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 triangle 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.

Proposal
We propose a reconfigurable direct 3D point based rendering system.
- System should be able to be 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.

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
- Design blocks.
- Simulate and verify.
- Synthesize for FPGA.
- Implement on FPGA.
- Verify functionality.

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.
- Model/view operation in software to reduce design size.
- Built in drivers for peripherals used to reduce time and effort in writing drivers.

References
- Real time 3-d graphics processing hardware design using field-programmable gate arrays, James Ryan Warner, M.S. University of Pittsburgh, 2008.
- 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