A Survey of Distributed Rendering Applications

Jacob Lambert
Computer Science Department, University of Oregon
CIS 630 Distributed Systems, Spring 2017
jlambert@cs.uoregon.edu

Keywords—**Distributed Systems, Rendering, Visualization**

I. Context

Rendering, the processes of generating an image from a model, is a necessary step for any visualization application. Because rendering can be computationally expensive, many applications have methods for accelerating the rendering process. One approach is to distribute the rendering computation across multiple computation units, even on different machines. That is, the rendering stage can be run on a distributed system.

I want to explore the different types of rendering software used in different contexts. The specific contexts I am exploring are:

- Distributed rendering applications designed for visualizing large scientific data-sets.
- Distributed rendering applications designed for visualizing detailed 3D models for use in digital animation.

II. Paper Selection

I chose the following two papers as seed papers:

- "A sorting classification of parallel rendering" - 1994 [1]
- "A data distributed, parallel algorithm for ray-traced volume rendering" - 1993 [2]

These papers have a high number of citations, and seem to have been very influential for the theoretical base of parallel rendering. Many of the current parallel distributed rendering applications cite these papers.

Next, primarily by exploring papers citing the seed papers, I generated a list of software used in academic research settings. These programs perform distributed rendering with the goal of visualizing scientific data:

- Chromium [3]
- VTK [4]
- Paraview [5]
- VisIT [6]
- Equalizer [7]

These applications all perform some form of distributed rendering. I would like to better understand the specifics of each application, and to see how the newer applications have made improvements over older implementations.

I will then survey several papers that use the above applications to solve problems, specifically focusing on their use of distributed systems. Paraview and VTK are showcased in [8], VisIT and VTK in [9], Equalizer in [10], and Chromium in [11].

Finally, I want to briefly look into some distributed parallel applications designed for rendering photo-realistic models and digital animations. Some of the most popular rendering software are:

- V-Ray, used by Maya, 3D Max, Rhino, and several other high-end modeling software
- Renderman, parallel rendering software developed and used by Pixar
- Mental-Ray, a competitor to V-Ray, available on many of the same modeling software packages

These commercial applications are not typically showcased in academic publications. However, information about how they utilize distributed systems for parallel rendering is available from web resources and each application's documentation. For example:

V-Ray: https://docs.chaosgroup.com/display/VRAY3MAX/Set+Up+Distributed+Rendering

Renderman (Tractor): https://rmanwiki. pixar

.com/ display/TRA/Implementation

III. Key Issues

I want to explore several key issues.

- I want to see how the theory of distributed rendering presented in the seed papers is applied in the development and execution of the parallel rendering applications.
- I want see how each application utilizes a distributed system to perform rendering. Even though all of the research-based applications are intended to visualize data, they may come from different domains, and may be used to solve problems with different kinds of requirements.
- I want to discover how these applications have been used in academic research, specifically where their ability to render across distributed systems was utilized.
- Finally, I want to compare the academic software designed for data visualization against the commercial software designed for digital animation and photo-realistic images. Because these two classes of rendering applications have very different requirements, I expect their use of distributed systems to vary.

References

[1] Steven Molnar, Michael Cox, David Ellsworth, and Henry Fuchs, "A sorting classification of parallel rendering," *IEEE computer graphics and applications*, vol. 14, no. 4, pp. 23–32, 1994.

[2] Kwan Liu Ma, James S Painter, Charles D Hansen, and Michael F Krogh, "A data distributed, parallel algorithm for ray-traced volume rendering," in *Proceedings of the 1993 symposium on Parallel rendering*. ACM, 1993, pp. 15–22.

[3] Greg Humphreys, Mike Houston, Ren Ng, Randall Frank, Sean Ahern, Peter D Kirchner, and James T Klosowski, "Chromium: a stream-processing framework for interactive rendering on clusters," *ACM transactions on graphics (TOG)*, vol. 21, no. 3, pp. 693–702, 2002.

[4] William J Schroeder and Kenneth M Martin, "The visualization toolkit-30," 1996.

[5] Utkarsh Ayachit, "The paraview guide: a parallel visualization application," 2015.

[6] Hank Childs, "Visit: An end-user tool for visualizing and analyzing very large data," 2013.

[7] Stefan Eilemann, Maxim Makhinya, and Renato Pajarola, "Equalizer: A scalable parallel rendering framework," *IEEE transactions on visualization and computer graphics*, vol. 15, no. 3, pp. 436–452, 2009.

[8] Jonathan Woodring, Katrin Heitmann, James Ahrens, Patricia Fasel, Chung-Hsing Hsu, Salman Habib, and Adrian Pope, "Analyzing and visualizing cosmological simulations with paraview," *The Astrophysical Journal Supplement Series*, vol. 195, no. 1, pp. 11, 2011.

[9] Thomas Fogal, Hank Childs, Siddharth Shankar, Jens Krüger, R Daniel Bergeron, and Philip Hatcher, "Large data visualization on distributed memory multi-gpu clusters," in *Proceedings of the Conference on High Performance Graphics*. Eurographics Association, 2010, pp. 57–66.

[10] Stefan Eilemann, Ahmet Bilgili, Marwan Abdellah, Juan Hernando, Maxim Makhinya, Renato Pajarola, and Felix Schürmann, "Parallel rendering on hybrid multi-gpu clusters," in *Eurographics Symposium on Parallel Graphics and Visualization*. The Eurographics Association, 2012, number EPFL-CONF-216016, pp. 109–117.

[11] Fabrizio Lamberti and Andrea Sanna, "A streaming based solution for remote visualization of 3d graphics on mobile devices," *IEEE transactions on visualization and computer graphics*, vol. 13, no. 2, pp. 247– 260, 2007.

The references used to meet the paper requirements are

1. [1]
2. [2]
3. [3]
4. [7]
5. [9]
6. [8]
7. [10]
8. [11]