IMPROVED EDGE DETECTION USING VARIABLE THRESHOLDING TECHNIQUE AND CONVOLUTION OF GABOR WITH GAUSSIAN FILTERS

Isaack Adidas Kamanga

Department of Electronics and Telecommunications engineering,
Dar es Salaam Institute of Technology (DIT), Dar es Salaam, Tanzania

ABSTRACT

Medical Field, Robotic vision, Pattern recognition, Hurdle detection, and smart city are examples of areas that require image processing to achieve automation. Detecting an edge is an important stage in any computer vision application. The performance of the edge detecting algorithm is largely affected by the noise present in an image. An Image with a low signal-to-noise ratio (SNR), imposes a challenge to locate its edges. To improve the observable image boundaries, an adaptive filtering technique is proposed in this article. The proposed algorithm uses convolution of Gabor filter with Gaussian (GoG) operator to clean the noise before non-Maxima suppression. Furthermore, using variable hysteresis thresholding can further improve edge locating. The implementation of the algorithm was done by Python and Matlab. The obtained results were compared to a number of reviewed algorithms such as the Canny method, Laplacian of Gaussian, The Marr-Hildreth method, Sobel operator, and the Haar wavelet-based method. Three performance factors were used; PNSR, MSE, and processing time. The simulation result shows that the proposed method has higher PNSR, lower MSE, and shorter processing time when compared to the Canny detector, the Marr-Hildreth, Haar wavelet-based, Laplacian of Gaussian, and the Sobel operator methods. The higher PNSR, lower MSE, and shorter processing time mean improved edge details of the processed image.

KEYWORDS

Edge, Edge locating, Filtering, Gabor filter, Python programming.

1. INTRODUCTION

Image processing is a kind of signal processing in which the input is the image and the output can be either an image or a collection of features or characteristics of an input image. Image processing is one of the main steps in the medical field, robotic vision, pattern recognition, Hurdle detection, and smart city [1], [2], [3]. We may perform image processing so as to enhance it for human interpretation, to enable information to be extracted for machine vision and interpretation [4], [5]. Image processing is also done to restore the image details and achieve pattern recognition [6]. Image processing can either be analog or digital. Digital image processing involves the use of computer-written algorithms to process it to get an enhanced version of it to extract important features for a given application. Digital image processing involves several stages. The stages are image acquisition, enhancement, transformation, morphological processing, segmentation, and finally representation and description [7]. The acquisition is converting light to digital images. Enhancement means improving an image by suppressing the noise and making the important features more observable. Transformation is the representation of an image into a format such as grayscale for ease of further processing. Morphological processing is the process of extracting image components necessary for shape representation. Segmentation involves separating a part or parts of an image that is useful for further processing. Segmentation is a difficult process in digital image processing. To successfully separate these parts from the rest of the image, we need to clearly define the boundaries of an image. An image boundary is often known as an edge [8]. An edge is a boundary separating two objects in an image. It is obtained by determining the continuous line of pixels separating two different brightness [9]. According to [10] and [11], An edge is a curve of pixel positions where a sudden change of brightness occurs.

To locate an edge, a number of methods can be used. Robert operator, Sobel operator, Prewitt, and Canny are the basic methods for edge detection. Out of these methods, Canny is optimal. However, Canny faces four main challenges. First, the Gaussian filtering applied in the Canny method may blur the weak edges out causing them to be undetected. Secondly, Canny uses the same-sized sigma leading to missing weak edges. Thirdly, the Canny method involves very complex computations, this wastes time, therefore, is not a good approach in the application that requires real-time responses such as robotic vision systems and car plate number identification in a smart city. The Canny approach has another problem, the pixels at corners may appear in the wrong directions for their neighbors leading to the missing junction. This problem has been solved by the SUSAN edge detector [12]. The remaining issues present a research opportunity to enhance the algorithm.

In this paper, a fast algorithm based on the variable thresholding technique, the application of the convolution of Gaussian with Gabor filters, and the use of different-sized Gabor filters before and after non-Maxima suppression is proposed. The main target is to overcome the two challenges of the Canny method. Firstly, is the possibility of losing the weak edges by using the same-sized Gaussian filter. Secondary, the long computation speed. The computational speed is improved by using variable-sized sigma and threshold depending on the brightness and noise level of an image pixel. The implementation is done using Python.

1.1. Problem Statement

Based on the literature review, the Canny method is the most effective method for edge detection, however, it has the following significant flaws. First, the Canny method's Gaussian filtering may blur the weak edges out, making them go undetected. Additionally, Canny employs the same-sized filter, which could result in the loss of weak edges. Last but not least, the Canny method requires time-consuming, extremely complex computations, making it a poor choice for real-time applications like smart city car plate recognition. To overcome these challenges, this paper proposes a fast algorithm based on the application of different-sized Gabor filters before and after non-Maxima suppression. Also, the algorithm adjusts the threshold and sigma size in accordance with the brightness and noise level of an image part to speed up computation. Python is used to carry out the implementation.

This article is organized into five sections. Section 1 gives the introduction and the contributions of the proposed method. Also, section 1 defines the keywords used in the study and the problem statement. Section 2 is a literature review giving a survey on scholarly related works. Section 3 is about the methodology. Section 4 gives the experimental results and discussion. Finally, section 5 provides the conclusion. The end of the paper also contains an acknowledgment of key contributors toward successfully of this work and a list of references used.

1.2. The Keywords

An edge is a line joining the pixels separating two different light intensities in an image. Edge locating of detection involves determining and drawing a line to connect all pixels that sits on the points of a sudden change of light intensity in an image. In order to locate an edge, the first and second derivatives of an image may be computed first. Filtering is a necessary part of edge detection. Filtering involves the suppression of noise to clearly define the areas with different light intensities. In this work, a Gabor filter is engaged. The Gabor filter was invented by Hungarian engineer Denis Gabor. Gabor filter is a band pass linear filter which is orientation sensitive. The filter examines whether the image contains any particular frequency content in particular directions in a focused area close to the point or region of analysis. Python is a general-purpose and interpreted high-level language for computer programming. Python is chosen because it has simple codes and can be executed much faster than other languages' codes. The syntax of Python is simple meaning that it can easily be learned, read, and developed.

2. Literature Review

The basic edge detection methods can be based on the first derivative or second derivative of an image function [13]. The operators are of two types, the gradient-based ones, and the Gaussian based. Gradient-based computes the first derivatives while the Gaussian based computes the second derivatives [14], [15]. Examples of Gradient-based operators are Sobel, Prewitt, and Robert operators. Examples of Gaussian-based operators are the Laplacian of Gaussian and the Canny method. Out of these methods, Canny is optimal but involves complex computation. Table 1 summarizes the advantages and limitations of each of these basic algorithms.

2.1. Related Works

The canny proposal came to solve challenges imposed by other operators in Table 1. Despite the success of Canny methods, it has limitations hindering its application in many real-time applications. A number of studies have been done to solve the limitations of the Canny approach. Few works are reviewed in the following sub-sections.

2.1.1. Bilateral Detector

The bilateral edge detector has been proposed by Chandra and Abin [16]. Traditional bilateral filtering is a nonlinear method for smoothing the image while retaining its edges. Using domain and range kernels, the bilateral filtering operation is applied to a pixel's spatial neighbourhood. A Gaussian function serves as the domain kernel, which smoothens the image by giving each pixel a weight based on how far it is from the central pixel at a given location. The range kernel takes into account the radiometric distance of the pixels from the central pixel to preserve the edge. In order to achieve adaptive smoothing of the image, the intensity variations are taken into consideration. This is unlike the Canny method which uses the same sized filter and sigma. The bilateral detector is not an iterative method and hence reduces the computation time when compared to the Canny operator.

Table 1. Summary of the advantages and limitations of the basic edge detection algorithms

S/N	Operator	Advantages	Limitations
1	Robert	Simple than all other operators, small sized mask (2x2 matrix) hence has the lowest computation time, Easy to search smooth edges	It is very sensitive to noise, it is not very accurate in edge detection
2	Sobel	Uses a small-sized mask (3x3 matrix) hence fast than Prewitt and Canny, it is easy to search smooth edges	Inaccurate for weak edges, highly sensitive to noise, may fail to preserve diagonal direction points.
4	Laplacian of Gaussian	It can detect edges and their various orientations easily, and it has fixed characteristics in all directions	It is sensitive to noise, localization mail fails at error curved edges, and It generates noisy responses that do not correspond to edges, so-called “false edges”
S/N	**Operator**	**Advantages**	**Limitations**
5	Canny	Suitable for step edge, non-sensitive to noise, and characterized by good localization	Has complex computation, takes a long time to complete computations, and uses the same sized filter in strong and weak edges

2.1.2. Haar Wavelet-based Algorithm

By computing the sums and differences of adjacent elements, the Haar wavelet analyses can data. In this proposed method, the Canny detector is implemented with adaptive parameters using the Haar wavelet. These variables include sigma and others. It has been noted that canny detectors outperform conventional canny detectors when adaptive parameters are used.

2.1.3. Soft Morphology-based Detectors

The algorithm for mathematical morphology edge detection is presented by Shang and Jiang in [17]. In order to measure and extract the shape of the corresponding image for the purposes of image analysis and identification, Mathematical Morphology is a novel type of nonlinear algorithm. The basic operations of erosion and dilation, as well as open and closed operations, complete the type of interaction, making it more suitable for processing and analyzing visual information. A modified soft morphological edge detection algorithm that can accurately detect image edges while limiting the impact of noise and having strong robustness has been proposed after analysis and comparison of the conventional morphological edge detection algorithm. However, the computation speed is low and sensitive to noise hence the high probability of missing the weak edges.

2.1.4. SUZAN Algorithm

Brady and Smith invented the SUSAN algorithm as an improvement for the Canny detector. The algorithm shows better performance on low-contrast input images [18]. The study [19] proposes an adjustment of this algorithm to improve its performance in high-contrast input images. Another advantage of this detector is that its sensitivity to the noisy image is very small giving more edge details on images with low SNR when compared to another algorithm.

Basically, SUZAN solved the inability of the Canny operator to locate junction edge pixels. With the Canny edge detector, the pixels at corners may appear in the wrong directions for their neighbors leading to the missing junction. In this paper, the digital image will be represented by a mathematical model to ease analysis.

However, the main weakness of this method is the high latency between taking the input, processing, and provision of results.

2.1.5. Detectors based on Variable Sigma

Many scholars have proposed an improved version of the Canny detector based on using the variable sigma, see [20], [21], [22]. These scholars suggest that if we engage variable sigma during image smoothing it also affects the non-Maxima suppression process. The Canny detector works by accepting the input image I and the sigma value to the Gaussian filter. The filter filters out the noise to reduce its effect. The ordinary Canny detector uses a constant-sized sigma throughout the smoothing of the image. The proposed approach determines the noise of a specific part of the image and assigns an appropriate sigma. As a result, the part of the image with more noise is applied with a large sigma while low noise parts are applied with a small valued sigma.

2.1.6. The Marr-Hildreth Algorithm

Another algorithm was suggested by [23] before the Canny edge detector. It is based on a gradient operator. It is superior to the Gaussian operator because it uses both Laplacian and Gaussian filters. Gaussian filter for blurring the image while Laplacian for detecting sharp edges. Thus the output image is the convolution of the input image and the Laplacian of the Gaussian operator (LoG).

If the Gaussian filter is given by

$$G(i,j) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp - \left(\frac{i^2 + j^2}{2\sigma^2}\right)$$

Whereby i and j are the image pixel coordinates and σ is a sigma value then the LoG operator is given by the equation below.

$$LoG = \frac{\partial^2}{\partial i^2} G(i,j) + \frac{\partial^2}{\partial j^2} G(i,j) = \frac{i^2 + j^2 - 2\sigma^2}{\sigma^4} \exp\left(-\frac{i^2 + j^2}{2\sigma^2}\right)$$

The major weakness of this method is that it is sensitive to noise, localization mail fails at error curved edges, and it generates noisy responses that do not correspond to edges.

2.2. Research Gap

With all the algorithms proposed to solve the limitations of the Canny detector, there is no single algorithm solving all the limitations. As discussed in section 2.1, most of the algorithms solve one or two limitations. There is no single proposed method for solving the iteration issue and the use of a constant filter issue. This motivated the researcher to propose an algorithm that uses different sigma and filter sizes in smoothing the digital image. The filtering is performed before and after non-Maxima suppression.

3. METHODOLOGY

3.1. Edge Detection Process

The proposed algorithm has seven main stages. These stages are image acquisition, noise suppression stage, gradient computation, non-Maxima suppression, hysteresis thresholding, combining individual edges to form a single image, and output stage. The flow chart in Figure 1 enlightens these stages.

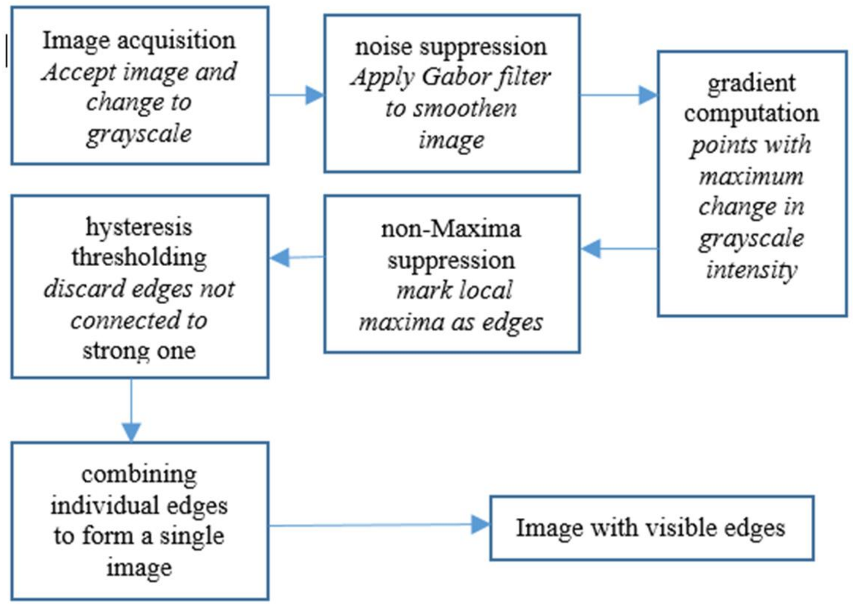


Image acquisition
Accept image and change to grayscale
noise suppression
Apply Gabor filter to smoothen image
gradient computation
points with maximum change in grayscale intensity
non-Maxima suppression
mark local maxima as edges
hysteresis thresholding
discard edges not connected to strong one
combining individual edges to form a single image
Image with visible edges

Fig 1. Edge detection process flow by the proposed algorithm

3.1.1. Image Acquisition

For easy processing, the input image is converted to a non-colored image called a grayscale image. The intensity of the image at any position ranges from 0 to 255.
Mathematical representation of an image

An image is nothing but a matrix of pixels packed in rows and columns [10-12], [13] also see [18] and [20]. The size of rows and columns depends on the resolution of an image. The resolution of an image is the total number of pixels represented as nx, ny. The digital image can be represented as a matrix of pixels' intensity at a location (xi, yi) where by nx is the width and ny is the height of the image. Let's assume the input to the algorithm has been already converted to a grayscale image $I_{i,j}, i = 1,2,3..n, j = 1,2,3,..n$ whereby $I_{i,j}$ represents the intensity of the digital image at a point (i, j). Practically the image contains a random noise $N_{i,j}$ which affect the pixel at the position (i, j). Thus, the gray level of the image's intensity can be represented by an additive mode shown as Equation.

$$G_{i,j} = I_{i,j} + N_{i,j} \tag{1}$$

3.1.2. Noise Suppression

First, image G is cleaned by the Gaussian of Gabor operator. Good results have been observed by the use of this combination. Gaussian filter is a low pass filter for removing high-frequency components in an image [13], [16]. Gabor filter is a band pass filter allowing the defined set of frequencies. The Gabor filter was invented by Denis Gabor in 1971 [21]. At any specific location and given orientation, Gabor filters are very sensitive to the boundaries of two different intensities in an image [21].

Mathematically a 1-D Gabor filter can be viewed as a product of Gaussian function and sinusoidal function defined by Equation 2.

$$g(x;\sigma_X, f, \phi) = \exp(-\frac{x^2}{2\sigma_X{}^2})\exp(i(\frac{2\pi x}{f}+\phi)) \qquad (2)$$

$$e^{ix} = \cos x + i\sin x; i = \sqrt{-1}$$

Whereby g is a Gabor function in the x-axis, σ_X standard deviation (sigma) in the x-axis direction, f and ϕ are the frequency and phase offset of the sinusoidal wave.

An image is a 2-D function hence we rewrite equation (2) into equation (3).

$$g(x,y;s,\lambda,\theta,\phi,\sigma) = \exp(-\frac{x^2+s^2y^{i2}}{2\sigma^2})\exp(i(2\pi\frac{x^i}{\lambda}+\phi)) \qquad (3)$$

Equation 3 has both real and imaginary parts:

Real part $g(x,y;s,\lambda,\theta,\phi,\sigma) = \exp(-\frac{x^2+s^2y^{i2}}{2\sigma^2})\cos(2\pi\frac{x^i}{\lambda}+\phi)$

Imaginary $g(x,y;s,\lambda,\theta,\phi,\sigma) = \exp(-\frac{x^2+s^2y^{i2}}{2\sigma^2})\sin(2\pi\frac{x^i}{\lambda}+\phi)$

Whereby $x^i = x\cos\theta + y\sin\theta$, $y^i = -x\sin\theta + y\cos\theta$

Fig. 2 (a) and (b) show a plot of 1-D and 2-D Gabor filters respectively.

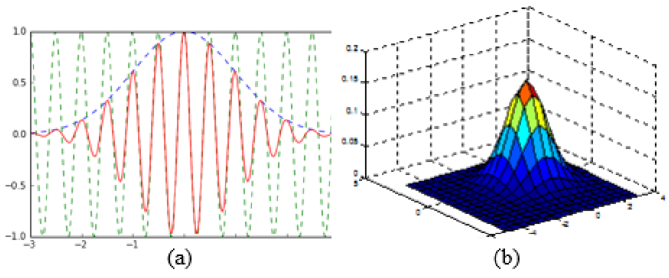


Figure 2. MATLAB plots of (a) 1-D and (b) 2-D Gabor filters

To filter the unwanted information, the grayscale image in its mathematical model is convoluted with the Gabor filter. The blurring of the image by the filter is specified by sigma σ [6].

Suppose the input image to the filter is $I(x,y)$ and the output image is $F(x,y)$, Figure 3 is the suggested smoothing architecture based on Equation (3).

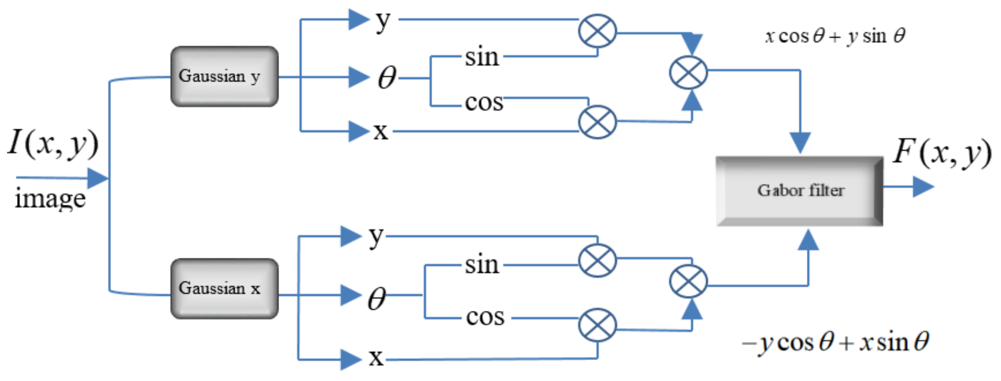


Figure 3. Noise smoothing architecture based on Gabor filter

3.1.3. Gradient Computation

The main concern is computing the gradient and finding the points pixel points with maximum change in grayscale intensity [13], [17]. Sobel operator is engaged to compute the gradients of the image pixels. The operator is applied in 2-D i.e. i and j directions. The Sobel operators in these two directions are given by matrices M_i and M_j.

$$M_i = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} \qquad M_j = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

The matrices M_i and M_j are convolved with blurred image pixels $F(i,j)$ ^{i} and j directions. Equations (4) and (5) are the gradients at pixel point (i , j).

$$g_i = M_i * F(i,j) \tag{4}$$
$$g_j = M_j * F(i,j) \tag{5}$$

Edge: The magnitude of the edge E is computed by applying Equation (6). look [17], [20], and [21].

$$E = \sqrt{g_i^2 + g^2{}_j} \tag{6}$$

The arc tan(g_j / g_i) give the direction of the computed gradient at pixel point (i , j).

3.1.4. Non-Maxima Suppressions

A non-Maxima is a small gradient of grayscale intensity at a point compared to that in either direction. Non-Maxima suppression means deleting this small gradient and preserve the larger one. This process is very important in object identification [14]. This stage involves marking of the local maxima as edges. The resulting gradient image contains non-maxima. According to [16] and [20], these are deleted so as to create thin and clear edges.

To achieve the suppression, the algorithm performs two process by using a Python do-while loop for all pixels. The proposed algorithm also adjusts the magnitude of the gradient for clear observation of the output image.

Step1: Define the magnitude of gradient for clear observation.
Step2: Round off the gradient of pixel $I(i, j)$ to nearest 45 degrees.
Step3: Compare the magnitude of its gradient $g(i, j)$ to that at position $(i-1, j-1)$ and that at position $(i+1, j+1)$.
Step4: If gradient magnitude at (i, j) is larger than the left and right ones, preserve the edge at (i, j).
Step5: Otherwise preserve the large gradient either left or right by discarding the one at (i, j).
Step6: Adjust the gradient to the defined fixed magnitude.
Step7: continue step1 to step4 for all pixels, i, j=1,2,3,4…. n

Hence when compared to the gradient in either direction, the non-Maxima gradient at a point has a small grayscale intensity.

3.1.5. Hysteresis Thresholding

The proposed algorithm suggests using different sized threshold values at different parts of the image depending on the noise level. An image may have banding, a fixed pattern, or random noise [16]. This work only considers fixed pattern noise. The assumption is made such that noise will tend to make an edge weak at the affected point, hence the algorithm lowers the threshold value. This stage involves the deleting the edges that are not connected to the strong edges. The results of the previous stage still contain edges created by noise. The proposed algorithm uses the variable threshold values depending on the noise level kept in memory from stage 2 of image blurring by the Gabor filter. Two threshold values are selected at every pixel point. The proposed algorithm passes through the following steps to achieve hysteresis thresholding.

Step1: If the gradient at point (i, j) > average of two threshold values, keep the gradient.
Step2: If the gradient at point (i, j) < average of two threshold values, discard the gradient
Step3: If the gradient at point (i, j) neither > nor < average of two threshold values, search in the 3*3 matrix region. If not found increase the region size to 5*5 matrices.
Step4: After step 3, repeat step 1 and step 2 to locate a maximum gradient.

3.1.6. Combining Individual Edges to form a Single Image

All gradients that pass a previous process are maintained in a matrix as an image with boundary traces.

3.2. Algorithm Implementation

The pseudocode for the process summarized in Fig.1 is given below

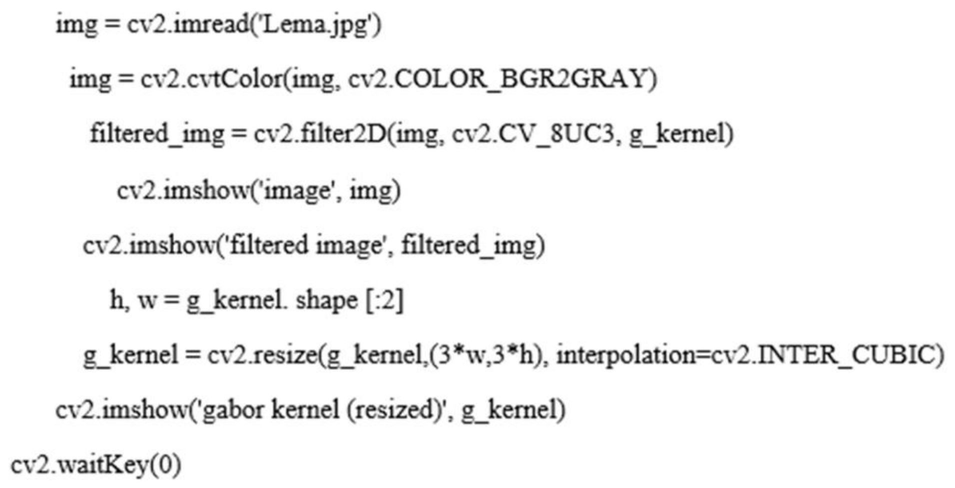
```
1. Read colored image I
2. Convert I to grayscale
```

```
3.  j=1
4       While j<512 do
5.          For i=1 to 512
6.              I(i,j)= 0.2989 * R + 0.5870 * G + 0.1140 * B//RGB are the red, green and blue pixel
            intensities
7           i=i+1, j=j+1
8         end while
9         Smoothen the grayscale image
10  Compute gradients and suppress the weak gradients
11  Perform hysteresis thresholding
12  Get the edge image
```

3.2.1. Python Programming

Python is opted for because of its simple syntax, is based on object-oriented, and can speed computation [24]. Python is easy to learn as a high-level programming language and runs faster on hardware such as Raspberry pi. For this study, a converter tool called small Matlab and octave to Python compiler was used to convert an m-file to corresponding Python code.

Part of Python code for image filtering



```
      img = cv2.imread('Lema.jpg')
        img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
          filtered_img = cv2.filter2D(img, cv2.CV_8UC3, g_kernel)
             cv2.imshow('image', img)
         cv2.imshow('filtered image', filtered_img)
            h, w = g_kernel. shape [:2]
         g_kernel = cv2.resize(g_kernel,(3*w,3*h), interpolation=cv2.INTER_CUBIC)
      cv2.imshow('gabor kernel (resized)', g_kernel)
cv2.waitKey(0)
```

3.2.2. MATLAB2019a

Matlab is a numeric computation environment developed and supported by MathWorks. Matlab2019 was selected due to its flexibility in code conversion and improved computer vision toolbox.

3.3. Algorithm Testing

Two DIT laboratory images were used for testing the implementation. The results summarized in Fig.4 was after several adjustments of sigma and threshold value.

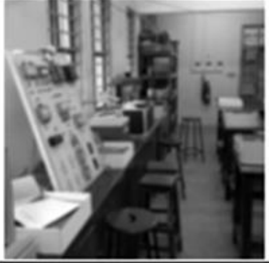
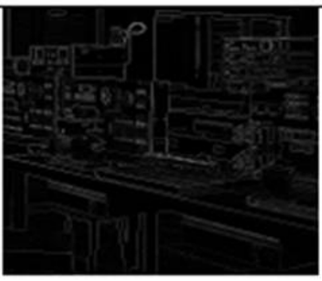
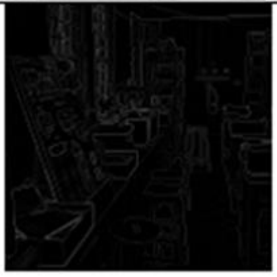
Input image		
Grayscale image		
Smoothened image		
Edge detection		

Fig 4. Results of algorithm testing

4. Experimental Results

4.1. Variables

Central spatial frequency: This is the frequency of the unidirectional sinusoidal signal. The range is (0 - 0.5) cycle per pixels. In simulations, a maximum frequency of 0.5 cycle per pixels was used. This help to avoid possible breakdowns of the filtered image. See [14] and [21]. Low and high thresholds used were 5% and 30% respectively.

Other variables such as sigma, lambda, s are shown in Table 2

Table 2. Values of variables applied in simulation

S/N	Variable	Value	Entry	Required value
1	Sigma σ	2.01	$1/\sigma^2$	0.2475
2	Lambda λ	3.3	$1/\lambda$	0.303
3	Spatial speed s	0.3	s^2	0.09

4.2. Datasets

Two standard images (Lena and sunflower) with a resolution of 512x512 were used to determine the algorithm's performance. PSNR, MSE, and processing time were recorded. Furthermore, to compare the results with other algorithms, it was necessary to use standardized datasets [9], [18], [25]. Two laboratory images taken at Dar es Salaam Institute of Technology (DIT) was used in testing the algorithm only.

4.3. Results Discussion

The results obtained with the proposed algorithm as detailed in Table 3. Figure 4 shows filtered images and corresponding edge detection by proposed algorithm.

Table 3. Simulation results

S/N	Image used	Size used	MSE (x10^3)	PSNR
1	Lena	512x512 standard	4.2398	12.06
2	Sun Flower Image	512x512 standard	3.8224	13.77

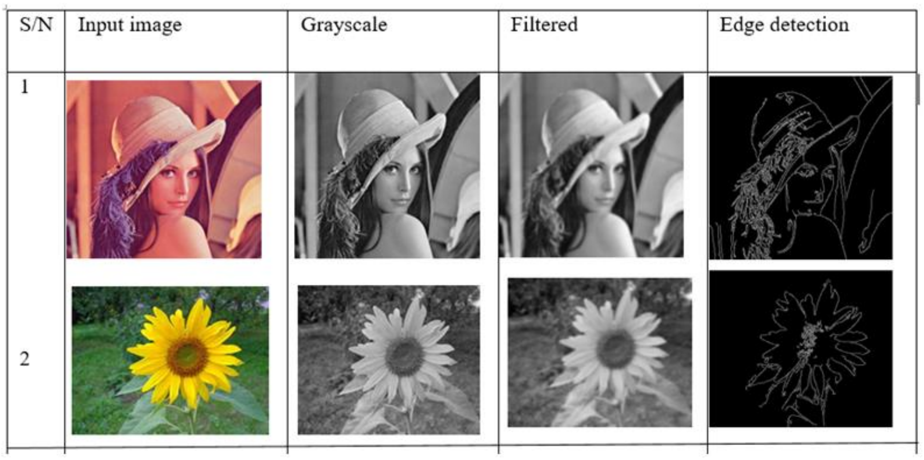


Figure 4. Original images and their corresponding filtered images and edge detection for Lena and sunflower standard test images

4.4. Performance Comparison

The proposed algorithm in this work is compared against other methods reviewed earlier. Even though not all works were compared but the best among them were selected. The proposed algorithm was compared against Canny detector, The Marr-Hildreth algorithm, Haar wavelet-based algorithm, Soft morphology-based detectors, Laplacian of Gaussian, and Sobel operator.

The performance comparison is done in terms of the following important terms in image processing:

1. Mean Squared Error (MSE): Varies inversely proportional with the quality of the edge detected [22].
2. Peak Signal to Noise Ratio (PSNR): Varies proportionally with the quality of edge detected [22].
3. Processing time: This is a response time from the moment of accepting image to provision of the edge image. Processing time of different algorithms is summarized by study conducted by [25].

Table 4 shows the comparison results; the results show a significant improvement with the proposed study.

From Table 4, the proposed method performs better in terms of PSNR and MSE and processing time when compared to the rest. The testing was not done for other standard images. It was conducted on 512x512-sized Lena and sunflower images. The reason is the unavailability of PSNR and MSE values of many algorithms. Other standard images such as salt pepper, apple, and Statue of liberty lack the values to compare with the proposed method [22], [25].

Table 4. Comparison between the proposed method and reviewed methods in terms of PSNR, MSE and processing time values on Lena and sunflower standard 512x512 images

S/N	Algorithm	Testing image	MSE (x10^3)	PSNR(dB)	Processing time(s)
1	Proposed	Lena	4.2398	12.0621	3.1312
		Sunflower	3.8224	13.7701	N/A
2	Canny detector [22],[23]	Lena	15.7000	6.2060	3.9411
		Sunflower	87.54	8.7087	No data
3	The Marr-Hildreth [22] ,[23] based on LOG	Lena	19.53	5.2217	40.7277
		Sunflower	87.66	8.7024	No data
4	Haar wavelet-based [23],[24]	Lena	6.2343	11.8700	No data
		Sunflower	No data	No data	No data
6	Laplacian of Gaussian [22], [23]	Lena	19.53	5.2217	40.7277
		Sunflower	87.66	8.7024	No data
7	Sobel operator [22],[23]	Lena	19.42	5.2476	29.6719
		Sunflower	87.78	8.6965	No data

5. CONCLUSION

This article proposes an algorithm based on the Gaussian of Gabor filter (GoG) operator. The implementation has been done in Python and some parts were tested by Matlab2019a. Two standard images of Lena and sunflower were used to compute PSNR and MSE (summarized in

Table 3). The results in Table 3 were compared to the PSNR and MSE obtained by other reviewed edge detection algorithms. Moreover, the processing time of the algorithms in these two images was recorded and compared to that of the proposed method. The comparison of which has been presented in Table 4.

Therefore, the proposed approach attains better edge detection with a thick edge or the boundaries of an image compared to the reviewed edge detecting algorithm.

Funding: This research received no external funding.
Conflicts of Interest: The authors declare no conflict of interest.

ACKNOWLEDGMENTS

The author acknowledges the valuable comments of Prof. Dr Dening Jiang of Tianjin University of Technology and Education and Dr Mbazigwa Mkiramweni of DIT for their valuable inputs in this work. The author is also grateful to the anonymous reviewers whose comments improved the presentation of the article contents.

REFERENCES

[1] Suman Shrestha, (2014) “Image denoising using new adaptive Based median filter”, Signal & Image Processing: An International Journal (SIPIJ) Vol.5, No.4, pp1-13.
[2] P. Srujana, J. Priyanka, V. Y. S. S. S. Patnaikuni and N. Vejendla, "Edge Detection with different Parameters in Digital Image Processing using GUI," 2021 5th International Conference on Computing Methodologies and Communication (ICCMC), 2021, pp. 795-802, doi: 10.1109/ICCMC51019.2021.9418327
[3] G. M. H. Amer and A. M. Abushaala, "Edge detection methods," 2015 2nd World Symposium on Web Applications and Networking (WSWAN), 2015, pp. 1-7, doi: 10.1109/WSWAN.2015.7210349.
[4] S. R. Joshi and R. Koju, "Study and comparison of edge detection algorithms," 2012 Third Asian Himalayas International Conference on Internet, 2012, pp. 1-5, doi: 10.1109/AHICI.2012.6408439
[5] Ajay Kumar Boyat & Brijendra Kumar Joshi, (2015) “A review paper: noise models in digital Image processing”, Signal & Image Processing : An International Journal (SIPIJ) Vol.6, No.2, pp63-75.
[6] Isaack Kamanga, "An Adaptive Approach to Improve Canny Method for Edge Detection", International Journal of Science and Research (IJSR), Volume 6 Issue 6, June 2017, pp. 164-168
[7] Chandra Sekhar Seelamantula and Abin Jose and,“Bilateral Edge Detectors",ICASSP,IEEE,pp:1449-1453,2013.
[8] Meiling Huang, Yuelei Liu, Yongmei Yang, Edge detection of ore and rock on the surface of explosion pile based on improved Canny operator, Alexandria Engineering Journal, Volume 61, Issue 12, 2022, Pages 10769-10777, ISSN 1110-0168, https://doi.org/10.1016/j.aej.2022.04.019
[9] Isaack Adidas Kamanga, Dening Jiang, "Securing Iris Recognition System for Access Control Based on Image Processing", International Journal of Science and Research (IJSR), Volume 6 Issue 10, October 2017, pp. 1131-1140,
[10] J. F. Canny, A Computational Approach to Edge Detection, IEEE Trans. on PAMI, vol 8, No.6, pp. 679-698 (1986).
[11] W. Rong, Z. Li, W. Zhang and L. Sun, "An improved Canny edge detection algorithm," 2014 IEEE International Conference on Mechatronics and Automation, 2014, pp. 577-582, doi: 10.1109/ICMA.2014.6885761.
[12] Ewaryst, R. SUSAN edge detector reinterpreted, simplified and modified, Proceedings of the 2007 International Workshop on Multidimensional (nD) Systems. nDS 2007 Aveiro, Portugal, June 27-29, 2007. [Piscataway, NJ] : IEEE, cop. 2007, 1-4244-1112-2/07
[13] Shashidhar Ram Joshi,Roshan koju," Study and Comparison of Edge Detection Algorithms", IEEE,2012.
[14] Shang Junna, Jiang Feng," An Algorithm of Edge Detection Based on Soft Morphology",IEEE,2012,pp: 166-169.
[15] Shokhan M. H.(2014). An efficient approach for improving canny edge detection algorithm. International Journal of Advances in Engineering & Technology. Issue 1 volume 7.

[16] Chao Gao, Hongjun Zhu, Yongcai Guo, Analysis and improvement of SUSAN algorithm, Signal Processing, Volume 92, Issue 10,2012,Pages 2552-2559, ISSN 0165-1684.

[17] M. Sharifi, M. Fathy and M. T. Mahmoudi, "A classified and comparative study of edge detection algorithms," Proceedings. International Conference on Information Technology: Coding and Computing, 2002, pp. 117-120, doi: 10.1109/ITCC.2002.1000371.

[18] Lejiang Guo,Yahui Hu Ze Hu, Xuanlai Tang "The Edge Detection Operators and Their Application in License Plate Recognition, IEEE TRANSATION 2010, 20978-1-4244-5392-4/10/.

[19] Jianjum Zhao, Heng Yu, Xiaoguang Gu and Sheng Wang. "The Edge Detection of River model Based on Self-adaptive Canny Algorithm And Connected Domain Segmentation" IEEE,Proceedings of the 8th World Congress on Intelligent Control and Automation July 6-9 2010, Jinan, China, 978-1-4244-6712-9/10/$26.00 ©2010 IEEE.

[20] Mee-Li Chiang, Siong-Hoe Lau "Automatic Multiple Faces Tracking and Detection using Improved Edge Detector Algorithm" IEEE 7th International Conference on IT in Asia (CITA),2011 ,978-1-61284-130-4/11/

[21] K. R. Namuduri, R. Mehrotra and N. Ranganathan, "Edge detection models based on Gabor filters," Proceedings., 11th IAPR International Conference on Pattern Recognition. Vol. III. Conference C: Image, Speech and Signal Analysis, 1992, pp. 729-732, doi: 10.1109/ICPR.1992.202090.

[22] Dakshayani, V.; Locharla, G.R.; Pławiak, P.; Datti, V.; Karri, C. Design of a Gabor Filter-Based Image Denoising Hardware Model. Electronics 2022, 11, 1063. https://doi.org/10.3390/electronics11071063

[23] D. Poobathy, R. Manicka Chezian,"Edge Detection Operators: Peak Signal to Noise Ratio Based Comparison", IJIGSP, vol.6, no.10, pp.55-61, 2014.DOI: 10.5815/ijigsp.2014.10.07

[24] Maini R, Aggarwal H. Study and comparison of various image edge detection techniques. IJIP. 2009 Mar; 3(1):1–12.

[25] https://www.python.org/doc/essays/blurb/ Visted on 12th June, 2022.

[26] Raver, C. (2020). LENA Data. Databrary. Retrieved August 24, 2022 from http://doi.org/10.17910/b7.1102

AUTHORS

Isaack Adidas Kamanga is a research assistance at Dar es Salaam Institute of Technology with a research base in computer vision, Machine learning, Artificial intelligence and Signal and information processing. Specifically, in application of image processing in medical diagnosis automation, Fault identification in systems, pests and diseases identification in crops, features extraction from image. Application of sound spectrum analysis in recognition. FPGA and VLSI technologies and Nano-electronics. He completed MEng. at Tianjin University of Technology and Education in 20018, Tianjin China P.R. Completed B.S.in Telecommunications Engineering from University of Dar Es Salaam in 2012. He is a registered professional engineer by Tanzania Engineers Registration Board (ERB) Tanzania