General Purpose Computation on the GPU
Spring 2009

Time: 2:30—3:45pm Tu&Thu
Location: LGRT 1322
Instructor: Rui Wang
Grader: TBA
Office: CS 270
Email: ruiwang@cs.umass.edu
Course Page: (twiki)
Course Forum: (phpBB)
Office Hours: 4:00—5:00pm Tu&Thu (or by appointment)

Prerequisites: Familiar with C/C++ Programming and Data Structures

Textbooks: (Recommended, not required)
NVIDIA CUDA Programming Guide (Downloadable from NVIDIA website)
Patterns for Parallel Programming, Addison Wesley, 2005

Course Objectives: Graphics processors (GPUs) on today's commodity video cards have evolved into powerful engines capable of a variety of computations beyond computer graphics. This course takes a detailed look at both basic and advanced topics related to general-purpose computation on graphics hardware (GPGPU). The aim of the course is to provide students with knowledge and hand-on experience in developing applications on modern GPUs using NVIDIA's CUDA programming interface. The first half of the class will focus on introductions to programmable GPUs, OpenGL shading language (GLSL), and CUDA; and the second half will focus on the GPU-based implementation of several key building blocks/applications such as prefix sum (scan), matrix operations, image convolution, k-means clustering, dynamic programming, collision detection, physically-based simulation, and ray tracing.

This course is primarily lecture-based. It will also involve student discussions about important research publications. A final project is required, the goal of which is for every student to take an interesting computational problem and provide an efficient GPU-based solution to it.

Course Workload and Grading:

<table>
<thead>
<tr>
<th>Programming Assignments:</th>
<th>45% (15% each)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Midterm Exam (in-class):</td>
<td>15%</td>
</tr>
<tr>
<td>Final Project:</td>
<td>35%</td>
</tr>
<tr>
<td>Class Participation:</td>
<td>5%</td>
</tr>
</tbody>
</table>

Course Schedule (subject to change):

<table>
<thead>
<tr>
<th>Date</th>
<th>Activity</th>
<th>Due</th>
</tr>
</thead>
<tbody>
<tr>
<td>01/27</td>
<td>Introduction</td>
<td></td>
</tr>
<tr>
<td>01/29</td>
<td>Graphics Hardware I</td>
<td></td>
</tr>
<tr>
<td>02/03</td>
<td>Graphics Hardware II</td>
<td>CUDA Warmup</td>
</tr>
<tr>
<td>02/05</td>
<td>GPU Programming Overview</td>
<td></td>
</tr>
</tbody>
</table>
02/10  CUDA Programming Basics I
02/12  CUDA Programming Basics II
02/17  CUDA Programming Basics III
02/19  CUDA Memory Model I  Assignment 1
02/24  CUDA Memory Model II
02/26  OpenGL Programming Language I
03/03  OpenGL Programming Language II
03/05  OpenGL Programming Language III
03/10  GPGPU Optimizations I
03/12  GPGPU Optimizations II  Assignment 2
03/17  Spring recess
03/19  Spring recess
03/24  Prefix Scan and Applications I
03/26  Prefix Scan and Applications II
03/31  Midterm Exam
04/02  Review
04/07  Image Processing
04/09  Matrix Computation  Assignment 3
04/14  Dynamic Programming
04/16  Project Proposal Presentation  Proposal Presentation
04/21  Monday schedule to be followed
04/23  Collision Detection
04/28  Physically-based Simulation
04/30  Student Presentation I
05/05  Student Presentation II
05/07  Student Presentation III
05/12  Ray Tracing
05/??  Final Project Presentation  Project Presentation

Collaboration Policy:

- For the programming assignments, you may discuss them with your classmates, but you must implement your own solutions.
- The midterm exam is an in-class, closed-book exam.
- The final project is to be completed by each student independently. You may discuss your project with other students, but you may not share code with each other.

Late Policy:

- You should do your best to complete each programming assignment on time. To cope with unforeseen circumstances, you are allowed five late days in total during the entire semester. You may use them at your own discretion. Beyond this, late assignments will not be graded. Note that these are calendar days: weekend and holiday days count equally with weekdays.
- No late day is permitted for the final project.
Resources:

- CUDA SDK
- CUDA Documents
- CUDA Zone
- CUDA Forum
- GPGPU.org