Computational Vision & Imaging - Lab 1

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

In this lab exercise, you will look at applying edge detectors to images and comparing two different methods. You will also learn how to write a function in MATLAB and evaluate the difference when using an approximation for calculation of a parameter.

At this stage, we will assume that you are able to use MATLAB, and understand its basic use.

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 files and the data files (.gif) for Lab 1 from CANVAS and save them in your working directorry
- In MATLAB type

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

  This will load up the .gif file from the current directory into the variable shakey. To see a help command for the function type

  help read_image
- You can display the image by typing

  show_image(shakey)

  This only works for grey images.
- Now you can load up different sets of masks. To load up the sobel masks type

  load sobel

  If you now type

  Who

  you will get a list of variable names. You can see the sobel horizontal mask by typing

  sobelX

  and the array will print on screen.

**STEP 2:**

- You can convolve the image with the mask by typing

shakey_sobelX = conv2(shakey,sobelX,'valid');

You can then display the new image.

- Apply both the sobelX and sobelY operators to the image. You can try to threshold the resulting images using

  show_image(abs(shakey_sobelX)>5)

  or whatever number you wish to use as the threshold.

**TASK 1:**

- Combine the two resulting arrays using Pythagoras theorem. To do this you will need to write your own m-file. Call it magnitude.m. You need to make it into a MATLAB function, such that:

  m = magnitude(x,y)

  Now display the resulting image using

  show_image(m)

  You can also display this edge image after thresholding it,

  show_image(m>40)

  Create several of these with different thresholds.

  **QUESTION 1: What do you notice regarding the effect of changing the threshold?**

**TASK 2:**

- Now load up the Roberts operator. Repeat your previous exercise,

  **QUESTION 2: What do you notice regarding the difference between Sobel and Roberts?**

**TASK 3:**

- You now need to write a new function, that rather than taking the magnitude of the gradients, it takes the absolute value of |Gx|+|Gy|. This is an approximation to the magnitude.

  **QUESTION 3: What do you notice regarding the difference between magnitude and absolute when calculating the edge?**