

20  
VIS 04  
austin, texas  
October 10-15

# GP GPU

---



Aaron Lefohn

Institute for Data Analysis  
and Visualization  
University of California, Davis



Ian Buck

Graphics Lab  
Stanford University



John Owens

Institute for Data Analysis  
and Visualization  
University of California, Davis



Robert Strzodka

Caesar Institute  
Bonn, Germany

---

---

# Introduction and Overview

---



Aaron Lefohn

Institute for Data Analysis and Visualization  
University of California, Davis

# Motivation

---

- Challenge Statement

- GPGPU signifies the dawn of the desktop parallel computing age

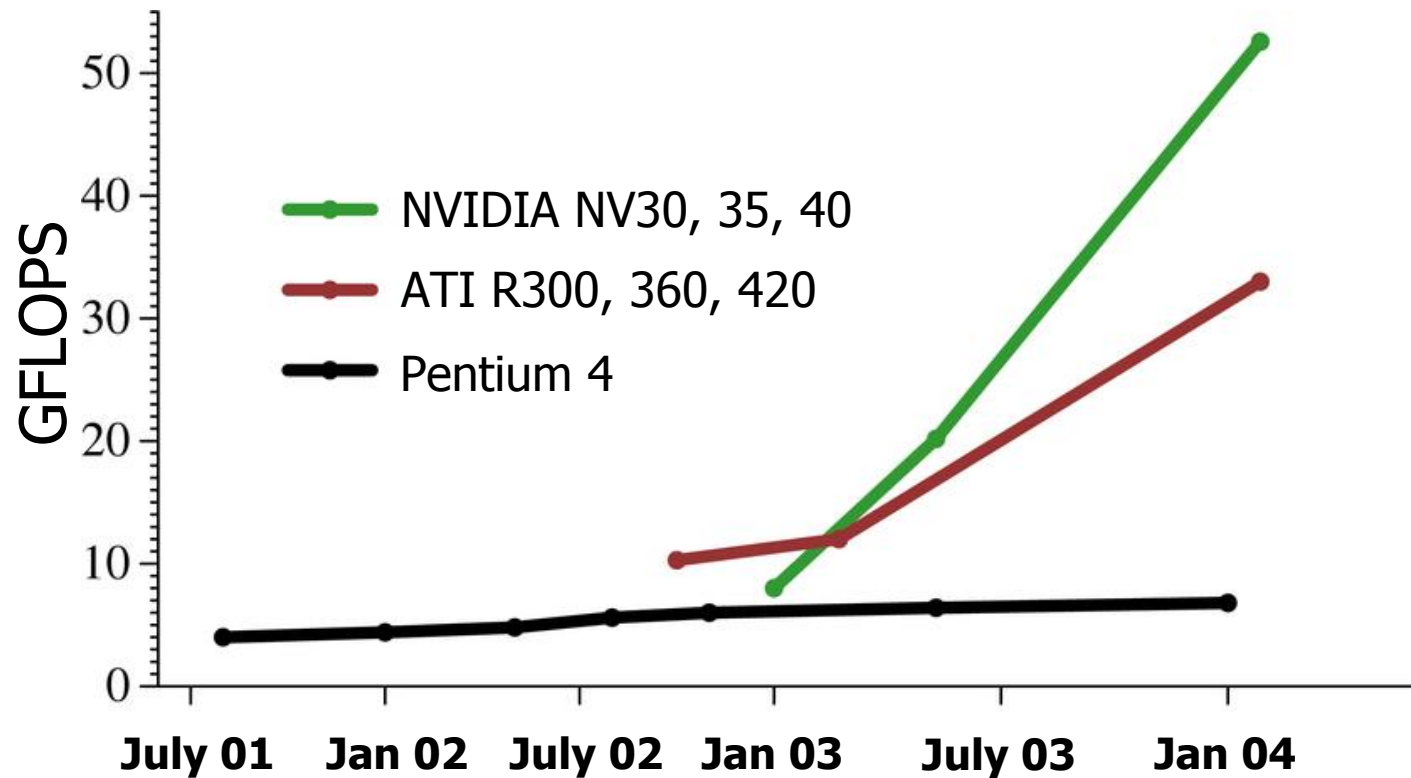
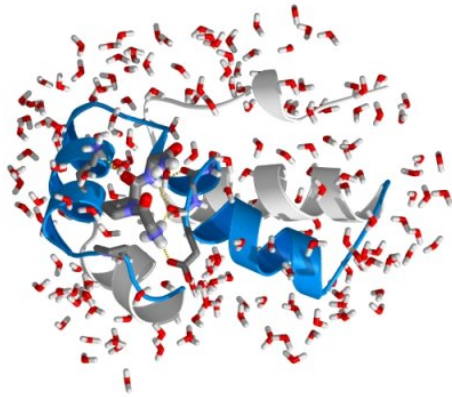


Figure courtesy of Ian Buck, Stanford University

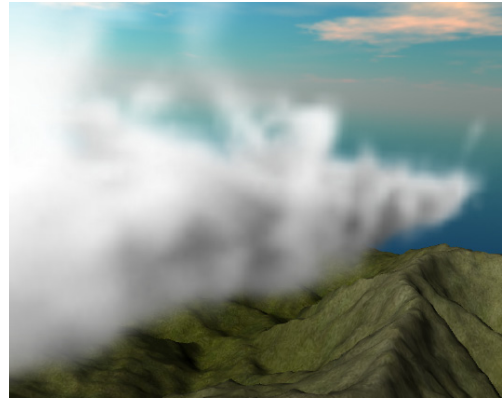


# Motivation: GPU Compute Power

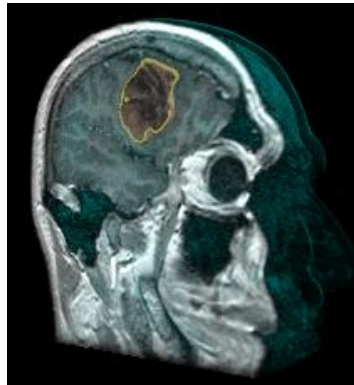
---



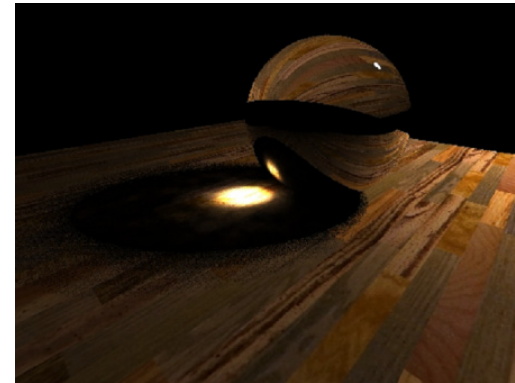
**Molecular Dynamics (Buck)**



**Cloud Simulation (Harris)**



**Level-Set Surface Deformation (Lefohn)**



**Photon Mapping (Purcell)**



# Tutorial Introduction

---

- General-Purpose Computation on GPUs (GPGPU)
  - Modern GPUs can accelerate “appropriate” algorithms  $>10x$
  - Achieving this speedup currently requires a large amount of GPU-specific knowledge—We want to change this!
- Tutorial goal
  - Give visualization researchers/engineers the knowledge and tools to leverage the computational power of the GPU



# Tutorial Topics

---

- What kinds of algorithms map well to GPUs?
- Why are GPUs faster than CPUs?
- GPGPU implementation details
  - Programming model
  - Basic building blocks
  - Nitty-gritty details
  - Real-world examples
- The future of GPGPU



# Motivation: Why GPGPU?

---

- Beginning of desktop parallel computing age
  - GPUs are the first **commodity, programmable parallel** architecture
- Advantages of data-parallelism
  - GPUs are >10x faster than CPU for appropriate problems
  - GPU performance increasing faster than CPU performance
- Advantages of commodity
  - GPUs are inexpensive
  - GPUs are ubiquitous: Desktops, laptops, PDAs, cell phones
  - 1980's data-parallel architectures cost millions of dollars



# Motivation: Why GPGPU Now?

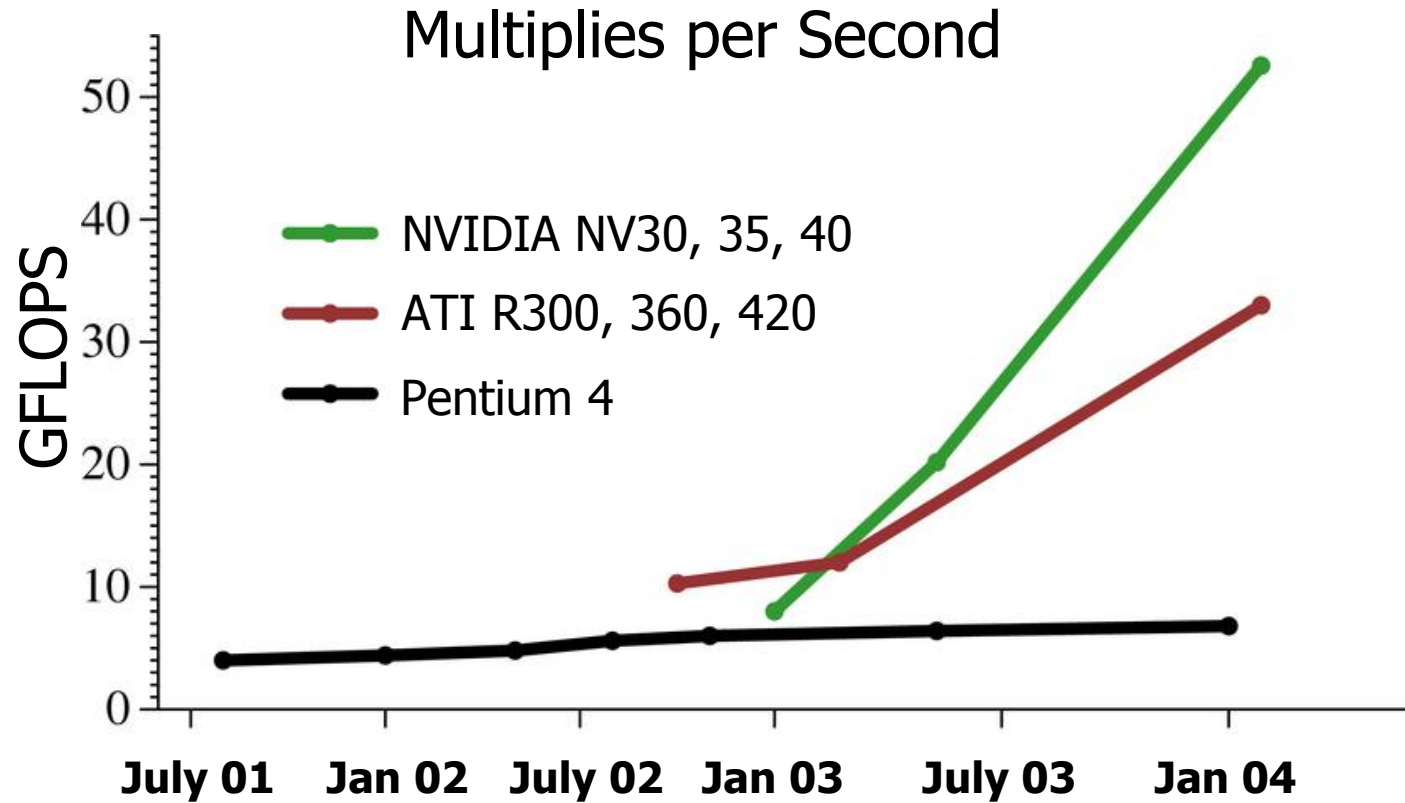
---

- GPU feature set only recently matured
  - Programmability 2001
  - Read/write memory 2001
  - Floating point 2002
  - Conditional execution (?) 2004...
- GPU evolution driven by computer game market
- GPU power rapidly increasing



# Motivation: GPU Compute Power

---



● Figure courtesy of Ian Buck



# Brief History of GPGPU

---

See <http://www.gpgpu.org> for a more complete history

1990	Lengyel	Motion planning
1999	Hoff	Voronoi diagrams
2000	Peercy	RenderMan with OpenGL
2001	Strzodka	2D PDE image processing
2002	Purcell / Carr Harris	Ray tracing Cellular automata
2003	Krueger / Boltz / Goodnight Lefohn Rottger / Krueger	Linear algebra 3D level-set solver Ray casting
2004	Govindaraju Buck / McCool	Database operations GPGPU languages

---



# Brief History of GPGPU

---

- Where are we now?
  - Transitioning from “What can we do” to “What should we do”
  - Examples
    - Lefohn et al., Univ. of Utah technical report 2002
      - Unpublished brute force solution gave no speedup
    - Buck et al., Siggraph 2004, BrookGPU Language
      - Clarifies GPGPU programming model
    - Fatahalian et al., Graphics Hardware 2004
      - Describes why matrix-matrix multiplication can never be fast on current GPUs



# Motivation: Why GPGPU Vis Tutorial?

---

- Visualization community can benefit from GPGPU
  - You need GFLOPS!
  - 2D and 3D image processing
    - Segmentation, registration, tone mapping, ...
  - Simulation
  - New rendering algorithms
  - Interactive *Visulation*
  - Familiarity with graphics programming makes transition easier
- Harnessing power of GPU is hard
  - Data-parallel algorithm mappings
  - Mapping from graphics primitives to compute primitives
  - Many performance pitfalls



# Tutorial Prerequisites

---

- We assume
  - Basic knowledge of interactive graphics programming
  
- Target audience
  - Researchers interested in GPGPU
  - Engineers interested in GPU acceleration of their applications
  - Attendees wishing a survey of this exciting new field



# Tutorial Speakers (Alphabetical)

---

- Ian Buck
  - Ph.D. student, Pat Hanrahan
  - Stanford University
  
- Aaron Lefohn
  - Ph.D. student, John Owens
  - University of California, Davis
  - Graphics software engineer, Pixar Animation Studios
  
- John Owens
  - Assistant professor, Electrical and Computer Engineering
  - University of California, Davis
  - Ph.D., Bill Dally and Pat Hanrahan, Stanford University
  
- Robert Strzodka
  - Staff researcher, Caesar Institute, Bonn, Germany
  - Ph.D., Martin Rumpf, University of Duisburg



# Tutorial Schedule Overview

---

- Morning

- Introduction
- GPU/data-parallel architecture overview
- GPGPU programming model and languages
- Computational building blocks

- Afternoon

- “Getting your hands dirty: Making it work”
- Case studies
- The future
- Q&A



# Tutorial Schedule

---

- Section 1: Introduction

- 8:30 Introduction and Tutorial Overview Lefohn  
*Motivation, introduction, and overview of the day*
- 9:00 A Data-Parallel Genealogy: The GPU's family tree Owens  
*Data-parallel architectures, stream processing, GPUs*

- Section 2: GPGPU Programming

- 9:30 The Programming Model Lefohn  
*Building computational primitives out of OpenGL calls*
- 10:00 Break
- 10:30 GPGPU Programming Languages Buck  
*Beyond OpenGL: Data-parallel GPU languages*



# Tutorial Schedule

---

- Section 3: GPGPU Computational Primitives

- 11:20      Mathematical Primitives      Strzodka  
*Linear algebra, PDEs, FEMs*
- 12:15      Lunch
- 1:45      General Algorithmic Primitives      Owens  
*Sorting, Searching*



# Tutorial Schedule

---

- Section 4: “Getting Your Hands Dirty”

- 2:15      Data Formatting and Addressing      Lefohn

*GPU memory model and data structures*

- 2:45      Computation Tips and Tricks      Buck

*Performance tips, working around GPU limitations*

- 3:15      Developer Tools      Strzodka

*Compiling, debugging, profiling*

- 3:45 - 4:15      Break



# Tutorial Schedule

---

- Section 5: Case Studies

- 4:15      Level-Set Surface Deformation      Lefohn  
*Computation and visualization of dynamic, sparse PDEs*
  
- 4:35      Advanced Image Processing      Strzodka  
*Registration, segmentation, and skeletons*
  
- 3:55      Ray Tracing and Molecular Dynamics      Buck  
*Particle simulations and light transport*



# Tutorial Schedule

---

- Section 6: Conclusions

- 5:15      The Future      Owens  
*The future of commodity data-parallel computing*
  
- 5:30      Open Question and Answer      All  
Q & A



# Coming Next...

---

- “A Data-Parallel Genealogy: The GPU’s Family Tree”
  - John Owens
  - Introduction to data-parallel and streaming architectures
  - The bigger picture of GPGPU

