Computational Vision & Imaging - Lab 2

Hamid Dehghani,
School of Computer Science,
University of Birmingham, U.K.
B15 2TT

In this lab exercise, you will look at applying different noise removal filters and investigate the effect on edge detection.

You are asked to write a short (no more than 2 pages) report of your work, answering specific questions, and showing example images. This work is not assessed (it will not count towards your module mark) but you will get formative feedback.

STEP 1:

- Download the zip file and extract the .m script files, .mat saved variable files and the data files (.gif) for Lab 2 from CANVAS and save them in your working directory
- In MATLAB type

  shakey = read_image('','shakey.150.gif');

  This will load up the .gif file from the current directory into the variable shakey.
- You should also load up some noise and edge filters, type

  load filters
- Using the built-in procedure ***conv2*** convolve the image with the 3x3 Gaussian filter, and then the 5x5 filter. Can you see any difference between them? Try applying an edge filter to each and thresholding.

TASK 1:

Can you describe the effect in comparison with applying the edge filter to the image directly?

STEP 2:

- Using the function ***N(m,s,-3:1:3)*** you can create a discrete sample from a Gaussian (Normal) density. You need to specify the mean ***m*** (keep it at 0, think about why) and the standard deviation ***s***. The last term simply uses the code to create a vector in Matlab. So you can create larger and smaller vectors by altering the step size (the number between the two colons) or the limits of the vector (the starting and ending numbers of the last term). So now try creating a 9x9 Gaussian mask. To do this you will need to use matrix multiplication in the right way. Try some initial exploratory experiments with this, what happens to the image as you increase

the size of the mask? What happens as you increase the size of ***s***? Make detailed notes as you proceed about what you did and what you observed.

- Now apply gradient operators such as the Sobel operators to the blurred images. What happens to the edges in the heavily blurred case?

TASK 2:

What is the effect of increasing the size of the Gaussian Filter (3x3 versus 5x5 for example)? What is the effect of changing the standard deviation ***s***? Why do you see what you see?

STEP 3:

- Now compare the speed of applying two large 1D Gaussian filters in sequence, with applying a single equivalent 2D Gaussian filter that results from their multiplication. To test the CPU time used you can use a function called “tic”-“toc”. Can you detect differences in the CPU times as the mask sizes increase? You should check that the results are the same by examining areas of the image matrix in detail. Are there any effects due to small floating point errors?

STEP 4:

- Look at your Lecture notes and produce a 2D Laplacian filter.
- Now try applying the Laplacian operator to the Shakey image. You will need to calculate the zero-crossing for edges: you can use ***I_out = edge(I_in,’zerocross’)***, where ***I_in*** is the image convolved with the Laplacian, and ***I_out*** is the calculated edges. Think about the result. Why does it produce a poor result compared to the other operators?

TASK 3:

- I mentioned the Laplacian of the Gaussian in the lecture. How could you combine the idea of the Laplacian operator with the idea of Gaussian smoothing? Try out your ideas.