

IMPLEMENTATION OF SIMULINK BASED MODEL USING SOBEL EDGE DETECTOR FOR DENTAL PROBLEMS

Deepika Nagpal

MTech Scholar from JCDVP, Sirsa

Lekha bhambhu

Assistant Prof. JCDVP Sirsa

Abstract- Image Segmentation is the process of partitioning a digital image into multiple regions or sets of pixels. Edge Detection is one of the main Technique used in Segmentation. In this paper we used Sobel edge detector for segmenting the dental X-ray image. Using MATLAB, Image is segmented. Sysytem Test tool is used for the verification of the Simulink Model. The Simulink Model based Image Segmentation is a new function in image processing and offers a model based design for processing. Dental Caries is the main problem occurred in the teeths. Segmentation help to identify the places where the problems of dental caries are present.

Keywords: Segmentation, Simulink, MATLAB, System Test, Dental Caries.

I. INTRODUCTION

Segmentation of an image entails the division or separation of the image into regions of similar attribute. The basic attribute for segmentation is image amplitude-luminance for a monochrome image and color components for a color image. Image edges and textures are also useful attributes for segmentation. The result of image segmentation is a set of regions that collectively cover the entire image, or a set of contours extracted from the image. Segmentation does not involve classifying each segment. The segmentation only subdivides an image; it does not attempt to recognize the individual segments or their relationships to one another. A major goal of image segmentation is to identify structures in the image that are likely to correspond to scene objects. Current approaches to segmentation mainly rely on image-based criteria, such as the grey level or texture uniformity of image regions, as well as the smoothness and continuity of bounding contours. In this work we describe a segmentation method that is guided primarily by high-level information and the use of class-specific criteria. The motivation for using such class-based criteria to supplement the traditional use of image-based criteria in segmentation has two parts. First, it stems from the fact that although recent image-based segmentation algorithms provide impressive results, they still often fail to capture meaningful and at times crucial parts. Second, evidence from human vision indicates that high-level, class-based criteria play a crucial role in the ability to segment images in a meaningful manner, suggesting that the incorporation of such methods will help improve the results of computer vision segmentation algorithms.

II. VARIOUS CRITERIA FOR IMAGE SEGMENTATION

Segmentation subdivides an Image into its constituent regions or objects. The level to which the subdivisions are carried depends on the problem being solved. That is, segmentation should stop when the objects of interest have been isolated. For example, in the automated inspection of electronic assemblies, interest lies in analyzing images of the products with the objective of determining the presence or absence of specific anomalies, such as missing components or broken connection paths.

Segmentation Algorithms for monochrome images generally are based on one of two basic properties of image intensity values: discontinuity and similarity.

- In the first category, the approach is to partition an image based on abrupt changes in intensity, such as edges (Detecting Intensities).
- The Principal approach in the second category is based on partitioning an image into regions that are similar according to a set of predefined criteria (Detecting discontinuities in Image).

The approach is to partition an image based on abrupt changes in intensity, such as edges in an image. Three basic types of gray-level discontinuities that are mostly detected in a digital image are: points, lines and edges. For detecting the three basic types of intensity discontinuities in a digital image is:-

1. Point Detection
2. Line Detection
3. Edge Detection

The detection of isolated points embedded in areas of constant or nearly constant intensity in an image is straightforward in principle. Using the mask we can show that an isolated point has been detected at the location on which the mask is centered if

$$|R| \geq T$$

Where T is a non-negative threshold. This approach to point detection is implemented in the toolbox using function `imfilter` with the mask.

Line detection is an important step in Image processing and analysis. Lines and edges are features in any scene from simple indoor scene to noisy terrain images taken by satellite. If the algorithm to detect these features is not properly designed then we have to introduce the intermediate step of line/edge completion in between feature extraction and interpretation, which serves to join the disrupted segments. To avoid this step an optimized feature extraction algorithm is necessary. For a line, the gray level is relatively constant along a thin strip. There will be a spike-like cross-section in the ideal case. If the gray levels on either side are same, this looks like an uneven spike if the gray levels on either side differ. When combined with noise and blurring, the cross-section may look like a roof in one dimension.

Edge detection is a well-developed field on its own within image processing. Region boundaries and edges are closely related, since there is often a sharp adjustment in intensity at the region boundaries. Edge detection techniques have therefore been used as the base of another segmentation technique. The edges identified by edge detection are often disconnected. To segment an object from an image however, one needs closed region boundaries. The desired edges are the boundaries between such objects. In an image, an edge is a curve that follows a path of rapid change in image intensity. Edges are often associated with the boundaries of objects in a scene. Edge detection is used to identify the edges in an image. To find edges, you can use the edge function. This function looks for places in the image where the intensity changes rapidly, using one of these two criteria:

- Places where the first derivative of the intensity is larger in magnitude than some threshold
- Places where the second derivative of the intensity has a zero crossing

Edge provides a number of derivative estimators, each of which implements one of the definitions above. For some of these estimators, you can specify whether the operation should be sensitive to horizontal edges, vertical edges, or both. Edge returns a binary image containing 1's where edges are found and 0's elsewhere.

III. PROPOSED MODEL FOR IMAGE SEGMENTATION

The model for image segmentation presents an efficient architecture for Image Segmentation. This architecture offers an alternative through a Graphical User Interface tool MATLAB. Image segmentation can be obtained by using various methods, but the drawback of most of the methods is that they use a high level language for coding. This model focuses on processing an image pixel by pixel and in modification of pixel neighborhoods that can be applied to the whole image. The objective lead to the use of a tool with a high-level graphical interface under the Matlab Simulink based blocks which makes it very easy to handle with respect to other software. The various applications where noise removal, enhancing edges and contours, blurring and so on. This model presents the architecture of filtering images for edge detection with the help of Video and Image Processing blockset. In this there is comparison between Coding technique and Model developed using Simulink. There is many techniques based on Coding for image segmentation but all that techniques require a high level language. So, with the help of Morphological operations, proposed model for Image Segmentation. This Model uses various blocks from 'VIDEO AND IMAGE PROCESSING BLOCKSETS'.

Block	Library	Quantity
Image From File	Video and Image Processing Blockset> Sources	1
Edge Detection	Video and Image Processing Blockset> Analysis & Enhancement	1
Minimum	Video and Image Processing Blockset> Statistics	2
Maximum	Video and Image Processing Blockset> Statistics	2
Video Viewer	Video and Image Processing Blockset> Sinks	8
Subtract	Simulink > Math Operations	2
Divide	Simulink > Math Operations	2
Opening	A. Video and Image Processing Blockset> Morphological Operations	1
Label	B. Video and Image Processing Blockset> Morphological Operations	1
Display	Simulink > Sink	2
Noise Source	C. Signal Processing Blockset> Signal Processing Sources> Signal from Workspace	1
Noise Level Control	Simulink > Sink	1
Relational Operator	Simulink >Relational Operator	1
Edge Difference	Simulink > Sink>To workspace	1
Bitwise XOR	Simulink > Logical and Bitwise Operator	1
Inport & Output	Simulink >Commonly Used	1

Table 1:Simulink Blocks used for Implementation

IV. IMPLEMENTATION

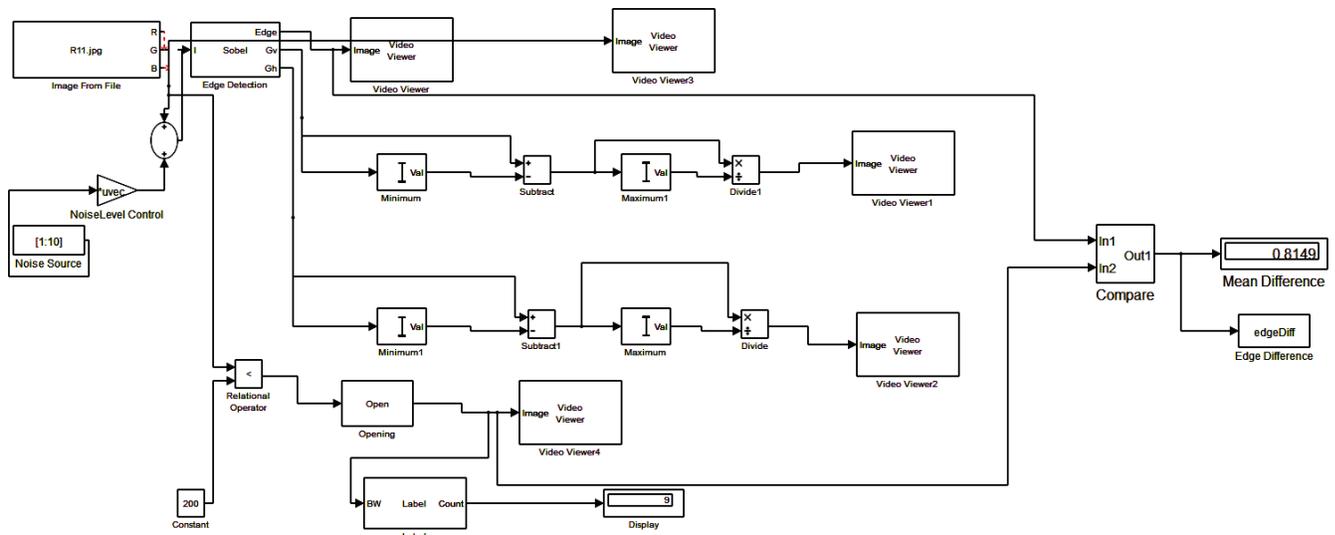


Fig. 1 shows simulink model for sobel edge detection

A.Simulation

The simulation results of the proposed work are presented in this paper. The image metrics like partition coefficient, partition entropy and the percentage of misclassified pixels are used in the chapter to compare between the various existing and proposed algorithms. Extensive qualitative and quantitative analysis is done for comparing the clustering and segmentation results obtained using the different algorithms, under increasing noise condition. The algorithms are tested on synthetic image, real world image and biomedical image. We discuss coding technique, the results and simulation for that technique (Sobel Edge Detector) are discussed below. Coding technique such as:

In an image, an edge is a curve that follows a path of rapid change in image intensity. Edges are often associated with the boundaries of objects in a scene. Edge provides a number of derivative estimators, each of which implements one of the definitions above. For some of these estimators, you can specify whether the operation should be sensitive to horizontal edges, vertical edges, or both. Edge returns a binary image containing 1's where edges are found and 0's elsewhere. The most powerful edge-detection method that edge provides is the Sobel method. The Sobel method differs from the other edge-detection methods in that it uses two different thresholds (to detect strong and weak edges), and includes the weak edges in the output only if they are connected to strong edges. This method is therefore less likely than the others to be fooled by noise, and more likely to detect true weak edges.

B.Creation of Histogram

Histogram is used for showing the comparison between the simulink based model and the coding based model. The no of pixels defines that the proposed model gives better result in segmentation.

C.Testing the algorithm using SystemTest tool

System Test tool is used for the verification of the Simulink model. The System Test software provides MATLAB and Simulink users with a framework that integrates software, hardware, simulation, and other types of testing in one environment. It uses predefined elements to build test sections that simplify the development and maintenance of standard test routines. The System Test software provides MATLAB and Simulink users with a framework that integrates software, hardware, simulation, and other types of testing in one environment. It uses predefined elements to build test sections that simplify the development and maintenance of standard test routines. There are four primary stages of testing: Planning, Building, Running the test, Viewing test results.

In order to perform the test we use some Golden reference values. The model must match the results produced by the golden reference within an absolute tolerance level of 10%. The test cases for which this condition must be met are:

- Threshold values ranging from 365 to 535
- Noise levels ranging from 0 to 100

Using System Test, test vectors are created representing this range of threshold and noise level values. The elements are used to perform the testing are: Simulink Element, Limit Check Element, General Plot Element.

For each main test iteration, the following information is saved as a result for post-processing by specifying them under Save Results:

- Measured difference between both algorithm implementations
- The pass/fail value determined by the Limit Check element

The Results of system Test are displayed on the Run Status. A test Report and Test Results file gives the Output generated by the System test Tool. Test Report Contains the information about running iterations with various Test Variables. By clicking on the Test Results File the output is displayed on the Command window of MATLAB. stresults gives the Test Results Object Summary for the running model. Whereas stresults.ResultsDataSet is used to display test results data. Using this information Graph are Plotted between various test Parameters.

V. RESULTS

Figure 2 shows original image used for segmentation. Figure 3 and figure 4 shows output of coding based model and simulink model. Figure 5, 6 and 7 shows the histogram of the above three figures which helps for the comparison between the various images. Figure 8 shows the output of the System test tool. The graph is plotted between edge difference and noiselevel which was obtained by running the system test tool.



Figure 2: Original Image

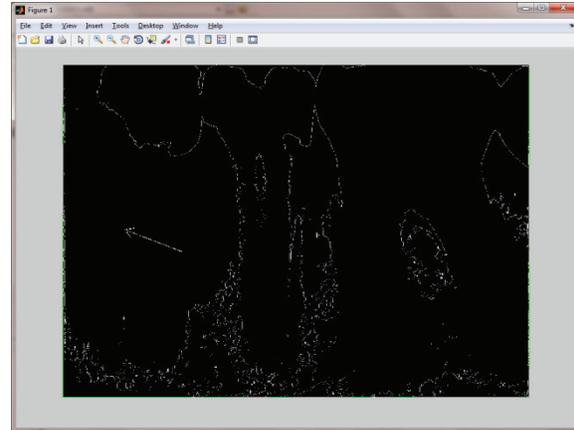


Figure 3: Sobel Edge Detection using MATLAB

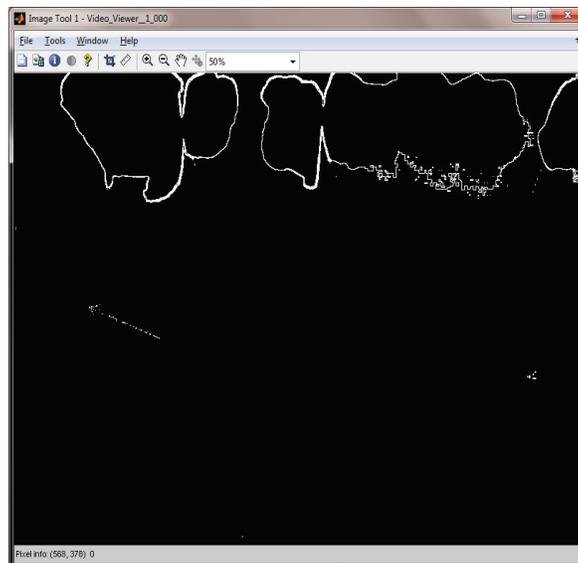


Figure 4.: Proposed Model Output

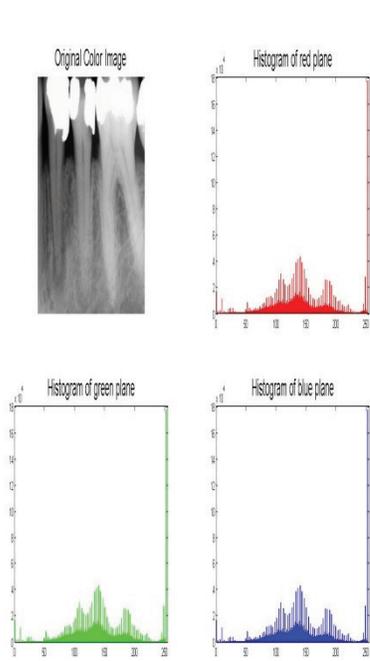


Fig 5: Histogram of Original Image

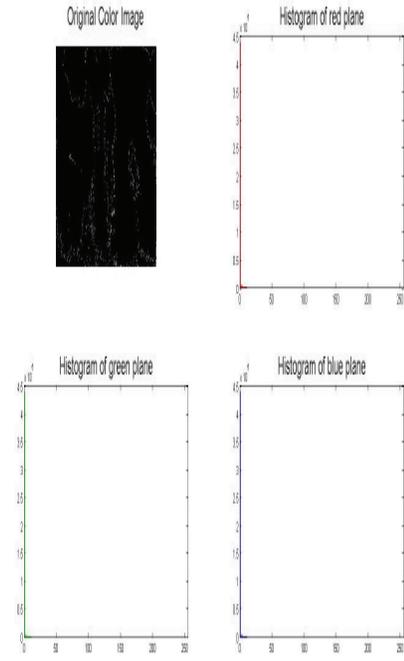


Fig 6: Histogram of Sobel Edge Detector Image using MATLAB

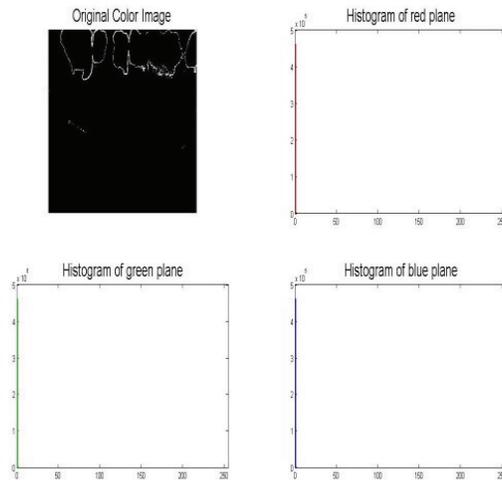


Fig 7: Histogram of Output image for Proposed Model

```

MATLAB 7.10.0 (R2010a)
File Edit Parallel Desktop Window Help
Current Folder: D:\MATLAB\
Shortcuts How to Add What's New
New to MATLAB? Watch this video, see Demos, or read Getting Started.
Undefined function or method 'updateTimeStatus' for input arguments of type 'handle.handle'.
Loading test results....
stresults =

Test Results Object Summary for 'main with running':

NumberOfIterations: 10
TestVectorNames: noiseLevel, threshold
SaveResultNames: PassFail, edgeDiff
ResultsDataSet: [10x4 dataset]

There are no Test Vector Groups associated with this test result object.

Artifacts associated with this test result object:
TEST-File (main with running.test)
Test Report (main with running.report.html)

Type stresults.ResultsDataSet to display test results data. For information
on working with test results data, refer to the Analyzing Test Results demo.

ans =

      noiseLevel      threshold      PassFail      edgeDiff
11 [ 0] [365] [1] [0.3654]
12 [ 20] [365] [1] [0.9875]
13 [ 40] [365] [1] [0.9884]
14 [ 60] [365] [1] [0.9859]
15 [ 80] [365] [1] [0.9814]
16 [100] [365] [1] [0.9798]
17 [ 0] [375] [1] [0.3654]
18 [ 20] [375] [1] [0.9875]
19 [ 40] [375] [1] [0.9884]
20 [ 60] [375] [1] [0.9859]
    
```

Fig 8: System test tool output

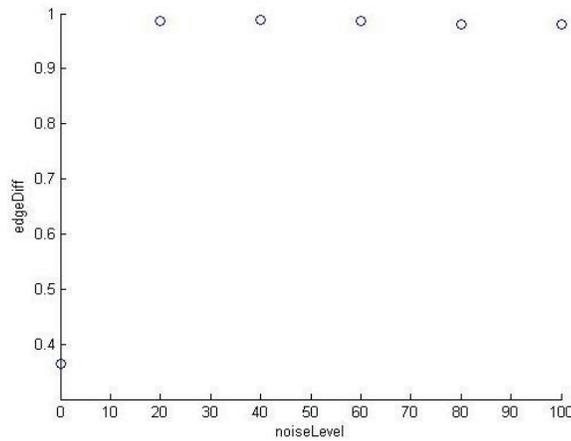


Figure 9: Graph between noiseLevel and edgeDiff

VI. CONCLUSION

This Simulink model focuses on processing an image pixel by pixel and in modification of pixel neighborhoods that can be applied to the whole image. This result gives comparison between Coding techniques and Simulink Model based technique and gives the conclusion that Simulink Based techniques are easy to understand and Implement. Comparisons are done with the help of Histogram. System test tool is used for the verification of the proposed simulink model.

VII. REFERENCES

- [1]. G.T. Shrivakshan, Dr.C. Chandrasekar, "A Comparison of various Edge detection Techniques used in Image Processing", IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 5, No 1, pp. 269-276, September 2012.
- [2]. Zolqernine Othman, Habibollah Haron, Mohammed Rafiq, Abdul Kadir, "Comparison of Canny and Sobel Edge Detection in MRI Images", Computer Science, Biomechanics & Tissue Engineering Group, and Information System, 133-136
- [3]. Law Todd, Itoh Hidenori, "Image filtering, edge detection, and edge tracing", IEEE transactions on pattern analysis and machine intelligence, vol. 18, no. 5, May(1996), pp. 481-491.
- [4]. Orlando J. Tobias and Rui Seara, "Image Segmentation by Histogram Thresholding Using Fuzzy Sets", IEEE Transactions on Image Processing, Vol.11, No.12, 2002, pp. 1457-1465.
- [5]. N. Senthilkumaran and R. Rajesh, "Edge Detection Techniques for Image Segmentation - A Survey", Proceedings of the International Conference on Managing Next Generation Software Applications (MNGSA-08), 2008, pp.749-760.
- [6]. Salem Saleh Al-amri and N.V. Kalyankar, "Image Segmentation By Using Edge Detection", International Journal on Computer Science and Engineering(IJCSE), Vol. 02, No. 03, 2010, pp. 804-807.
- [7]. Rezaei-Rad, G. and M. Aghababaie (2006). Comparison of SUSAN and Sobel Edge Detection in MRI Images for Feature Extraction. In Information and Communication Technologies ICTTA '06. 2nd. pp 1103 - 1106.
- [8]. Samir Shah, Ayman Abaza, Arun Ross, Hany Ammar, "Automatic Tooth Segmentation Using Active Contour without Edges", IEEE 2006 Biometrics Symposium
- [9]. Pedro H. M. Lira, Gilson A. Giraldi, "Panoramic Dental X-Ray Image Segmentation and Feature Extraction".
- [10]. Russo F., "Edge detection in noisy images using reasoning", IEEE transactions on instrumentation and measurement, vol.47, no.5 1998, pp. 1102-1105.
- [11]. Nassir Salman, "Image Segmentation Based on Watershed and Edge Detection Techniques", The International Arab Journal of Information Technology, Vol. 3, No. 2, April 2006.
- [12]. S. Sridhar "Digital Image Processing" 2nd ed, Oxford University Press 2011.
- [13]. Digital Image Processing" by Rafael C. Gonzalez, Richard E. Woods, Steven L. Eddins, Second Edn.
- [14]. Mathworks Inc. "Simulink 3.0", <http://www.mathworks.com/products/Simulink/>.