CIS 465&602 Computer Vision

Assignment 2

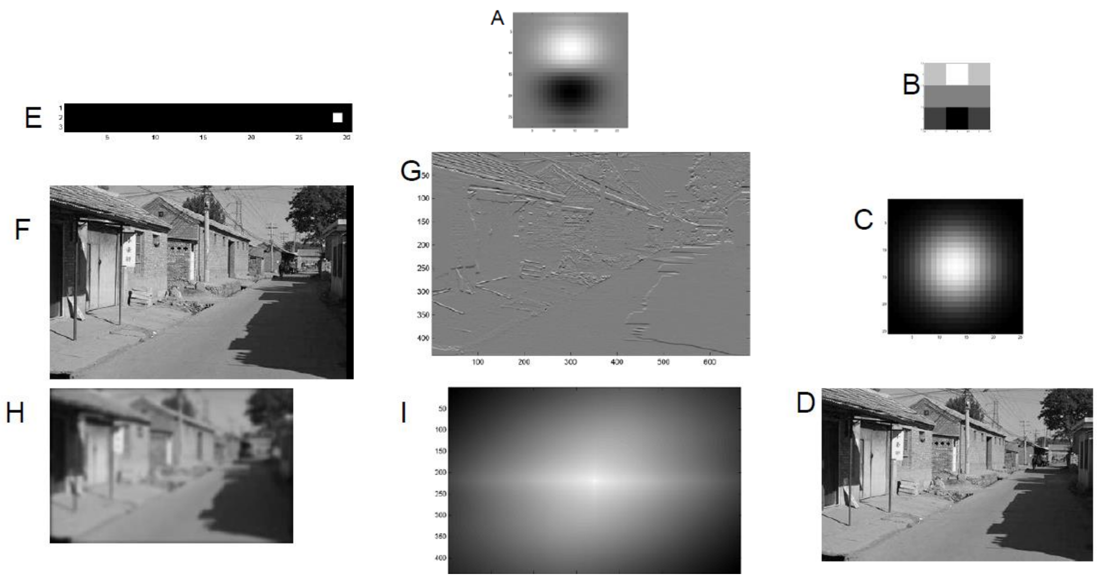
Due: 3/3/2021 23:59PM Submit to myCourses

1. Fourier Image (30 pts)

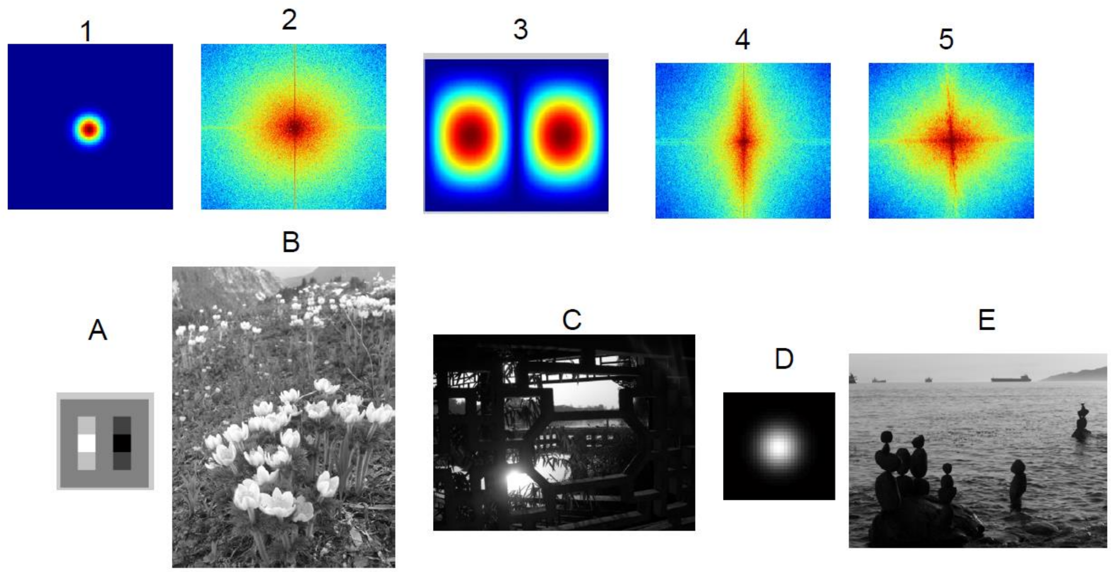
Hint: Feel free to use MATLAB to figure out Fourier images

(a) Fill the blank:

(i) _ = D * B (ii) A = _ * _ (iii) F = D * _ (iv) _ = D * D



(b) Match the following Fourier images to their original image, e.g., 1->E



2. Difference-of-Gaussian (DoG) Detector (24 pts)

(a) The 1-D Gaussian is

$$g_\sigma(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{x^2}{2\sigma^2}\right)$$

Calculate its 2nd derivative with respect to x, and use MATLAB to plot it (use $\sigma = 1$).

(b) Use MATLAB to plot the difference of Gaussians in 1-D given by

$$D(x, \sigma, k) = \frac{g_{k\sigma}(x) - g_\sigma(x)}{k\sigma - \sigma}$$

using $k = 1.2, 1.4, 1.6, 1.8$ and 2.0. State which value of k gives the best approximation to the 2nd derivative with respect to x. You may assume that $\sigma = 1$. Note: Submit your MATLAB code in the pdf as a “code” block.

(c) The 2D equivalents of the plots above are rotationally symmetric. To what type of image structure will a difference of Gaussian respond maximally?

3. Canny Edge Detector (16 pts)

(a) Suppose that the Canny edge detector successfully detects an edge in an image. The edge (see Figure 1) is then rotated by θ, where the relationship between a point on the original edge (x, y) and a point on the rotated edge (x', y') is defined as

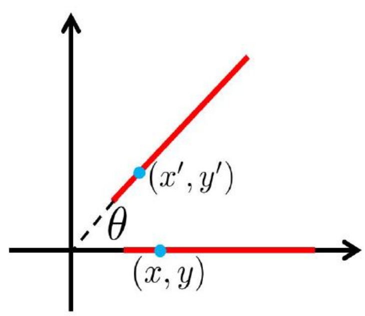


Figure 1: A rotated detected edge

Will the rotated edge be detected using the same Canny edge detector? Provide either a mathematical proof or a counter example.

Hint: The detection of an edge by the Canny edge detector depends only on the magnitude of its derivative. The derivative at point (x, y) is determined by its components along the x and y directions. Think about how these magnitudes have changed because of the rotation.

(b) After running the Canny edge detector on an image, you notice that long edges are broken into short segments separated by gaps. In addition, some spurious edges appear. For each of the two thresholds (low and high) used in hysteresis thresholding, explain how you would adjust the threshold (up or down) to address both problems. Assume that a setting exists for the two thresholds that produces the desired result. Briefly explain your answer.

4. Normalized Cross-correlation (30 pts)

Background: In class, we covered cross-correlation, in which a template image is multiplied with sections of a larger image to measure how similar each section is to the template. Normalized cross-correlation is a small refinement to this process. In rough terms, it works like this: before multiplying the template with each small section of the image, the image section is scaled and offset so it has zero mean and variance of 1. This increases accuracy by penalizing image sections which have high intensity but do not match the pattern of the template. The MATLAB function normxcorr2 performs this entire process for you.

(a) Use the provided *crossCorrelation.m* file to load the provided photo and template. Read the MATLAB documentation for normxcorr2 and use it to perform cross-correlation to find the section of the image that best matches the template. Note: include your MATLAB code (.m file) as part of the submission and describe why the peak occurs where it does. Also explain the straight-line artifacts in the cross-correlation. Please include the cross-correlation image in your solution.

(b) In *crossCorrelation.m*, perform cross-correlation using the larger template. Note that the larger template does not exactly match the image. Describe your results, and why they are different from part (a). What does this tell you about the limitations of cross-correlation for identifying objects in real-world photos? Also justify what applications cross-correlation would be good at. Hint: The provided code auto-scales image brightness so that it covers the full range. Therefore, same colors do not necessarily have the same value in the two cross-correlation images.

(c) Above, we saw that cross-correlation can be fragile. One way to make it less fragile is to perform cross-correlation using many templates to cover the different ways an object may appear in an image. Suppose we wish to search for N_R possible rotations of an object at N_S possible sizes. Assume the image is size $n \times n$ and the template is roughly size $m \times m$. How many math operations will the entire search require? Hint: We're just looking for a "Big-O Notation" estimate. In other words, you may neglect constant factors, such as the effects of image edge padding, and smaller terms.