LAB 5

DUE DATE: Fri Oct 27 5pm (upload to polylearn) Individual Project

Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Write a program in CUDA to Change an Image from Color to Gray

I’m providing a file in C that does :read a color image, change the image to gray and write the color image

Your job is to write the cuda kernel that change the image to gray, the code for the cuda allocation contains //TODO that you will need to complete

Since we are using a library to read/write images called OpenCV to be able to use you need to perform the following steps:

1. I’m giving you for this program a file that reads a color image , change it into gray and writes back to image file in C

Download the file and place it in your local directory in the MPAC lab

ImagetoGray.cu

2. To be able to use this program you need to install OpenCV library

2.a Download and execute the following bash file: build\_opencv.bash

2.b Open a terminal and Run the bash file: ./build\_opencv.bash

2.c Type the following command to add the library path to bashrc (CHANGE the yellow directory name to yours dont use my directory it will not work)

export LD\_LIBRARY\_PATH=$LD\_LIBRARY\_PATH:/home/mpanto01/opencv\_build/lib

2. d Test that the library is correctly installed by compiling:

/usr/local/cuda-8.0/bin/nvcc -I/home/mpanto01/opencv\_build/include -I/home/mpanto01/opencv\_build/include /opencv

-L//home/mpanto01/opencv\_build/lib

-g -o output OpenCVCUDA.cu -lopencv\_core -lopencv\_imgproc

-lopencv\_highgui -lopencv\_imgcodecs

Change the local directory name to yours

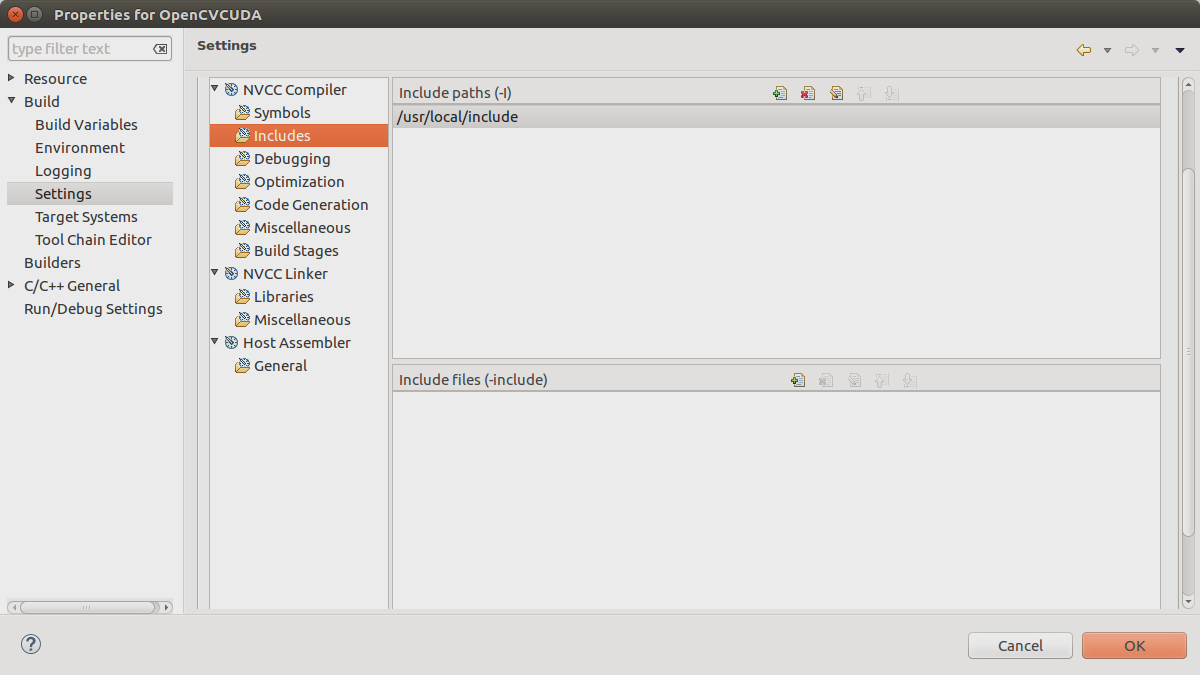
Change the loutput file name to one of your liking

Change the file name you want to compiler to yours

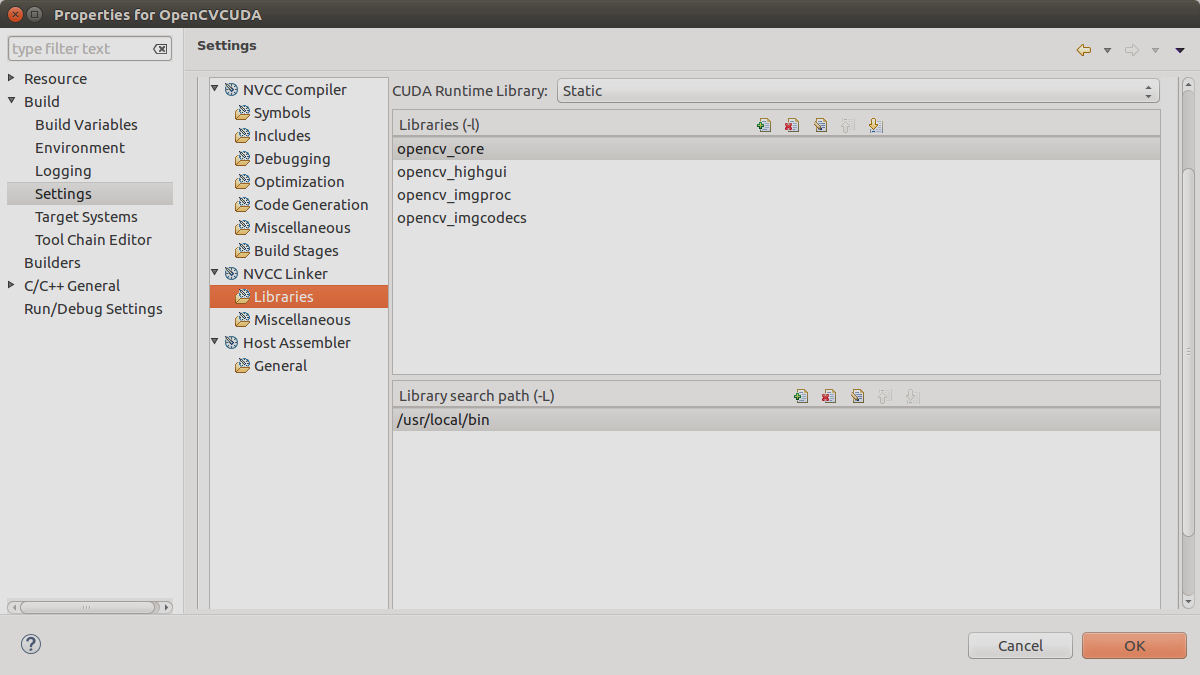
If you are like using Nsight IDE you will need to add to project settings the following:

Project→properties>build > settings>compile settings

1. Compiler: Include files



2. Linker: Lib path



3. Modify the code so it will change the image to gray on the GPU. The code has the //TODO statements written in the code for you to fill up

4. Profile the code and compare times in CPU and GPU

GPU time:

CPU time:

5. Answer the questions from book (2nd edition):

4.2

4.3

4.4

4.6

Appendix : Add the GPU Kernel code and the kernel call (example of what a kernel call is : matmul<<<blocks, threads>>(….) ) from main()

Add the original and gray image.

Note :if you think the image maybe inappropriate it most probably is so pick a different one

Result should look like

