

Real-Time Rendering (Echtzeitgraphik)



Michael Wimmer
wimmer@cg.tuwien.ac.at





- Michael Wimmer
 - Associate Professor am
 - Institut für Computergraphik und Algorithmen
(<http://www.cg.tuwien.ac.at>)
- Lehre:
 - UE Einführung in die Computergraphik, UE Computergraphik, VU Echtzeitgraphik
- Forschung:
 - Echtzeitgraphik: Schatten, Sichtbarkeit, Image-Based Rendering, Games, Modellierung, Point-Based Graphics, ...



- Zeit: Mittwoch, 15:15-16:45
- Termine: 12 Einheiten
genaue Einteilung am Web (wichtig!!!)
- Ankündigungen: TISS
- Vorlesungshomepage:
www.cg.tuwien.ac.at/courses/Realtime
- Beurteilung:
praktischer Teil + mündliche Prüfung
- Anrechenbarkeit...



- Ort: Seminarraum 186
- Lecture support:
 - Videoaufnahme?
 - ??



- VU: Vorlesung mit Übung
- Kleines Demo-Projekt in 2er-Gruppen
 - Implementierung von “ein paar Techniken”
 - Ev. in bestehendes **CGUE-Spiel**

2002

2007

- 3 Abgaben
- Präsentation am Ende des Semesters (25.1. Anwesenheitspflicht!!)
- Betreuung durch Tutoren im Informatik-Forum (Echtzeitgraphik-Forum)



- 0. Abgabe (17.10.): **Projektvorschlag**
 - Welche Effekte
 - Quellenangaben!!!
- 1. Abgabe (28.11.): **“Rendering-”Engine**
 - OpenGL-Rendering
 - Kamera
 - Texturen
- 2. Abgabe (16.1.): **Fertiges Projekt**
 - Implementierung der Effekte, “schönes” Demo



- Prerequisites
 - Needs to run on Windows 8.1 x64!
 - PC with NVIDIA GTX 960 or AMD R9 380!
 - More details on website when PCs are ready
 - Graphics API:
 - OpenGL 3.2+ core profile
 - DirectX 10+
 - Vulkan
 - Needs to use fragment shaders
 - Need to explain in assignment
 - what effects
 - what sources (web pages, papers, tutorials, ...) were used



■ Content

- Total Textures
- We have access to the full repository!
- <https://lva.cg.tuwien.ac.at/cgue/textures/>
 - user: student
passw: we4tex13



■ Abgabesystem

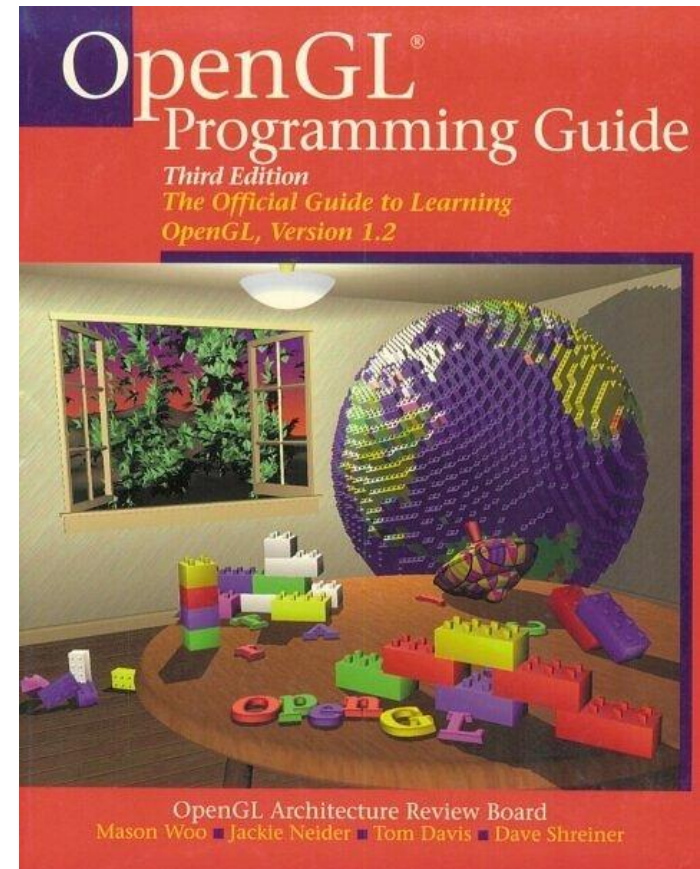
- Same as CGUE!
- Need to subscribe in TISS, then login to Abgabesystem (lva.cg.tuwien.ac.at/rtr)
- Need to use GIT and do regular commits!
 - Assignment is not complete without an up-to-date GIT repository!



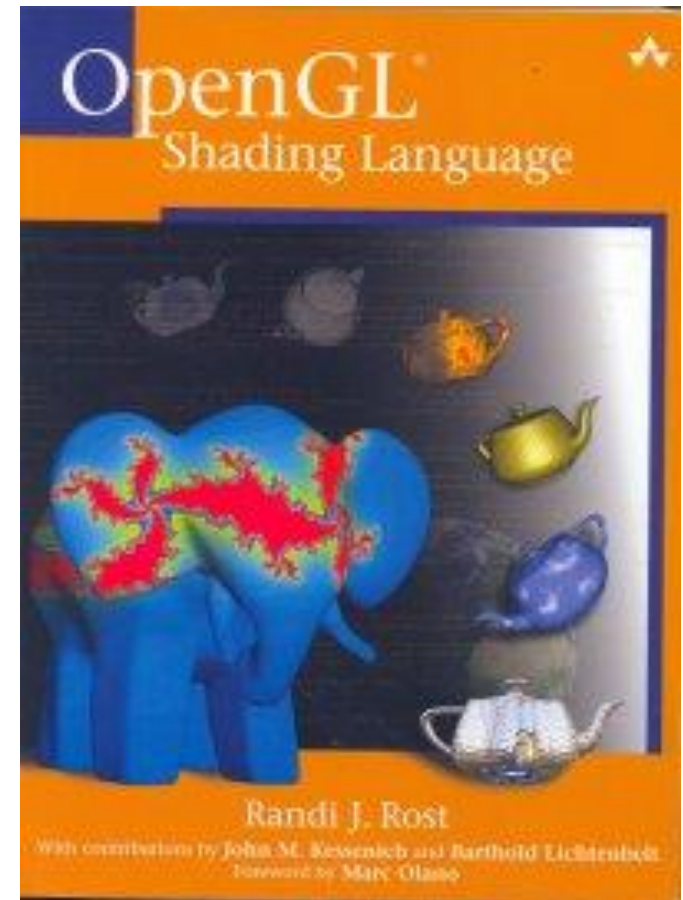
- Debugging
 - AMD CodeXL
 - NVIDIA nSight



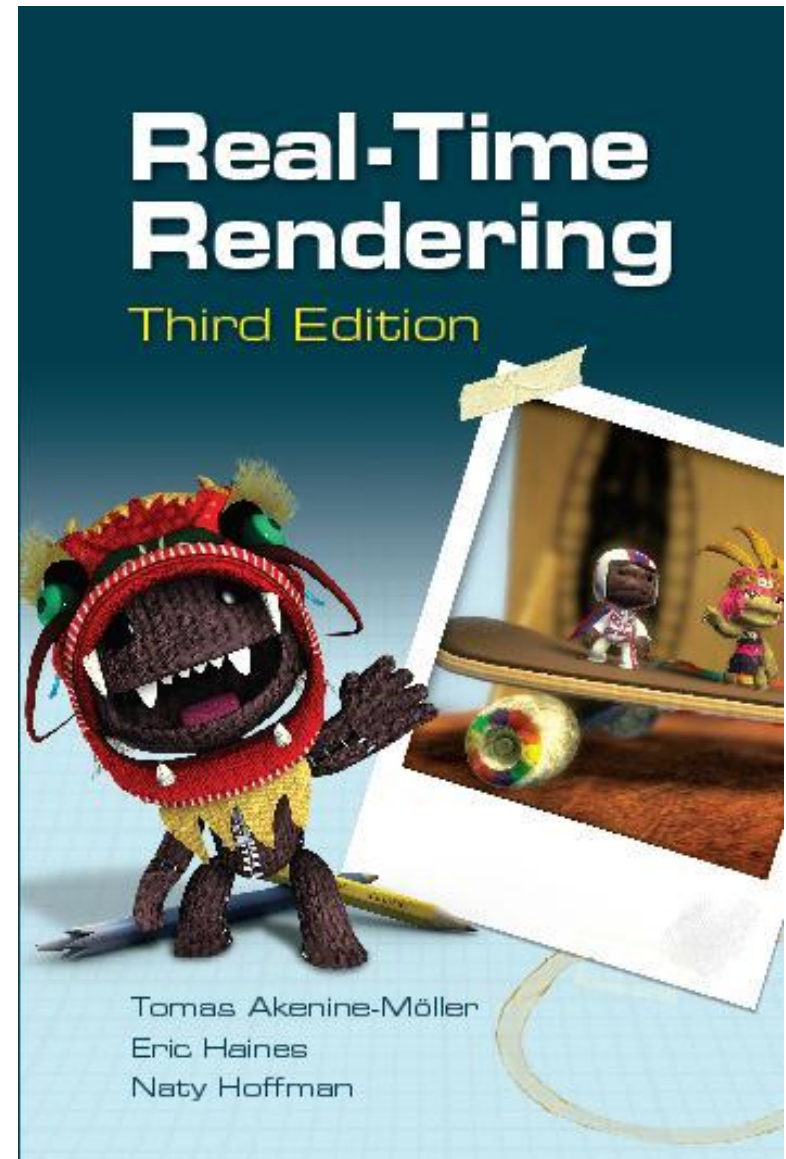
- Basic Computer Graphics course
- Some knowledge about OpenGL
 - Google: “redbook pdf”



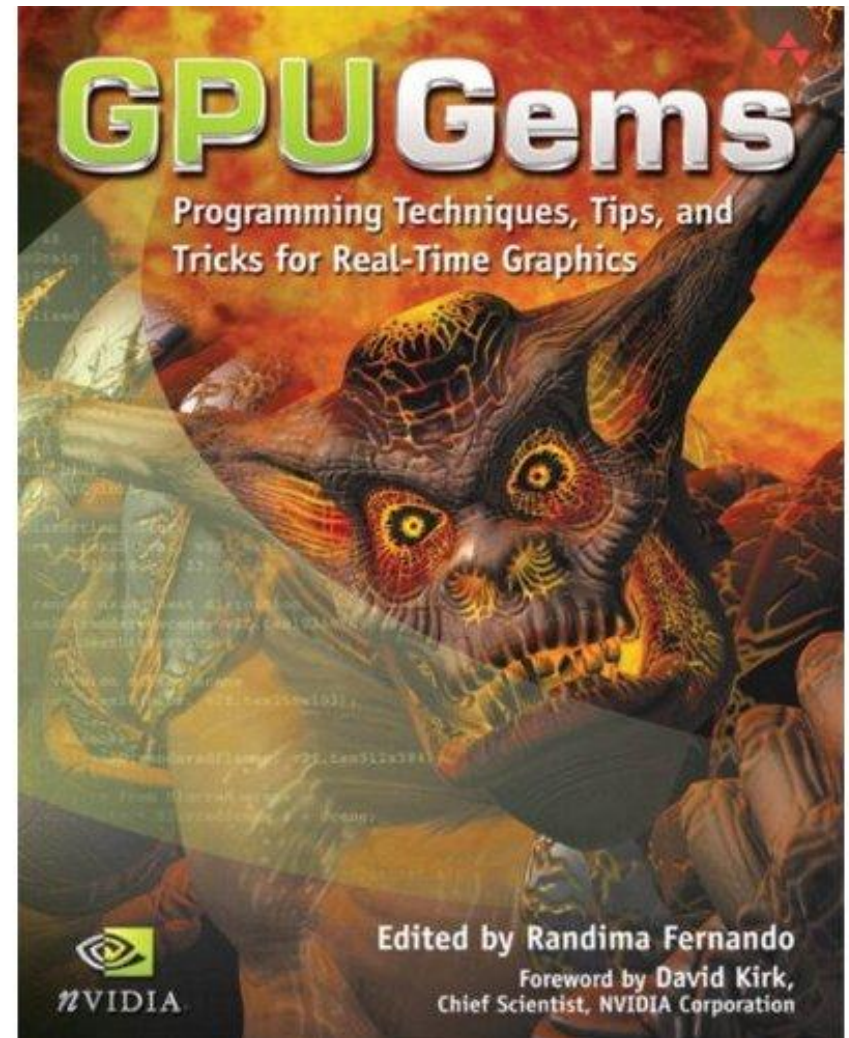
- Some knowledge about GLSL
 - Orange Book



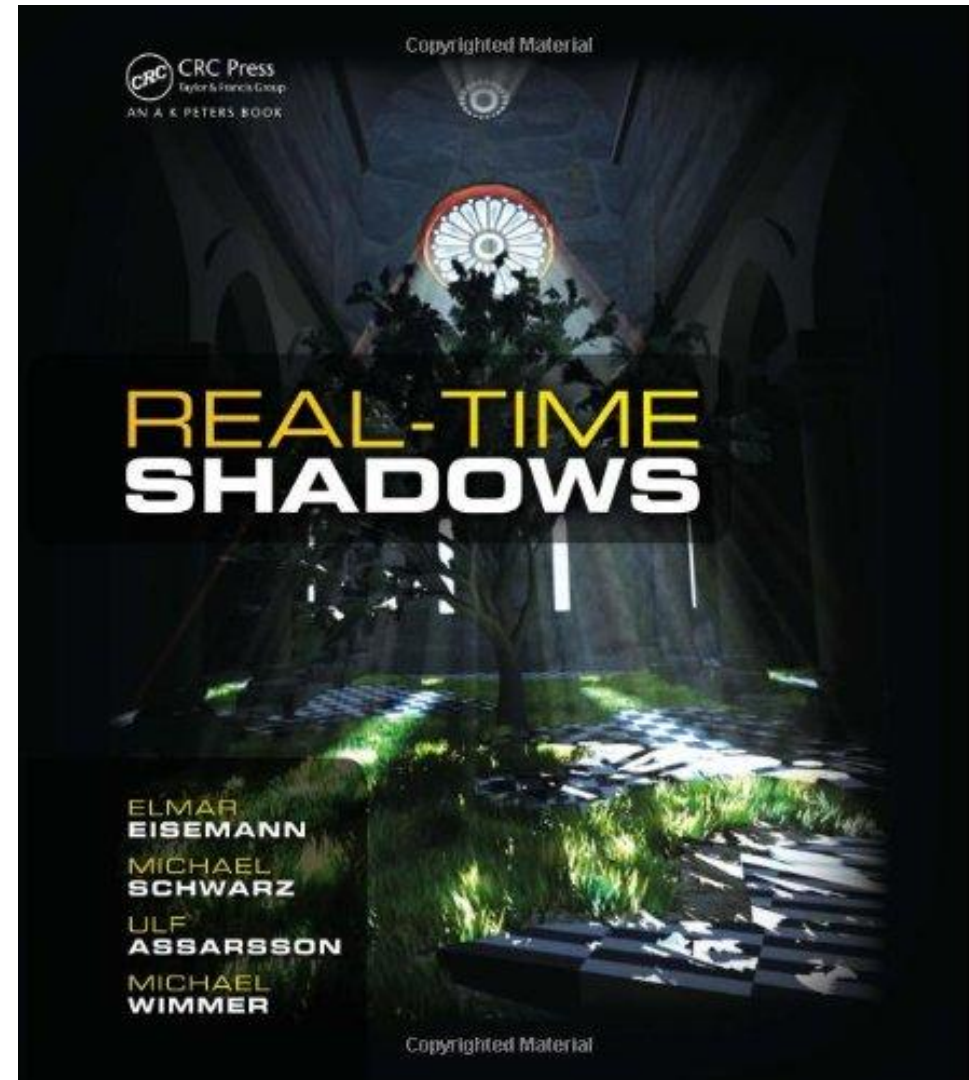
- Real-Time Rendering, Third Edition
 - AK Peters, 2008 (new: third edition)
- Not mandatory
 - But covers all standard methods
- Lecture slides!



- GPU Gems 1, 2, 3
 - Many nice effects
 - Available online
- ShaderX/GPU Pro series



- Real-Time Shadows
 - Shameless plug 😊



What's it all about

Culling
Visibility

Illumination

Shading and Lighting

Reflections

Shadows



- Evolution of graphics hardware
- Perception issues
- Level of detail
- Graphics programming
- Performance techniques
- Shading models
- Terrain rendering



But most importantly..



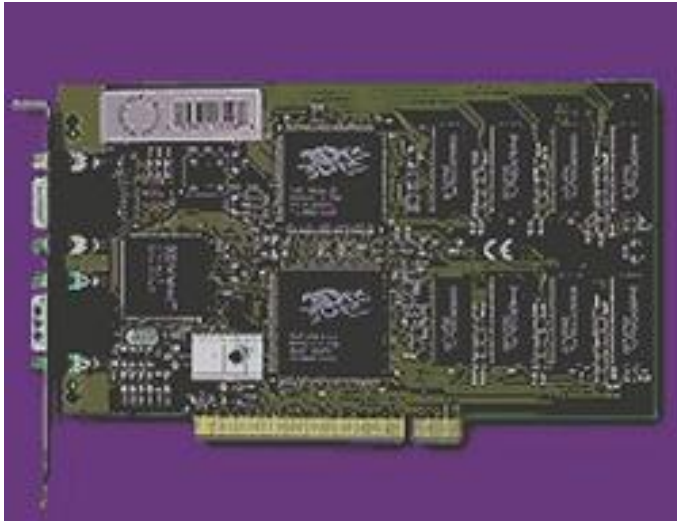
all of this at 60 frames per second!



- Actually, more might be needed
 - CRT refresh rate!
- Explanation: eye sees double images
- LCDs might have different artifacts (softness, ghosting)
- Also...
 - Multiple displays
 - Stereo rendering



Real-Time Rendering Hardware Development



3DFX Voodoo
(1996)



GeForce GTX 980
Radeon/Fury/...



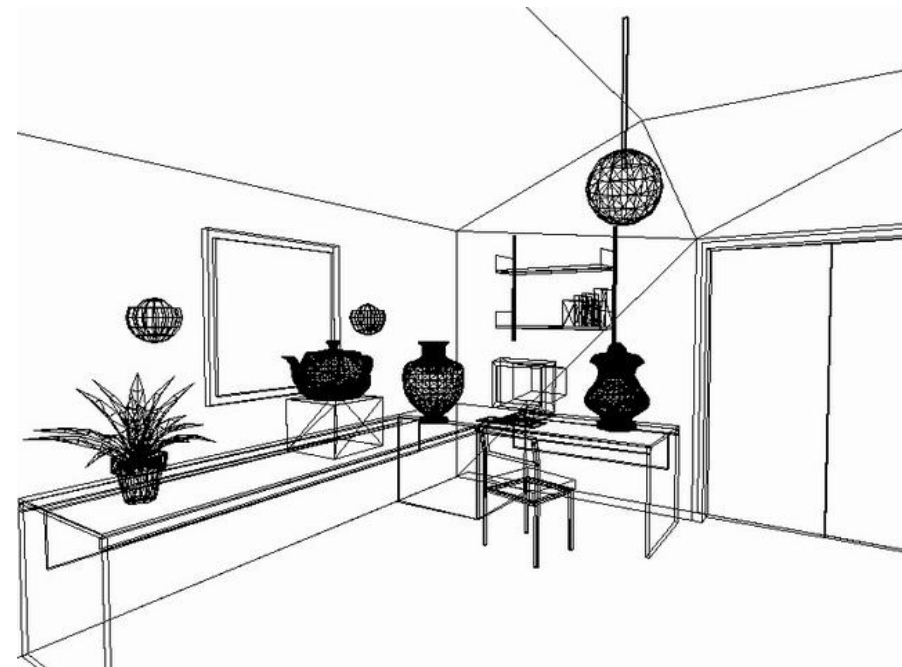
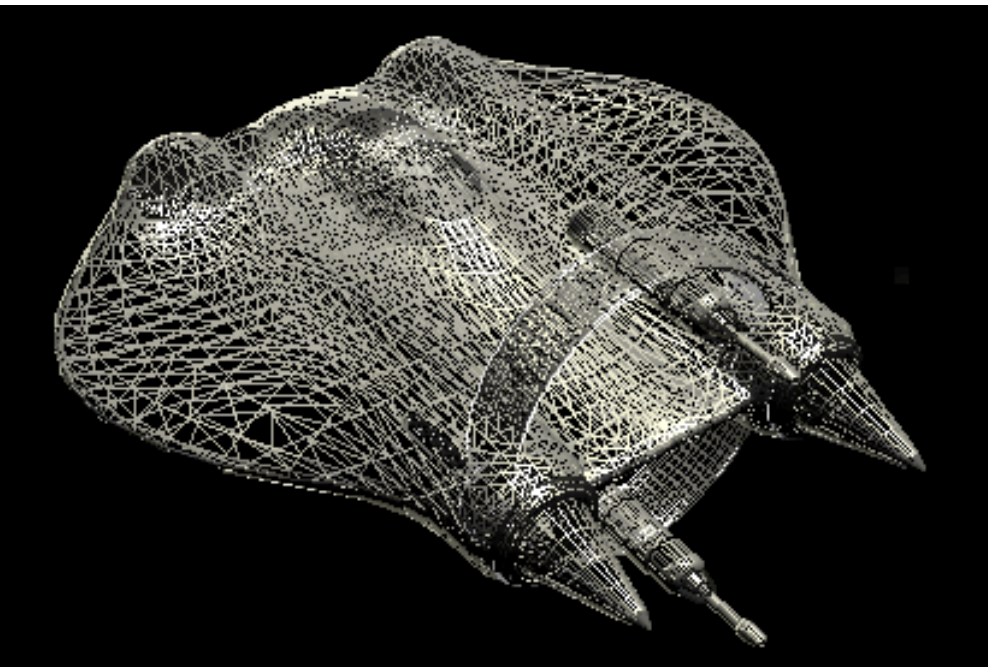
- Performance
 - Triangles / second
 - Pixel fragments / second
 - Shader ops / second
- Features
 - Hidden surface elimination
 - Image (texture) mapping
 - Programmable shading
- Quality
 - Numeric precision (8/10/16 bit, 16/24/32/128 bit FP)
 - Texture filters, antialiasing



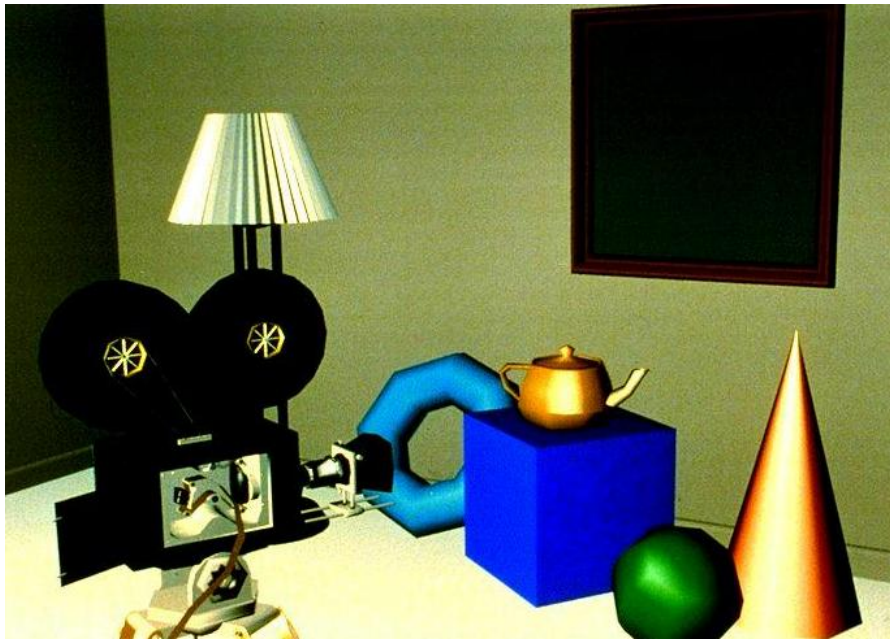
- Some important phases
 - Early research
 - Flight simulation
 - SGI workstations
 - PC
- Hardware generations
 - Different development track for SGI/PC
 - Defined by feature set, but:
 - Any feature can be implemented in hardware
 - Early SGIs: hardware geometry, no texturing
 - Early PCs: hardware texturing, no geometry



- Vertex: transform, clip, project
- Pixel: color interpolation of lines
- Frame buffer: overwrite
- When: prior to 1987



- Vertex: lighting calculations
- Pixel: depth interpolation, triangles
- Frame buffer: depth buffer, color blending
- Dates: 1987-1992



- Vertex: texture coordinate transformation
- Pixel: texture coordinate interpolation
texture evaluation and filtering
- Dates: 1992-2000



- Programmable shading
 - Vertex shading
 - Pixel shading
 - Geometry shading
 - Tessellation
- Heavily used for other calculations (GPGPU)

Date: 2001-2008



- Real-time photorealistic rendering kind of possible...



- But a lot of highly specialized methods/fakes
- Realism mostly due to artist tuning/content creation (100 artists, ~3 years for AAA titles)



- Starts 2009:
 - IBM: **Cell** (already in use in PS3, though not primarily for graphics)
 - Intel: **Larabee** (16 x86 'mini-cores') (-> Xeon Phi)
 - AMD: **Fusion** (CPU+GPU on a chip, low-end)
 - NVIDIA: **CUDA: FERMI, KEPLER, MAXWELL**
- Better surface representations (subdivision surfaces, true displacements, true B-reps)
- Real-time raytracing/pathtracing, radiosity
- Might make many state-of-the-art methods useless!
- **But:** Exciting new research areas!!!



- Paradigm shift to heterogeneous architectures
 - Merging CPU and GPU on one chip
 - Often seen in mobile GPUs
 - GPU is treated as a parallel streaming PU
- High bandwidth interconnect of CPU and GPU
 - CPU and streaming units working together
- New: algorithm decomposition, dynamic data structures, efficient data structure traversal and adaptive refinement...
- **Good-bye to the one way graphics pipeline!**



- Intel Larrabee
 - x86 cores
- NVIDIA Fermi/Kepler/...
 - streaming cores
- Extremely powerful multi-core processors
 - Usually 8-16 cores
 - Optimized for SIMD instructions
 - Synchronized caches for communication
 - C++ programmability

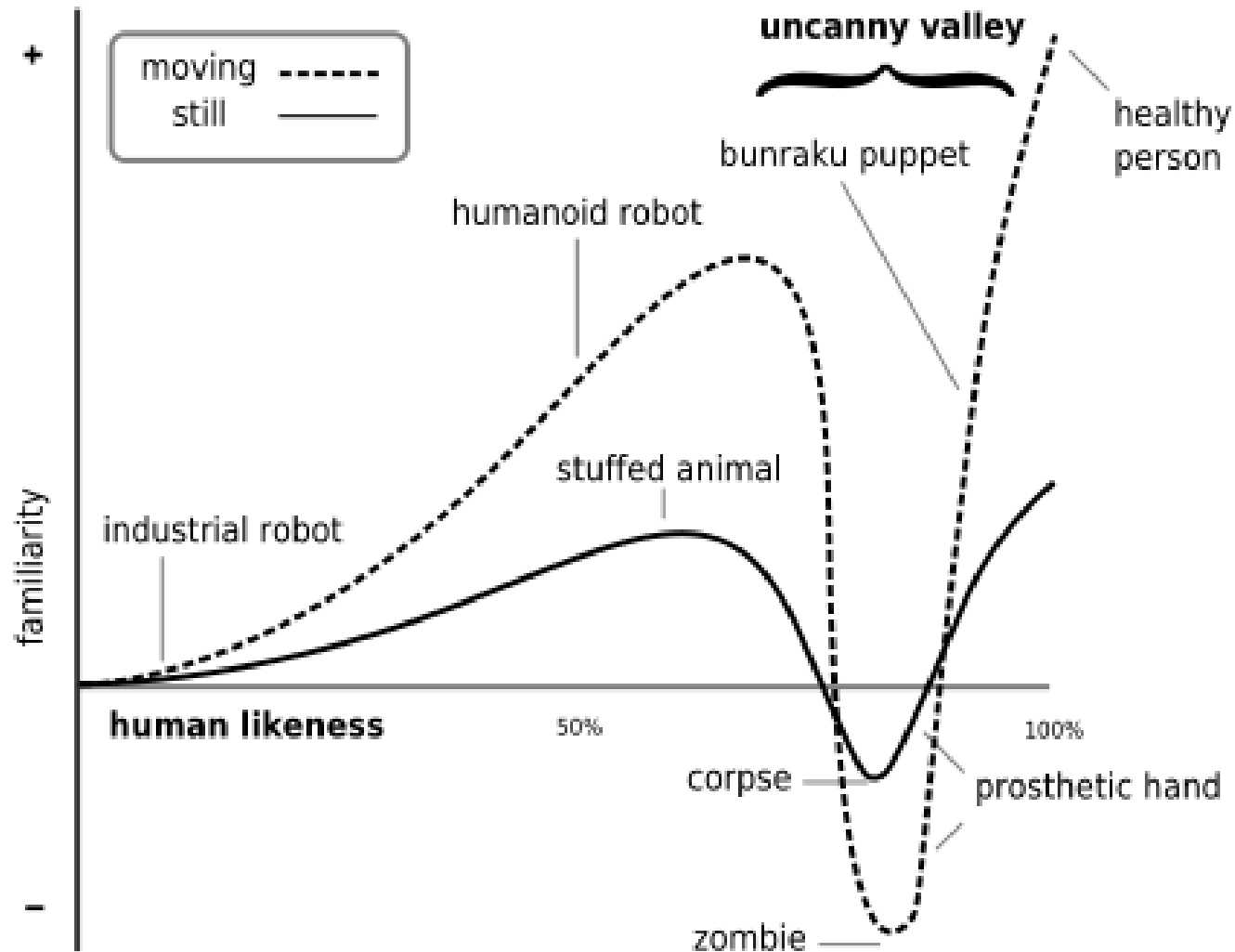


Fermi's 16 SM are positioned around a common L2 cache. Each SM is a vertical rectangular strip that contain an orange portion (scheduler and dispatch), a green portion (execution units), and light blue portions (register file and L1 cache).



- Axes of realism:
 - Rendering
 - Content
 - Animation
 - Behavior
- Have to be in equilibrium!
- Current content and animation gap!
- Hard problem: Requires complex simulations/captures or cannot be formulated as equations
- Especially hard for animation (uncanny valley)





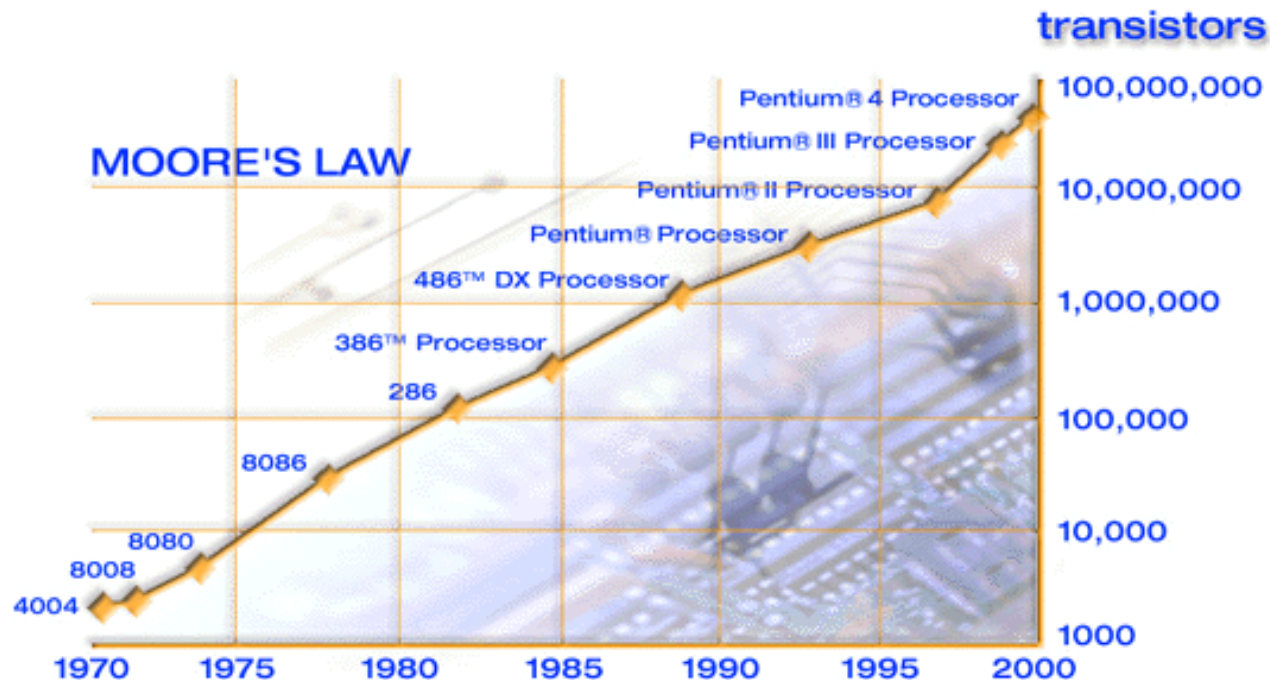
- Started with introduction of **3DFX Voodoo** in late 1996
 - First real 3D card (but no 2D)
 - Bilinearly filtered textures
 - No performance hit for texturing
 - 2x performance advantage for over 1 year!
- 3rd generation, minus all vertex capabilities!
- Let's forget about...
 - Matrox Millenium (no textures), S3 Virge (slower than software), NVidia NV-1 (bad architecture)
- Enter NVidia...



- Up to 1995
 - 2D only (S3, Cirrus Logic, Tseng Labs, Trident)
- 1995 Scanlines (Proprietary APIs)
- 1996 Trapezium rendering (introduction of DX3)
- 1997 Triangle rendering (... DX5)
- 1998 Triangle setup (...DX6)
- 1999 Multi-Pipe, Multitexture (...DX7)
- 2000 Transform and lighting (...DX8)
 - finally caught up to full 3rd generation!
- 2001 Programmable shaders
 - PCs surpass SGI workstations, 4th generation
- 2002 Full floating point
- 2004 Full looping and conditionals (...DX9)
- 2007 Geometry shaders, more flexible programming model (...DX10)



- Gordon Moore, 1965
- Exponential growth in number of transistors
- Doubles every 18 months (holds for CPUs)
 - yearly growth: 1.6
- Not visible in clock speeds anymore
- Trend: multiple cores...



Season	Product	32-bit AA Fill	Yr rate	MPolys	Yr rate
2H97	Riva 128	20M	-	3M	-
1H98	Riva ZX	31M	2.4	3M	1.0
2H98	Riva TNT	50M	2.6	6M	4.0
1H99	TNT 2	75M	2.3	9M	2.3
2H99	GeForce256	120M	2.6	15M	2.8
1H00	GeForce 2 GTS	200M	2.6	25M	2.8
2H00	Geforce 2 Ultra	250M	1.6	31M	1.5
1H01	GeForce 3	416M	2.5	25M	0.6
2H01	GeForce 3 Ti500	500M	1.4	30M	1.4
1H02	GeForce 4	625M	1.6	75M	6.3
1H03	GeForceFX 5800	1041M	1.7	375M	5
2H03	GeForceFX 5900	938M	0.8	338M	0.8
2H04	GeForceFX 6800	~2500M	2.7	600M	1.8
2H05	GF 7800 GTX	~5000M	2	800M	1.4
	(Cost: 500 Euro)	AVG:	2.1	AVG:	2.4



8800 GTX (2006)	GTX 280 (2008)	GTX 980Ti (2015)
120 cores	240 cores	2816 cores/22 SMM
single precision	single/double precision	
0.5 TeraFLOPs	1 TeraFLOPs	5,6 TeraFLOPs
37 GigaTexels/sec	48 GigaTexels/sec	176 GigaTexels/s

- As fast as fastest Supercomputer in 1995
- Becoming slower than Moore's law
- Used in HPC parallel computers (CUDA, Tesla)
 - Molecular dynamics, climate simulations, fluid dynamics
 - ...everything highly parallel computable
- Speedup 10-100x compared to standard processors



- Beware peak numbers! Usually less due to:
 - State changes, pipeline stalls
 - CPU/driver issues
 - Non-optimal geometry arrangement
 - Memory bandwidth (for geometry!)
 - Non-trivial transform/lighting
 - Cache inefficiencies
 - Non-trivial shading/texturing
- But may sometimes be larger:
 - e.g.: z/texture compression can help
 - no antialiasing



- Peak numbers not so relevant nowadays...
- More important considerations:
 - Vertex shader instructions
 - Pixel shader instructions
 - Feature set
 - Quality (antialiasing)
 - Single/Double precision Teraflops

