Aynur Süne - 21041

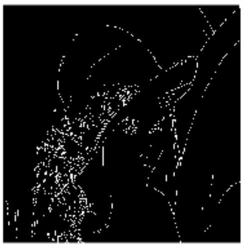
EE 417 – Assignment #1

Comparison of Edge Detectors

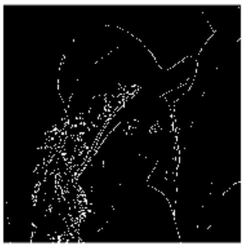
Edges happen when pixel values of an image undergo sudden variation. To detect edges in an image, various methods can be applied and each will give different solution. In this assignment, Prewitt, Roberts and Canny edge detectors are applied as 1st Derivative Edge Detectors; and Laplacian of Gaussian (LoG) is applied as 2nd Derivative Edge Detector. Outputs of three different images are as follows:

original image

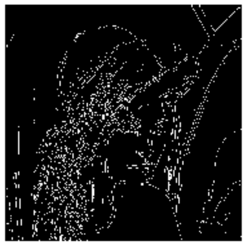
canny



prewitt

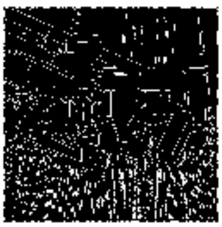


roberts



log

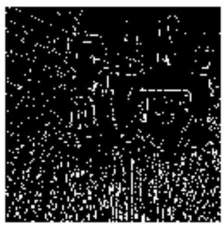
original image



canny

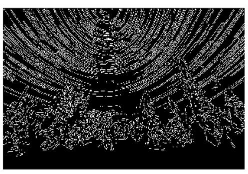
prewitt

roberts

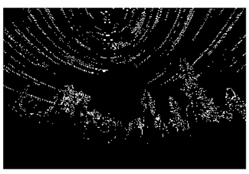


log

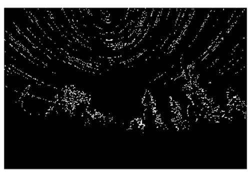
original image



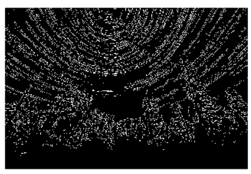
canny



prewitt



roberts



log

Canny Edge Detection Algorithm

For best filter, Canny had some performance criteria. First one is ‘Good Detection’, Canny argues that the filter must have a stronger response at the edge location than the noise. Although a smoothing operation is applied, there may still be noisy pixels in an image. Filter must be able to detect an edge among noisy pixels. Second criteria is ‘Good Localization’. The filter response must be maximum very close to x=0. It should be able to detect real local points. Last one is ‘Single Response Constraint’. There should be only one maximum in a reasonable neighborhood of x=0. Filter should be able to detect real maximum one among local points. Canny found a filter that maximizes these three performance criteria and this looks very similar to first derivative of Gaussian.

Canny Edge detection algorithm applies Gaussian Filter first, to smooth the image and reducing noise. Then, image gradient is calculated and edge strengths are estimated. Higher strength implies an edge. Local maxima of strengths must be found. But there may be other large values around maximum point so they are counted as an edge as well. This leads a thicker edge than it should be. But as 3rd criteria states, there should be only one maximum. To detect the highest value, Canny applied a technique called ‘Non-Max Suppression’. This technique suppressed values smaller than its neighbours, eliminated points smaller than real maximum strength value. This resulted thinner edges. These thinned edges are later thresholded. If the threshold is too high, there can be misdetections. If threshold is too low, many of the pixels appear as an edge, causing spurious edges. For more accuracy, Canny selected two different threshold values, low and high threshold. If the strength of an edge is higher than high threshold, it is counted as an edge; if lower than low threshold, it is not counted as an edge and discarded. If an edge is between low and high threshold, it is called a weak edge. If a weak edge is adjacent to strong edges, it is included to edge list. This thresholding is called Hysteresis Thresholding.

Prewitt Edge Detection Algorithm

To compute gradient value of an image, Prewitt uses two image which are gradient approximations of the derivatives obtained from convolution of the original image and two different kernels. Kernels are different and one is used for Vertical Edge Detection and another is for Horizontal Edge Detection.

-1	0	1
-1	0	1
-1	0	1

Noise Smoothing

Vertical Edge Detection

-1	-1	-1
0	0	0
1	1	1

Horizontal Edge Detection

Noise Smoothing

Formula for gradient magnitudes is as follows:

$$\mathbf{G} = \sqrt{{\mathbf{G}_x}^2 + {\mathbf{G}_y}^2}$$

One threshold value is used to detect high gradient values of the formula's output as an edge.

Roberts Edge Detection Algorithm

Roberts edge detection algorithm operates same as Prewitt, only difference is the kernels. Kernels used for convolution is as follows:

$$\begin{bmatrix} +1 & 0 \\ 0 & -1 \end{bmatrix} \qquad \begin{bmatrix} 0 & +1 \\ -1 & 0 \end{bmatrix}.$$

Laplacian of Gaussian

Laplacian of Gaussian operation first smooths the image by applying Gaussian Filter because derivative is sensitive to noise. In second derivative, zero crossings imply an edge. Peaks of the first derivative corresponds to zero crossings of the second derivative. But second derivative which is equal to 0 is not enough. Second derivative has to change sign. To find zero crossings, Laplacian (in this case second derivative) is applied after Gaussian.

Output of different images indicates that Canny Edge Detection is better on all of them. It can detect edges better than other algorithms. Setting two different thresholds is really precise for finding the edges. Canny is able to find the edges that other edge operators are not able to. It gives the general outline of the images better than other operators. Laplacian of Gaussian gives good results as well, but since it is a second derivation operation and derivation is sensitive to noise, although image is smoothed by Gaussian filter, usually there are still noisy pixels and these noisy pixels are detected as an edge as well. Prewitt and Roberts operators does not have smoothing operations and only one threshold is used for detecting edges. These cause thicker edges and some noisy pixels detected as an edge as well. Less edge found with Robert and Prewitt due to thresholding system when compared to Canny. Also Prewitt is better at finding edges more than Robert where main difference of these operators are only kernels used. In this case, bigger kernel gives better result but it is also expected to be computationally slower than Robert, but when matrix sizes are not very large, computation time is very close so this difference can be neglected. To conclude, Canny Edge Detection Algorithm gave best results among others, Laplacian of Gaussian was slightly worse than Canny because of some undetected edges and noise but it deserves second row because despite second derivative issues it gave better results than Roberts and Prewitt. Among Roberts and Prewitt, Prewitt gave better outputs than Roberts so it should be the 3rd best operator. And the worst among all was the Roberts operator. It can be observed that Prewitt and Roberts did not give any details about the images, especially Roberts.