Graph Theory Based Approach For Image Segmentation Using Wavelet Transform

Vikramsingh R. Parihar vikramparihar05@gmail.com
Department of PG Studies (Electrical and Electronics Engg.)
Prof Ram Meghe Collge of Engineering and Management
Badnera-Amravati. 444701, INDIA

Nileshsingh V. Thakur thakurnisvis@rediffmail.com
Department of PG Studies (Electrical and Electronics Engg.)
Prof Ram Meghe Collge of Engineering and Management
Badnera-Amravati. 444701, INDIA

Abstract

This paper presents the image segmentation approach based on graph theory and threshold. Amongst the various segmentation approaches, the graph theoretic approaches in image segmentation make the formulation of the problem more flexible and the computation more resourceful. The problem is modeled in terms of partitioning a graph into several sub-graphs; such that each of them represents a meaningful region in the image. The segmentation problem is then solved in a spatially discrete space by the well-organized tools from graph theory. After the literature review, the problem is formulated regarding graph representation of image and threshold function. The boundaries between the regions are determined as per the segmentation criteria and the segmented regions are labeled with random colors. In presented approach, the image is preprocessed by discrete wavelet transform and coherence filter before graph segmentation. The experiments are carried out on a number of natural images taken from Berkeley Image Database as well as synthetic images from online resources. The experiments are performed by using the wavelets of Haar, DB2, DB4, DB6 and DB8. The results are evaluated and compared by using the performance evaluation parameters like execution time, Performance Ratio, Peak Signal to Noise Ratio, Precision and Recall and obtained results are encouraging.

Keywords: Segmentation, Graph Theory, Threshold, Wavelet Transform.

1. INTRODUCTION

Segmentation is the process of partitioning a digital image into set of pixels or regions. Among the various existing segmentation approaches, graph theoretic approach found to have several good features in practical applications. The graph theoretic approach organizes the image elements into mathematically sound structures. It makes the formulation of the problem more flexible and the computation more resourceful. The problem is modeled in terms of partitioning a graph into several sub-graphs; such that each of them represents a meaningful object of interest in the image. The segmentation problem is then solved in a spatially discrete space by the efficient tools from graph theory [1].

All the existing graph based approaches involves the use of following terminologies. Let $G = (V, E)$ be a graph, where $V= \{v_1, v_2, \ldots, v_n\}$ is a set of vertices corresponding to the image elements, which might represent pixels or components. E is a set of edges connecting pairs of neighboring vertices. Each edge $(v_i, v_j) \in E$ has a corresponding weight $\omega(v_i, v_j)$ which measures a quantity based on the property between the two vertices connected by that edge e.g., color, motion, location or some other local attribute (in our case the difference in intensity). For image segmentation, a segmentation S is a partition of V into components such that each component $C \in S$ corresponds to a connected component in a graph $G'=(V',E')$, where $V' \subseteq V$, $E' \subseteq E$.

A segmentation approach should capture perceptually important components or regions. Now three problems arises as to provide description of what is perceptually important, to specify what a developed segmentation approach does and precise definitions of the properties of a resulting segmentation, in order to better understand the method as well as to facilitate the comparison of different approaches. The segmentation approach should run at speeds similar to edge detection or other low-level visual processing techniques in order to be of practical use. Also the visual quality of segmentation is to be maintained at the same time.

This paper presents the image segmentation approach based on graph theory. The problem is modeled in terms of partitioning a graph into several sub-graphs; such that each of them represents a meaningful region in the image. The segmentation problem is then solved in a spatially discrete space by the well-organized tools from graph theory. The boundaries between the regions are determined as per the segmentation criteria and the segmented regions are labeled with random colors. In presented approach, the image is preprocessed by discrete wavelet transform and coherence filter before graph segmentation. The experiments are carried out on a number of natural images taken from Berkeley Image Database as well as synthetic images from online resources.

The organization of this paper is as follows. Section 2 includes the literature review. The section concludes with our findings from the literature review and motivation behind identified problems. Section 3 focuses on the formulation of the identified problem regarding the graph based representation of image and threshold function. Section 4 is dedicated to the proposed approach; where the working of the graph based algorithm for segmenting an image is described along with its implementation and our contribution to the work. Section 5 emphasize on the experimental results for a number of images along with comparison of the obtained results followed by the thorough discussion about the experimental results. Section 6 addresses the conclusions along with the future work.

2. LITERATURE REVIEW

The earliest graph-based approaches use fixed thresholds and local measures in computing segmentation. Later the focus was moved towards segmenting the image based on minimum spanning tree (MST) of the graph. For image segmentation, the edge weights in the graph are based on the differences between pixel intensities. The segmentation criterion is to break MST edges with large weights. The inadequacy of simply breaking large edges is that it would result in the high variability region being split into multiple regions. The splitting of such highly variable region is inappropriate.

Another class of graph based approaches is introduced [2-5] where the technique primarily focuses on finding minimum cuts in a graph. The cut criterion is designed in order to minimize the similarity between pixels that are being split. This bias is addressed with the normalized cut criterion. These cut-based approaches to segmentation capture non-local properties of the image, in contrast with the early graph-based approaches. However, they provide only a characterization of each cut rather than of the final segmentation.

The normalized cut criterion [6-8] provides a significant advance over the previous works. However, the normalized cut criterion also yields an NP-hard computational problem. In practice these approximations are still fairly hard to compute, limiting the approach to relatively small images.

Later the eigenvector-based approximations [9-10] are related to more standard spectral partitioning approaches on graphs. However, all such approaches are too slow for many practical applications. Also the eigen vector approach captures computationally important groupings or clusters and not according to human perception. Hence, our focus is moved towards another approach.

Pedro F. Felzenszwalb and Daniel P. Huttenlocher [11] using a graph-based representation of the image developed a segmentation algorithm and found that their approach satisfy global properties. The algorithm runs in time nearly linear in the number of graph edges and is also fast in practice. The specialty of the approach is that it is able to preserve detail in low-variability image regions and ignore detail in high-variability regions. Further improvements in [11] are made by Ming Zhang and Reda Alhajj [12] by re-defining the internal difference used to define the property of the components and the threshold function, which is the important factor in determining the size of the components. They claimed the efficiency and effectiveness of the adjusted approach through experimentations. However, no performance evaluation parameter is presented by both [11] and [12].

2.1 Our findings from the Literature Review

- Fixed threshold and local measures cannot be employed for good segmentation as it has many drawbacks.
- Simply breaking the MST edges or edges with high weights would result in improper segmentation.
- The eigen-vector based segmentation approaches are two slow for practical applications and the segments obtained by these approaches are computationally important but perceptually important regions are not obtained.
- The normalized cut criterion provides a significant advance over the previous works. However, the normalized cut criterion also yields an NP-hard computational problem. In practice these approximations are still fairly hard to compute, limiting the approach to relatively small images or requiring computation times of several minutes.
- Graph cuts algorithm based on iterated region merging requires lot of user interaction.
- In the image segmentation based on mean shift and normalized cuts, the spatial structure and the detailed edge information of an image are not preserved.
- If the image is treated as an undirected weighted non-planar finite graph and image segmentation is handled as graph partitioning problem, then the approach could not segment the images having high overlapping of objects or very dark images.
- If weighted Euclidean distance is used to calculate the edge weight, then the efficiency becomes less.
- When the segmentation is done based on the principle that in an Eulerian circuit, each edge is traversed only once and further segregation in open and closed sub-graphs is done by choosing critical vertices at a minimum directed distance, the algorithm itself cannot trace the boundary in images. The input of traced boundary is, thus, to be given; so more user interaction is required.
- When the objects to be segmented contain similar colors with the background, Grab Cut might fail to correctly segment them.
- The iterated region merging-based graph cuts algorithm requires a lot of user interaction.

2.2 Motivation Behind Identified Problem

From the critical analysis of the related work, we find that graph partitioning problem is categorized as NP-hard problem. Since image segmentation can be reduced to graph partitioning therefore it is also a NP-hard problem. Though, the different approaches exist to perform the color image segmentation, no particular approach produce the most efficient segmentation for the given color image. Also there is no standard basis on which an image can be segmented. Therefore, the scope of contribution exists in this area and this motivated us for problem formulation. Our goal is to develop an image segmentation approach that can be broadly useful, just like the other low-level techniques such as edge detection which are utilized in a wide range of computer vision tasks. In order to achieve such broad utility, we believe it is important that a segmentation approach should have the two properties. First is to capture perceptually important groupings or regions, which often reflect global properties of the image. And second is to run the segmentation approach at the speeds similar to edge detection or other low level process. We have developed an approach for image segmentation considering these two factors.

3. PROBLEM FORMULATION

This section presents the formulation of the identified problem, which involves the use of graph based representation of an image along with threshold function. Although, the literature consists of a various approaches to represent an image onto a graph, all the graph theoretic approaches involve the same common terminologies. The problem of graph based segmentation can be formulated as:

- The image is initially mapped on a graph $G = (V, E)$, where, $V= \{v_1, v_2,\ldots, v_n\}$ is a set of vertices corresponding to the image elements, which might represent pixels or regions. E is a set of edges connecting certain pairs of neighboring vertices.
- Each edge $(v_i ,v_j) \in E$ should have a corresponding weight $\omega\ (v_i ,v_j)$ which measures a certain quantity based on the property between the two vertices connected by that edge.
- For image segmentation, an image should be partitioned into mutually exclusive components, such that each component C is a connected graph $G' = (V',E')$ where $V' \subseteq V$, $E' \subseteq E$ and E' contains only edges built from the nodes of V'.
- In other words, non empty sets $C_1,\ldots, C_k$ form a partition of the graph G such that $C_i \cap C_j = \phi\ (i , j \in \{1,2,\ldots, k\}, i \neq j)$ and $C_1 \cup \ldots \cup C_k = G$.
- Although, there are different aspects to measure the quality of segmentation but, in general, it is believed that the elements in a component are supposed to be homogeneous and the elements in different components to be heterogeneous.
- This means that edges between two vertices in the same component should have relatively low weights, and edges between vertices in different components should have higher weights.
- A threshold function is used to manage the extent to which the difference between the components must be larger than the minimum internal difference within each component.
- An approach is to be produced which when there are more components than expected, the threshold function should "encourage" merging. When there are fewer components than expected, the threshold function should "discourage" merging.
- Before graph based segmentation, the image should pass through a filter which will remove noise. However, the edges should be preserved for proper segmentation.
- To increase the speed of computation, some preprocessing should be done on image so that the smaller insignificant regions will be merged and the computational complexity of the graph based segmentation algorithm is thus reduced.
- The segmentation results should be evaluated based on appropriate performance evaluation parameters.
- Finally, the segmentation result of the proposed approach is to be compared.

4. PROPOSED APPROACH

This section is subdivided into three parts wherein the paper presents the working of the graph based representation approach that is incorporated in our work along with our approach. Section 4.1 primarily focuses on the study and working of the graph based representation of image. Our proposed approach is introduced in Section 4.2. Section 4.3 provides detailed working of the proposed approach.

4.1 Graph-Based Segmentation [11]

- Let $G = (V , E)$ be an undirected graph
- Vertices $v_i \in V$, the set of elements to be segmented
- Edges $(v_i , v_j) \in E$ corresponding to pairs of neighboring vertices.
- Each edge $(v_i , v_j) \in E$ has a corresponding weight $\omega\ ((v_i , v_j))$ which is a non-negative measure of the dissimilarity between neighboring elements v_i & v_j.
- In the case of mentioned approach, the elements in V are pixels and the weight of an edge is the difference in intensity between the two pixels connected by that edge.

The systematic working of the graph based approach is demonstrated by means of flow chart, prepared by us, as shown in Figure 1. The input image is initially mapped on a graph $G = (V, E)$ with n vertices and m edges. The output is a segmentation of V into components $S = (C_1 , \ldots\ldots, C_r)$. The edges E are sorted into non decreasing edge weight order $\pi = (o_1, \ldots. , o_m)$. The segmentation is started with S^0, where each vertex v_i is in its own component. S^q is constructed from S^{q-1} as shown below. The following process is repeated for $q = 1, \ldots.., m$. Let v_i and v_j denote the vertices connected by the q^{th} edge in the ordering, i.e., $o_q = (v_i , v_j)$ If v_i and v_j are in disjoint components of S^{q-1} and $\omega(o_q)$ is small compared to the internal difference of both the components, then the two components are merged.

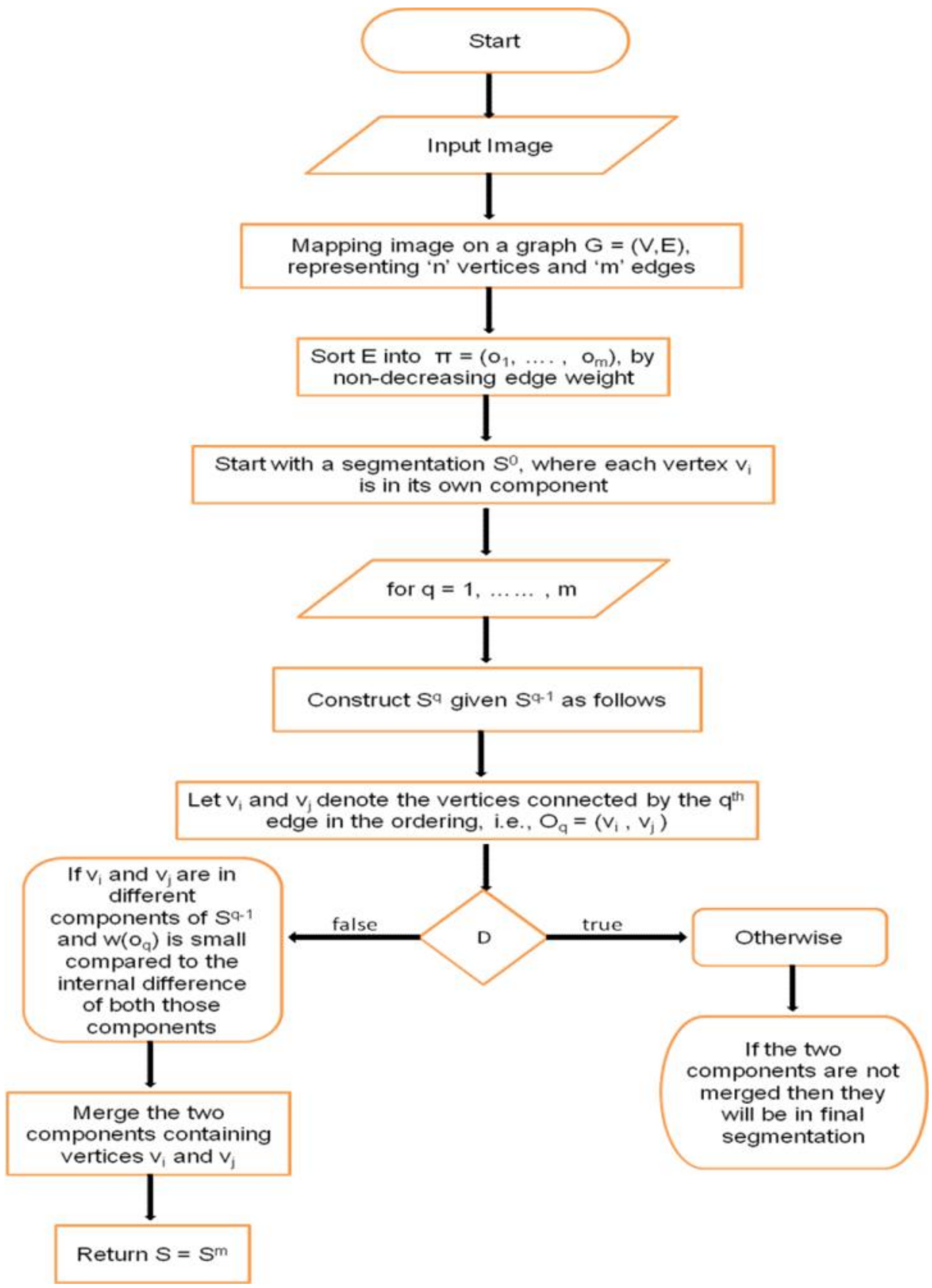


FIGURE 1: Flowchart, prepared by us, for the graph based segmentation approach [11].

In other words, the input image is considered as a graph where the pixels are vertices and the edges connecting two pixels have some weights that are the difference between the intensity values of the two pixels. These edges are initially sorted according to the non decreasing order. The segmentation process is then initialized with the consideration that each vertices belong to its own components. Now the edges connecting two vertices in the neighboring regions are evaluated. Based on the threshold value, the predicate decides whether the two regions have to be merged or to be considered as segmented. If the edges connecting two pixels of different components have less value than the threshold, then the two regions are merged together. If the edges connecting two pixels of different components have equal or larger value than the threshold then the two regions remain separated and are obtained in the final segmentation results. Similar calculation is performed for all the edges and thus the boundaries between the two pixels are determined. The regions are finally labeled with random colors so as to distinguish the adjacent regions. The above process can be interpreted with the help of the flowchart.

4.5 Proposed Approach

From the study of the graph based segmentation approach, it is found that changing the definition to use the median weight, or some other property, in order to make the computation more robust, makes the problem of finding a good segmentation NP-hard. Thus a small change to the segmentation criterion vastly changes the difficulty of the problem. So changing the segmentation criteria is not appropriate. So, while maintaining the segmentation criteria of [11], we carried out experimentations by preprocessing the input image by using wavelets transforms like Haar, DB2, DB4, DB6 and DB8 as well as filtering the image using coherence filter [13]. A number of natural images and synthetic images are used for experimentations. The evaluation of the proposed graph based segmentation approach which includes the execution time, Performance ratio (PR) [14], Precision and Recall and Peak Signal to Noise ratio [PSNR].

4.5.1 Wavelet Transform

Wavelets have the special ability to examine signals simultaneously in both time and frequency. In the DWT, an image is analyzed by passing it through an analysis filter bank. This process is followed by a decimation operation. This analysis filter bank of a low pass and a high pass filter is commonly used in image compression. A signal is split into two bands when it passes through these filters. The coarse information of the signal is extracted by low pass filter which corresponds to an averaging operation. The high pass filter extracts the detail information of the signal which corresponds to a differencing operation. The output of the filtering operations is then decimated by two.

By performing two separate one-dimensional transforms, a two-dimensional transform can be accomplished. Here, initially, the image is filtered along the x-direction using low pass and high pass analysis filters and decimated by two. On the left part of the matrix, low pass filtered coefficients are stored and on the right part of the matrix, high pass filtered coefficients are stored. Later, the same process is followed by filtering the sub-image along the y-direction and decimated by two. On the lower part of the matrix, low pass filtered coefficients are stored and on the upper part of the matrix, high pass filtered coefficients are stored. Finally, the image is split into four bands. These bands are denoted by HH, LH, HL and LL after one-level decomposition.

The following process demonstrated how reconstruction of the image is carried out. Initially, the image is upsampled by a factor of two on all the four subbands at the coarsest scale and filters the subbands in each dimension. Then the four filtered subbands are sum up to reach the low-low subband at the next finer scale. This process is repeated until the image is fully reconstructed.

Among the various wavelet transforms, we carried out experimentations by preprocessing the image by using Haar transform, DB2 transform, DB4 transform, DB6 transform and DB8 transform and found that the execution speed is marginally increased and also the visual quality of the segmentation output is maintained and even improved in many cases.

4.5.2 Coherence Filter [13]

In order to compensate for digitization artifacts and removal of the noise inculcated in the images, we used a Coherence filter to smooth the image slightly before computing the edge weights. When the image is passed through a coherence filter, the coherence filter performs Anisotropic Diffusion of the color or grayscale image. This process reduces the noise in an image while preserving the region edges. Anisotropic diffusion is a technique that aims at reducing image noise while preserving significant parts of the image details like edges, lines or other parameter that are important for the analysis of the image. As a result, the images obtained after filtering preserves linear structures while at the same time smoothing is made along these structures. A generalization of the usual diffusion equation describes both these cases where the diffusion coefficient is a function of image position and assumes a matrix value. In Anisotropic diffusion each new image in the family is computed by applying the above mentioned generalized equation to the previous image. As a result, anisotropic diffusion is an iterative process where a relatively simple set of computation are used to compute each successive image in the family and this process is continued until a sufficient degree of smoothing is obtained. Due to the above mentioned advantages, we preprocessed the image by means of the coherence filter.

4.6 Working of the Proposed Approach

Flowchart for the proposed approach is shown in Figure 2. This flowchart helps in visualization of the stepwise working of the proposed approach. Flowchart represents the process for color image segmentation.

- As a preprocessing step discrete wavelet transform is done on the images. In our experimentations we used the single-level discrete 2-D wavelet transform (DWT2) which performs single-level 2-D wavelet decomposition with respect to either a particular wavelet or particular wavelet filters specified. We used the wavelets like Haar, DB2, DB4, DB6 and DB8 for experimentations.
- Before passing the image to the coherence filter, the gray scale component image for each color plane, i.e. red, green and blue colors, is extracted by simple operation.
- The grayscale color plane image is then given to the coherence filter where the noise is removed while preserving the edges.
- The graph based segmentation is done on this filtered image. Some input parameters have to be initiated before segmentation is done. These parameters includes
    1. *neighbor_radius*: the neighborhood radius of each pixel [1 by default]
    2. *Coefficient k*: segmentation algorithm coefficient (large prefer large segmented component)
    3. *min_size*: the minimum size allowed for each segment.
- The graph segmentation is then done on the three color planes respectively depending upon the parameters provided.
- As discussed earlier, the boundaries between the two regions are determined based on the definition of predicate.
- Gradient operator help visualize the boundaries between the components. The white color indicates the presence of boundaries. The black color regions are the components separated by the boundaries.
- Morphological operations are done on the gradient image from where the contours are obtained. Finally the contours obtained are more prominent as the insignificant boundaries get eliminated.
- The image is then labeled with random intensity values for each color plane. This image is normalized for display purpose.
- Two neighboring pixels are put in the same component when they appear in the same component in all three of the color plane segmentations.
- The contours obtained from the three color planes are intersected together to form the final contours and the regions are determined based on these contours for color images.
- The regions are finally assigned random colors so that the neighboring regions can be differentiated.

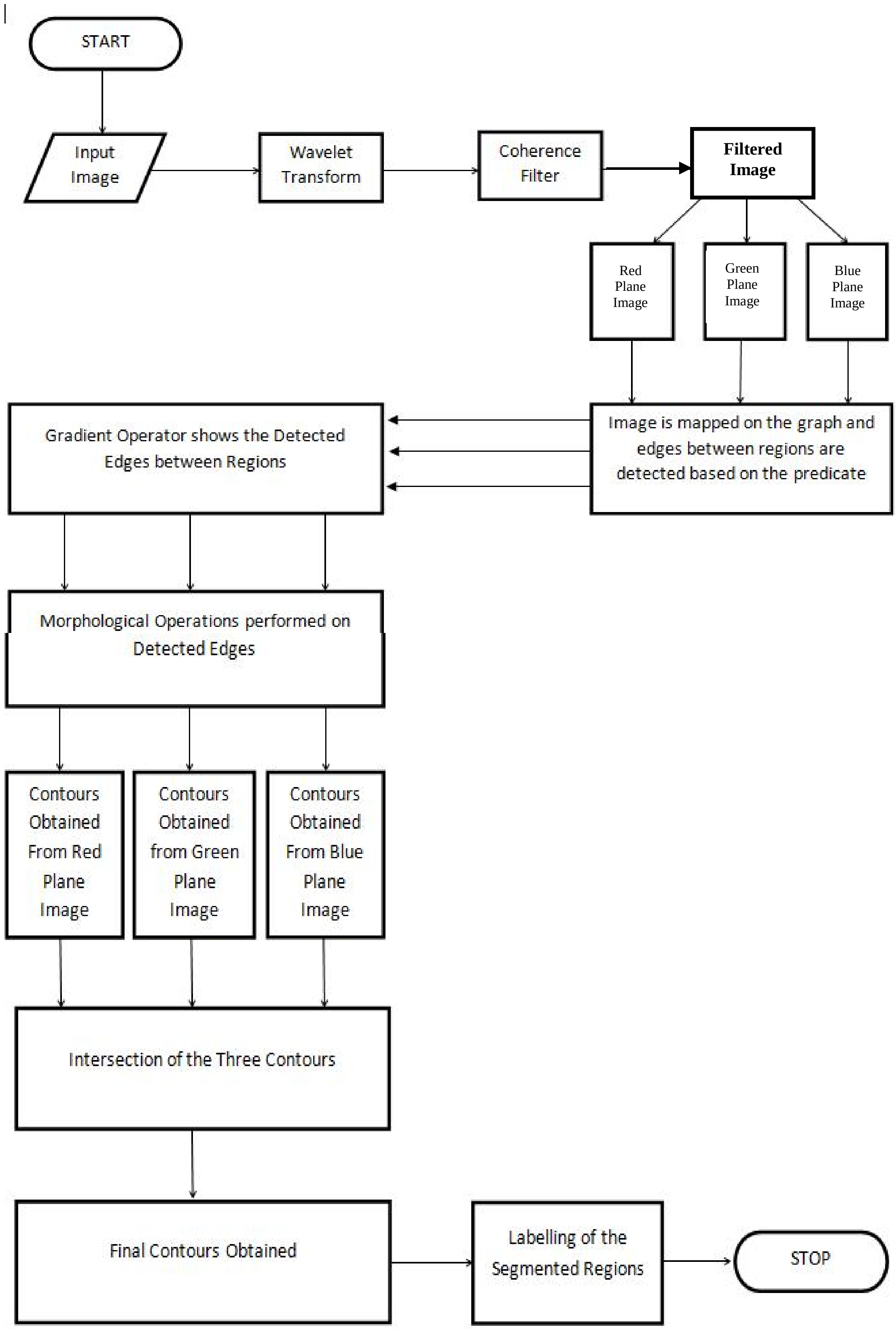


FIGURE 2: Flowchart of the proposed graph based segmentation approach for color images.

5. EXPERIMENTAL RESULTS AND DISCUSSIONS

This section presents the experimental results and discussion on obtained results. Section 5.1 demonstrates the stepwise working of the proposed approach for the different wavelets. The comparison on perceptual basis of various segmented images obtained by our approach is done in Section 5.2. The evaluation and comparison of the obtained results is mentioned in Section 5.3. Thorough discussion about the obtained results is presented in Section 5.4.

The approach is implemented using (MATLAB 8.1.0.604) (R2013a). The experimentations are carried out on Intel (R) Core (TM) 2 Duo T6570, 2.10 GHz processor. The RAM of the system used is 3GB and ROM is 300GB. The operating system is 32-bit and the processor is x64 installed on Windows 8 platform. The experimentations are carried out on natural color and grayscale images taken from Berkeley Image Database [15] as well as synthetic images [16–19] taken from online resources.

5.1 Stepwise Output of the Proposed Approach

This section presents the stepwise results obtained from our proposed approach for each of the wavelet used. The input image “296059.jpg” of size 481 x 321 taken from Berkeley Image Database [15] is shown in Figure 3. The final contours obtained and the labeled images are also displayed for each approach. A tabular representation is provided for the display of intermediate results which are obtained at the mentioned stages. Finally the screenshots of the Graphical User Interface (GUI), which is created in order to visualize the segmentation output in a more effective manner, is presented. In the GUI, the input image, final segmented image, the graph segmented images for each color planes as well as the stepwise results are displayed.

5.1.1 Stepwise results obtained when the image is not preprocessed by any wavelet transform

The final contours obtained after segmentation when the image is not preprocessed by any wavelet transform is as shown in Figure 4 (a). The segmented regions are then labeled as shown in Figure 4 (b). The intermediate results obtained at various stages are provided in Figure 5. Finally screenshot of GUI for the mentioned approach is provided in Figure 6.

FIGURE 3: The input image “296059.jpg” of size 481 x 321 taken from Berkeley Image Database.

FIGURE 4 (a): Final contours obtained after intersecting the contours of the three color planes.

FIGURE 4 (b): Final Labeled Image showing Segmented Regions.

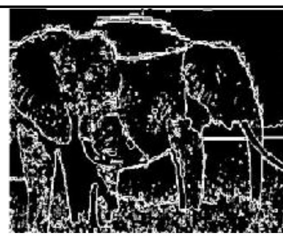
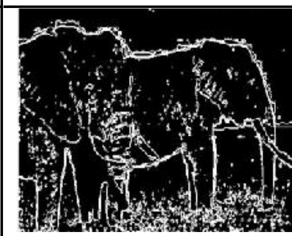
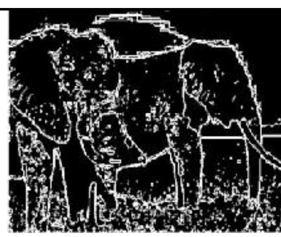
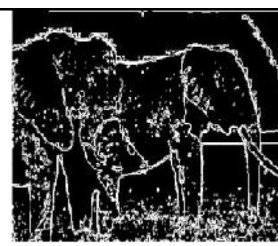
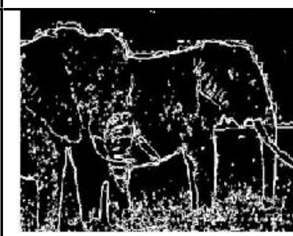
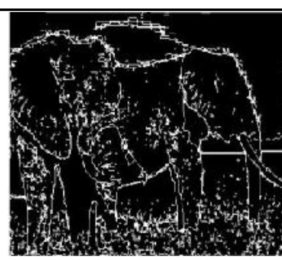
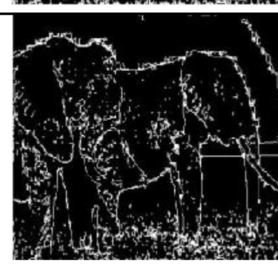
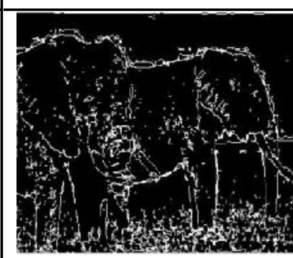
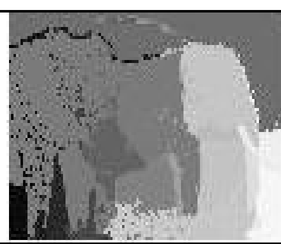
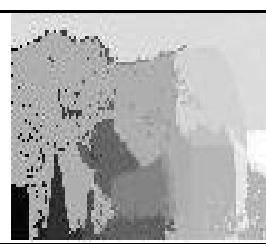
Component	Red Color Plane	Green Color Plane	Blue Color Plane
Grey scale Component of Input Image given to Coherence Filter			
Filtered Image			
Gradient after Graph Based Segmentation			
Contours obtained before morphological operation			
Contours obtained after morphological operation			
Labeled Image			

FIGURE 5: Stepwise output of the proposed segmentation approach when the input image is not preprocessed by any wavelet transforms.

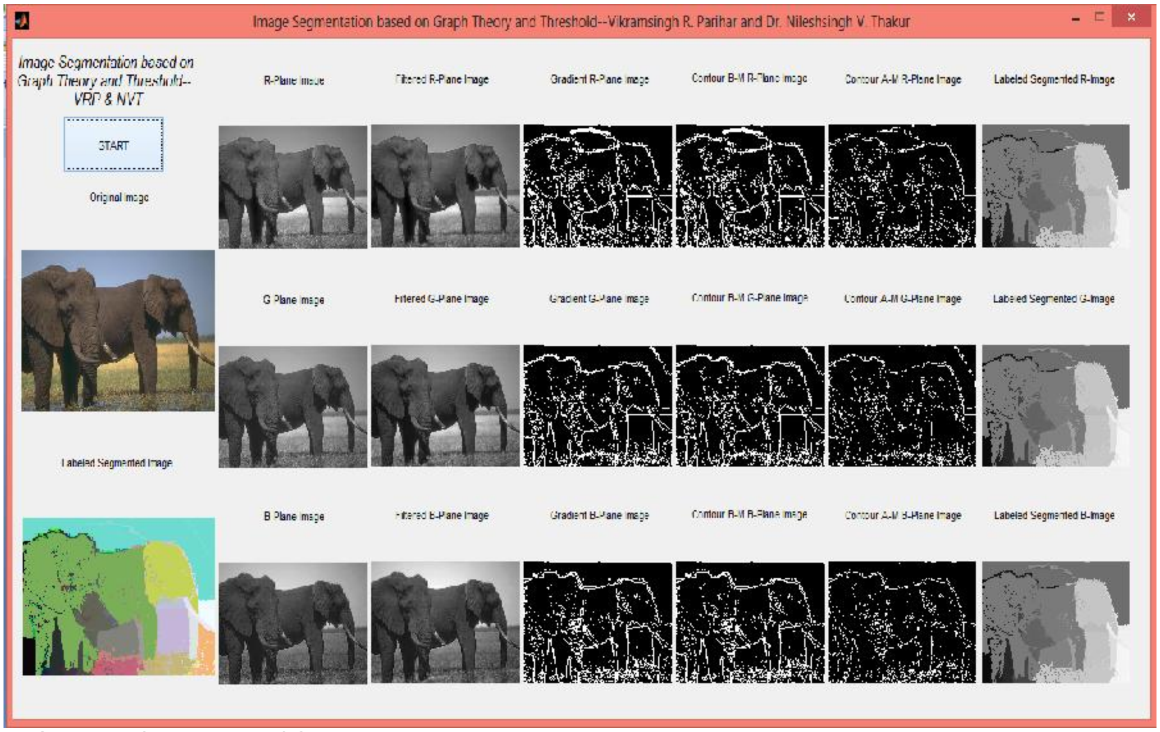


FIGURE 6: Screenshot of GUI showing the obtained results which help in visualizing the segmented output of each color plane along with the labeled image, when the image is not preprocessed by any wavelet transform.

The final contours obtained after segmentation when the image is preprocessed by discrete wavelet transform using different wavelets are as shown in Figure 7 (a) through Figure 11 (a). The segmented regions are then labeled as shown in Figure 7 (b) through Figure 11 (b).

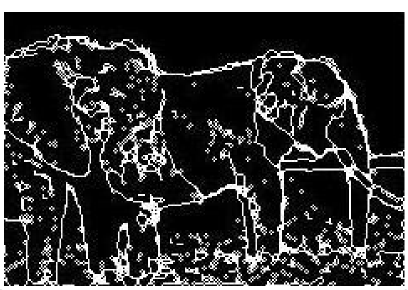


FIGURE 7 (a): Final contours obtained after intersecting the contours of the three color planes.

FIGURE 7 (b): Segmented image obtained after preprocessing the image with Haar Transform.

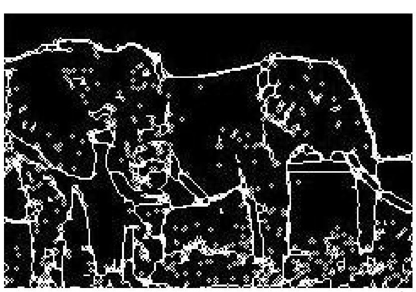


FIGURE 8 (a): Final contours obtained after intersecting the contours of the three color planes.

FIGURE 8 (b): Segmented image obtained after preprocessing the image with DB2 Transform.

FIGURE 9 (a): Final contours obtained after intersecting the contours of the three color planes.

FIGURE 9 (b): Segmented image obtained after preprocessing the image with DB4 Transform.

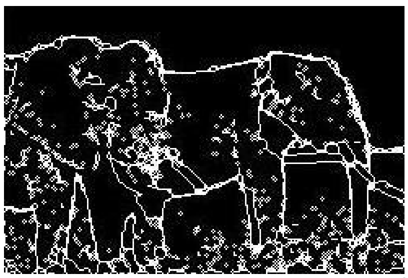


FIGURE 10 (a): Final contours obtained after intersecting the contours of the three color planes.

FIGURE 10 (b): Segmented image obtained after preprocessing the image with DB6 Transform.

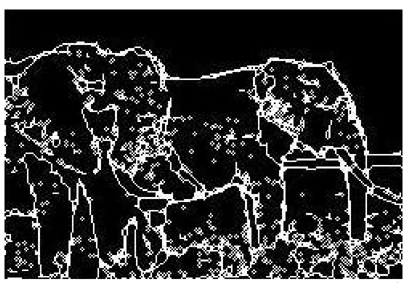


FIGURE 11 (a): Final contours obtained after intersecting the contours of the three color planes.

FIGURE 11 (b): Segmented image obtained after preprocessing the image with DB8 Transform.

5.2 Contours and Labeled Images

Based on the proposed approach, we carried out experimentations on natural as well as synthetic images and then comparative study is done based on the results obtained. The natural images are taken from Berkeley Image Database. These images include color as well as grayscale images both from the 'Test' and 'Train' datasets. We also experimented on some synthetic images taken from the online resources. All these experimentations are carried out using MATLAB (R2013a). Experimental results are shown in Figure 12 through Figure 15.

For all the experiments, we initialized the input parameters as given below:

neighbor_radius = 1 (the neighborhood radius of each pixel [1 by default])

Coefficient k = 350 (segmentation algorithm coefficient [large prefer large segmented component])

min_size = 0.01 (the minimum size allowed for each segment).

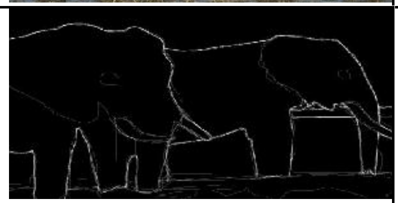
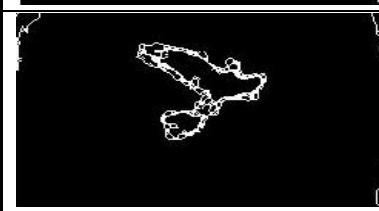
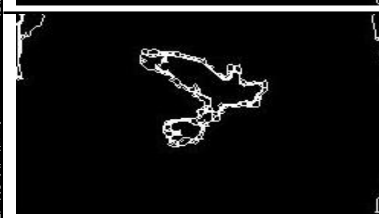
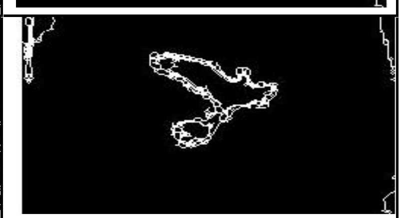
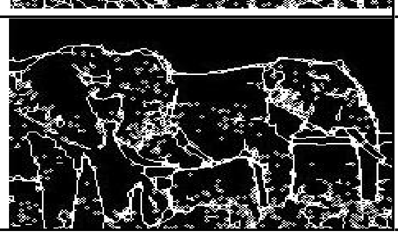
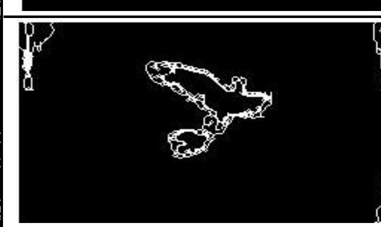
Image name	'296059.jpg'	'135069.jpg'
Image		
Ground Truth Data for Edge Detection		
Contours obtained without preprocessing by any Wavelet Transform		
Contours obtained after DWT2 using Haar Wavelet		
Contours obtained after DWT2 using DB2 Wavelet		
Contours obtained after DWT2 using DB4 Wavelet		
Contours obtained after DWT2 using DB6 Wavelet		
Contours obtained after DWT2 using DB8 Wavelet		

FIGURE 12: Demonstration of Contour Images obtained for given images.

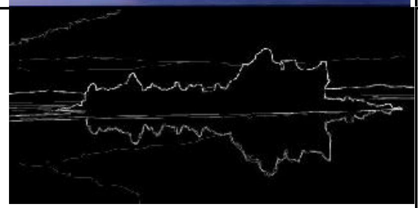
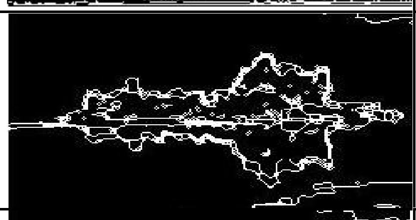
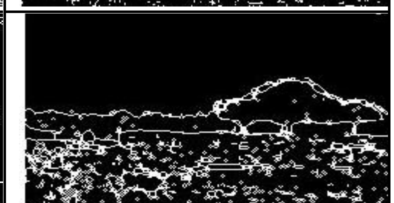
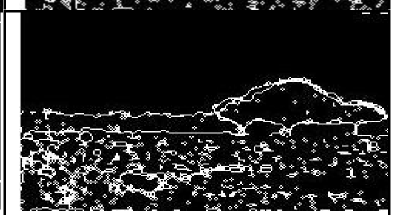
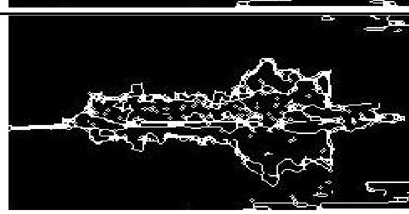
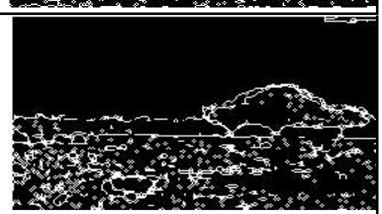
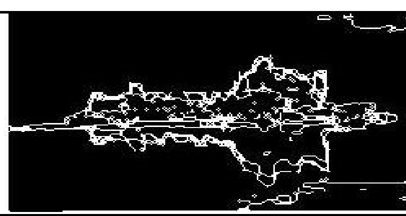
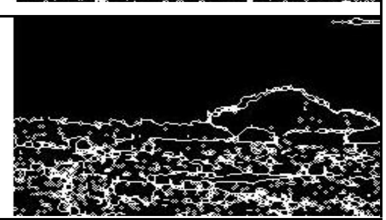
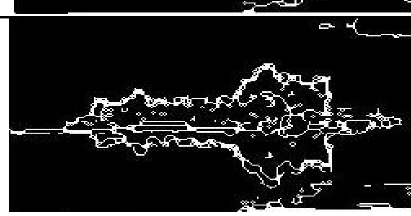
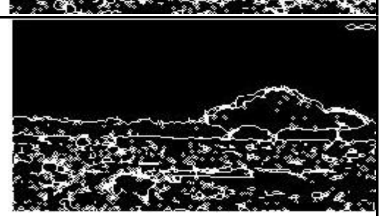
Image name	'143090.jpg'	'296007.jpg'
Image		
Ground Truth Data for Edge Detection		
Contours obtained without preprocessing by any Wavelet Transform		
Contours obtained after DWT2 using Haar Wavelet		
Contours obtained after DWT2 using DB2 Wavelet		
Contours obtained after DWT2 using DB4 Wavelet		
Contours obtained after DWT2 using DB6 Wavelet		
Contours obtained after DWT2 using DB8 Wavelet		

FIGURE 13: Demonstration of Contour Images obtained for given images.

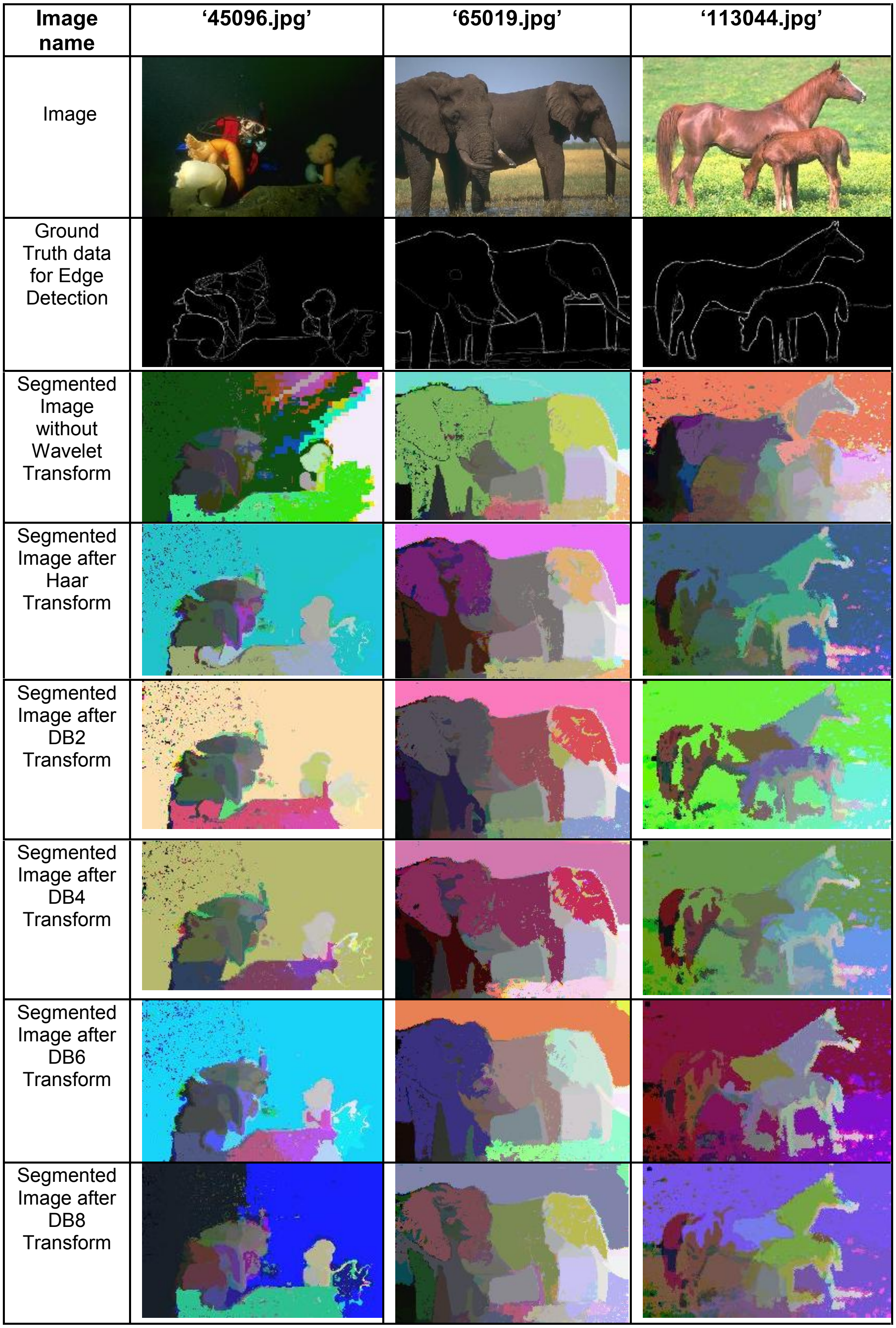


FIGURE 14: Labeled images obtained after segmentation.

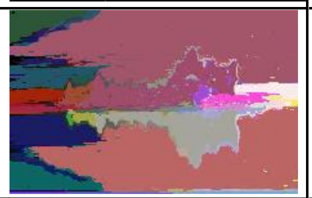
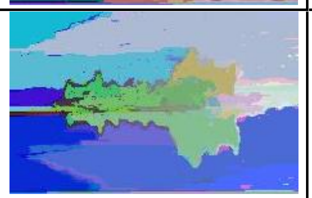
Image name	**'135069.jpg'**	**'143090.jpg'**	**'296007.jpg'**
Image			
Ground Truth Data for Edge Detaction			
Segmented Image without Wavelet Transform			
Segmented Image after Haar Transform			
Segmented Image after DB2 Transform			
Segmented Image after DB4 Transform			
Segmented Image after DB6 Transform			
Segmented Image after DB8 Transform			

FIGURE 15: Labeled images obtained after segmentation.

5.3 Evaluation and Comparison of Obtained Experimental Results

In this section, the evaluations of the obtained results are done. The comparison of the results is also carried out. However, the comparison with the existing approach of [11] or [12] is not attained. This is due to the fact that the authors in these papers did not evaluate their results. They used the term “Human Perception” as to evaluate their result which is, in fact, a vague term. However, we find that our segmentation results also produce regions that are meaningful.

We, here, used the following parameters used for evaluations of our results.

1. Time Required for Graph Based Segmentation
2. Peak Signal to Noise ratio (PSNR)
3. Performance Ratio (PR)
4. Precision and Recall.

1. Time Required for Graph Based Segmentation

The time required for graph based segmentation approach to execute after preprocessing the image by DWT2 using the wavelets like Haar, DB2, DB4, DB6 and DB8 is shown in the Table 1. Table 1 helps for comparative study of the results.

2. Peak Signal to Noise Ratio (PSNR)

The PSNR of the final contour is calculated with reference to the ground truth dataset for edge detection. The PSNR is calculated using the following formula:

$$\text{PSNR (dB)} = 10 * \log\left(\frac{255^2}{MSE}\right), \quad \text{where} \qquad MSE = \sum_{i=1}^{x} \sum_{j=1}^{y} \frac{x*y}{(|A_{ij}-B_{ij}|)^2}$$

3. Performance Ratio (PR) [14]

PR is the ratio of true edges to false edges. Here true edges mean the edge pixels identified as edges in the ground truth data and false edges means the non edge pixels identified as edges and the edge pixels identified as non edges. The PR is calculated from the given formula

$$\text{P R} = \frac{\text{True Edges (Edge pixels identified as Edges)}}{\text{False Edges (Non edge pixels identified as edges)} + \text{(Edge pixels identified as Non-Edge pixels)}} \times 100$$

4. Precision and Recall

Precision is the fraction of the edges that are obtained by our approach that are relevant with the edges obtained from ground truth data. Whereas recall is the fraction of all relevant instances that are retrieved. There are four cases which have to be first evaluated:

- TN / True Negative: case is negative and predicted negative
- TP / True Positive: case is positive and predicted positive
- FN / False Negative: case is positive but predicted negative
- FP / False Positive: case is negative but predicted positive

Now, $\text{Precision} = \frac{TP}{TP + FP}$ and $\text{Recall} = \frac{TP}{TP + FN}$

Experimental results are shown in Table 2 through Table 5 and Figure 16 through Figure 19.

Input Image	Haar	DB2	DB4	DB6	DB8
‘3096.jpg’	0.46	0.46	0.47	0.48	0.48
‘42049.jpg’	0.43	0.46	0.45	0.46	0.46
‘62096.jpg’	0.45	0.46	0.46	0.47	0.48
‘108082.jpg’	0.45	0.46	0.48	0.49	0.49
‘167062.jpg’	0.45	0.46	0.45	0.47	0.47

TABLE 1 (a): Time required for segmentation of natural gray images in seconds

Input Image	Haar	DB2	DB4	DB6	DB8
‘45096.jpg’	0.46	0.46	0.48	0.48	0.49
‘65019.jpg’	0.47	0.46	0.47	0.48	0.49
‘113044.jpg’	0.47	0.47	0.48	0.48	0.50
‘135069.jpg’	0.40	0.41	0.42	0.42	0.44
‘143090.jpg’	0.42	0.43	0.44	0.45	0.45
‘296007.jpg’	0.43	0.44	0.45	0.46	0.48
‘296059.jpg’	0.46	0.46	0.46	0.48	0.49
‘306005.jpg’	0.46	0.47	0.47	0.48	0.50
‘beach.jpg’	0.24	0.25	0.25	0.24	0.25
‘rice.jpg’	0.51	0.65	0.50	0.53	0.52

TABLE 1 (b): Time required for segmentation of natural color images in seconds

Image	Haar	DB2	DB4	DB6	DB8
‘im1.jpg’	0.21	0.21	0.22	0.22	0.22
‘Syntheticim1.jpg’	0.12	0.13	0.13	0.13	0.14
‘Syntheticim2.jpg’	0.21	0.26	0.21	0.22	0.24
‘Syntheticim3.jpg’	0.23	0.22	0.23	0.38	0.23

TABLE 1 (c): Time required for segmentation of synthetic images in seconds

Input Image	Haar	DB2	DB4	DB6	DB8
‘45096.jpg’	46.7611	50.0392	49.1753	51.2983	43.1365
‘65019.jpg’	70.7948	71.9072	70.4551	70.5546	67.2279
‘113044.jpg’	34.9189	35.0555	34.3365	35.1997	33.7685
‘135069.jpg’	41.6328	42.4072	38.3769	42.1317	33.0302
‘143090.jpg’	44.1998	45.2006	43.4386	42.8078	41.0001
‘296007.jpg’	58.5074	59.4026	57.2390	59.1842	58.3477
‘296059.jpg’	59.3160	61.5131	59.6177	60.2192	59.5920
‘306005.jpg’	60.0831	61.6381	58.8643	56.7069	57.6688

TABLE 2: Performance Ratio of the Segmented Results.

Input Image	Haar	DB2	DB4	DB6	DB8
‘45096.jpg’	12.2143	11.4409	12.5631	12.7120	12.2341
‘65019.jpg’	9.6443	9.5434	9.1652	9.0504	9.101
‘113044.jpg’	8.6669	8.8337	8.4517	8.5042	8.5383
‘135069.jpg’	17.721	17.6791	16.8576	16.4634	16.7061
‘143090.jpg’	12.7865	12.2663	12.0917	12.6261	12.3636
‘296007.jpg’	12.0733	12.3503	12.2443	10.9233	11.7055
‘296059.jpg’	10.6418	11.089	10.7377	10.9542	10.3224
‘306005.jpg’	10.0494	10.3964	10.0695	9.6637	9.6162

TABLE 3: PSNR of the Segmented Results in decibels.

Input Image	HAAR	DB2	DB4	DB6	DB8
‘45096.jpg’	0.0022	0.0020	0.0023	0.0035	0.0019
‘65019.jpg’	0.0024	0.0024	0.0017	0.0018	0.0016
‘113044.jpg’	0.0009	0.0007	0.0010	0.0011	0.0007
‘135069.jpg’	0.0026	0.0025	0.0012	0.0025	0.0028
‘143090.jpg’	0.0035	0.0031	0.0030	0.0019	0.0026
‘296007.jpg’	0.0038	0.0041	0.0047	0.0021	0.0025
‘296059.jpg’	0.0026	0.0027	0.0030	0.0024	0.0042
‘306005.jpg’	0.0014	0.0015	0.0019	0.0013	0.0015

TABLE 4: Precision of the Segmented Results.

Input Image	HAAR	DB2	DB4	DB6	DB8
‘45096.jpg’	0.0499	0.0420	0.0472	0.0682	0.0446
‘65019.jpg’	0.0428	0.0428	0.0308	0.0317	0.0300
‘113044.jpg’	0.0248	0.0189	0.0272	0.0295	0.0189
‘135069.jpg’	0.0610	0.0563	0.0282	0.0563	0.0751
‘143090.jpg’	0.0449	0.0402	0.0393	0.0234	0.0327
‘296007.jpg’	0.0337	0.0360	0.0422	0.0186	0.0219
‘296059.jpg’	0.0404	0.0404	0.0461	0.0362	0.0643
‘306005.jpg’	0.0274	0.0283	0.0365	0.0256	0.0292

TABLE 5: Recall of the Segmented Results.

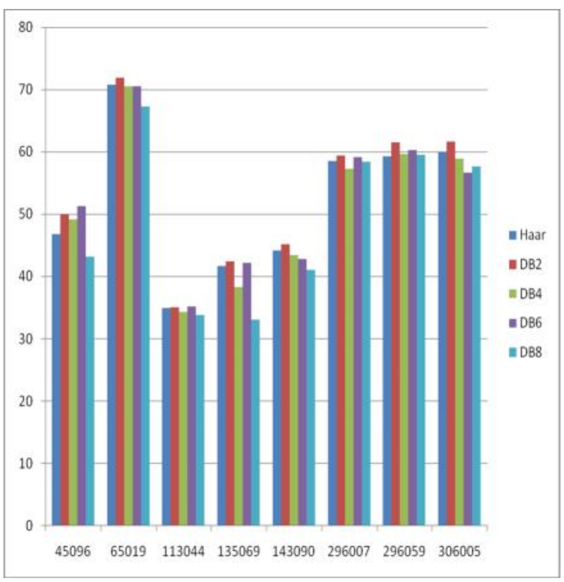


FIGURE 16: Bar Graphs showing comparison between the Performance Ratio of the results obtained for different wavelets.

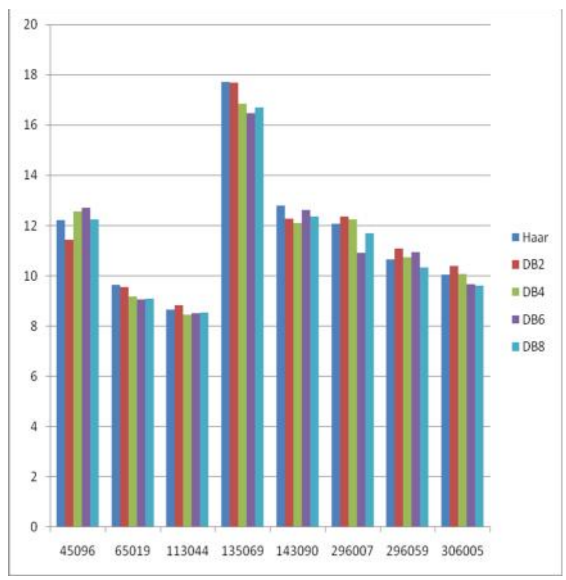


FIGURE 17: Bar Graphs showing comparison between the PSNR of the results obtained for the different wavelets.

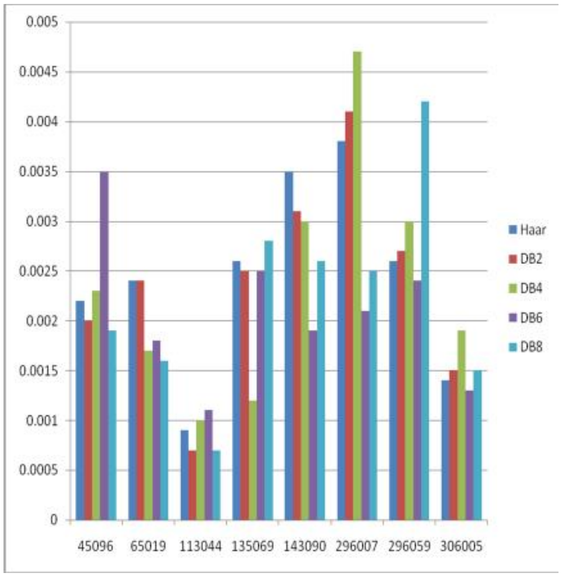


FIGURE 18: Bar Graphs showing comparison between the Precision of the results obtained for the different wavelets.

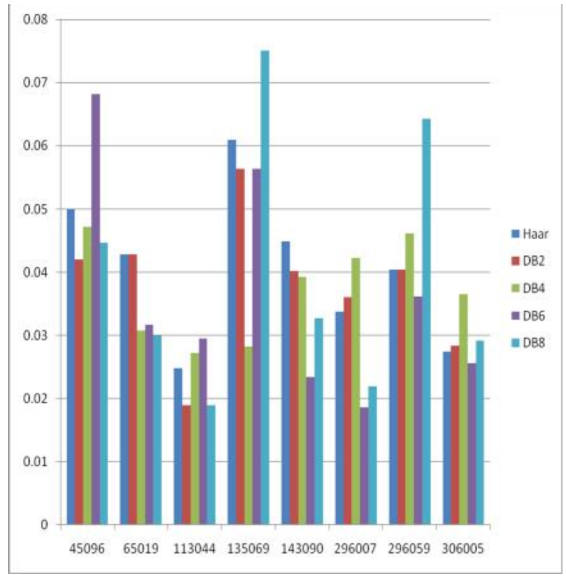


FIGURE 19: Bar Graphs showing comparison between the Recall of the results obtained for the different wavelets.

5.4 Discussion

The stepwise results obtained from each of the wavelet used for the image '296059.jpg' helps us to understand the proper working of the segmentation process. It also provides us the visualization of the transformation of image at almost each and every level. The contours mentioned are nothing but the boundaries detected by the segmentation process. Through the stepwise observation, it is found that the contours obtained are inherently related with the true edges of the image when compared with ground truth data for edge detection. This is justified by analytical as well as perceptual evaluation.

Later, the segmentation output for various images are displayed which involves both the contours obtained for the various wavelets along with the labeled images. From observations, it can be identified that all the wavelets provided segmentation results which are perceptually important. It is, however, very difficult to evaluate on the basis of perception the quality of obtained images. This motivated us to use the performance evaluation parameters like Time, PR, PSNR, Precision and Recall.

The Table 1 (a) through Table 1 (c) presented the execution time after the image is preprocessed by the respective wavelets. From Table 1 it is very tough to suggest which of the approaches is more preferable as the time differences is only of a few milliseconds. However, it is found that time taken for graph segmentation after DWT2 using Haar wavelets required less time for almost all the mentioned images.

The PR, PSNR, Precision and Recall parameters are generally used for edge detection. As our approach is also incorporating the determination of edges between the regions, we applied these parameters to the detected boundaries. The comparative study shows that all the wavelet images performed in nearly an equal conduct as long as PR is concerned. The PR of DB2 is, however, found to be better when the comparison is done based on the bar graphs and is closely followed by DB6.

The PSNR is also an important parameter which can evaluate the experimental results. The comparison of the obtained results for each of the wavelet used is done with the ground truth image and the edges of both the images are compared. The higher the value of PSNR indicates the minimum is the value of MSE and ultimately the better is the obtained results. From the study of the Table 3 and the bar graph shown in Figure 17, we found that the PSNR values of Haar wavelet is more in few cases closely followed by DB2 wavelets.

The Precision and Recall have also emerged as one of the wisely used evaluation parameters. The precision and recall are indirectly proportional to each other; wherein as the precision increases the recall decreases and vice versa. Higher precision is preferable and so does lower recall. The precision and recall parameters provide information about the relevancy between the obtained results with respect to some standard quantity.

We compared the edges obtained by our approach to the ground truth data set of edge detection. The precision and recall of each wavelet are shown in Table 4 and Table 5, plotted as bar graphs shown in Figure 18 and Figure 19 for comparison. By observing the results we found that precision of DB2 and Haar wavelets are comparatively higher. The recall of DB6 is minimum, closely followed by DB4 and DB2.

6 . CONCLUSION AND FUTURE SCOPE

This section presents the conclusions drawn from the evaluation and comparison of experimental results. The section concludes with future scope.

6.1 Conclusion

Based on the experimental results and discussion, the following conclusions are drawn:

- The contours obtained from graph segmentation are relevant to the true edges of the image. The observations regarding the edges are done in Figure 12 and Figure 13.

- The algorithm captures perceptually important regions. This can be justified from the Figure 14 and Figure 15 where the segmented results are compared with the input image as well as ground truth data.

- From the Table 1, it is found that time taken for graph segmentation after DWT2 using Haar wavelets required less time for almost all the mentioned images.

- The comparative study from the Table 2 shows that all the wavelet images performed in nearly an equal conduct as long as PR is concerned. The PR of DB2 is, however, found to be better when the comparison is done based on the bar graphs and is closely followed by DB6.

- The precision and recall of each wavelet are calculated and presented in the Table 4 and Table 5 and plotted as bar graphs in Figure 18 and Figure 19 for comparison. By observing the results, we found that precision of DB2 and Haar wavelets are comparatively higher. The recall of DB6 is minimum, closely followed by DB4 and DB2.

6.2 Future Scope

This section provides the possible future directions to extend the presented work.

- We, here, considered the wavelets of Haar, DB2, DB4, DB6 and DB8. In the future work, one can carry out experimentations considering other families of wavelets for preprocessing the image before segmentation and observe the results.

- Also, after the contours are extracted, the regions are labeled with random colors. This however may provide an improper segmentation results sometime as the neighboring regions are assigned colors that may not be distinguished. So, instead of random colors, a function can be developed that can assign largely varying colors to the neighboring regions.

- Developed approach can be used in other image processing work, in particular, image compression and image recognition.

- Presented paper deals with the segmentation of the still images, but, can be extended for the analysis of video. One can use the proposed approach as the basis for video compression.

- Self similarity check can be explored to have the better segmentation in combination with the proposed approach.

- Advanced non-classical optimization techniques, like, neural network and genetic algorithm can be used to optimize the obtained results. From this point of view, one can model the proposed approach in terms of the problem of neural network and genetic algorithm.

6. REFERENCES

[1] B. Peng, L. Zhang and D. Zhang, "A survey of graph theoretical approaches to image segmentation", *Pattern Recognition*, Vol. 46, pp.1020-1038, (2013).

[2] B. Peng, L. Zhang, D. Zhang and J. Yang, "Image segmentation by iterated region merging with localized graph cuts", Pattern Recognition Vol. 44, pp. 2527-2538, (2011).

[3] W. Tao, F. Chang, L. Liu, H. Jin and T. Wang, " Interactively multiphase image segmentation based on variation formulation and graph cuts", Pattern Recognition Vol. 43, pp. 3208-3218, (2010).

[4] J. Kim and K.Sang, "Color–texture segmentation using unsupervised graph cuts", Pattern Recognition Vol. 42, pp. 735-750, (2009).

[5] M. Bleyer and M. Gelautz, "Graph-cut-based stereo matching using image segmentation with symmetrical treatment of occlusions", Signal Processing: Image Communication, Vol.22, pp.127-143, (2007).

[6] J. Shi and J. Malik, "Normalized Cuts and Image Segmentation", IEEE Transactions on pattern analysis and machine intelligence, Vol. 22, Issue No. 8, (2000).

[7] S. Wang and J. M. Siskind, "Image Segmentation with Ratio Cut", IEEE Transactions on pattern and machine intelligence, Vol. 25, Issue No. 6, (2003).

[8] W. Tao, H. Jin, and Y. Zhang, "Color Image Segmentation Based on Mean Shift and Normalized Cuts", IEEE Transactions on systems, man and cybernetics, Vol. 37, Issue No. 5, (2007).

[9] R. C. Wilson, E. R. Hancock, and B. Luo, " Pattern Vectors from Algebraic Graph Theory", IEEE Transactions on pattern analysis and machine intelligence, Vol. 27, Issue No 7, (2005).

[10] Y. Yang, S. Han, T. Wang, W. Tao and X. Tai, "Multilayer graphcuts based unsupervised color–texture image segmentation using multivariate mixed student's t-distribution and regional credibility merging", Pattern Recognition, Vol. 46, pp. 1101-1124, (2013).

[11] P.F.Felzenszwalb and D.P.Huttenlocher, "Efficient graph based image segmentation", International Journal of Computer vision, Vol.59, Issue No.2, (2004).

[12] M. Zhang and R. Alhajj, “Improving the Graph-Based Image Segmentation Method”, Proceedings of the 18th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'O6), 0-7695-2728-0106, (2006).

[13] J. Weickert, “Coherence-Enhancing Diffusion Filtering”, International Journal of Computer Vision, Vol. 31, pp. 111-127, (1999).

[14] P. A. Khaire and N. V. Thakur, “ A Fuzzy Set Approach for Edge Detection”, International Journal of Image Processing (IJIP), Vol. 6, Issue No. 6, (2012).

[15] http://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/ (accessed on 24/04/2014)

[16] http://fin6.com/2013/12/color/ (accessed on 24/04/2014)

[17] http://clancyartpages.files.wordpress.com/2011/09/color_grid_dl.jpg (accessed on 24/04/2014)

[18] http://www.clker.com/clipart-grayscale-flower-decoration.html (accessed on 24/04/2014)

[19] http://furbeeconsulting.com/blog/ (accessed on 24/04/2014)