Using a Commodity GPU in an Undergraduate Parallel Computing Course

Joshua Steinhurst
St. Mary's College of Maryland
jsteinhurst@smcm.edu

Thorsten Scheuermann
Advanced Micro Devices, Inc.
thorsten.scheuermann@amd.com

Abstract

A modern graphics processor unit (GPU), contains dozens of high performance floating point processors. In raw computational power, they far outstrip current CPUs. GPUs have become a popular parallel platform among researchers. After graduation, our students are more likely to have a GPU available than a traditional parallel machine, yet they rarely gain experience with SIMD machines. We have developed a short module suitable for an undergraduate parallel computing course with very limited prerequisites.

What is a GPU?

- Programming Model
  - ❖Single Instruction Multiple Data (SIMD)
  - ❖Streaming computations
- Extremely limited communication
- Can buy >1TFLOPS for $3,000
  - ❖≥48 floating point processors per chip
  - ❖A commodity market driven by gamers

Not just for video games!

- Financial modeling
- Seismic simulations
- Video compression
- Physics calculations

Course Context

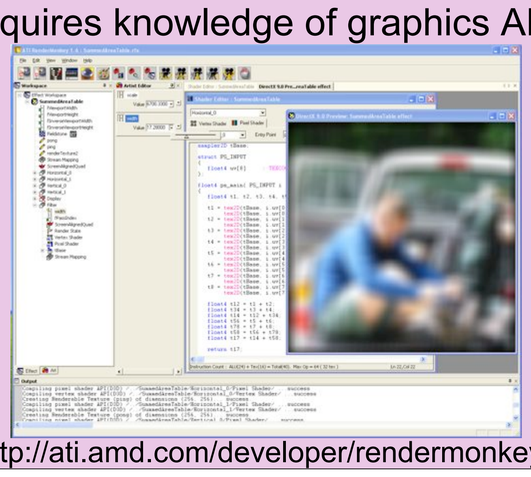
- Public liberal arts college
  - ❖Shallow prerequisite tree
    - ▪Only CS 1/2 (Java with data structures)
  - ❖Limited lab resources
- Distributed and Parallel Computing
  - ❖Cover both topics in one semester
  - ❖Both theory and practice
  - ❖Java RMI
    - ▪Distributed objects
    - ▪Synchronization
  - ❖Java Threads
    - ▪Shared memory architecture
    - ▪Multi-core workstations

Module Outline

- 2 Weeks including 4 meetings
- Introductory lecture on GPU (110 minutes)
  - ❖Basic graphics terminology and pipeline
  - ❖Review of SIMD programming model
  - ❖Brief introduction to language (HLSL)
- Three labs (55 minutes)
  - ❖Interactively develop algorithm as class
  - ❖In pairs:
    1. Write code in a provided framework
    2. Gather performance results
    3. Interpret the data focusing on differentiating computation and communication bottlenecks

Development Environment

- ATI RenderMonkey
  - ❖IDE for development of real-time shaders
    - ▪Immediate feedback on code changes
  - ❖Requires knowledge of graphics API

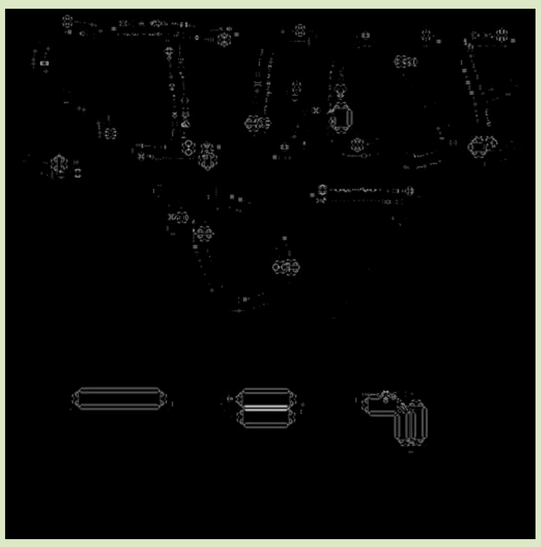


http://ati.amd.com/developer/rendermonkey/

Results

- Student outcomes
  - ❖Experience programming SIMD machine
  - ❖Excitement of using cutting edge system
  - ❖Experiments where outcome is unclear
    - ▪Performance model is complex and involves several interrelated factors
- Reusable module developed

Game of Life



```
float4 ps_main(float2 texCoord:TEXCOORD0):COLOR {
  float numNeighbors = GetNeighborCount(texCoord);
  float myCell = tex2D(Texture0, texCoord);

  if(!bPauseSimulation ) {
    if(myCell>0.0) {
      if(numNeighbors>=2.0 && numNeighbors<=3.0){
        myCell = 1.0;
       } else {
        myCell = 0.0;
       }
    } else {
       if(numNeighbors==3.0) {
        myCell = 1.0;
       }
    }
  }
  return myCell;
}
```

Interactive Control
2048^2 Cells at 120 iterations/sec
ATI Radeon X1900XTX 512MB

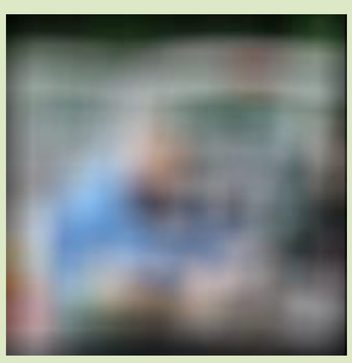
2D Heat Distribution


```
float GetAvgNeighborTemp(float2 texCoord) {
  [...]
  sum+=tex2D(Texture0, texCoord+float2(-dx, -dy));
  sum+=tex2D(Texture0, texCoord+float2(  0, -dy));
  sum+=tex2D(Texture0, texCoord+float2( dx, -dy));
  sum+=tex2D(Texture0, texCoord+float2(-dx,   0));
  [...]
  return sum/8.0;
}
float4 ps_main(float2 texCoord:TEXCOORD0):COLOR {
  if( !bPauseSimulation ) {
    myCell = GetAvgNeighborTemp(texCoord);
  }
  return myCell;
}
```

Interactive Control
1024^2 Cells at 216 iterations/sec
ATI Radeon X1900XTX 512MB

Large Kernel Filtering with Summed Area Table

Hensley et al. SIGGRAPH Sketch 2005



Future Work

- Non-graphical development environment
  - ❖Remove awkward syntax (i.e. TEXCOORD)
  - ❖Leverage emerging tools such as
    - ▪AMD's Close To the Metal
    - ▪NVIDIA's CUDA
- Adapt the module for
  - ❖Different institutions
    - ▪Stricter prerequisites
  - ❖Different course structures
    - ▪Dedicated parallel course
  - ❖Tighter integration with the entire course
- Develop measurable outcomes
  - ❖What learning is taking place?
  - ❖How does it compare to alternatives?
- Develop detailed labs for distribution

Interested in applying this at your school?
We are happy to provide assistance!

Funding and Support