NOVA SCHOOL OF
SCIENCE & TECHNOLOGY

Advanced Topics in Digital Image Processing

Lesson 6 - *Image Manipulation in OpenCL*

Objectives:

The main purpose of this lesson is to create and test image manipulation kernels in the OpenCL architecture and compare it to traditional CPU processing.

Exercises:

1 – Brightness and contrast of an image

The purpose of the exercise will be to perform an adjustment of brightness and contrast operation on an image using Buffer memory objects.

To process an image, kernels can be executed with an image-like 2D organization (1 work-item per pixel). However, we must not forget that there is padding at the end of each line and that the kernel should handle these situations. The kernel parameters should be:

- Pointer to the input image
- Pointer for output image
- Width and height of the image
- Padding value at the end of a given line (calculated with numpy.ndarray properties)
- Brightness (int) and Contrast (float)

2 – Evaluate execution time

Compare the execution time of the B&C in OpenCL against the same in CPU (optimized OpenCV). Calculate separately the time to prepare and configure OpenCL and the time to run the function. Use the Time object to assess the time differences.
Test with images of different sizes, preferably large ones.

Example:

```
import time

start_time = time.time()

…. code

print("- execute --- %s seconds ---" % (time.time() - start_time))
```

3 – Sobel and thresholding filtering

Implement a sobel + threshold filter in OpenCL using Image objects. After obtaining the sobel result for all RGB components, apply a filter to:

- remove all colored pixels (not gray scale)
- set to white all highest intensity ones.

This filter can be done setting the output pixel to white if the difference (*diff*) between all components is below a predefined threshold *t1* (received as parameter) and if average value of the 3 RGB components is above a second *t2* threshold (received as parameter). All other cases are set to black.

$$diff = |sobel_r - sobel_g| + |sobel_r - sobel_b| + |sobel_g - sobel_b|$$

$$Average(sobel) = \frac{sobel_r + sobel_g + sobel_b}{3}$$

$$output(x,y) = \begin{cases} white \quad if \quad diff < t1 \; and \; average(sobel) > t2 \\ \\ black \quad if \quad else \end{cases}$$