Graphics Research Journal

COMP220- Research Journal

1507516

November 22, 2016

1 Introduction

For my graphics and simulation project I have chosen to make a voxel based game, where the player is a light source that will light up the terrain. The terrain will be generated using perlin noise [3] to change the height of the ground.

Paper One: Perlin Noise Pixel Shaders [1]

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 enviroments.

They use the perlin noise function to generate these procedural shading techniques [2]. The paper also illustrates the pros and cons of generating the noise algorithm on different graphics hardware.

Paper Two: Improving Noise [2]

This paper is what I used to implement the perlin noise algorithm into my project. It work by calculating a random vector for each of the nearest verticies in a cube. This paper is improves upon his previous paper *An Image Synthesizer* [3].

Paper Three: The Multilevel Finite Element Method for Adaptive Mesh Optimization and Visualization of Volume Data [4]

This paper is about optimising 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 geometic 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.

Paper Four: Dual/primal mesh optimization for polygonized implicit surfaces [5]

Paper Five: Illumination for computer generated pictures [6]

This paper was published in June 1975, so it is fairly dated now, but the principles behind it still work fine for todays standsards. However there are improved lighting models, but are generally more complex.

Paper Six: An improved illumination model for shaded display [7]

Paper Seven: Texturing techniques for terrain visualization [8]

Paper Eight: Hypertexture [9]

Another paper by Ken Perlin.

Paper Nine: An Image Synthesizer [3]

The original paper by ken perlin, this paper has been cited over 1900 times.

References

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

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

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

[4] 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.

[5] Y. Ohtake and A. G. Belyaev, "Dual/primal mesh optimization for polygonized implicit surfaces," in *Proceedings of the seventh ACM symposium on Solid modeling and applications*, pp. 171–178, ACM, 2002.

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

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

[8] 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.

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