

Lab 1 - Edge Detectors

Computational Vision & Imaging

Student ID: XXXXXXXX

Random Student, School of Computer Science,
University of Birmingham, UK, B15 2TT

Abstract—This document is a formal solution for the Lab 1 Formative Assignment based on the Edge Detection. It explains the difference between using *Sobel* and *Roberts* filters when identifying image edges as well as shows the impact of different thresholds and what happens when the magnitude function is approximated.

Index Terms—Lab 1, formative assignment, computer vision, imaging, edge detection, filters, kernels, Sobel, Robert, MATLAB.

I. INTRODUCTION

DURING the lectures we were shown that an image can be represented as a matrix of intensity values, a so called **intensity image**. By evaluating how the intensity changes throughout the image pixels, we are able to determine where the biggest variations occur, thus signaling where a potential edge can be present. Such technique is a powerful way to detect the region changes in the image while reducing the amount of data but preserving the structural properties.

II. SETUP

We are given an image "shakey.150.gif" (1) on which we need to apply edge detection filters.

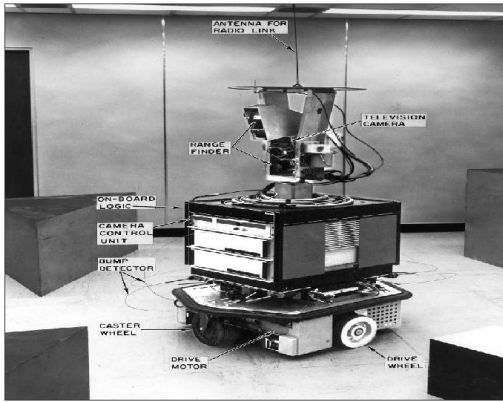


Fig. 1: grey-scale "Shakey 150" image

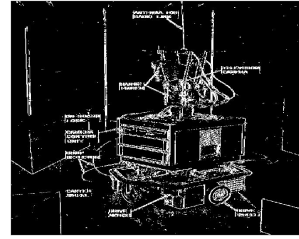
We are also given 2 MATLAB files "sobel.mat" and "roberts.mat" each containing horizontal and vertical weight matrices for calculating gradients at local regions in the image. The *Sobel* kernels can be seen in equation 1 and the *Roberts* kernels in equation 2.

This is a solution for the Formative Assignment by Hamid Dehghani
The report was submitted to Canvas on Feb 18 and reviewed by a TA

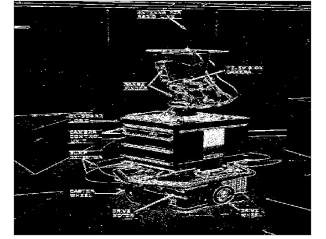
$$\text{SobelX} = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}; \text{SobelY} = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad (1)$$

$$\text{RobertsA} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}; \text{RobertsB} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \quad (2)$$

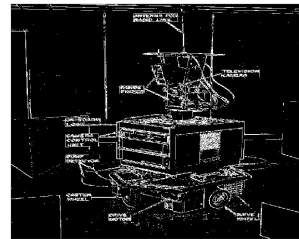
If we convolve the image with these filters (we already assume the filters are flipped, otherwise it is a *cross-correlation*), we get a different map for each filter. If we apply a threshold, we can see exactly which pixels "pass the criteria" for being considered as an edge. This can be seen in figure 2 where a threshold of 60 was used for *Sobel* maps and a threshold of 20 for *Roberts* maps.



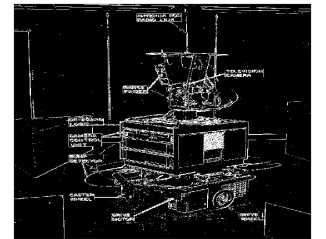
(a) Sobel horizontal



(b) Sobel vertical



(c) Roberts horizontal



(d) Roberts vertical

Fig. 2: filter maps after convolving edge kernels with the image

III. METHODOLOGY

Before we get into tasks, we need to define certain methods on which the experiments/findings depend on in section IV. We know that if we compute the gradient in x direction $\nabla_x G$ and in y direction $\nabla_y G$, we can compute the total magnitude by applying Pythagorean theorem. The acquired **intensity map** can be seen in figure 3 - *Sobel* map is more clear as it has higher weights towards the centre filter than *Roberts*.

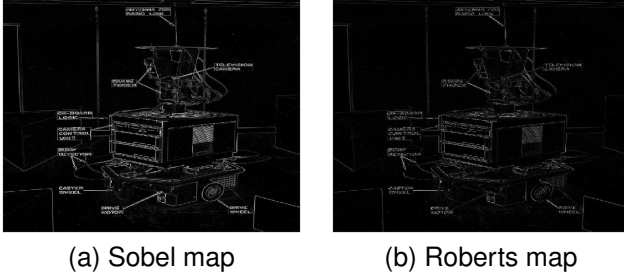


Fig. 3: non-thresholded magnitude maps

The computation for the exact magnitude can be seen in formula 3 but we can also approximate the magnitude using formula 4.

$$\text{magnitude} = \sqrt{\nabla_x G^2 + \nabla_y G^2} \quad (3)$$

$$\text{magnitude_approximation} = |\nabla_x G| + |\nabla_y G| \quad (4)$$

As well as computing the magnitudes, we will need to compute the absolute and the relative errors to compare how similar the approximate magnitude is to the exact magnitude. Absolute error can be found by equation 5 and the relative error by 6 if we compare true value x and predicted value \hat{x} .

$$\varepsilon_{\text{abs}} = |x - \hat{x}| \quad (5)$$

$$\varepsilon_{\text{rel}} = \frac{\varepsilon_{\text{abs}}}{|x|} \quad (6)$$

For matrices, we just compute the average over all entries.

IV. TASKS

A. Task 1

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

Answer 1: The smaller the threshold, the more noise and the more details there are in the image. The bigger the threshold, the less noise and the less details there are.

What happens with a low threshold is that we allow even relatively small values to be identified as image edges, and since at least some amount of the rate of change in intensity is present even in plain areas (due to noise), we get lots of pixels identified as edges. In contrast, with high threshold, we care only about the biggest changes in intensity, therefore, only the major contours are identified, leaving out most of the noise and minor edges undetected. In general, we can see a trade-off between image noise and contour smoothness.

This can be witnessed in figure 4 where 4 different *Sobel* magnitude maps show how increasing the threshold diminishes the high value (white) pixels in the image (i.e., edges or potential edges).

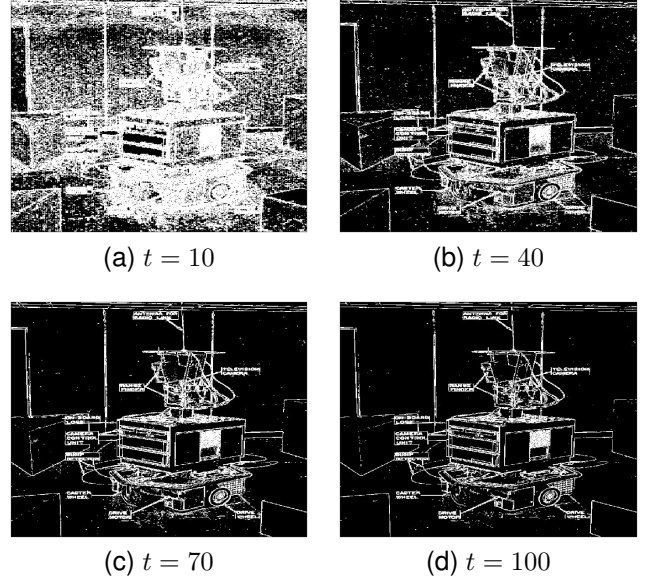


Fig. 4: Sobel magnitude maps after applying different thresholds

B. Task 2

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

Answer 2: Using Roberts filters, the contours are identified with relatively low thresholds, for example, with threshold 20, strong edges are detected and some small ones are even undetected, meaning 20 is slightly too high to capture small details. However, there is more noise too, so the trade-off is more overlapping compared to the trade-off when using Sobel filters.

The magnitude maps acquired by applying Roberts filter with different thresholds can be seen in figure 5.

As we can see, compared with Sobel maps, more noise is present even for higher threshold values. This happens because we compute the gradients at local regions of size 2-by-2 which is more sensitive than when we calculate gradients at 3-by-3 regions because we don't take into account the surrounding pixels which provide more generalized information about the rate of change of intensity in that region and smoothen the gradient.

C. Task 3

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

Answer 3: When using absolute values we introduce more noise because there is an approximation error (for example, in the case of using Roberts filters the relative error is 31.48%). However less time is taken to calculate the absolute measure than the exact magnitude (for example, in the case of using Sobel filters, the absolute measure is calculated 3-6 times faster than the magnitude). However, the result is still

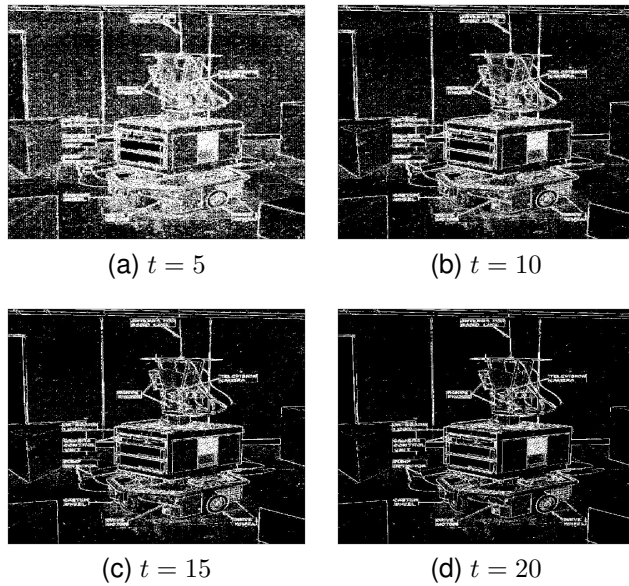


Fig. 5: Roberts magnitude maps after applying different thresholds

similar and, if efficiency is required, the absolute measure could be used as a substitute for the exact magnitude measure.

The example of maps acquired through magnitude approximation can be seen in figure 6. It is noticeable that the samples contain more noise than when computing the exact magnitudes, such as in figures 4, 5.

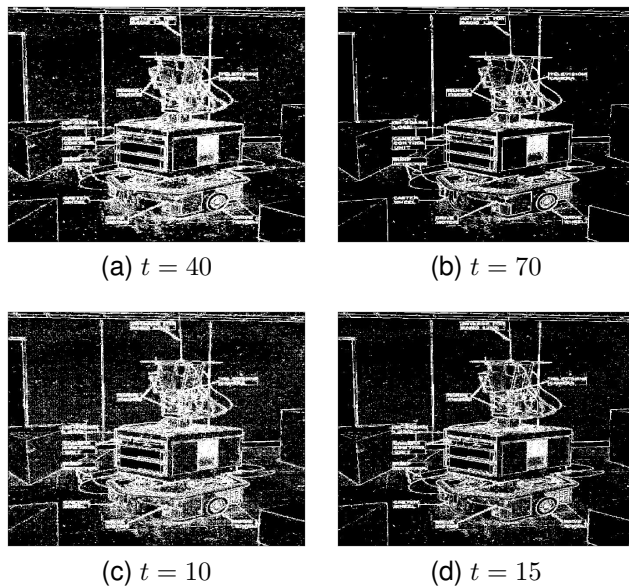


Fig. 6: Sobel (upper row) and Roberts (lower row) maps after applying magnitude approximation on different thresholds

REFERENCES

- [1] Sobel, Irwin. (2014). *An Isotropic 3x3 Image Gradient Operator*. Presentation at Stanford A.I. Project 1968.

- [2] LS. Davis, "A survey of edge detection techniques", *Computer Graphics and Image Processing*, vol 4, no. 3, pp 248-260, 1975.