ME 290R – Topics in Manufacturing, Spring 2009
General Purpose Computation (CAD/CAM/CAE) on GPUs
http://www.me.berkeley.edu/ME290R

Instructor: Prof. Sara McMains
5145 Etcheverry Hall, 642-9359
Office Hours: Th 2-3 pm, F 3-4 pm (tentative)
mcmains@me.berkeley.edu

Lectures:

Tu/Th 11am-12:30pm, 108 Wheeler (or bigger room?)

Description:

Modern programmable Graphics Processing Units (GPUs) dramatically outperform CPUs on a variety of arithmetically intense computations. Furthermore, the performance gap is growing, with GPU performance doubling roughly every 6 months instead of the 18 months it takes CPU performance to double. Unlike earlier graphics cards, today's programmable GPUs allow the user to perform general purpose computations on the GPU, such as dense matrix multiplication or simulations of fluid dynamics. Making maximum use of this power requires understanding the unique programming model of the GPU, and creatively mapping a problem onto the graphics computation steps. This course will cover the fundamentals of the graphics pipeline, how to program the GPU, and the latest research on general-purpose GPU computation, with an emphasis on applications in Computer Aided Design, Computer Aided Manufacturing, and Computer Aided Engineering analysis.

Course Topics:

Computer Graphics Background, GPU programming models and basic techniques :

- Introduction to graphics hardware: capabilities, trends, strengths and weaknesses; graphics pipeline introduction.
- Projective geometry: 2D scales, rotations, and translations using homogeneous coordinates; 3D scales, rotations, and translations.
- Projection: orthographic projection, perspective projection, vanishing points.
- Primitive assembly and rasterization, clipping, antialiasing; visibility, interpolation, shading, and texture mapping.
- Raster operations: scissor, alpha, stencil, and depth testing; blending and logic operations.
- The streaming model; high level programming for GPUs
- Bottleneck identification, debuggers
- Depth Peeling and Projective Textures
- Reductions, Sorting, Searching on the GPU
- CUDA
- OpenCL

Sample applications may include:

- GPU fluid visualization and simulation

- Sparse matrix solvers, conjugate gradient, and multigrid
- Dense linear solvers
- PDE solving
- Image processing
- Boolean operations, constructive solid geometry, CSG evaluation
- Manufacturability analysis
- Distance fields
- Splines
- Ray tracing

Required Texts:

Research papers, to be made available (generally in electronic form).

Reference Texts:

will be on reserve in the Kresge Engineering Library:
Fundamentals of computer graphics / Peter Shirley et al. [2nd ed.] AK Peters, 2005. Engineering T385.S434 2005
The Cg tutorial : the definitive guide to programmable real-time graphics. Fernando, Randima. Addison-Wesley, 2003. Engineering T385.F47 2003
OpenGL shading language. Randi J. Rost, Addison-Wesley, 2006. Engineering T385.R665 2006
also available on-line:
OpenGL programming guide : the official guide to learning OpenGL, version 2 / OpenGL ARB, Dave Shreiner et al. 5th ed. Addison-Wesley, 2006. Engineering T385.O635 2006
http://fly.cc.fer.hr/~unreal/theredbook/
GPU gems : programming techniques, tips, and tricks for real-time graphics / edited by Randima Fernando. Addison-Wesley, 2004. Engineering T385.G688 2004
http://developer.nvidia.com/object/gpu_gems_home.html
GPU gems 2 : programming techniques for high- performance graphics and general-purpose computation / edited by Matt Pharr ; Addison-Wesley, 2005. Engineering T385.G688 2005
http://developer.nvidia.com/object/gpu_gems_2_home.html
GPU gems 3 / edited by Hubert Nguyen. Addison-Wesley, c2008. Engineering T385.G6882 2008
http://developer.nvidia.com/object/gpu-gems-3.html

Requirements:

Short homework assignments involving GPU programming.
Final project, paper & presentation.
Oral presentation of a GPU research paper.

Grading:

50% Project
40% Homework
10% Oral Presentation

Pre-requisites:

Graduate standing in engineering or permission of instructor. Familiarity with C or C++.