BLG 453E Homework - 2

Due 26.10.2017 23:59

Policy: Please do your homework on your own (Do not copy paste your solutions from the internet or your friends). The code and the report you submitted must be your own work. All code must be implemented using **Python** programming language and **OpenCV Python wrapper**.

For your questions: albay@itu.edu.tr

1. **Spatial Filtering**

   Implement both of the following filters, and apply them on the following three images: cameramanN1.jpg (has Gaussian noise), cameramanN2.jpg (has impulsive noise), and cameramanN3.jpg (has a combination of both noises). For this homework, do not use built-in filtering related functions since here the goal is for you to get practice with spatial neighborhood operations

   (a) Implement a mean filter, display your results for the three images.

   (b) Implement a median filter, display your results for the three images.

   (c) Implement a mean-median filter with a blending parameter alpha. Play with alpha to obtain the visually most pleasing/denoised (noise removed) image. Display your results for the three images. Print your alpha parameter to the titles.

   (d) Which filter is more appropriate for which image? Explain/discuss why.

Figure 1: cameramaN1, cameranN2, cameramanN3 in order.

2. **Kirsh Compass Operator**

In this assignment you will implement an edge detection algorithm using the Kirsch Compass operator in eight directions (in your lecture notes). Load the image StairsBuildingsN.png, name it as I.

Figure 2

(a) Define the Kirsch operator matrices, and compute the convolution of each of the operator matrices with image I. Display your results. Hint: a sample result you will get from convolution with one of the masks:

Figure 3

(b) Recall that the gradient direction is determined by the operator rotation giving the maximum response. Determine which operator rotation gives the greatest response. Store this magnitude in an image called Jmag. To compute an edge detection result (binary edge map), threshold Jmag image with an appropriate threshold T_1. Display your result.

(c) Create a direction image Jdir where you encode each of the 8 directions with $N = 1, E = 2, S = 3, W = 4, NE = 5, SE = 6, NW = 7, SW = 8$. In fact

at each pixel coordinate of Jdir, we have a 2D vector field $V(y,x), U(y,x)$. You can fill in these two images (U, V) and the Jdir image as follows: Traverse/scan your image, if the magnitude image has value grater than some threshold T_2, record the corresponding encoding value in $[1, 8]$ in Jdir and the corresponding edge direction vector components. E.g. if your edge response at a pixel location (y, x) gives NW direction $(value = 7)$ then your unit direction vector is given by $U(y,x) = -sqrt(2)/2, V(y,x) = sqrt(2)/2$. Display the 2D vector field on top of your Jdir (or Jmag). (Hint: You can use ?quiver? function to display 2D vectors with an appropriate scale) To be able to better visualize your 2D vectors, you can decimate the 2D vector field by 4 for instance (e.g. $U(1:4:rows, 1:4:cols), V(1:4:rows, 1:4:cols)$) Hint for a portion of the image, you will get something like:

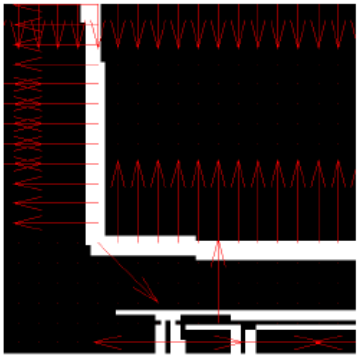

Figure 4

(d) Calculate the gradients of your image (compute the first derivatives of your image, i.e. the I_x and I_y). Compute the magnitude of your gradient image, Imag. Pick an appropriate threshold (you can check the intensity histogram of your gradient magnitude image to pick one). Display/print your gradient vector field superimposed on your Imag.

(e) Compare the results from part iii and iv.

(f) Repeat parts i-v now after filtering your image with a Gaussian filter of appropriate variance. Comment on your observations, how are the computations affected?