PERFORMANCE EVALUATION OF GRAPHICS ACCELERATOR

Juraj Vanek
Master Degree Programme (4), FIT BUT
E-mail: xvanek29@stud.fit.vutbr.cz

Supervised by: Adam Herout
E-mail: herout@fit.vutbr.cz

ABSTRACT

This paper deals with possibilities and functions of modern graphic accelerators and with measuring performance under OpenGL interface. Widespread algorithms to render scene in real-time are used. It focuses on how to test every part of accelerator's graphic pipeline by implementing multiple test series and evaluation of these measurements. Final application enables setting of test parameters and outputs a score, by which it is possible to judge accelerator's performance in comparison to others.

1. INTRODUCTION

Today we witness rapid development of graphics accelerators, so we also need tools to effectively evaluate their performance. This is done by rendering complex 3D test scenes with various advanced effects. For these measurements specialized applications („3D Marks“) are used, as they test graphics card on various complex scenes producing final score, by which is possible to compare performance among different graphic accelerators.

2. PERFORMANCE ANALYSIS

The speed of graphics card depends primarily on its graphics processor (GPU), which can be based on it's core clock, number of texture units, stream processors and memory bandwidth. To effectively evaluate performance of GPU, we need to know a way how output image is being progressively created in graphics pipeline. Main stages of graphic pipeline are as follows [3]:

- **Vertex shader stage:** works with vertices and performs its transformation.
- **Geometry shader stage:** works with whole primitives, can emit new vertices.
- **Fragment (pixel) shader stage:** applies per-pixel operations like texturing for instance or many other effects.

All these stages are fully programmable using shading languages (HLSL for DirectX, GLSL for OpenGL [2]). Today's GPU with its general purpose stream processors can be also used for non-graphics parallel computations (interfaces OpenCL and Direct Compute), so tests to measure raw computational power of GPU will also be useful.

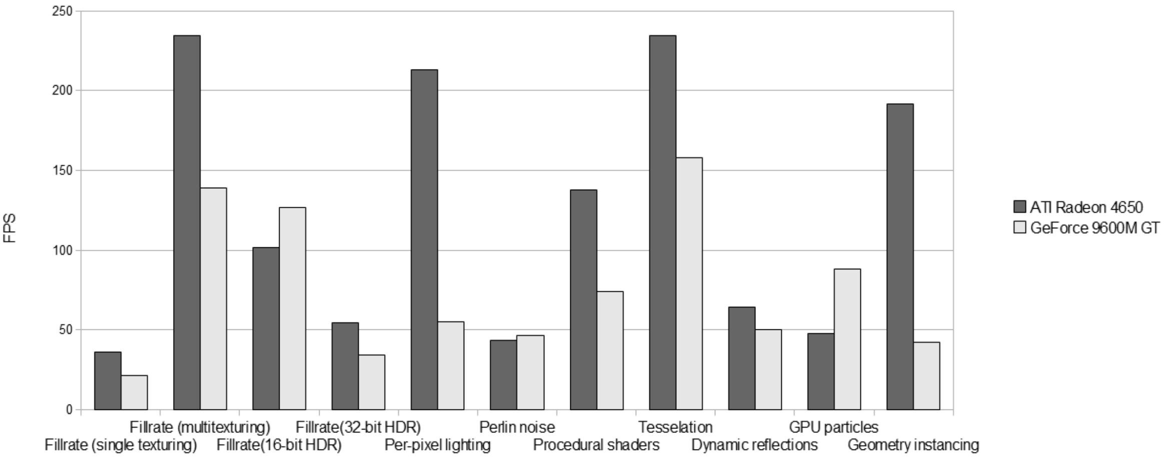
3. TESTS

Test design is derived from principles described in previous section, when every test series will put performance load on particular part of graphics pipeline (or whole pipeline). All tests are designed to be as little CPU speed independent as possible. Application runs under OpenGL [1] interface which is multiplatform and enables measuring performance under different operating systems and different system setups. Now we will describe individual test series:

- **Fillrate tests –** we measure how many pixels/texels is GPU theoretically capable to draw in one second by using single or multiple texture units. Some tests use floating point HDR textures. Speed is measured in milions of pixels per second (Mpixels/s).
- **Fragment shader tests –** use various per-pixel effects to put load on fragment shader like per-pixel Phong lighting with multiple lights, Perlin noise with multiple octaves, procedural textures generated entirely in shader [2], parallax mapping and static reflections and refractions using cube maps.
- **Geometry shader tests –** implemented are operations producing new primitives: surface tesselation with depth map, dynamic cubemapped reflections, shader generated particle system and geometry instancing by duplicating original object.
- **Vertex shader tests –** used scenes are as follows: high polygon model with many per-vertex lights (to measure polygon throughput in milions of polygons per second), displacement mapping with detailed texture and simulation of water surface with complex functions.
- **Complex tests –** stresses whole pipeline and video memory with advanced effects like HDR lighting, real-time ambient occlusion, dynamic shadow mapping and image supersampling (good method to measure memory bandwidth).

3.1. TEST RUN AND RESULTS

Application has GUI to set basic test parameters (test series, resolution, antialiasing etc.) and after test run the results are stored into XML file alongside with system informations. From those results the final score is calculated (every test has its own percentual weight calculated from frames per second, FPS). Following graph shows results of two widespread graphics accelerators from ATI and nVidia (from currently implemented tests):



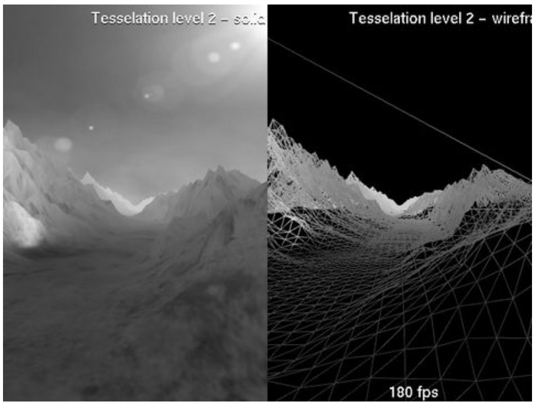
Graph 1: measured test results

3.2. Example screenshots from test application

Picture 1: HDR fillrate test

Built-in refract function with chromatic aberation
382 fps

Picture 2: Reflection/refraction test



Picture 3: Dynamic tesselation test

3600 particles
57 fps

Picture 4: GPU particles test

4. CONCLUSION

The main purpose of this application is to independently evaluate performance of modern graphic accelerators using portable OpenGL interface, because currently there is no application which can measure performance in advanced scenes under this interface. Results heavily depend also on display drivers, because many technologies used are supported only at short period of time. In the future, more tests will be available, measuring performance also in general purpose calculations through GPU. All test results can be uploaded to internet database to effectively compare different GPU's and to create statistics from these results.

ACKNOWLEDGEMENT

This work was partially supported by the BUT FIT grant FIT-S-10-2 and the research plan MSM0021630528.

REFERENCES

[1] Segal, M., Akeley K.: *The OpenGL® Graphics System: A Specification (Version 3.2 (Compatibility Profile))*

[2] Kessenich, J.: The OpenGL® Shading Language: Language Version: 1.50

[3] MSDN: Pipeline Stages (Direct3D 10): http://msdn.microsoft.com/en-us/library/ee415715%28VS.85%29.aspx