MATLAB Acceleration for Image Processing using CUDA-Enabled GPUs

March 2009

John Melonakos
AccelerEyes
john.melonakos@accelereyes.com

Sumit Gupta
NVIDIA Tesla GPU Computing
sumitg@nvidia.com
What is GPU Computing?

Computing with CPU + GPU

Heterogeneous Computing
Computation Discontinuity

Gflops (log scale)

- NVIDIA GPU
- Intel CPU

- Tesla 10-series
- Tesla 8-series
- Intel Xeon Quad-core 3 GHz
- Intel Core2 Dual-core 3.0 GHz
- Intel Pentium 4 Dual-core 3.0 GHz
- Intel Pentium 4 3.2 GHz

Double Precision debut
Medical Imaging
U of Utah

Molecular Dynamics
U of Illinois, Urbana

Video Transcoding
Elemental Tech

Matlab Computing
AccelerEyes

Astrophysics
RIKEN

50x – 150x

Financial simulation
Oxford

Linear Algebra
Universidad Jaime

3D Ultrasound
Techniscan

Quantum Chemistry
U of Illinois, Urbana

Gene Sequencing
U of Maryland
CUDA Parallel Computing Architecture

- Parallel computing architecture and programming model
- Includes a C compiler plus support for OpenCL and DX11 Compute
- Architectured to natively support all computational interfaces (standard languages and APIs)
NVIDIA Tesla 10-Series GPU

Massively parallel, many core architecture

240 Processor Cores

1 Teraflops - 1,000 times Cray X-MP

IEEE Compliant Double Precision Floating Point

Designed for Scientific Computing
CUDA Facts

- 900+ Research Papers
- 115+ universities teaching CUDA

www.NVIDIA.com/CUDA

- 200+ papers and applications
- 110 Million CUDA-Enabled GPUs
- 60,000+ Active Developers
Background

• Who is AccelerEyes?
  – AccelerEyes is a MathWorks partner
  – Simple software for visual computing
Background

• Who is AccelerEyes?
  – AccelerEyes is a MathWorks partner
  – Simple software for visual computing

• What is Jacket?
  – GPU engine for MATLAB
  – CUDA powered language extension
Background

• Who is AccelerEyes?
  – AccelerEyes is a MathWorks partner
  – Simple software for visual computing
• What is Jacket?
  – GPU engine for MATLAB
  – CUDA powered language extension
• Why Jacket?
  – Challenges in technical computing
  – Low-cost speed, high-value graphics
  – Increased productivity
MATLAB Options

- **CPU Solutions** (blue arrows)
  - MATLAB and the Parallel Computing toolbox enable PC and clustered MATLAB computing

- **GPU Solutions** (green arrows)
  - Jacket enables CUDA MATLAB computing
Jacket Benefits

Jacket combines the speed of CUDA and the graphics of the GPU with the user friendliness of MATLAB.
Functionality

- Generators: `geye`, `gones`, `gzeros`
- Element-wise: `+`, `*`, `-`, `/`
- Reductions: `sum`, `min`, `max` ...
- Indexing: subscripted referencing / subscripted assignment
- Linear Algebra: matrix multiply, ...
- FFT: `fft`, `ifft`, `fftn`, `ifftn`
- Filtering: `filter`, `filter2`, `convn`
- Interpolation: `interp2`
- Parallel for-loops: `gfor`
Kernel Benchmarks

SGEMM Kernel

Matrix Size

- 2048x2048: 48 secs (Tesla C1060) vs 2637 secs (Intel Core 2 Duo (2.0 GHz)), 54x Speedup
- 1024x1024: 7.1 secs (Tesla C1060) vs 117 secs (Intel Core 2 Duo (2.0 GHz))
- 512x512: 1.3 secs (Tesla C1060) vs 12.7 secs (Intel Core 2 Duo (2.0 GHz))

FFT Kernel

Vector Length

- 1024x1024: 1.1 secs (Tesla C1060) vs 17.7 secs (Intel Core 2 Duo (2.0 GHz)), 16x Speedup
- 512x512: 0.43 secs (Tesla C1060) vs 3.6 secs (Intel Core 2 Duo (2.0 GHz))
- 256x256: 0.28 secs (Tesla C1060) vs 0.91 secs (Intel Core 2 Duo (2.0 GHz))

Application Benchmarks

**Canny Edge Detection**
- **Image Size:** 3936x3936
  - **Time:** 3.8 minutes
  - **Speedup:** 114x
  - **Time (log scale):** 7.2 hours
- **Image Size:** 2048x2048
  - **Time:** 8 seconds
  - **Time (log scale):** 1.86 hours
- **Image Size:** 1024x1024
  - **Time:** 2 seconds
  - **Time (log scale):** 28.4 minutes

**Black-Scholes Simulation**
- **Vector Length:** 3.84
  - **Time:** 0.13 seconds
  - **Speedup:** 143x
  - **Time (log scale):** 18.1 seconds
- **Vector Length:** 2.24
  - **Time:** 0.12 seconds
  - **Time (log scale):** 10.6 seconds
- **Vector Length:** 1.04
  - **Time:** 0.11 seconds
  - **Time (log scale):** 5.0 seconds

*Note: Tesla 8 Series GPU vs. Intel Core 2 Duo (2.4 GHz) for Canny Edge Detection; Tesla C1060 vs. Intel Core 2 Duo (2.0 GHz) for Black-Scholes Simulation.*
function [u v] = optflow_HS(I1, I2, alphasq, iter)

    u = I1;
    v = I2;

    [Ix Iy It] = diffs(I1, I2);

    for i=1:iter
        u_ = lapAverage(u);
        v_ = lapAverage(v);

        num = Ix .* u_ + Iy .* v_ + It;
        den = alphasq + Ix.^2 + Iy.^2;

        u = u_ - Ix .* num./den;
        v = v_ - Iy .* num./den;
    end

end

>> [u v] = optflow_HS(gsingle(image1), gsingle(image2), 0.1, 100);
Optical Flow (Horn & Schunck)

Speedup: 12X on 128x256
Image Thresholding

CPU

GPU

Speedup: 20X on 512x512

Image Smoothing

Speedup: 12X on 915x915

```
1  Igpu = gsingle(Iin);  % Moon image
2  X = ones(3) / 9;     % Smoothing kernel
3  for i = 1:nIter,
4      Igpu = filter2(X, Igpu, 'full');
5  end;
```
Image Interpolation

```
1 - A = gzeros( nn );
2 - for ii = 1:100
3 -    Z_ = interp2( A_ );
4 -    gforce(Z_);
5 - end
```

Speedup: 200X on 256x256
Image Morphing

Speedup: 40X on 512x512
Custom CUDA Functions
Integration using MEX

#include <cuda.h>
#include "mex.h"

// each element gets its index */
static __global__ void kernel(float *d_out, float *d_in)
{
    int x = blockIdx.x * blockDim.x + threadIdx.x;
    d_out[x] = d_in[x] + x;
}

void mexFunction(int nlhs, mxArray *plhs[], int nrhs, const mxArray *prhs[])
{
    /* attach to context */
    C2context *ctx = (C2context *) (unsigned int) mxGetScalar(prhs[0]);
    cuCtxAttach(ctx, 0);

    /* get device pointer of gsingle */
    float *d_in = (float *) (unsigned int) mxGetScalar(prhs[1]);

    /* run kernel to initialize 10 elements */
    float *d_out;
    cudaMalloc((void **)&d_out, 10*sizeof(float));
    kernel<<<5,2>>>(d_out, d_in);

    /* pull back to CPU and print */
    float h_out[10];
    cudaMemcpy(h_out, d_out, 10*sizeof(float), cudaMemcpyDeviceToHost);
    for (int i = 0; i < 10; i++)
        printf("%f", h_out[i]);

    /* detach from context */
    cuCtxDetach(ctx);

    /* return device pointer */
    plhs[0] = mxCreateDoubleScalar((unsigned int)d_out);
}

---(Unix)--- mex.cu  Bot (53,0)  (C/)---(mexFunction)----17:54 [Wed]---
Graphics Toolbox

Jacket includes the Graphics Toolbox

- True visual computing
- OpenGL API in MATLAB
- Interactive OpenGL
- Key functions: `gsurf`, `gimage`, `gscatter3`, `gplot`, ...
- Visualization scripts are open and modifiable.
Some Jacket Customers

Roadmap for New Features

– more gfor
– gdouble
– multi-GPU support (for clusters of GPUs)
– LAPACK (eig, inv, etc.)
– signal processing
– image processing (and computer vision)
– Simulink® on the GPU
– statistical functions
– handle graphics
– lots of other MATLAB functions (finance, biology, etc.)
Tesla GPU Computing Products
Built for High Performance Computing
### Tesla GPU Computing Products

<table>
<thead>
<tr>
<th></th>
<th>Tesla S1070 1U System</th>
<th>Tesla C1060 Computing Board</th>
<th>Tesla Personal Supercomputer (4 Tesla C1060s)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>GPUs</strong></td>
<td>4 Tesla GPUs</td>
<td>1 Tesla GPU</td>
<td>4 Tesla GPUs</td>
</tr>
<tr>
<td><strong>Single Precision Perf</strong></td>
<td>4.14 Teraflops</td>
<td>933 Gigaflops</td>
<td>3.7 Teraflops</td>
</tr>
<tr>
<td><strong>Double Precision Perf</strong></td>
<td>346 Gigaflops</td>
<td>78 Gigaflops</td>
<td>312 Gigaflops</td>
</tr>
<tr>
<td><strong>Memory</strong></td>
<td>4 GB / GPU</td>
<td>4 GB</td>
<td>4 GB / GPU</td>
</tr>
</tbody>
</table>
**Tesla Personal Supercomputer: Cluster Perf**

**Supercomputing Performance**
- 960 cores. 4 TeraFlops
- Performance of a 64-node CPU cluster

**Personal**
- One researcher, one supercomputer
- Plugs into standard power strip

**Accessible**
- Program in C for Windows, Linux
Tesla S1070: Supercharge your cluster

PCI-e Gen2 Host Interface Cards

PCle Gen2 Cables (0.5m length)

Host Server

Tesla S1070

- Hess
- Chevron
- Petrobras
- NCSA
- CEA
- Tokyo Tech
- JFCOM
- SAIC
- Federal
- Motorola
- Kodak

- BNP Paribas
- University of Heidelberg
- University of Illinois
- University of North Carolina
- Max Planck Institute
- Rice University
- University of Maryland
- Eotvas University
- University of Wuppertal
- Chinese Academy of Sciences
- National Taiwan University
$5 Million Cluster: Lower Power, Higher Perf

CPU 1U Server

2 Quad-core Xeon CPUs: 8 cores

- 0.17 Teraflop (single)
- 0.08 Teraflop (double)

1819 CPU servers

- 310 Teraflops (single)
- 155 Teraflops (double)

Total area 16K sq feet

Total 1273 KW

CPU 1U Server

8 CPU Cores + 4 GPUs = 968 cores

- 4.14 Teraflops (single)
- 0.346 Teraflop (double)

1961 Teraflops (single)

196 Teraflops (double)

Total area 9K sq feet

Total 682 KW

Tesla 1U System

- 4.14 Teraflops (single)
- 0.346 Teraflop (double)

455 CPU servers

455 Tesla systems

- 1961 Teraflops (single)
- 196 Teraflops (double)

Total area 9K sq feet

50% fewer systems

6x more perf

40% smaller

½ the power

40% smaller
½ the power
<table>
<thead>
<tr>
<th>Life Sciences &amp; Medical Equipment</th>
<th>Productivity / Misc</th>
<th>Oil and Gas</th>
<th>EDA</th>
<th>Finance</th>
<th>CAE / Mathematical</th>
<th>Communication</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max Planck</td>
<td>GE Healthcare</td>
<td>Hess</td>
<td>Synopsys</td>
<td>Symcor</td>
<td>AccelerEyes</td>
<td>Nokia</td>
</tr>
<tr>
<td>FDA</td>
<td>Siemens</td>
<td>TOTAL</td>
<td>Nascentric</td>
<td>Level 3</td>
<td>MathWorks</td>
<td>RIM</td>
</tr>
<tr>
<td>Medtronic AGC</td>
<td>TechniScan</td>
<td>WRF Weather Modeling</td>
<td>TOTAL</td>
<td>SciComp</td>
<td>Wolfram</td>
<td>Philips</td>
</tr>
<tr>
<td>Evolved machines</td>
<td>Boston Scientific</td>
<td>OptiTex</td>
<td>CGG/Veritas</td>
<td>Hanweck</td>
<td>National</td>
<td>Samsung</td>
</tr>
<tr>
<td>Robarts Research</td>
<td>Eli Lilly</td>
<td>Tech-X</td>
<td>Chevron</td>
<td>Quant</td>
<td>Instruments</td>
<td>LG</td>
</tr>
<tr>
<td>Medtronic</td>
<td>Silicon Informatics</td>
<td>Elemental</td>
<td>Headwave</td>
<td>Catalyst</td>
<td>Ansys</td>
<td>Sony</td>
</tr>
<tr>
<td>AGC</td>
<td>Stockholm Research</td>
<td>Technologies</td>
<td>Acceleware</td>
<td>RogueWave</td>
<td>Access Analytics</td>
<td>Ericsson</td>
</tr>
<tr>
<td>Smith-Waterman DNA sequencing</td>
<td>Harvard</td>
<td>Dimensional Imaging</td>
<td>Seismic City</td>
<td>BNP Paribas</td>
<td>Tech-X</td>
<td>NTT DoCoMo</td>
</tr>
<tr>
<td>AutoDock</td>
<td>Delaware</td>
<td>Manifold</td>
<td>P-Wave</td>
<td>SORI</td>
<td>RIKEN</td>
<td>Mitsubishi</td>
</tr>
<tr>
<td>NAMD/VMD</td>
<td>Pittsburg</td>
<td>Digisens</td>
<td>Seismic Imaging</td>
<td>SOFA</td>
<td>SOFA</td>
<td>Hitachi</td>
</tr>
<tr>
<td>Folding@Home</td>
<td>ETH Zurich</td>
<td>General Mills</td>
<td>Mercury Imaging</td>
<td>Renault</td>
<td>SOFA</td>
<td>Radio</td>
</tr>
<tr>
<td>Howard Hughes Medical</td>
<td>Institute Atomic Physics</td>
<td>Rapidmind</td>
<td>Computer ffA</td>
<td>Boeing</td>
<td>Access Analytics</td>
<td>Research Laboratory</td>
</tr>
<tr>
<td>CRIBI Genomics</td>
<td>Rhythm &amp; Hues</td>
<td>Rhythm &amp; Hues</td>
<td>Elcomsoft</td>
<td>xNormal</td>
<td>Tech-X</td>
<td>US Air Force</td>
</tr>
<tr>
<td></td>
<td>xNormal</td>
<td>Elcomsoft</td>
<td>LINZIK</td>
<td>Access Analytics</td>
<td>RIKEN</td>
<td>NTT DoCoMo</td>
</tr>
</tbody>
</table>
|                                   | Elcomsoft           | LINZIK
More Information

- **Tesla main page**
  - [http://www.nvidia.com/tesla](http://www.nvidia.com/tesla)

- **Vertical Solutions**

- **CUDA Zone**
  - CUDA Tutorials, Applications

- **Hear from Developers**
  - [http://www.youtube.com/nvidiatesla](http://www.youtube.com/nvidiatesla)

---

Download Jacket Now

- [http://www.accelereyes.com](http://www.accelereyes.com)

Further Jacket Questions

- [http://www.accelereyes.com/forums](http://www.accelereyes.com/forums)
- [http://www.accelereyes.com/blog](http://www.accelereyes.com/blog)

---

Sumit Gupta
sumitg@nvidia.com

John Melonakos
john.melonakos@accelereyes.com