IEEE Visualization 2003 Workshop Proposal

Title: Workshop on Parallel Visualization and Graphics

Organizers: Praveen Bhaniramka, SGI
Mike Houston, Stanford University
Greg Humphreys, University of Virginia

Length: Full day workshop

Abstract

Ever increasing data sizes coupled with availability of commodity components in recent years has opened up exciting opportunities in handling large data visualization problems using parallel rendering systems. This workshop aims to bring together various trends in Parallel Visualization and Graphics by covering different aspects of the problem. As part of this workshop, we will provide a comprehensive survey of existing parallel rendering systems and present emerging technologies in this domain. We will bring together ideas from academia, government, and industry to cover a broad spectrum of applications and views. The workshop will consist of a set of invited talks.

We propose the workshop be divided into three main sessions, corresponding to the three main aspects of parallel rendering systems. While these topics have been discussed individually in the past, they have not been addressed together in a coherent manner. The goal of this workshop is to provide such a perspective on the topic. The three sessions will be on:

System Architectures
Software and Toolkits
Applications, Algorithms and Techniques

Introduction

The workshop will begin with an introductory talk on some of the trends in the visualization industry. This talk will focus mainly on the following trends over the last few years pertaining to visualization:

<u>Data Explosion</u> – There has been an exponential growth in data sizes in various application domains, especially those using acquisition devices or computer simulations to generate their data sets. Effective visualization of this data is extremely critical to the success of the application, and is often is the only way of understanding the data.

<u>Commoditization of graphics</u> – The computer graphics industry has recently undergone a revolution whereby innovation is driven by consumer demand rather than specialized, high-end requirements. This has brought real-time graphics to the desktop at commodity prices, creating new opportunities and posing new challenges to developers of visualization applications.

Importance of scalability – The most obvious and efficient way to solve large data visualization problems is to use scalable architectures comprised of commodity graphics components. These architectures are quite new and still evolving. By bringing together the key players in this exciting new direction for graphics architecture research, this workshop will provide an important forum to ensure that the community is working together towards a common vision.

Participants

Participants in the workshop will come from three groups. Several leading researchers will be directly invited to attend based on the scope and importance of their work. We have already contacted the following people who have committed to attend as invitees. Additional invitees will be selected so as to provide broad coverage of the topics listed below.

Core Participants

- Allen McPherson (LANL)
- Mike Houston (Stanford University)
- Greg Humphreys (University of Virginia)
- Ian Buck (Stanford University)
- Shrijeet Mukherjee (SGI)
- Phillip Slusallek (Saarland University)
- Steve Parker (University of Utah)

A second group of participants will be selected based on the submission of a one-page position paper describing their work and what aspects of it they intend to present. Should the number of submissions exceed the available time, those submitters not selected to present will still be invited to attend and contribute to all discussions--We don't want to limit the participation of any interested parties. All selections (additional invitees and position paper submitters) will be chosen so as to achieve a broad coverage of the topics of interest listed below (though we will probably not be able to address all of them).

Finally, depending on IEEE policy and room size, attendees may simply drop in if they have paid for the "tutorial track". These attendees will primarily be "listeners", though they'll certainly be able to contribute to discussions and ask questions along with the others.

Preliminary structure of workshop (all subject to change due to external factors such as date of workshop, room availability, IEEE policy on "tutorial" passes, etc):

We'll assume 7 actual hours in the room (9.0 hours, minus 1-hour lunch,
.5 hour for breaks, and .5 hour for intro/conclusion).

- Each of the invited and selected attendees will present a 30-minute talk on their work, and also present results, plans, and issues. Speakers will be asked to focus on specific portions of their overall project that relate to the topics of interest listed below. This will avoid duplication of mundane issues endemic to all parallel rendering and visualization projects. They will also be asked to identify issues and barriers to success, as well as potential future developments (or desired developments) that will aid their work.

Speakers will be grouped by topic area, with a short question and answer session following each speaker.

- Wrap-up session. The final session will include a panel of four to five attendees that have been selected to summarize the results of the workshop. They will present their quick summary of results, barriers, and issues to the audience. Follow on discussion with the panel and the audience will iterate these results for presentation to the overall audience of conference attendees (which will be presented by the organizer as requested in the call for proposals).

Sessions

System Architectures

This session will cover some of the major categories of parallel visualization systems available today. The systems themselves can be divided broadly into two kinds of systems – Systems which exploit parallelism between a set of dedicated graphics subsystems (GPUs) and those which use more general-purpose computational units (CPUs). More specifically, as part of this session, we will like to cover the following architectures:

<u>Graphics Cluster comprised of independent GPUs</u> – This talk will cover the basic principles behind the design of a graphics cluster. The talk will discuss various issues related to system configurations (CPU types, network interconnects, etc) and touch upon the advantages, issues, and challenges with these systems. Examples of such architectures include the Stanford/Tri-Labs' Chromium cluster and IBM's DeepView system.

<u>Single System Image architectures</u> – This talk will cover the basic principles behind single system image architectures and contrast them with traditional clusters. The talk will cover issues related to system configurations, proprietary interconnects, and performance specifications as applied to SSI architectures as well as touch upon the various pros and cons and challenges with such architectures. Examples of such architectures are systems from vendors such as SGI, SUN, and HP.

<u>CPU-based systems</u> – This talk will look at some massively parallel software rendering systems. Such systems usually contrast with the GPU-based systems in the granularity of decomposition schemes used and in certain cases can surpass them in image quality, scalability and system flexibility. There is a large number of such parallel software rendering systems including University of Utah's StarRay project, University of Saarland's parallel ray-tracing system, and the NASA Ames Unstructured grid renderer.

Software and Toolkits

The second session will cover various rendering software and toolkits that enable developers/researchers to take advantage of these massively powerful computing/rendering systems. This software can be divided into two main categories from a usage and scalability perspective:

Rendering Toolkits – These are software application programming interfaces (APIs), which allow application developers to write scalable rendering software by employing various parallelization techniques depending on the problem at hand. These toolkits essentially provide some kind of a library, which a developer can integrate into his/her application. Commonly known examples of toolkits which support parallel rendering are the Visualization Toolkit (VTK), OpenGL Performer, OpenGL Multipipe SDK and more recently the OpenRT system from University of Saarland. These toolkits mainly differ in the data organization (Data Flow vs. Scene Graph) and parallel rendering schemes (Data decomposition vs. Pipelining) they use.

Transparent Software – These software toolkits are transparent to the application and work by intercepting the stream of OpenGL calls made by the application. These toolkits apply intelligent state management techniques to provide some speedup and parallel rendering schemes to provide scalability to existing applications. These systems however provide only limited scalability and pose some interesting research problems from a transparent scalability point-of-view. Some projects in this area include Stanford's Chromium and SGI's OpenGL Multipipe.

Applications and Algorithms

The third session of the workshop will focus on new algorithms and techniques that exploit the scalability and efficiency of parallel rendering systems. These talks will identify a particular application domain and discuss specific algorithms in this domain. Some potential candidates for this session will be:

Parallel rendering techniques – This talk will focus on new ways of looking at different rendering problems and solving them using multiple GPUs. For example, researchers at University of North Carolina use multiple GPUs to do occlusion culling by pipelining the rendering and computation of occlusion mask across multiple GPUs, and the medical imaging community uses multiple GPUs for parallel volume rendering.

New GPU-based Applications – With the advent of the new programmable graphics hardware, it is now possible to implement certain algorithms on the graphics hardware that exploit the SIMD architecture of these GPUs. There is a completely new set of applications that emerge out of this domain. Common examples of these would be doing general-purpose computation, and implementing new rendering algorithms, on GPUs.

Products and Results of Workshop

Although the workshop, by design, will not result in any published papers, there are two mechanisms for collecting the content of the workshop and making it available to a wider audience. We will manage the construction of a web site dedicated to the workshop containing the results summary statement and links to each individual presenter's materials. A written summary of the workshop will be completed with the help of the invited speakers including the questions and future directions discussed during the panel. It is also possible that certain publications may be interested in presenting a two-page summary of results from the workshop. Should the workshop be accepted, we will pursue this option with IEEE CG&A or other publications of interest.

Organizers

Praveen Bhaniramka, Silicon Graphics, Inc

Praveen Bhaniramka got his engineering degree in 1998 in Computer Science and Engineering from Institute of Technology, Banaras Hindu University, India. He went to graduate school in Computer and Information Science at The Ohio State University, USA. At OSU, he worked on various research projects, including his thesis work on higher dimensional isosurface construction. After graduating with a Masters' in 2000, he joined the Advanced Graphics Division at Silicon Graphics, Inc. At SGI, he has been working on OpenGL Volumizer (http://www.sgi.com/software/volumizer). He is currently the Technical Lead on the project. His research interests include visualization of extremely large data sets, scalable graphics, and data fusion for multi-modal visualization.

Selected Publications:
"Isosurfacing in Higher Dimensions", Praveen Bhaniramka, Rephael Wenger, Roger Crawfis, IEEE Visualization 2000.

"OpenGL Volumizer: A Toolkit for High Quality Volume Rendering of Large Data sets", Praveen Bhaniramka, Yves Demange, IEEE Volume Visualization and Graphics Symposium, 2002.

"Isosurface Construction in Any Dimensions Using Convex Hulls", Praveen Bhaniramka, Rephael Wenger, Roger Crawfis, To appear in IEEE Transactions on Visualization and Computer Graphics.

Mike Houston, Stanford University

Mike Houston did his undergraduate studies at University of California, San Diego, getting a BS with honors in Computer Science 6/2001. As part of his time at UCSD, he worked in the San Diego Supercomputer Center's Visualization Group working on parallel volume rendering and data management systems (http://vistools.npaci.edu). He began his graduate work at Stanford University and research in the Stanford Graphics Lab in 9/2001. As part of this research, he has authored several papers on different subjects in computer graphics, including the development of Chromium (http://chromium.sourceforge.net). His current interests are the design and implementation of graphics clusters, massively parallel volume rendering, and computation on GPUs.

Selected Publications:
"Chromium: A Stream Processing Framework for Interactive Graphics on Clusters", Greg Humphreys, Mike Houston, Yi-Ren Ng, Randall Frank, Sean Ahern, Peter Kirchner, and James T. Klosowski, SIGGRAPH 2002.

"Non-Invasive Interactive Visualization of Dynamic Architectural Environments", Christopher Niederauer, Mike Houston, Maneesh Agrawala, Greg Humphreys, ACM SIGGRAPH 2003 Symposium on Interactive 3D Graphics.

"Fast Volume Segmentation With Simultaneous Visualization Using Programmable Graphics Hardware", Anthony Sherbondy, Mike Houston, Sandy Napel, To appear in IEEE Visualization 2003.

Greg Humphreys, University of Virginia

Greg Humphreys is an assistant professor at the University of Virginia. He received his Ph.D. in Computer Science from Stanford in 2002 under the supervision of Pat Hanrahan. His dissertation, "A Stream Processing Approach to Interactive Graphics on Clusters of Workstations", described several approaches to building flexible and scalable graphics architectures on clusters of workstations with commodity graphics accelerators. His flagship software system, Chromium, is in production use in hundreds of academic, research, and industrial labs to meet their needs for both big data visualization and support for large tiled displays. Currently, he is working on making scalable commodity-based graphics into a remotely accessible service that can easily be shared by multiple users and provide the combined rendering power of multiple graphics accelerators to a remote display. He is also very interested in exploiting GPUs for general purpose computation and novel visualizations of complex environments.

Selected publications:
"Scene Graph Enabled Distributed Memory Visualization and Rendering". E. Wes Bethel, Greg Humphreys, Brian Paul, and J. Dean Brederson. IEEE Symposium on Parallel and Large Data Visualization 2003.

"A Multigrid Solver for Boundary Value Problems Using Programmable Graphics Hardware". Nolan Goodnight, Cliff Woolley, Greg Lewin, David Luebke, and Greg Humphreys. Graphics Hardware 2003.

"Interactive Time-Dependent Tone Mapping Using Programmable Graphics Hardware". Nolan Goodnight, Rui Wang, Cliff Woolley, and Greg Humphreys. Eurographics Symposium on Rendering 2003.

"Non-Invasive Interactive Visualization of Dynamic Interactive Environments". Chris Niederauer, Mike Houston, Maneesh Agrawala, and Greg Humphreys. ACM Symposium on Interactive 3D Graphics 2003.

"Chromium: A Stream Processing Framework for Interactive Graphics on Clusters". Greg Humphreys, Mike Houston, Yi-Ren Ng, Randall Frank, Sean Ahern, Peter Kirchner, and James T. Klosowski. SIGGRAPH 2002.

"WireGL: A Scalable Graphics System for Clusters". Greg Humphreys, Matthew Eldridge, Ian Buck, Gordon Stoll, Matthew Everett, and Pat Hanrahan. SIGGRAPH 2001.