CS5234 Advanced Computer Graphics Spring 2013
Homework 2: GPU Memory

1 Due Date

Homework 2 is due on Friday February 15th, 2013, 11:59pm.

2 Introduction

This homework is designed to help you understand the memory architecture of the GPUs. You need to learn how to use different types of the memory, and how to boost the performance by using the right type of GPU memory.

3 Project requirement

Please apply CUDA programming framework on an image processing application: image filtering with convolution. Image convolution is a commonly used image operation for image processing techniques, such as image smoothing, sharpening, and edge detection. An introduction of image convolution can be found in this web-link. You should also look into this example for a better description.

Please use the following image sharpening convolution kernel in your implementation.

$$\frac{1}{6}\begin{bmatrix} -1 & -4 & -1 \\ -4 & 26 & -4 \\ -1 & -4 & -1 \end{bmatrix} \tag{1}$$

Three BMP images file are provided in this location for testing your implementation. They have different sizes, 512x512, 1024x1024, and 2048x2048. And they are all store as 8-bit gray scale images. You need to load the image file, process it with your GPU-based convolution, and store it on the disk in BMP format. For accessing BMP file format, you can refer to this link.

You need to implement the following three parallel memory access strategies.

1. Load the image into global memory. Each thread processes one pixel. Threads are organized into blocks. Try three different block size 8x8, 16x16 and 32x32. Please find out which block size give you the best performance.

2. Load the image into texture memory (use 2D texture). Each thread processes one pixel. Try three different block size 8x8, 16x16 and 32x32. Please find out which block size give you the best performance.

3. Load the image into texture memory (use 2D texture). Each thread processes one pixel. In each thread block, all the image data required for the block is loaded into shared memory first. Then, each thread processes the data in the shared memory and store the result in the shared memory. Finally, the result is written into global memory.

Please also implement a pure CPU version and report the performance speedup for your GPU implementation versus CPU one. The comparison is on the kernel execution time.

4 What to Submit

Put your solution in one or more source files. The main file (which includes function main()) should be named homework2.cpp or homework2.cu. Include all your source files and linux makefile in a zip file and upload to class Scholar site in your own dropbox. Please also include a text file to report the kernel execution time for the three GPU implementations and the CPU implementation. In the text file, please also indicate the performance speedup for your GPU implementations against CPU implementation.

Note: Please use CUDA 5.0 for all the homeworks for this course.