Graphics Research Journal

COMP220- Research Journal

1507516

November 28, 2016

1 Introduction

This paper will be covering three different game components that went into the Graphics and Simulation project. These components are *Procedural Terrain Generation*, *Optimization of Complex Meshes* and *Illumination*.

2 Procedural Terrain Generation

2.1 An Image Synthesizer [1]

This paper by ken perlin has been influential to the generation of realistic Computer Generated Imagery (CGI) as it has cited over 1900 times and has been integrated into various software packages such as Autodesk *Maya* and *3D Studio Max* [2].

This paper describes the concept of a "Pixel Stream Editor" (PSE) which lets designers work with a high level programming environment to produce realistic CGI.

The PSE allows operations to be performed asynchronously at different pixels, which means that this computation can be performed on a Graphics Processing Unit to speed up rendering times.

However this paper has been improved upon in his more recent paper *Improving Noise* [3].

2.2 Improving Noise [3]

This paper revises some issues that arose in his previous paper that was introduced 17 years before. The original noise algorithm suffered from two defects;

- Discontinuity across coordinate-aligned integer boundaries.
- An expensive and problematic method of computing the gradient.

Both these issues were removed in the improved algorithm proposed in this paper.

2.3 Hypertexture [4]

This paper focuses on the rendering of an intermediate region between object and non-object called “Hypertexture”. Examples of this can be complex 3D textures such as hair, fur or smoke. This paper is based of Ken Perlins previous paper [1].

Hypertexture objects have no well defined boundaries, the paper proposes a function that describes how the objects texture transitions between the outside and inside of the object.

3 Optimization

3.1 Perlin Noise Pixel Shaders [5]

This paper demonstrates how they used procedural shading techniques to compute high resolution textures efficiently in real time. This means that materials like wood and stone can be generated using procedural texturing to quickly produce dynamic animated environments.

They use the perlin noise function to generate these procedural shading techniques [3]. The paper also illustrates the pros and cons of generating the noise algorithm on different graphics hardware.

3.2 The Multilevel Finite Element Method for Adaptive Mesh Optimization and Visualization of Volume Data [6]

This paper is about optimizing a 2 or 3 dimensional mesh.

Rendering large amounts of geometric primitives (i.e. cubes or spheres etc..) can be very compute intensive, this paper presents two algorithms that show significantly increased rendering quality and speed of these geometric primitives from large datasets. It does this by generating approximate meshes and then refining the meshes iteratively.

The mesh is refined based on a sequence of approximations that are adaptively generated by the algorithm. This approach can be used on very large datasets of complex meshes to generate a lower poly mesh from, i.e. a low poly approximate height map can be generated iteratively from a large topographical height map.

This method could be very useful in games to adaptively reduce the Level of Detail in models as they got further away from the camera.

Another paper that goes into more depth about this topic: [7]

4 Illumination

4.1 Illumination for computer generated pictures [8]

This paper was published in June 1975, but the principles behind it still work well for 3D video game graphics, where performance is important. However for more realistic lighting effects for 3D Design, ray tracing is a slower, but more accurate solution to complex 3D lighting [9].

Phong's paper breaks up the lighting calculations into three separate parts; *specular reflection*, *diffuse* and *ambient*.

4.2 Models of Light Reflection for Computer Synthesized Pictures [11]

This paper by Blinn builds upon Phongs paper [8]. As it has "a noticeable effect primary for non-metallic and edge lit objects" [11]. However Phongs paper is apparently better at rendering curved surfaces [11].

4.3 An improved illumination model for shaded display [10]

This paper presents a shading model that uses global information to calculate intensities, then provides extensions to the ray tracing visible surface algorithm. This paper builds upon the work done by Phong [8] and Blinn [11]. It has improved the lighting effects, however it is not perfect.

References

[1] K. Perlin, "An image synthesizer," *ACM Siggraph Computer Graphics*, vol. 19, no. 3, pp. 287–296, 1985.

[2] K. Perlin, "Noise and turbulence." [Online]. Available: `http://mrl.nyu.edu/~perlin/doc/oscar.html`. [Accessed: 27-Nov-2016].

[3] K. Perlin, "Improving noise," in *ACM Transactions on Graphics (TOG)*, vol. 21, pp. 681–682, ACM, 2002.

[4] K. Perlin and E. M. Hoffert, "Hypertexture," in *ACM SIGGRAPH Computer Graphics*, vol. 23, pp. 253–262, ACM, 1989.

[5] J. C. Hart, "Perlin noise pixel shaders," in *Proceedings of the ACM SIGGRAPH/EUROGRAPHICS workshop on Graphics hardware*, pp. 87–94, ACM, 2001.

[6] R. Grosso, C. Lurig, and T. Ertl, "The multilevel finite element method for adaptive mesh optimization and visualization of volume data," in *Visualization'97., Proceedings*, pp. 387–394, IEEE, 1997.

[7] G. F. Carey and D. L. Humphrey, "Mesh refinement and iterative solution methods for finite element computations," *International Journal for Numerical Methods in Engineering*, vol. 17, no. 11, pp. 1717–1734, 1981.

[8] B. T. Phong, "Illumination for computer generated pictures," *Communications of the ACM*, vol. 18, no. 6, pp. 311–317, 1975.

[9] P. Shirley, "Ray tracing," in *ACM SIGGRAPH 2005 Courses*, SIGGRAPH '05, (New York, NY, USA), ACM, 2005.

[10] T. Whitted, "An improved illumination model for shaded display," in *ACM Siggraph 2005 Courses*, p. 4, ACM, 2005.

[11] J. F. Blinn, "Models of light reflection for computer synthesized pictures," in *Proceedings of the 4th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '77, (New York, NY, USA), pp. 192–198, ACM, 1977.

[12] J. Döllner, K. Baumman, and K. Hinrichs, "Texturing techniques

for terrain visualization," in *Proceedings of the conference on Visualization'00*, pp. 227–234, IEEE Computer Society Press, 2000.