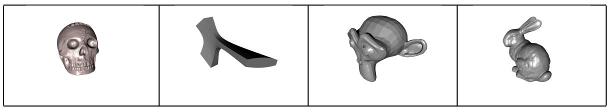


Figure 3: Medium-size mesh used.

TT Depth	TT Generated	Final triangles	Final points
1	24	47232	27334
2	120	53599	33610
3	504	64967	44978
4	2040	82757	62768

Figure 4: Evolution of the number of triangles and points when subdivision is applied. Distribution of the intersection cases. C is the intersection case number, and TTD is the tetra-tree depth.

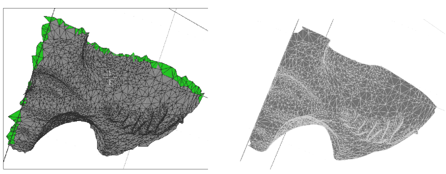


Figure 5: Before (left) and after (right) the remeshing algorithm, the shared triangles have been removed. In green, the shared triangles from the original mesh

TT Depth	Final triangles	Final points
1	2648	1954
2	3533	2834
3	4413	3707
4	4803	4097

Figure 6: Mesh, new size of the mesh and cases. The original mesh has 1854 triangles and 1172 points. TTD is the tetra-tree depth

TT Depth	Final triangles	Final points
1	3131	2072
2	4044	2978
3	4634	3566
4	5462	4389

Figure 7: Mesh, new size of the mesh and cases. The original mesh has 2180 triangles and 1132 points. TTD is the tetra-tree depth

TT Depth	Final triangles	Final points
1	74173	40667
2	84008	50501
3	93907	60403
4	127036	93565

Figure 8: Mesh, new size of the mesh and cases. The original mesh has 69459 triangles and 35947 points. TTD is the tetra-tree depth

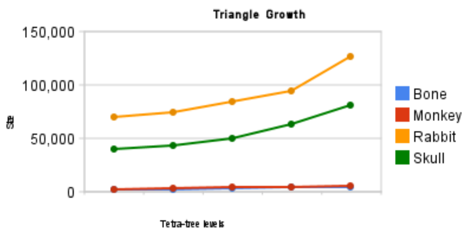


Figure 9: Evolution of the mesh size after the remeshing procedure in relation with the tetra-tree level.

5 CONCLUSIONS AND FUTURE RESEARCH

A new hierarchical spatial decomposition for dealing with complex objects has been presented. This new approach is based on regular spatial decomposition performed by tetra-trees. In order to achieve this goal, a local remeshing method is applied. This method transforms the relation between triangle and tetra-cone into a direct relationship, increasing the size of the mesh. The splitting process adds triangles to the original mesh, but this mesh is simplified with the hierarchical structure. So we increase the time for tetra-tree building (only performed at the beginning), attempting to avoid an updating process on the interaction environment.

Various tests with real meshes were performed to compare the original meshes with the remeshed ones. Regarding the main disadvantage of our model, the computational cost of the building process, some preliminary work for translating the building process to the GPU are in progress. The geometric operations of the split method are easily implemented in GPU as well so the whole process could be performed in the GPU thus reducing the associated computational cost. With regard to the geometrical operations applied to hierarchical decomposition some ideas are under development, such us: a fast way to locate nodes and determine the concavities and holes in the mesh, definition of depth levels to classify the triangles inside the nodes, and a space-filling index to perform transversal operations on the tree.

6 ACKNOWLEDGEMENTS

This work has been partially supported by the Spanish Ministry of Education and Science and the European Union (via ERDF funds) through the research project TIN2007-67474-C03-03, by the Consejería de Innovación, Ciencia y Empresa of the Junta de Andalucía through the research projects P06-TIC-01403 and P07-TIC-02773, and by the University of Jaén through the research project UJA-08-16-02, sponsored by Caja Rural de Jaén. The rabbit mesh comes from The Stanford 3D Scanning Repository, the suzane mesh (monkey) comes from Blender and the other meshes are of free distribution.

REFERENCES

[1] Pierre Alliez, Mark Meyer, and Mathieu Desbrun. Interactive geometry remeshing. *ACM Trans. Graph.*, 21(3):347–354, 2002.

[2] Pierre Alliez, Giuliana Ucelli, Craig Gotsman, and Marco Attene. *Shape Analysis and Structuring*, chapter Recent Advances in Remeshing of Surfaces, pages 53–82. Mathematics and Visualization. Springer Berlin Heidelberg, 2008.

[3] Pierre Alliez, Éric Colin de Verdière, Olivier Devillers, and Martin Isenburg. Isotropic surface remeshing. In *SMI '03: Proceedings of the Shape Modeling International 2003*, page 49, Washington, DC, USA, 2003. IEEE Computer Society.

[4] F. Betul Atalay and David M. Mount. The cost of compatible refinement of simplex decomposition trees. In *IMR*, pages 57–69, 2006.

[5] Vincent Francois, Jean Christophe Cuilliere, and Michel Gueury. Automatic meshing and remeshing in the simultaneous engineering context. *Research in Engineering Design*, 11:55 – 66, 1999.

[6] Pascal J. Frey. About surface remeshing, 2000.

[7] J. Jimenez, F. Feito, R. Segura, and C. Ogayar. Particle oriented collision detection using simplicial coverings and tetra-trees. *Computer Graphics Forum*, 25(1):53 – 68, 2006.

[8] Juan Jose Jimenez, Francisco R. Feito, and Rafael Jesus Segura. Robust and optimized algorithms for the point-in-polygon inclusion test without pre-processing. *Comput. Graph. Forum*, 28(8):2264–2274, 2009.

[9] Juan José Jiménez, Rafael Segura, and Francisco Feito. A robust segment/triangle intersection algorithm for interference tests. eficiency study. *Computational Geometry: Theory and Applications*, 43(1):474–492, 2010.

[10] Dae-Young Kwak and Yong-Taek Im. Hexahedral mesh generation for remeshing in three-dimensional metal forming analyses. *Journal of Materials Processing Technology*, 138(1-3):531 – 537, 2003.

[11] Tomas Möller and Ben Trumbore. Fast, minimum storage ray/triangle intersection. *SIGGRAPH '05: ACM SIGGRAPH 2005 Courses*, page 7, 2005.

[12] Carlos Ogayar, Rafael Segura, and Francisco Feito. Point in solid strategies. *Computers & Graphics*, 26(1):616–624, 2005.

[13] Juan J. Pombo, Jose C. Cabaleiro, and Tomas F. Pena. Parallel complete remeshing for adaptive schemes. *Int. J. Comput. Sci. Eng.*, 1(2-4):207–215, 2005.

[14] Emil Praun. Spherical parametrization and remeshing. *ACM Trans. Graph*, 22:340–349, 2003.

[15] Hanan Samet. *Foundations of Multidimensional and Metric Data Structures (The Morgan Kaufmann Series in Computer Graphics and Geometric Modeling)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2005.

[16] Vitaly Surazhsky and Craig Gotsman. Explicit surface remeshing. In *SGP '03: Proceedings of the 2003 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pages 20–30. Eurographics Association, 2003.