A Tree-based Approach Towards Edge Detection of Medical Image using MDT

Indra Kanta Maitra\textsuperscript{1}, Sanjay Nag\textsuperscript{2}, Pradip Saha\textsuperscript{2} and Samir K. Bandyopadhyay\textsuperscript{2}

\textsuperscript{1}\textit{B. P. Poddar Institute of Management and Technology, Kolkata, India}
\textsuperscript{2}\textit{Department of Computer Science and Engineering, University of Calcutta, India}

ikm.1975@yahoo.com, sanjaynag75@gmail.com, pradip.saha93@gmail.com and skb1@vsnl.com

Abstract

To develop any autonomous computerized algorithm for abnormality detection based on medical image analysis essentially requires a robust segmentation algorithm. To achieve this goal of accurate segmentation of medical image it requires an effective edge detection algorithm. Generalized edge detection algorithms like Sobel, Canny, though effective in normal images but often fails in case of medical images. Medical images obtained from devices such as Ultrasonography, X-Ray, CT and MRI exhibit diverse image characteristics but are essentially collection of intensity variations from which specific abnormalities are needed to be isolated. Homogeneity enhancement, followed by edge detection and resulting segmentation is essential prerequisite for all medical image analysis system. This paper proposes a robust medical image enhancement and edge detection algorithm that is equally effective with most medical images that are popular now-a-days yielding excellent segmentation results in all cases.

Keywords: Radiograph, Mammogram, CT-scan, MRI, DICOM, Full and Complete Tree, Histogram, Average Bin Distance (ABD), Maximum Difference Threshold (MDT), Prominent Bins, Prominent Intensity Distances

1. Introduction

Medical imaging has been undergoing a revolution in the past decade with the advent of faster, more accurate and less invasive devices. This has driven the need for corresponding software development which in turn has provided a major impetus for new algorithms in signal and image processing. Edge detection is an important part of image preprocessing aimed to their segmentation and automatic recognition of their contents. Medical image analysis is critical in numerous biomedical applications such as detection of abnormalities, tissue measurement, surgical planning and simulation, and many more. In particular, image segmentation is an essential step, which partitions the medical image into different non-overlapping regions such that each region is nearly homogeneous and ideally corresponds to some anatomical structure or region of interest. It is the main tool in pattern recognition, object recognition, image restoration, image segmentation, and scene analysis. An edge detector is principally a high-pass filter that can be applied to extract the edge points in an image.

Process of identification of sharp discontinuities of an image is called edge of an image i.e., edges are significant local changes of intensity. Here discontinuities mean abrupt changes of pixel intensity of image in a scene. Thus intensity causes basically geometric events and non-geometric events; geometric events basically discontinuity in depth and/or color and texture i.e., object boundary and discontinuity in surface and/or color and texture i.e., surface boundary and non-geometric events basically direct reflection of light called specularity and inner reflection or shadows from other object or same object. In high level image vision, edge detection is used in the interpretation of 3D objects from 2D
images obtained from an image occlusion in radiological imaging. The goal of edge detection is to produce a continuous line drawing of a scene from an image of that scene. Important features can be extracted from the edges of an image (e.g., corners, lines, curves) and these features are used by higher-level computer vision algorithms (e.g., recognition) for analysis.

Medical images are images of the human body or parts of the body intended for clinical purposes for revealing or diagnosis of disease in medical science. Digital X-Ray, mammogram, Ultrasound (USG), Computed Tomography (CT), Magnetic Resonance Imaging (MRI) are some well accepted imaging techniques used for clinical diagnosis. The quality and characteristics of images obtained depend on the different sensors, parameters set by the operators and individual characteristics of the patients. Generalized feature extractions from such images are difficult as they involve diverse technologies. The only commonality among all images is the intensity features exhibited by them. DICOM (Digital Imaging and Communication in Medicine) image format is universally well accepted for all above imaging technologies. In general, the grayscale image pixel is represented by 8 bits in DICOM format having 256 ($2^8$) grayscale color intensities. In this research paper a new edge detection algorithm for grayscale medical images using DICOM format has been proposed. In the later part of this citation contains a description of popular edge detection techniques. The third section contains proposed edge detection method, and in the fourth section results of the proposed method are discussed. Final section contains the conclusion of this citation.

2. Brief Review

All edge detection algorithms are not involved with intensity change. Effects such as refraction or poor focus can result in objects with boundaries defined by a gradual and continuous change in intensity. Choosing a specific operator is dependent on responsiveness to such gradual change in intensity. Contemporary wavelet-based techniques describe the nature of the transition for each edge in order to distinguish edges. There are many edge detection methods which are based on the gradients in the image. The methods return non-zero values in the uneven regions that typically occur on the boundary between two diverse regions in an image. There are large numbers of edge detection operators available, each designed to be responsive to certain particular types of edges. The majority of different methods may be grouped into two categories: i) The gradient method detects the edges by looking for the maximum and minimum in the first derivative of the image, ii) the Laplacian method searches for zero crossings in the second order derivative of the image to find edges.

The gradient is a vector which has certain magnitude and direction and the magnitude of gradient gives information about the strength of the edge and the direction of gradient is always perpendicular to the direction of the edge. This was implemented in Sobel operator [1, 2].

$$\nabla f = \left( \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right)$$

(1)

Two separate kernels are designed to respond maximally to edges running vertically i.e., North-South direction and horizontally i.e., East-West direction relative to the pixel grid. The kernels can be applied concurrently to the input image and can then be combined together to obtain the absolute magnitude of the gradient at each point and the orientation of that gradient. The gradient magnitude is given by:

$$\text{mag}(\nabla f) = \sqrt{(\partial f/\partial x)^2 + (\partial f/\partial y)^2} = \sqrt{M_x^2 + M_y^2}$$

(2)

The angle of orientation of the edge giving rise to the spatial gradient is given by:

$$\text{dir}(\nabla f) = \tan^{-1} \left( M_x/M_y \right)$$

(3)
To compute magnitude, it can be also being written as:

\[
\text{mag}(\nabla f) = |M_x| + |M_y|
\]  

(4)

The Roberts Cross operator \[1][3\] performs a simple and quick calculation based on 2-D spatial gradient measurement on an image. Each output pixel values represent the magnitude of the spatial gradient of the input image at that point. One kernel is perpendicular rotation of the other kernel. This is very similar to the Sobel operator.

The angle of orientation of the edge giving rise to the spatial gradient is given by:

\[
dir(\nabla f) = \tan^{-1}\left(\frac{M_x}{M_y}\right) - \frac{3\pi}{4}
\]  

(5)

Prewitt operator \[1][4\] again is similar to the Sobel operator and is used for detecting vertical and horizontal edges in images. If the arrangement of pixels about the pixel \((i, j)\) are:

\[
\begin{array}{ccc}
    a_0 & a_1 & a_2 \\
    a_7 & [i,j] & a_3 \\
    a_6 & a_5 & a_4
\end{array}
\]

Then the partial derivatives can be computed by:

\[
M_x = (a_2 + ca_3 + a_4) - (a_0 + ca_7 + a_6)
\]  

(6)

\[
M_y = (a_6 + ca_5 + a_4) - (a_0 + ca_1 + a_2)
\]  

(7)

The constant \(c\) implies the emphasis given to pixels closer to the center of the mask.

The Laplacian of Gaussian \[1][5\] is a 2-D measure of the 2nd order spatial derivative of an image. The rapidly changing regions are identified and hence they are useful for edge detection. The Laplacian is often applied to an image that is smoothed with a Gaussian Smoothing in order to reduce its sensitivity to noise.

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

\[
L(x,y) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}
\]  

(8)

Since these kernels resemble a second derivative measurement on the image, they are very sensitive to noise. To avoid misclassification of noise elements as edge the image is blurred using Gaussian smoothing operator before applying the Laplacian filter. This pre-processing step reduces the high frequency noise components prior to the differentiation step.

The Laplacian of Gaussian \[5\] kernel can be pre-calculated in advance so only one convolution needs to be performed at run-time on the image. The 2-D LoG function centered on zero and with Gaussian standard deviation \(\sigma\) has the form:

\[
\text{LoG}(x,y) = -\frac{1}{\pi\sigma^4} \left[ 1 - \frac{(x^2 + y^2)}{2\sigma^2} \right] e^{-\frac{(x^2 + y^2)}{2\sigma^2}}
\]  

(9)

The Canny edge detection algorithm \[6\] is known to many as the optimal edge detector and has shown that the first derivative of the Gaussian closely approximates the operator that optimizes the product of signal-to-noise ratio and localization. The algorithm then tracks along these regions and suppresses any pixel that is not at the maximum. The steps for canny edge detection is follows, Compute \(f_x\) and \(f_y\)

\[
f_x = \frac{\partial}{\partial x} (f * G) = f * \frac{\partial}{\partial x} G = f * G_x
\]  

(10)

\[
f_y = \frac{\partial}{\partial y} (f * G) = f * \frac{\partial}{\partial y} G = f * G_y
\]  

(11)

\(G(x, y)\) is the Gaussian function

\(G_x(x, y)\) is the derivate of \(G(x, y)\) with respect to \(x\):

\[
G_x(x,y) = \frac{\partial_x}{\partial^2} G(x,y)
\]  

(12)

\(G_y(x, y)\) is the derivate of \(G(x, y)\) with respect to \(y\):
\[
G_y(x, y) = \frac{-y}{\sigma^2} G(x, y)
\]  
(13)

The performance of the canny algorithm [6] depends heavily on the adjustable parameters, \(\sigma\), which is the standard deviation for the Gaussian filter, and the threshold values, ‘T1’ and ‘T2’. \(\sigma\) also controls the size of the Gaussian filter. The greater the value for \(\sigma\), the larger the size of the Gaussian filter. This indicates more blurring which is necessary for noisy images and also can detect larger edges. Gradient-based algorithms such as the Prewitt filter have a major drawback of being very sensitive to noise. The size of the kernel filter and coefficients are fixed and cannot be adapted to a given image. An adaptive edge-detection algorithm is necessary to provide a robust solution that is adaptable to the varying noise levels. The complexity of Canny edge detection algorithm is high compared to Sobel, Prewitt and Robert’s operator but the algorithm performs better than all these operators under almost all scenarios.

The Frei-Chen [7] edge detection method determines edge points on the size of the angle between the sub-image \(b\) and its projection on the edge sub-space. The Frei-Chen edge detector actually uses a combination of 4 pairs of wavelets. Two pairs are the first degree partial derivatives of the Gaussian smoothing function, which are sensitive to step edges and the other two pairs are second and fourth mixed partial derivatives of the Gaussian smoothing function, which are sensitive to Dirac edges.

3. Proposed Method

Any grayscale image is represented as a two-dimensional array of pixel intensities. A grayscale image can be expressed as a combination of \(k\) intensity values with a certain frequency \(f(k)\) where \(k = 0\) to \(n\). In this research paper a new structure is proposed to represent images using a modified full and complete binary tree that will accommodate both the intensity and frequency measures. The objective in constructing such a tree is to obtain an image with reduced number of color, yet maintaining the full color palette; thus achieving color quantization at every tree level.

A binary tree can be defined as full binary tree if the entire node contains exactly two child nodes or it is leaf node and all leaf nodes must contains in the same level. Similarly a binary tree \(T\) with \(n\) levels is complete if all levels except possibly the last are completely full, and the last level has at least all its nodes to the left side. But in case of full and complete tree it must satisfy both the conditions laid down by their definitions. Hence it is possible to obtain a tree that is complete at all levels having all nodes and all child nodes; at the last level have only leaf node.

The proposed data structure, all the possible colour of an image can be represented by the leaf nodes of the said data structure i.e. if the image contains \(2^n\) number of distinct colours then the tree will have \(2^n\) leaves at level \(n\). To represent a 256 grayscale DICOM image, it is required to construct a full and complete binary tree with leaf nodes at level 8 (as \(2^8\) equals 256). The node structure of the said binary tree will contain pointers for left and right child along with image data. The data will have three components i.e. colour intensity, its frequency present in the image and a balancing factor.

The frequency of intensity of left child node \(f(L)\) and right child node \(f(R)\), whichever is greater, will be the intensity of the parent node. The frequency for the node will be the summation of \(f(L)\) and \(f(R)\). The balancing factor will be determined by its level \(l\) and total number of colour intensity \(C\), for left node it will be balancing factor of parent node \((C-2^l)\) and for right it will be balancing factor of parent node.

Initially, a tree is created having level 8 with balancing factor as the only image data. At level 8 the value of balancing factor equals the intensity value; hence the balancing factor value is copied to the intensity data. The remaining two data fields of image data are empty in the beginning except the leaf nodes. Then the image bitmap is read in row major order. Each pixel’s intensity will be compared with the balancing factor of each
node starting from the root up to the last level. Every time it compares the intensity value with the balancing factor, to check whether it has a value greater or lower than the balancing factor. If it has a value lower than the balancing factor it moves on to the left child node for comparison. Similarly, if it is equal or greater than the balancing factor then it moves on to the right child node for comparison. Every time it will traverse a node the frequency value for that node will be incremented. At the end of the reading process, the tree contains frequency value of image data set for all the nodes in the tree, with the root node having the total pixel count whereas every leaf having the frequency value of that particular intensity.

The entire tree is constructed but only the leaf has all their image data set. In the next phase, the intensity value of image data for all the intermediate nodes including the root node are required to be calculated. As per the proposed algorithm the parent node will hold the intensity value of the child node that has a greater frequency among the two child nodes. To achieve the same the tree has to be traversed in postorder manner. According to the postorder traversal of tree the left child node and the right child node is traversed before the parent node, making it possible to compare the intensity frequencies of the child nodes. The intensity value of the parent node is updated based on the comparison results of the child nodes. This process is continued till the intensity value of the image data for root node is achieved.

The tree structure that is obtained by the above procedure contains the histogram of the original image with 256 grey shades at level 8 and in every subsequent upper level contains 2ⁿ number of grey shades. The intensity values of intermediate nodes can contain any grey shade value, thus preserving the original colour palette and performing uniform colour quantization at every level. The manifestation of different intensity in different medical imaging technologies, are divergent. Even similar type of medical image may show different intensity characteristics, for example, MRI of brain and knee will exhibit significant intensity differences. Appropriate level of the aforesaid data structure can be utilized to accommodate different cases depending on the intensity characteristics to obtain optimal result. Moreover, to highlight or segment out a portion of any image intensity, different parts of multiple levels can be utilized. Say for example, to isolate a tumor that exhibit higher intensities than the rest of the image, it is possible to take a lower level for those high intensity parts, thus taking greater number of colour shades. For the rest of the image intensities a higher level is used with reduced number of colour shades.

Here some of the basic functions are defined which are used in later algorithms:

Parent (i)

Return \left\lfloor i/2 \right\rfloor

Left (i)

Return 2i

Right (i)

Return 2i+1

// Total number of nodes in a Complete Binary Tree
tNode (h)

Return \text{2}^{h+1}

// The number of terminal nodes (leaf nodes) in a Complete Binary Tree
lNode (h)

Return \text{2}^{h+1}
3.1. Generation of Image Histogram

Image histogram is a graphical representation of the intensity distribution in a digital image. It exhibits the number of pixels for each intensity value. In mathematical term, a histogram is a function that counts the number of observations that fall into each of the disjoint categories. Let \( n \) be the total number of observations and \( k \) be the total number of disjoint categories, the histogram \( H_i \) meets the following:

\[
n = \sum_{i=1}^{k} H_i
\]

In this proposed method, colour intensities and their frequencies are extracted from the original image colour space to generate histogram. The extracted data are stored in the leaf nodes of proposed tree i.e. each leaf node representing individual disjoint intensity sequentially. ORIGINAL-HISTOGRAM is the process to obtain the histogram from the source medical image. Here, the leaf node is starting from \( iNode (h) + 1 \) to \( tNode (h) \).

**Figure 1. Original MRI along with Histogram of Level 0**

**Algorithms for:** Storing Original Colour Space at Leaf Nodes of Tree

```
ORIGINAL-HISTOGRAM (Image, height, width)
Loop x← 1 to height
    Do Loop y← 1 to width
        Do
            Intensity ← Image [x, y]
            Tree [(iNode (h) + 1) + Intensity].count ← Tree [(iNode (h) + 1) + Intensity].count + 1
        End
        x←x +1
        y←y +1
    End
Return Tree
```

**Correctness**

Loop invariant: At start of every iteration of outer loop, each row of image \( x = 1, 2, \ldots, n \).
Initialization: Since $x = 1$ i.e. it is at first row of the image before the first iteration of the outer loop, so, the invariant is initially true. It is at the position $\text{Image}[1][y]$.

Maintenance: In each successive iteration loop invariant moves to next row by incrementing $x$. Loop works by moving $\text{Image}[x+1][y]$, $\text{Image}[x+2][y]$, $\text{Image}[x+3][y]$ and so on.

Termination: The outer loop ends when $x>\text{height}$, i.e. all the row of the image is already traversed.

**Complexity Analysis**

Assuming the height = width = $n$, the running time of the algorithm is $\Theta(n^3)$ for all cases.

### 3.2. Generation of Level Histogram

LEVEL-HISTOGRAM is an important process to generate colour quantised histograms for each subsequent level of tree with reduces number of colour i.e. half number of colours upto the root in bottom-up mode. The way LEVEL-HISTOGRAM works:

- It will work for all leaf nodes i.e. $\text{iNode}(h)+1$ to $\text{tNode}(h)$.
- Compare frequency count of Tree $[i]$ with child nodes i.e. $\text{Tree}[\text{Left}(i)]$ and $\text{Tree}[\text{Right}(i)]$, intensity of the larger will be propagated to parent intensity and sum of frequency count of children will be the frequency count of parent.
- The iteration will be terminated when the parent node is the root of tree.

**Figure 2. Level 1, Level 2 and Level 3 Histograms**

**Algorithms for**: generate quantize colour spaces in different level of tree

**LEVEL-HISTOGRAM (Tree)**

**Loop** $x \leftarrow \text{iNode}(h)+1$ to $\text{tNode}(h)$
- **Do** $\text{Lcount} \leftarrow \text{Tree}(x).\text{count}$
  - **Loop** $y \leftarrow \text{Parent}(x)$ downto 0
    - **Do** **If** $x \mod 2 \neq 0$
Then  \( \text{Tree}[y].\text{intensity} \leftarrow \text{Tree}[x].\text{intensity} \)
\( \text{Tree}[y].\text{count} \leftarrow \text{Tree}[y].\text{count} + \text{Lcount} \)

Else  If  \( \text{Tree}[y].\text{count} < \text{Tree}[x].\text{count} \)
Then  \( \text{Tree}[y].\text{intensity} \leftarrow \text{Tree}[x].\text{intensity} \)
\( \text{Tree}[y].\text{count} \leftarrow \text{Tree}[y].\text{count} + \text{Lcount} \)

\( x \leftarrow y \)
\( y \leftarrow \text{Parent}(x) \)
\( x \leftarrow x + 1 \)

Return Tree

Correctness

Loop invariant: At start of every iteration of outer loop, each leaf node of tree \( x = \text{iNode}(h) + 1, \text{iNode}(h) + 2, \ldots, \text{tNode}(h) \).

Initialization: Since \( x = \text{iNode}(h) + 1 \) i.e. it is at first leaf of the tree before the first iteration of the outer loop, so, the invariant is initially true.

Maintenance: In each successive iteration, loop invariant moves to next leaf by incrementing \( x \). Loop works by moving \( \text{iNode}(h) + 1, \text{iNode}(h) + 2, \text{iNode}(h) + 3 \) and so on.

Termination: The outer loop ends when \( x > \text{tNode}(h) \), i.e. all the leaves of the image is already traversed.

Complexity Analysis

Assuming that the Tree with height \( h \), the number of leaves is \( \text{lNode}(h) = 2^h \), so, the outer loop will execute \( 2^h \) number of times. To traverse from a leaf to root, \( \lg (h-1) \) number of iteration is required. The inner loop will be executed \( 2^h. \lg (h - 1) \) times. So, the running time of the algorithm is \( \Theta(2^h. \lg (h - 1)) \) for all cases.

3.3 Average Bin Distance (ABD) Calculation

In each level, half of intermediate colours bin are truncated depending on the condition stated above. This intermediate truncation will generate different bin distance in the histogram of particular level of the Tree. The process BIN-DISTANCE will calculate the average bin distance of that histogram. The Average Bin Distance (ABD) is the mean of different bin distance in the histogram. The way BIN-DISTANCE has been calculated.

- It will work for a particular height \( (h_1) \) of the Tree.
- The iteration will start from first node i.e. \( \text{iNode}(h_1) \) up to the last node i.e. \( \text{tNode}(h_1) \) of particular height of the Tree.
- Summing the intensity distance between all adjacent nodes sequentially and calculate average bin distance (ABD) dividing by the number of bins.

Algorithms for: Calculate the Average Bin Distance

BIN-DISTANCE (Tree, \( h_1 \))
\( \text{TotBin} \leftarrow 0 \)
\( \text{TotBinDist} \leftarrow 0 \)
\( \text{Loop} x \leftarrow \text{iNode}(h_1) + 2 \) to \( \text{tNode}(h_1) \)
\( \text{Do} \)
\( \text{TotBin} \leftarrow \text{TotBin} + 1 \)
\( \text{TotBinDist} \leftarrow \text{TotBinDist} + (\text{Tree}[x].\text{intensity} - \text{Tree}[x - 1].\text{intensity}) \)
\( x \leftarrow x + 1 \)
\( \text{AvgBinDist} \leftarrow \text{TotBinDist} / \text{TotBin} \)
\( \text{Return} \; \text{AvgBinDist} \)
Correctness

Loop invariant: At start of every iteration of loop, each node of tree for a particular height \( h_1 \), \( x = iNode(h_1) + 2, iNode(h_1) + 3, \ldots, tNode(h_1) \).

Initialization: Since \( x = iNode(h_1) \) i.e. it is at first node of the tree for a particular height \( h_1 \) before the first iteration of the loop, so, the invariant is initially true.

Maintenance: In each successive iteration, loop invariant moves to next node by incrementing \( x \). Loop works by moving \( iNode(h_1) + 3, iNode(h_1) + 4 \) and so on.

Termination: The loop ends when \( x \leq tNode(h_1) \), i.e. all the leaves of the image is already traversed.

Complexity Analysis

Assuming that the Tree with height \( h \), the number of node is \( tNode(h) - iNode(h) \), so, the loop will execute \( tNode(h) - iNode(h) \) number of times. It is for the leaf node of the Tree, so, loop will be executed in maximum times. For intermediate height of Tree loop will be executed in lesser number of times. Therefore, the upper bound of the running time will be \( O(2^h) \).

3.4 Calculation Maximum Difference Threshold (MDT)

In previous process, ABD of histogram of particular level of the Tree has been derived. CALCULATE-MDT process first segregates the bins into two categories namely, Prominent Bins and Truncated Bins. Prominent Bins are the points of histogram from where sharp change intensity values are recorded. Whereas, Truncated Bins has an insignificance difference of intensity with its adjacent bins. Prominent bins have a significant role to determine edges of an image. Using the prominent bins, CALCULATE-MDT process generates the Maximum Difference Threshold (MDT). The outline of CALCULATE-MDT:

- It will work for a particular height \( h_1 \) of the Tree.
- The iteration will start from first node i.e. \( iNode(h_1) \) upto the last node i.e. \( tNode(h_1) \) of particular height of the Tree.
- It will compare the intensity difference of a node with its previous node with ABD. If intensity difference is greater than the ABD, it will be marked as prominent node else it will be marked as truncated.

![Figure 3. Level 2 Histogram showing Prominent Bins along with other bins](image)
Algorithms for: Calculation of MDT by Identifying the Prominent Bins and Truncate the Non-Prominent Bins

CALCULATE-MDT (Tree, \(h_1\))

Tree [iNode (\(h_1\)) + 1].prominent ← 1

TotPrmBin ← 0

TotPrmBinDist ← 0

Loop \(x ← iNode (h_1) + 2\) to tNode (\(h_1\))

Do If Tree [\(x\)].intensity - Tree [\(x - 1\)].intensity ≥ AvgBinDist

Then Tree[\(x\)].prominent ← 1

TotPrmBin ← TotPrmBin + 1

TotPrmBinDist ← TotPrmBinDist + (Tree [\(x\)].intensity - Tree [\(x - 1\)].intensity)

Else Tree[\(x\)].prominent ← 0

\(x ← x + 1\)

MDT ← TotPrmBinDist / TotPrmBin

Return MDT

Correctness

Loop invariant: At start of every iteration of loop, each node of tree for a particular height \(h_1\), \(x = iNode (h_1) + 2, iNode (h_1) + 3, \ldots, tNode (h_1)\).

Initialization: Tree [iNode (\(h_1\)) + 1].prominent ← 1, the points of histogram from where sharp change intensity values are recorded set to 1. TotPrmBin and TotPrmBinDist are set to 0 for counting total prominent bin and total prominent bin distance.

Maintenance: In each successive iteration, loop invariant moves to next node by incrementing \(x\). Loop works by moving iNode (\(h_1\)) + 3, iNode (\(h_1\)) + 4 and so on.

Termination: The loop ends when \(x ≤ tNode (h_1)\), i.e. all the leaves of the image is already traversed which is finite number.

Complexity Analysis

Assuming that the Tree with height \(h\), the number of node is tNode (\(h\)) - iNode (\(h\)), so, the loop will execute tNode (\(h\)) - iNode (\(h\)) number of times. Including Inner conditions, loop will be executed in maximum times for all leaf node of the Tree. For intermediate height of Tree loop will be executed in lesser number of times. Therefore, the upper bound of the running time will be \(O (2^h)\).

3.5 Generation of Enhanced Image

In this process, the enhanced image will be generated from the original medical image. The New Intensity of a particular pixel has been calculated using the truncated histogram from the tree of a particular level. Initially mapping between original histogram at level (\(h-1\)) and the desired level histogram at level \(h_1\) has been done. From mapping process a particular intensity has been selected from the level histogram and checking has been performed weather the obtained intensity is prominent or not. If the obtained intensity is prominent one then it will be propagated to the image pixel, else the next higher prominent intensity will be selected from the level histogram to propagate.
Figure 4. Enhanced MRI and Graph Showing Prominent Intensity Distances in Level 2

**Algorithms for:** Redraw the Image Using Truncated Histogram

\[
\text{REDRAW - IMAGE (Image, height, width, Tree, h, h)}
\]

**Loop** \( x \leftarrow 1 \) to height

\[ \text{Do Loop } y \leftarrow 1 \text{ to width} \]

\[ \text{Do NewIntensity } \leftarrow (\text{Image}[x, y] / (\text{tNode}(h) / \text{tNode}(h_1))) + 1 \]

\[ \text{If Tree[iNode}(h_1) + \text{NewIntensity + 1}.\text{prominent} \neq 1 \]

\[ \text{Then While Tree[iNode}(h_1) + \text{NewIntensity + 1}.\text{prominent} \neq 1 \]

\[ \text{Do NewIntensity } \leftarrow \text{NewIntensity} - 1 \]

\[ \text{NewImage}[x, y] \leftarrow \text{NewIntensity} \]

\[ y \leftarrow y + 1 \]

\[ x \leftarrow x + 1 \]

**Return** NewImage

**Correctness**

Loop invariant: At start of every iteration of loop, inner loops maximum limit width of the image and outer loop maximum limit height of the image.

Initialization: height and width are set by the height and width of the image.

Maintenance: In each successive iteration, loop invariant moves to pixel of the image incrementing 1. Inner Loop works by moving 1, 2, 3,..., width and outer Loop works by moving 1, 2, 3,..., height.

Termination: Inner loop width of the image, but check only Tree[iNode(h1) + NewIntensity + 1].prominent not equals to 1 i.e. maximum width.

**Complexity Analysis**

Assuming that the height of the image \( h_t \) and width of the image \( w_t \), then maximum number of iteration of loop is \( h_t \times w_t \). Thus if \( h_t = w_t = n \) then the complexity of the above algorithm is \( O(n^2) \).

**3.6. Generation of Horizontal and Vertical Edge Map**

In the previous process, the enhanced image has been generated by using the level histogram of a desired level. In this part of the algorithms the horizontal edge map i.e. HozEdgeMapImage will be obtained using the aforesaid enhanced image. The process will scan the enhanced image in row major order. It will be started from the left most
pixel from first row and terminated at the right most pixel of the last row. Here two consecutive pixels i.e. NewIntensity and NxtNewIntensity respectively are compared. If the absolute value of the difference is greater than the MDT then the corresponding pixel position of the HozEdgeMapImage image will be set to 0 i.e. black else 255 i.e. white.

**Algorithms for:** derive the Horizontal Edge of the image

**HozEdgeMap** (NewImage, height, width, MDT)

Loop $x \leftarrow 1$ to height

Do flag $\leftarrow 1$

Loop $y \leftarrow 1$ to width

Do If Flag = 1

Then NewIntensity $\leftarrow$ NewImage [$x$, $y$]

NxtNewIntensity $\leftarrow$ NewImage [$x$, $y$]

If |NewIntensity $-$ NxtNewIntensity| $\geq$ MDT

Then Flag $\leftarrow 1$

HozEdgeMapImage [$x$, $y$] $\leftarrow$ BLACK

Else Flag $\leftarrow 0$

HozEdgeMapImage [$x$, $y$] $\leftarrow$ WHITE

$y \leftarrow y + 1$

$x \leftarrow x + 1$

Return HozEdgeMapImage

**Correctness**

Loop invariant: At start of every iteration of loop, inner loops maximum limit limit width of the image and outer loop maximum limit height of the image.

Initialization: Height and width are set by the height and width of the image and a constant flag is set 1 in the outer loop and in the inner loop flag is set by some condition like if (NewIntensity $-$ NxtNewIntensity) $\geq$ MDT then set flag 1 else 0.

Maintenance: In each successive iteration, loop invariant moves to pixel of the image incrementing 1. Inner Loop works by moving 1, 2, 3,…, width and outer Loop works by moving 1, 2, 3,…, height depends on flag value.

Termination: Inner loop width of the image does not execute if flag value is 0, until inner loop executes maximum width and outer loop always performs maximum height of the image.

**Complexity Analysis**

Assuming that the height of the image $h_i$ and width of the image $w_i$, then maximum number of iteration of loop is $h_i * w_i$. Thus if $h_i = w_i = n$ then the complexity of the above algorithm is $O(n^2)$. For vertical edge detection, algorithms is similar to the above horizontal edge detection algorithm except it sets pixels vertically i.e. in column major order rather than horizontally.
In previous two methods horizontal edge map and vertical edge map images has been obtained. In this process the horizontal and vertical edge map are super impose in each other using logical OR operation. The superimposed output image will be the obtained edge of the medical image i.e., EdgeMapImage. It is the final outcome of the research paper which can be applicable to any medical image.

**Figure 5. Horizontal and Vertical Edge Map in Level 2**

**Figure 6. Derived Edge of the MRI Image in Level 2**

**Algorithms for:** derive the Edge of the image

EDGEMAP (HozEdgeMapImage, VerEdgeMapImage, height, width)

Loop \(x \leftarrow 1\) to height

Do Loop \(y \leftarrow 1\) to width

Do EdgeMapImage \([x, y] \leftarrow \text{HozEdgeMapImage} [x, y] \lor \text{VerEdgeMapImage} [x, y]\)

\(y \leftarrow y + 1\)

\(x \leftarrow x + 1\)

Return EdgeMapImage

**Correctness**

Loop invariant: At start of every iteration of loop, inner loops maximum limit width of the image and outer loop maximum limit height of the image.
Initialization: Height and width are set by the height and width of the image and in the inner loop horizontal edge map and vertical edge map are logical OR-ed together.

Maintenance: In each successive iteration, loop invariant moves to pixel of the image incrementing 1. Inner Loop works by moving 1, 2, 3, ..., width and outer Loop works by moving 1, 2, 3, ..., height.

Termination: loop terminates until after of its full scan i.e. inner loops executes maximum width of image and outer loop executes maximum height of the image.

Complexity Analysis

Assuming that the height of the image $h_I$ and width of the image $w_I$, then maximum number of iteration of loop is $h_I \times w_I$. Thus if $h_I = w_I = n$ then the complexity of the above algorithm is $O(n^2)$.

4. Experimental Results

Different medical images like computed tomography (CT scan), mammogram, digital radiograph, MRI are used to obtain the results of the proposed algorithms. Here the proposed method is implemented on 322 number of mammogram images of the MIAS database, 247 number of digital radiograph images of chest lung nodules and non-nodules of the Japanese Society of Radiological Technology (JSRT) in cooperation with the Japanese Radiological Society (JRS), over 500 number of MRI and CT images of DICOM sample image sets provided by OsiriX and other medical images available in public domain. For the experiment purpose different body parts are considered like breast, brain, chest, knee, abdomen, spine etc. to establish the robustness of the method. Some of the outputs of different types of medical images are cited in figure 7 to figure 10. After consulting with experts and by observation, it has been recognised in general that 2nd level decomposition i.e., with 64 colour bin is most acceptable level for further investigation. In this level most of important edges are preserved leaving behind the redundant edges. Further decomposition may lead to loss of significant information related to edge. But at the same time different types of medical images like CT, MRI, Mammogram, Radiograph etc. exhibit best results in different levels depending on the body parts and the technical parameters like contrast etc.
Figure 7. Lung Radiograph: (a) Original Image, (b) Edge Map with level 1 decomposition, (c) Edge Map with level 2 decomposition and (d) Edge Map with level 3 decomposition.

Figure 8. Breast Mammogram: (a) Original Image, (b) Edge Map with level 1 decomposition, (c) Edge Map with level 2 decomposition and (d) Edge Map with level 3 decomposition.
Figure 9. Knee MRI: (a) Original Image, (b) Edge Map with level 1 decomposition, (c) Edge Map with level 2 decomposition and (d) Edge Map with level 3 decomposition
Figure 10. CT of Abdomen with contrast: (a) Original Image, (b) Edge Map with level 1 decomposition, (c) Edge Map with level 2 decomposition and (d) Edge Map with level 3 decomposition

5. Results Evaluation

Performance evaluation in algorithm design is an important step that is commonly neglected. What constitutes an “acceptable” result differs significantly, and is often based on visual subjective opinion with very little quantitative endorsement [8]. The accuracy is measured for our proposed algorithms using quantitative measures by comparing the obtained results representing an abnormal mass, as a “mask” with its equivalent “gold standard”. The "gold standard" is obtained by manually drawing the breast region representing an abnormal mass within each mammogram. The boundary of the regions corresponding to the abnormal breast is manually traced to extract the region and generate a "ground truth (GT) image".

It is needed to derive a quantitative measure to get the accuracy of the proposed segmentation method. The "masks" derived is compared with the GT to obtain the measures of "True Positive (TP)", "False Negative (FN)" and "False Positive (FP)". "True Positive" is denotes the intersection of obtained mask and GT indicating the actual match between the two. "False Negative" is the under-segmented region that is absent in "mask" whereas in "False Positive" it is the over segmented region that is absent in GT. By using these parameters on the output of all the images having abnormalities, the following results are obtained on different quality measures.

Table 1. Common Measures Used in the Evaluation of the Proposed Method

<table>
<thead>
<tr>
<th>Common measures</th>
<th>Computation</th>
<th>Radiograph</th>
<th>Mammogram</th>
<th>MRI</th>
<th>CT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy (Percentage agreement)</td>
<td>$\frac{</td>
<td>TN</td>
<td>+</td>
<td>TP</td>
<td>}{</td>
</tr>
<tr>
<td>Dice similarity coefficient (DSC)</td>
<td>$\frac{2</td>
<td>TP</td>
<td>}{2</td>
<td>TP</td>
<td>+</td>
</tr>
<tr>
<td>Error rate</td>
<td>$\frac{</td>
<td>FP</td>
<td>+</td>
<td>FN</td>
<td>}{</td>
</tr>
<tr>
<td>Sensitivity (Percentage of Correct Estimation)</td>
<td>$\frac{</td>
<td>TP</td>
<td>}{</td>
<td>TP</td>
<td>+</td>
</tr>
</tbody>
</table>
Specificity (True Negative Fraction/Rate) | \(\frac{|TN|}{|TN|+|FP|}\) | 0.9521 | 0.9687 | 0.9609 | 0.9814
--- | --- | --- | --- | --- | ---
False Positive Fraction/Rate | 1 - Specificity | 0.0478 | 0.0312 | 0.0390 | 0.0185
Under estimation fraction (UEF) | \(\frac{|FN|}{|TN|+|FN|}\) | 0.0004 | 0.0003 | 0.0003 | 0.0002
Over estimation fraction (OEF) | \(\frac{|FP|}{|TN|+|FN|}\) | 0.0005 | 0.0004 | 0.0004 | 0.0002

6. Conclusion

Edge detection is very significant step in any image segmentation and analysis algorithm. Proper classification of regions based on local intensity characteristics leads to prominent edges that isolate different anatomical regions as separate objects for further consideration. Presence of noise may hinder the process of edge determination. The proposed algorithm produces edge map image that is single pixel continuous edge line so edge thinning process is eliminated. The continuity of the edge line in the proposed algorithm ensures that process of edge linking is unnecessary. The most significant achievement of this proposed algorithm is that it is universally applicable to a wide range of medical imaging technology. The algorithm is equally suitable for MR, CT, X-Ray, Mammographic as well as Sonographic images that exhibit diverse image characteristics. The dynamicity of the algorithm helps it to adapt to any kind of medical image and intensity nature as it adapts to different image conditions. Moreover, the data-structure accommodates selection at any level of the tree exhibiting reduced number of bin at that level of abstraction. The user has the choice of selecting the number of color bins that he feels appropriate for the image. Higher number of bin will give more dense edges whereas selection of lower bins will give sparse edges. The edge map image is devoid of any noise that is observed in other known methods of edge detection. It can be observed from the edge map image that each region within the breasts, chest, knee, brain etc. are clearly visible and isolated. The added advantage of this edge detection process is that it separated the main object i.e. breasts, knee, brain etc. from the exterior non object region. Experimentation results showed almost cent percent accuracy with almost negligible values of over or under segmentation regions. With this edge detection method, the image segmentation process becomes simple and efficient as it does not require convolution. It is also robust, highly accurate and can be very useful for diverse medical imaging technologies.

Acknowledgement

The authors would like to thank Department of Electronics and Information Technology (DeitY), Government of India, New Delhi for budget support under the project Fully Automated Computer Aided Diagnosis (CAD) System of Human Breast Cancer using Digital Mammogram vide approval no. 1(2)/2012-ME&TMD/ESDA dated 26.03.2014.

References

Authors

**Dr. Indra Kanta Maitra**, he is the Senior System Analyst of B. P. Poddar Institute of Management and Technology, Kolkata. He obtained Ph.D. from University of Calcutta in the year 2015. He is the Co-investigator of the project Fully Automated Computer Aided Diagnosis (CAD) System of Human Breast Cancer using Digital Mammogram sponsored by Department of Electronics and Information Technology (DeitY), Government of India, New Delhi. He received award in 96th Indian Science Congress, 2009 at Shillong for Best Poster in the section of Information and Communication Science and Technology (including Computer Science). He is a Member of IEEE, Computer Society of India and Indian Science Congress Association. He was Managing Committee Member, Nomination Committee Member and Student Coordinator (East) of CSI. He is author of two books and more than 30 publications in National and International Journal and Conference including IEEE, Elsevier, etc. His working areas of specialization are Image Processing, Network Security, Data Structure, Programming Language, Computer Organization and Architecture, etc. He is now working in the field of Biomedical Image Analysis and CAD.

**Sanjay Nag**, he is Research Scholar at University of Calcutta, India. He obtained Master in Computer Application in 2009 from IGNOU and Microsoft Certified Professional (MCP) in 2011. Now he is working as Senior Programmer in the project Fully Automated Computer Aided Diagnosis (CAD) System of Human Breast Cancer using Digital Mammogram. He is the author of more than 10 publications in National and International Journal and Conference including IEEE, Elsevier, etc. His fields of specialization are Biomedical Image Analysis, Image Processing, Data Communication and Network, Software Engineering, etc.

**Dr. Pradip Saha**, he is Director, Roy’s Diagnostic and Imaging Centre and Sevayan Diagnostic and Imaging Centre, Kolkata. He obtained MD (Radio-Diagnosis) from University of Calcutta in the year 1989. Currently he is appointed by University of Calcutta as medical consultant of the project Fully Automated Computer Aided Diagnosis (CAD) System of Human Breast Cancer using Digital...
Mammogram sponsored by Department of Electronics and Information Technology (DeitY), Government of India, New Delhi. Earlier he was Lecturer at R.G. Kar Medical College, Kolkata.

Prof. Samir K. Bandyopadhyay, he is Professor of Computer Science and Engineering, University of Calcutta, India. He obtained his Ph.D. in Computer Science and Engineering in 1989 from University of Calcutta, M.Tech. in Radio-Physics and Electronics in 1979 from University of Calcutta and B.E. in Electronics and Tele-Communication in 1975 from B.E. College, University of Calcutta. Currently he is the Principal Investigator of the project Fully Automated Computer Aided Diagnosis (CAD) System of Human Breast Cancer using Digital Mammogram sponsored by Department of Electronics and Information Technology (DeitY), Government of India, New Delhi. He is the Chairman of Science and Engineering Research Support Society (SERSC, Indian Part), Fellow of Computer Society of India, Sectional President of ICT of Indian Science Congress Association, 2008–2009, Senior Member of IEEE, Member of ACM, Fellow of Institution of Engineers (India), Fellow of Institution of Electronics and Tele-Communication Engineering, India, Reviewer of International Journals IEEE Transactions on Neural Networks, ACM and Springer Publications. His fields of specialization are Bio-medical Engineering, Mobile Computing, Pattern Recognition, Graph Theory, Image Processing, Handwritten Signature Verification, Graphical Password Verification, etc. He has 25 years of Teaching and Research experience in the Post-graduate and under-graduate studies. He published books like Data structure Using C, Addison Wesley, 2003; C Language, Pearson Publication, 2010. He is author of more than 100 publications in National and International Journals and Conferences.