GPU Accelerated Particle Methods for Simulating and Rendering Fire and Water Effects

A thesis presented in partial fulfillment of the requirements
for the degree of

Doctor of Philosophy
in
Computer Science

at Massey University, Albany
New Zealand.

Timothy Lyes

2015

Abstract

The simulation of complex natural phenomena such as fire and water is a complicated problem and with the surge in popularity of video games and other interactive media, it has become an area of interest in computer graphics to be able to simulate these phenomena in real-time. The challenge exists not only to simulate as accurately as possible for the best degree of visual realism, but also to use a method which allows for this real-time interaction.

In this thesis, the use of particle systems as a method for simulating fire and water effects is explored, as well as the rendering methods used to visualize them. Particle systems are well suited to this type of problem as they can be parallelized and provide many methods of behavioural customization in order to produce a wide range of different effects. Realistic looking results can be achieved when a sufficient number of particles are able to be simulated within an adequate time frame.

It can be shown that particle system methods such as Smoothed Particle Hydrodynamics and Velocity-Vortex methods are able to simulate these phenomena well. These methods are implemented using NVIDIA CUDA to parallelize the governing algorithms on the graphics processor, and with the use of spatial grid division techniques to reduce the computational complexity, they are able to run at real-time interactive rates.

Additionally, when utilizing point-based approaches for rendering fire, and a surface generation approach using the Marching Cubes algorithm for rendering water, it can be shown that these particle systems are able to be rendered with realistic-looking visualizations while maintaining interactivity. Combining both the computational aspects of the particle system and the rendering aspects directly on the graphics device produces good quality rendered fire and water effects at speeds fast enough to be used with interactive media applications.

Acknowledgements

I would like to thank everyone who has supported me. In particular I would like to thank my parents, Joanna and Rex, and my two sisters, Amy and Susannah, who have continued to love and support me during this endeavour. I would also like to thank my supervisors, Dr. Arno Leist, Dr. Daniel Playne and Prof. Ken Hawick for their guidance and expertise, as well as all my colleagues at Massey University for their camaraderie and encouragement.

CONTENTS

List of Figures **6**

List of Tables **9**

1 Introduction **11**

1.1 Aim and Research Methods of the Thesis 13
1.2 Key Contributions of the Thesis 14
1.3 Structure of the Thesis 14

2 Introduction to Particle Systems **17**

2.1 Particle Systems Overview 17
2.2 Fluid Flow and Hydrodynamics 20
2.3 Other Related Particle Models 23
2.3.1 Agent-Based Models 23
2.3.2 Spring-Mass Models 24
2.3.3 Other Notable Work 25
2.4 Rendering Effects 26
2.5 Numerical Methods and Integration 27
2.5.1 Euler Integration Method 27
2.5.2 Euler-Cromer Method 28
2.5.3 Runge-Kutta (2nd Order) 29
2.5.4 Runge-Kutta (4th Order) 29
2.5.5 Leapfrog Method 31
2.6 Computational Complexity 32
2.7 Introduction to Parallelization 32

3 Introduction to Parallelization **35**

3.1 CUDA Programming Model 36
3.2 Kepler Architecture 38

3.3 CUDA Memory 40
3.4 Parallelization of Particle Methods 42
3.5 CUDA Libraries 44
3.5.1 Thrust 44
3.5.2 CURAND Random Number Generation 45
3.6 Graphics Cards 46
3.7 Summary 46

4 Particle System Models 47
4.1 Non-interacting Particle Model 48
4.2 Spring-Mass Model 49
4.3 Smoothed Particle Hydrodynamics 52
4.3.1 Navier-Stokes Fluid Flow 53
4.3.2 SPH Equations 54
4.4 Velocity-Vortex Model 56
4.4.1 Rigid-Body Interaction 61
4.5 Summary 62

5 Spatial Grid and Integration Methods 63
5.1 Spatial Grid Algorithm 63
5.1.1 Spatial Grid-based Collision Detection 64
5.1.2 Sorting Considerations 67
5.1.3 Spatial Grid Implementation 68
5.2 Numerical Integration 70
5.2.1 Integration Implementation 70
5.2.2 Integration Method Comparison 74
5.3 Summary 82

6 Implementation 85
6.1 Fire Introduction 86
6.2 Non-interacting Fire Implementation 86
6.3 Plasma Implementation 90
6.3.1 Plasma System Overview 91
6.3.2 Plasma-specific Spatial Grid Analysis 93
6.3.3 Interaction Processing 98
6.4 Velocity-Vortex Fire Implementation 98
6.4.1 Fire System Initialization 101
6.4.2 Dynamic Spatial Grid 101
6.4.3 Vorticity Dynamics 103
6.4.4 Buoyancy Handling 107
6.4.5 Advection Step and Body Collision 107
6.4.6 Fire Performance Comparison 108

6.5 Water Implementation 109
6.5.1 Spring-Mass System Implementation 110
6.5.2 Smoothed Particle Hydrodynamics Implementation 113
6.5.3 SPH vs Spring-Mass Comparison 117
6.6 Summary 118

7 Rendering **121**
7.1 Rendering Introduction 122
7.2 Point-Based Rendering Method 123
7.3 Fire Visual Comparison 125
7.4 Volumetric Rendering 127
7.5 Surface Generation 130
7.5.1 Isosurface Introduction 132
7.5.2 Marching Cubes 134
7.5.3 Initialization 137
7.5.4 Implementation Overview 138
7.5.5 Cell Classification Approaches 138
7.5.6 Generating the Surface 142
7.6 Surface Rendering 144
7.6.1 Surface Rendering 144
7.6.2 Cubemapping and Fresnel Reflection 146
7.7 Results Discussion 152
7.8 Realism Comparisons 156
7.9 Summary 159

8 Discussions and Conclusions **161**
8.1 Discussion 161
8.1.1 Numerical Integration and Spatial Partitioning 162
8.1.2 Choices in Fire and Water Simulation Models 163
8.1.3 Surface Generation and Rendering 165
8.2 Conclusions 166
8.3 Future Work 168

LIST OF FIGURES

2.1 Example screenshot of a particle system 18
2.2 Screenshot of FFT waves approach 22
2.3 Example screenshot of a cloth simulation for a flag 25
2.4 Leapfrog Method diagram 31

3.1 Visual representation of the Kepler SMX layout. 39

4.1 Diagram of non-interacting particle motion 48
4.2 Diagram of spring-mass particle motion 50
4.3 Diagram of SPH particle dynamics 53
4.4 Diagram of velocity-vortex particle dynamics 60

5.1 Diagram for particle-to-particle interactions in a spatial grid 64
5.2 Diagram for first and last arrays calculation 69
5.3 Energy conservation comparison for an active system 77
5.4 Energy conservation comparison for an calm system 78
5.5 Graph for Euler method instability demonstration 78
5.6 Conservation of energy over multiple runs and time scales 79
5.7 Comparison of execution times for a spring-mass model 80
5.8 Comparison of execution times for a SPH model 81
5.9 Visual comparison using different integration methods 82

6.1 Visual comparison and diagram for a small electrostatic field size 94
6.2 Visual comparison and diagram for a large electrostatic field size 95
6.3 Visual comparison and diagram for an appropriate electrostatic field size 96
6.4 Graph for execution times for increasing electrostatic influence sizes 97
6.5 Comparison of execution times for different fire system simulations 109
6.6 Example screenshot of fire system rendering 110
6.7 Comparison of execution times for different water system simulations 118

6.8 Visual comparison of different water system simulations 119
6.9 Time lapse of water simulation 120

7.1 Example texture used for alpha values in point-based rendering 125
7.2 Visual comparison of different fire rendering methods 126
7.3 Volumetric rendering of a water system 130
7.4 Sample surface rendering screenshots 131
7.5 Diagram of a particle field representation 134
7.6 Marching Cubes states 136
7.7 Diagram of particle fields influencing the Marching Cubes state 137
7.8 Screenshot of Blinn-Phong rendering of a water simulation 146
7.9 Diagram of the Fresnel effect 148
7.10 Screenshots of water systems rendered using the Fresnel effect 151
7.11 Screenshots of interactive scenarios rendered using the Fresnel effect 151
7.12 Screenshots demonstrating differences in surface smoothness due to the configuration of the method 154
7.13 Differences in surface smoothness at a smaller Marching Cubes resolution . . . 155
7.14 Differences in surface smoothness using a large field size 155
7.15 Fire comparison between simulated image and real photo 156
7.16 Water surface comparison between simulated image and real photo 157
7.17 Water droplet comparison between simulated image and real photo 158
7.18 Splash comparison between simulated image and real photo 159

LIST OF TABLES

3.1 Kepler GK110 architecture statistics. 38
3.2 NVIDIA GeForce GTX 780 Specifications 46

6.1 Simulation parameters for the spring-mass model 111
6.2 Simulation parameters for the SPH model 113

7.1 Test results for water simulation using 262,000 Marching Cubes voxels 152
7.2 Test results for water simulation using 2.1 million Marching Cubes voxels . . . 153