Volume Reconstruction and Parallel Rendering Algorithms:
A Comparative Analysis

by
Ulrich Neumann

A dissertation submitted to the faculty of The University of North Carolina at Chapel Hill in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Computer Science

Chapel Hill
1993

Approved by:

____
Advisor: Henry Fuchs

____
Reader: Turner Whitted

____
Reader: Stephen M. Pizer

____
Reader: Jan F. Prins

ULRICH NEUMANN. Volume Reconstruction and Parallel Rendering Algorithms: A Comparative Analysis. (Under the direction of Henry Fuchs.)

Abstract

This work focuses on two issues of concern to designers and implementers of volume-rendering applications - finding the most efficient rendering method that provides the best image possible, and efficiently parallelizing the computation on multicomputers to render images as quickly as possible.

Three volume rendering methods: ray casting, splatting, and volume shearing, are compared with respect to their reconstruction accuracy and computational expense. The computational expense of the rendering methods is modeled and measured on several workstations. The most-efficient rendering method of the three is found to be splatting. Three reconstruction-filter kernels are evaluated for their accuracy. Two error measurement methods are used. The first is image-based and uses a heuristic metric for measuring the difference between rendered images and reference images. The second method is analytical and uses a scale-space measure of feature size to compute an error bound as a function of feature size and sampling density. Of the three filter kernels tested, the separable cubic filter to found to produce the most-accurate reconstruction of the volume function.

Parallel volume-rendering algorithms may be classified and described by the partitioning of tasks and data within the system. A taxonomy of algorithms is presented and the members are analyzed in terms of their communication requirements. Three optimal algorithm-classes are revealed: image partitions with both static and dynamic data distributions, and object partitions with contiguous dynamic block data distributions. Through analysis and experimental tests, a 2D mesh network-topology is shown to be sufficient for scalable performance with an *object-partition* algorithm when the image size is kept constant. Furthermore, the network-channel bandwidth-requirement actually decreases as the problem is scaled to a larger system and volume data size.

Implementations on Pixel-Planes 5 and the Touchstone Delta demonstrate and verify the scalability of object partitions. As part of these implementations, a new load balancing approach is demonstrated for object-partition algorithms.

Acknowledgments

Many thanks to my committee members and especially to Henry Fuchs for acting as my advisor and supporting me through this effort. Their suggestions were helpful in raising the quality of this work in may ways. Any remaining flaws are certainly my responsibility.

Professor Steve Taylor and Mike Palmer at Caltech supported my efforts by acting as hosts for my visit to Caltech and suppling me with information and access to the Touchstone Delta. David Ellsworth, a kindred spirit at UNC who also visited Caltech at the same time, was helpful during many discussions and fruitful late-night coding sessions (when the Delta was up and available). A special thanks goes to Terry Yoo who coordinated the Caltech visit as the UNC site-coordinator of the Graphics and Visualization Science and Technology Center in addition to helping me with countless other details.

Andre State, Qin Fang, and Tim Cullip are fellow-volume renderers who helped keep my thinking straight by participating in numerous discussions about volume rendering and parallel algorithms. Andre State also deserves special thanks for his help with the VVEVOL tests. Tim Cullip must be recognized and thanked for continuously raising the standard with respect to parallel-rendering rates.

Last, but not by any means least, I thank my wife Patricia. Over the years she has mastered the art of balancing encouragement and occasional whip-cracking. Her strength and faith always provided support along this journey. This work is dedicated to her patience and perseverance.

Financial support for this work has come from the Pixel-Planes 5 and VistaNET grants.

Table of Contents

List of Figures

List of Abbreviations

A	Alpha (or Opacity)
ALU	Arithmetic Logic Unit
B	Blue
CAT	Computer-Aided Tomography
G	Green
I/O	Input / Output
K	$2^{10} = 1024$
M	$2^{20} = 1048576$
MIMD	Multiple Instruction Multiple Data
MRI	Magnetic Resonance Imaging
R	Red
RISC	Reduced Instruction Set Computer
SIMD	Single Instruction Multiple Data
VLSI	Very Large Scale Integration
1D	One-Dimensional
2D	Two-Dimensional
3D	Three-Dimensional