

Copyright is owned by the Author of the thesis. Permission is given for a copy to be downloaded by an individual for the purpose of research and private study only. The thesis may not be reproduced elsewhere without the permission of the Author.

SIMULATION AND MODELLING OF GRAVITATIONAL
MICROLENSING EVENTS USING GRAPHICAL
PROCESSING UNITS

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

DOCTOR OF PHILOSOPHY
IN COMPUTER SCIENCE

at Massey University,
Albany (Auckland), New Zealand.

Cho Hong Ling

2013

Abstract

This thesis presents the results of a study into the use of graphical processing units (GPUs) in the simulation and modelling of gravitational microlensing. Two simulation approaches were investigated: magnification maps and the use of a dynamic engine for directly simulating gravitational microlensing light curves. It was found that the GPUs are able to speed up the generation of magnification maps dramatically. Very high performance in light curve extraction from magnification maps using GPUs is also achieved. Furthermore, the use of texture memory speeds up the extraction of light curves in a further 75% improvement in performance. They provide a speed up of over a $100\times$ faster than CPUs in light curve simulations with finite source effects. The dynamic engine approach use a hybrid computation method with both CPUs and GPUs to simulate light curves for complex microlensing events. It allows us to model microlensing events with orbital motion effects, which are usually done on a cluster computer, on just a desktop computer with GPUs. Modelling strategies and optimization techniques are developed and applied to model different types of microlensing events.

GPU architectures show great promise for tackling the computationally expensive task of numerical modelling of microlensing events. With the modelling strategies developed here, microlensing modelling can be performed on a desktop computer at only a fraction of the cost of a cluster computer. The approach in this thesis provides a very cost-effective solution for the microlensing modelling challenge.

Acknowledgements

I would like to thank my supervisor Ian Bond and my co-supervisor Winston Sweatman for their guidance over the years. This thesis could not have been completed without their continued support and encouragement. I enjoyed the weekly meeting and interesting discussions of my research and life with them.

I would also like to thank everyone at the Institute of Information & Mathematical Sciences for their inspirational seminars, discussions and support.

My family means the world to me and I could not have completed my degree without them. Special thanks to my wife Linh for her selfless support and sacrifices over the years.

Contents

Abstract	iii
Acknowledgements	v
List of Figures	xiv
List of Tables	xv
1 Introduction	1
2 Gravitational Microlensing and Current Challenges in Microlensing	5
2.1 Introduction	5
2.2 Simple microlensing	6
2.3 Multiple lenses	11
2.3.1 Inverse ray shooting	11
2.3.2 Critical and caustic curves	13
2.4 Higher-order effects	14
2.5 Computational challenges in microlensing modelling	17
2.5.1 Multi-dimensional parameter space	17
2.5.2 Finite-source effect	18
2.5.3 Orbital motion in modelling	18
3 GPUs as potential tools for microlensing modelling	21
3.1 Introduction	21
3.2 Massively parallel processor	22
3.3 NVIDIA GPU	23
3.3.1 GPU architecture	23
3.3.2 Device Memory Space	24
3.3.3 CUDA threads organization	29
3.3.4 Coalescing memory access	31
3.3.5 Shared memory bank conflicts	37
3.4 Floating point accuracy	39
3.5 Performance tuning	41
3.5.1 Control flow	41
3.5.2 Thread execution	43
3.5.3 Memory latency hiding	45
3.5.4 Arithmetic Instructions	46
3.6 Conclusion	46

4	Microlensing Modelling using Magnification Maps	49
4.1	Introduction	49
4.1.1	Magnification map	49
4.1.2	From cluster computer to GPU	52
4.2	Magnification map generation	53
4.2.1	Magnification map generation using CPU	54
4.2.2	Magnification map generation using GPU	59
4.2.3	Parallel magnification map generation by both CPU and GPU	67
4.2.4	Performance comparison	69
4.3	Track extraction	70
4.3.1	Track extraction using CPU	71
4.3.2	Track extraction using GPU	73
4.3.3	Performance comparison and discussion	82
4.4	Conclusion	89
5	Dynamic Light Curve Engine for Complex Microlensing Modelling	91
5.1	Introduction	91
5.2	Dynamic light curve engine	92
5.2.1	Light curve partitioning	94
5.2.2	Efficient image solving algorithm	97
5.2.3	Efficient image area searching method	104
5.2.4	GPU inverse ray shooting	108
5.2.5	GPU ray shooting kernel	109
5.3	Dynamic light curve engine models	110
5.3.1	Light Curve Engine 1 - Sequential CPU and GPU computation	111
5.3.2	Light Curve Engine 2 - Overlapping CPU and GPU computation	115
5.3.3	Light Curve Engine 3 - Multi-threading image area search + GPU ray shooting kernel 1	116
5.3.4	Light Curve Engine 4 - Multi-threading image area search + GPU ray shooting kernel 2	120
5.3.5	Light Curve Engine 5 - Multi-threading plus overlapping computation between CPU and GPU	122
5.4	Conclusion	123
6	Development of Modelling Strategies and Microlensing Event Case Studies	125
6.1	Introduction	125
6.2	Modelling Strategies	126
6.2.1	Microlensing modelling using magnification map technique	126
6.2.2	Systematic grid search	133
6.2.3	Complex modelling using the dynamic light curve engine	135
6.3	Event showcase	137
6.3.1	Binary microlensing event - MOA2002-BLG-042	137
6.3.2	Binary microlensing event with parallex - MOA2008-BLG-199	142
6.3.3	Planetary microlensing event with a strong finite-source effect - MOA2007-BLG-400	146
6.3.4	Complex modelling event - MOA2004-BLG-33	149
6.4	Exclusion Region mapping	152
6.4.1	MOA2007-BLG-397	153
6.4.2	MOA2007-BLG-312	155
6.5	Conclusion	157

7 Conclusion and Discussion	159
7.1 GPU solution for microlensing modelling	159
7.1.1 Magnification map generation and light curve extraction	159
7.1.2 Dynamic light curve engine for complex microlensing events	160
7.1.3 Modelling strategies	160
7.2 Future work	161
7.2.1 Real-time automatic microlensing modelling system	161
7.2.2 GPU mini cluster	162
7.2.3 Multi planetary system modelling	162
7.2.4 Parallel computing in the near future for microlensing modelling	163

List of Figures

2.1	Microensing geometry. The lens L at a distance D_l from the observer O deflects light from the source S	7
2.2	Microensing light curves with single lens point source of $u_0 = 0.1, 0.2, 0.4, 0.8$. t is the normalised time unit.	9
2.3	Theoretical light curve of a binary-lens finite-source microlensing event.	10
2.4	Inverse ray shooting.	12
2.5	Schematic of the source star track and the caustic.	14
2.6	Critical and caustic curves for a binary lens with planetary mass fraction of 0.001. The top figure (a) shows the critical curve (red line) and the caustic curve (green line) of a lens system.	15
2.7	Light curve comparing the effects of a finite source and a point source. This figure demonstrates the effect on the theoretical light curve simulation by the finite source effect.	16
3.1	Fermi Architecture. This figure show multiple SMs and shared L2 Cache on the Fermi GPU.	24
3.2	Fermi Streaming Multiprocessor.	25
3.3	Thread organization. Each grid consists of multiple blocks and each block consists of multiple threads.	30
3.4	GPU Memory Access. Figure (a) shows coalescing global memory access which only takes one memory transaction.	33
3.5	GPU Memory Access. Three memory access patterns that are non-coalescing on devices of compute capability 1.0 and 1.1, but are coalescing memory access on devices of compute capability of 1.2 or higher.	34
3.6	GPU Memory Access. Some memory bandwidth is wasted due to not all memory locations being fetched within the 128 bytes memory segment.	35
3.7	GPU Shared Memory Access. Both (a) and (b) are a conflict-free shared memory access,	38
3.8	GPU Shared Memory Access. Two threads in (a) are accessing <i>different bytes</i> of a shared memory bank,	39
3.9	GPU Shared Memory Access. Both (a) and (b) are conflict-free access.	40
4.1	The top figure shows a magnification map generated with a binary lens system for a given ε and d	50
4.2	General procedure of magnification map generation.	54
4.3	Coordinate system used in the source plane and the lens plane. (x_i, y_i) is the image position and (x_s, y_s) is the source position. The image area on the lens plane is coloured in grey.	55
4.4	(a) is the source plane with boundaries $(sx_1, sy_1) = (-0.3, -0.3)$ and $(sx_2, sy_2) = (0.3, 0.3)$	56

4.5	Image area and active cells. The area coloured in grey are the image area. Grid cells that coloured in grey are the active cells that form the image area.	57
4.6	CPU performance on magnification map generation using single CPU core.	60
4.7	General procedure of magnification map generation on GPU.	61
4.8	Magnification map generation workflow.	64
4.9	GPU performance comparison on magnification map generation between unrolled kernels and kernels without unrolling.	66
4.10	Parallel magnification map generation by both CPU and GPU.	68
4.11	Performance comparison between CPU and GPU on magnification map generation.	69
4.12	Schematic of source star limb darkening profile and track extraction across a magnification map.	70
4.13	Bilinear interpolation. The four red dotted squares are pixels on the magnification map.	71
4.14	In Kernel 1 and 2, each thread writes the partial solution to GPU global memory.	74
4.15	This figure shows threads within a block taking advantage of locality to utilize caching capability of texture memory.	76
4.16	In Kernel 2, the partial solutions are being calculated by the first thread of each block and only the final solution is being transferred back to the CPU memory. The final solution is computed by the GPU before transfer back to the CPU memory.	78
4.17	Top figure (a) shows the performance on track extraction in four different kernels on G80 series GPUs and the bottom figure (b) shows performance on Fermi GPUs.	81
4.18	CPU vs. GPU on track extraction.	82
4.19	Top figure shows track extraction performance on a 2.66GHz Core i7 CPU turbo boosted to 2.93GHz. Bottom figure shows track extraction performance on a GTX480 GPU.	87
5.1	Dynamic light curve engine procedure	94
5.2	Critical curve and caustic curve. For binary lensing with planetary mass fraction 0.001 and projected separation of 1.2 Einstein radii between the components.	96
5.3	A source star and its trajectory as it crosses the central caustic. The lensing geometry is the same as in Figure 5.2.	97
5.4	The light curve corresponding to the source star track shown in Figure 5.3. The solid blue line is generated by the dynamic engine with a finite source. The dotted green line is generated by the point source approximation.	98
5.5	The image area (arc-like dark line on the lens plane) shown in the above figure is mapped from the source star (circle on the source plane).	99
5.6	Image locations for a binary lens. The black crosses are the image positions obtained by solving the inverted binary lens equation.	100
5.7	Source star lies at the edge of the caustic.	101
5.8	The black crosses are the point image positions obtained by solving the nine point-source locations show in Figure 5.7.	102
5.9	The above figure shows five disjointed image areas on the lens plane that map to a source star on the source plane.	103
5.10	Custom queue data structure for storing the active cells array. These are ready to be used by the GPU without rearrangement.	104

5.11	The image area searching algorithm at work at different stages. The point images are marked as black crosses and labeled active cells are in grey. The dark solid line is the real image area boundary.	107
5.12	Ray Shooting by GPU	110
5.13	Light Curve Engine 1	111
5.14	Light curve engine performance CPU vs. GPU Engine 1	112
5.15	CPU-only light curve engine performance. The CPU is an Intel Core i7 3.0GHz CPU running in turbo mode.	113
5.16	Light curve GPU Engine 1 performance. CPU is an Intel Core i7 3.0GHz CPU running in turbo mode and GPU is an NVIDIA GTX480.	114
5.17	Light Curve Engine 2	115
5.18	Light curve engine performance GPU Engine 1 vs. GPU Engine 2.	116
5.19	Sequential vs. Parallel computation using CPU and GPU	117
5.20	Light curve engine 3. Solving image area using multiple CPU threads and performing ray shooting using GPU ray shooting kernel 1.	118
5.21	Light curve engine performance GPU Engine 1 vs. GPU Engine 2 vs. GPU Engine 3. CPU is a Quad core Intel Core i7 3.0GHz CPU and GPU is an NVIDIA GTX480. The error bars are smaller than the plot symbols.	119
5.22	Light curve GPU Engine 2 performance. CPU is an Intel Core i7 3.0GHz CPU running in turbo mode and GPU is an NVIDIA GTX480. The error bars are smaller than the plot symbols.	119
5.23	Light curve engine 4	120
5.24	Light curve engine performance GPU Engine 1 versus GPU Engine 2 versus GPU Engine 3 versus GPU Engine 4. The error bars are smaller than the plot symbols.	121
5.25	Light Curve Engine 5	122
5.26	Light curve engine performance GPU Engine 3 vs. GPU Engine 4 vs. GPU Engine 5. The error bars are smaller than the plot symbols.	123
6.1	Work flow for the grid search using the magnification map technique.	127
6.2	Optimization strategy 1 work flow	130
6.3	Optimization strategy 2 work flow	131
6.4	Optimization strategy 3 work flow	132
6.5	Light curve from 4 telescopes of the microlensing event MOA2002-BLG-042, together with the best-fitting binary-lens model.	138
6.6	Diagram representing the geometry of the best-fitting model for MOA2002-BLG-042.	139
6.7	MOA2002-BLG-042 χ^2 map resulting from a grid search of binary models over the binary mass fraction, ε , and the projected separation, d . This was generated using optimisation 1 described in Section 6.2.	140
6.8	MOA2002-BLG-042 χ^2 map as in Figure 6.7. The stretch in χ^2 ranges from 0 to 5000.	140
6.9	MOA2002-BLG-042 χ^2 map as in Figure 6.7. The stretch in χ^2 ranges from 0 to 2500.	141
6.10	Searching time on each magnification map in MOA2002-BLG-042 modelling	142
6.11	Light curve, obtained by MOA, of event MOA2008-BLG-199, together with the best-fitting binary lensing plus parallax model.	143
6.12	Magnification map image and source star track for the best-fitting parallax model of event MOA2008-BLG-199.	144
6.13	MOA2008-BLG-199 optimization flowchat	145
6.14	MOA2008-BLG-199 χ^2 map resulting from a grid search over the two parallax parameters π_E and π_N	146

6.15	Light curve obtained by MOA and CTIO of the event MOA2007-BLG-400 together with the best-fitting finite-source single lens model.	147
6.16	MOA2007-BLG-400 χ^2 map resulting from a grid search over the binary mass fraction and projected separations.	148
6.17	Light curve of MOA2007-BLG-400 as in Figure 6.15. Here the best-fitting binary-lens model is shown.	149
6.18	Diagram representing the geometry of the best-fitting model for MOA2007-BLG-400.	150
6.19	Light curve from 6 telescopes of the microlensing event MOA2004-BLG-33, together with the best-fitting xarallap model.	151
6.20	Detailed view of xarallap fitting model of MOA2004-BLG-33 as in Figure 6.19.	151
6.21	Light curve obtained by MOA of the event MOA2007-BLG-397 together with the best-fitting point-source single lens model.	153
6.22	Planetary exclusion regions for MOA2007-BLG-397.	154
6.23	Light curve obtained by MOA of the event MOA2007-BLG-312 together with the best-fitting point-source single lens model.	155
6.24	Planetary exclusion regions for MOA2007-BLG-312.	156

List of Tables

3.1	Architecture specifications showing comparisons between the different generations of NVIDIA GPU boards.	26
3.2	GPU architecture. Information on threads and warps in different generations of GPU.	32
5.1	GPU rays shooting performance. CPU is an Intel Core i7 3.0GHz CPU running in turbo mode and GPU is an NVIDIA GTX480.	114