GPU Computing
and
General Purpose Computation on GPUs

Summer Term 2016
Kick-Off Meeting
20th April 2016

Max-Gerd Retzlaff, Emanuel Schrade, Tamás Szép
Computer Graphics Group
Karlsruhe Institute of Technology

Evolution of GPUs

1995 NV1 1 Million Transistors

1999 GeForce 256 22 Million Transistors

2002 GeForce 4 63 Million Transistors

2003 GeForce FX 130 Million Transistors

2004 GeForce 6 222 Million Transistors

2005 GeForce 7 302 Million Transistors

2006 GeForce 8 754 Million Transistors

2008 GeForce GTX 200 1.4 Billion Transistors

2010 GeForce GTX 400 3 Billion Transistors

2013 GeForce GTX 700 Up to 7 Billion Transistors

1995 (Command & Conquer)

2000 (Diablo II)

2004 (Far Cry)

2006 (Gears of War)

2011 (Crysis 2)

2014 (The Vanishing of Ethan Carter)

Data and the images courtesy of David Luebke: http://s08.idav.ucdavis.edu/luebke-nvidia-gpu-architecture.pdf
Additional images from www.nvidia.com

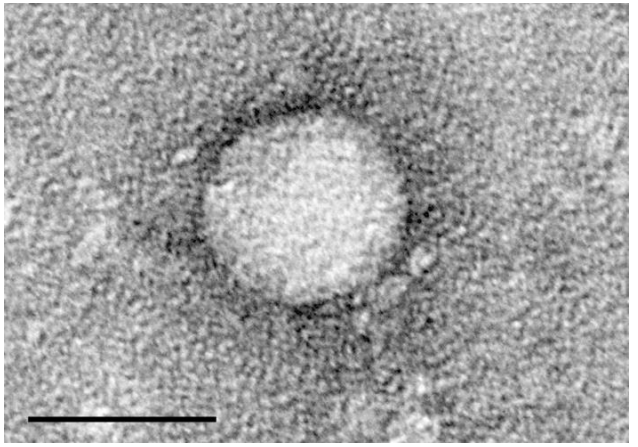
Evolution of GPUs

- Stunning evolution of entertainment graphics...

- ...but GPUs can be used for **far more** than „just“ graphics.

GPU Computing: Example #1

Mutation modeling of the Hepatitis C Virus (HCV)



- HCV
  - Major cause of liver diseases worldwide.
  - Difficult to study viral functions and drug resistance.
  - BUT: the mutation follows specific rules that can be modeled.

- Implementation on the GPU: JACKET, MATLAB®

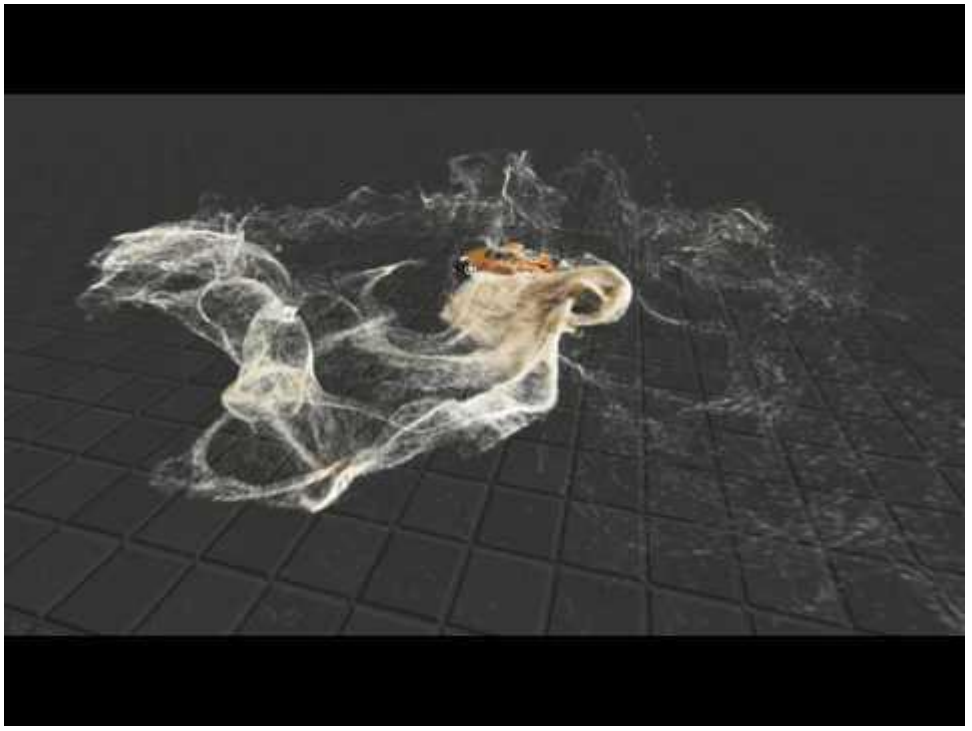
	Days	Hours	X speed up	HW cost	X $ spent	Relative P-P
Workstation-CPU Only	39	936		$2,000		
Workstation-CPU & GPU		22.5	41.6	$5,000	2.50	1664.00%
Compute Cluster		5	187.2	$250,000	125	149.76%

More about this project at http://www.accelereyes.com/examples/virus_detection_hepatitis_c

GPU Computing: Example #2

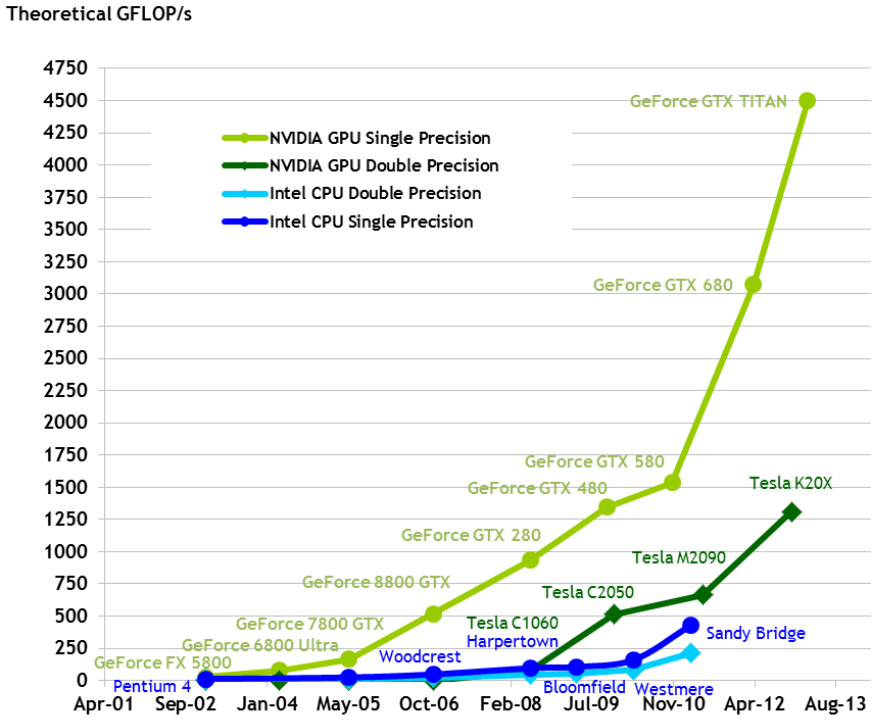
Real Time 3D Fluid and Particle Simulation and Rendering

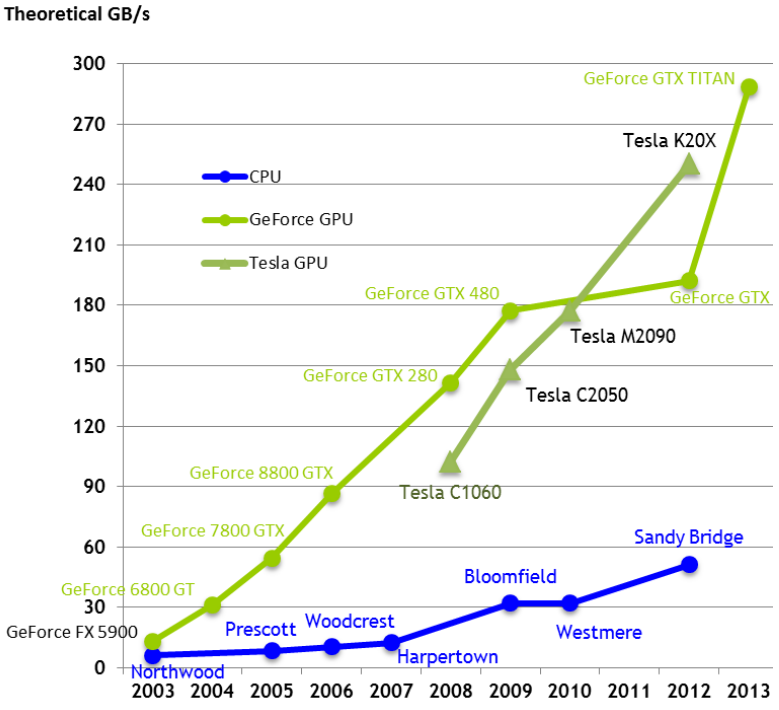
- 3D fluid solver in CUDA
- CPU → GPU speedup: 40x



GPU Computing

- Evolution of parallel computing architectures

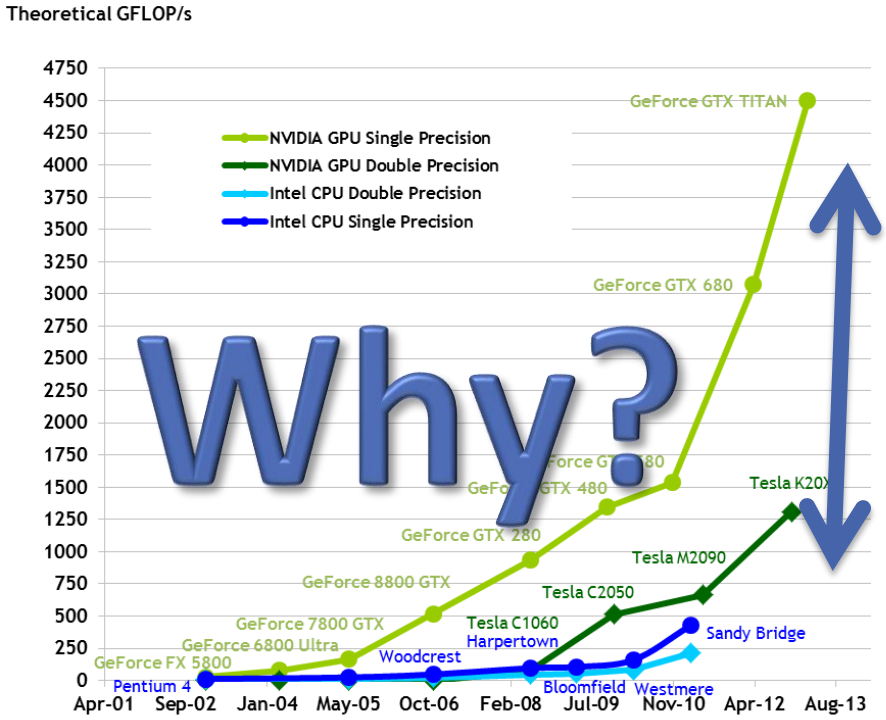


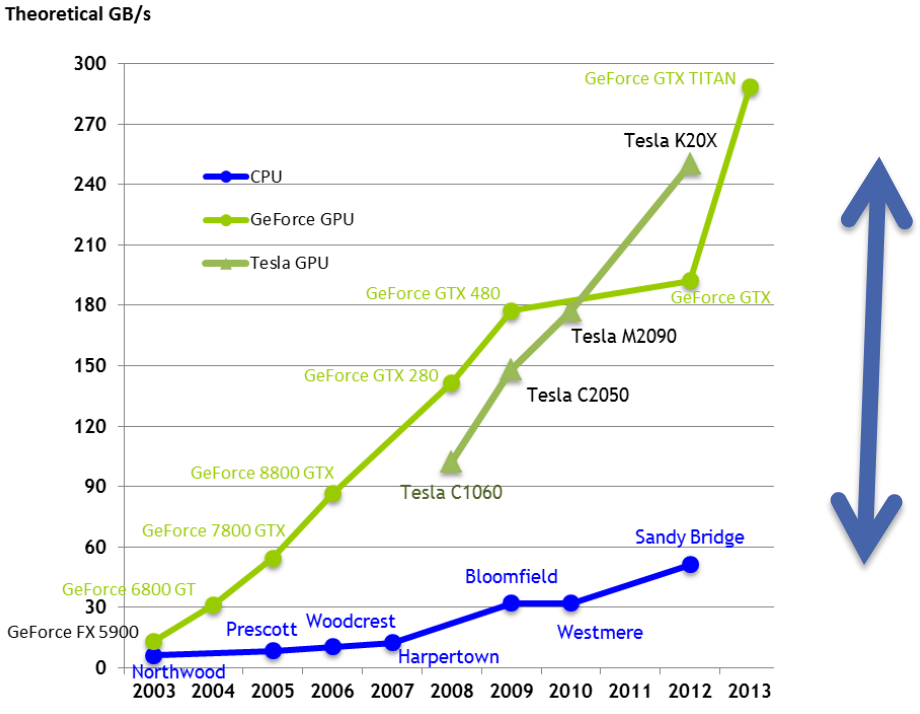


Fact: 10% of the top 500 supercomputers are GPU-accelerated

GPU Computing

- Evolution of parallel computing architectures



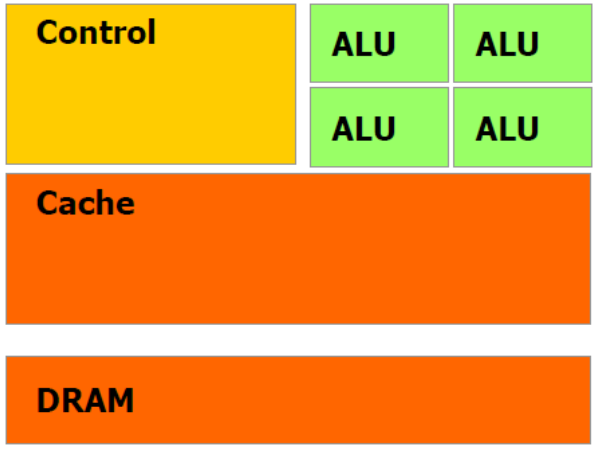


Fact: 10% of the top 500 supercomputers are GPU-accelerated

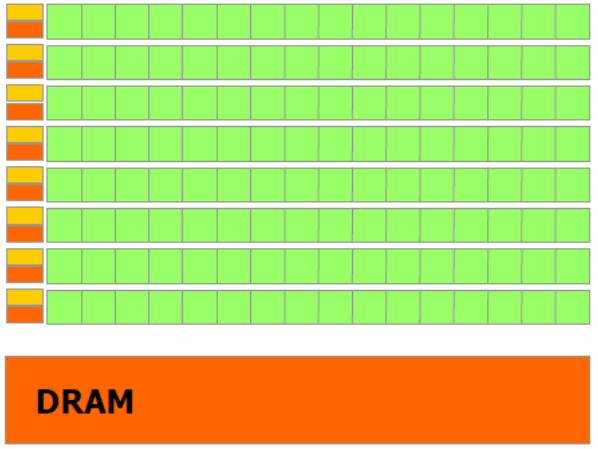
Images courtesy of Nvidia

GPU Architecture

- Task Parallelism vs. Data Parallelism
- Modern GPUs: highly data parallel suitable, compute-intensive problems



CPU

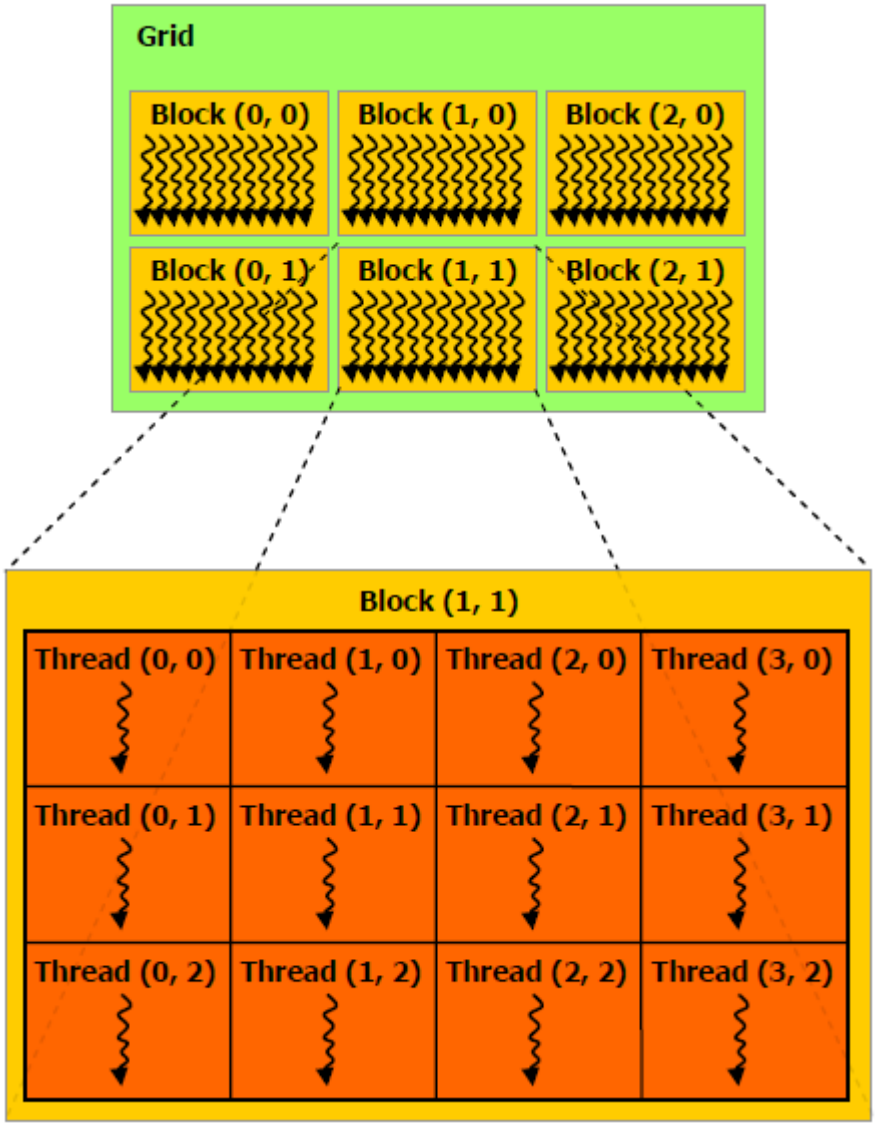


GPU

Images courtesy of Nvidia

GPU Architecture

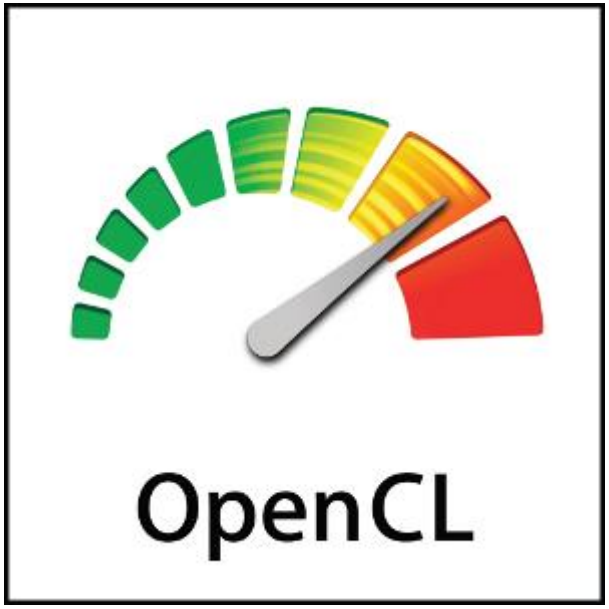
- Hardware Multithreading
  - Many-core architecture
  - Thousands of lightweight threads
  - In-order execution,
    cheap flow control
  - Latency is hidden by
    the raw number of threads.



Images courtesy of Nvidia

OpenCL

- Open Computing Language
  - cross-platform standard for computing on heterogeneous platforms
  - maintained by Khronos Group (OpenGL, OpenAL)
  - introduced at SIGGRAPH 2008



OpenCL SDK

- From **NVIDIA**:
  - http://developer.nvidia.com
    1. Download the developer driver.
    2. Download the CUDA toolkit.
    3. Download the GPU Computing SDK (optional, but worth it).

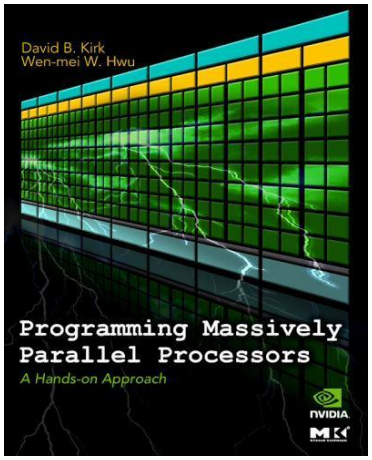
- From **ATI**:
  - http://developer.amd.com
    1. Download Accelerated Parallel Processing (APP, formerly ATI Stream).

Read the documentation and additional materials at vendors' websites!

Recommended Literature

- Programming Massively Parallel Processors
  A Hands on Approach

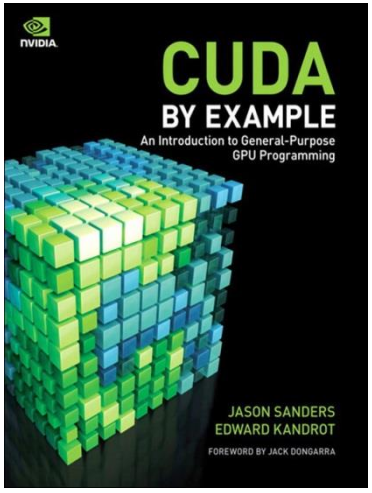
  David B. Kirk, Wen-mei W. Hwu
  Morgan Kaufmann, 2010



- CUDA by Example: An Introduction to
  General-Purpose GPU Programming

  Jason Sanders, Edward Kandrot

  Morgan Kaufmann, 2010



- The OpenCL Specification

  Khronos Group
  http://www.khronos.org/registry/cl/specs/opencl-1.1.pdf

Course Organization

Advisors

Max-Gerd Retzlaff
Room 145
retzlaff@kit.edu

Emanuel Schrade
Room 140
schrade@kit.edu

Tamás Szép
Room 166
tamas.szep@kit.edu

Please prefix the mail subject always with [gpuc] (GPU Computing) or [gpgpu] (GPGPU)!

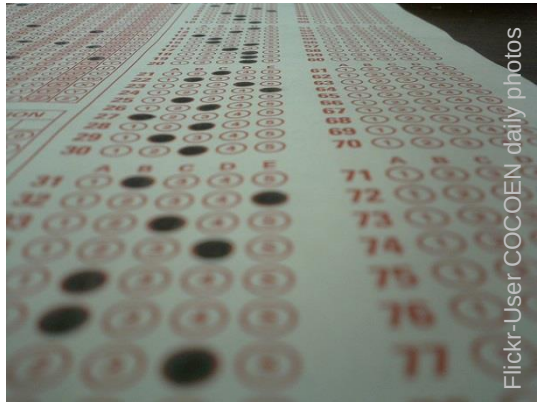
Course Organization

	GPU Computing	GPGPU
Credits:	**4 SWS**	**2 SWS**
Workload:	4 assignments + 1 free-style	4 (reduced) assignments
Requirements:	each individual assignment: at least **40%** all assignments in total: at least **50%** **... otherwise failed!**	

It is not possible to get credits for both courses!

Course Organization

	GPU Computing	GPGPU
Points:	**20** points for each assignment **100** in total (**50** to pass)	**20** points for each assignment **80** in total (**40** to pass)
Grading:	**Graded!**	**Pass/not pass**



Evaluation

- Assignments **must be submitted** to our submission system.
- The deadline is usually midnight before the evaluation, but check the submit system.
- Assignments **must be presented** by the author (in person).
- Evaluations will be held in ATIS computer pool.
  Make sure your code compiles and runs there!
- You can also use your own **laptop**. If you encounter problems, use the ATIS machines as fallback! *Especially for benchmarking!*
- Participants will have priority access on ATIS computers during Praktikum time slots.
- ATIS machines 50-53 and 66-82 are equipped with NVidia GPUs suitable for the assignments.

Evaluation

- Evaluations will be held in the course time slot (GPUC:We, 14:00; GPGPU:We, 15:45).
- GPU-Computing only:
  For the evaluation, the course will be split into three sub-groups to keep down waiting times:

Group	identified by least significant digit of the student id being…	Evaluation starts at time slot time + …
1	0, 1, 2 or 3	0
2	4,5 or 6	30
3	7, 8 or 9	60

Course Organization

Grading:

GPU Computing			GPGPU
Min Points	Max Points	Grade	
96	100	1.0	
91	95	1.3	
86	90	1.7	
81	85	2.0	
76	80	2.3	
71	75	2.7	
66	70	3.0	
61	65	3.3	
56	60	3.7	
50	55	4.0	
0	49	FAIL	

Submitting your solution

- Upload your solutions to the CG Submission system:
  https://submit.ivd.kit.edu

- Sign up if you do not have an account yet.
- In case of troubles, send us an e-mail!

Late delivery is penalized!

- Deadlines are strict, can be extended only in emergency cases.
- Late penalty:
  - -4 points if you submit after the deadline
    (Yes, even if you are only a few minutes late!)
  - -2 points/day for any additional day delay after the first
- <8 points for **any** assignment → you fail the course!

- You have to show up personally for the evaluation at the appointed date and time.
- -2 points/day if you do not show up for the evaluation!

Support

- Assistance and mentoring
  - Feel free to send us e-mails (see advisors page).
    - Please prefix the mail subject always with:
      [gpuc] (GPU Computing) or **[gpgpu]** (GPGPU)!
  - Drop by our offices.
- Mailing list **cg.praktikum@ira.uni-karlsruhe.de**
  - We'll add you to the list using the address you entered in submit.
  - announcements
  - questions of interest, (topic related) discussions

Working together and discussing the assignments is encouraged but each participant must be able to present and explain his own solution (no copies)!

Further Information

- Check the submit system for new assignments (sheet + start-up kit):

  https://submit.ivd.kit.edu

- You also have to register at the **Studierendenportal** for this Praktikum.

Assignment #1

	GPU Computing	GPGPU
Tasks:	Add two vectors of integers (one in reverse order). (Efficiently) rotate a matrix of numbers.	
Deadline:	in just under two weeks: **3.5.2016 at 23:59**	
Next Meeting:	ATIS computer pool (evaluation) in two weeks: **4.5.2015**	
	14:00	**15:45**

Assignment #2 – #5

	GPU Computing	GPGPU
Topics:	■ #2: Parallel algorithms (parallel reduction, prefix sum) ■ #3: Image filtering (discrete convolution) ■ #4: Particle systems, cloth simulation ■ #5: Freestyle	

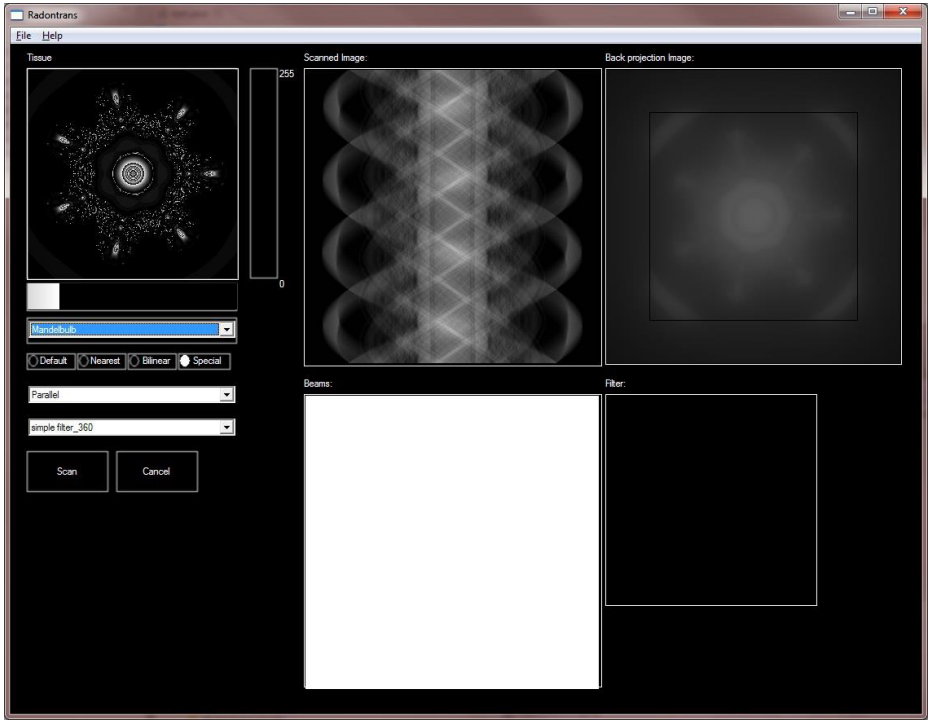
The next assignment will usually be published on the day of the evaluation.

Freestyle assignment (GPU-Computing only)

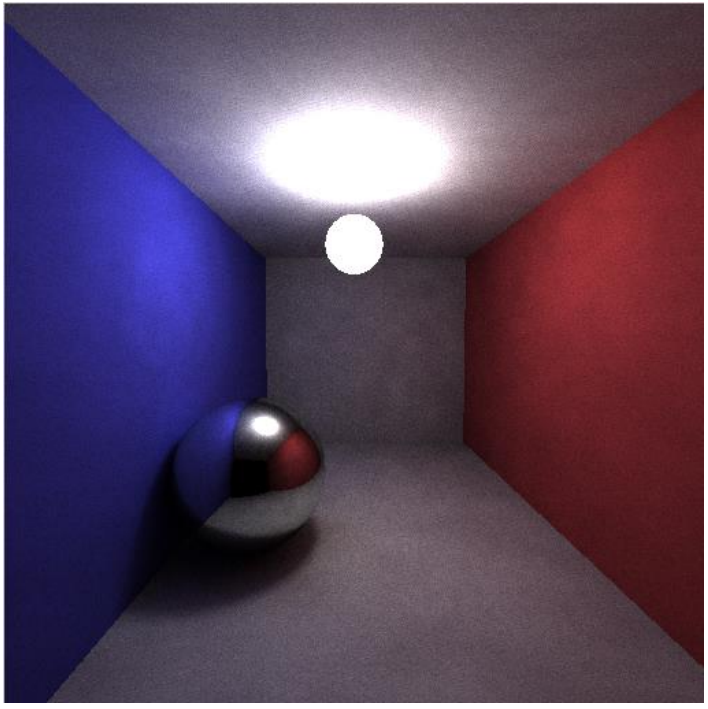
- You propose a project you would like to work on as a final assignment.
- We will check your proposal and maybe ask you to improve it.
- After the proposal has been accepted you can start working.
- Each participant will be required to present the results of the freestyle assignment in a ~10min talk to all participants.

Freestyle Examples

Simulated CT on the GPU
Thorsten Gröninger

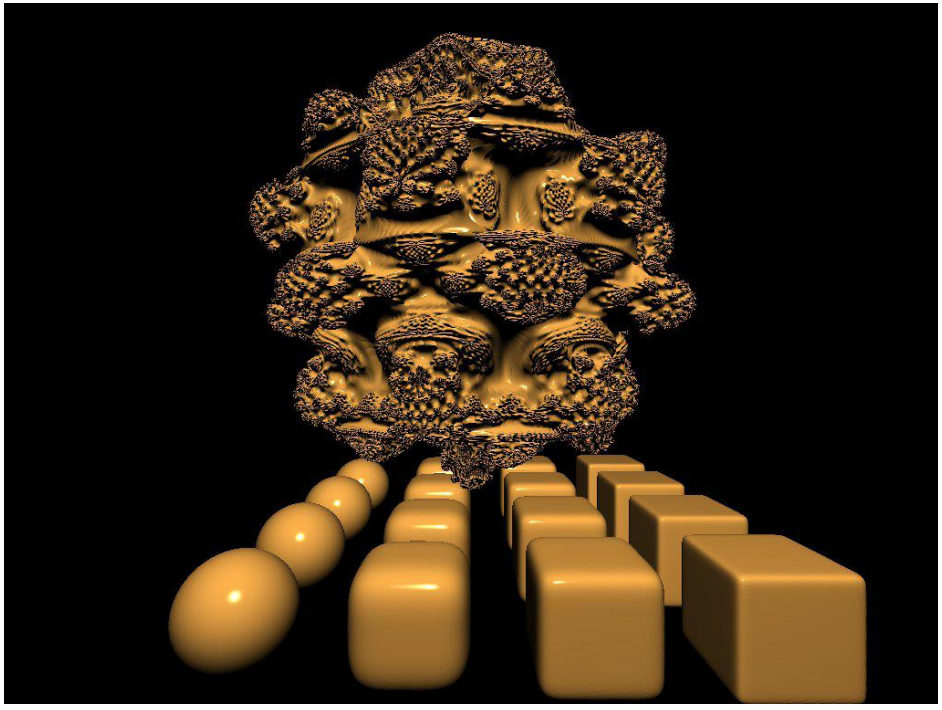


Path tracing on the GPU
Martin Tillmann

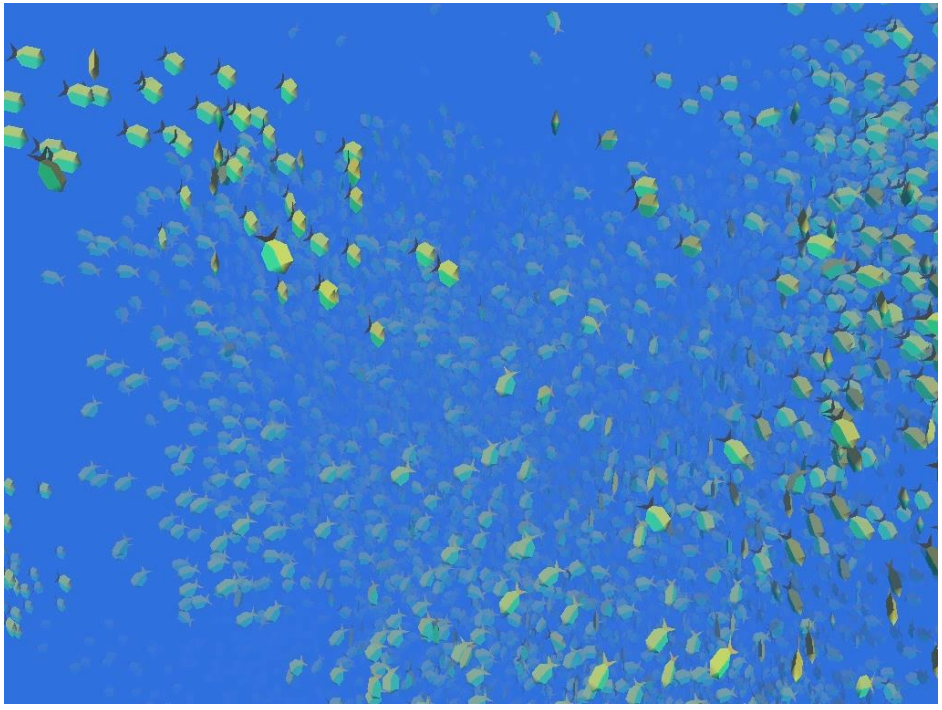


Freestyle Examples

Fractal ray-tracing on the GPU
Manuel Martin



Simulation of fish schools
Alexander Wirth



Karlsruhe Institute of Technology

Questions?