Thrust documentation

Thrust is a C++ parallel programming library which resembles the C++ Standard Library. Thrust's high-level interface greatly enhances programmer productivity while enabling performance portability between GPUs and multicore CPUs. Interoperability with established technologies (such as CUDA, TBB, and OpenMP) facilitates integration with existing software. Develop high-performance applications rapidly with Thrust!

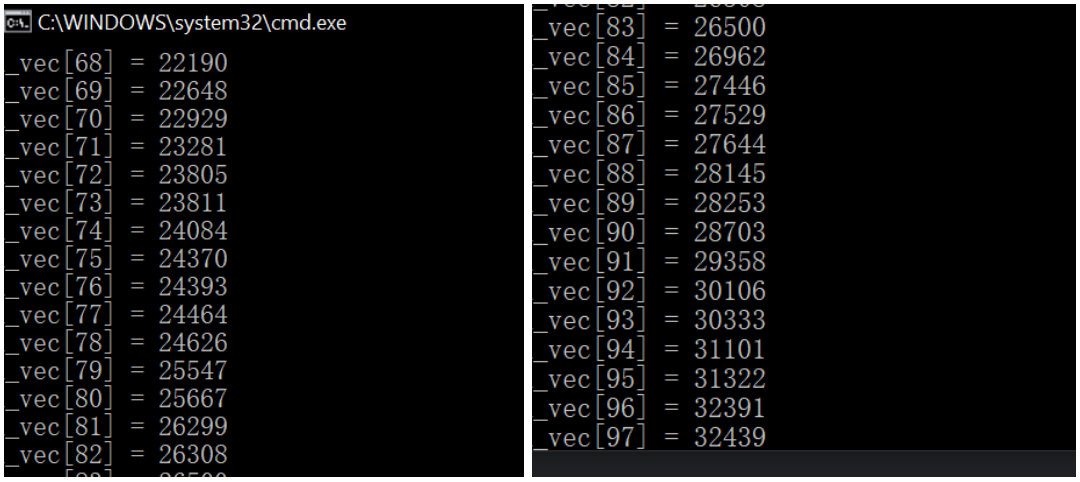
Thrust is included in the NVIDIA HPC SDK and the CUDA Toolkit.

Question 1:

Edit the code in the P8Q1 to perform the following:
• Copy results from **device to host** by using this:
thrust::copy(XXXXXXX)
• Use a for loop to display the host vector.
Instructions about where to place each part of the code is demarcated by the
//@@ comment lines.

This program generates random numbers serially and then transfers them to a parallel device where they are sorted. Display the sorted integers.

Output:

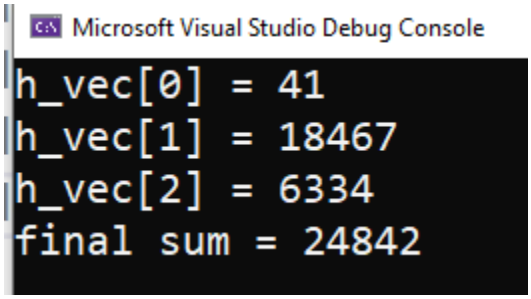


Question 2:

Edit the code in the P8Q2 to perform the following:
• Generate a thrust::host_vector<int> for host input arrays
• Copy host input array from host to device
• Parallel Vector Addition is executed by
int x = thrust::reduce(d_vec.begin(), d_vec.end(), 0, thrust::plus<int>());
• Display the final sum of x
Instructions about where to place each part of the code is demarcated by the //@@ comment lines.

This program generates random X numbers serially and then executes vector addition in parallel. Display the sum.

Sample output for 3 random generated values:



CUDA Image Color to Grayscale

Question 3:

The purpose of this lab is to convert an RGB image into a grayscale image. The input is an RGB triple of float values and the student will convert that triple to a single float grayscale intensity value. A pseudo-code version of the algorithm is shown below:

```
for ii from 0 to height do
     for jj from 0 to width do
          idx = ii * width + jj
          # here channels is 3
          r = input[3*idx]
          g = input[3*idx + 1]
          b = input[3*idx + 2]
          grayImage[idx] = (0.21*r + 0.71*g + 0.07*b)
     end
end
```

Edit the code in P8Q3 to perform the following:

- allocate device memory
- copy host memory to device
- initialize thread block and kernel grid dimensions
- invoke CUDA kernel
- copy results from device to host
- deallocate device memory

Instructions about where to place each part of the code is demarcated by the //@@ comment lines.

Hints: size in bytes to copy = imageWidth * imageHeight * imageChannels * sizeof(float)

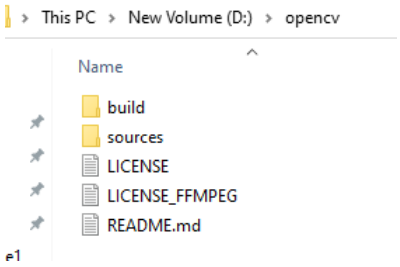
Resource: OpenCV for C++ Installation Guide

Step to configure OpenCV in cpp project:

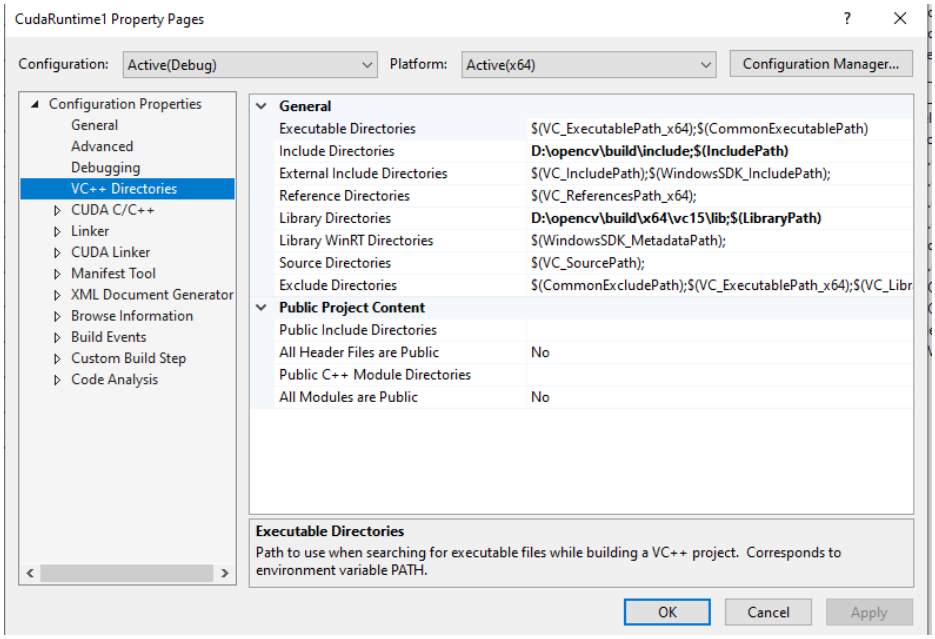
Step 1:

Directly download **opencv 4.2.0** [Win pack] from here
release Archives - OpenCV
* The latest **opencv 4.6.0 CANNOT** be used for running the sample code.

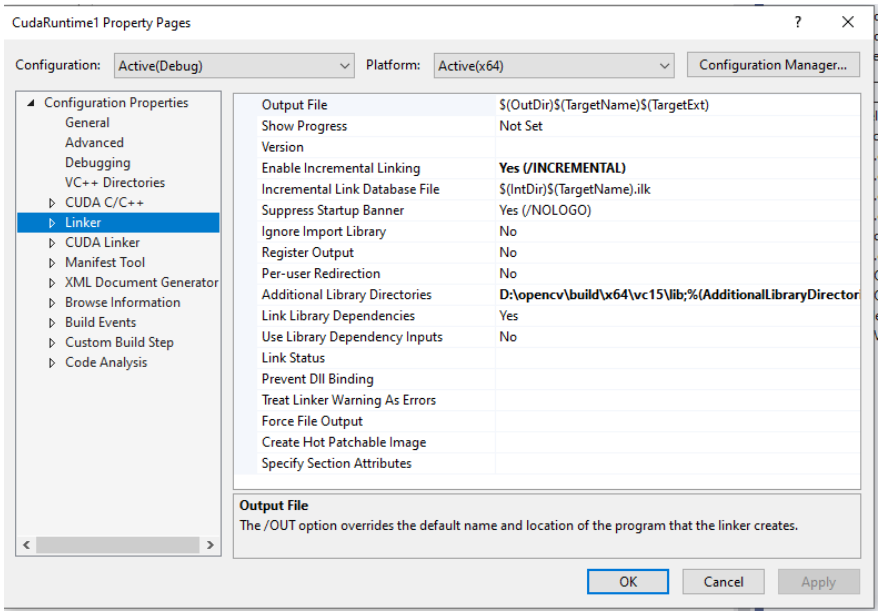
Step 2:
Run “opencv-4.2.0-vc14_vc15.exe” to unzip it and put at “D:\”
* Make sure that you have “D:\opencv\build…” folders



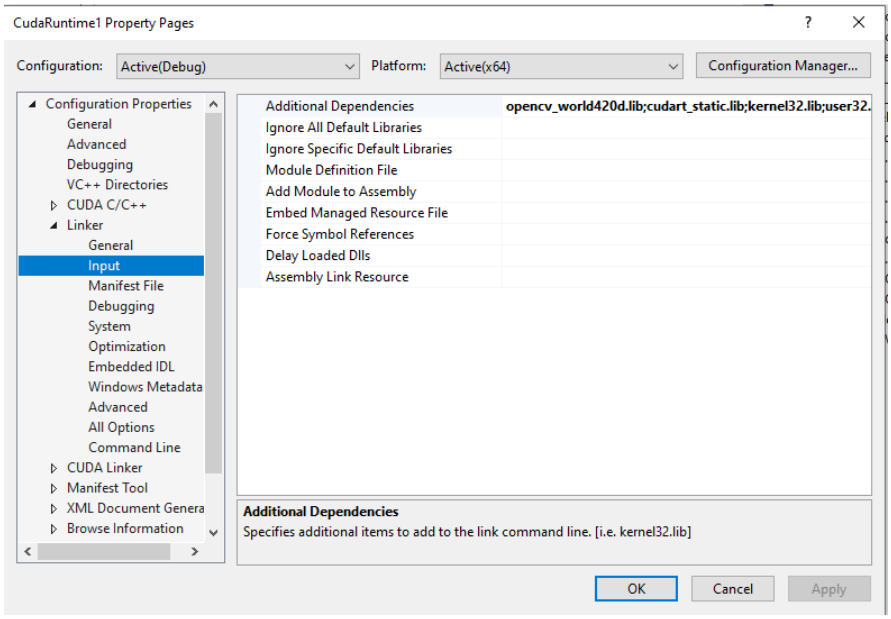
Step 3:
Go to visual studio project properties > VC++ Directories, add in these:
Include Directories “D:\opencv\build\include;”
Library Directories “D:\opencv\build\x64\vc15\lib;”



Step 4:
Go to visual studio project properties > Linker, add in this:
Additional Library Directories "D:\opencv\build\x64\vc15\lib;"

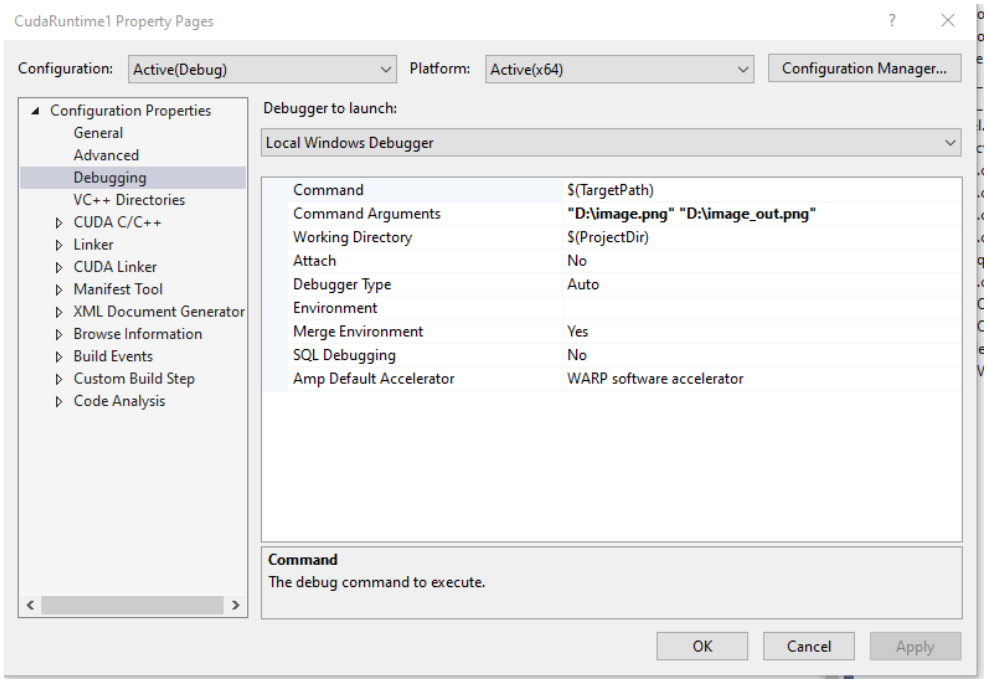


Step 5:
Go to visual studio project properties > Linker > Input, add in this:
Additional Dependencies "opencv_world420d.lib;"

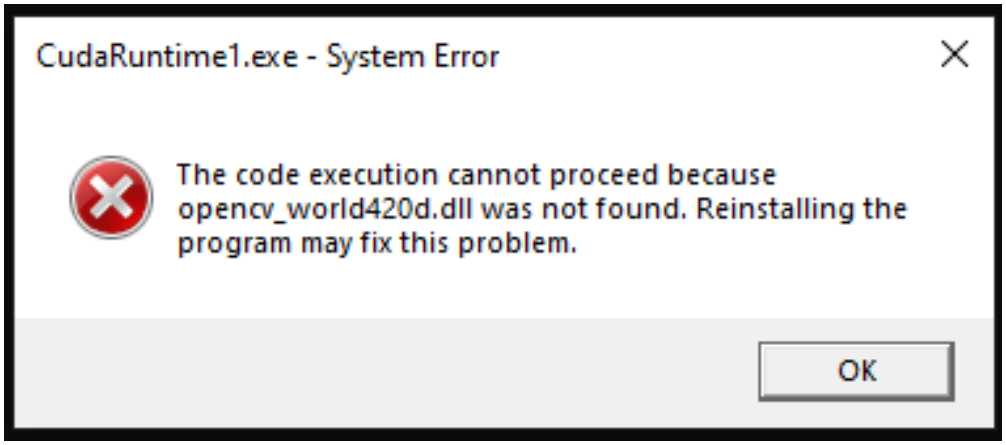


Step 6:
Go to visual studio project properties > Debugging , add in this:
Command Arguments “D:\image.png” “D:\image_out.png”

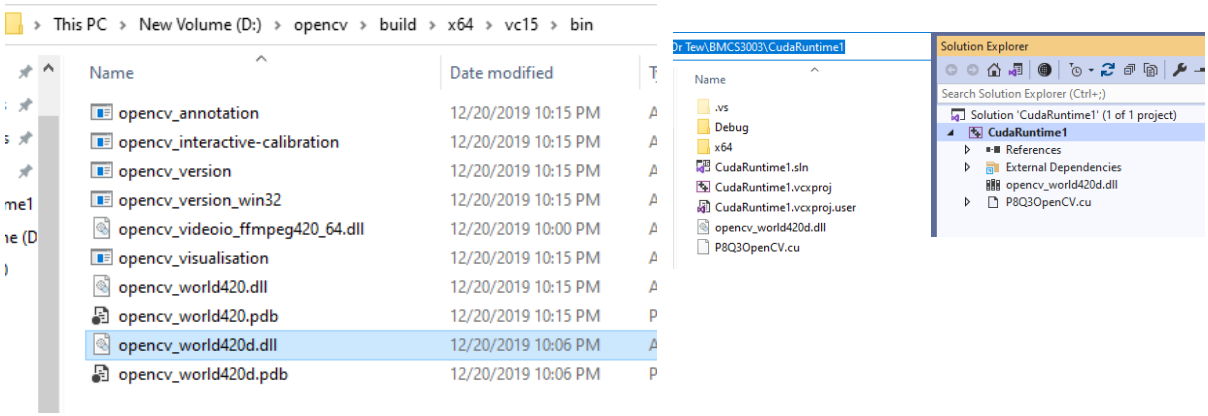
*** Take note that the Open / Close Double Quote “” is typed using your keyboard (Not Ctrl+C and Ctrl+V from HERE!)**



Step 7:
Put your **image.png** at D:\image.png location, and press F5 in your project:
It should show this Error.



Step 8:
Go to D:\opencv\build\x64\vc15\bin, copy the “opencv_world420d.dll” file, and paste it in your project folder. Repeat Step 7 again, you should be able to get D:\image_out.png.



*** Please make sure that all your coding files are copied and pasted in Windows Explorer (DO NOT drag and drop / DO NOT include that file from visual studio). The file icon in Solution Explorer IS NOT WITH a blue square at the left bottom icon that indicates “shortcut”.**

Output:

Original Image	Converted Image