Computer Graphics Lab

PIXELSTROM

http://pixelstrom.inf.h-bonn-rhein-sieg.de/

Fraunhofer IAO

Bonn-Rhein-Sieg University of Applied Sciences

The Eighth Computer Science Research Exposition - 2011 - [UNB]

FPGA-Based Image Combining for Parallel Graphics Systems

Jens Maiero[1], André Hinkenjann[2], Marco Winzker[3] and Matthias Bues[4]

Introduction

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

Implementation

Integration in VE-Framework:
in interactive and dynamic scenes the need for **good load balancing** is essential. The implemented **dynamic view frustum** splits the scene into sub-frusta. The size of the sub-frusta is coordinated by the render client and calculated every frame depending on the response time of each server. The integration into our VE-Framework basho provides the opportunity to use different renderers.

FPGA combiner implementation:
the graphics format is DVI-D, which is imperative for the use of standard DVI-D encoder and decoder for the signals. Due to the limited storage of the FPGAs, the sub-images have to be framelocked. The combiner has to **balance small variations** in graphics timings by fitting all signals to a master signal.

Results

Setup for the evaluation:
one render client and up to two render servers with a Nvidia FX 3000G. The following results are measured using a XGA resolution and a **dynamic view frustum** (sort-first approach [MC94]). 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

Bottleneck network traffic: 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 [BH04].

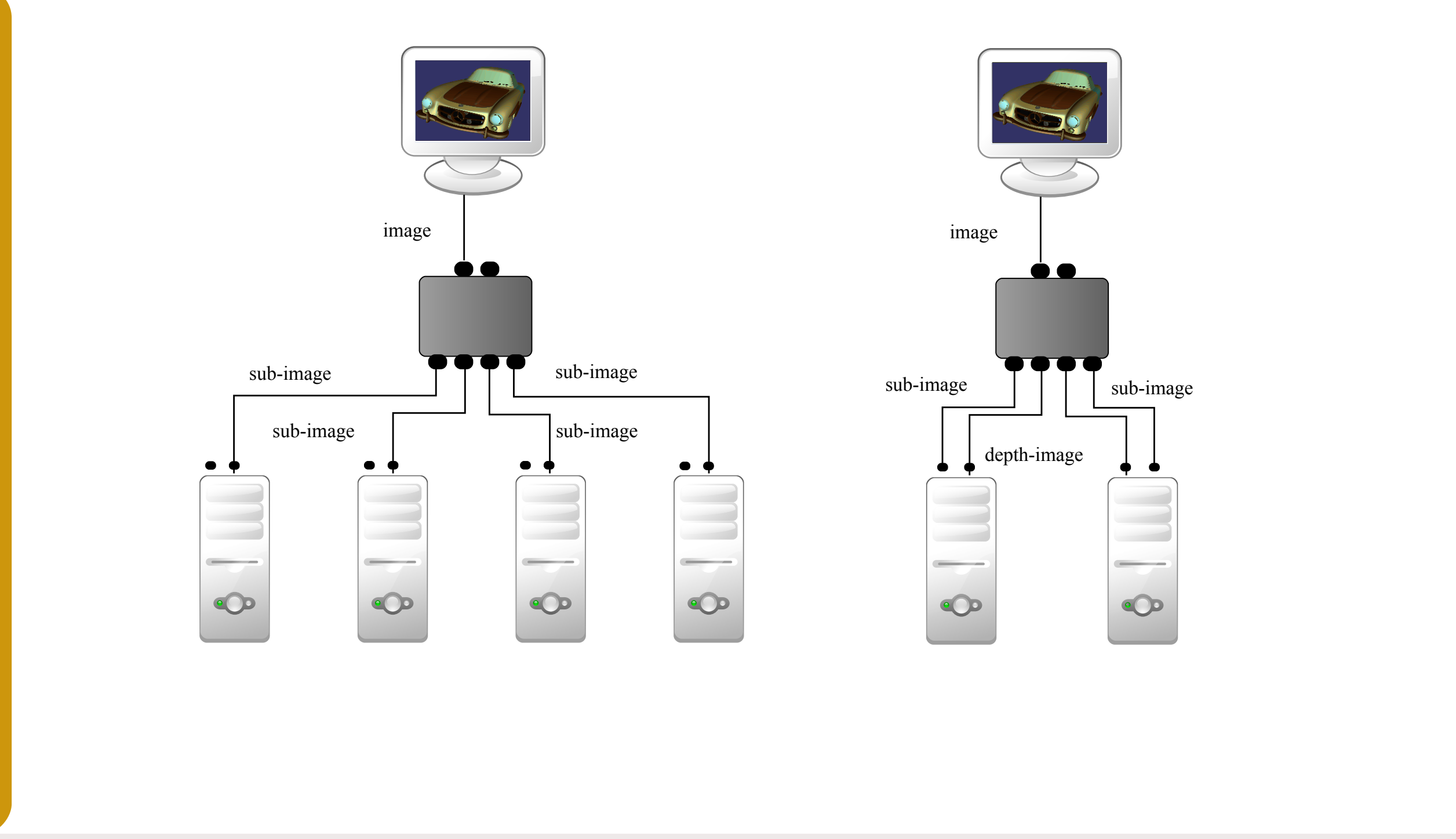

Fig 1: Example combiner setup (sort-first / sort-last)

Related Work

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]. In comparison with Lightning-2 our system does not buffer image-data, because it relies on framelocked video streams. Graphics hardware manufacturers produce more and more affordable GPUs supporting frame- and genlock. Also the simplicity and price differs from Lightning-2. In comparision to SLI our system is not limited to a fix number of GPUs.

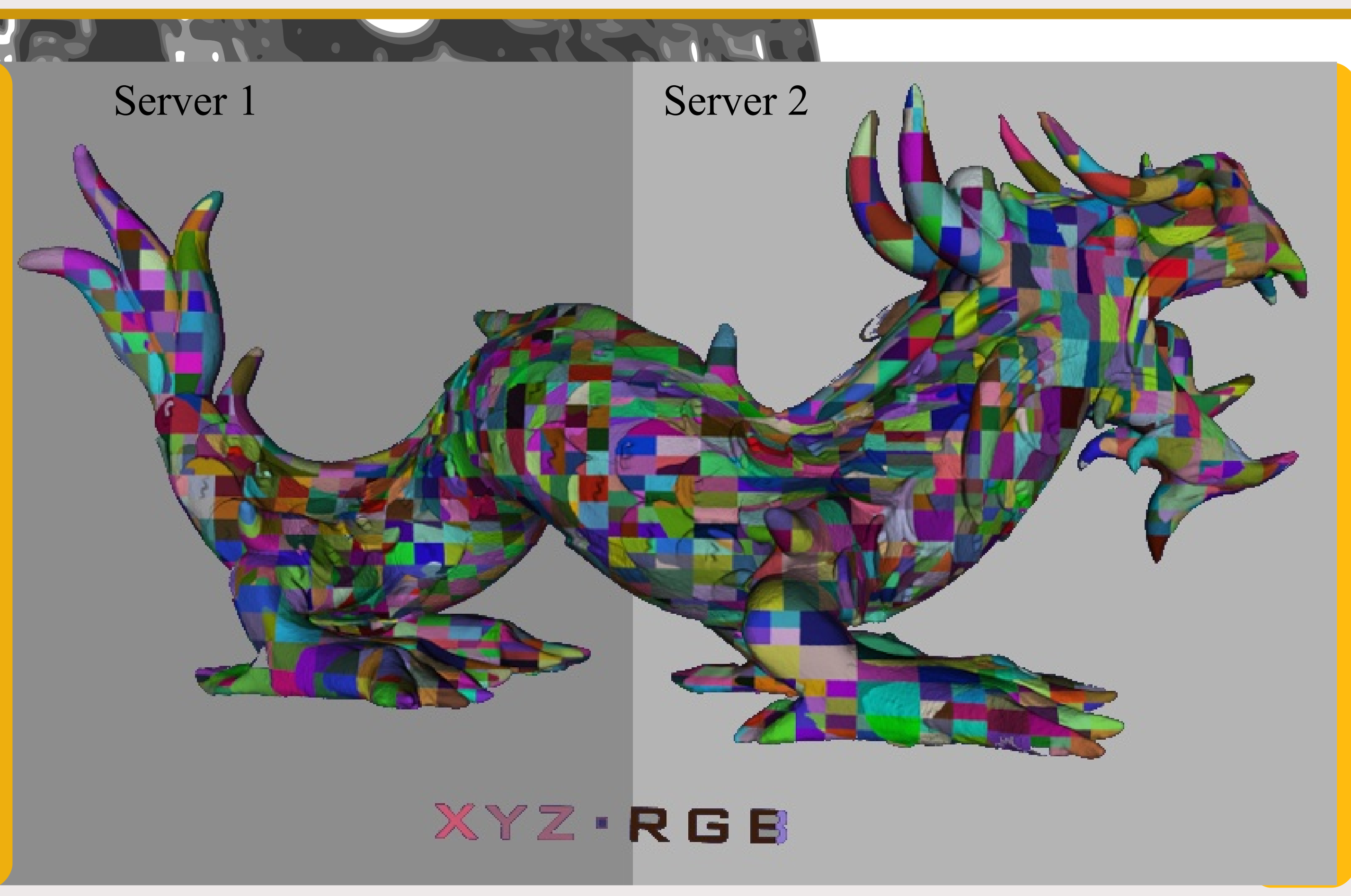

Fig.2: Spatialized Stanford Model (dragon_xyz_rgb)

References

[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-Visualization System. IEEE Transactions on Visualization and Computer Graphics, Piscataway, NJ, USA.Based Image Compositing in a Distributed

Contact:
Bonn-Rhein-Sieg University of Applied Sciences: {jens.maiero, andre.hinkenjann, marco.winzker}@h-brs.de
Fraunhofer IAO, Stuttgart: matthias.bues@iao.fhg.de