Simulation Modelling Practice and Theory 77 (2017) 32–48

Contents lists available at ScienceDirect

Simulation Modelling Practice and Theory

journal homepage: www.elsevier.com/locate/simpat

Lightweight particle-based real-time fluid simulation for mobile environment

Jin-Chun Piao[a], Jue-Min Lu[a], Chung-Pyo Hong[b], Shin-Dug Kim[a,*]

[a] Department of Computer Science, Yonsei University, 50, Yonsei-ro, Seodaemun-gu, Seoul, 03722, South Korea
[b] Mobile Communications, LG Electronics, South Korea

ARTICLE INFO

Article history:
Received 23 November 2016
Revised 6 May 2017
Accepted 8 May 2017

Keywords:
Mobile device
Real-time fluid simulation
SPH algorithm
Physics engine
3D effect

ABSTRACT

This paper presents a real-time lightweight fluid simulation based on a particle fluid technique developed for mobile environment . The Bullet physics engine and smoothed particle hydrodynamic (SPH) fluid algorithm will be used for our lightweight fluid simulation. First, we describe an advanced collision detection mechanism that will be used. By using this method, less computational resources are required. Secondly, we present a simplified SPH algorithm where nearby particles are grouped together to minimize the number of calculations. By decreasing the number of particles, an improved computational performance is expected. Finally, the ARM NEON based parallel computing technique was enabled to reduce execution time by lowering the number of arithmetic instructions. Several experiments are carried out where the experimental results indicate the first technique led to a 50% improvement in performance. The second technique provided a 17% overall improvement. The third technique delivered a performance improvement within the range of 26%–40%. Overall, the experimental results show that the proposed techniques provided an accumulative performance improvement of approximately 120% for all applied methodologies.

© 2017 Elsevier B.V. All rights reserved.

1. Introduction

Simulating natural phenomena has become one of the trends in computer science. Specifically developing real-time simulations in some applications has become one of the methods used for interactions between machines and humans. In mobile devices, only 2D fluid effects are used because of the relative lack of computing power available. Fluid simulations using real-time 3D are only used in PC environment since performance bottlenecks occur in the mobile environment. In this research, we will develop a 3D lightweight, particle-based real-time fluid effect for the mobile environment. The use of the Bullet physics engine and an SPH algorithm allows for implementation of a, particle-based fluid simulation based on physical rules. OpenGL ES was used to render objects on the mobile screen. OpenGL ES is a subset of the OpenGL computer graphics application programming interface (API) for rendering 2D and 3D computer graphics, and is a widely used graphics API for mobile applications.

Typically, fluid simulations are employed for a PC or other similar environments where significant computer hardware resources are available that allow for the use of optimization techniques. As demand for powerful graphics performance and

* Corresponding author.
E-mail addresses: kumcun@yonsei.ac.kr (J.-C. Piao), jmlu0619@yonsei.ac.kr (J.-M. Lu), hulkboy@yonsei.ac.kr (C.-P. Hong), sdkim@yonsei.ac.kr (S.-D. Kim).

http://dx.doi.org/10.1016/j.simpat.2017.05.003
1569-190X/© 2017 Elsevier B.V. All rights reserved.

quality increases, display of fluid simulation output on a mobile platform does not meet user expectations. There are many generally used software optimization methods focused on graphics. For example, we could use a graphics processing unit (GPU) to rapidly manipulate the graphics and alter memory to accelerate the creation of images. Alternately, we can simplify the internal algorithm. To overcome the limitations of mobile platforms, several optimization methods were considered as part of this research. First, the entire project is divided into three modules. Experiments were then conducted to find bottlenecks such as lower performance or graphics quality. Then a new algorithm and mechanism to enhance the performance of the modules is presented.

For the purposes of this research, a lightweight fluid system was considered to be composed of three modules. The first module was the Bullet physics engine component, which is an open source program with robust computational capabilities. As a real-time physics engine library, it can provide an approximate simulation of certain physical systems, such as rigid body dynamics, soft-body dynamics, and fluid dynamics. The physics engine assists in creating a physical rules enabled world for simulation purposes. The second module is the SPH algorithm module, which is one of the most commonly used fluid algorithms. The algorithm module uses a significant number of particle objects to represent a real-time fluid. SPH can calculate properties such as density, pressure force, internal force, and external force of every particle. Motion states are calculated from the forces of the particles based on the flow of particles creating the fluid effect. The last module is based on mathematical operations, which require a large proportion of the resources used in a graphics related application.

In this paper, we demonstrate a method for improving the performance of a fluid simulation application for the mobile environment. Three methods will be illustrated with the goal of improving performance. First, we suggest an advanced collision detection mechanism in Bullet physics engine, which can eliminate a number of nonessential computations. Secondly, we simplify the SPH fluid algorithm, using an approximate SPH algorithm that groups nearby particles to reduce CPU execution time. The particles are considered to have a similar motion state. Hence, they are grouped into one particle. Some properties regarding the motion state are calculated and simultaneously assigned to decrease the computing overhead. In the Bullet physics engine, a significant number of arithmetic calculations are executed. The ARM NEON will help decrease the computing time by executing several calculations using only one instruction.

2. Related work

In the past, real-time fluid simulations were only considered to be capable of working in a PC environment due to the hardware and resource requirements. However mobile devices, such as smartphones and tablet PCs, are now more popular. Therefore, the demand for simulating high performance, real-time fluid effects in a mobile environment has grown. Based on previous works prepared by other researchers, many open source libraries are available for developers [1–3] including physics engines and real-time fluid algorithms. A way to achieve real-time simulation in a mobile environment by simplify was proposed [4,5]. It was not physically accurate, but it has the advantage of being easy to modify in real-time.

The use of a physics engine ensures the motion of objects in a virtual world act in accordance with the physical rules of the real world. The main functions of the physics engine include collision detection, calculating various forces, and motion state. Physics engine also have a number of utilities. For example, an unmanned vehicle simulator uses a physics engine to simulate the use of unmanned vehicle [6], a surgery simulator using a physics engine is of great value for medical training [7], and physics simulation in augmented reality is also possible [8]. A number of physics engines have been developed including commercial and open source engines. Among them, the most frequently used physics engines are PhysX, Havok Physics, Bullet Physics, and some frameworks of significant functions such as ODE, Box2D. Havok [9] was developed by the Irish company Havok, and is a physics engine which is now a world leader in physics middleware. Havok offers a physics engine component and associated functions which can be used to develop games, simulators, etc. PhysX [10] is the proprietary real-time physics engine middleware SDK; PhysX was acquired by NVidia and is one kind of technique that grafts PPU and CUDA in order to accelerate hardware. The Bullet physics engine [11] supports discrete collision detection and simulation of rigid-body and soft-body. Despite the Bullet physics engine library being an open source, it offers similar functions as other paid libraries.

Real-time fluid techniques can be divided into three methods. First, a procedural technique is good at exhibiting an endless lake or ocean which can also include a sine wave function. The disadvantage, though, is that the difficultly in implementing reciprocal action when object falls through the fluid. The second method is the height field technique. The height field technique allows for a continuous water surface such as a shallow pond or lake. The amount of computing is diminished as it uses 2D effects to represent a 3D fluid surface. However, the height field fluid technique does not allow for display of breaking waves. The final method is the particle based fluid technique that uses particles to represent a fluid. If particles such as water-drops were used, the required computational time would slow down the simulation. Therefore to reduce the number of particles, a particle is used to represent a specific area of a fluid which is then grouped with other particles. The particles maintain a certain distance from each other during the computation. This allows for displaying fluid motions such as splashing and spray.

SPH is one of the most widely used Lagrangian particle based fluid simulation techniques. The SPH algorithm was first developed for solving astrophysical problems [12]. The SPH algorithm was introduced into the computer graphics field by Stam and Fiume to model fire and other natural phenomena such as gas [13]. SPH was also used to simulate smoke, ballistic events, and volcanology since the algorithm is flexible enough to be used in all types of simulation. Fluid simulation with free surfaces based on SPH algorithm was firstly developed by Muller [14]. The SPH algorithm is an interpolation technique

Download English Version:

https://daneshyari.com/en/article/4962640

Download Persian Version:

https://daneshyari.com/article/4962640

Daneshyari.com