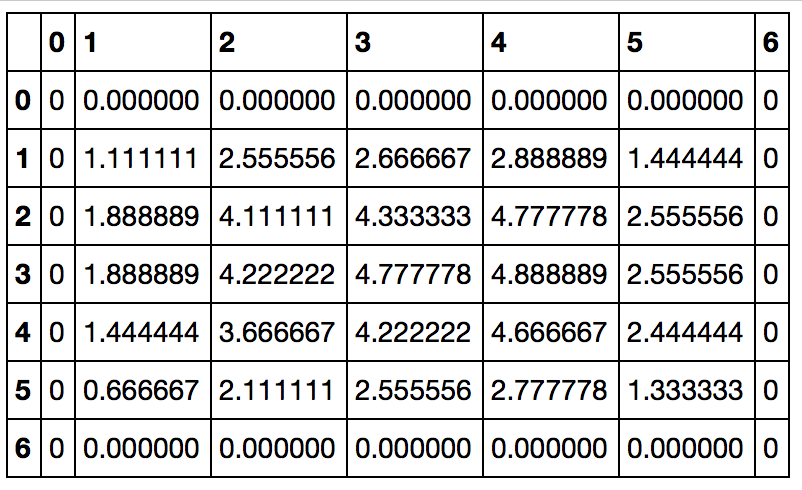
**CS 5670: Computer Vision HW #1**

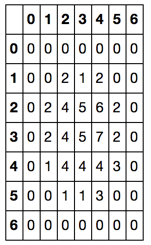
Jeff Ponnor, Daniel Reidler

**Task 1: Image Filtering and Enhancement**

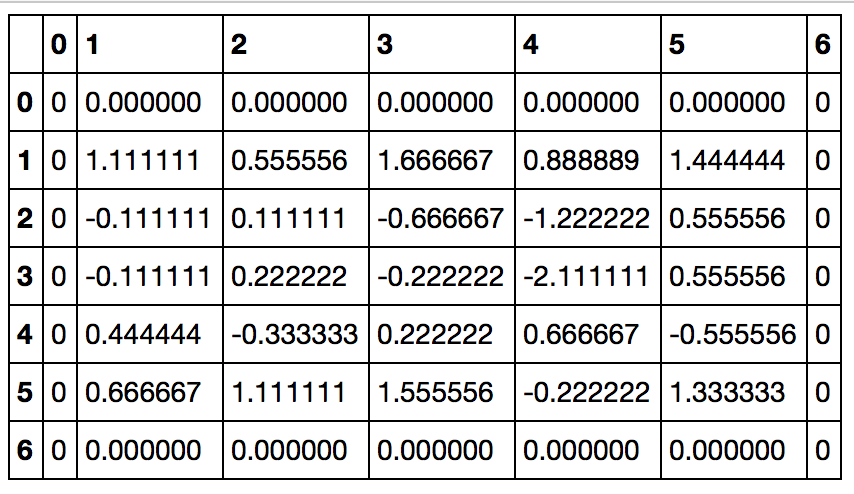
1. Output of Image Convolution:



1. Output of Convolution With Median Filter:



The key difference from 1:



1. Gradient Magnitude and Direction using Sobel:

Gradient Magnitude: 4.472136

Gradient Direction: -1.107149

1. a) Filter using distance between pixels: We use a simple Gaussian 3x3 kernel.

[0.0625,0.125,0.0625],

[0.125,.25,0.125],

[0.0625,0.125,0.0625], (Please see comments in code for more details.)

b) Filter using distance between pixel values:

-- For each 3x3 neighborhood, we order pixels values least to greatest. If center pixel value is the least or greatest in the neighborhood, it will weigh 6x and we will equally weigh the other 8 pixels. Else, the center pixel value is weighted 4x, Weight nearest pixel value below center pixel- 2x, Weight nearest pixel value above center pixel- 2x. In both cases, we weight the remaining pixels 1x. We set the center pixel value to weighted average of values.

c) Filter that takes into account both distances:

-- We take the average of the values computed in a and b.

Results:

|  |  |  |
| --- | --- | --- |
| Filter 1 | Filter 2 | Filter 3 |
| 1.4ResultsImages/camera_man_filter1applied.png | 1.4ResultsImages/camera_man_filter2applied.png | 1.4ResultsImages/camera_man_filter3applied.png |
| 1.4ResultsImages/house_filter1applied.png | 1.4ResultsImages/house_filter2applied.png | 1.4ResultsImages/house_filter3applied.png |
| 1.4ResultsImages/lena_filter1applied.png | 1.4ResultsImages/lena_filter2applied.png | 1.4ResultsImages/lena_filter3applied.png |

1. Unsharp Masking:

|  |  |  |
| --- | --- | --- |
| Gray | Sigma=0.75 | Sigma=2.5 |
| 1.5Results/camera_man_gray.png | 1.5Results/camera_man_sigma0_75.png | 1.5Results/camera_man_sigma2_5.png |
| 1.5Results/house_gray.png | 1.5Results/house_sigma0_75.png | 1.5Results/house_sigma2_5.png |
| 1.5Results/lena_gray.png | 1.5Results/lena_sigma0_75.png | 1.5Results/lena_sigma2_5.png |

The differences in the images

Cameraman -

1. The original grayscale image is the sharpest.

2. The image with sigma = 2.5 is "dull" in the sense that the image seems to be more gray and use a smaller range of values.

3. The image with sigma = 0.75 is not as sharp as the original but also still clearer than the image with sigma = 2.5

House

1. The original grayscale image is the least sharp of the images.

2. The image with sigma = 2.5 is the sharpest. Specifically, the contrast between the bricks and the house gutter is more sharp.

3. image with sigma = 0.75 is sharper than the original but not as sharp as sigma=2.5

Lena

1. Like the cameraman, the original is the sharpest. Though, the image with sigma= 0.75 is almost as sharp.

2. The image with Sigma = 2.5 is not as sharp.

**Task 2: Color Quantization with K-Means**

1. A
2. B
3. C
4. D
5. E

The results

**Task 3: Edge Detection**

1. Explain in a few sentences the way of operation of each of the three detectors.

Canny:

The Canny edge detector uses the fact that the derivative of the Gaussian Laplace can be used to approximate edges in an image. It is typically used after a Gaussian Laplace smoothing on an image. The Canny edge detector uses the sigma value as the input.

Sobel:

The Sobel edge detector is comprised of two 3x3 kernels that are convolved with an image to produce an edge detected graph. This operator requires a threshold as it is not consistent

Gaussian-Laplace Filtering:

The Gaussian-Laplace Filter is a distance based filter than heavily weights the center cell and decreases in weight outward. This filter approximates the derivate of the image. Once a Gaussian-Laplace filter is applied, region’s values reflects its

2.

Sobel Operation

|  |  |  |
| --- | --- | --- |
| Threshold = 0.05 | Threshold = 0.1 | Threshold = 0.3 |
| 3A/Sobel/sobel_threshold=0_05church.png | 3A/Sobel/sobel_threshold=0_1church.png | 3A/Sobel/sobel_threshold=0_3church.png |
| 3A/Sobel/sobel_threshold=0_05golf.png | 3A/Sobel/sobel_threshold=0_1golf.png | 3A/Sobel/sobel_threshold=0_3golf.png |
| 3A/Sobel/sobel_threshold=0_05nuns.png | 3A/Sobel/sobel_threshold=0_1nuns.png | 3A/Sobel/sobel_threshold=0_3nuns.png |

|  |  |  |
| --- | --- | --- |
| Threshold = 0 | Threshold = 0 | Threshold = 0 |
| 3A/Sobel/sobel_threshold=0church.png | 3A/Sobel/sobel_threshold=0golf.png | 3A/Sobel/sobel_threshold=0nuns.png |

Gaussian-Laplace, threshold =0

|  |  |  |
| --- | --- | --- |
| Sigma = 0.75 | Sigma = 1.75 | Sigma = 2.75 |
| 3A/Gauss-Laplace-Thres=0/LoG_thres=0_sigma=0_75church.png | 3A/Gauss-Laplace-Thres=0/LoG_thres=0_sigma=1_75church.png | 3A/Gauss-Laplace-Thres=0/LoG_thres=0_sigma=2_75church.png |
| 3A/Gauss-Laplace-Thres=0/LoG_thres=0_sigma=0_75golf.png | 3A/Gauss-Laplace-Thres=0/LoG_thres=0_sigma=1_75golf.png | 3A/Gauss-Laplace-Thres=0/LoG_thres=0_sigma=2_75golf.png |
| 3A/Gauss-Laplace-Thres=0/LoG_thres=0_sigma=0_75nuns.png | 3A/Gauss-Laplace-Thres=0/LoG_thres=0_sigma=1_75nuns.png | 3A/Gauss-Laplace-Thres=0/LoG_thres=0_sigma=2_75nuns.png |

Canny Filter Operation

|  |  |  |
| --- | --- | --- |
| Sigma = 0.75 | Sigma = 1.75 | Sigma = 3.75 |
| 3A/canny/canny_church_sigma0_75.png | 3A/canny/canny_church_sigma1_75.png | 3A/canny/canny_church_sigma3_75.png |
| 3A/canny/canny_golf_sigma0_75.png | 3A/canny/canny_golf_sigma1_75.png | 3A/canny/canny_golf_sigma3_75.png |
| 3A/canny/canny_nuns_sigma0_75.png | 3A/canny/canny_nuns_sigma1_75.png | 3A/canny/canny_nuns_sigma3_75.png |

The threshold parameter decides how many total “edges” we include. The higher the threshold, the fewer edges we will include. The lower the threshold, the more “edges” we will include. The threshold parameter determines the balance between precision and recall. If we use a higher threshold, we may increase our precision because we will have fewer false positive, however, we may decrease our recall because some edges may not be greater than the threshold. Conversely, the lower the threshold, the lower our precision since we will likely falsely label edges. But, we will have a higher recall value because we will likely capture most of the “true edges”.

The sigma parameter influences the amount of noise or how smooth the image will be. However, if sigma is too high, we may also miss some edges. If the original image has similar intensities we will likely use a smaller sigma value to get the maximum edge detection.

In the images below, we see that for each image, a different sigma value works best. And, they will each have a different optimal sigma value.

B)

Precision and Recall are important because Precision explains the percentage of edges reported that are truly edges and Recall explains the percentage of edges detected from all the edges. A low recall value means that we are not detecting all of the edges and may mean that we are missing critical edges from the image. This happens when a poor threshold value is used.

A low precision value means we are incorrectly classifying some points: either missing real edges (true positives) or including false edges (false positives). If we’re missing real edges, then we are simply not detecting edges. And, if we are including false edges we are adding false edges to the image.

3. Zero Crossing is taken preference over thresholding.