Reconfigurable Point Based 3-D Graphics Rendering on FPGA

Amit Ghare, Siddharth Rajagopal

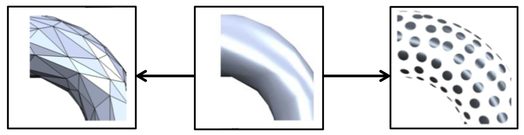
Department of Electrical Engineering, San Jose State University, San Jose, California 95112.

The problem

- Triangle meshes are dominant primitives in graphics processing.
- Complex 3D object models with triangles meshes are expensive to store and process
- With millions of triangle meshes, meshes that are smaller than a pixel do not get displayed on the screen
- Complex model demands faster processing and efficient storage

Point based rendering

- Represent 3D objects using points rather than triangle meshes
- Point has location, normal, color, alpha etc. information
- Conceptually simpler than triangle meshes
- Flexible and scalable processing possible
- No connectivity information => lesser demands on storage



Mesh Rendendering *Sample* *Point Rendendering*

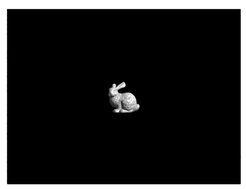
Proposal

We propose a reconfigurable direct 3D point based rendering system.

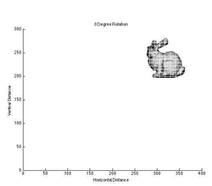
- System should be able to programmable
- 60 fps frame rate
- Double buffering
- 18 bit color
- Modular design approach targeting Xilinx ML505 evaluation platform

High Level Modeling

- OpenGL model gives high level overview of operations involved
- Not much insight into hardware operation
- MATLAB is closer to hardware operations

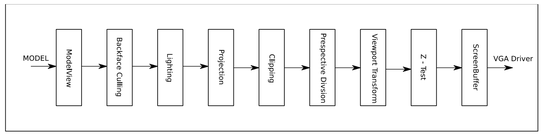


OpenGL model



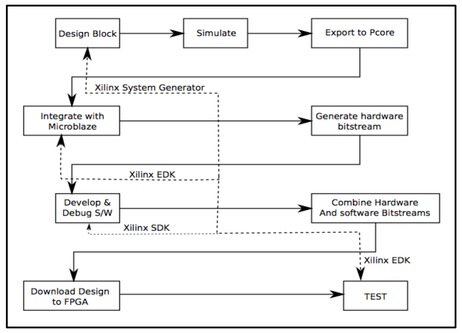
MATLAB model

Point based rendering pipeline

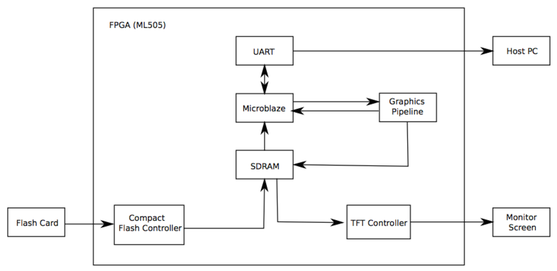


- Determine operations and optimizations involved in each stage
- Decide hardware/software implementation
- Figure out control mechanism and implementation strategy
- Prepare test plan for block level and system level design

Implementation process

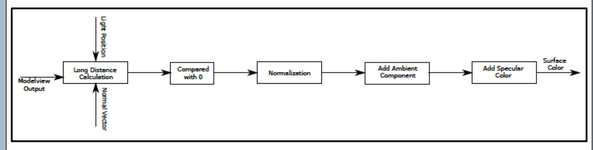


Hardware



- Design using available Xilinx IPs
- Microblaze processor is central control processor
- On board SDRAM is used as frame buffer and z buffer
- UART for debug
- Rendering pipeline attached as slave to peripheral bus
- Vertex, normal and color data stored on Compact Flash card
- Hardware reconfigurability achieved by storing different designs as different bitstreams.

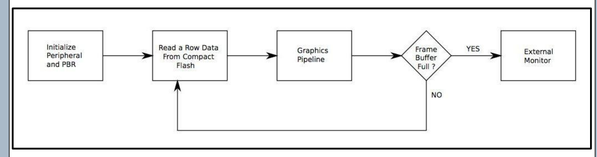
Color pipeline



Phong Shading implemented in hardware

- Square root required for normalization approximated to less than 10% error in about 5 clocks
- Specular component stored as lookup table
- Diffuse component calculated in hardware

Software



- Lightweight standalone OS provided by Xilinx run on Microblaze
- Software initializes, configures, monitors and manipulates hardware
- Modelview operation in software to reduce design size
- Built in drivers for peripherals used to reduce time and effort in writing drivers

Testing and Results

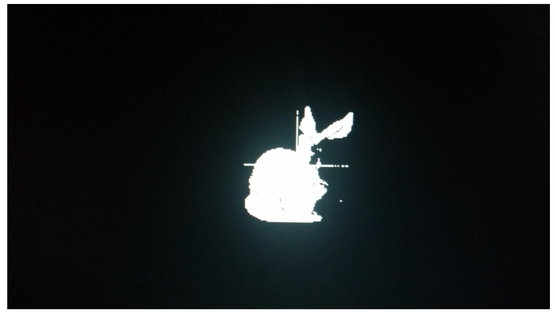


Figure 4: FPGA rendering of point model

- Individual blocks tested with UART
- Incremental test strategy followed to allow easy debug
- Rendering pipeline tested using software accessible registers
- 1,00,000 point model rendered at 60 fps
- Hardware uses about 60% of FPGA resources

Takeaways

- Use of existing IPs reduces design turn around time, improves modularity.
- Reconfigurable design increases flexibility and scalability.
- Can complement existing mesh based rendering techniques
- Design has room to incorporate complex point rendering algorithms and reconfiguration options

Principal Contributions

- Hardware/Software co-design framework for the design of reconfigurable PBR architecture
- Design and implementation of a hardware architecture for PBR targeted towards an FPGA implementation
- Modeling and verification of PBR in MATLAB and OpenGL

References

- M. Levoy and T. Whitted, The use of points as a display primitive. University of North Carolina, Department of Computer Science, 1985.
- Majer, M.; Wildermann, S.; Angermeier, J.; Hanke, S.; Teich, J., "Co-design Architecture and Implementation for Point-Based Rendering on FPGAs,"Rapid System Prototyping, 2008. RSP '08. The 19th IEEE/IFIP International Symposium on , vol., no., pp.142,148, 2-5 June 2008
- Real time 3-d graphics processing hardware design using field-programmable gate arrays, James Ryan Warner, M.S. University of Pittsburgh, 2008
- T. Akenine-Moller, T. Moller, and E. Haines, Real-Time Rendering, 2nd ed. Natick, MA, USA: A. K. Peters, Ltd., 2002.
- Xilinx online support

Acknowledgements

We wish to thank Dr. Thuy Le for his guidance through this challenging and rewarding project.
We would also like to thank San Jose State University for providing a conducive environment and access to resources

The team

Siddharth Rajagopal

Amit Ghare