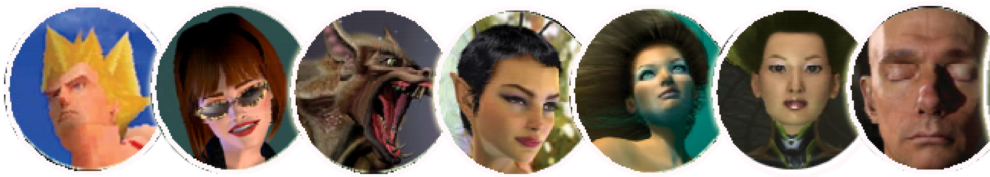
GPU Computing
and
General Purpose Computation on GPUs

Winter Term 2013
Kick-Off Meeting
23 October 2013

Anton S. Kaplanyan, Gregor Mückl and Christoph Schied
Computer Graphics Group
Karlsruhe Institute of Technology

Evolution of GPUs

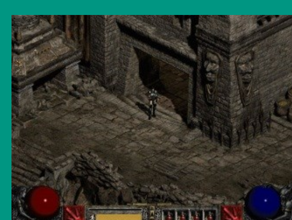


1995	1999	2002	2003	2004	2005	2006
NV1	GeForce 256	GeForce 4	GeForce FX	GeForce 6	GeForce 7	GeForce 8
1 Million Transistors	22 Million Transistors	63 Million Transistors	130 Million Transistors	222 Million Transistors	302 Million Transistors	754 Million Transistors

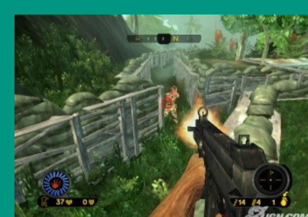
2008
GeForce GTX 200
1.4 Billion Transistors

2010
GeForce GTX 400
3 Billion Transistors

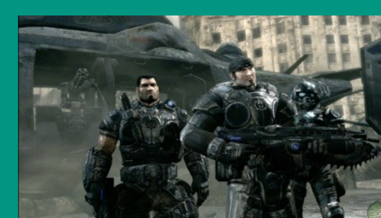
1995
(Command & Conquer)



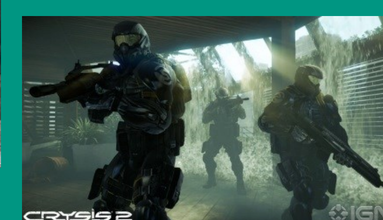
2000
(Diablo II)



2004
(Far Cry)



2006
(Gears of War)



2011
(Crysis 2)

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

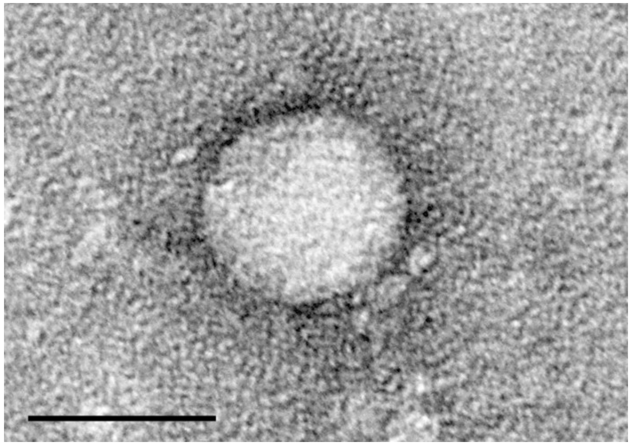
Evolution of GPUs

- 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®

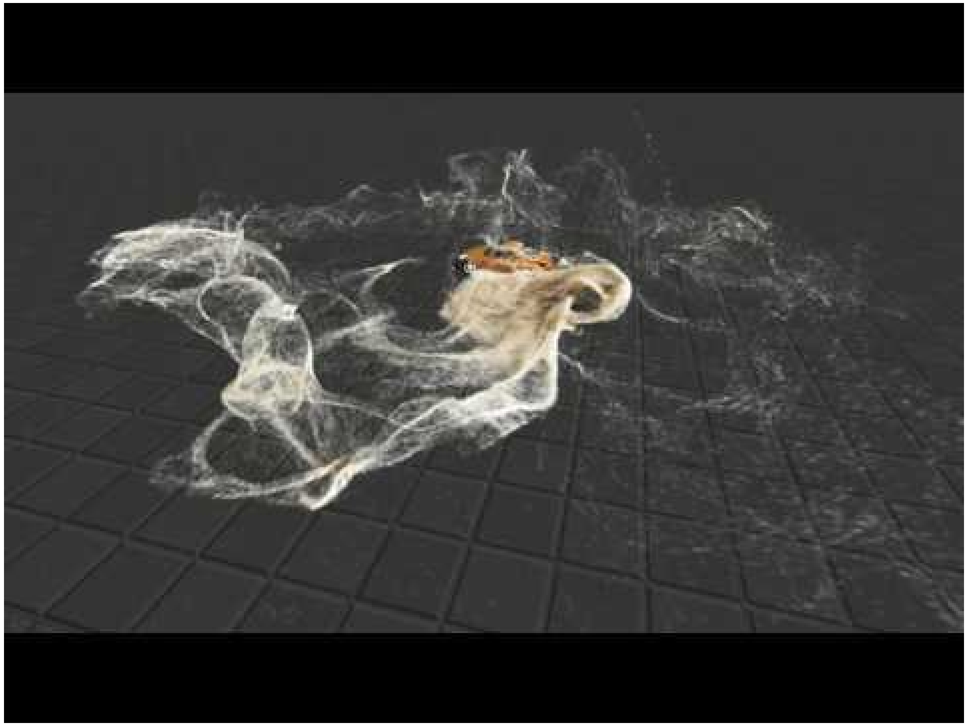
	Days	Hours	X speed up	HW cost	X $ spent	Relative P-P
Workstation-CPU Only	39	936		$2,000		
Workstation-CPU & GPU		22.5	41.6	$5,000	2.50	1664.00%
Compute Cluster		5	187.2	$250,000	125	149.76%

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

GPU Computing: Example #2

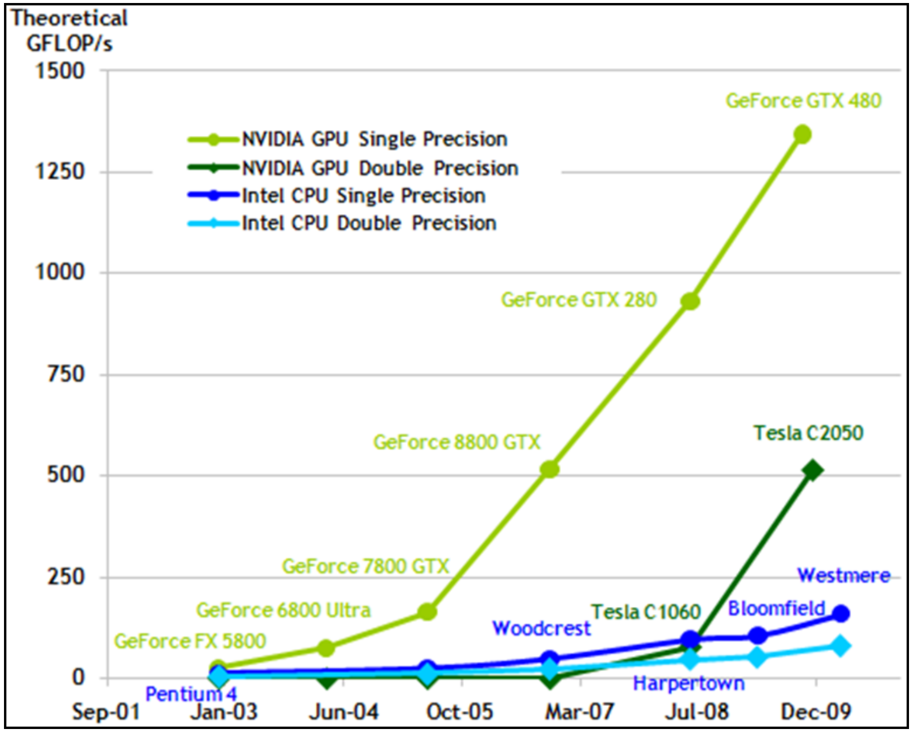
Real Time 3D Fluid and Particle Simulation and Rendering

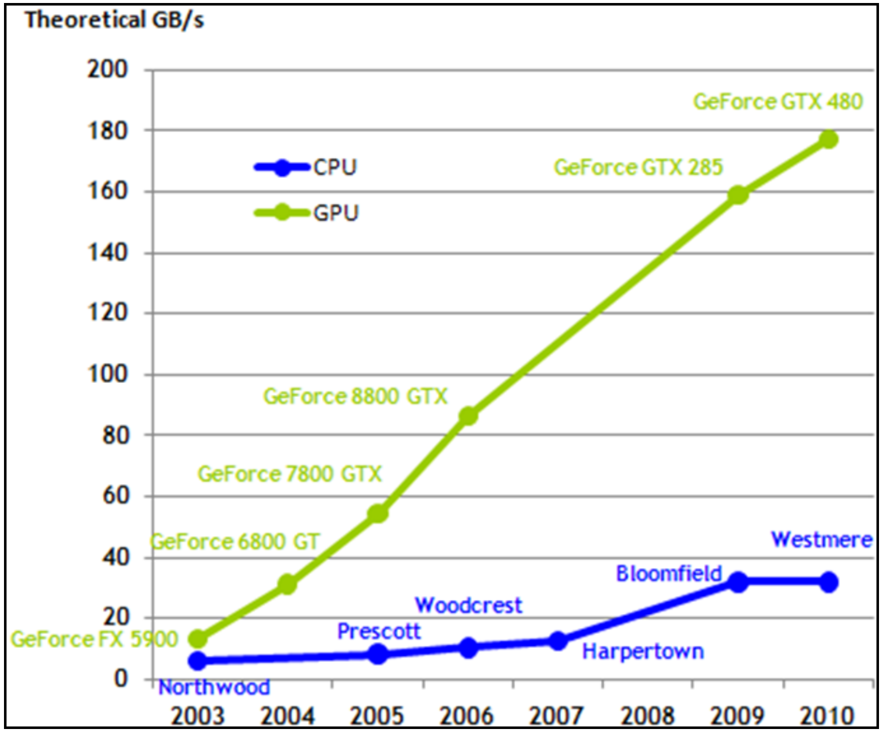
- 3D fluid solver in CUDA
- CPU → GPU speedup: 40x



GPU Computing

- Evolution of parallel computing architectures

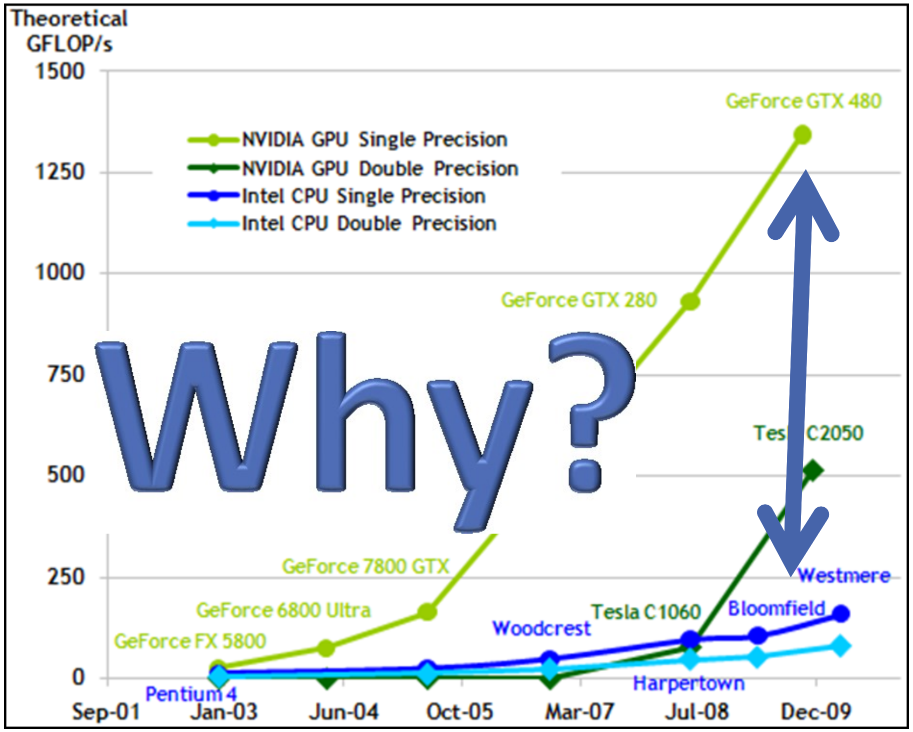


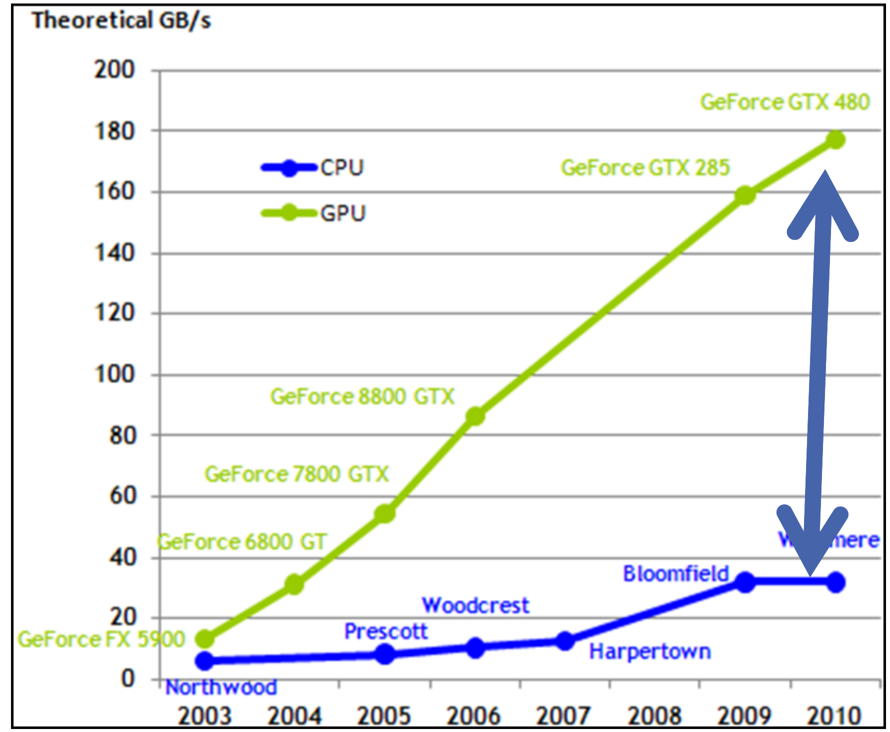


Fact: 10% of the top 500 supercomputers
are GPU-accelerated

GPU Computing

- Evolution of parallel computing architectures



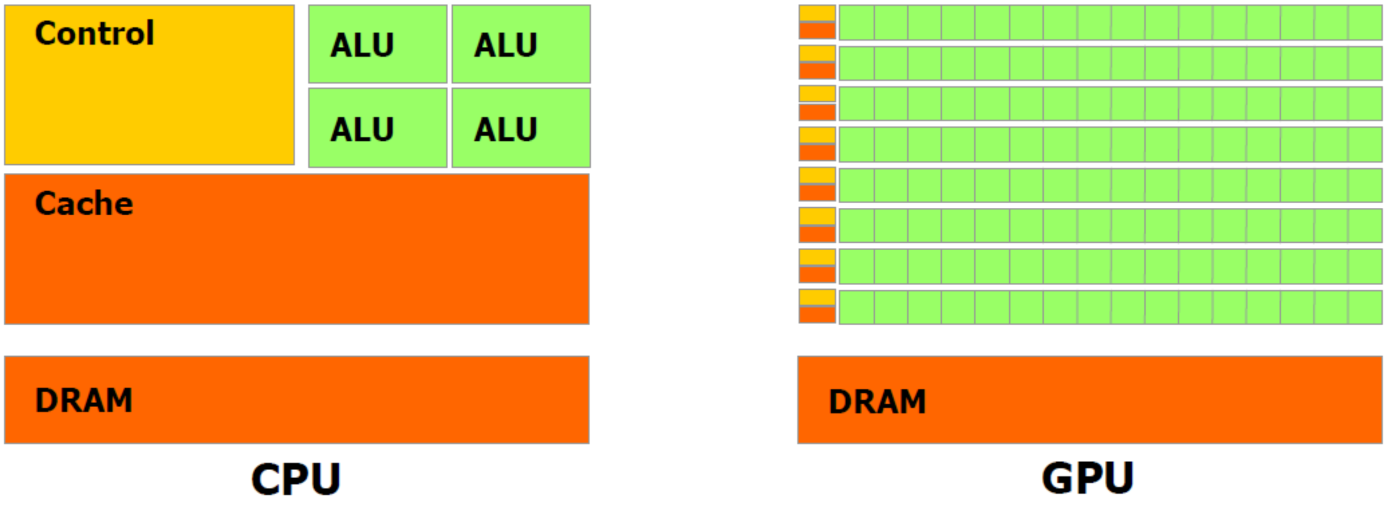


Fact: 10% of the top 500 supercomputers
are GPU-accelerated

Images courtesy of Nvidia

GPU Architecture

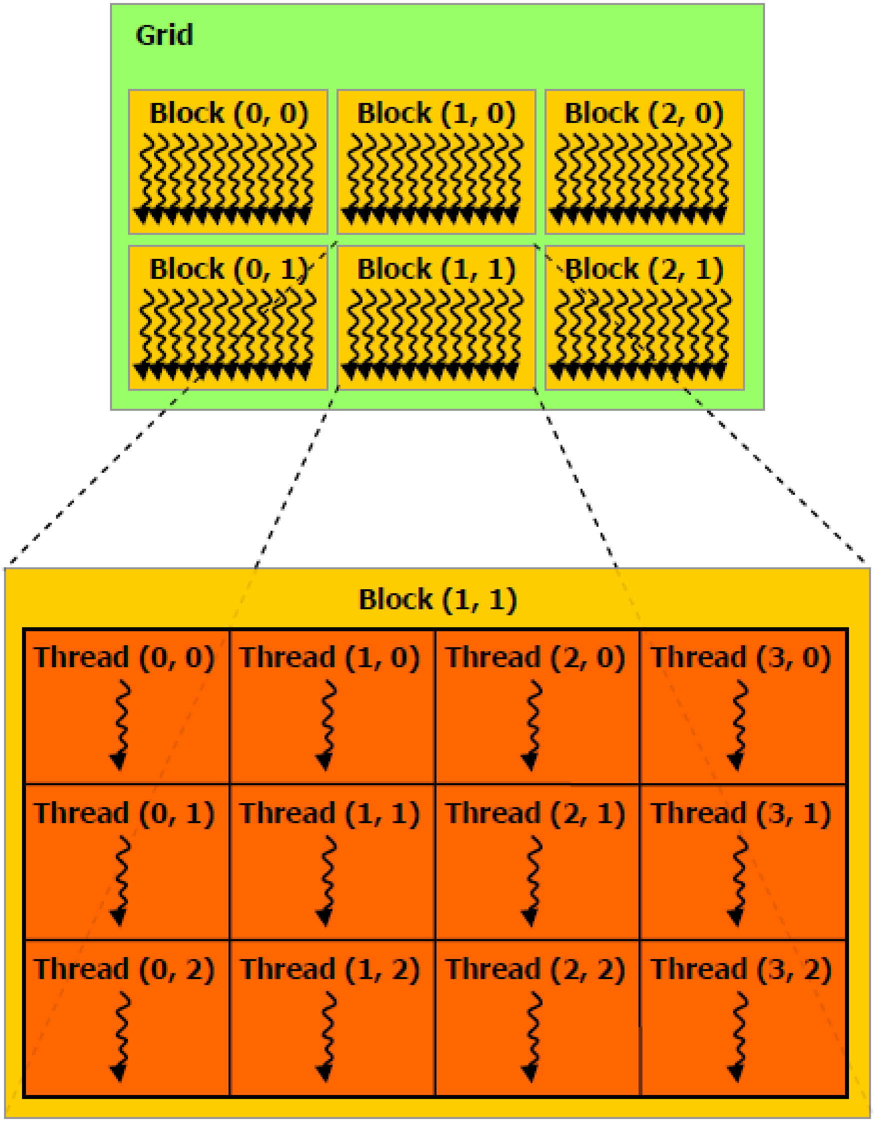
- Task Parallelism vs. Data Parallelism
- Modern GPUs: highly data parallel suitable, compute-intensive problems



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)

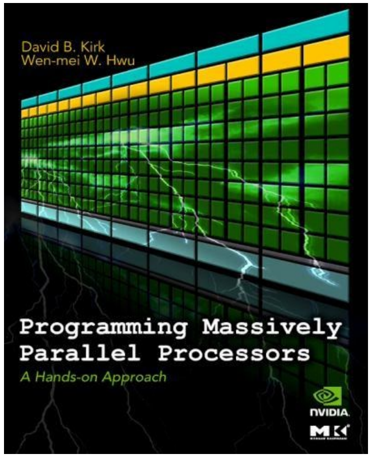
Read the documentation and additional materials at vendor's websites!

Karlsruhe Institute of Technology

Recommended Literature

- Programming Massively Parallel Processors
  A Hands on Approach

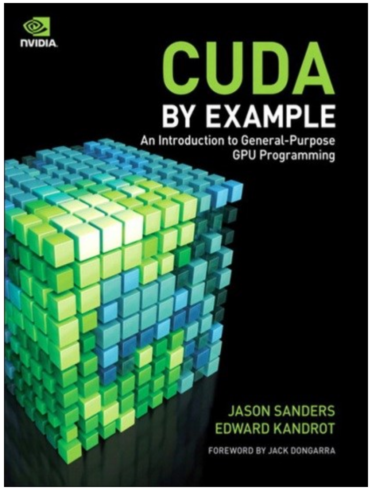
  David B. Kirk, Wen-mei W. Hwu
  Morgan Kaufmann, 2010



- 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-1.0.29.pdf

Course Organization

People

Anton S. Kaplanyan

Room 141

anton.kaplanyan@kit.edu

Gregor Mückl

Room 145

gregor.mueckl@kit.edu

Christoph Schied

Room 136

christoph.schied@kit.edu

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!**	

Course Organization

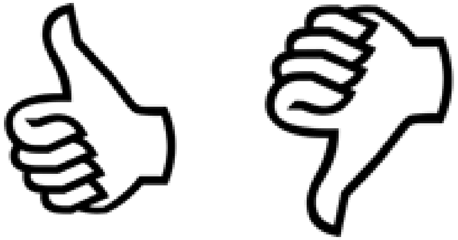
	GPU Computing	GPGPU
Points:	**20** points for each assignment **100** in total (**50** to pass)	**20** points for each assignment **80** in total (**40** to pass)
Evaluation:	- Assignments **must be submitted** to our submission system - Assignments **must be presented** by the author (in person) - Presentations will be held in ATIS computer pool **Make sure your code compiles and runs there!** - You can also use your own laptop.	
Grading:	**Graded!**	**Pass/not pass**

Course Organization

	GPU Computing	GPGPU

Grading:

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



Assignment #1

	GPU Computing	GPGPU
Tasks:	Add two vectors of integers (one in reverse order) (Efficiently) rotate a matrix of numbers	
Next Meeting:	ATIS computer pool (evaluation)	
	In two weeks (**6.11.**) **12:00**	In two weeks (**6.11.**) **11:30**

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	

Submitting your solution

- Upload your solutions to the CG Submission
  submit.ibds.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
- -2 points/day after the deadline
- < 8 points for **any** assignment → you fail the course!

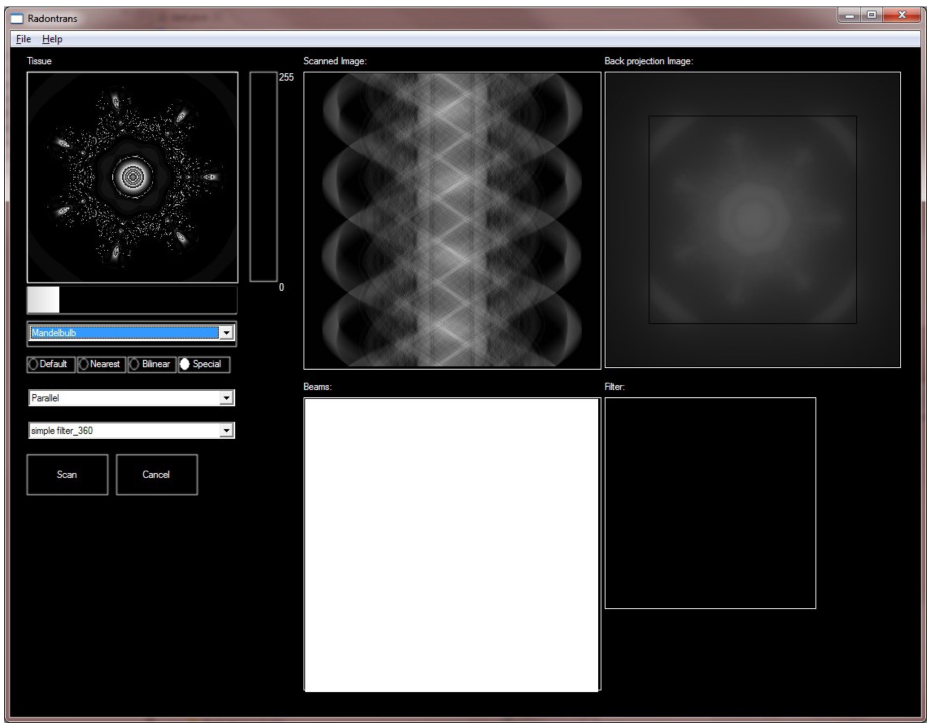
Further Information

- Check website for new assignments (sheet + start-up kit):
  - cg.ibds.kit.edu/lehre/ws2013/
- Assistance and mentoring
  - Feel free to send us e-mails
  - Drop by our offices

Freestyle Examples

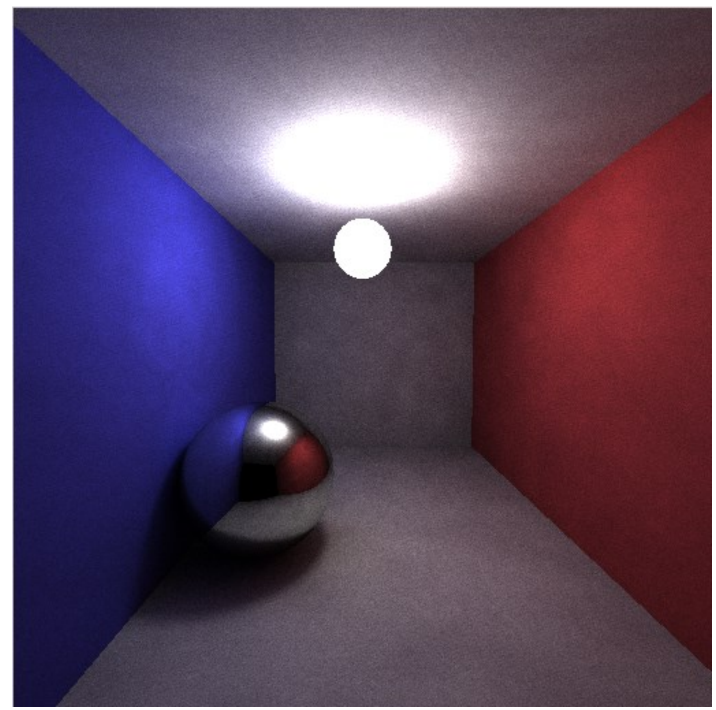
Simulated CT on the GPU

Thorsten Gröninger



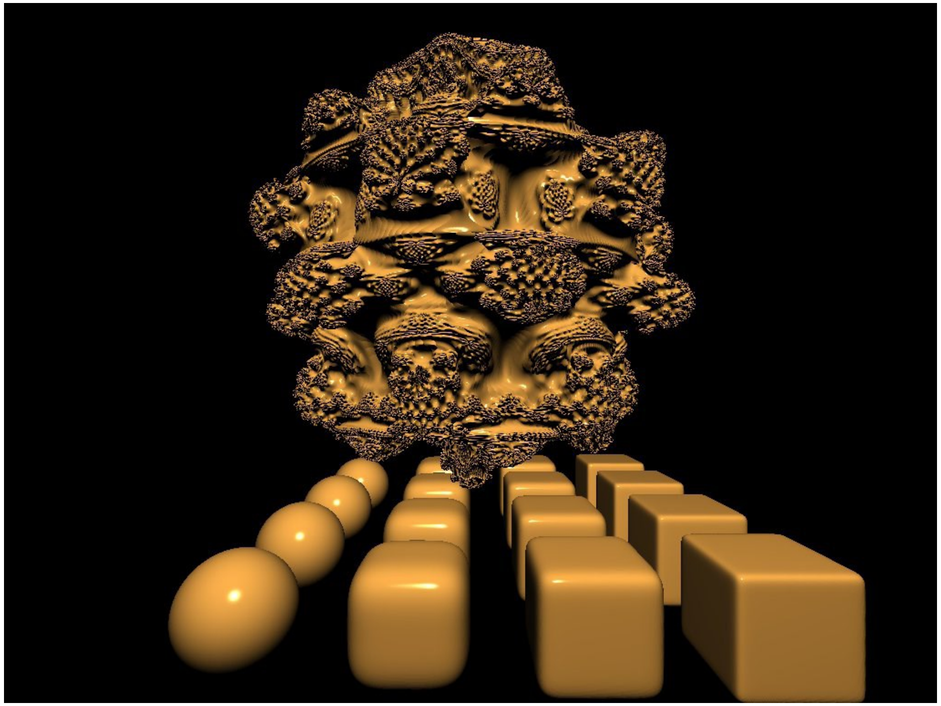
Path tracing on the GPU

Martin Tillmann

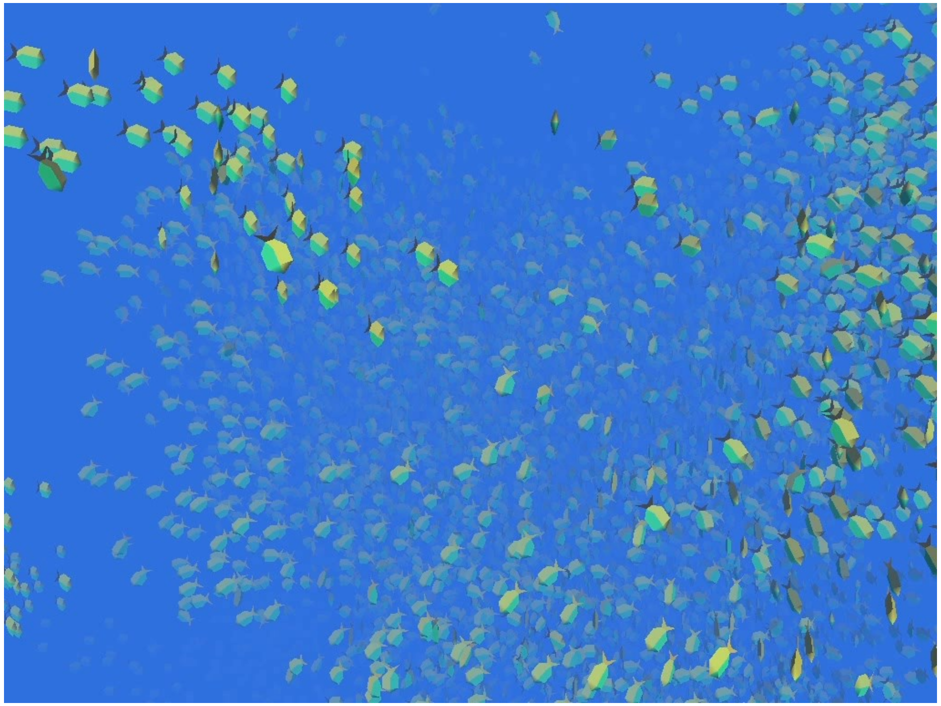


Freestyle Examples

Fractal ray-tracing on the GPU
Manuel Martin



Simulation of fish schools
Alexander Wirth



Questions?