Depth of Field Project Proposal

Objective:

To create my own implementation of the Depth of Field technique effect in computer 3D graphics using personal experience combined with other implementations and the science behind Depth of Field and its perception.

Personal Experience:

The exposure I have witnessed in regards to the Depth of Field technique mainly comes from two areas, video games and photography. This exposure is small, however this exposure will allow a wide array of approaches to the implementation of a personal Depth of Field in 3D graphics. Photography generally tackles the 'idea' behind Depth of Field without actually making objects in space appear in or out of focus. Video games, in particular, games like Call of Duty handle the visual effect of Depth of Field in an actual 3D space in real time. This real time processing is normally done with the performance in mind, which helps provide depth of field's key qualitative features in both the foreground and the background with minimal impact on total system performance or engine architecture.[1] Combining the exposure from photography and video game 3D rendering, I would like to meet halfway by using Depth of Field to render the effect in the realm of a 3D space.

Science Behind Depth of Field:

Depth of Field is the distance between the nearest and farthest objects in a scene that appear acceptably sharp in an image,[2] a real human interpretation through focusing on objects, both near and far away. The human eye focuses on an object based on that object's distance and what the focal point is at a current moment.

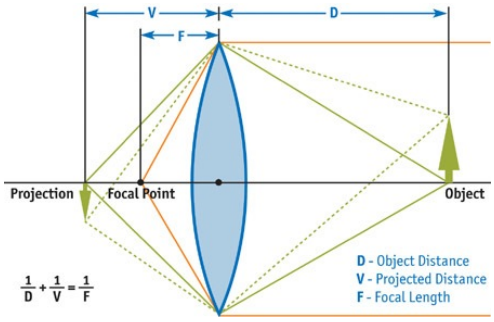

Figure 1. Focus of an object through a thin lens

Based on several factors, equations, and algorithms, this effect can accurately be projected on images and in 3D worlds. However, the accuracy behind depth of field and the performance cost behind making it seem realistic compete. Over the course of the project, I want to find a balance between accuracy and performance in order to present an appealing, yet efficient implementation of Depth of Field on 3D spaces.

Techniques of Implementing Depth of Field:

Depth of Field is usually categorized into two main methods of implementation, object space and image space. Object space methods tends to operate on a 3D space and build the effect while rendering in the pipeline. Image space methods take an image, and by utilizing the image's depth map, can determine how much to blur a pixel based on its depth in a field. Comparison of the two focus on the method's benefit(s). Object space methods tend to generate more realistic effects, with the cost of performance. Using image space methods, the rendering tends to be faster, but may yield more artifacts and less realism in comparison to object space methods. [3]

Figure 2.

Depicts a character from *Call of Duty 4: Modern Warfare* rendered with different methods of Depth of Field. These images particularly depict blurring objects in the foreground. This image also helps reinforce the idea that there is a general preference for blurring background objects as opposed to foreground objects, due to loss of quality.[4]

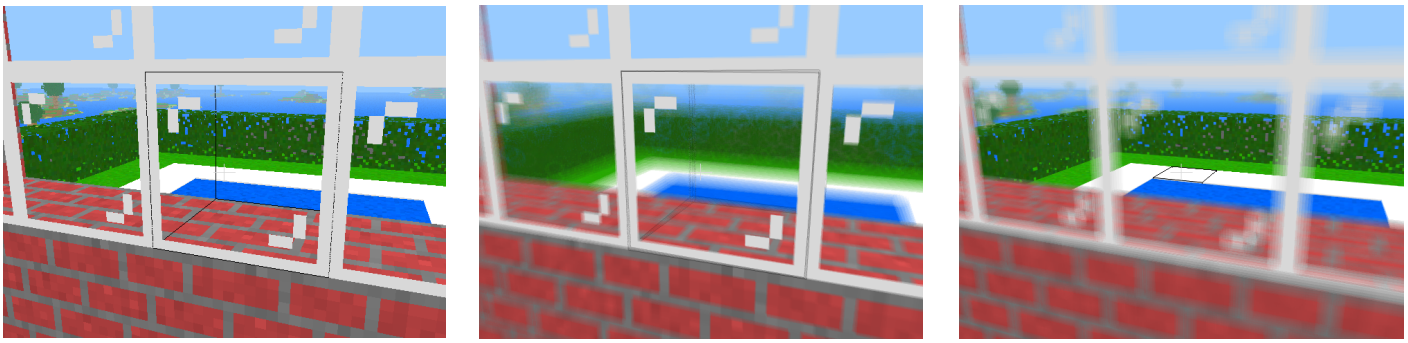

Figure 3. Simulation of Depth of Field using an accumulation buffer. Original image (left), focus on the foreground window pane (middle), and focus on the background pool (right).[5]

1. http://http.developer.nvidia.com/GPUGems3/gpugems3_ch28.html. *Depth of Field: A Survey of Techniques.* Earl Hammon, Jr. Infinity Ward.
2. http://en.wikipedia.org/wiki/Depth_of_field.
3. Barsky, Brian A & Todd J. Kosloff. *Algorithms for Rendering Depth of Field Effects in Computer Graphics*. University of California Berkley. Berkley, California. http://www.cs.berkeley.edu/~barsky/Blur/survey.WSEAS.2008.Algorithms.for.Rendering.Depth.of.Field.Effects.in.Computer.Graphics.pdf
4. http://http.developer.nvidia.com/GPUGems3/gpugems3_ch28.html. *Practical Post-Process Depth of Field*. Earl Hammon Jr. Infinity Ward.
5. http://en.wikibooks.org/wiki/OpenGL_Programming/Depth_of_Field.