FLEXRENDER: A DISTRIBUTED RENDERING ARCHITECTURE FOR RAY TRACING HUGE SCENES ON COMMODITY HARDWARE.

A Thesis

Presented to

the Faculty of California Polytechnic State University

San Luis Obispo

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in Computer Science

by

Robert Edward Somers

June 2012

COMMITTEE MEMBERSHIP

TITLE:	FlexRender: A distributed rendering architecture for ray tracing huge scenes on commodity hardware.
AUTHOR:	Robert Edward Somers
DATE SUBMITTED:	June 2012
COMMITTEE CHAIR:	Zoë Wood, Ph.D.
COMMITTEE MEMBER:	Chris Lupo, Ph.D.
COMMITTEE MEMBER:	Phillip Nico, Ph.D.

Abstract

FlexRender: A distributed rendering architecture for ray tracing huge scenes on commodity hardware.

Robert Edward Somers

As the quest for more realistic computer graphics marches steadily on, the demand for rich and detailed imagery is greater than ever. Unfortunately, our appetite for large and complex geometry is quickly outpacing advances in the hardware used to render it. Scenes with hundreds of millions or even billions of polygons are not only desired, they are demanded.

Techniques such as normal mapping and level of detail have attempted to address the problem by reducing the amount of geometry in a scene. This is problematic for applications that desire or demand access to the scene's full geometric complexity at render time. More recently, out-of-core techniques have provided methods for rendering large scenes when the working set is larger than the available system memory.

We propose a distributed rendering architecture based on message-passing that is designed to partition scene geometry across a cluster of commodity machines in a spatially coherent way, allowing the entire scene to remain in-core and enabling the construction of hierarchical spatial acceleration structures in parallel. The results of our implementation show over an order of magnitude speedup in rendering time compared the traditional approach, while keeping memory overhead for message queuing around 1%.

TODO: Triple check that claim with the Toy Store scene results.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

TODO

1.1 Our Contribution

TODO

Chapter 2

Background

TODO

2.1 Light and Radiometry

TODO

2.2 Ray Tracing

TODO

2.3 Bounding Volume Hierarchies

TODO

2.4 Morton Coding and the Z-Order Curve

TODO

Chapter 3

Related Work

TODO

Chapter 4

FlexRender Architecture

TODO

Chapter 5

Results

TODO

Chapter 6

Future Work

TODO

Image Results