Simulation of Smoke to Improve Unity 3D Game Engine Particle System based on FDS

XU YONGZHE[1] , Kyunjoo Lee[2] , Eunju Kim[1] , Jaesug Ki[1,*] , Byungsoo Lee [1,*]

[1]Department of Computer Engineering, Universty of Incheon, Korea
[2]MaxOn Soft, Co., Ltd, Korea
yongzhexu@hotmail.com, lisun@maxonsoft.com, leebone28@hanmail.net,
{skyblueki, bsl}@incheon.ac.kr

Abstract. In this paper, our proposed method based on FDS (Fire Dynamic Simulation) outputs the data the Excel file, which improves a game engine for effective smoke and the fire model. The Excel file coordinates in the Unity 3D game engine's particle system, coordinates in the original particle system. After computing the coordinates of Unity 3D simple particle moving location boundary, make them flow like simulation's real world smoke fluid. Recent game engine computer graphic developers usually control resources focused on visualization smoke and the fire model. But in the fire simulation, the smoke simulation game or firefighter simulation game part requires a different approach. In this paper, we focus on smoking location boundary. The proposed method can change particles more accurately than other real time fluid dynamics graphics method.

Keywords: Computer Graphics, Fire Dynamics Simulator (FDS), Smoke, Particle System, Fluid Dynamics, Unity 3D, Fire Simulation.

1 Introduction

The first particle systems in computer graphics used the generation of the explosion of a planet for special film trick effects. In 2001 Stanford University researcher Ronald Fedkiw, Jose Stam, Henrik Wann Jesen simulated smoke, by using 300,000 particles [8]. That was the first computational particle system simulation smoke used fluid dynamics. Now researchers use 2~3 million particles simulation smoke. From this simulation or game engine, FDS needs a long time to simulate real-world fire and smoke. Real time simulation and debugging 3D fluid solver are very hard tasks. In this paper, proposed smoke coordination which is based on FDS results in smoke data, by getting the smoke boundary from FDS output Excel file. After computing smoke boundary coordinates, graphic card does not need computing coordinates. Therefore, computer performance was improved.

2 Related works

From the Opacity Mapping demo, Particle system is the point of 3D space that is determined by position x, y, z and orientation given by three vectors x, y, z. NVIDIA smoke is based on fluid physics mechanism [1][2]. The approach from the 2d fluid simulation and implementation of 3D fluid simulation by Harris (2004) [13] was very challenging, and difficult for real time simulation. The form (1) is a Partial differential equations (PDEs) function.

$$\sim\ = (,). \qquad (1)$$

The x is changed by wavelet f, which may be depended on x and t. [14]

$$^{n+1} = \qquad {}^{n} + ({}^{n}, t^{n}). \qquad (2)$$

Harris 2D fluid simulation doesn't have an unique coordinate z [7]. Improved NVIDIA Box of smoke demo is added to function z from the form (3) [3].

$$\Delta f = \frac{^{2}}{\partial x^{2}} + {}^{2}{}_{\partial^{2}} + {}^{2}\ {}_{2} = 0. \qquad (3)$$

Set the world space coordination. Using the Function is the problem to cause more graphics card resources in running time.

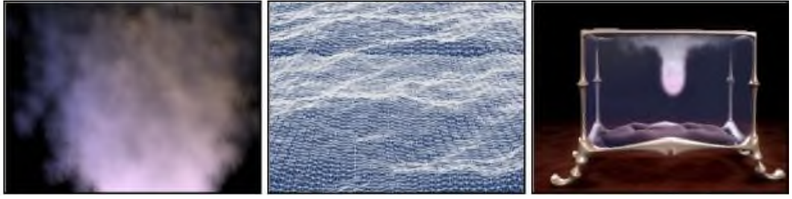

Fig. 1 NVIDIA Opacity Mapping demo, Harris 2004 2D fluid simulation, NVIDIA Box of smoke demo

3 FDS coordinate and Unity 3D particle system

From this section, we discuss about FDS, Unity 3D particle system, designed program codes. Firstly, we propose a method constructed in game's environment boundary, Secondly, FDS simulation environment about fire and wind [4], Thirdly, we use FDS output excel file to find smoke and fire fluid boundary coordinates, finally Unity 3D game engine particle system loads these particle coordinates. In figure 2, simulation x, y, z = 27m mesh, small room's x, y, z = 9m, the left picture shows simulation start, and the right picture displays fire burning. From right picture, white points are smoke detector device for detection smoke density. Users can set smoke detector device (white points) of coordinates in a simulation scene. The smoke detector device coordinate can be written into smoke density data in every second (user can change the time). Unity 3D game engine's particle system is a specialized in a method of 3D object [10]. From the FDS output file, we can know smoke density and also know smoke boundary coordinates. From program design part, Encapsulation is indispensable in program codes. The setting of the particle properties of program codes is cited from reference [6][9]. The particle's life time (particle moving time) follows location and boundary. The Source codes we used are required in changing c# in which Unity 3D game engine interface is made. Unity 3D Program code uses Particle System interface method 'position.x' to change position of particles [13]. If we know the timing of moving particles, Particle System life time can be implemented through location and boundary. It is possible to get the particle coordinate, by knowing life-cycle and current-time.

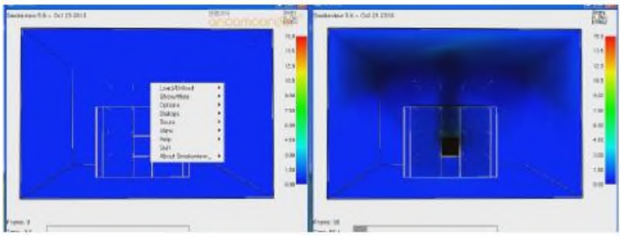

Fig. 2 FDS Simulation fire model

4 Simulation result

Figure 3 shows our method applied in Unity 3D. The particle follows program code coordinates and boundaries. Particle coordinates in life time follow program code coordinates. The left of picture 3 shows smoke in 3D space mesh. The center picture shows that created particles follow boundaries. The right picture shows the change of texture in life time of particles [12]. Figure 4 shows moving particle and computing coordinates in next time [11]. Figure 4 illustrates Life time of the particles which move from 'Start' to 'End' and the particles disappear when arrive at End-position. Each space send particles to next Random Function space coordinate.

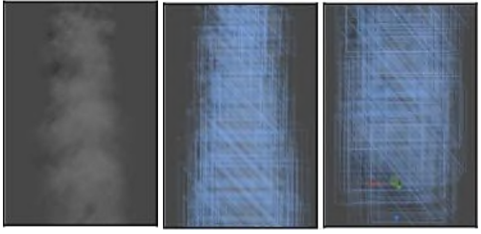

Fig.3 Unity 3D simulation result

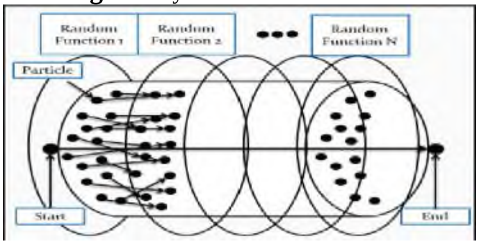

Fig.4 Particle in life-cycle compute coordinate

5 Conclusions and Future Work

The excel file based on FDS smoke detector device coordinates compute smoke boundary coordinates. The result of our research improves the performance better than other visualization smoke particle system, because the graphic card can skip the computing the particle coordinates. This method is possible to be used in steam effect not only smoke particle system but also fire model system. This approach is different from common simple particle system visualization. Future work we plan is to change particle boundaries in more detail from CFD(computational fluid dynamics).

Acknowledgement

This paper was Researched (2012) The Next Generation Fire protection & Safety Core Technology Development program of National Emergency Management Agency (NEMA) of Republic Korea.

References

1.Eberhardt, B. et. al.: A Fast, Flexible, Particle-System Model for Cloth Draping. IEEE, Computer Graphics and Application, 16 (50:52-59). September 1996.
2.Jansen, J., Bavoil, L.: Fast rendering of opacity-mapped particles using DirectX 11 tessellation and mixed resolutions. Opacity Mapping SDK White Paper NVIDIA Graphics SDK 11 document.
3.Keenan, C., L, lamas Ignacio., T, Sarah.: Real Time Simulation and Rendering of 3D Fluids. GPU GEMS 3 2007.
4.M, Kevin., K, Bryan., H, Simo., F, Jason.: Fire Dynamics Simulator (version 5). NIST Special Publication 1019-5. NIST.
5.Lander, Jeff.: The Ocean Spray in Your Face. Game Developer Magazine July (1997): 13-20.
6.Luna, Frank.: 3D Game Programming with DirectX 9.0. Plano. Texas: Wordware Publishing, 2003.
7.J. Harris Mark.: Fast Fluid Dynamics Simulation on the GPU, GPU GEMS, chapter 38.
8.F, Ronald*., S, Jos., J, Henrik Wann.: Visual Simulation of smoke. Proceedings of the 28th annual conference on Computer graphics and interactive techniques ACM 2001.
9.Reeves, William.: Particle Systems: A Technique for Modeling a Class of Fuzzy Objects. ACM Transactions on Computer Graphics Volume 17, Number 3 (1983):359 – 376.
10. Unity Team.: Unity Manual. Particle System, positioning Game Objects.
11. D. Wensheng, Z. Xinyan, Z. caijun .: Smoke Simulation based on particle system in Virtual Environments, Multimedia Communications (Mediacom), 2010 International Conference on
12. Hae Won Byun, Hye Moon Jung.Drawing Style Capture for Cartoon Rendering, International Journal of Multimedia and Ubiquitous Engineering Vol. 8, No. 1, January, 2013
13. D. Wensheng, Z. Xinyan, Z. caijun.Smoke Simulation based on particle system in Virtual Environments, Multimedia Communications (Mediacom), 2010 International Conference on
14. Sheng feng He, Hon-Cheng Wong, Un-Hong Wong.An Efficient Adaptive Vortex Particle Method for Real-Time Smoke Simulation, IEEE 2011 12th International Conference on Computer-Aided Design and Computer Graphics