StyleSwap Assignment1

Akshat Sharma

May 14, 2024

1 Introduction

This assignment was based on edge detection by convolving a given digital image with predefined kernels(filters).
The most fundamental step in the edge detection algorithms is *convolution.* For this I have defined a method *'Convolve2'* which has as its parameters, two matrices, and returns the feature map obtained on convolving them. Using this, I have defined functions to implement various edge detection algorithms including the Sobel, Prewitt and Canny algorithms.

2 Analysis of Various kernels used

2.1 Gaussian kernel

Edge detection kernels such as Sobel and Laplacian, are quite sensitive to noise in images, since they compute the derivative (or second derivative) of the image signal. This is taken care of by convolving the image with the Gaussian kernel, which smoothens out the pixel values, thereby reducing noise. I have used the 3x3 Gaussian kernel :

$$\begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix}$$

2.2 Sobel kernels

The Sobel-x and Sobel-y kernels, when convolved with the image, give the derivative maps G_x and G_y, along x and y directions respectively. These operators perform two operations simultaneously:

1.Computation of derivative using finite difference approximation
2.Gaussian blurring

They are composed of the x and y derivative filters ($\begin{pmatrix} -1 & 0 & 1 \end{pmatrix}$ and $\begin{pmatrix} 1 \\ 0 \\ -1 \end{pmatrix}$) multiplied with the 1-D Gaussian filter in the other direction.

$$\text{Sobel-x} = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \qquad \text{Sobel-y} = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix}$$

The Sobel operator is widely used, but may struggle with noisy images. Explicit Gaussian blurring may be used to improve on this.

2.3 Prewitt kernels

To obtain the Prewitt-x and Prewitt-y kernels, the x and y derivative filters are multiplied with standard averaging filters.

$$\text{Prewitt-x} = \begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix} \qquad \text{Prewitt-y} = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{pmatrix}$$

The only difference with the Sobel operator, is therefore, that the equal weight is given to the central and neighbouring pixel differences. The two have similar computational complexity, accuracy and processing time.

2.4 Scharr kernels

Compared to the Sobel and Prewitt operators, the Scharr operator assigns higher weight to the central pixels in the kernel.

$$\text{Scharr-x} = \begin{pmatrix} -3 & 0 & 3 \\ -10 & 0 & 10 \\ -3 & 0 & 3 \end{pmatrix} \qquad \text{Scharr-y} = \begin{pmatrix} 3 & 10 & 3 \\ 0 & 0 & 0 \\ -3 & -10 & -3 \end{pmatrix}$$

This results in better noise insulation and detecting finer details, giving higher accuracy compared to Sobel and Prewitt operators. It is as fast as the Sobel operator but with slightly increased computational complexity.

3 Comparative study of various Edge detection algorithms

The various edge detection techniques I have employed for this assignment are:

1. Sobel edge detection
2. Gaussian blur combined with Sobel operator
3. Scharr edge detection
4. Prewitt edge detection
5. Canny edge detection
6. Laplacian method

The results of the Sobel, Gauss+Sobel, Scharr and Prewitt algorithms appear to be very similar, with decent accuracy.

The Canny edge detection, involves multiple steps including Gaussian blurring, Sobel filtering, non-maximum suppression, double thresholding, and edge tracking by hysteresis. Non-maximum suppression comprises determining the direction of gradient at each pixel(using the angle matrix), and of the neighbouring pixels at that coordinate in the same direction, preserving only the pixel with the highest intensity.

This is followed by double thresholding, i.e. setting two threshold intensity values, *highThreshold* and *lowThreshold*, and using these, classifying the pixels into three categories(where I is the pixel intensity):

1. Strong edges, if $I > highThreshold$
2. Weak edges, if $highThreshold > I > lowThreshold$
3. Non edges, if $lowThreshold > I$

The pixels values at strong edges are then set to 255, those at weak edges to a lower intensity, and the non edges to zero. Edge tracking is then performed, by giving nonzero value to pixels in the neighbourhood of strong edges. As a result, in the Canny edge-detected image, only the most prominent edges are highlighted. Edges are well-defined and continuous.

The final method used involved obtaining the Laplacian of the image(the second derivative map) by convolving with the Laplacian kernel. The computation of second order derivative is particularly sensitive to noise; therefore Gaussian blurring was used before finding the Laplacian.