

Assignment 2. Feature Detection (Due Oct 13 before midnight)

Note: Turn in a PDF document (**in writing; please type**) containing a list of your .m files (not the code itself), images showing the results of your experiments, and an analysis of the results. All the writings must be soft copies in print and be sent to Prof. Zhu <cv.zhu.ccny@gmail.com> via email. For the programming part, **send ONLY your source code by email**; please don't send in your images and executable (even if you use C++). You are responsible for the loss of your submissions if you don't write **"CSc 471 Computer Vision Assignment 2"** in the subject of your email. Do write your names and IDs (last four digits) in both both of your report and the code. **Please don't zip your report with your code and other files; send me the report in a separate PDF file. The rest can be in a zipped file.**

Choose one or two of the images on the course web page for this assignment or use an image of your own choice. If you use a different image, be careful that the image has not been saved under JPEG using a high level of compression. This often introduces artifacts into the image that confound subsequent analyses of the results. For the most part, you should apply your operators to a gray-scale version of the image.

1. (20 points) Generate the histogram of the image you are using, and then perform a number of histogram operations (such as contrast enhancement, thresholding and equalization) to make the image visually better for either viewing or processing (10 points). If it is a color image, please first turn it into an intensity image and then generate its histogram. Try to display your histogram (5 points), and make some observations of the image based on its histogram (5 points). What are the general distributions of the intensity values? How many major peaks and valleys does your histogram have? How could you use the histogram to understand, analyze or segment the image? Please also display the histograms of the processed images and provide a few important observations.

2. (20 points) Apply the 1x2 operator and Sobel operator to your image and analyze the results of the gradient magnitude images (including vertical gradients, horizontal gradients, and the combined) (10 points). Please don't forget to normalize your gradient images, noting that the original vertical and horizontal gradients have both positive and negative values. I would recommend you to display the absolute values of the horizontal and vertical gradient images. Does the Sobel operator have any clear visual advantages over the 1x2 operator? Any disadvantages (5 points)? If you subtract the 1x2 edge image from the Sobel are there any residuals? You might use two different types of images: one ideal man-made image, and one image of a real scene with more details (5 points). **(Note: don't forget to normalize your results as shown in slide # 29 of feature extraction lecture: part 2)**

3. (20 points) Generate edge maps of the above two combined gradient maps (10 points). An edge image should be a binary image with 1s as edge points and 0s as non-edge points. You may first generate a histogram of each gradient map, and only keep certain percentage of pixels (e.g. 5% of the pixels with the highest gradient values) as edge pixels (edgels). Use the percentage to automatically find a threshold for the gradient magnitudes. In your report, please write up the description and probably equations for finding the threshold, and discuss if 5% is a good value. If not what is (5 points)? You may also consider to use local, adaptive thresholds to different portions of the image so that all major edges will be shown up nicely (5 points). In the end, please try to generate a sketch of an image, such as the ID image of Prof. Zhu.
4. (20 points) What happens when you increase the size of the edge detection kernel from 1×2 to 3×3 and then to 5×5 , or 7×7 ? Discuss computational cost (in terms of members of operations, and the real machine running times - 5 points), edge detection results (5 points) and sensitivity to noise, etc. (5 points). Note that your larger kernel should still be an edge detector. Please list your kernels as matrices in your report, and tell us what they are good for (5 points).
5. (20 points) Suppose you apply the Sobel operator to each of the RGB color bands of a color image. How might you combine these results into a color edge detector (5 points)? Do the resulting edge differ from the gray scale results? How and why (5 points)? You may compare the edge maps of the intensity image (of the color image), the gray-scale edge map that are the combination of the three edge maps from three color bands, or a real color edge map that edge points have colors (5 points). Please discuss their similarities and differences, and how each of them can be used for image enhancement or feature extraction (5 points). Note that you want to first generate gradient maps and then using thresholding to generate edge maps. In the end, please try to generate a color sketch of an image, such as the ID image of Prof. Zhu. You may also consider local, adaptive thresholding in generating a color edge map.