Assignment 2: Edge Detection

Rami Saad and Raymond Lam

Section 4 and Section 2 (respectively)

October 21, 2019

Introduction

The purpose of this lab was to learn properties behind Edge Detection. Edge detection will often be the first stage in many image processing pipelines because image edges often correspond to the edges of actual objects[1]. This is not always the case but for the scope of this lab that assumption will be made.

The Canny Edge detector is actually composed of smaller modular functions. The purpose for the modular approach is repeatability; many of these functions will be re-utilized at a later time.

The lab was mainly conducted using Matlab and the C programming language. The combination of these two programming languages was achieved by using MEX functions which provide an interface between Matlab and C. Matlab was used as the scripting language for the project while the actual functions were implemented in C.

Theory

The Edge Detector pipeline is comprised as follows:

1. Find the horizontal and vertical image gradients.
2. Obtain the magnitude of the gradients (“strength” of an edge).
3. Post-process the gradients to remove weak or possibly incorrect edges.
4. Threshold the gradient image to obtain the final edge map.

The first step in any edge detector is to first process I(x, y) to find salient edges, by treating the image as a scalar field it allows us to define the notion of an “image derivative”, which is identical to the notion of partial derivatives that you have in vector calculus[1]. One type of derivative approximation filter is the Sobel operators:

$$h_x(x,y) = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \qquad h_y(x,y) = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

,

Then we find the magnitude of the gradients obtained through the Sobel spatial filtering:

$$||\nabla f(x,y)|| = \sqrt{(f_x(x,y))^2 + (f_y(x,y))^2}$$

Next, the image must be Post-processed using a non-maximum suppression filter. The non-maximum suppression filter will set all the values of a certain NxN window size to zero if they are not equal to the largest index in that particular window. For this lab, the NMS filter is run in two steps, first the Nx1 filter is run vertically and then a 1xN filter is run horizontally. This functionality can be expressed by this function:

$$NMS(w[n]) = \begin{cases} 0 & if w[n] \leq w_{max} \\ w[n] & otherwise \end{cases}$$

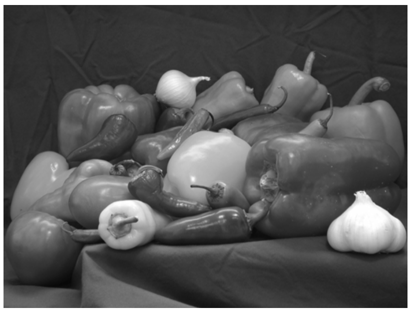
Finally, thresholding must be applied to the input image I(x,y) with the thresholding operator T, resulting in a binary image B(x,y). The expression takes the form:

$$B(x,y) = \begin{cases} 1 & I(x,y) \geq T \\ 0 & I(x,y) < T \end{cases}$$

Results

2.1.1

Q. 1


(a) Original Image

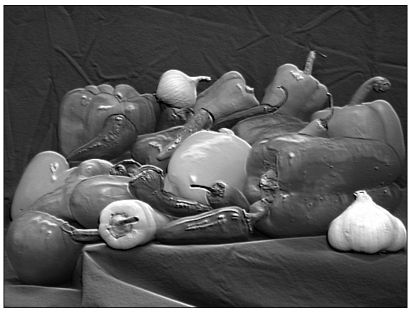

(b) Transformed Image

Figure 1: Sobel Spatial Filter

Q. 2

(a) Original Image

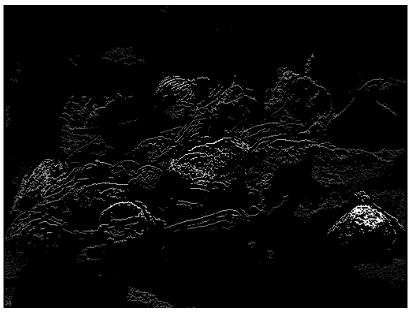

(b) Transformed Image

Figure 2: Non Maximum Suppression

Q. 3

(a) Original Image

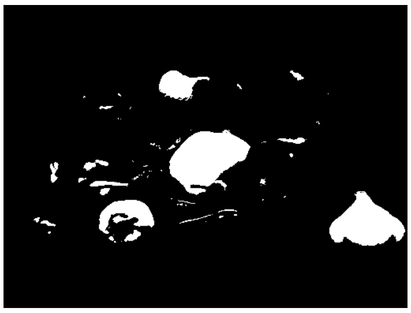
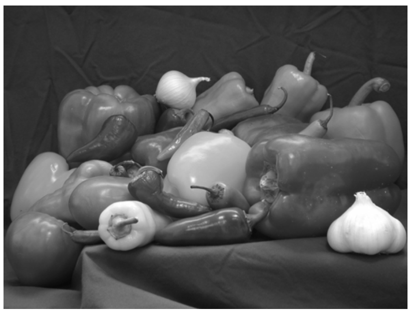

(b) Transformed Image

Figure 3: Thresholding Operations

Q. 4


(a) Original Image

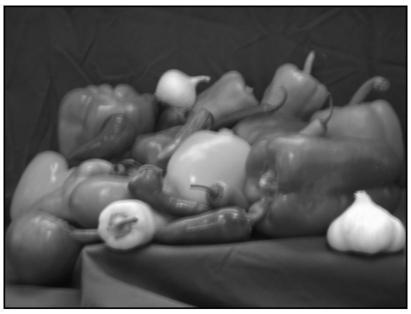

(b) Transformed Image

Figure 4: Gaussian blur Spatial Filter test

Q. 4 cont.

(a) Original Image

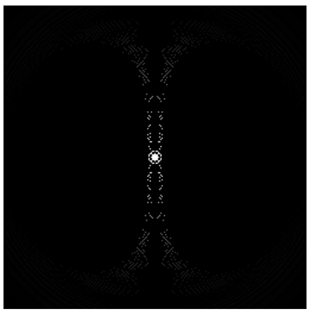
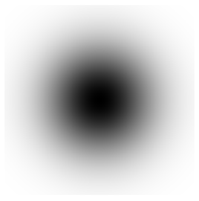
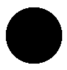

(b) Transformed Image

Figure 5: NMS filter test

Q. 4 cont


(a) Original Image


(b) T = 0.25

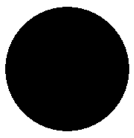
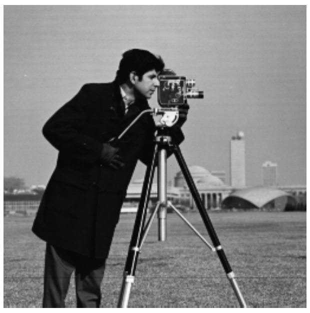

(c) T = 0.75

Figure 6: Thresholding Operation

2.2.1

Q. 1


(a) Original Image

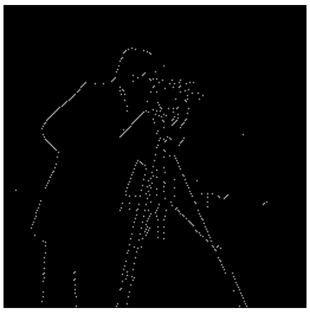
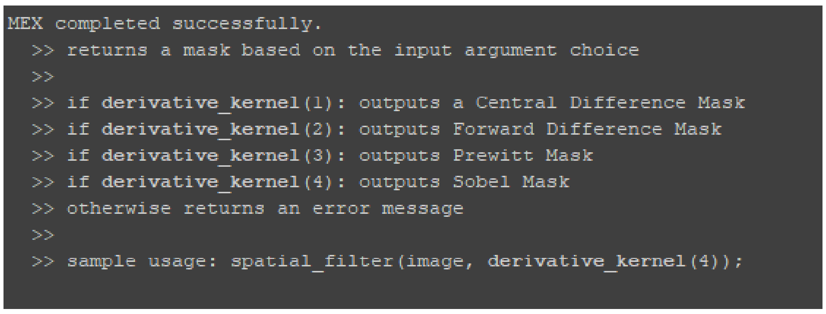

(b) Transformed Image

Figure 7: Complete Edge Detection Function

2.3.1

Q. 1



```
MEX completed successfully.
  >> returns a mask based on the input argument choice
  >>
  >> if derivative_kernel(1): outputs a Central Difference Mask
  >> if derivative_kernel(2): outputs Forward Difference Mask
  >> if derivative_kernel(3): outputs Prewitt Mask
  >> if derivative_kernel(4): outputs Sobel Mask
  >> otherwise returns an error message
  >>
  >> sample usage: spatial_filter(image, derivative_kernel(4));
```

(a) Help console output

Figure 8: derivative kernel help functionality

Analysis

Q. 1

Depending on the method of dealing with boundaries, values near the image boundaries will be truncated more heavily with NxN filter rather than two 1xN and Nx1 filters. Another aspect would be that computation time would be reduce for latter case. With a single 2D filter there are (M x N)(n x m) computations but using separable kernels will result in (M x N)(n + m) computations where M and N are the rows and columns of the image and m and n are the rows and columns of the filter.

Q. 2

There are a variety of advantages when implementing a NMS filter with two 1D arrays rather than 1 2D array. First, with two separate passes the magnitudes are obtained from the horizontal and vertical gradients and that information is reflected in the gradient magnitude. Secondly, This operation is not affected by the offset from the central pixel. It also lets us consider shapes other than square and rectangles when collecting the neighborhood around the current pixel.

Q. 3

Pre-filtering is an important part of edge detection because it allows for enhancement of desirable attributes of an image. In this lab the Sobel filter resulted in a more pronounced divergence between local areas of differing image intensities (edges). This allowed for edges to be more simple to locate in the rest of the edge detection pipeline.

Conclusion

In this lab, the process of creating a simple Cannery edge detector were accomplished. The spatial filter dealt with image boundaries by simply ignoring the outside edges. The non maximum suppression filter was by far the most challenging aspect to implement for this lab because of its requirement of having to traverse the 1D image array not only horizontally but vertically too. This compounded with the necessity of combining both NMS images back together made it very challenging to complete. Finally the thresholding filter was simple to implement as it only creates a binary image based on a threshold value T. While not perfect the edge detection is serviceable as demonstrated in figure (7b).

References

1. "Assignment 2 – Edge Detection." Assignment 2 – Edge Detection, 2016.

2. Gonzalez, Rafael C., and Richard E. Woods. Digital Image Processing. 4th ed. New York, NY: Pearson, 2008.