**CSC 545/645 Computer Speech, Music and Images**

**Exercise No. 5, March 3, 2020**

**Edge Detection**

**Goal**

1 Experiment with edge detection using various kernels

2. Experiment with edge enhancement through adding edges and unsharp filtering

**Procedure**

Write a Processing program to perform edge detection on an image. Try the Prewitt or Sobel (try both if you have time) and the Laplacian edge detector; kernels for these operators, and for the Laplacian operator, are provided in the skeleton code—Ex06b\_edgeDetection. You can decide whether you want to convert the image to grayscale before filtering; if you don’t convert to grayscale, you may see that some of the operators don’t work very well on color images.

Experiment with the Laplacian operator—compare Laplacian on the original image with Laplacian on the blurred image (use the low pass box filter, also provided, to blur the image). Does pre-blur improve the result? Edge detectors normally preserve only edges, with all other pixels set to black. You can get an “emboss” effect by adding 128 to pixels after edge detection (try it).

Implement a sharpening filter by adding the output of the Laplacian filter to the original image (when you add images, be sure to keep your pixel values in range).

*Unsharp filtering* is popular in some applications (astronomy, for example). Unsharp filtering gets its name because a blurred (unsharpened) image is subtracted from the original, then the output is added back to the original image (be sure all pixel values stay in range). The result, in spite of the filter’s name, is to sharpen the image. Implement an unsharp filter and compare it to sharpening using the Laplacian operator (add the Laplacian edge image to the original image—again, make sure pixel values stay in range).

As usual, the skeleton code is in the Exercises folder on the class server. There is an image in the data folder; feel free to add others.