

GPU-Assisted Techniques for Image-based Rendering and
Physically Based Computer Animation

by

Jian Zhu

Doctor of Philosophy in Software Engineering



2012



Faculty of Science and Technology

University of Macau



GPU-Assisted Techniques for Image-based Rendering and
Physically Based Computer Animation

by

Jian Zhu

SUPERVISOR: Prof. EnHua Wu

CO-SUPERVISORS: Prof. PhengAnn Heng, Prof. HanQiu Sun

Department of Computer and Information Science

Doctor of Philosophy in Software Engineering

2012

Faculty of Science and Technology

University of Macau



Author's right 2012 by
ZHU Jian





Acknowledgements

Firstly, I would like to express my sincere appreciation to my supervisor, Prof. WU Enhua, not only for his professional guidance and valuable advices throughout the whole seven years during my master and PhD study, but also for his kind support and constant encouragement. His conscientious attitude at work impressed and influenced me a lot. I am grateful to him for leading me to such an exciting research area in computer science: Computer Graphics. I am also grateful to my two co-supervisors: Prof. PhengAnn Heng and Prof. HanQiu Sun from CUHK.

In particular I would like to express my great thanks to Prof. LI Yiping, Ms. LIANG Shouzhuang, and my mathematics teacher Mr. ZENG Chuping in No.2 Middle School of ShaoYang, for their kind concern, help, trust, and encouragement all the time. Special thanks are given to Dr. PUN Chiman, who taught me a lot about image processing. I worked as his teaching assistant for more than two years, and we collaborated very well. I am also grateful to other teachers who taught me the master courses. They are Prof. GONG Zhiguo, Dr. XU Qiwen, Dr. SI Yain Whar, Prof. LI Xiaoshan, Prof. THAM Yiu Kwok and Dr. SIMON Fong.

Thanks should go to Donald Lam WengHong, the lab technician who helped me a lot in my daily work. I have also benefited from time spent with the following colleagues and students: CHANG Yuanzhang, GUO Qinghui, CAI Zhongmou, CHEN Yadang, HAO Chuanyan, YANG Lijie, ZHANG Shixue, SHENG Bin, YAO

Wenli, and LI Dong. I also would like to thank my roommates. They are ZHU Haitao, ZHENG Haibin, ZHENG Chenpei, CHEN Shipo, and ZHANG Bin.

My thanks also extend to the professors and students from the State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing. They are Prof. WANG Wencheng, LIU Xuehui, CHEN Yanyun, LIU Youquan, LI Sheng, Dr. BAO Kai, HUANG Peijie, JIAO Shaohui, CHI Xiaoyu, CHEN Xin, YANG Liming, MENG Weiliang, HUANG Mengcheng, LIU Fang, PhD candidates YANG Meng, WU Xiaolong, LIU Feitong, WANG Shandong, XIE Guofu, LI Xiaosheng, SUN Gang, MA Ziyang, Master students LIU Ming, LIU Yang, ZHANG Man, CHEN Shunbin, LIU Le, LU Jian, et al.

I am grateful to my parents for their selfless and endless love to me. They are not highly educated persons, but they educated me a lot. I am so grateful for my upbringing. Most important of all, the deepest thanks go to my wife and my little one year old daughter. I owe them too much. Without my wife's big consideration and encouragement, I am afraid I cannot finish the study. My gratitude is far beyond the words I can express. They are most precious to me in the world, and they are the motivating force that makes me go forward.

Finally, I would like to express my appreciation to University of Macau for the support of postgraduate studentship to me.

Abstract

For a long time, real-time rendering of highly realistic scene has been a goal pursued by computer graphics researchers. The parallelism of programmable graphics hardware provides us a way to accelerate the simulation and rendering speed. In this thesis, we present two GPU-assisted techniques for image-based rendering and physically based computer animation, respectively.

In Chapter 3 of this thesis, we present a real-time depth image-based rendering algorithm by GPU acceleration. By the algorithm, we uniformly sample an object in all directions in a polar coordinates system, to construct a *Spherical Depth Image*. With two deduced Warping Equations, we then pre-warp the image onto a view-dependent plane to get an intermediate image, which is further rendered onto the target image plane using standard texture mapping. By exploiting the inherent parallelism of modern programmable GPU, we transport the pre-warping process into vertex shader. Furthermore, the hardware pipeline's rasterization function is utilized to conduct the image re-sampling efficiently to generate hole-free rendering results.

Significant achievements have been made in the past two decades to model the dynamic behavior of fire, but almost none of them have addressed the decomposition process of the solid objects under combustion. In Chapter 5 of this thesis, we present a realistic, fast, and controllable model to simulate burning solids on GPU. A hybrid structure of grids is employed to simulate the whole process efficiently. The fuels inside the solid are consumed in a physically based way based on combustion theory, and fire is physically simulated by solving the fluid dynamics equations. They are well coupled by treating the solids as part of the fluid with high viscosity. Fire

propagation along the burning surface is well simulated with level set method. With proposed GPU-based ray-marching and multi-textures methods to visualize the turbulent fire and burning solid respectively, convincing results are produced. To achieve interactive simulation speed, a few acceleration techniques are employed, including a moving grid to dynamically track the fire propagation, a refined Marching Cubes method to reconstruct the burning surface, and a hardware-implemented fluid solver with CUDA.

Keywords: GPU, Image-based rendering, Depth image, Hardware acceleration, Pre-warping, Pixel Shader, Vertex Shader, CUDA, Physically based animation, Fluid simulation, Navier-Stokes Equations, Fire propagation, Object decomposition, Level set, Moving grid, Marching cubes

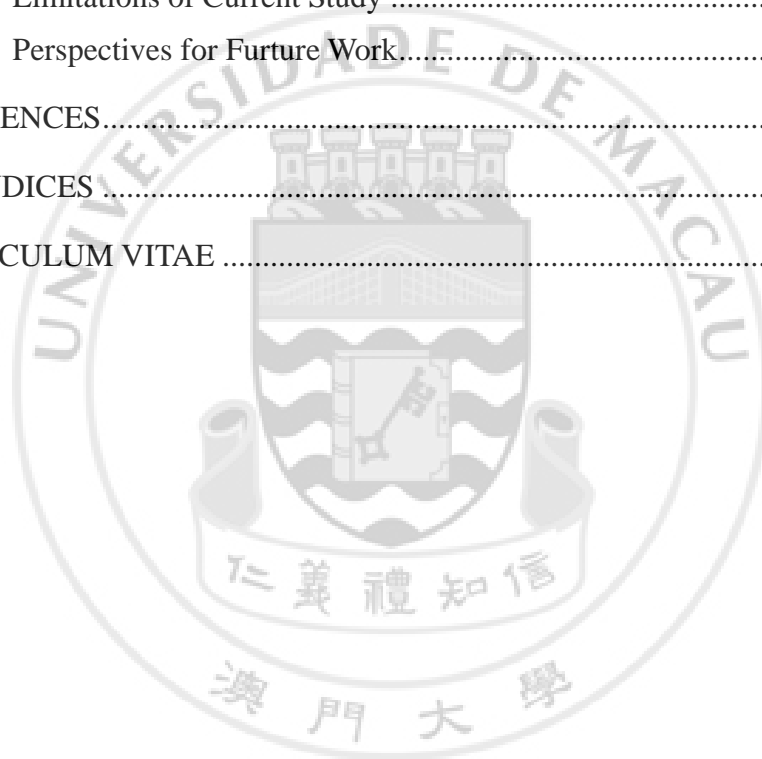


Table of Contents

Acknowledgements	i
Abstract	iii
Table of Contents	v
List of Tables and Figures	viii
Chapter 1: INTRODUCTION	1
1.1. General Background	1
1.2. Specific Background	1
1.2.1. Graphics Hardware Acceleration	1
1.2.2. Image-based Rendering	2
1.2.3. Physically Based Fluid Animation	3
1.3. Research Goals and Objectives	4
1.4. Research Methodology and Design	7
1.5. Potential Contributions	8
1.6. Thesis Organization	9
1.7. Statement of Originality	10
Chapter 2: GRAPHICS PROCESSING UNIT	13
2.1. Introduction	13
2.2. Graphics Hardware Pipeline	15
2.3. Shading Language	17
2.4. GPGPU with CUDA	20
2.4.1. Computing Architecture of CUDA	21
2.4.2. Programming Model of CUDA	23
Chapter 3: GPU-ACCELERATED RENDERING OF A SINGLE SPHERICAL DEPTH IMAGE IN REAL-TIME	27
3.1. Image-based Rendering	28
3.1.1. Classification of IBR Techniques	29
3.2. Related Work	31
3.2.1. 3D Image Warping	31
3.2.2. Relief Texture Mapping	33
3.2.3. Double Projective Cylindrical Texture Mapping	35
3.3. Spherical Depth Image-based Rendering on GPU	38
3.3.1. Spherical Depth Image-based Modeling	39

3.3.2. Rendering Algorithm.....	43
3.4. Experiment Results and Comparison	55
Chapter 4: PHYSICALLY BASED FLUID SIMULATION	61
4.1. Introduction	61
4.1.1. Eulerian Method.....	63
4.1.2. Lagrangian Method	65
4.2. Different Kinds of Fluids	67
4.2.1. Smoke and Fire.....	67
4.2.2. Water	69
4.2.3. Bubbles.....	70
4.2.4. Granular Materials.....	71
4.2.5. Deformable Materials.....	72
4.3. Phenomena Related to Fluids.....	74
4.3.1. Mixing Fluids.....	74
4.3.2. Fluid-Solid Coupling.....	75
4.3.3. Controlling Fluid	76
4.4. Numerical Schemes for Fluid Simulation	78
4.4.1. Fluid Solver.....	78
4.4.2. Surface Tracking	81
4.4.3. Grid Generation.....	83
4.4.4. Vortex-based Methods.....	84
4.4.5. Procedural Turbulence Methods.....	86
4.4.6. Hybrid Methods.....	86
Chapter 5: GPU-ASSISTED SIMULATION OF BURNING SOLID	88
5.1. Related Work.....	90
5.1.1. Simulation of Fire.....	90
5.1.2. Rendering of Fire	92
5.1.3. Erosion and Decomposition	94
5.1.4. GPU Acceleration in Fluid Simulation.....	95
5.2. Proposed Model.....	97
5.2.1. Modeling Structure.....	99
5.2.2. Object Decomposition.....	100
5.2.3. Fire Propagation	102
5.2.4. Flame Simulation	103
5.3. Visualization.....	105

5.3.1.	GPU-based Volume Rendering of Fire by Ray Marching.....	105
5.3.2.	GPU-based Visualization of Burning Surface with Multi-Textures Using GLSL.....	107
5.4.	Acceleration Techniques	109
5.4.1.	Moving Grid.....	109
5.4.2.	Refined Marching Cubes.....	110
5.4.3.	CUDA Based Fluid Solver	111
5.5.	Results and Comparisons	115
Chapter 6:	CONCLUSIONS.....	125
6.1.	Conclusions.....	125
6.2.	Limitations of Current Study	127
6.3.	Perspectives for Furture Work.....	128
REFERENCES.....		129
APPENDICES		148
CURRICULUM VITAE		152



List of Tables and Figures

Table 3-1: Comparison of different rendering methods on the same platform.....	57
Table 4-1: Several high order operators for advection.....	79
Table 4-2: Several pressure solvers.....	81
Table 5-1: The workflow of our GPU-based ray marching method	106
Table 5-2: Refined Marching Cubes	111
Table 5-3: Speed of our model without/with different acceleration methods	118
Figure 2-1: Computational horsepower of modern GPU and CPU [CUDA2012]	15
Figure 2-2: 3D rendering pipeline.....	16
Figure 2-3: A programmable rendering pipeline.....	17
Figure 2-4: Workflow in Cg.....	19
Figure 2-5: Computing architecture of CUDA [CUDA2012]	22
Figure 2-6: Grid of blocks in CUDA [CUDA2012]	24
Figure 2-7: A simple program written with CUDA	26
Figure 3-1: IBMR rendering pipeline	29
Figure 3-2: Illustration of 3D warping.....	33
Figure 3-3: Relief texture mapping pipeline.....	34
Figure 3-4: Double projective cylindrical camera model	36
Figure 3-5: DPCI: (a) color information (b) depth value.....	37
Figure 3-6: DPCI rendering with a top and cap planar image	38
Figure 3-7: 3D polar coordinate system.....	40
Figure 3-8: Relief textures: (Top) color and depth, (Bottom) normal	41
Figure 3-9: Re-projection of the relief texture in the front side.....	42
Figure 3-10: Spherical Depth Image: (a) color (b) depth (c) normal.....	43
Figure 3-11: Two-pass rendering of Spherical Depth Image	44
Figure 3-12: 2D view of our pipeline	45

Figure 3-13: 3D view of the pre-warping stage.....	47
Figure 3-14: Running results	58
Figure 3-15: (a) Color texture for Per-pixel lighting; (b) (c) (d) Lighting effects	58
Figure 3-16: Environment mapping effects (I)	59
Figure 3-17: Environment maps of a Quadrangle	59
Figure 3-18: Environment mapping effects (II).....	60
Figure 3-19: Environment maps of the Nvidia Lobby.....	60
Figure 4-1: Computational grids in Eulerian methods.....	64
Figure 4-2: Finite difference operators used in Eulerian methods.....	64
Figure 4-3: Physically based simulation of smoke [Schechter08].....	68
Figure 4-4: Physically based simulation of fire [Hong07].	69
Figure 4-5: Physically based simulation of water [Enright02].	70
Figure 4-6: Physically based simulation of bubbles [KimD10].	71
Figure 4-7: Physically based simulation of granular materials [Narain10].	72
Figure 4-8: Physically based simulation of deformable materials [Wojtan08].....	73
Figure 4-9: Physically based simulation of mixing fluids [Bao10].	75
Figure 4-10: Physically based simulation of fluid-solid coupling [Lenaerts08].	76
Figure 4-11: Physically based simulation of fluid control [Thuerey06].	77
Figure 5-1: Burning solids in real world.....	89
Figure 5-2: Fire simulation with physically based methods. (a) From [Nguyen02], (b) From [Hong07], (c) From [Horvath09].	92
Figure 5-3: Realistic fire rendering results in [Pegoraro06].	94
Figure 5-4: Simulation result of burning solid in [Melek03].	95
Figure 5-5: Overview of our proposed model.	99
Figure 5-6: Our proposed hybrid grids structure: (Red) Global Grid, (Yellow) Local Grid, and (Green) Moving Grid	100
Figure 5-7: (a) GPU-based ray marching method for fire rendering. Each cast ray stops marching once colliding with the solid fuel boundary. (b) Volume rendering of fire by drawing a cluster of slices from back to front. Some	

of the slices intersect with the solid fuel, making the solid surface look discontinuous.....	107
Figure 5-8: Each node obtains a texture coordinate from its nearest point on the surface, and each generated vertex obtains its texture coordinate by interpolation.....	108
Figure 5-9: A simplified model leads to complete combustion. The three grids (red, yellow, and green ones) are the global grid, local grid and moving grid, respectively. Compared with Example 1 (the top row), burning surface in Example 2 (the bottom row) is bigger with a different configuration of K	119
Figure 5-10: Two examples of incomplete combustion. d_s has a higher value in Example 4 (bottom row) than in Example 3 (top row), so Example 4 demonstrates a higher degree of combustibility.	120
Figure 5-11: A wooden dinosaur burns from different locations in Example 5 (top row) and Example 6 (bottom row). The global grid, local grid, and moving grid are marked in red, yellow, and green colors, respectively.	121
Figure 5-12: Burning bunny.....	122
Figure 5-13: As the wood decomposes gradually, the gold characters (i.e., Chinagraph) inside the wood show up.	123
Figure 5-14: As the wood decomposes gradually, the gold characters (i.e., CASA2011) inside the wood show up.	124

