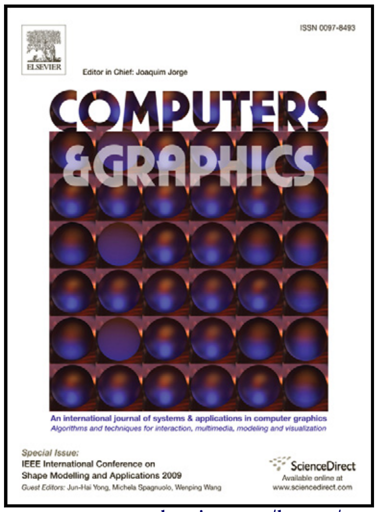
Author's Accepted Manuscript



www.elsevier.com/locate/cag

Simplified and Tessellated Mesh for Realtime High Quality Rendering

Yazhen Yuan, Rui Wang, Jin Huang, Yanming Jia, Hujun Bao

PII: S0097-8493(15)00114-4
DOI: http://dx.doi.org/10.1016/j.cag.2015.07.011
Reference: CAG2623

To appear in: *Computers & Graphics*

Received date: 12 April 2015
Revised date: 11 July 2015
Accepted date: 12 July 2015

Cite this article as: Yazhen Yuan, Rui Wang, Jin Huang, Yanming Jia, Hujun Bao, Simplified and Tessellated Mesh for Realtime High Quality Rendering, *Computers & Graphics,* http://dx.doi.org/10.1016/j.cag.2015.07.011

Simplified and Tessellated Mesh for Realtime High Quality Rendering

Yazhen Yuan, Rui Wang[1], Jin Huang, Yanming Jia, Hujun Bao

State Key Lab of CAD&CG, Zhejiang University, China

Abstract

Many applications require manipulation and visualization of complex and highly detailed models at realtime. In this paper, we present a new mesh process and rendering method for realtime high quality rendering. The basic idea is to send a simplified mesh to hardware pipeline, while use the online tessellation on the GPU to facilitate the rendering of complex geometric details. We formulate it into an inverse tessellation problem that first computes the simplified mesh, and then optimizes the tessellated mesh with geometric details to approximate the original mesh. To solve this problem, we propose a two-stage algorithm. In the first stage, we employ an iterative surface simplification technique, where we take the requirement of hardware tessellation into consideration to obtain an optimal simplified mesh. In the second stage, to better utilize the hardware tessellation, we propose a moving vertex strategy to approximate the tessellated mesh to the original mesh. Results show that our method achieves 2-4 times faster at rendering but still retains high quality geometrical details.

Keywords: hardware tessellation, mesh simplification, mesh rendering and visualization

1. Introduction

Realtime rendering of complex, highly detailed model is of great interest in variety of performance demanding applications, such as game, visualization, virtual reality, etc. However, due to the bottleneck of I/O, it becomes a popular strategy to highly simplify the complex model so as to achieve desired frame rate. But, with the simplification, the rendering quality significantly degenerates in general.

As a result of recent advances in graphics hardware, large number of geometry primitives can now be efficiently and flexibly generated online with the highly parallel GPU tessellation units. Technically, to fully utilize the power of the GPU tessellation units, it requires a two-layer representation of model. A coarse model to be sent from CPU to GPU, and a fine model that are tessellated at rendering. However, given a complex model, it is challenge to decompose the input model into such a two-layer representation. First, the coarse model provides a base to be rendered and tessellated at runtime. The hardware tessellation performance and quality depend on this coarse representation. Thus, how to get the best coarse model for hardware tessellation is a problem. Second, the hardware tessellation requires tessellation parameters and vertex data to recreate the details of original model. How to obtain optimal parameters and vertex data to approximate the original mesh is another problem.

Inverse subdivision techniques [1, 2, 3] can be regarded as potential solutions for these two problems. They decomposed the connectivity of original mesh into a coarse representation for subdivision. Guskov et al. [4] proposed normal meshes to compress the storage of meshes by constructing a multiresolution mesh with normal offsets. Cook [5] introduced the ideal of displacing a surface by a function. Lee et al. [6] introduced the displacement maps as an inverse subdivision process so that the original mesh can be approximated by displaced subdivision surface. However, even with the recent progress to carry out realtime subdivision using hardware tessellations [7], these inverse approximation approaches [4, 6] still suffer from two main limitations. First, in computing the coarse representation, these methods do not consider the hardware tessellation stage, therefore the coarse representation is not optimal for hardware tessellation. Second, no matter the normal offset or the displacement map, these methods only move vertices along normals, which may fail at capturing some geometric features of original mesh, especially in case of using a small number of tessellated surfaces.

To address these limitations, in this paper, we propose a new solution to find optimal decomposition for the