

# Research And Analysis On Segmentation And Thresholding Techniques

Ms.K.Priyadharshini  
BL.EN.P2EBS11016,  
M.Tech, Embedded Systems  
Amrita School of Engineering, Bangalore

Ms.Tripty Singh  
Assistant Professor  
Department of CSE  
Amrita School of Engineering, Bangalore

**Abstract - This paper discusses on the analysis of various segmentation and thresholding techniques applied on an image. This is achieved by developing a GUI using Matlab tool implementing the techniques on a baby scan image. Various segmentation techniques such as Edge detection, Line Detection, Region-based segmentation and Watershed – Segmentation are applied to the input image to obtain the segmented output image. Similarly two types of global thresholding techniques – Basic Global and Otsu’s method of thresholding are implemented on the input image to obtain the respective output image.**

*Index Terms:* Edge Detection, Line detection, Thresholding, Region-based segmentation, Watershed-segmentation

## 1) INTRODUCTION

Image processing techniques have become more prevalent in almost all applications such as Medical, Defense, Robotics, Remote Sensing etc.

Image Segmentation is an important technique in image processing and, thresholding is a technique which is commonly used for image segmentation. Image segmentation is a process of dividing an image into different regions such that each region is uniform in a certain sense and the union of any two regions is not uniform [2]. The goal of segmentation is typically to locate certain objects of interest which may be depicted in the image.

There are four popular segmentation based approaches: Thresholding techniques, Region-based, Edge- based, Contour – based (Line-based). Watershed – segmentation is a segmentation technique where the segments of images are found by applying transforms.

This paper briefly describes the approaches which are given above. These techniques are useful to find the edges of the image and the form of the image. For a baby scan image the position of the baby and the form of the baby inside the womb can be detected by using these segmentation techniques.

Region growing algorithm is used which grows in the particular shade of the region to partition the image’s darker and lighter areas. The watershed transform finds "catchment basins" and "watershed ridge lines" in an image by treating it as a surface where light pixels are high and dark pixels are low.[3] GUI provides user-interaction and ease of use to achieve this goal.

## 2) STATE – OF – THE - ART

There are previous works which describe on the image segmentation techniques and thresholding techniques. Various papers discuss on the edge detection techniques using operators such as

Sobel, Prewitts, and Roberts etc. and also describe the mathematical formulation of those operators. [4, 5, 6, 7]

Many papers discuss about the thresholding techniques, and also the segmentation applied for 2D images and for obtaining 3D meshes from segmentation. There is description of segmentation techniques used with thresholding and methods of fuzzy sets. [1, 2, 3] [Websites 1, 2, 3]

Some papers discuss on the region-based segmentation techniques which describe both the region growing and region split merge algorithms in detail. These algorithms are based on the continuities and discontinuities in image. This is used to partition the image into regions of darker and lighter shades by identifying a seed region. [2] [Books 1, 2]

Watershed transformation is discussed in various papers, where this technique is used for detection of contours in breast cancer, for partitioning 3D mesh surfaces and so on. This is one technique which uses the concept of “catchment basins”, i.e., partitioning the regions w.r.t “catchment basins”. [8, 9]

## 3) IMAGE SEGMENTATION

As told previously image segmentation can be done by four basic approaches: Edge-based, Contour-based (Line Detection), Region-based and Watershed transformations.

### 3.1) Edge-Based Segmentation

Edge detection is essentially the operation of detecting significant local changes in an image. Edge detection techniques transform images to edge images benefiting from the changes of grey tones in the images. Edges are the sign of lack of continuity, and ending. As a result of this transformation, edge image is obtained without encountering any changes in physical qualities of the main image.

There are five major edge detectors which are most commonly used. They are: 1) Sobel operator, 2) Robert operator, 3) Prewitts operator, 4) LoG (Laplacian of Gaussian), 5) Zero-crossings, 6) Canny Edge operator.

#### 3.1.1) Sobel operator

The Sobel operator performs a 2-D spatial gradient measurement on an image and so emphasizes regions of high spatial frequency that correspond to edges. Typically it is used to find the approximate absolute gradient magnitude at each point in an input gray scale image.

The operator consists of a pair of 3x3 convolution kernel given in Figure.1 as given below. One kernel is simply rotated by 90 degrees. The Sobel mask is given in the Figure.1

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

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

Gx                      Gy

Figure.1.Sobel Mask

3.1.2) Prewitts operator

The Prewitts edge detector is an appropriate way to estimate the magnitude and orientation of an edge. Although differential gradient edge detection needs a rather time consuming calculation to estimate the orientation from the magnitudes in the x and y-directions, the compass edge detection obtains the orientation directly from the kernel with the maximum response.

The Prewitts operator is limited to 8 possible orientations, however experience shows that most direct orientation estimates are not much more accurate. This gradient based edge detector is estimated in the 3x3 neighborhood for eight directions. All the eight convolution masks are calculated. One convolution mask is then selected, namely that with the largest module. The Prewitts mask is given in Figure.2.

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

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

0 degrees 45 degrees

Figure.2.Prewitts Mask

3.1.3) Roberts Operator

The Roberts Cross operator performs a simple, quick to compute, 2-D spatial gradient measurement on an image. It thus highlights regions of high spatial frequency which often correspond to edges. In its most common usage, the input to the operator is a gray scale image, as is the output. Pixel values at each point in the output represent the estimated absolute magnitude of the spatial gradient of the input image at that point.

1	0
0	-1

1	0
0	-1

Gx                      Gy

Figure.3.Roberts Mask

3.1.4) LoG (Laplacian of Gaussian) operator

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 gray level image as input and produces another gray level image as output.

The Laplacian  $L(x, y)$  of an image with pixel intensity values  $I(x, y)$  is given by:

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

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. Two commonly used small kernels are shown in Figure 4.

1	1	1
1	-8	1
1	1	1

-1	2	-1
2	-4	2
-1	2	-1

Figure.4.LoG Mask

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.

Thus the fundamental characteristics of the Laplacian of Gaussian edge detector are:

1. The smoothing filter is a Gaussian.
2. The enhancement step is the second derivative (Laplacian in two dimensions).
3. The detection criterion is the presence of a zero crossing in the second derivative with a corresponding large peak in the first derivative.
4. The edge location can be estimated with sub-pixel resolution using linear interpolation.

3.1.5) Zero-crossings Edge Detector

Zero-crossings edge detector is a special type of Laplacian detector where the filter can be any type of filter specified by the user. It finds the edges by looking for zero-crossings after filtering with the specified filter.

3.1.6) Canny Edge Detector

Canny technique is very important method to find edges by isolating noise from the image before it finds the edges of image, without affecting the features of the edges in the image and then applying the tendency to find the edges and the critical value for threshold.

The algorithmic steps for canny edge detection technique are follows:

1. Convolve image  $f(r, c)$  with a Gaussian function to get smooth image  $f^\wedge(r, c)$ .  
 $f^\wedge(r, c) = f(r, c) * G(r, c, \sigma)$
2. Apply first difference gradient operator to compute edge strength then edge magnitude and direction are obtain as before.

3. Apply non-maximal or critical suppression to the gradient magnitude.

4. Apply threshold to the non-maximal suppression image.

### 3.2) Contour – Based (Line detection) segmentation:

Line-detection algorithms based on contour extraction usually involve the following stages of image processing: smoothing, edge detection, edge thinning, edge linking, chain straightening and line correction. The three last stages process the lines extraction.

The line detection operator consists of a convolution kernel tuned to detect the presence of lines of a particular width  $n$ , at a particular orientation  $\theta$ .

Figure. 5 shows a collection of four such kernels, which each respond to lines of single pixel width at the particular orientation shown.

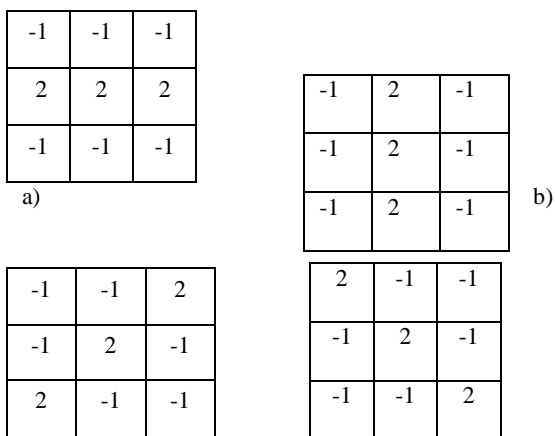


Figure. 5. Four line detection kernels which respond maximally to horizontal, vertical and oblique (+45 and - 45 degree) single pixel wide lines.

#### 3.2.1) Hough Transform

The Hough transform is a technique which can be used to isolate features of a particular shape within an image. Because it requires that the desired features be specified in some parametric form, the classical Hough transform is most commonly used for the detection of regular curves such as lines, circles, ellipses, etc. A generalized Hough transform can be employed in applications where a simple analytic description of a feature(s) is not possible. Due to the computational complexity of the generalized Hough algorithm, we restrict the main focus of this discussion to the classical Hough transform. Despite its domain restrictions, the classical Hough transform retains many applications, as most manufactured parts (and many anatomical parts investigated in medical imagery) contain feature boundaries which can be described by regular curves. The main advantage of the Hough transform technique is that it is tolerant of gaps in feature boundary descriptions and is relatively unaffected by image noise.

The Hough technique is particularly useful for computing a global description of a feature(s) given (possibly noisy) local measurements. The motivating idea behind the Hough technique for line detection is that each input measurement indicates its contribution to a globally consistent solution. A convenient equation for describing a set of lines uses *parametric* or *normal* notion:

$$x\cos\theta + y\sin\theta = r,$$

Where  $r$  is the length of a normal from the origin to this line and  $\theta$  is the orientation of  $r$  with respect to the X-axis.

#### 3.2.2) Hough lines

The algorithm for finding Hough lines is given by:

1. The input image  $E$  is binary array with edge pixels marked with ones and other pixels marked as zeroes. Let be the arrays containing the discretized intervals of the parameter space.

2. Discretize the parameter spaces of  $\rho$  and  $\theta$  using sampling steps,  $\delta\rho$  and  $\delta\theta$ , yielding acceptable and manageable resolution of  $R, T$  in the parameter space

3. Let  $A(R, T)$  be the counter array, initialized as zeroes

4. For each pixel  $E(i, j) = 1$ , and for  $h=1 \dots T$

- Let  $\rho = i\sin\theta_d(h) + j\cos\theta_d(h)$

- Find index  $k$  so that  $\rho_d(k)$  is closest to  $\rho$

- Increment  $A(k, h)$  by one

5. Find all local maxima  $(kp, hp)$  such that  $(kp, hp) > \tau$ , where  $\tau$  is a user defined threshold.

6. The output is a set of lines described by  $(\rho_d(kp), \theta_d(hp))$ .

#### 3.2.3) Hough Peaks

Hough peaks are mainly used to identify the peaks in the Hough transformation matrix,  $H$ . The peaks suppress other peaks that are close to a peak just detected, and that if peaks are too weak, they will not be detected. These behaviors can be controlled with parameters.

#### 3.3) Region-based Segmentation

A region-based method usually proceeds as follows: the image is partitioned into connected regions by grouping neighboring pixels of similar intensity levels. Adjacent regions are then merged under some criterion involving perhaps homogeneity or sharpness of region boundaries. Over stringent criteria create fragmentation; lenient ones overlook blurred boundaries and over merge.

The two techniques in region-based segmentation are: Region Growing and Region Split and Merge.

##### 3.3.1) Region Growing

Region-growing approaches exploit the important fact that pixels which are close together have similar gray values.

This is based on the concept of "Start with a single seed pixel, and add new pixels slowly". The algorithm is given by:

- (1) Choose the seed pixel
  - (2) Check the neighboring pixels and add them to the region if they are similar to the seed
  - (3) Repeat step 2 for each of the newly added pixels; stop if no more pixels can be added.
- Region growing algorithm for an image with the formulation is given by:
- (1) Partition the image into initial seed regions  $R(0) i$  (e.g., split the image in  $7 \times 7$  regions).
  - (2) Fit a planar model to each seed region. If  $E(R(0) i, a, m)$  is small enough, accept  $R(0) i$  and its model; otherwise reject  $R(0) i$ .
  - (3) For each region, find all points that are *compatible* with the region by considering the neighbors of the region (i.e., to avoid outliers).

$C(k) = \{ (x, y) : (g(x, y) - f(x, y, a, m)) < \epsilon \text{ and } (x, y) \text{ is a 4-neighbor of } R(k) \}$

(4) If there were no compatible points, then  $m = m + 1$ . If  $m > M$ , do not grow  $R(k)$  further; otherwise, go to step 3.

(5) Form the new region

$R(k+1) = R(k) \cup C(k)$ , refit the model to  $R(k+1)$ , and compute  $E(R(k+1), a, m)$

(6) Compute the difference error:

$$\rho(k) = E(R(k+1), a, m) - E(R(k), a, m)$$

(7) If  $\rho(k) < T1$ , go to step 3.

(8)  $m = m + 1$ ; If  $m > M$ , do not grow the region further.

(9) Refit the region at the new model  $f(x, y, a, m)$ . If the error of fit decreases, accept the new model and go to step 3; otherwise, do not grow the region further.

### 3.3.2) Region Split-merge:

Regions split and merge algorithm works on iteratively splitting and merging regions to form the "best" image segmentation.

The opposite approach to region growing is region shrinking (splitting). It is a top-down approach and it starts with the assumption that the entire image is homogeneous. If this is not true, the image is split into four sub images. This splitting procedure is repeated recursively until the image is split into homogeneous regions.

If the original image is square  $N \times N$ , having dimensions that are powers of 2 ( $N = 2^n$ ): All regions produced by the splitting algorithm are squares having dimensions  $M \times M$ , where  $M$  is a power of 2 as well ( $M = 2^m, M \leq n$ ). Since the procedure is recursive, it produces an image representation that can be described by a tree whose nodes have four sons each. Such a tree is called a Quad tree.

The split/merge algorithm is given as below:

1. If a region  $R$  is inhomogeneous ( $P(R) = \text{False}$ ) then the region is split into four sub regions
2. If two adjacent regions  $R_i, R_j$  are homogeneous ( $P(R_i \cup R_j) = \text{TRUE}$ ), they are merged
3. The algorithm stops when no further splitting or merging is possible

### 3.4) Watershed – Segmentation

The watershed algorithm derives its name from the manner in which regions are segmented into catchment basins. The purpose of the watershed transformation is to determine the watershed lines on a topographic surface.

The steps of the watershed segmentation algorithm are as follows:

1. Compute the curvature (or some other height function) at each vertex.
2. Find the local minima and assign each a unique label.
3. Find each flat area and classify it as a minimum or a plateau.
4. Loop through plateaus and allow each one to descend until a labeled region is encountered.
5. Allow all remaining unlabeled vertices to similarly descend and join to labeled regions.
6. Merge regions whose watershed depth is below a preset threshold.

Watershed segmentation is achieved by three different transforms in this paper: Distance Transform, Gradient Transform and Marker Controlled Segmentation.

#### 3.4.1) Distance Transform

To use watershed to segment the connected blobs, use distance transforms (DTs) to preprocess the image to make it suitable for watershed segmentation.

Defining the DT of a binary image as the distance from every pixel of the object component (black pixels) to the nearest white pixel. There are many different ways to define the distance between two pixels  $[i1, j1]$  and  $[i2, j2]$  in a digital image. Several commonly used DT functions for image processing are:

a) Euclidean:

$$d_e(i1, j1, i2, j2) = \sqrt{(i1 - i2)^2 + (j1 - j2)^2}$$

b) City Block:

$$d_{cb}(i1, j1, i2, j2) = \text{mod}(i1 - i2) + \text{mod}(j1 - j2)$$

c) Chess Board

$$d_{chb}(i1, j1, i2, j2) = \max(\text{mod}(i1 - i2), \text{mod}(j1 - j2))$$

#### 3.4.2) Gradient Transform:

This is used to pre-process a gray-scale image prior to using the water-shed transform for segmentation. The gradient magnitude has high pixel values along object edges and low pixel values everywhere else.

#### 3.4.3) Marker-controlled Segmentation

Marker-controlled watershed segmentation follows this basic procedure:

1. Compute a segmentation function. This is an image whose dark regions are the objects to segment.
2. Compute foreground markers. These are connected blobs of pixels within each of the objects.
3. Compute background markers. These are pixels that are not part of any object.
4. Modify the segmentation function so that it only has minima at the foreground and background marker locations.
5. Compute the watershed transform of the modified segmentation function.

### 4) THRESHOLDING TECHNIQUES

Threshold is one of the widely methods used for image segmentation. It is useful in discriminating foreground from the background. By selecting an adequate threshold value  $T$ , the gray level image can be converted to binary image. The binary image should contain all of the essential information about the position and shape of the objects of interest (foreground).

Threshold techniques can be categorized into two classes: *global threshold and local (adaptive) threshold*. In the global threshold, a single threshold value is used in the whole image. In the local threshold, a threshold value is assigned to each pixel to determine whether it belongs to the foreground or the background pixel using local information around the pixel.

This paper uses only the global thresholding techniques: Basic global thresholding and Otsu's method of thresholding. The thresholded image  $G(x, y)$  can be defined as

$$G(x, y) = \begin{cases} 1 & \text{if } f(x, y) > T \\ 0 & \text{if } f(x, y) < T \end{cases}$$

#### 4.1) Basic Global Thresholding

The basic global thresholding technique follows the algorithm as below

A) Select an initial estimate for T.

B) Segment the image using T. This will produce two groups of pixels. G1 consisting of all pixels with gray level values  $>T$  and G2 consisting of pixels with values  $\leq T$ .

C) Compute the average gray level values mean1 and mean2 for the pixels in regions G1 and G2.

D) Compute a new threshold value  $T = (1/2)(\text{mean1} + \text{mean2})$

E) Repeat steps 2 through 4 until difference in T in successive iterations is smaller than a predefined parameter  $T_0$ .

#### 4.2) Otsu Method

The basic idea for Otsu method is to find the threshold that minimizes the weighted within-class variance.

The assumptions made in Otsu's model are:

- Histogram (and the image) is bimodal.
- There is no use of *spatial coherence*, nor any other notion of object structure.
- Assumes stationary statistics, but can be modified to be locally adaptive.
- Assumes uniform illumination (implicitly), so the bimodal brightness behavior arises from object appearance differences only.

#### 5) DATA ACQUISITION

An image of a baby in womb is taken for analysis of segmentation. The original image is a bitmap color image of size 462x290. The input color image is taken as a part of the original image. The gray level image of the original image is given as input for most of the segmentation and thresholding techniques. The gray level image is converted to binary image and given as input for certain techniques.

The output image, after the application of appropriate functions displays either a segmented image, thresholded image or the extracted lines from the image.

The functions to obtain the segmentation and thresholding are described above. These functions are implemented directly through the Image Processing Toolbox in MATLAB. The flowchart for data acquisition is given in figure.6.

#### 7) DEVELOPMENT OF GUI

The GUI is developed using the GUI tool in the MATLAB. The GUI figure contains 3 axes for display of images. The push buttons are designed for obtaining the original image, gray scale image and histogram of the original image. The original image is display in axes1, the gray scale image is display in axes2 and all the other outputs are displayed in axes3.

Three pop-up menus are configured to select the edge detection, line detection and thresholding respectively. The pop-up menus are coded with the "switch - case" statements. The region-based segmentation and water-shed segmentation are configured as button groups, with radio-buttons. These button groups are also coded with "switch-case" statements. The "Apply" push button is used with Edge detection techniques' pop-up menu. The static text is used for "Edge detection", "Line Detection" and "Thresholding".

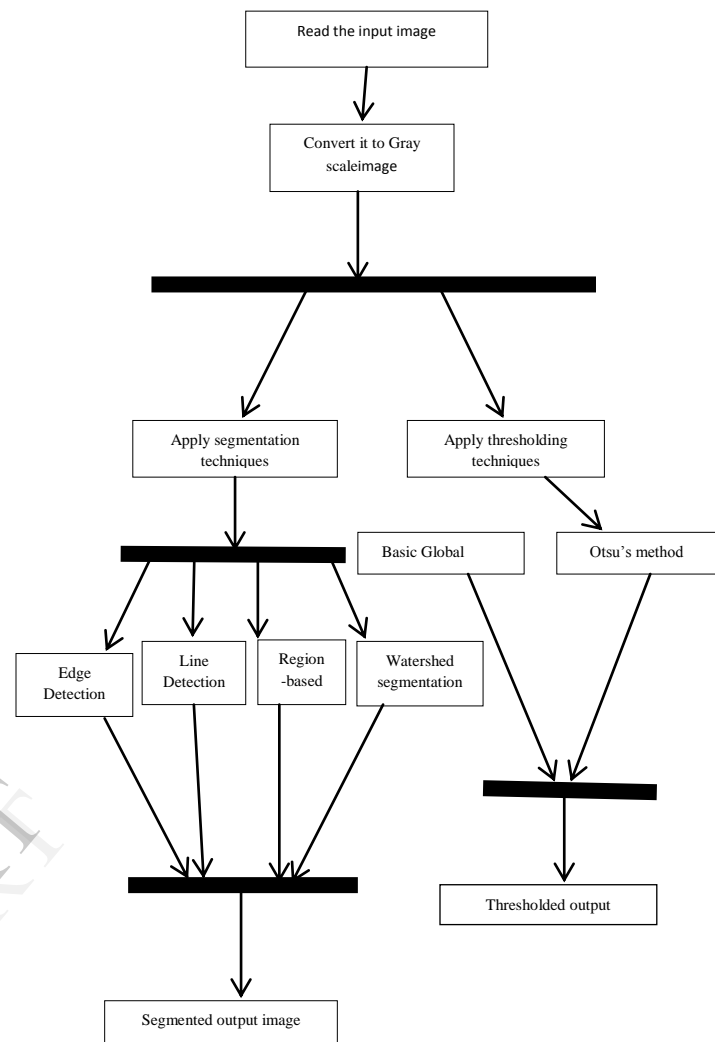


Figure.6. Flowchart for data acquisition

#### 8) RESULT AND CONCLUSION

The Edge detection techniques with Roberts, Prewitts, and Sobel etc. operators are applied to the gray level image with the threshold applied to it implicitly and the operators are applied from the MATLAB functions prescribed.

Similarly the line detection techniques are applied to identify the Hough lines and peaks using the Hough transform. Hough lines are implemented in such a way that it finds the longest connected line in that image.

The region-growing algorithm is implemented by writing a user defined function which identifies the regions in the image by identifying the continuities in both x and y- axis directions.

The watershed transformation is implemented with Distance transform, Gradient transform which also uses the Sobel edge detector and the marker-controlled segmentation.

The Global thresholding techniques are applied to the image and Otsu's method uses the gray level thresholding to get the thresholded image.

Finally the paper has tried to implement and establish the various segmentation techniques on the baby scan image to produce various results. Image segmentation being an important aspect in various applications, and especially in Medical applications, a GUI which is user-interactive is designed to apply these segmentation and thresholding techniques to apply and view the results easily.

The PSNR,MSE and Max error values are analysed for the output images of various techniques in the following table using the EZW compression method.

Image	PSNR(Peak Signal to Noise Ratio)	MSE(Mean Square Error)	Max Error	Nb. Encoding loops
Original image	33.27	30.61	58	5
Gray image	28.4	93.99	186	5
Edge Detection Sobel operator	32.2	39.19	91	5
Prewitts operator	32.74	34.57	81	5
Roberts operator	33.85	26.82	94	5
LoG operator	31.16	49.79	88	5
Zero-crossings	30.61	56.47	97	5
Canny Edge	22.99	326.4	171	5
Line-detection – Hough transform	28.85	84.65	112	5
Hough Peaks	32.31	38.24	82	5
Region growing	24.61	224.9	166	5
Distance transform	24.82	214.5	129	5
Gradient transform	23.26	306.7	156	5
Marker controlled segmentation	27.01	129.4	121	5
Thresholding – Basic Global	25.47	184.3	164	5
Otsu’s method	25.4	187.4	134	5

Table.1. PSNR,MSE and Max error values



Figure.7. Original Image

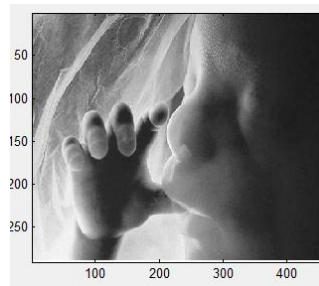


Figure.8.Gray scaled image

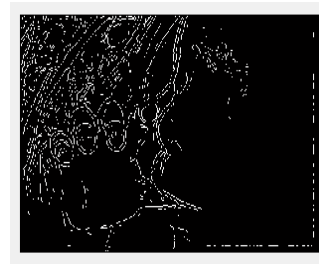


Figure.9. LoG (Laplacian of Gaussian)

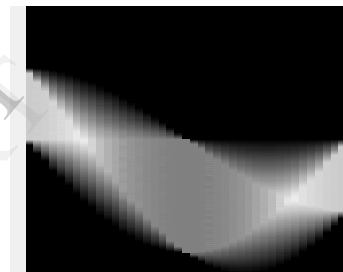


Figure.10. Hough peaks

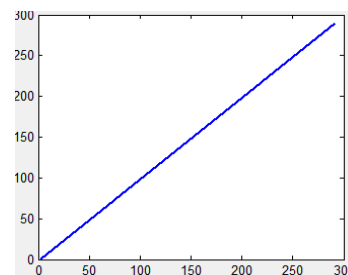


Figure.11.Hough Lines

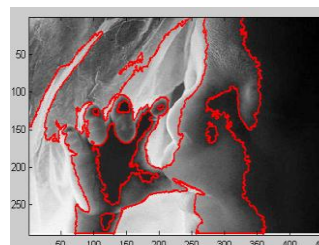


Figure.12. Growing Region



Figure.13.Region Growing-Thresholded image



Figure.14. Thresholded image (Otsu)



Figure.15. Marker controlled watershed segmentation

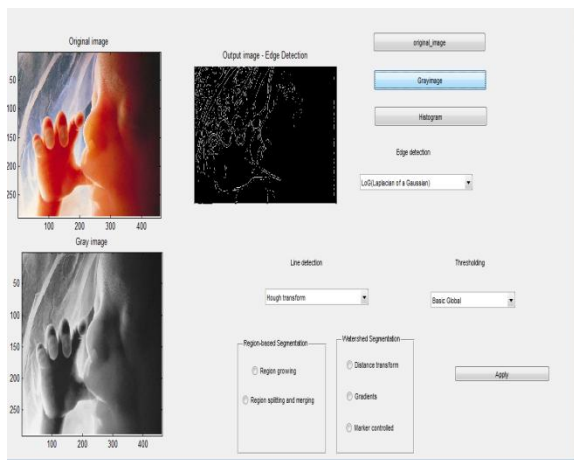


Figure.16.GUI – Edge Detection

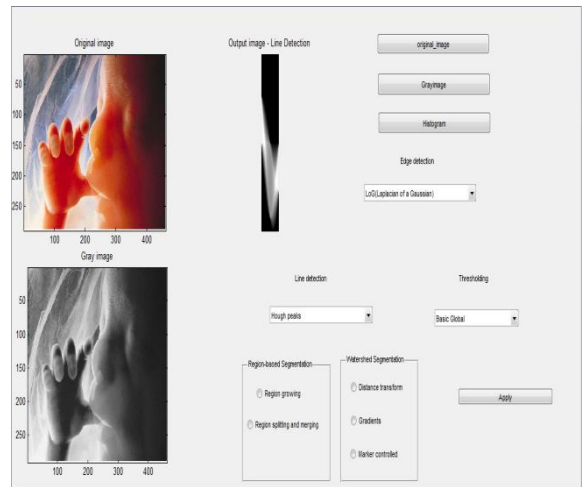


Figure.17. GUI – Line Detection

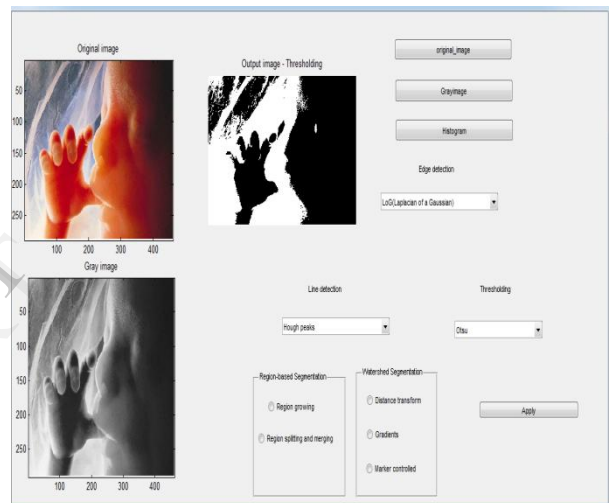


Figure.18. GUI – Thresholding

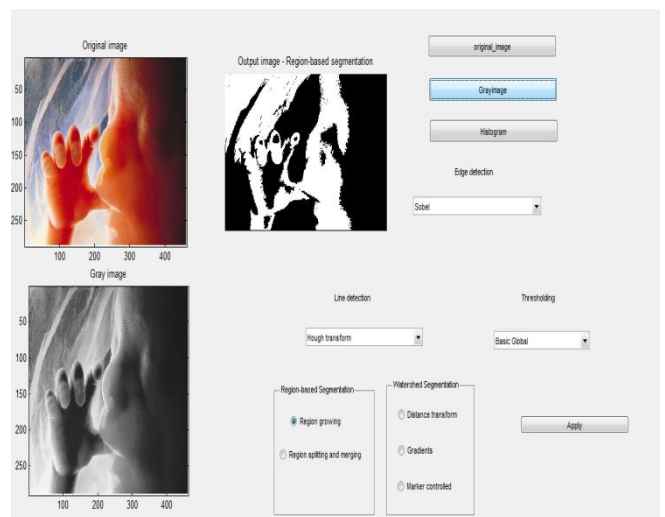


Figure.19. GUI – Region –based segmentation



Figure.20. GUI – Watershed – Segmentation

## REFERENCES

### Links:

- 1) A.D. Jepson and D.J. Fleet, 2007 “Image Segmentation.
- 2) S. C. Zhu and A. Yuille, “Region competition: Unifying snakes, region growing, and bayes/mdl for multiband image segmentation,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 18, pp. 884–900,1996.
- 3) N.R. Pal, S.K. Pal, A review on image segmentation techniques, *Pattern Recognition* 26 (1993) 1277–1294
- 4) L. Alvarez, P.L. Lions, J.M. Morel, Image selective smoothing and edge detection by nonlinear diffusion, II, *SIAM J. Numer. Anal.*, Vol. 29, 845–866, 1992.
- 5) N. Senthilkumar, R. Rajesh, "Edge Detection Techniques for Image Segmentation and A Survey of Soft Computing Approaches", *International Journal of Recent Trends in Engineering*, Vol. 1, No. 2, PP.250-254, May 2009.
- 6) Raman Maini & Dr. Himanshu Aggarwal “Study and Comparison of Various Image Edge Detection Techniques “*International Journal of Image Processing (IJIP)*, Jan-Feb 2009, Volume (3) Issue (1) pp.1-11.
- 7) Salem Saleh Al-amri, Kalyankar and Khamitkar S.D,”Image Segmentation by using edge detection”, *International Journal on Computer Science and Engineering*, Volume2, 2010, Pages: 804-807.
- 8) Mangan, A., Whitaker, R.: Partitioning 3D surface meshes using watershed segmentation. *IEEE Trans. Visual. Compute Graph* 5(4), 308–321 (1999)
- 9) Y. L. Huang and D. R. Chen, “Watershed segmentation for breast tumor in 2-D sonography,” *Ultrasound Med. Biol.* 30, 625–632 (2004).

### Websites:

- 1) <http://en.wikipedia.org/>
- 2) [www.cse.unr.edu/~bebis/CS791E/Notes/](http://www.cse.unr.edu/~bebis/CS791E/Notes/)
- 3) [http://vplab.iitm.ac.in/courses/CV\\_DIP/PDF/lect-Segmen.pdf](http://vplab.iitm.ac.in/courses/CV_DIP/PDF/lect-Segmen.pdf)

### Books:

- 1) R. C. Gonzalez and R. E. Woods. *Digital Image Processing*, 2nd ed., Pearson Education, 2000.
- 2) R. C. Gonzalez and R. E. Woods. *Digital Image Processing Using Matlab*, 2nd ed., Pearson Education, 2000.