

## Report for Stereo Vision task in OpenCL

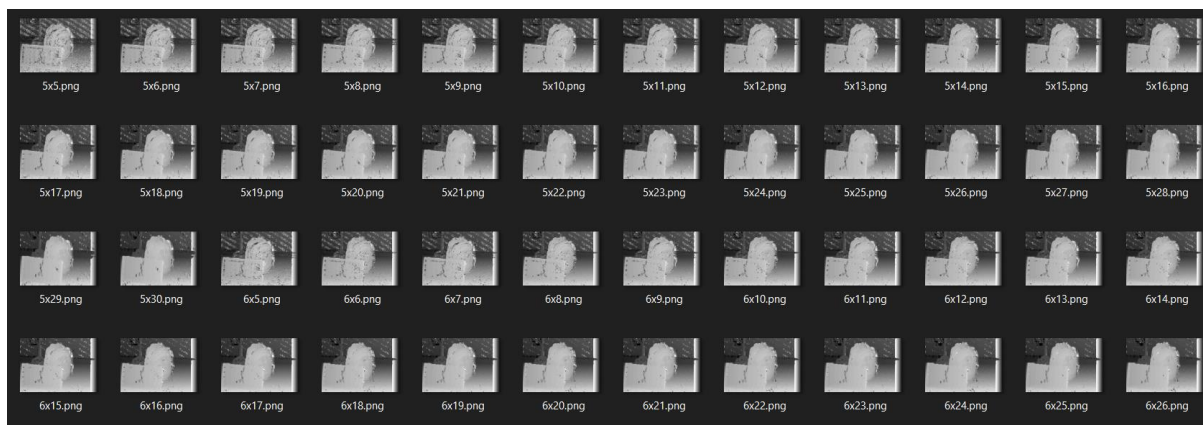
Nikunj Arora – 2646352

Dennis Goyal – 2627601

With the same constants as Stereo Vision Task in plain C:

```
C:\Users\nikun\Documents\Visual Studio 2019\Projects\OpenCL\Debug\StereoVisionCL.exe
Reading Left Image...Done (1.32633 s)
Reading Right Image...Done (1.19134 s)
Device: Intel(R) Corporation
OpenCL Version: OpenCL 2.1 NEO
Converting Left Image to grayscale...Done (0.000901333 s)
Converting Right Image to grayscale...Done (0.00086925 s)
Calculating Left Disparity Map...Done (1.82574 s)
Calculating Right Disparity Map...Done (1.82505 s)
Performing Cross Checking...Done (0.000391833 s)
Performing Occlusion Filling...Done (0.0081235 s)
The program took 8.4591 s
```

We observed that unlike running the program of CPU which took several minutes, running on GPU took just over 1 second. Therefore, we can run this program for different window sizes.



After trying out window sizes from 5x5 all the way to 30x30, we settled with a window size of 15x15. Therefore, the new constants:

maxDisp = 64

windowWidth = 15

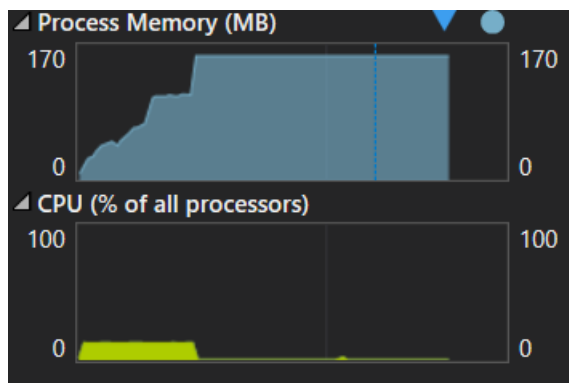
windowHeight = 15

crossCheckingThreshold = 2

occlusionNeighbours = 256

## OpenCL on Intel UHD Graphics 620

### Resource Usage:



RAM: 154.7 MB

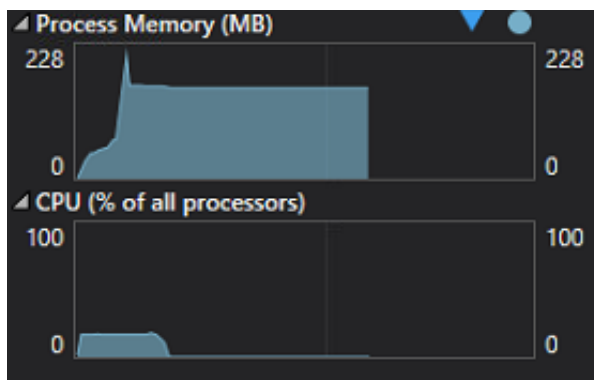
CPU: 13%

### Output:

```
C:\Users\nikun\Documents\Visual Studio 2019\Projects\OpenCL\Debug\StereoVisionCL.exe
Reading Left Image...Done (1.23567 s)
Reading Right Image...Done (1.25279 s)
Device: Intel(R) Corporation
OpenCL Version: OpenCL 2.1 NEO
Converting Left Image to grayscale...Done (0.0009455 s)
Converting Right Image to grayscale...Done (0.000880666 s)
Calculating Left Disparity Map...Done (5.57571 s)
Calculating Right Disparity Map...Done (5.59769 s)
Performing Cross Checking...Done (0.000168 s)
Performing Occlusion Filling...Done (0.0240486 s)
The program took 16.5016 s
```

## OpenCL on Nvidia GTX 1070ti

### Resource Usage:



Ram: 153.4 MB

CPU: 17%

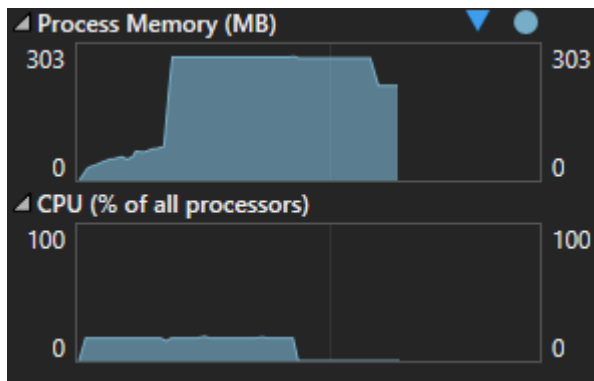
### Output:

```
D:\CppProjects\OpenCL\Debug\StereoVisionCL.exe
Reading Left Image...Done (0.879246 s)
Reading Right Image...Done (0.915627 s)
Device: NVIDIA Corporation
OpenCL Version: OpenCL 1.2 CUDA
Converting Left Image to grayscale...Done (0.00017056 s)
Converting Right Image to grayscale...Done (0.000171008 s)
Calculating Left Disparity Map...Done (0.720361 s)
Calculating Right Disparity Map...Done (0.692959 s)
Performing Cross Checking...Done (2.272e-05 s)
Performing Occlusion Filling...Done (0.01374 s)
The program took 3.43049 s
```

As we can see the Nvidia GPU is approximately 5x faster than the integrated Intel GPU.

## CUDA implementation on Nvidia GTX 1070ti

### Resource Usage:



RAM: 274.1 MB

CPU: 17%

### Output:

```
D:\CppProjects\OpenCL\x64\Debug\StereoVisionCuda.exe
NBody.GPU
=====

CUDA version:    v10020
CUDA Devices:

0: GeForce GTX 1070 Ti: 6.1
  Global memory:  8192mb
  Shared memory:  48kb
  Constant memory: 64kb
  Block registers: 65536

  Warp size:      32
  Threads per block: 1024
  Max block dimensions: [ 1024, 1024, 64 ]
  Max grid dimensions: [ 2147483647, 65535, 65535 ]

Reading Left Image...Done (1.67785 s)
Reading Right Image...Done (1.61086 s)
Converting Left Image to grayscale...Done (0.000286304 s)
Converting Right Image to grayscale...Done (0.000291456 s)
Converting Left Disparity Map...Done (2.36797 s)
Converting Right Disparity Map...Done (2.35821 s)
Performing Cross Checking...Done (5.0176e-05 s)
Performing Occlusion Filling...Done (0.0895211 s)
The program took 8.55352 s
```

From what we observe, CUDA implementation performs worse than OpenCL implementation. This is probably due to inefficient CUDA code.