ECE 6397: Parallel Algorithms for GPUs
David Mayerich, Assistant Professor
Office: Engineering Building 2, W330
Email: mayerich@uh.edu

UNIVERSITY of HOUSTON
CULLEN COLLEGE of ENGINEERING

Programming Assignment 2: Matrix Multiplication

Your second programming assignment is the implementation of a general matrix multiplication (gemm) subroutine. Matrices are available on the class website (http://stim.ee.uh.edu/education/ece-6397-gpu-programming/). The matrix file contains two (2) 32-bit unsigned integers representing the matrix size, followed by the matrix data as 32-bit floating point values in column-major order.

Your application will load two matrices and calculate their product:

$$\mathbf{C} = \mathbf{AB}$$

The matrices will be specified as files from the command line:

```
>> multiply a.mtx b.mtx c.mtx
```

Each argument is a file associated with the matrix multiplication. The first two arguments specify the matrices to be multiplied. The third argument is the file name where the result will be output.

Your program will have the following functionality:

- accept arbitrary matrix sizes (you can assume that both matrices will fit in global memory)
- implement the matrix multiplication using shared memory
  - profile this implementation and output the estimated GFLOPS
- implement the matrix multiplication using the cuBLAS library
  - profile this implementation
  - compare your output to the result of the cuBLAS gemm function (use MSE)

Recommendations

1. As always, I recommend starting with a CPU implementation. Use the lecture notes as a template.

2. Perform the multiplication using cuBLAS and verify that you have the correct result. In order to test if your results match cuBLAS, use the mean squared error (MSE):

$$MSE = \sum_{i=1}^{N} (g[i] - x[i])^2$$

   where the array x is an array containing every element in the matrix. Basically, you'll loop through the final matrices and calculate the difference between each element and square the result. This value should be very close to zero.

3. Implement the multiplication in global memory, and verify the result.
4. Complete your shared memory implementation

ECE 6397 – GPU Programming
Programming Assignment 2: General Matrix Multiplication
Rubric

Name_____

Correctly reading input and output are crucial for this assignment. Assignments that don't correctly read input or produce images won't be graded. Do that first.

cuBLAS implementation **_____ / 20**

GPU implementation **_____ / 60**

correct result	_____ / 10
code is correct/robust	_____ / 20
shared memory is used	_____ / 20
code is efficient (reasonable comparison to gemm)	_____ / 10

comments (I expect about 1-2 lines per line of code) **_____ / 10**
(also note that comments help me understand your thought process, making it easier to give better scores on implementation and efficiency)

profiling (timing results) **_____ / 10**