Design and Implement of a CUDA Based SPH Particle System Editor

Xianjun Chen[1,3] and Yongsong Zhan[1,2]

[1] Guangxi Key Laboratory of Trusted Software, Guilin University of Electronic Technology, Guilin, Guangxi, 541004, China
[2] College of Computer Science and Engineering, Guilin University of Electronic Technology, Guilin, Guangxi, 541004, China
[3] Information Engineering School, Haikou College of Economics, Haikou, Hainan, 570203, China
hingini@126.com

Abstract. Particle system is the key technique for special effects visualization in many computer graphics applications such as computer games and animations. In this paper, we propose a novel approach for simplifying particle effect simulation in a particle system editing environment, where a SPH (Smoothed Particle Hydrodynamics) solver is used for particles movement simulation. We leverage the parallelism computing architecture of NVIDIA CUDA to power up the SPH solver implementation. Experimental results show the robustness and efficiency of the proposed system for real time graphics applications.

Keywords: Particle system editor, SPH, CUDA, Special effect.

1 Introduction

Particle systems have long been recognized as an essential building block for visualizing various types of effects, such as smoke, fire or dust, in numerous of Computer Graphics (CG) applications such as computer games, animations and virtual reality, etc. With the dramatically increasing power of modern graphics process hardware, the particle system based special effects in most real-time applications have been extremely complex. As a consequence, it is difficult to control and direct the particles motion optionally and freely.

Smoothed particle hydrodynamics (SPH) is a computational method used for simulating fluid flows. It has been used in many research fields, including astrophysics, ballistics, volcanology, and oceanography. It is a mesh-free Lagrangian method, needing a large number of particles to achieve smooth surfaces and to resolve fine-scale surface details, and the resolution of this method can easily be adjusted by simulation parameter setting.

Nowadays, GPU have been designed with a parallel throughput architecture that emphasizes executing many concurrent threads slowly, rather than executing a single thread very quickly. To utilize the computation power of modern GPU, a novel

Y. Tan, Y. Shi, and Z. Ji (Eds.): ICSI 2012, Part II, LNCS 7332, pp. 465–472, 2012.

technology called CUDA is released by NVIDIA as a software development kit (SDK), which allows recent NVIDIA graphic cards to be programmed for general-purpose tasks using C++ in a parallel schema, with extensions to manage the interaction between the card and the host machine. CUDA gives developers access to the virtual instruction set and memory of the parallel computational elements in CUDA-enable GPU. CUDA-enabled graphics cards are equipped with multi-core process units with both local and shared memory. Local memory is read/write per-thread, at the mean time shared memory exposed by CUDA is a fast common memory region (up to 48KB per Multi-Processor) that can be shared amongst threads, and can be used as a user-managed cache, enabling higher bandwidth than is possible using texture lookup. CUDA allowing the programmer to define functions for solving computational tasks is called kernels, which will be executed by many different CUDA threads in parallel.

In this paper, a novel CUDA based approach for particle system editor is proposed to produce complex special effects for computer games. An editing environment is established for special effects simulation controlled by animators freely, and a SPH solver is employed for simulating the particles motion. Moreover, our system leverage the powerful computation ability of NVIDIA CUDA GPU architecture, a novel technology for solving general purpose parallel computing problems for accelerating the simulation of the SPH solver.

2 Related Work

Particle system is a method of modeling a class of fuzzy objects which have undefined form and are usually in constant motion [1]. Lamorlette [2] simulated the flame with procedure noise and Kolmogorov turbulence noise, which was governed by artificial wind field and buoyancy. Flows built up from a superposition of flow primitives all have the disadvantage that they do not respond dynamically to user-applied external forces. To simulate the spread of flame, a particle set was used by Perry [3] to define the flame boundary, where new particles should be added periodically. Dynamical models of fluids based on physical equations were first implemented in two-dimensions. Gamito [4] used a vortex method coupled with a Poisson solver to create two-dimensional animations of fluids. Foster [5] clearly showed the advantages of using the full three-dimensional Navier-Stokes equations in creating fluid-like animations. Stam [6] proposed a stable algorithm that used both Lagrangian and implicit methods to solve the full Navier-Stokes equations. Simulating fluid motion using SPH was introduced in astrophysics [7] and successfully used in computer graphics to simulate a few thousand particles at interactive frame rates [8]. On the other hand, recently, parallel technology has become more and more popular for its powerful computing capacity, nowadays parallel computing can be conducted in ordinary PC. Although GPU are mainly designed for fast polygon rendering, graphics hardware has been extended to various applications of general-purpose computations [9]. Harris [10] had implemented on GPU the coupled map lattice, and had simulated cloud dynamics using partial differential equations. The capability of GPU to simulate SPH was demonstrated by