A Platform-Oblivious Approach for Heterogeneous Computing: A Case Study with Monte Carlo-based Simulation for Medical Applications

Shih-Hao Hung , Min-Yu Tsai , Bo-Yi Huang , Chia-Heng Tu

National Taiwan University

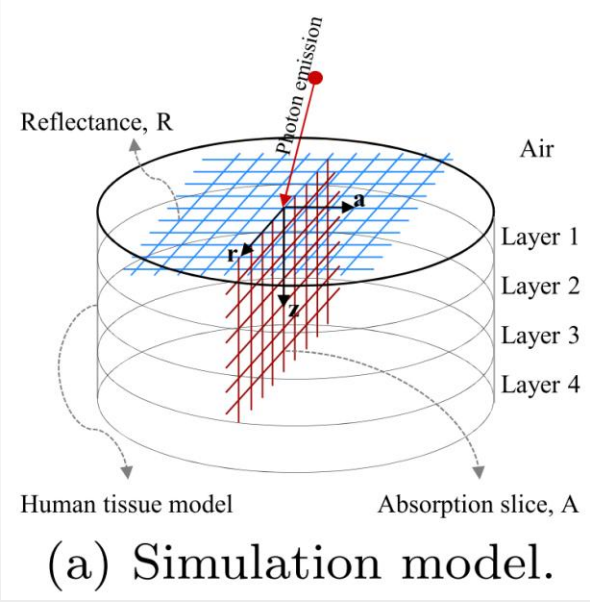
NTU PAS Lab
Performance, Applications, Security

ABSTRACT

- Light is important and helpful in many medical applications, such as cancer treatment.
- Monte Carlo-based simulations are considered to deliver accurate results, but require intensive computational resources.
- GPU and FPGA have been proposed to accelerate MC simulations with platform-specific programming schemes.
- Our platform-oblivious approach provides a unified, highly portable code and delivers competitive performance and power efficiency.

MCML OVERVIEW

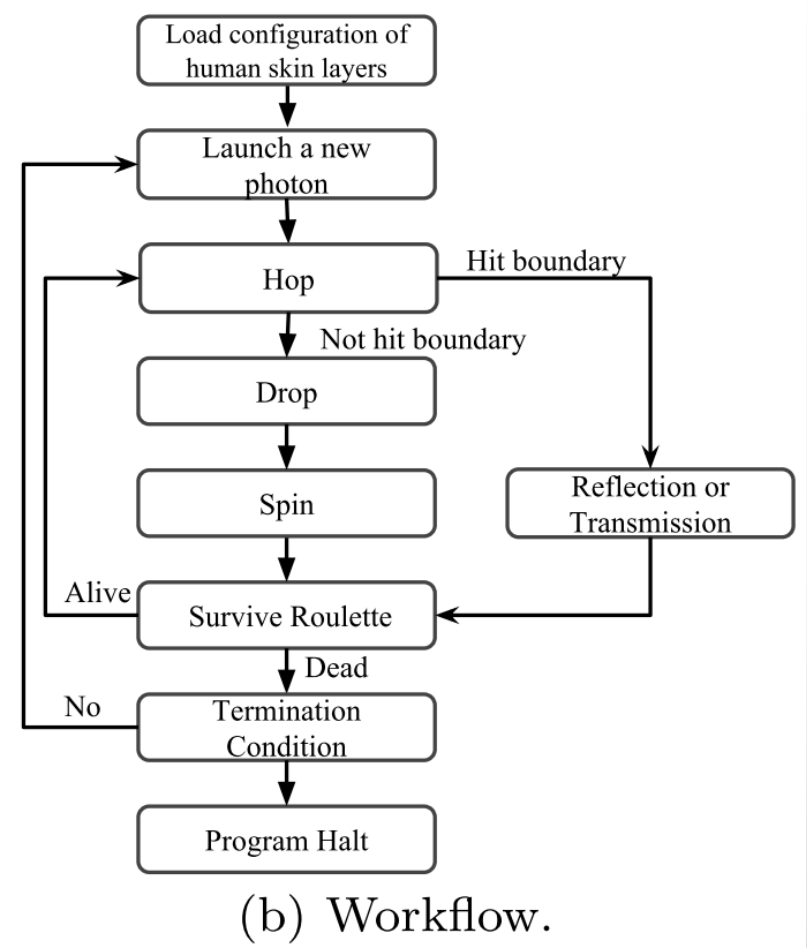
- Lihong Wang's version of Monte Carlo modeling of light transport in multi-layered tissues (MCML) was released in 1995.
- C-based MCML program simulates the photon propagation in a multi-layered tissue, assumes
  - plane-parallel layers
  - infinitely wide with homogeneous media
  - Media parameters with absorption, scattering, anisotropy, and refractive index.



(a) Simulation model.

Workflow of MCML

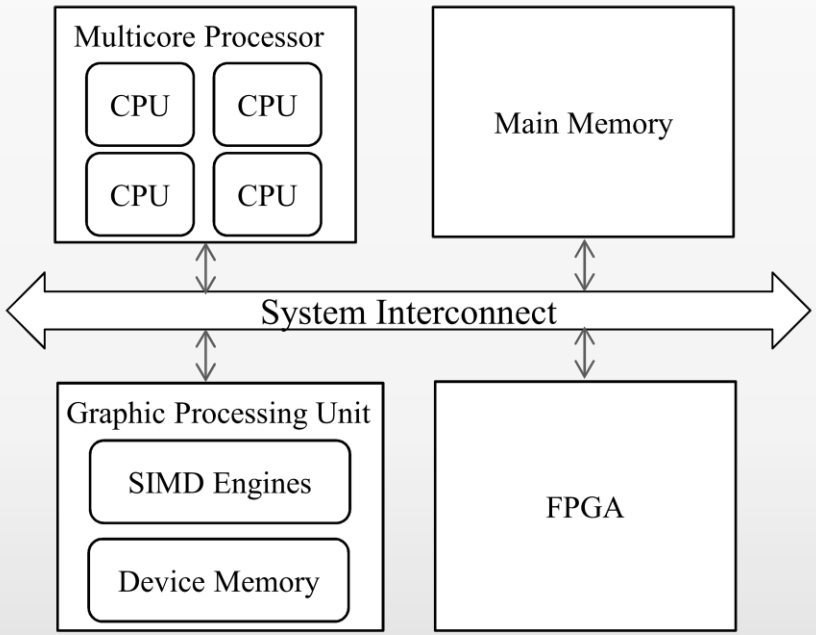
- Starts with loading the parameters of the given human skin model.
- Set up initial random sampled configuration, e.g., weight, position, direction, and step size.
- Undergoes the three key operations, hop, drop, and spin, iteratively until it is considered as dead.



(b) Workflow.

MCML ACCELERATION

- A general system organization for running OpenCL applications.
  - The multi-core processor acts : Host or Device
  - GPU and FPGA : Devices
  - Main memory stores the initial OpenCL program and data.
  - Device memory is used to keep the data transferred from/to the main memory at the device side.



ACCELERATING METHODS

- Batch Parallel Execution Model
- Buffering Photon Records
- Reducing Branch Divergence
- Caching Data with Constant Memory
- Sectioning the Thread Execution
- Fast Relaxed Math

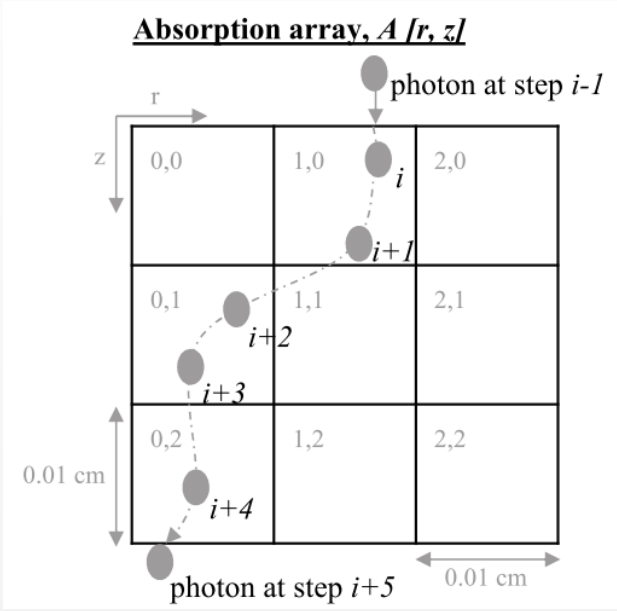
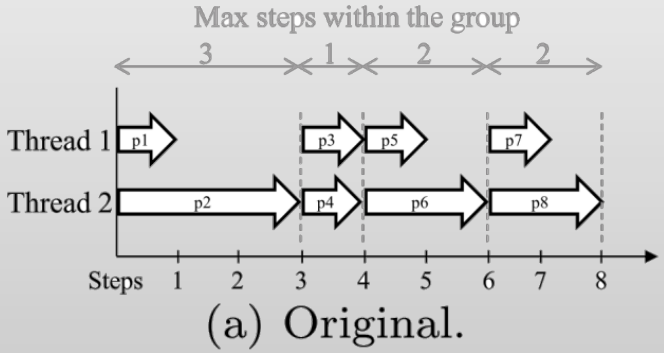
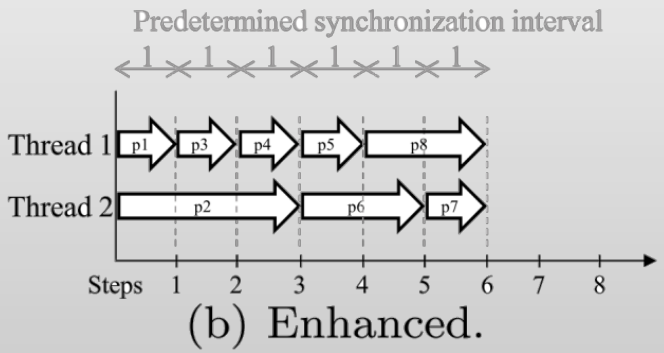


Figure 3. Mapping the photon's travel path to the absorption array.



(a) Original.



(b) Enhanced.

EXPERIMENTAL ENVIRONMENT

Table 1. Experiment environment.

Accelerator	Process (nm)	Memory bandwidth (GB/s)	Last level cache	Board power (W)	OpenCL SDK	Operating system
Intel Core i7-920	45	25.6	8 MB	130	Intel OpenCL 14.2	CentOS 6.5,
NVIDIA GTS 450	40	57.7	256 KB	106	CUDA Toolkit 3.2	kernel 2.6.32-431,
Altera Stratix V	28	25.6	None	25	Altera OpenCL 14.1	gcc/g++ 4.4.7

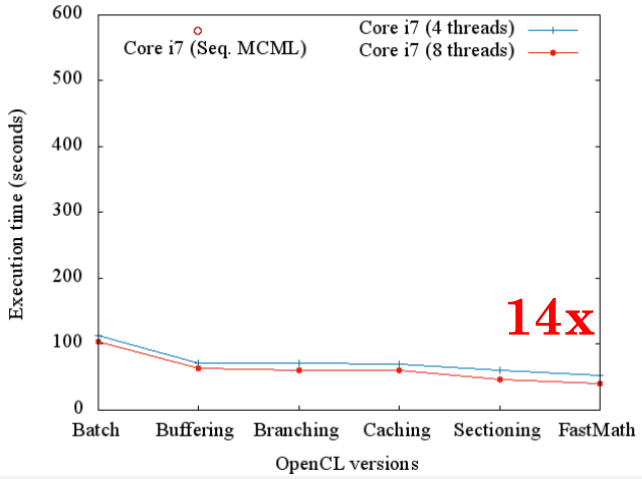
Table 2. Properties of the five-layer skin tissue at 633nm [*].

Layer	Absorption coefficient (cm^{-1})	Scattering coefficient (cm^{-1})	Anisotropy factor	Refractive index	Thickness
1st epidermis (top)	4.3	107	0.79	1.5	0.01
2nd dermis	2.7	187	0.82	1.4	0.02
3rd dermis plexus superficialis	3.3	192	0.82	1.4	0.02
4th dermis	2.7	187	0.82	1.4	0.09
5th dermis plexus profundus	3.4	194	0.82	1.4	0.06

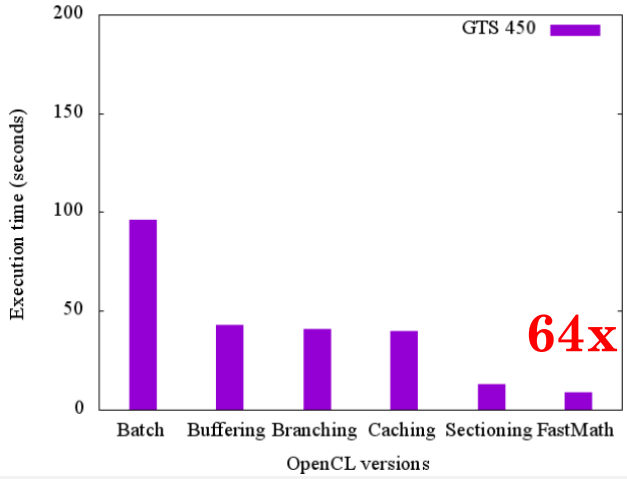
[*] W. C. Y. Lo, T. D. Han, J. Rose, and L. Lilge. GPU-accelerated Monte Carlo simulation for photodynamic therapy treatment planning. In European Conferences on Biomedical Optics, pages 737313–737324, 2009.

EXPERIMENTAL RESULTS
EXECUTION TIME COMPARISON

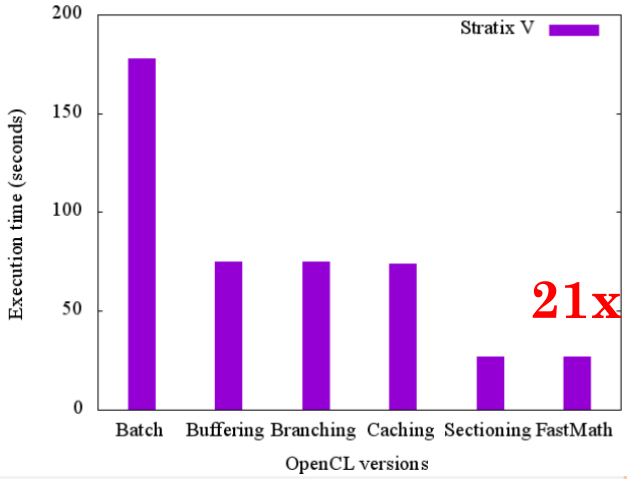
Intel i7 CPU



NVIDIA GTS 450



Altera Stratix V



Efficiency of the three computing engines.

	Time (s)	Power (W)	Photons/Joule
Core i7 (Seq. MCML)	577	20	867
Core i7 (4 threads)	52	70	2,747
Core i7 (8 threads)	40	110	2,247
GTS 450	9	95	11,103
Stratix V	27	13	28,490

Performance vs. Development Time

- GPU

Programming scheme	Exec. time (Seconds)	Dev. time (Months)
OpenCL (Our work)	9	1.5 (1 programmer)
CUDA (Lo et al. [9, 1])	2	2.5 (1-2 programmers)

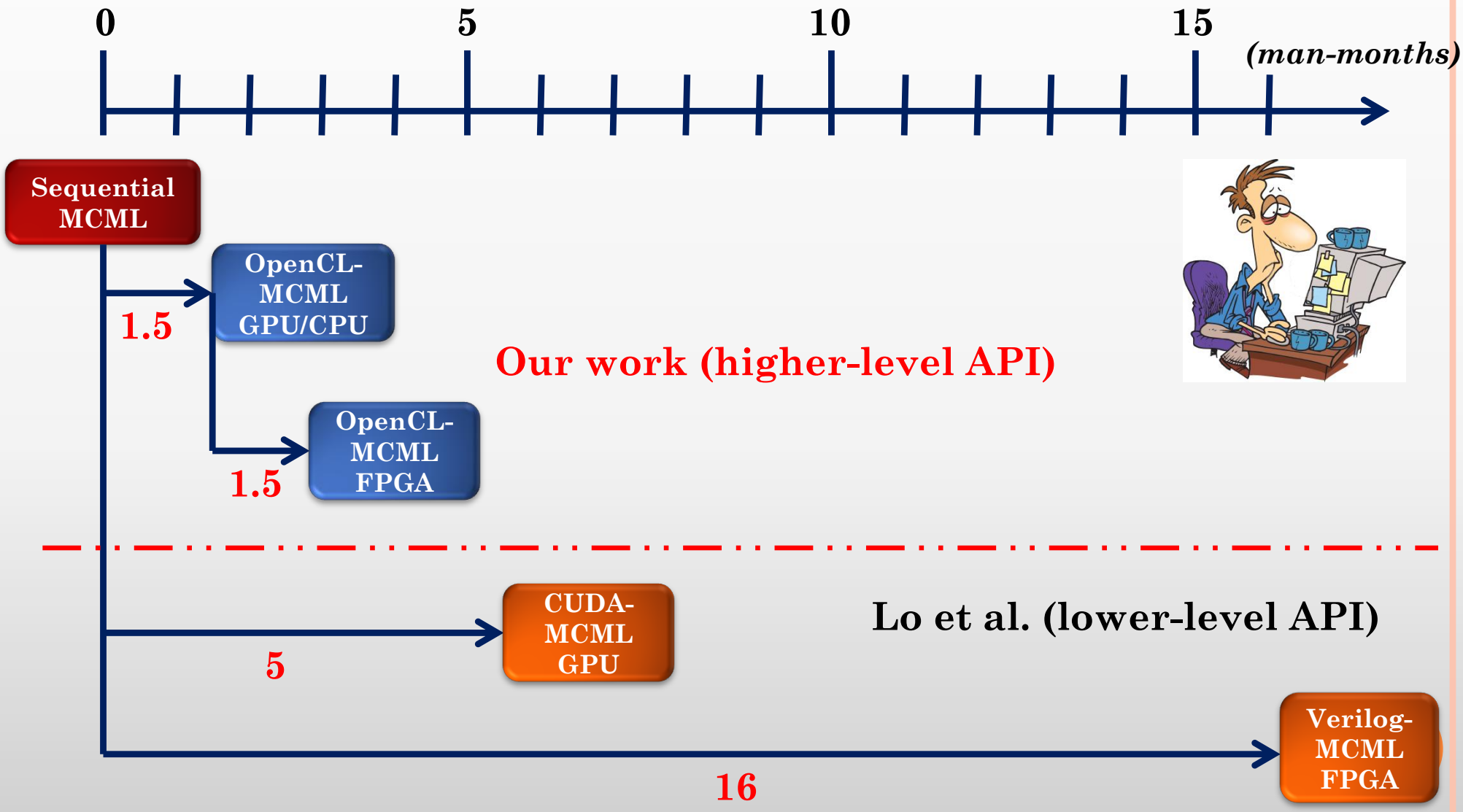
- FPGA

Performance and development efforts for FPGA.

Programming scheme	Processed photons per second	Dev. time (Months)
OpenCL (Our work)	370.3K	3 (1 programmer)
Verilog (Lo et al. [10])	454.5K	4 (2-4 programmers)

NTU PAS Lab
Performance, Applications, Security

Faster Development Time



http://bortomljusochmorker.blogg.se/2012/january/

NTU PAS Lab
Performance, Applications, Security

CONCLUSION

- With high-level programming scheme (OpenCL)
  - Short development time, higher software portability and simpler software maintenance.
  - Achieve up to 21x and 64x speedups with FPGA and GPU respectively against the sequential version.
  - FPGA version consumes 7.3x less power than GPU version.
  - We believe that such an agile application development and deployment path is desirable as GPU and FPGA both become increasingly popular in the future.


臺灣大學
National Taiwan University


NTU PAS Lab
Performance, Applications, Security

THANK YOU