International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 3 Issue 11, November-2014

Development of a Matlab Guided Based Interactive Platform for Edge Detection in Noisy Coloured Images

Sadiq Bashir Olaniyi
Department of Electrical and Computer Engineering
Ahmadu Bello University Zaria
Kaduna State, Nigeria.

Abstract **- This work is aimed at the development of a MATLAB GUIDE-based interactive platform for Traditional-Median edge detection algorithms in coloured images such as Sobel-Median, Prewitt-Median, Robert-Median and Laplacian-Median. The Median filter was introduced into the existing Traditional edge detection algorithms with a view to performing on noisy images. GUIDE is a development environment in MATLAB that provides a set of tools for creating Graphical User Interfaces (GUIs). The developed Graphical user interface provides a comparative analysis between the existing Traditional edge detection algorithms and presents the output results in figures.**

Keywords: MATLAB, GUIDE, Traditional Edge Detectors, Medial Filter

1. INTRODUCTION

GUI design has been around for a short time, but the universal qualities of a good design have remained unchanged by its arrival [1]. GUIDE is a development environment in MATLAB designed to make building Graphical User Interfaces (GUIs) faster and easier [1].It also makes it easier to adjust parameters and visualize your programs. Edge detection in digital images is defined as a set of connected pixels that forms a boundary between two disjoints regions [2]. It can also be defined as the process of locating and identifying sharp discontinuities in images [3].Different factors causes these discontinuities in images, amongst which are reflection of light, shadows or illuminations [4]. Traditional edge detection algorithms are very fast but they cannot perform well on noisy images. Hence, the significant problem of these edge detection algorithms are displacement, removed edges, false and broken edges [5]. In order to perform in noisy environments, the Median filter is introduced as an improvement over the existing traditional edge detection algorithms, with a view to reducing the noise present in the images before detection of edges. Image filters are used to improve the detectability of important image details or objects by man or machine [6].

The developed standalone GUI application will help researchers obtain a comparative analysis between the existing traditional edge detection algorithms without having to resort to coding the algorithms when needed.

2. TRADITIONAL EDGE DETECTORS

The existing traditional edge detection algorithms are presented as follows [5]

i Sobel Edge Detectors

The Sobel edge detection algorithm uses a 3x3 convolution kernels designed to operate in both horizontal and vertical directions denoted as G_p and $G_q$respectively. One kernel is simply the other rotated by 90^0 [5]. The mask used by Sobel edge detection algorithm is shown in Figure 1.0

-1	0	+1
-2	0	+2
-1	0	+1

G_p

+1	+2	+1
0	0	0
-1	-2	-1

G_q

Figure 2.0. A 3x3 kernel of Sobel Edge Detection Algorithm

ii Prewitt Edge Detectors

The Prewitt edge detection algorithm is very similar to the Sobel edge detection algorithm, but uses a different convolution kernel. The Prewitt edge detection algorithm uses a 3x3 convolution kernels designed to operate in both horizontal and vertical directions denoted as G_p and $G_q$respectively. It performs better than the Sobel edge detection algorithm [3]. The maskused by Prewitt edge detection algorithm is shown in Figure 2.1 [5]

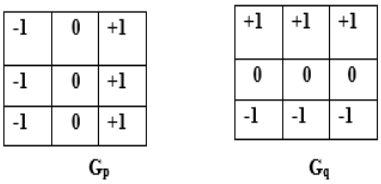

Figure 2.1 3x3 kernel of Prewitt Edge Detection Algorithm

ii Roberts Edge Detectors

The Robert Edge detection algorithm uses a 2x2 convolution kernel to compute the gradient magnitude and direction. Its computes the difference between the sum of squares of the diagonally adjacent pixel using discrete differentiation. Thus finding the approximate gradient in the image. The 2x2 kernel are shown in Figure 2.2 [5]

Figure 2.2. A 2x2 kernel of Robert Edge Detection Algorithm

iv Laplacian Edge Detectors

The Laplacian edge detection algorithm is a typical example of the second order edge detection algorithm, based on the second order derivatives. In Laplacian edge detection algorithm, the edges are usually marked at the position were the second order derivative becomes zero. The three commonly used laplacian kernel are shown in figure 2.3 [5].

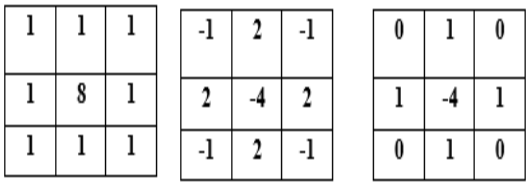

Figure 2.3. Three Commonly Used Laplacian Convolution Kernel

2.1 Median Filter

Median filtering is a nonlinear signal processing technique developed with a view suppressing noise in images [7]. In one-dimensional form, the Median filter consists of a sliding window consisting of a number of odd pixels. The center pixel in the window is replaced by the Median of the pixels in the window. In Median filtering, there are various strategies of applying of the Median filter with a view to suppressing noise. The first method will use a window of length 3, if there is no significant signal loss the window length could be increased to 5 for effective Median filtering of the original noisy input image [7]. For example, if the values of the pixels within a window are 0.1, 0.2, 0.9, 0.4, 0.5, the center pixel would be replaced by the value 0.4, which is the Median value of the sorted sequence 0.1, 0.2, 0.4, 0.5, 0.9. [7].

There exists other types of smoothing techniques which are effective at removing noise in an image, but adversely affect edges. While reducing noise in images, it is important to preserve the edges because they are critical importance to the visual appearance of images [8]. The Median filter is capable of removing moderate level of noise whilst preserving edges for given window size. It is also effective in the removal of impulsive noise. Due to this fact, the Median filter is widely used in digital image processing applications [8]

3. METHODOLOGY

In developing a GUI for edge detection in coloured images based on traditional-Median edge detection algorithms, the following methodology is adopted:

i) Initializing the guide by typing the word "guide" in the MATLAB command prompt.
ii) Creating a GUI that plots a given function by selecting the "axes" button that will display input image.
iii) Configuring the basic control button that performs an action when a user clicks on it.
iv) Saving your current version and running the program by pressing the green arrow at the top of the GUI editor.
v) Writing the codes for the various windows of the GUI using traditional- Median edge detection algorithm

The GUI development flow chat for the traditional-Median edge detectors is shown in Figure 3.1

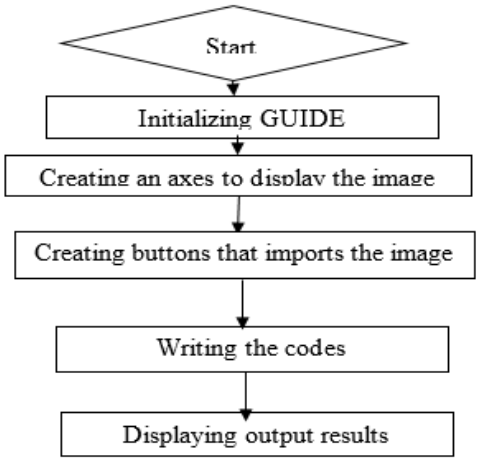

Figure 3.1 MATLAB GUI Flow Chart

3.1 Initializing the GUIDE

Figure 3.2 and Figure 3.3 shows how to initialize the "guide" and how to select an axes to display the input and output image.

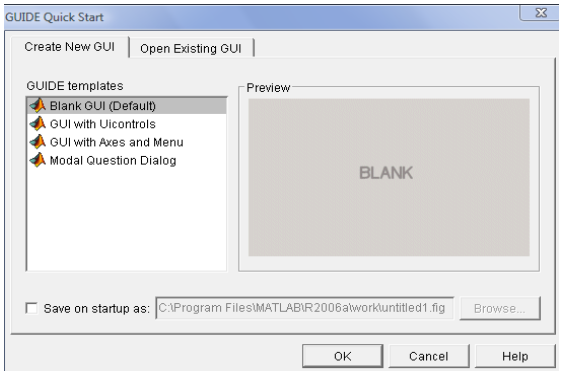

Figure 3.2 Guide Window Interface

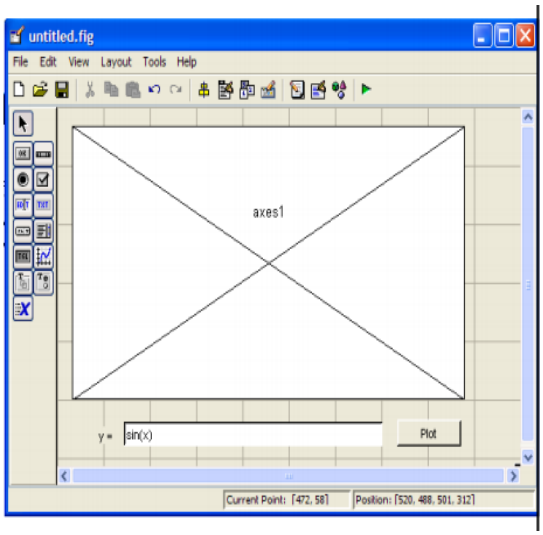

Figure 3.3 Axes to Display the Imported Image

Figure 3.4 shows the overall platform after all axes and buttons have been selected. The overall Platform of the GUI consists of four axes and buttons to display the individual output image as a result of applying the edge detection algorithms.

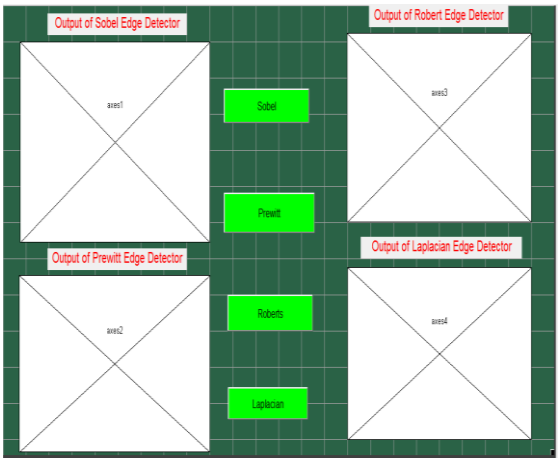

Figure 3.3 Overall Platform of the GUI

4. PERFORMANCE EVALUATION

Figure 4.1 shows the test input image used in evaluating the effectiveness of the algorithms

Figure 4.1. Noisy Lena Image

The Lena image is a standard benchmark image used for edge detection in image processing as a low level feature. Figure 4.2 shows the result of applying a Traditional-Median filter to a noisy image with a view to detecting the edges present in the image.

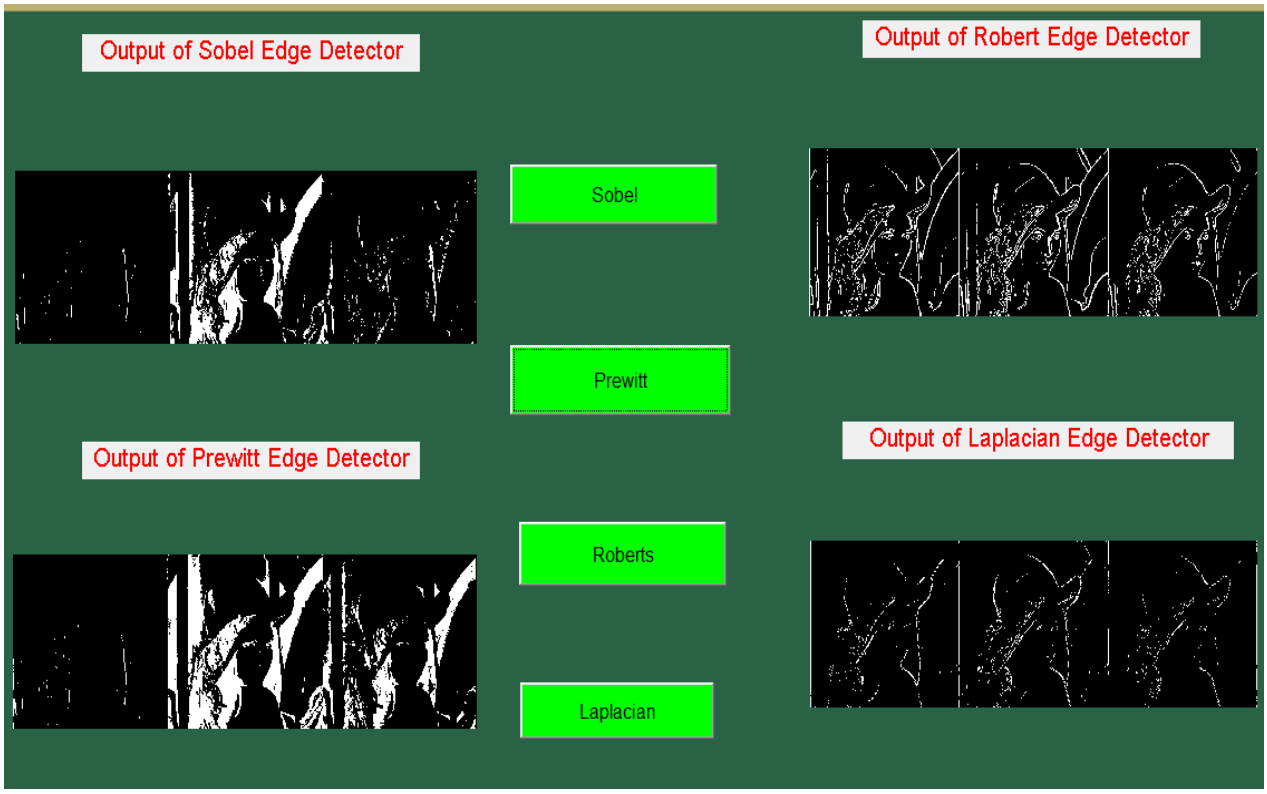

Figure 4.2 Output Result of Detected Edges

The displayed results showed the comparative analysis between the existing traditional edge detectors in a graphical form using Matlab image processing Toolbox. The Roberts edge detection algorithm has a small kernel size which makes it highly sensitive to noise and not compatible with modern technology. But with the modification by applying the median filter to smoothening the image, the algorithm can perform reasonably in noisy environment. The Sobel edge detection algorithm is less sensitive to noise as compared to the Robert edge detection algorithm due to its larger kernel size. This performs better than the Robert edge detection algorithm. The Prewitt edge detection algorithm is said to perform better than the Sobel edge detection algorithm due to its improved kernel size [5]. The kernel used in laplacian edge detection algorithm combined with the Gaussian kernel to produce the Laplacian of Gaussian (LoG) kernel with a view to dealing with noise present in images. However, this method produce responses that do not match edges, thus resulting in 'false edges' and localization error may be severe at curved edges [3].

5. CONCLUSION

This paper presents a graphical user interface (GUI) comparison between the existing traditional edge detection algorithms which are the Sobel, Prewitt, Roberts and the Laplacian. In order for the algorithm to perform in noisy environments, a Median filter was introduces with a view to smoothening the images. These reduces the effect of false and broken edges that exists with the traditional edge detection algorithms.

REFERENCES

[1] Building GUIs with MATLAB version 5 retrieved http://www.mathworks.com
[2] Gang L., Fan Y., & Linlin W. (2008). An Algorithm for Remote Sensing Image Edge Detection Based on Fuzzy Sets. *Second International Symposium on Intelligent Information Technology Application*, 1087-1090.
[3] Rashmi, Kumar M., & Saxena R. (2013). Algorithm and Technique on Various Edge Detection: A survey. *Signal & Image processing : An International Journal (SIPIJ), 4*, 65-75.
[4] Ghasemi M., Koohi M., & Shakery A. (2011). Edge detection in multispectral images based on structural elements. *The International Journal of Multimedia & Its Applications (IJMA), Vol.3*(No.1), 90-99.
[5] Maini R., & Aggarwal H. (2011). Study And Comparism of Various Image Edge Detection Technique. *internation Journal of Image Processing (IJIP), 3*(1), 1-12.
[6] Petrou M., & Bosdogianni P. (1999). *Image Processing the Fundermentals* (pp. 1-325).
[7] William k. Pratt (2007). Digital image processing 4th Edition. A John Wiley & Sons, Inc., Publication (pp. 277-282)
[8] Ramamoorthy K., Chelladurai T., Sundararajan P., & Krishnamurthy M. (2014). Noise suppression using weighted Median filter for improved edge analysis in ultrasound kidney images. *Int. J. Comp. Sci. Mobile Comput, 3*, 97-105.