BEST: International Journal of Management, Information Technology and Engineering (BEST: IJMITE)
ISSN 2348-0513
Vol. 2, Issue 7, Jul 2014, 65-70

IMPLEMENTATION OF SOBEL EDGE DETECTION ALGORITHM

LAVANYA K B[1], K V RAMANA REDDY[2] & SIVA S YELLAMPALLI[3]

[1]Research Scholar, VLSI Design and Embedded Systems, VTU Extension Centre,
UTL Technologies, Bangalore, Karnataka, India

[2]Assistant Professor, VTU Extension Centre, UTL Technologies, Bangalore, Karnataka, India

[3]Professor, VTU Extension Centre, UTL Technologies, Bangalore, Karnataka, India

ABSTRACT

Edge of image is one of the most fundamental and significant features, Edge detection is always one of the classical studying projects of computer vision and image processing field. In this paper we deal with MATLAB/SIMULINK model for Sobel edge detection technique and the corresponding simulation results. The edge detection technique used in feature detection and feature extraction from an image

KEYWORDS: Edge Detection, Gradient Method, Sobel Edge Detection

INTRODUCTION

The edge detection is a terminology in the Image processing particularly in the area of feature extraction to refer to algorithm which aims at identifying points in a digital image at which the image brightness changes sharply [1]. There are some well-known methods for edge detection such as Sobel, Canny, Prewitt and Robert algorithms which are different in terms of performance on hardware, speed and simplicity [1]. The Sobel operator is mainly used for hardware implementation due to efficiency and simple mathematical model that make it easy for real-time edge detection application [2]. The edge detection is mainly applicable in case of data transmission; where the edge detected data reduce the amount of data to be transmitted. Edge detection finds its application in Computer vision applications, boundary detection, motion detection, segmentation, texture analysis and object identification [2].

The sobel edge detector is very popular than simple gradient operators, because its less complex and easy for computations [2]. The accuracy of the Sobel for edge detection is relatively less because it uses only two masks which detect the edges in horizontal and vertical directions only. The accuracy can be enhanced by using the Soble operator which uses a larger set of masks [1,2] In this paper we discuss an Sobel edge detection algorithm is implemented using simulink blockset.

Image Edge Detection

Edge detection refers to the extraction of the edges in a digital image. It is a process whose aim is to identify the points in an image where discontinuities or sharp changes in intensity occur. This process is crucial to understanding the content of image and has its applications in image analysis and machine vision. It is usually applied in initial stages of computer vision applications [3, 4]. Edge detection aims to localize the boundaries of object in an image and is a basis for many image analysis and machine vision applications.

The purpose of finding edges of an image is to significantly reduce the amount of data to be stored and to filter out useless information, while the important structural properties of an image being preserved [5]. Edge detection is a type of image segmentation technique which determines the presence of an edge in an image and outlines it in the correct way.

With the advent of artificial intelligence systems and forensic science the process of edge detection has achieved the most sought status. Edge detection finds its applications in boundary detection, texture analysis, motion detection/estimation, segmentation, and object identification.

The Sobel filter is a widely used filter for image edge detection. The idea is to compute the first derivative of a grayscale input image. The magnitude of the first derivative is used to detect the presence of an edge in an image.

Sobel Edge Detection Algorithm

In edge detection the Sobel operator is used commonly [1]. The Sobel operator is a classic first order edge detection operator, computing an approximation of the gradient of the image intensity function. Any approximation used for first derivative must satisfy the following conditions:[1]

- It must be zero in the area of constant intensity.
- It must be nonzero at the onset of intensity step or ramp.
- It must be nonzero at points along an intensity ramp.

First derivative are implemented using magnitude of gradient. For an image f (x,y) the gradient of image f at (x,y) is defined as [5]

$$f = \text{grad}(f) = [Gx\ ,Gy]^T = [\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}]^T \qquad (1)$$

Where Gx is the gradient in the horizontal direction, Gy is the gradient in the vertical direction.

The two gradient are calculated using the partial derivative of image function f with respect to x and y at every pixel location of the image. Digital approximations of these partial derivatives can be expressed as [1].

$$Gx = \partial f / \partial x = [f(x + 1, y) - f(x - 1, y)]/2 \qquad (2)$$

$$Gy = \partial f / \partial y = [f(x, y + 1) - f(x, y - 1)]/2 \qquad (3)$$

The Gradient magnitude is given by

$$G = |Gx| + |Gy| \qquad (4)$$

Here G represents the value of rate of change in the direction of gradient vector.

High Level Implementation of Sobel Edge Detection Algorithm

As design become larger and more complex, it is necessary to describe the design at a high level, which will help the designer to run simulations faster and identify bugs in the early stages. Simulink is developed by MathWorks, is a data flow graphical programming language tool for modeling, simulating and analyzing multi domain dynamic systems. Its primary interface is a graphical block diagramming tool and a customizable set of block libraries. Simulink is widely used in control theory and digital signal processing for multi domain simulation and Model-Based Design.

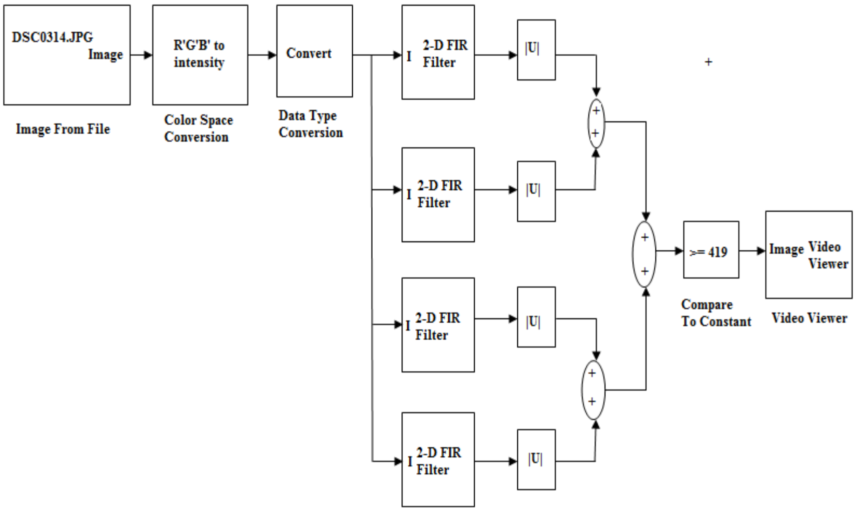


Figure 1: Design of Sobel Edge Detection Using Simulink

From the figure 1 we can see ‘Image from file’ block which is used to loaded an image from the file specified by the filename. Here the image is a static image of M –by-N-by-P array and the block outputs a color image, where M is the number of rows and N is the number of columns in each color plane, P. Next we have color space conversion block converts color information between color spaces. Use the conversion parameter to specify the color spaces. This block converted color image to gray scale image. The Converted block is used for data type conversion from unsigned integer to double. The 2-D FIR filters are is used to perform convolution operation with input image matrix to get the gradient. The gradient is calculated along four directions, such as a 0, 45, 90 and 135 degree orientations and added together to get the final gradient value. Thresholding block is used to comparing the gradient value coming from the 2-D filter and user defined value to get the binary edge detected image.

Pre Processing Block

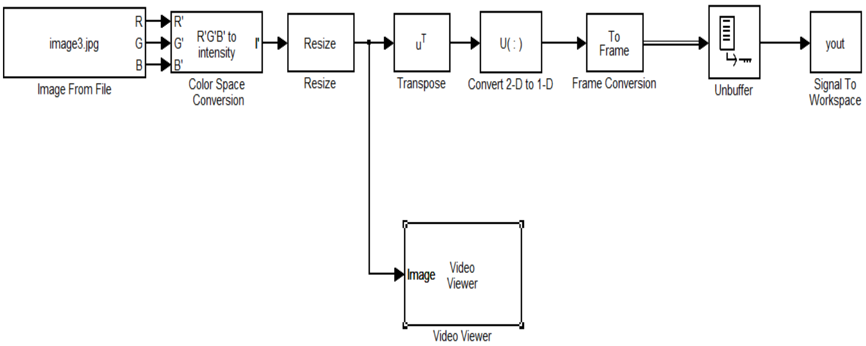
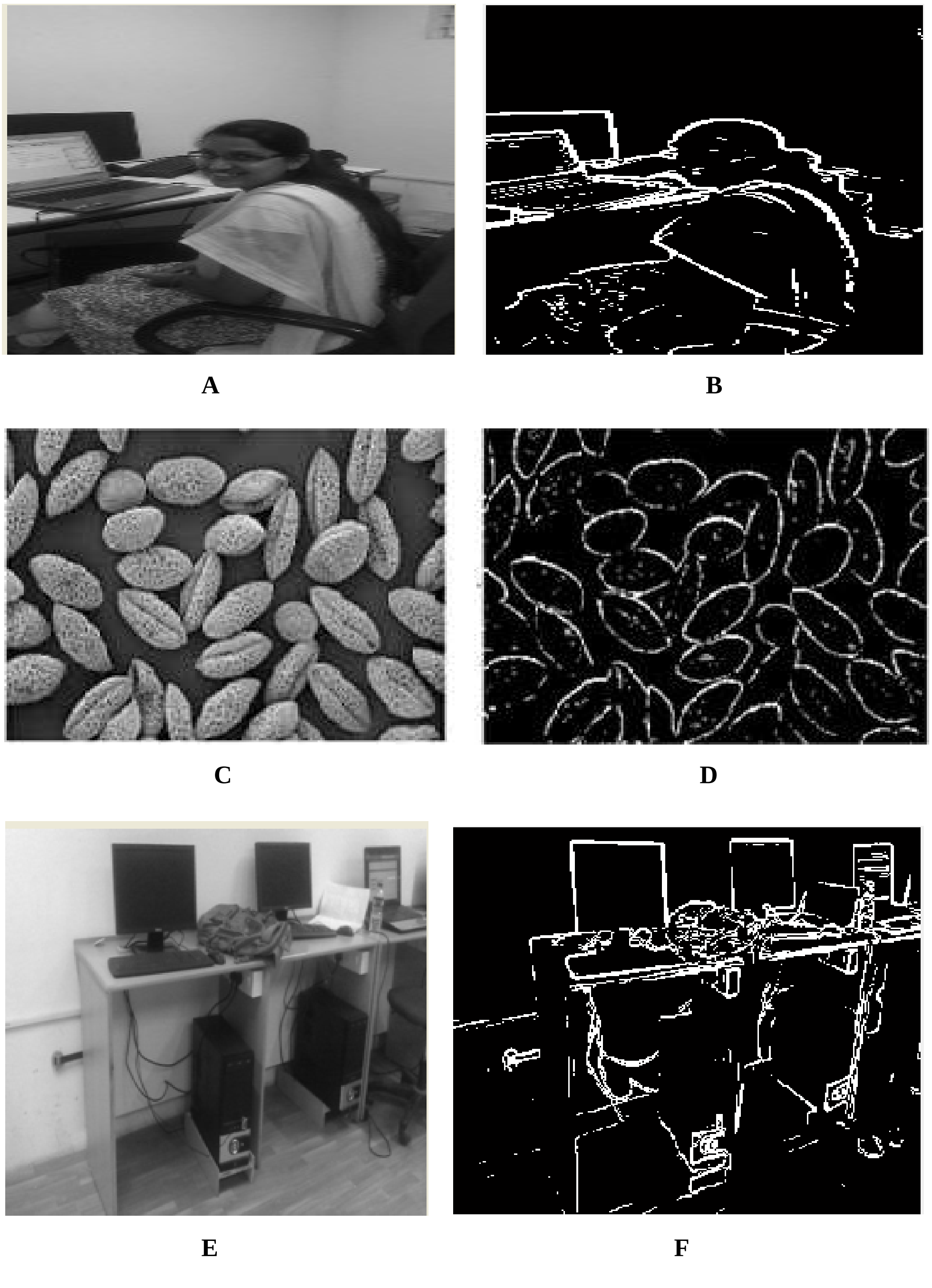


Figure 2

The above figure shows the pre processing block is designed using Simulink block, it is used to convert the two dimensional (2D) image data into one dimensional data. Input is color image of size 512 x 512 is given as input to the file block. A color space conversion block convert RGB image to gray scale image and then image is resized to get an image size of 128 x 128. This gray scale image data which is 2D is to be converted into 1D for further processing. After conversion blocks sets output signal to frame based data. The frame buffer block does not make any changes to the input signal other than the sampling mode. Unbuffer block converts this frame to scalar samples output at a higher sampling rate. The pixels values from the unbuffered block are stored in the work space.

Simulation Result

Different input images and their edge detected images are shown the Figure 2. The figures A, C, E are the input images and figures B, D, F are their edge detected results respectively.



A B

C D

E F

Figure 3

CONCLUSIONS

Edge detection forms a pre-processing stage to remove the redundant information from the input image, thus dramatically reducing the amount of data to be processed while at the same time preserving useful information about the boundaries. Here we are discussed with implementation of Sobel edge detection technique in modelling type by using the MATLAB/SIMULINK. . It detects the edges of an image. It can be efficiently used as a part of complex computer vision system.

ACKNOWLEDGEMENTS

I acknowledge, Dr. V. Venkateswarlu, Principal, VTU Extension Centre, UTL Technologies Ltd., Bangalore for his guidance and suggestion and UTL Technologies Ltd., Bangalore for providing lab facility during the design and implementation.

REFERENCES

1. Rafel C. Gonzalez, Richard E. Woods “Digital Image Processing” Prentice Hall, 2nd edition (January 15, 2002)
2. Rahul R, Gaulkar, Swati R. Dixit and A.Y. Deshmukh “Design of VHDL based Multi-Direction Sobel Edge Detection Processor”, International Journal of current Engineering and Technology, vol, 4, No. 2 pp. 745-748. April 2014.
3. Pratt, William K. “Digital Iamage Processing”
4. Mamta Juneja, Parvinder Singh Sandhu, Performance Evaluation of Edge Detection Techniques for Images in Spatial Domain, International Journal of Computer Theory and Engineering, Vol. 1, No. 5, December, 2009
5. Anna Fabijańska, Dominik Sankowski (2008), Edge Detection in Brain Images, MEMSTECH’2008, pp 60-62.