Computer Science Tripos – Part II – Project Proposal

Smart Anti-Aliasing for Virtual Reality

G. Ash, Fitzwilliam College

12 October 2016

Project Supervisor: Dr R. Mantiuk

Director of Studies: Dr R. Harle

Project Overseers: Prof R. Anderson & Prof J. Bacon

Introduction

A problem with modern Virtual Reality headsets is that they use low resolution displays to cover a huge Field of View. Graphical artefacts, such as moiré patterns and pixellated edges (jaggies), are pronounced on these displays. A good technique to ameliorate these artefacts if super-sampling, but super-sampling is often too expensive in VR devices where low latency is a requirement - to avoid simulation sickness.

Unfortunately, modern consumer headsets suffer from astigmatism because of a single lens between the viewer and the display. We can't remove these distortions without (another) anastigmatic lens, however we do currently give equal preference to image quality across the whole of the display even when the edges of the image become distorted.

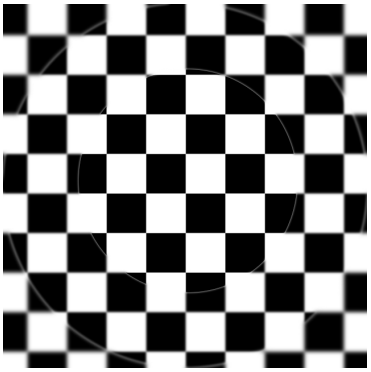

Figure 1: The center of the perceived image is sharp, while the edges get progressively blurrier

We could exploit astigmatism in single lens VR headsets by sampling more in the sharp center of the image, and sampling less as we move towards the blurrier edges (Figure 1). This project aims to extend an existing open source system to include this optimisation, and to measure the impact on both the performance of the system, and on the image quality to the user.

Starting point

Novel anti-aliasing techniques, such as subpixel-reconstruction and temporal antialiasing, are still actively being developed. I intend to build on Intel's research[1] into a hybrid raytracing/rasterizing VR renderer by creating an entirely rasterized solution that retains image quality while allowing for GPU hardware acceleration.

Free and extensible renderers exist for VR headsets, such as Unity. However I will be extending the existing open-source OSVR-RenderManager, which will allow me to easily create separate reusable demos to show off certain graphical artefacts, and, if neccesary, to modify the entire rendering pipeline. OSVR is compatible with all modern VR headsets, and supports all modern graphics APIs (OpenGL, Vulkan, Direct3D).

I have good experience in C/C++ through small projects and the IB C/C++ course. I'm familiar with the graphics pipeline and have experience in WebGL, with some experience writing vertex and fragment shaders in OpenGL. I will need to refresh my knowledge of shader programming, and will need to take some time learning most of the OpenGL API.

Resources required

For this project I shall be using my own quad-core machine with a VR capable GPU. I will also be using an Occulus Rift Development Kit 2 [3](lent to me by the Hackers at Cambridge group) for testing and user studies. Source backups will be made both to a private Github repository and MCS daily. I will use an MCS machine as a failsafe incase my machine should break.

Work to be done

The project breaks down into the following sub-projects:

1. **Setup** Fork the existing Open Source Virtual Reality-RenderManager repository. Create a daily cronjob to backup this to github and MCS. Research my chosen renderer's pipeline.

2. **Core development** Develop/modify a simple super-sampling algorithm. Extend the algorithm to allow for areas of the screen to be ignored. Further extend to seamlessly composite draws that we sample differently.

3. **Demo creation** Create a couple of OpenGL demos that highlight both moire patterns and pixellated edges, for use in visual quality experiment.

4. **Optimisation** Make use of a GPU profiler to determine any redundancy or inefficiency. Refactor the algorithm to make it easily configurable.

5. **Evaluation** Make use of benchmarking/profiling software to evaluate the algorithm with regards to performance. Perform visual quality experiment to evaluate the impact of the algorithm on image quality to the user, recruit college members across

fields to participate. Compare my approach against no anti-aliasing, and fullscreen anti-aliasing.

Success citeria

The project will be a success if I manage to do the following:

1. Improve the performance of the open source renderer with anti-aliasing enabled
2. Provide an alternative anti-aliasing technique that suffers only negligible loss in image quality to the end user.
3. Evaluate both full screen antialiasing, my selective antialiasing approach, and no antialiasing from a user perspective by constructing demos that show off artefacts masked by antialiasing.

Possible extensions

If I achieve my main result early I shall try the following alternative experiment or method of evaluation:

1. Research using a heuristic to determine salient objects or regions in the scene, and extend my algorithm to more closely resemble a foveated rendering[2] technique.
2. Further reduce the requirement to super-sample by determining which objects/samples can be shared between each eye.

Timetable

Planned starting date is 16/10/2011.

1. **Michaelmas weeks 2–4** Start project Setup, Begin refreshing knowledge on OpenGL. Research the rendering pipeline of my chosen renderer. Formulate an implementation strategy.
2. **Michaelmas weeks 5–6** Complete project setup. Test writing custom code in the renderer, start implementation of selective antialiasing algorithm
3. **Michaelmas weeks 7–8** Continue development of algorithm. Start on demo creation
4. **Michaelmas vacation** Finish development of algorithm and demos.
5. **Lent weeks 0–2** Write progress report. Generate corpus of test examples. Begin optimisation.
6. **Lent weeks 3–5** Finish optimisation, begin evaluation of image quality on users.
7. **Lent weeks 6–8** Finish user studies, start performance analysis. Write up User studies in disseration.

8. **Easter vacation:** Begin on extensions, flesh out dissertation, complete evaluation.
9. **Easter term 0–2:** Complete dissertation, proof read. Submit to DoS and supervisor for comments.
10. **Easter term 3:** Further proof reading/refactoring and submit dissertation.

References

1. Using Astigmatism in Wide Angle HMDs to Improve Rendering D. Pohl, T. Bolkart, S. Nickels, O. Grau, 2015.
2. Foveated 3d graphics. B. Guenter, M. Finch, S. Drucker, D. Tan, and J. Snyder. ACM SIGGRAPH Asia, 2012.
3. Oculus VR. Oculus Rift, 2014. http://www.oculus.com/