

COMP220: Research Journal

COMP220

1507866

November 27, 2016

This essay will look at a lighting in computer graphics and using particle effect systems to simulate dynamic objects such as fire.

1 Introduction

The intention of this paper is to look at computer graphic lighting and particle effects. For lighting it will look at Phong's model for computer generated images from 1975 [1]. For particle effects it will look at Reeves technique for simulating irregular and dynamic objects [2]. It will also look at the use of particle effects systems in simulating realistic fire, fire-works and water.

2 Lighting

Phong originally proposed a lighting model for more realistic computer graphic lighting [1]. Previously computer graphics only used diffuse lighting, Phong expanded this into separate components and introduced highlights [3, 4, 1]. Phong's model splits computer graphic lighting into three components which

are specular, ambient and diffuse. Lighting models previous to this could take a long time to calculate or did not give realistic results [1]. Blinn refers to Phong's model as a more realistic model and says it improves the appearance of curved objects [5]. However Blinn and Newell suggested the Phong model was not suitable for reflections on planar polygons [4]. Another issue with Phong's model is that it is designed to be used on distant light sources and is not suited to light sources within the scene [6]. Extension to Phong's model may be able to remove those issues [4, 6].

3 Particle Effects

Reeves first proposed the idea of a particle system in 1983 [2]. A particle system was to be used to represent dynamic or irregular objects such as clouds, smoke and fire [2, 7, 8]. A particle system would create new particles and make older particles "die". Reeves' particle technique was one of the first to be proposed [9]. There are now alternative methods for simulating dynamic and irregular phenomena as Reeves original technique is now over 30 years old [8, 9].

Huang *et al* outline what the main properties of a particle and particle effect system are [10]. A particle system should initialise, emit, render and finally eliminate particles. This corresponds with Reeves as he says particle should be "born" and "die", both papers say that particles should have limited lifespans. The properties a particle should have are spatial location, velocity, survival time, shape, size and a gravity factor [10].

3.1 Particle Effects for Fireworks and Fire Simulation

Fireworks are a irregular and dynamic objects so are suited to being simulated using a particle system [7, 2]. There are many other papers that look at using

particle effects for fireworks [7, 11, 12]. Lei and Wang looked using a particle system to simulate fireworks [7]. Like Reeves they mention particles need to have a lifetime and can be used to create dynamic objects. Their method also takes other forces into account such as simulated temperature, wind speed and physics and what effect this would have on the particle[7]. Lei and Wang’s method does not follow Huang *et al*’s model as all particles are created at the same time. However, the particles are similar to Huang *et al*’s model, they have attributes such as position, velocity and lifespan.

Another use for particle effects is simulating fire [9]. There are many ways to simulate fire with computer graphics [13, 14, 15]. Fire is complex to simulate therefore most methods simplify fire often to just the yellow flame component [9]. Pegoraro and Parker use particle effects to simulate flames but researched fire at a molecular level first to make their simulation more realistic. Their physically based rendering model takes biological and chemical factors into account. Their simulation also warps the simulation background to simulate refraction. Despite producing realistic results Pegoraro and Parker’s method takes 40 to 80 seconds to produce a flame, which would not be viable in a video game [9].

3.2 Particle Effects for Water Simulation

Water is another dynamic object that can be simulated using particle effects. Three components needed for realistic water simulation is correct atmospheric conditions, wave generation and light transport through the water body [16]. Like Pegoraro, Wan *et al* look at photo-realistic simulation and uses simulated refraction to improve appearance and reflections [9, 17]. Shi and Wang mention Premoze and Ashikhmin’s complex illumination model for water but not whether they use it [16].

Shi and Wang used particle effects to simulate a waterfall [18]. Like Lei, Shi and Wang used physics to make the particle movement more natural [7, 18]. Collision detection was also used to make the water move more fluidly. Shi and Wang’s definition for a water particle is “a particle with initial position, initial velocity, life cycle, quality, size, shape, falling height” [18, p.1]. Many of those properties match the properties suggest by Huang *et al.*

Using a clustering tree can reduce computing power as it reduces the amount of particles that need to perform a collision detection [18]. If one particle collides others generated afterwards do not need to perform the check [18].

4 Conclusion

In conclusion Phong’s model for lighting is still widely used however it requires extensions to make it more realistic for reflections. Reeves proposed the original technique for particle effect systems. There are now many iterations of this techniques. Many of these other versions take other factors such as gravity and collisions into account to make them more realistic.

References

- [1] B. T. Phong, “Illumination for computer generated pictures,” *Commun. ACM*, vol. 18, pp. 311–317, June 1975.
- [2] W. T. Reeves, “Particle systems: A technique for modeling a class of fuzzy objects,” *ACM Trans. Graph.*, vol. 2, pp. 91–108, Apr. 1983.
- [3] J. T. Kajiya, “Anisotropic reflection models,” *SIGGRAPH Comput. Graph.*, vol. 19, pp. 15–21, July 1985.

- [4] J. F. Blinn and M. E. Newell, “Texture and reflection in computer generated images,” *Commun. ACM*, vol. 19, pp. 542–547, Oct. 1976.
- [5] J. F. Blinn, “Models of light reflection for computer synthesized pictures,” *SIGGRAPH Comput. Graph.*, vol. 11, pp. 192–198, July 1977.
- [6] T. Whitted, “An improved illumination model for shaded display,” in *ACM SIGGRAPH 2005 Courses*, SIGGRAPH ’05, ACM, 2005.
- [7] S. Lei and W. Wang, “Study on algorithm for fireworks simulation based on particle system,” in *Computer Science and Network Technology (ICCSNT), 2013 3rd International Conference on*, pp. 231–234, Oct 2013.
- [8] T. Nishita and Y. Dobashi, “Modeling and rendering of various natural phenomena consisting of particles,” in *Computer Graphics International 2001, CGI ’01*, (Washington, DC, USA), pp. 149–158, IEEE Computer Society, 2001.
- [9] V. Pegoraro and S. G. Parker, “Physically-based realistic fire rendering,” in *Proceedings of the Second Eurographics Conference on Natural Phenomena*, NPH’06, pp. 51–59, 2006.
- [10] X. Huang, D. Zhao, N. Li, and R. Xu, “Design and application of general data structure for particle system,” in *Computer Science and Electronics Engineering (ICCSEE), 2012 International Conference on*, vol. 1, pp. 489–492, March 2012.
- [11] W. Dong, X. Zhang, and C. Zhang, “Firework simulation based on particle system in virtual scene,” in *Proceedings of the 2010 International Conference on Multimedia Communications, MEDIACOM ’10*, (Washington, DC, USA), pp. 39–41, IEEE Computer Society, 2010.

- [12] S. Zhang, “Fireworks simulation based on particle system,” in *Proceedings of the 2009 Second International Conference on Information and Computing Science - Volume 01*, ICIC ’09, (Washington, DC, USA), pp. 187–190, IEEE Computer Society, 2009.
- [13] F. Bridault-Louchez, M. Leblond, and F. Rousselle, “Enhanced illumination of reconstructed dynamic environments using a real-time flame model,” in *Proceedings of the 4th International Conference on Computer Graphics, Virtual Reality, Visualisation and Interaction in Africa*, AFRIGRAPH ’06, pp. 31–40, 2006.
- [14] P. Beaudoin, S. Paquet, and P. Poulin, “Realistic and controllable fire simulation,” in *Proceedings of Graphics Interface 2001*, GI ’01, pp. 159–166, 2001.
- [15] A. Lamorlette and N. Foster, “Structural modeling of flames for a production environment,” *ACM Trans. Graph.*, vol. 21, pp. 729–735, July 2002.
- [16] S. Premoze and M. Ashikhmin, “Rendering natural waters,” in *Proceedings the Eighth Pacific Conference on Computer Graphics and Applications*, pp. 23–434, 2000.
- [17] D. C. Wang, W. G. Wan, and C. Hua-Jie, “Research and implementation of real-time rendering algorithm for water surface based on programmable graphics hardware,” in *2007 IET Conference on Wireless, Mobile and Sensor Networks (CCWMSN07)*, pp. 375–377, Dec 2007.
- [18] L. Shi and W. Wang, “The status quo and realization of waterfall simulation based on particle system,” in *2015 4th International Conference on Computer Science and Network Technology (ICCSNT)*, vol. 01, pp. 91–93, Dec 2015.