MASSEY UNIVERSITY
TE KUNENGA KI PŪREHUROA
UNIVERSITY OF NEW ZEALAND

Computer Vision 159.731
S1 2020

Assignment 1 - Filtering and blob finder – due date 12/06/2020

Your task is to implement the Kuwahara Filter as described in the study guide. However, your program must take 3 parameters, e.g.

```
kuwahara input.jpg output.jpg 7
```

The first two parameters are the input image and the output image, after passing the Kuwahara filter. The last parameter is the neighbourhood size. Your program should allow **odd neighbourhood sizes** between 3 to 15.

The implementation needs to be as *fast* as possible. In order to achieve that, you need to use **Integral Images** (or Summed-Area Tables). Two SATs are needed, one with the *sum of the pixel values*, and the other with the *sum of the squares of the pixel values*. The time (runtime) should be measured using the utility time. This is a unix/linux utility available on the machines in the CV lab. Usage: '**`time ./<yourcode>`**'.

The input images should be grayscale. Remember to either convert colour images or loading with the option:

```
imageinput = imread( argv[1], CV_LOAD_IMAGE_GRAYSCALE);
```

When using a kernel, ignore neighbour pixels that are outside the image. They should not be part of the calculation (this is different than considering that these values are zero). Therefore the edges of the images tend to maintain the mean value of one of the sub-window that lies within the boundaries of the image.

The code must be in C or C++, it must use OpenCV and it must compile with the gcc compiler in the Linux laboratory or equivalent (you can use Windows, but double check that your code works in the lab machines).

You can test your solution using the images available on Stream. The result with a kernel of size 7 should be similar to the image below (you can check it by using subtraction).

Figure 1: limes.tif before and after the Kuwahara 7x7 filter.

The assignment is worth **10 marks.**
5 marks – correctness (as similar as possible to the expected result)
5 marks – speed (ranking the codes submitted and comparing to the optimal solution), measured using kernels from 3x3 up to 15x15.
Submit your assignment on Stream **by 12/06/2020**. Late assignments that are not justified may lose 10% per day after the due date.