International Journal of Future Generation Communication and Networking
Vol. 13, No. 3s, (2020), pp. 239–243

Performance Improvement and Efficiency Comparison of 3D Game using High Performance Computing

Rohit Jadhav, Ajinkya Patil, Sahil Raj, Rahul Kumar, Professor S.G. Pawar

Computer Engineering Department, Final Year Engineering SavitriBai Phule University

Abstract

Performance improvement is an important task for achieving higher frame rate for various 3D Gaming and high-quality video service providers. We are trying to improve performance of 3D Open World Game. When we improve quality of a game, performance suffers. But why performance suffers when we improve quality. The answer to that question is, quality increases then the detailing of that game/scene increases which results in increase in mathematical computation per vertex, this happens 60 times per second (which is 60FPS), so that rendering that image on Frame Buffer takes much time which results in Frame Drop which hurts performance. Hence, we are trying to minimize the rendering time required per frame/scene on Frame Buffer by shifting heavy mathematical calculations on GPU. After shifting the mathematical calculations required per frame on GPU using High Performance Computing technique OpenCL to increases Frame Rate.

Keywords: Frame Rate, FPS, Frame Drop, Frame Buffer, GPU, OpenCL.

I. INTRODUCTION

Simulation is used in many contexts, such as simulation of technology for Performance tuning or optimizing, safety engineering, testing, training, education, and video games. Often, computer experiments are used to study simulation models. Simulation is also used with scientific modelling of natural systems or human systems to gain insight into their functioning, as in economics. Simulation can be used to show the eventual real effects of alternative conditions and courses of action. Simulation is also used when the real system cannot be engaged, because it may not be accessible, or it may be dangerous or unacceptable to engage, or it is being designed but not yet built, or it may simply not exist. Key issues in simulation include the acquisition of valid source information about the relevant selection of key characteristics and behaviours, the use of simplifying approximations and assumptions within the simulation, and fidelity and validity of the simulation outcomes. Procedures and protocols for model verification and validation are an ongoing field of academic study, refinement, research and development in simulations technology or practice, particularly in the work of computer simulation Since 1993 the Development of Games and Simulations with better quality using Graphics Rendering Pipeline concepts of OpenGL. OpenGL was introduced by Silicon Graphics Inc. in 1992 and was used in various fields that required complex visualization of 2D and 3D images such as CAD, Virtual Reality, Flight Simulation and scientific visualization. It was also very obvious to enthusiasts that OpenGL could be used to render graphics in video games as well. As the time goes the quality of Graphics is improved realism in Games. For realistic effect it takes more time to render Frame/Scene on Frame Buffer.

ISSN: 2233-7857 IJFGCN

II. LITERATURE SURVEY

Literature survey tells about the description of this project. It gives an idea about how the project is distributed in Literature parts and the techniques that will be used to implement the project. This chapter includes the related work studied in relation with this project. These papers are close to the objectives of the project and the observations of these research papers are analysed in the project

Gang Xie, Ya-lin Zhang et. al. [2].Problem of programming for heterogeneous computer systems consist of CPUs and various accelerating devices such as GPUs. They introduce a few of the most popular models for heterogeneous parallel programming, including OpenCL (Open Computing Language), CUDA (Compute Unified Device Architecture). Heterogeneous systems are becoming more and more important because of their computation throughput and performance per watt advantage over homogeneous systems based only on CPUs. They use coprocessors or accelerators with different instruction sets and different performance characteristics. They are optimized to work from their own memory for different algorithms and applications. These devices typically exhibit internal parallelism, and execute asynchronously and concurrently with the host processor.

Wai-Kiu Lee and Rocky K.C. et. al. presented aspect of mathematical modelling that is necessary for successful development of modelling and simulation.[3]. The simulation is implemented on GPU (using CUDA). In this paper we are considering the Automatic Flight Control Simulation example. In this case the task of the automatic control of aircraft (system) is one of the basic tasks consists of defining the flight, the mechanical properties of aircraft and determining selected parameters. Here five types of flights (and conditions) are considered for automatic control: 1) Aircraft flight along a specified trajectory, 2) Target-oriented navigation, 3) Interruption of aircraft flight path, 4) Control of rocket along a space trajectory, 5) Aircraft take-off. Due to the development of GPU computing, the mathematical flight modeling offers a variety of simulation of flight phases. And GPU tends to extend the capabilities of these phases. As in end the aim of paper is to offer the data about flight dynamics and simulating automated aircraft control while the GPU is employed. That is to say it describes in details the technology to perform the simulation on the GPU (using CUDA).

Lag refers to the delay between a player's input and the corresponding output in games is presented by Peter Kvasnica, Igor Kvasnica et. al. [4]. In the commonly used client-server architecture, factors responsible for lags come from either the client, the server or the network between them.This study evaluates the impacts of two sources of lag stick rate and vertical synchronization (VSync), and the effectiveness of two common lag compensation techniques: lag compensation1 and interpolation by measuring the player's shooting accuracy, which is defined by the number of shots hit divided by the number of shots fired, under different configurations. To our best knowledge, this is the first study to evaluate these four issues in multiplayer games. This paper presented the results of experiments which investigated the impacts of two sources of lag: tick rate and vertical synchronization (V-Sync), and the effectiveness of two lag compensation techniques: lag compensation and interpolation.

Marcelo P. M. Zamith, Esteban W. G. Clua et. al. [5] discussed a new game loop architecture concept that employs graphics processors (GPUs) for generalpurpose computation (GPGPU). A critical issue in this field is the concept of process distribution between CPU and GPU. The presented architecture consists in a very efficient model for distribution, and it was mainly designed to support math and physics on the GPU, but any kind of generic computation can be easily adapted. Balancing the load between CPU and GPU is an interesting approach for achieving a better use of the computational resources in a game or virtual and augmented reality applications. Collision detection among rigid bodies was the task chosen to validate the proposed model. Processor time after 500 collision was 1.8967000 seconds for the GPU versus 35.2438000 seconds for the CPU

ISSN: 2233-7857 IJFGCN

III. RELATED WORK

Real-time Terrain Modeling using CPU–GPU Coupled Computation: Adrien Bernhardt, el. al.[6] have Motivated by the importance of having real-time feedback in sketch-based modeling tools, here a framework for terrain edition capable of generating and displaying complex and high resolution terrains presented. This system is efficient and fast enough to allow the user to see the terrain morphing at the same time the drawing editing occurs. There are two types of editing interactions: the user can draw strokes creating elevations and crevices; and previous strokes can be interactively moved to different regions of the terrain. One interesting aspect of this CPU– GPU coupled computation solution is the usage of the CPU to control the GPU solver iterations and stop at a certain resolution and then resume computing when the GPU is idle. Another interesting feature of this method is the balance between terrain generation in the CPU and in the GPU – one can control this balance by simply changing the quadtree refinement. It achieved realtime performance in both modeling and rendering using a hybrid CPU–GPU coupled solution.

Practical GPU and voxel-based indirect illumination for real time computer games: Andrei Simion, el. al.[7] have proposed the Indirect illumination techniques greatly improve the realism of computer generated 3D scenes. However, most of these techniques are used offline and rely on pre-baked solutions which do not support dynamic updates. There are some real time solutions but usually the degree of realism that they achieve is not convincing enough, Hence a real time solution based on a voxel representation of the scene generated from the mesh representation paired with an approximate voxel cone tracing technique which achieves good visual results and interactive frame rates is presented here. They've implemented a state of the art global illumination technique in a real game engine that runs at interactive frame rates and yields good visual results comparable with other state of the art approaches. Even if the algorithm runs at the same time as other game engine systems, it still have good performance and we can deliver a complete real-time experience to the user. Although the hardware requirements are fairly high to be able to run these algorithms, both hardware and software improvements are being developed to support these techniques and will undoubtedly become integral parts of modern game engines.

High-Performance Navigation and Rendering of Very-Large Scale Landscape and Seascape: Pi Xuexian, el. al.[8] has tried to use the idea for navigation of large-scale landscape, some difficulties must be overcome: Firstly, the huge geometry height map data and corresponding texture data may exceed the capability of current computer. Secondly, the view camera will move freely and arbitrarily. For simulation and rendering of unbounded ocean surface, the realistic ocean wave and real-time calculation of illumination of ocean surface may overwhelm the CPU and GPU. This paper presents a new framework based on some new data structures and new rendering algorithms for navigation and rendering of very large-scale landscape and seascape. Firstly, construction of a new terrain data model, which consists of Terrain Tile Pyramid and Terrain Summary Pyramid carried out. Secondly, in order to stitch the terrain tile boundaries seamlessly, the Index Template is introduced, and an algorithm of LOD based on "Remarkability" is described in detail. Finally, Perlin noise is used to simulate the ocean surface and a framework of navigation of very large-scale scene is completed. Experimental results show that this system can satisfy the requirements of real-time navigation of large-scale landscape and seascape.

ISSN: 2233-7857 IJFGCN

Ocean Wave Real-time Simulation Based on Adaptive Fusion: Shunli Wang, el. al.[9] has described the key of this paper is in using Perlin noise to generate height map at distant, FFT method at near sight and fuse the height map with adaptive fusion algorithms during the transition phase. Aim at the poor fidelity in large-scale ocean wave scene real-time simulation, a simulation method based on adaptive fusion is proposed. Perlin noise is used to generate height map at distance and FFT method is adopted at near sight. The adaptive fusion of height map is realized during transition phase. Meanwhile, simulation software is designed and the ocean wave real-time simulation with a 100x100km2 range is realized. The method is compared with the ocean wave that only uses Perlin noise method or FFT method. Simulation results show that the method can effectively improve the fidelity of ocean wave simulation in real time. This method makes full use of the advantages the high fidelity of Perlin noise at distant and FFT method at near sight. It effectively enhances the fidelity of ocean wave real-time simulation

Realistic Real-time Rendering of 3D Terrain Scenes Based on OpenGL: Dewen Seng, el. al.[10] has explored that the scientific visualization, computer animation and virtual reality are three hotspots of the modern computer graphics, and 3D realistic real-time rendering is the core technique of them. Generation of realistic images of virtual nature scenes is a challenging topic in computer graphics. As an important part of natural scenery, realistic rendering of 3D terrain scenes has gradually attracted much attention. It is applied widely in cartoon, games, films, virtual reality and engineering domain. OpenGL is the current international standard of 3D image. This paper discussed the implementing methods for the representation of 3D terrain realistic scenes based on OpenGL, such as the digital terrain model mapping, the blend of water and terrain data, the blend of thematic information and terrain data. The main contents of this paper include the research on illumination reality and the motion reality, which are the key issues of virtual environment generation. The experiences of using OpenGL to generate highly realistic images with realistic real-time rendering techniques were summarized. A visualization system for 3D terrain scenes was developed to demonstrate the effectiveness and efficiency of OpenGL as an interfacing tool for realistic real-time image rendering of 3D terrain scenes, and the results fit the anticipant goals well.

IV.CONCLUDING REMARKS

Performance of the 3D Open World Game is a big issue which can be solved with the help of High-Performance Computing. Various researchers have put their best efforts to solve this issue. The literature review has yielded the insights into some of the useful approaches to solve this problem, also minimization of the rendering time required per frame/scene. One particular High-Performance Computing technique known as CPU-GPU interoperability, is highly capable in case of improving efficiency. This makes it the perfect fit for implementing an 3D Open World Game. Other methods also exist but are not capable of performing the best in interoperability

REFERENCES

[1] Ajinkya Gavane Benjamin Watson, “The Effects of Adaptive Synchronization on Performance and Experience in Gameplay” North Carolina State University, USA Rachit Shrivastava, Microsoft, , North
Carolina State University, USA May 2019.

[2] Ya-lin Zhang , “A few of the most popular models for heterogeneous parallel programming”

ISSN: 2233-7857 IJFGCN

Gang Xie Institute of Computer Applications China Academy of Engineering Physics, Institute of Computer Applications China Academy of Engineering Physics 2017.

[3] Wai-Kiu Lee and Rocky K. C. Chang , "Evaluation of Lag-Related Configurations in First-

Person Shooter Games" Department of Computing, The Hong Kong Polytechnic University

Hung Hom,Kowloon, Hong Kong fcswklee, csrchangg@comp.polyu.edu.hk 2016.

[4] Peter Kvasnica, Igor Kvasnica, "Application of CUDA Computing Principles in Automatic Flight Control Simulation" -2013.

[5] Marcelo P. M.Zamith, Esteban W. G. Clua, Aura Conci and Anselmo Montenegro Marcelo P. M.

Zamith, Esteban W. G. Clua, Aura Conci and Anselmo Montenegro , "Parallel processing between GPU and CPU: Concepts in a game architecture" -2007

[6] Adrien Bernhardt, Andre Maximo, Luiz Velho, Houssam Hnaidi, Marie-Paule Cani,"Real-Time Terrain Modeling Using CPU-GPU Coupled Computation" -2011 24th SIBGRAPIConferenceon Graphics, Patterns and Images.

[7] Andrei Simion, Victor Asavei, Sorin Andrei Pistirica, Ovidiu Poncea "Practical GPU and Voxel-Based Indirect Illumination for Real Time Computer Games", 2015 20th International Conference on Control Systems, Computer Science.

[8] Pi Xuexian, Yang Xudong, Li Sikun, Song Junqiang "High-performance navigation and rendering of

very-large scale landscape and seascape" , 2005 Ninth International Conference on Computer Aided Design, Computer Graphics (CAD-CG'05).

[9] Fengju Kang, Dinghua Wang "Ocean wave real-time simulation based on adaptive fusion ; Shunli Wang"

, Proceedings of the 32nd Chinese Control Conference, 2013, Conference Paper.

[10] " Dewen Seng, Hongxia Wang "Realistic Real-Time Rendering of 3D Terrain Scenes Based on OpenGL", 2009 First International Conference on Information Science and Engineering, Conference Paper.

ISSN: 2233-7857 IJFGCN