My thesis involves a new method for expanding the depth of field and simplifying geometry in real-time graphics applications. Large background scenes are rendered onto flat surfaces and then modified quickly as viewing angle and depth change.

1 Outline

Since my work applies to real-time graphics, the speed at which it can run is critical to its effectiveness. My method not only needs to be able to display background imagery at real-time speeds, it needs to be able to adapt to new scenery without drastically reducing the average frame rate or creating any large time gaps between two consecutive frames.

In addition to being constrained to time, the imagery that my method produces should not be significantly different from the imagery that would be created with a fully real-time rendered scene. In other words, the imagery produced by my method should be consistent with the "real" imagery. It should not contain any significant graphical artifacts.

2 Hypothesis

The success of my work relies on several assumptions. My method will be able to run in real-time without causing a significant decrease in frame rate over comparable real-time methods. My method will be able to display significantly more background geometry than other methods at comparable frame rates. My method will not produce any significant graphical artifacts and will produce imagery consistent with a fully real-time method rendering at the same depth.

3 Variables

The independent variables in my experiment are the scenery to display and the position of the camera in the scene. These variables determine what scenery will be displayed on the screen. The dependant variables are the frame-rate, and the images produced on the screen.
4 Measures

The only objective measurement I will be able to take is the time taken to draw each frame. This time can be represented as frames per second (fps) or simply as a measurement of time taken to draw each frame (typically milliseconds).

There are three ways that I can measure this time. First is fps measured at intervals of one second. This will represent the real-world experience of users running an application using my technique. Since fps is not necessarily an accurate representation of performance due to buffering and vertical sync, I will also be measuring the amount of time it takes to draw the scene measured in milliseconds. This value differs from fps because instead of measuring the number of times a new image is sent to the screen, I will be measuring the amount of time it takes to do the actual work of drawing and is a much more accurate representation of performance. I will be looking closely at the average time to draw the scene as well as the longest or group of longest frames to draw. It is possible that average drawing time is very good but there are a few long pauses between two consecutive frames.

5 Experiment Procedure

My experiment will consist of a set of one or two minute instrumented runs of fly throughs in different scenes. Each fly through will be tested several times using my method, an alternative method and a minimalist method that shows no background scenes.

I will provide a set of very large scenes composed of a height map decorated with a variety of textures and static models. The primary scene will be a realistic looking landscape with tree models. This is type of scene I expect my method to perform well on. I will also use several different, possibly less realistic scenes in order to reveal advantages and flaws in my method as well as the ones I will be testing against.

I will perform several different types of fly throughs on each of these scenes with each of the methods mentioned. One fly through will be a simple stationary camera. This will test my method in the best possible situation where viewing angles never change. The next type of fly through will be a realistic and continuous fly through where the camera moves at a slow speed. This is the expected real-world application of my method. The next two fly throughs are designed to push the limits of the method. One will be a very fast-moving continuous fly through each of the scenes, and the last fly through will choose a random position in the scene for the camera every couple of seconds. These fly throughs will have very rapidly changing background scenery.

For my method to be considered effective, I expect my method to perform with no less than a 50% decrease in fps over the alternative rendering method while being able to expand the depth of field at least 100% with no visual artifacts. These requirements will need to be met in the realistic scene in both the stationary and slow moving fly through.