

Comp3402 Unit1 Edge Detection

C.B.Price September 2021

Purpose	Edge detection is a fundamental image processing operation. It is useful in identifying objects in an image.
Files Required	Octave scripts in the zipped folder “Octave_ImProc_Release” on the module site.
ILO Contribution	Position Paper 1 (part 1) Learning Outcome 2.
Send to Me	
Homework	

Activities

1 Applying the Sobel edge detector to a synthetic image.

Here we shall use the script “NumFilterSobel” to apply the Sobel kernel to a synthetic image, both with and without noise. The image has a vertical brightness step, and you will get a plot of a horizontal row of pixels before and after the edge detector is applied.

(a) Run the script “**NumFilterSobel**” and choose (i) image size = 10, (ii) noise level = 0. Look at the output image row, you should clearly see the edge has been detected. In “flat” regions where there is no edge, the output should be 0.0. Type in **kernel** to print out the actual Sobel kernel used. Also type in **I** to see the original image and **I_f** to see the result of the Sobel operator.

You should see the operator correctly identifies the edge in the middle of the image. You will also see artefacts at the edges since the kernel cannot be applied to edge pixels.

(b) Now let’s have a look at some slices through the image as 1D plots. Run the script “**NumFilterSobel**” and choose (i) image size = 100, (ii) noise level = 0. Look at the output image row, you should clearly see the edge has been detected. In “flat” regions where there is no edge, the output should be 0.0. Type in **kernel** to print out the actual Sobel kernel used.

(b) (i) Repeat but now set the noise level to 0.01. You should see that the edge has been detected in the output image, but the original noise has also led to small amounts of edges being detected.

(ii) Now increase the noise level to 0.05. The Sobel edge detector will probably fail. So what do we conclude?

(iii) Could you suggest a pre-processing operation before applying the edge detector?

2 Applying the Sobel edge detector to images.

Run the script “**EdgeSobel**” which applies the Sobel operator to an original image, and the image with some added noise. Using the results of Activity 1, explain what you see. Feel free to investigate different images. Here the magnitude (size) of the edges is calculated, by

applying a horizontal Sobel and a vertical Sobel and combining these.

3 Smoothing before Edge Detection

You will have found that the Sobel edge detection operation is sensitive to noise. Therefore it makes sense to filter out as much noise as possible before applying the edge detection operator. This is the purpose of the script "**SmoothThenSobel**".

(a) Open up the script and look for the code which (i) adds noise to the original image, (ii) smooths the noisy image and (iii) applies the Sobel operator.

(b) Run the script using 9 for the width and 2 for sigma and compare the results with Activity 2.

(c) You may want to experiment with other filter widths and sigmas based on your previous investigation of Gaussian smoothing.
