FPGA-Based Image Combining for Parallel Graphics Systems

Jens Maiero, André Hinkenjann and Marco Winzker
Bonn-Rhein-Sieg - University of Applied Sciences
{jens.maiero, andre.hinkenjann, marco.winzker}@h-brs

Matthias Bues
Stuttgart - Fraunhofer IAO
matthias.bues@iao.fhg.de

Abstract

This paper describes FGPA-based image combining for parallel graphics systems. The goal of our current work is to reduce network traffic and latency for increasing performance in parallel visualization systems. Initial data distribution is based on a common ethernet network whereas image combining and returning differs to traditional parallel rendering methods. Calculated sub-images are grabbed directly from the DVI-Ports for fast image compositing by a FPGA-based combiner.

1. Introduction

The system introduced in this paper is designed to visualize large 3D models/scences and highly complex shaders in real-time. In comparison to traditional parallel graphics systems the realized FPGA-based system reduces network traffic, latency and memory accesses by directly grabbing the rendered sub-images from the DVI-Ports of the render servers [MH09] [BH04]. An especially designed hardware combiner merges the rendered sub-images.

Related publications or commercial systems in this area often have complex system configurations, are too expensive or merge sub-images in cascaded network processing units [OW06] [PM07]. An additional advantage is that graphics hardware manufacturers produce more and more affordable GPUs supporting frame- and genlock.

2. Implementation

Because of the flexibility of FPGAs, different algorithms for parallel rendering can be implemented. Due to the genlocked signal of the rendered image data the hardware combiner has no need for an external RAM. This is one reason for the cost-effectiveness of the hardware. Another reason for the economic efficiency is that the hardware is based on standard components like FPGAs and DVI-D, which also makes the system small and compact. The current implementation of the FPGA is the sort-first algorithm [MC94]. In this case each connection via DVI-D transfers only color information to the combiner. In a final step the combiner merges the calculated sub-images additively. To provide a high speed-up in parallel rendering clusters using hardware rendering, the models have to be spatialized for an efficient culling process.

In interactive and dynamic scenes the need for good load balancing is essential. The implemented dynamic view frustum splits the scene into *n* new frusta. The size of the frusta is coordinated by the render client and calculated every frame depending on the response time of each server. Due to the synchronization between all render servers and the static latency of the combiners it is guaranteed that more than one hardware combiner can be used. The cascading composition of the combiners unlocks the limitation of a fix number of render servers, which enables an arbitrary performance benefit.

Concerning the FPGA combiner implemetation: the graphics format is DVI, which is imperative for the use of standard DVI encoder and decoder for the signals. Because the limited storage of the FPGAs, the sub-images have to be framelocked. The combiner has to balancing small variations in graphics timing by fitting all signals to a master signal.

Bottleneck network trafic: let the network bandwidth be 1 GBit/s, and let the image resolution be 1000·1000 pixels (RGB), hence we can achieve a maximum frame rate of 44,7 Hz.

3.Results

The setup for the first evaluation: one render client and two render servers with a Nvidia FX 3000G. The following results are measured using a XGA resolution and a dynamic view frustum. The latency of the FPGA combiner is about 40 µs.

Model	Tris	Spatialized	1 Server	2 Server
Mercedes 300SL	800 000	no	20 fps	30 fps
Infinity Triant	1 227 000	yes	29 fps	59 fps
Synthetic Scene	2 000 000	yes	22 fps	44 fps

Future work of this project is to implement the sort-last algorithm and to analyze cascading, alpha-blending and other renderers e.g. a ray tracer or a volume renderer.

References

[MH09] M. Bues, 2009. *A system for high-quality real-time rendering for Virtual Environments. Dissertation: IPA-IAO-Forschung und Praxis.*

[BH04] A. Hinkenjann and M. Bues, 2004. *Network aware parallel rendering with pcs. In VRCAI '04: Proccedings of the 2004 ACM SIGGRAPH, New York.*

[MC94] S. Molnar, M. Cox, D. Ellsworth and H. Fuchs, 1994. *A sorting classification of parallel rendering. IEEE Computer Graphics and Applications 14.*

[OW06] openWarp, 2006. *openwrap combiner, http://www.openwarp.com/download/openWARP_combiner.pdf, last viewed 2010.*

[PM07] D. Pugmire, L. Monroe, Laura , C. Davenport, A. DuBois, D. DuBois and S. Poole, 2007. *NPU-Based Image Compositing in a Distributed Visualization System. IEEE Transactions on Visualization and Computer Graphics, Piscataway, NJ, USA.*