Fast Sorting in OpenGL Shaders for Order Independent Transparency

Pyarelal Knowles  Geoff Leach  Fabio Zambetta

RMIT University, Melbourne

Abstract

Order independent transparency (OIT) is a class of techniques to render transparency which sort surfaces as fragments per-pixel rather than sorting polygons. First a deep image is constructed, containing all fragments. Then in a second pass, they are sorted and composited. In exact OIT, where all surfaces are sorted and blended, unlike approximate approaches, the sorting stage is the bottleneck for large scenes. With a combination of two shader based techniques a rendering performance increase of over a factor of ten is achieved for OIT. This improvement is shown to have overcome the original GPU generations.

Constructing a Deep Image

With the ability to write to arbitrary memory locations from the fragment shader and atomic operations, all fragments can be captured and stored in a single rendering pass of the geometry. Per-pixel linked lists are a common capture technique and storage data structure.

Backwards Memory Allocation (BMA)

BMA [1] groups pixels by depth complexity intervals and composite with separate shaders. Each has the correct memory defined. This improves occupancy in the lower intervals, allows easy per-interval optimization and has potential to reduce divergence.

Register-based Block Sort (RBS)

Rather than sort in global memory, the temporary fragments array, using local memory, is used in sortFragments. However, the high numbers of compare and swap operations are still relatively expensive. By explicitly unrolling loops, the compiler is able to store values in and use much faster registers.

Results

BMA and RBS improve OIT rendering performance for most scenes, in particular the power plant where it is more than a factor of ten. In cases where there is little or no opportunity for improvement, such as the hairball with nearly already sorted data, only small overhead is observed.

Power plant rendering time and speedup for different GPUs.

Conclusion

An order of magnitude speed increase allows real-time OIT rendering of more complex scenes that were not previously possible. Power plant now only 3× slower than normal.

Future Work

Sorting is superlinear and further improvements will be important for scenes even larger than the powerplant and below, deep image construction time is becoming more significant and needs addressing. Techniques that address atomic contention (perhaps with ARB_fragment_shader_interlock) and memory locality would be beneficial.

For example, fragments are saved in the order the fragment shader executes. This loosely follows the pattern below, but still varies significantly. Fragments are read in pixel arrival order, which is different, and impacts cache performance. This is particularly important for deep images that are read many times, for example in impostor rendering.

Deep Image Visualization

The UNC power plant model rendered with transparency. It is a large scene with 12 million triangles and a high depth complexity.

Code is available at https://github.com/pknowles/oit

References


Acknowledgements

Special thanks to Paul Philips from HP and Mike Wang from Nvidia for providing the hardware used to develop and run the experiments. Also to the open source community for their software.