

## Assignment 4 (40 points)

### Edge Detection (10/20)

Use the `edge` function to generate results for Roberts, Canny, Sobel, and Prewitt operators on an image of your choice. Note also that the various edge functions support a number of parameters – feel free to explore those to get more interesting results. State which operator gives the best performance and why you think so.

### Edge Filter (10/20)

Design a 7x7 “Sobel” operator and filter your image from the “Edge Detection” task above with your filter. The main idea behind the design of a proper Sobel-ish operator is to model the Gaussian derivate in one direction and the Gaussian in the perpendicular direction:

$$\frac{\partial G(x, y, \sigma)}{\partial x} = -\frac{x}{2\pi\sigma^4} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$

In this formulation,  $\sigma$  becomes a parameter of the filter; you can choose whatever you like, although choosing 1 makes the math much simpler.  $x$  and  $y$  are the distance from the center pixel of your filter. Aside from the size of your filter, how does it differ from the standard Sobel operator? How are the image results different than what you saw applying the standard Sobel operator?

### 481 Students: Noise reduction (20/0)

- On an image of your choice, use the `imnoise` function to generate two noise corrupted images as follows: (the article at <https://stackoverflow.com/questions/14435632/impulse-gaussian-and-salt-and-pepper-noise-with-opencv> show one way to do this with Python and opencv)  

```
noise_gaussian = imnoise(Image, 'gaussian', 0, 0.05);  
noise_saltAndpepper = imnoise(Image, 'salt & pepper', 0.02);
```
- Use `subplot` to display the original image and the two noise corrupted images.
- Use the function `fspecial` to design averaging filters of size (3x3), (5,5), and (7x7). Use `subplot` to display the `noise_saltAndpepper` image and the three averaged filtered results. Do the same for the `noise_gaussian` image.
- Use the `medfilt2` function to perform median filtering on the `noise_saltAndpepper` image. Design the median filters to work with window sizes of (3x3), (5x5), and (7x7). Use your filters on the `noise_gaussian` image also and display as in part c).

### General submission instructions:

- Be kind to your aging, over-worked professor and submit only a single document. This can be pdf, MS Word, OpenOffice, etc. Do not submit a zip file.

- (b) Your single document should include the input image for your problem, if required, and answers to each of the sub-problems (text, image or both, as appropriate). Your document should also include code that you wrote to generate your answers.
- (c) You may use any images you like for the programming; I encourage you to use images that might be useful/interesting for your final project.
- (d) Feel free to use whatever functions MatLab supplies (unless otherwise specified). Also feel free to write your own, if you are so inclined; it will take more time, but you will gain a deeper understanding of the material.
- (e) Point values for each question are indicated as  $(x/y)$  in which  $x$  is the point value for 481 students and  $y$  is the point value for 381 students.