A Study of Edge Detection Methods and Providing a New Simple Algorithm for Edge Detection of Noisy Images

Amin Ghasemi Nejad Raeini, Mehran Abdali, Zoheir KordRostami

Department of Electronics, Sirjan Science and Research Branch, Islamic Azad University, Sirjan, Iran
amin.gh1392@yahoo.com

Department of Electronics, Sirjan Science and Research Branch, Islamic Azad University, Sirjan, Iran

Department of Electrical and Electronic Engineering, Shiraz University of Technology, Shiraz, Iran

Abstract

Noisy image edge detection is one of the major areas of work in image processing. Edge detection of the noise-free images is much easier than finding the edge of noisy images. But in practice, many of the images are imbued with noise. This paper will present a new algorithm based on morphology operation to eliminate noise and perform correct edge detection simultaneously. The algorithm is applied on several images and the results are observable.

Keywords: Morphology Operation, Noisy Image, Edge Detection, Structure Element.

I. Introduction

The quality of edge detection is highly dependent on lighting conditions, the presence of objects of similar intensities, density of edges in the scene. Since different edge detectors work better under different conditions, the objective of this paper is to identity the suitable edge detector for images with the noise present in it (Anurag et al., 2012).

The study begins by taking images, and then to add salt and pepper noise. Edge detector operators are applied to all the noisy images. The main problem is that different edge detectors work differently. Some takes more time with respect to other, while some finds more edges (works deeply) with respect to other. The detection of edges in an image depends upon illumination, blur, noise, intensity, objects.

The actual difference in working of various edge detectors can be analyzed by using these different algorithms in a same program or system. That is one of the goal of our project. We tested five edge detectors that use different methods for detecting edges and compared their results by a new Algorithm to determine which detector works better for different Noisy images. We have observed in many theoretical and practical environments that Sobel Operator is better than Roberts and Prewitt operator. So in this paper we compare our Proposed model with Sobel Operator, Laplacian of Gaussian edge detection method, Prewitt edge detection method, Prewitt
edge detection method, Canny edge detection method and Edge detection using mathematical morphology operations.

The classical operator such as Sobel, and Prewitt which uses first derivative has very simple calculation to detect the edges and their orientations but has inaccurate detection sensitivity in case of noise. Laplacian of Gaussian (LOG) operator is represented as another type of edge detection operator which uses second derivative. It finds the correct places of edges and testing wider area around the pixel. The disadvantages of LOG operator is that it cannot find the orientation of edge because of using the Laplacian filter. The other type of edge detection operator is the Gaussian edge detectors such as Canny, which is using probability for finding error rate and localization. Also it is symmetric along the edge and reduces the noise by smoothing the image. Canny method can produce good edge with the smooth continuous pixels and thin edge. Sobel edge detection method cannot produce smooth and thin edge compared to canny method. So it performs the better detection in noise condition but unfortunately it has complex computing (Beant Kaur et al., 2010).

In the mean time we have applied our Proposed Operator to detect the edges which gives better results than other operators. We can make decision by observing the subjective and object comparisons that our Proposed Operator is optimal. By doing comparison between traditional and morphological operators result, we come to know that the result of applying proposed model is better than all. So, it can be used in medical for better results because resultant output has continuous edges as compared to traditional operator’s results. The main advantages of mathematical morphology are direct geometric interpretation, simplicity and efficiency in hardware implementation.

II. REVIEW THE OPERATIONS OF MATHEMATICAL MORPHOLOGY

Mathematical morphology uses structuring element, which is characteristic of certain structure and feature, to measure the shape of image and then carry out image processing. Based on set theory, mathematical morphology is the operation which transforms from one set to another (Beant Kaur et al., 2010).

The basic mathematical morphological operators are dilation and erosion and the other morphological operations are the synthesization of the two basic operations. In the following, we introduce some basic mathematical morphological operators of grey-scale images.

Morphological techniques probe an image with a small shape or template called a structuring element.

The structuring element is positioned at all possible locations in the image and it is compared with the corresponding neighborhood of pixels. Some operations test whether the element "fits" within the neighborhood, while others test whether it "hits" or intersects the neighborhood. A structuring element is simply a binary image that allows us to define arbitrary neighborhood structures. The structuring element is a small binary image, i.e. a small matrix of pixels, each with a value of zero or one. The matrix dimensions specify the size of the structuring element. The pattern of ones and zeros specifies the shape of the structuring element. An origin of the structuring element is usually one of its pixels, although generally the origin can be outside the structuring element (Rama Bai et al., 2010).

The dilation process is performed by laying the structuring element SE(s,t) on the image F(x,y) and sliding it across the image in a manner similar to convolution. The difference is in the operation performed. The different steps of dilation are:
1. If the origin of the structuring element coincides with a white pixel in the image, there is no change, and move to the next pixel.

2. If the origin of the structuring element coincides with a black pixel in the image, and at least one of the black pixels in the structuring element falls over a white pixel in the image, then change the black pixel in the image (corresponding to the position on which the center of the structuring element falls) from black to a white.

Erosion of a grey-scale image \( F(x,y) \) by a grey-scale structuring element \( SE(s,t) \) is denoted by

\[
(F \Theta SE)(x,y) = \min\{ F(x + s, y + t) - SE(s, t) \}
\]  

These two basic operations, dilation and erosion, can be combined into more complex sequences. The most useful of these for morphological filtering are called opening and closing. Opening consists of an erosion followed by a dilation and can be used to eliminate all pixels in regions that are too small to contain the structuring element. In this case the structuring element is often called a probe, because it is probing the image looking for small objects to filter out of the image. Opening and closing of grey-scale image \( F(x,y) \) by grey-scale structuring element \( SE(s,t) \) are denoted respectively by

\[
A \circ SE = (A \Theta SE) \oplus SE
\]

\[
A \bullet SE = (A \oplus SE) \Theta SE
\]

Erosion is a transformation of shrinking, which decreases the grey-scale value of the image, while dilation is a transformation of expanding, which increases the grey-scale value of the image. But both of them are sensitive to the image edge whose grey-scale value changes obviously. Erosion filters the inner image while dilation filters the outer image. Opening is erosion followed by dilation and closing is dilation followed by erosion. Opening generally smoothes the contour of an image, breaks narrow gaps. As opposed to opening, closing tends to fuse narrow breaks, eliminates small holes, and fills gaps in the contours. Therefore, morphological operation is used to detect image edge, and at the same time, denoise the image (Rama Bai et al., 2010).

**III. Review of common edge-detection methods**

**A. Edge detection using mathematical morphology operations**

The operation’s mathematical relationship is as follows:

\[
(Picture \oplus Strel) - (Picture \Theta Strel)
\]  

A Structure Element (Strel) which varies according to the type of image will be used in this operation. The operation is performed for edge detection and recovery of image features and the extracted features are dependent on the type and size of Structure Element. Dilation and Erosion are the operators of the operation.

The combination of these operators is also used in other morphology operations. Erosion reduces the amount of pixels and makes the objects in the image become thinner and shorter. While Dilation is an operation in which the amount of image pixels increases and the image becomes thicker. The subtraction operation is used to retrieve the image features.

**B. Sobel edge detection method**

The operator consists of a pair of 3×3 convolution kernels as shown in Figure 1. One kernel is simply the other rotated by 90°.
These kernels are designed to respond maximally to edges running vertically and horizontally relative to the pixel grid, one kernel for each of the two perpendicular orientations. The kernels can be applied separately to the input image, to produce separate measurements of the gradient component in each orientation (call these $G_x$ and $G_y$). These can then be combined together to find the absolute magnitude of the gradient at each point and the orientation of that gradient. The gradient magnitude is given by (Vineet Saini et al., 2012):

$$|G| = \sqrt{G_x^2 + G_y^2}$$  \hspace{1cm} (5)

Typically, an approximate magnitude is computed using:

$$|G| = |G_x| + |G_y|$$  \hspace{1cm} (6)

which is much faster to compute. The angle of orientation of the edge (relative to the pixel grid) giving rise to the spatial gradient is given by:

$$\theta = \arctan(G_x / G_y)$$  \hspace{1cm} (7)

In this method the edges are recognized using the Sobel approximation as derivatives (Rama Bai et al., 2010).

\begin{table}[h]
\centering
\begin{tabular}{|c|c|c|c|}
\hline
Masks Used in Sobel Operator & Gx & Gy \\
\hline
-1 & 0 & 1 & 1 & 2 & 1 \\
-2 & 0 & 2 & 0 & 0 & 0 \\
-1 & 0 & 1 & -1 & -2 & -1 \\
\hline
\end{tabular}
\end{table}

C. Canny edge detection method

The Canny edge detection algorithm is known to many as the optimal edge detector. Canny's intentions were to enhance the many edge detectors already out at the time he started his work. He was very successful in achieving his goal and his ideas and methods can be found in his paper, "A Computational Approach to Edge Detection". In his paper, he followed a list of criteria to improve current methods of edge detection. The first and most obvious is low error rate. It is important that edges occurring in images should not be missed and that there be NO responses to non-edges. The second criterion is that the edge points be well localized. In other words, the distance between the edge pixels as found by the detector and the actual edge is to be at a minimum. A third criterion is to have only one response to a single edge. This was implemented because the first 2 were not substantial enough to completely eliminate the possibility of multiple responses to an edge.

Based on these criteria, the canny edge detector first smoothes the image to eliminate and noise. It then finds the image gradient to highlight regions with high spatial derivatives. The algorithm then tracks along these regions and suppresses any pixel that is not at the maximum (nonmaximum suppression). The gradient array is now further reduced by hysteresis. Hysteresis is used to track along the remaining pixels that have not been suppressed. Hysteresis uses two thresholds and if the magnitude is below the first threshold, it is set to zero (made a nonedge). If
the magnitude is above the high threshold, it is made an edge. And if the magnitude is between the 2 thresholds, then it is set to zero unless there is a path from this pixel to a pixel with a gradient above T2.

In order to implement the canny edge detector algorithm, a series of steps must be followed. The first step is to filter out any noise in the original image before trying to locate and detect any edges. And because the Gaussian filter can be computed using a simple mask, it is used exclusively in the Canny algorithm. Once a suitable mask has been calculated, the Gaussian smoothing can be performed using standard convolution methods (Rama Bai et al., 2010). A convolution mask is usually much smaller than the actual image. As a result, the mask is slid over the image, manipulating a square of pixels at a time. The larger the width of the Gaussian mask, the lower is the detector's sensitivity to noise. The localization error in the detected edges also increases slightly as the Gaussian width is increased. The Gaussian mask used in my implementation is shown below:

<table>
<thead>
<tr>
<th></th>
<th>2</th>
<th>4</th>
<th>5</th>
<th>4</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>9</td>
<td>12</td>
<td>9</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>12</td>
<td>15</td>
<td>12</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>9</td>
<td>12</td>
<td>9</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>5</td>
<td>4</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

After smoothing the image and eliminating the noise, the next step is to find the edge strength by taking the gradient of the image. The Sobel operator performs a 2-D spatial gradient measurement on an image. Then, the approximate absolute gradient magnitude (edge strength) at each point can be found. The Sobel operator uses a pair of 3x3 convolution masks, one estimating the gradient in the x-direction (columns) and the other estimating the gradient in the y-direction (rows).

The magnitude, or edge strength, of the gradient is then approximated using the formula:

\[ |G| = |G_x| + |G_y| \]  \hspace{1cm} (8)

The direction of the edge is computed using the gradient in the x and y directions. However, an error will be generated when sum X is equal to zero. So in the code there has to be a restriction set whenever this takes place. Whenever the gradient in the x direction is equal to zero, the edge direction has to be equal to 90 degrees or 0 degrees, depending on what the value of the gradient in the y-direction is equal to. If GY has a value of zero, the edge direction will equal 0 degrees. Otherwise the edge direction will equal 90 degrees. The formula for finding the edge direction is just:

\[ \theta = \arctan(G_x/G_y) \]  \hspace{1cm} (9)

Once the edge direction is known, the next step is to relate the edge direction to a direction that can be traced in an image. So if the pixels of a 5x5 image are aligned as follows:
Then, it can be seen by looking at pixel "a", there are only four possible directions when describing the surrounding pixels - 0 degrees (in the horizontal direction), 45 degrees (along the positive diagonal), 90 degrees (in the vertical direction), or 135 degrees (along the negative diagonal). So now the edge orientation has to be resolved into one of these four directions depending on which direction it is closest to (e.g. if the orientation angle is found to be 3 degrees, make it zero degrees). Think of this as taking a semicircle and dividing it into 5 regions.

![Figure 1. A Semicircle That Shown 5 Regions](image)

Therefore, any edge direction falling within the yellow range (0 to 22.5 & 157.5 to 180 degrees) is set to 0 degrees. Any edge direction falling in the green range (22.5 to 67.5 degrees) is set to 45 degrees. Any edge direction falling in the blue range (67.5 to 112.5 degrees) is set to 90 degrees. And finally, any edge direction falling within the red range (112.5 to 157.5 degrees) is set to 135 degrees.

After the edge directions are known, non maximum suppression now has to be applied. Non maximum suppression is used to trace along the edge in the edge direction and suppress any pixel value (sets it equal to 0) that is not considered to be an edge. This will give a thin line in the output image.

Finally, hysteresis is used as a means of eliminating streaking. Streaking is the breaking up of an edge contour caused by the operator output fluctuating above and below the threshold. If a single threshold, T1 is applied to an image, and an edge has an average strength equal to T1, then due to noise, there will be instances where the edge dips below the threshold. Equally it will also extend above the threshold making an edge look like a dashed line. To avoid this, hysteresis uses 2 thresholds, a high and a low. Any pixel in the image that has a value greater than T1 is presumed to be an edge pixel, and is marked as such immediately. Then, any pixels that are connected to this edge pixel and that have a value greater than T2 are also selected as edge pixels. If you think of following an edge, you need a gradient of T2 to start but you don't stop till you hit a gradient below T1.

In this method the edges are recognized through finding a local maximum of gradient F(x,y).

This method uses two thresholds to detect strong and weak edges and the weak edges which are connected to the strong edges, are considered in the output.
D. Prewitt edge detection method

Prewitt edge detector uses the masks displayed below to digitally estimate the first order derivatives. Implementation of the Prewitt method is slightly easier than the Sobel method, but its results are noisier. Prewitt operator is similar to the Sobel operator and is used for detecting vertical and horizontal edges in images.

Table III

<table>
<thead>
<tr>
<th>Prewitt Operator</th>
</tr>
</thead>
</table>
| h1 = \[
  \begin{bmatrix}
  1 & 1 & 1 \\
  0 & 0 & 0 \\
 -1 & -1 & -1 \\
  \end{bmatrix}
\] |
| h2 = \[
  \begin{bmatrix}
 -1 & 0 & 1 \\
 -1 & 0 & 1 \\
 -1 & 0 & 1 \\
  \end{bmatrix}
\] |

The Prewitt edge detector is a gradient based edge detector. The detector is considered to be poor due to its bad approximation to the gradient operator. However, the ease of implementation and low computational cost overcome these disadvantages. Figure 3 shows Prewitt mask suitable for HIP framework using spiral addressing scheme.

E. Laplacian of Gaussian edge detection method

In this method after filtering F(x,y) using a Gaussian filter, the edges are detected by finding zero crossing. Filtering an image with \( \nabla^2 h(r) \) is equivalent to filtering it using the Smoothing function and then computing its Laplace. Thus filtering an image with \( \nabla^2 h(r) \) has two effects. First it makes the image become Smooth, so it reduces noise; Second, by calculating the Laplace of binary edges the image becomes recognizable (Rama Bai et al., 2010). The Laplacian is a 2-D isotropic measure of the 2nd spatial derivative of an image. The Laplacian of an image highlights regions of rapid intensity change and is therefore often used for edge detection. The Laplacian is often applied to an image that has first been smoothed with something approximating a Gaussian Smoothing filter in order to reduce its sensitivity to noise. The operator normally takes a single graylevel image as input and produces another graylevel image as output. The Laplacian L(x,y) of an image with pixel intensity values I(x,y) is given by:

\[
L(x, y) = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}
\]  

(10)

Since the input image is represented as a set of discrete pixels, we have to find a discrete convolution kernel that can approximate the second derivatives in the definition of the Laplacian. Three commonly used small kernels are shown in Table 4.
Because these kernels are approximating a second derivative measurement on the image, they are very sensitive to noise. To counter this, the image is often Gaussian Smoothed before applying the Laplacian filter. This pre-processing step reduces the high frequency noise components prior to the differentiation step (Gonzalez et al., 2008).

In fact, since the convolution operation is associative, we can convolve the Gaussian smoothing filter with the Laplacian filter first of all, and then convolve this hybrid filter with the image to achieve the required result. Doing things this way has two advantages:

- Since both the Gaussian and the Laplacian kernels are usually much smaller than the image, this method usually requires far fewer arithmetic operations.
- The LoG ('Laplacian of Gaussian') kernel can be precalculated in advance so only one convolution needs to be performed at run-time on the image.

The 2-D LoG function centered on zero and with Gaussian standard deviation $\sigma$ has the form (Vineet Saini et al., 2012):

$$LOG(x, y) = -\frac{1}{\pi\sigma} \left[ 1 - \frac{x^2 + y^2}{2\sigma^2} \right] e^{-\frac{x^2 + y^2}{2\sigma^2}}$$  \hspace{1cm} (11)

and is shown in Fig 2.

![Figure 1. 2-D LOG Function](image)

<table>
<thead>
<tr>
<th>Table IV</th>
<th>Three Commonly Used Small Kernels</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 1 0</td>
<td>1 1 1</td>
</tr>
<tr>
<td>1 -4 1</td>
<td>1 -8 1</td>
</tr>
<tr>
<td>0 1 0</td>
<td>1 1 1</td>
</tr>
</tbody>
</table>

Note that as the Gaussian is made increasingly narrow, the LoG kernel becomes the same as the simple Laplacian kernels shown in Figure 1. This is because smoothing with a very narrow
Gaussian ($\sigma < 0.5$ pixels) on a discrete grid has no effect. Hence on a discrete grid, the simple Laplacian can be seen as a limiting case of the LoG for narrow Gaussians.

IV. The new algorithm

The morphology operation uses the Dilation and Erosion operators for better retrieval of the edges (Gonzalez et al., 2008). These two operators are not appropriate for eliminating the noises, so the Opening and Closing operators are first used to eliminate noise and finally the other two operators are used for retrieval and better identification of the edges.

In performing morphology operation, choosing a Structure Element which is used for better edge detection and a better display of the edges, is very important.

This algorithm uses a $3 \times 3$ square Structure Element.

The algorithm is as follows:

$$F = (\{(I \circ SE) \cdot SE\} \bigoplus SE) - (\{(I \circ SE) \cdot SE\} \bigoplus SE)$$

- $F$: Final Image
- $I$: Initial Image
- SE: Structure Element

In the presented algorithm, the Opening operator is first used to remove the salt noise, and then the Closing operator is applied to eliminate the pepper noise.

Dilation operator has been used to highlight details and to better display the edges, and finally the subtraction operation is utilized to retrieve the edges.

V. Results and Analysis

In this section the results of conventional methods of edge detection on an image including pepper - salt noise is shown and the problems with these methods in edge detection of a noisy image are specified and the edge detection results using the proposed algorithm are also observable.
VI. Conclusions

Edge detection has become a crucial step for detecting a correct object of an Image. We have observed in many theoretical and practical environments that Sobel Operator is better than Roberts and Prewitt operator. So in this paper we compare our Proposed model with Sobel Operator, Laplacian of Gaussian edge detection method, Prewitt edge detection method, Prewitt edge detection method, Canny edge detection method and Edge detection using mathematical morphology operations.
The classical operator such as Sobel, and Prewitt which uses first derivative has very simple calculation to detect the edges and their orientations but has inaccurate detection sensitivity in case of noise. Laplacian of Gaussian (LOG) operator is represented as another type of edge detection operator which uses second derivative. It finds the correct places of edges and testing wider area around the pixel. The disadvantages of LOG operator is that it cannot find the orientation of edge because of using the Laplacian filter. The other type of edge detection operator is the Gaussian edge detectors such as Canny, which is using probability for finding error rate and localization. Also it is symmetric along the edge and reduces the noise by smoothing the image. Canny method can produce good edge with the smooth continuous pixels and thin edge. Sobel edge detection method cannot produce smooth and thin edge compared to canny method. So it performs the better detection in noise condition but unfortunately it has complex computing.

In the mean time we have applied our Proposed Operator to detect the edges which gives better results than other operators. We can make decision by observing the subjective and object comparisons that our Proposed Operator is optimal. By doing comparison between traditional and morphological operators result, we come to know that the result of applying proposed model is better than all. So, it can be used in medical for better results because resultant output has continuous edges as compared to traditional operator’s results. The main advantages of mathematical morphology are direct geometric interpretation, simplicity and efficiency in hardware implementation.

Acknowledgements

Special thanks to Dr Mehran Abdali and Dr Zoheir KordRostami who helped me in writing this paper.

References
