Performance and Precision: Mobile solutions for high quality engineering drawings

Ravi Krishnaswamy
Autodesk Inc.
ravi.krishnaswamy@autodesk.com

ABSTRACT
Engineering documents e.g. 'blueprints' are one of the traditional forms of paper based information moving more to the digital realm. With mobile and the evolution of GPUs on mobile, there are tremendous opportunities for applications that view and interact with engineering documents.

SOLUTIONS
The solutions we discuss are:

1. Solutions for rendering analytical curves that are high performance and produce high quality graphics. We discuss innovations around ellipses.
2. Solutions for line types (line styles) that provide order of magnitude gains through leverage of texture efficiencies on the GPU while improving quality over CPU generated patterns at the same time.
3. A topological sort based preprocessing stage to handle draw order and transparency.

For empirical results, we look at two different mobile GPUs on Android – the Adreno 320 (Samsung S4, Nexus 7) and Tegra 4 k1 (Nexus 9). We present some of our findings and encouraging results.

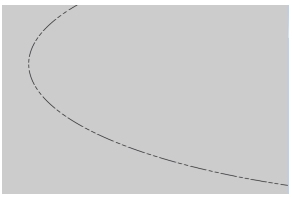
ANALYTIC RENDERING
We apply techniques from the references mentioned to the implicit form of the ellipse equation to estimate the distance from the curve using gradients. This is computed on mobile device using dFdx and dFdy intrinsics.

Memory Savings
Our solutions use hardware instancing, available now in OpenGLES 3.0. Through that we get substantial savings, over 75% for certain curve types.

LINETYPES
Zoom dependent line types (line styles) and handling dots on line styles as a special case, are one of the important topics to solve well for engineering drawings.

SIGGRAPH 2015 Posters, August 09 – 13, 2015, Los Angeles, CA.
ACM 978-1-4503-3632-1/15/08.
http://dx.doi.org/10.1145/2787626.2792615

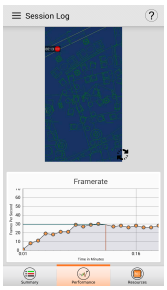
By defining a bucketing data structure stored on the GPU using texture memory we are able to implement a fast and highly efficient line type rendering solution.



Challenges included parameterization of ellipse length, which we implemented via an approximation formula, and handling zoom variance. Handling dots required a unique bucketing approach using texture memory, which provided high quality as well as high performance results.

DRAW ORDER
We discuss a topological sort based algorithm to solve the issue of draw order defined transparency without having to rely on more advanced GPU features not available on most mobile devices.

PERFORMANCE
Our experiments with both synthetic and real world data showed good results with interactive performance, and the ability to handle large data sets on smaller devices.



References
CHARLES LOOP, JIM BLINN. 2005. Resolution independent curve rendering using programmable graphics hardware. In *ACM Transactions on Graphics (TOG) - Proceedings of ACM SIGGRAPH 2005* Volume 24 Issue 3, July 2005 pages 1000-1009.

CHARLES LOOP, JIM BLINN. 2007. Chapter 25 – rendering vector art on the GPU. In *GPU Gems 3*. Hubert Nguyen (editor) Addison Wesley 2007 pages 543 – 562