UNIVERSITY
of
ABERTAY DUNDEE

Honours Project Proposal

CMP401

An evaluation of the Visibility Buffer rendering technique in real-time graphics applications.

Author:
Alberto Taiuti

Supervisor:
Dr. Paul Robertson

28th October 2016

Abstract

Over the past couple of years

1 Introduction

Computer graphics is a field which is constantly in evolution. New technologies and techniques are developed every day and there is a constant push for innovation both in terms of performance and of visual quality. Over the last decade Deferred Rendering has become one of the primary choices of rendering technique for game developers due to its advantages over Forward Rendering and there have been other major publications showcasing novel techniques which propose to improve over one or more aspects of Deferred or Forward Rendering, with one example of these being Forward+ rendering.

As this area of research is extremely active, new techniques are published every year. However, one specific area of improvement being researched is how to reduce the memory footprint of the rendering framework used by an application so that more GPU memory is available for other purposes, such as storing textures or general usage buffers. This is especially relevant when put in the context of the games industry's push to achieve real-time 4K high-resolution rendering in computer games, where because of the high pixel density of the images being rendered, a large amount of the memory available on GPUs is used for storing the images used by the rendering engine, leaving less space left for the texture maps used by the 3D models which also need to be of higher resolution to avoid aliasing with the higher resolution render targets. More so, with the advent of Virtual Reality headsets and their high rate of adoption by customers, it becomes even more important to explore how to reduce the memory footprint of the available rendering techniques given that to render a 3D VR environment, the whole scene needs to be rendered twice and the results of the rendering stored in memory for both the left and right eye, basically doubling the amount of memory required by the renderer.

A solution to this problem is to simply increase the memory available on the GPU, which some hardware vendors have implemented. However this approach requires the end users to buy a new GPU and reduces the reachable audience if your game or 3D application requires such hardware.

Another solution is to develop and assess new software approaches which allow to improve the memory efficiency of 3D renderers with the currently available hardware. On this side, there has recently been an example of such technique being brought forward, named in various ways by different authors but more generally referred to as *The Visibility Buffer*.
The Visibility Buffer rendering technique proposes to reduce the number of intermediary image buffers required to render a scene at the expense of more complex real-time calculations on the GPU, thus reducing the overall memory footprint of the renderer implementation. However, on its own this is not appealing due to the more complex work done by the shaders which makes it potentially too slow for real-time applications and hence research is going towards mixing the basic Visibility Buffer with other methods in order to improve its real-time rendering speed. This technique is very novel and is yet to be fully explored, which makes it a good candidate for an in-depth research on its feasibility as a real-time rendering method when compared to consolidated ones such as Forward, Forward+ and Deferred Rendering and whether its advantages in memory footprint are met by a decent rendering speed performance.

Hence the research question for this project is: 3em *How Visibility Buffer technique compare to current real-time rendering techniques and what techniques exist to improve its real-time rendering performance?*

many of them set to improve the currently available ones, whereas only a few actually propose a new way of achieving real-time 3D rendering using the present software and hardware.