Image Segmentation Based on Watershed and Edge Detection Techniques

Nassir Salman
Computer Science Department, Zarqa Private University, Jordan

Abstract: *A combination of K-means, watershed segmentation method, and Difference In Strength (DIS) map was used to perform image segmentation and edge detection tasks. We obtained an initial segmentation based on K-means clustering technique. Starting from this, we used two techniques; the first is watershed technique with new merging procedures based on mean intensity value to segment the image regions and to detect their boundaries. The second is edge strength technique to obtain an accurate edge maps of our images without using watershed method. In this paper: We solved the problem of undesirable oversegmentation results produced by the watershed algorithm, when used directly with raw data images. Also, the edge maps we obtained have no broken lines on entire image and the final edge detection result is one closed boundary per actual region in the image.*

Keywords: *Watershed, difference in strength map, K-means, edge detection, image segmentation*.

Received September 30, 2004; accepted February 13, 2005

1. Introduction

Any gray tone image can be considered as a topographic surface. If we flood this surface from its minima and, if we prevent the merging of the waters coming from different sources, we partition the image into two different sets: The catchment basins and the watershed lines. If we apply the watershed transformation to the image gradient, the catchment basins should theoretically correspond to the homogeneous gray level regions of this image. However, in practice, this transform produces an important over-segmentation due to noise or local irregularities in the gradient image.

To perform image segmentation and edge detection tasks, there are many methods that incorporate region-growing and edge detection techniques. For example, it is applying edge detection techniques to obtain Difference In Strength (DIS) map. Then employ region growing techniques to work on the map as in [2, 10]. In [3], combining both special and intensity information in image segmentation approach based on multi-resolution edge detection, region selection and intensity threshold methods to detect white matter structure in brain. In [3, 6, 7, 9] adaptive clustering algorithm and K-means clustering algorithm are generalizing to include spatial constraints and to account for local intensity variations in the image. The spatial constraints are included by the use of a Gibbs Random Field model (GRF). The local intensity variations are accounted for in an iterative procedure. In [8], Vincent introduced a fast and flexible algorithm for computing watersheds in digital gray scale images.

In this paper a combination of *K-means clustering and Watershed* techniques then region merging and edge detection procedures were used. The clustering method was applied to obtain an image of different intensity regions based on minimum distance to examine each pixel in the image and then to assign it to one of the image clusters. Then a *watershed transformation technique* worked on the gradient of that image was employed to reduce the oversegmentation of the *watershed algorithm*. But the result is oversegmentation image if we use the *watershed algorithm* with the gradient of raw data image without clustering method above. To get rid oversegmentation, merging method based on *mean gray values and edge strengths* (T_1, T_2) were used. The *watershed algorithm* can segment image into several homogeneous regions which have the same or similar gray levels. To perform meaningful segmentation of image, regions of different gray levels should be merged if the regions are from the same object. The *watershed segmentation* generates spatially homogeneous regions which are oversegmented. But the objective of this work is to obtain one closed boundary per actual region in the final segmentation results of the image under study.

2. K-Means Clustering Method

1. The initial intensity mean value of each region in the image was defined according to the image histogram because the locations of peaks and valleys of a histogram indicate the clusters of similar-spectral pixels in an image.

2. The goal of this method is to find a partition S_j of the data points that minimizes the sum of squared distance to the center of the cluster. At first, points were assigned at random into K sets Sj. Then each point was assigned to the set whose mean center is the closest.
   This was repeated until no point changes of set [6, 7].
3. At the *kth* iterative step distribute the samples {x} among the *K* cluster domains, using the relation

$$x \in S_j(k) \quad \text{if} \quad \|x - u_j(k)\| < \|x - u_j(k)\| \qquad (1)$$

   for all i =1, 2, …, K, $i \neq j$, where $S_j(k)$ denotes the set of samples whose cluster center is $u_i(k)$.
4. From the results of step 2, the new cluster centre (new average value) $u_j(k + 1)$, j = 1, 2, .., *K*, was computed so that the sum of the square distances from all pixels in $S_j(k)$ to the new cluster center was minimized. In other words, the new cluster center $u_j(k + 1)$ was computed so that the performance index J_j:

$$J_j = \sum_{x \in S_j(k)} \left\| x - u_j(k+1) \right\|^2, \; j = 1,2,...,K, \qquad (2)$$

   is minimized. The $u_j(k + 1)$ which minimizes this performance index is simply the sample mean of $S_j(k)$. Therefore, the new cluster center is given by

$$u_j(k+1) = \frac{1}{N_j} \sum_{X \in s_J(k)} x, \qquad j = 1,2,..K, \qquad (3)$$

   Where N_j is the number of samples or pixels in $S_j(k)$. This was repeated until no point changes of set. And now:
5. *If* $u_j(k + 1) = u_j(k)$ *for* $j = 1, 2,\ldots, K$, the algorithm has converged and the procedure is terminated. Otherwise go to step 2. In this method the average value of each group was initialized from image histogram then the labels of the pixels that belong to which group is initialized using gray levels difference between every pixel and the mean value of each group, then compared the results with minimum distance (denoted equal 255 gray levels).

The mean value of the group that have been calculated and the labeled values were updated. The output image has different intensity regions. Then the gradient values of this image were calculated using gradient operator as defined in equation (4).

3. Watershed Segmentation Method

This method includes three main steps:

1. Calculated gradient image values.
2. Using watershed algorithm step.
3. Merging steps; as follows.

3.1. Gradient Calculations

In contrast to a classical area based segmentation, the *watershed transform* [8] was executed on the gradient image. The gradient defined the *first partial derivative* of an image and contains a measurement for the change of gray levels. The gradient values (G (x, y)) of the initial segmented image were obtained using firstly the approximation of the *gradient operator* [1] in *x, y* directions (equation (4)) as two 3x3 masks.

$$U_x(i,j) = (2 + 4c)^{-1}\{u(i + 1, j) - u(i - 1, j) + c\,[u(i + 1, j + 1) - u(i - 1, j + 1) + u(i + 1, j - 1) - u(i - 1, j - 1)]\}$$

$$U_y(i,j) = (2 + 4c)^{-1}\{u(i, j + 1) - u(i, j - 1) + c\,[u(i + 1, j + 1) - u(i + 1, j - 1) + u(i - 1, j + 1) - u(i - 1, j - 1)]\}$$

$$G(x,y) = \sqrt{(\partial f / \partial x)^2 + (\partial f / \partial y)^2} \qquad (4)$$

Where $c = (\sqrt{2} - 1)/(2 - \sqrt{2})$. Then the gradient image values (G (x, y)) were calculated. The gradient values on the border of input image are the same as in the it's inner pixels. This gradient image values are useful to calculate edge strength values as in equation (6). See edge strength merging step in section 3.3.

3.2. Watershed Algorithm and Processing Procedures

Because the regions in the image characterized by small variations in gray levels have small gradient values, Thus in practice, we often see watershed segmentation applied to gradient of an image, rather than to the image itself. The aim of the watershed transform is to search for regions of high intensity gradients (*watersheds*) that divide neighbored local minima (basins). For applying the *watershed transform* an advanced, fast, and an accurate algorithm proposed by Vincent [8] was used. From this algorithm: A marker image included zero marker values of watershed line pixels was obtained. Then in our work:

1. We used 3x3 mask to scan this marker image to find these zero marker values, turn them to their intensity values as in original image and compared these values with their neighbor pixels intensity to assign them to one marker region. We did that because discontinuity in the watershed pixels happened by using this algorithm. We deleted all *watershed pixels* (zero marker values) to obtain second marker image represents the markers of image regions only.
2. At this point we calculated mean gray values of every region as in equation (5). The block diagram of our proposed method as shown in Figure 1.

3. To find adjacency relations of the regions in our image, to get rid of oversegmentation results and to do merging process (see next section), RAG was created same as seen in Figure 2-a and its data structure was shown in Figure 2-c.
4. Then to get edge pixels list, RAB was created and its data structure was shown in Figure 2-b, where edge strengths come from this data structure and there is one and only one edge between every two adjacent regions. In this procedure, for every two regions *i*, *j*, we calculated:

   a. Edge pixel pointer.
   b. Edge length (N).
   c. Edge strengths.
   d. Next edge pointer for another two regions *k*, *l* [4].

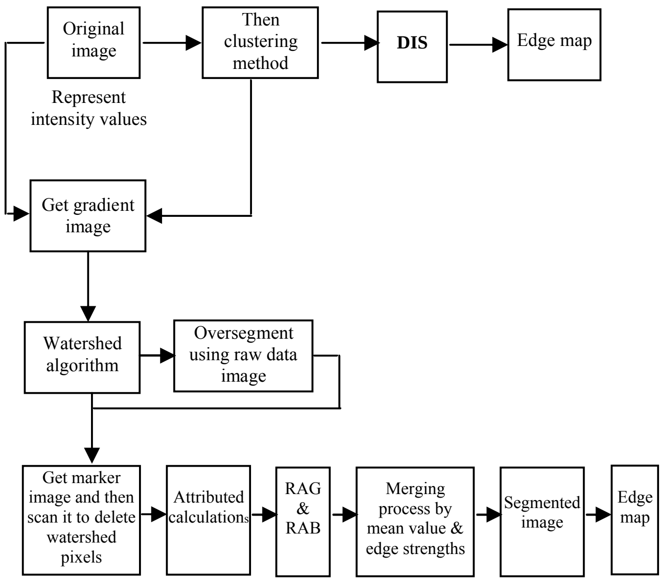
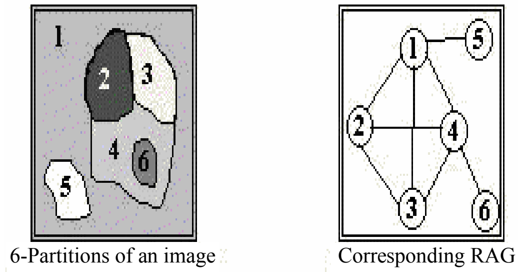


Figure1. Block diagram of our proposed method.

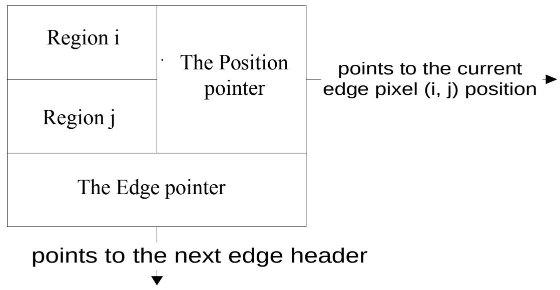
In RAG data structure Figure 2-c, R_1 is a head, and R_2, R_3, R_5, R_4 are chain connected with R_1. In creating edge pixels RAB Figure 2-b, a pointer from every two adjacent regions was used to assign the position of the current pixel (i, j) as an edge pixel and so on for other edge points positions (x_1, y_1 to x_N, y_N) which represent the positions of the edge pixels (e. g., between regions R_i & R_j). Also, another pointer was assigned for another edge pixels (if it exists) between other two regions, having relation between the two pointers.

The *edge strength* between two different regions R_i, R_j comes from this data structure. After completing image scanning we got on RAG and edge pixels RAB for all regions in the entire image. The procedures above were done by scanning the second marker image row by row manner where every pixel in the second marker image was scanned into two directions (x, y) as shown in Figure 2-d and as follows:

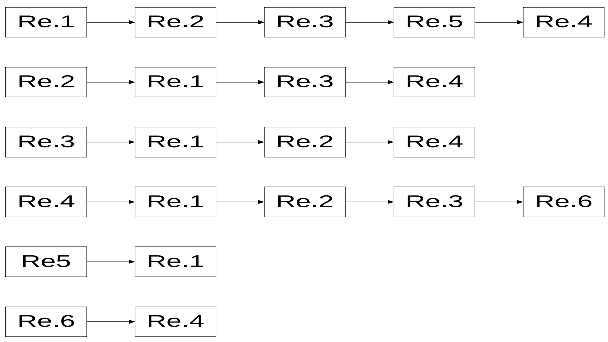
1. For i = 0 to row numbers, for j = 0 to column numbers, if k != m goto 2, else goto 1 // k represents the black pixel in the figure.
2. Create RAG (i. e., R_1 is a head, R_2, R_3, R_5, R_4 are chain connected with R_1) & edge points list (current pixel i is an edge pixel). In the final of this procedure, RAG and edge pixels for our image were obtained. Then merging procedure (next section) based on mean value of each region and edge strengths (T_1, T_2).



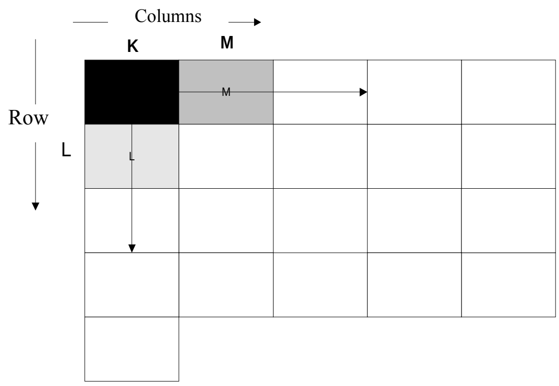
(a) Segmented regions and adjacency graph.



(b) RAB data structure.



(c) RAG data structure correspond to (a).



(d) Scanning every pixel in the 2nd marker in (x, y) directions.

Figure 2. Shows RAG, RAB and their data structures.

3.3. Merging Procedures

The marker image was used to calculate the number of pixels of each region in that image (N_i) and then the mean intensity value (μ) of each region (i), was calculated as follow:

$$(\mu_i) = \frac{\sum_{N \in i} origion\ \ pixels\ \ intensity\ \ of\ \ (i)}{N_i} \tag{5}$$

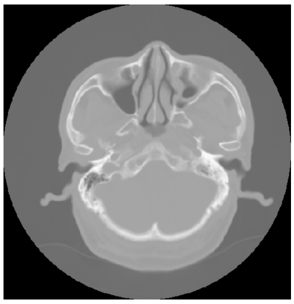
The original intensity values of these marker pixels of region *i* was obtained from the original input image because they have the same positions in the two images (e. g., input image and marker image). After we calculated the average mean value, RAG & RAB were created and calculated. Then merging procedure was started based on one of the two following criterions:

1. *Merge the pair of regions (R_i, R_j)* if the difference of means value of the two regions is less than our tested threshold (T), where the T value was chosed experimentally to obtain good image segmentation. This step of merging is very useful to get rid over-segmentation results produced by *watershed algorithm* especially when we used the gradient of raw data image directly with this algorithm instead of using the output of clustering method. The output segmented image has no oversegmentation result and the final edge detection results are one closed boundary per actual region in the image as shown in Figure 3-(c, d).
2. *Edge Strength Merging Process*: Two *edge strengths* gradient values (T_1, T_2) were used in one subroutine, T_1 is less than T_2. For example if we choose $T_1 = 1$ and if the Edge strength as in equation (6) is less than 1, we get merging of every two adjacent regions because the watershed algorithm [8] we used based on immersion procedure and in this procedure it looks to the topographic surface. It means we related intensity values as an altitude (height) and we got merging results by comparing the gradient values of the edge points (pixels) between the two regions and the region itself. If the points have low gradient values, that means the merging was done and the region becomes large.

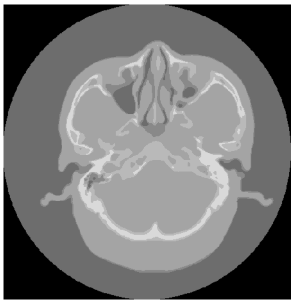
So; in this procedure it is very important and useful to choose values of T_1, T_2 in our merging process. See the results in Figure 3.

$$Edge\ \ strength = \frac{\sum_{p \in Edge} Gradient\ \ (p)}{N} \tag{6}$$

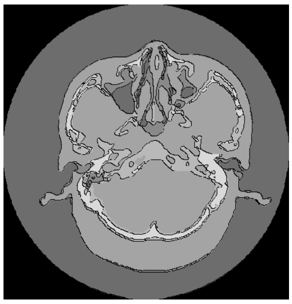
Where Gradient (p) represents *edge points* gradient values which come from the gradient image step (see block diagram in Figure 1) for all pixels (p) on the edge between every two regions, and N are the number of edge pixels.



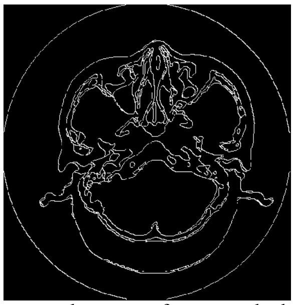
(a) Original image of brain image (512x512).



(b) Seg. image into 6 regions by k-means method.



(c) Then segmented image with edges (region map) by watershed algorithm.



(d) An accurate edge map after watershed & merge process by mean value.

Figure 3. The results of K-means method, watershed algorithm, and merging techniques.

[cpu = 27.835s. [20.936 *s* for K-means. 6.229 *s* for watershed and merging. And 0.670 *s* for two edge strength].

4. Difference in Strength Technique

The DIS for each pixel was calculated using equation (7) [10]. After processing all the input pixels, the DIS map was obtained. In DIS map, the larger the DIS value is, the more the pixel is likely located at the edge. At this step, a 3x3 window runs pixel by pixel on the input image. When the window runs over the bolder of the input image, pixels outside the bolder are given the gray level of the input nearest to it. The DIS for the center pixel as in Figure 4, for example, was calculated as in equation (7) [10].

$$\begin{aligned}&|Z_1-Z_3|+|Z_1-Z_5|+|Z_1-Z_6|+|Z_1-Z_7|+|Z_1-Z_8|\\&+|Z_2-Z_4|+|Z_2-Z_5|+|Z_2-Z_6|+|Z_2-Z_7|+|Z_2-Z_8|\\&|Z_3-Z_4|+|Z_3-Z_6|+|Z_3-Z_7|+|Z_3-Z_8|+|Z_4-Z_5|\\&+|Z_4-Z_7|+|Z_4-Z_8|+|Z_5-Z_6|+|Z_5-Z_7|+|Z_6-Z_8|\end{aligned} \tag{7}$$

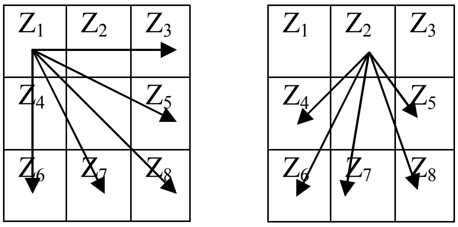
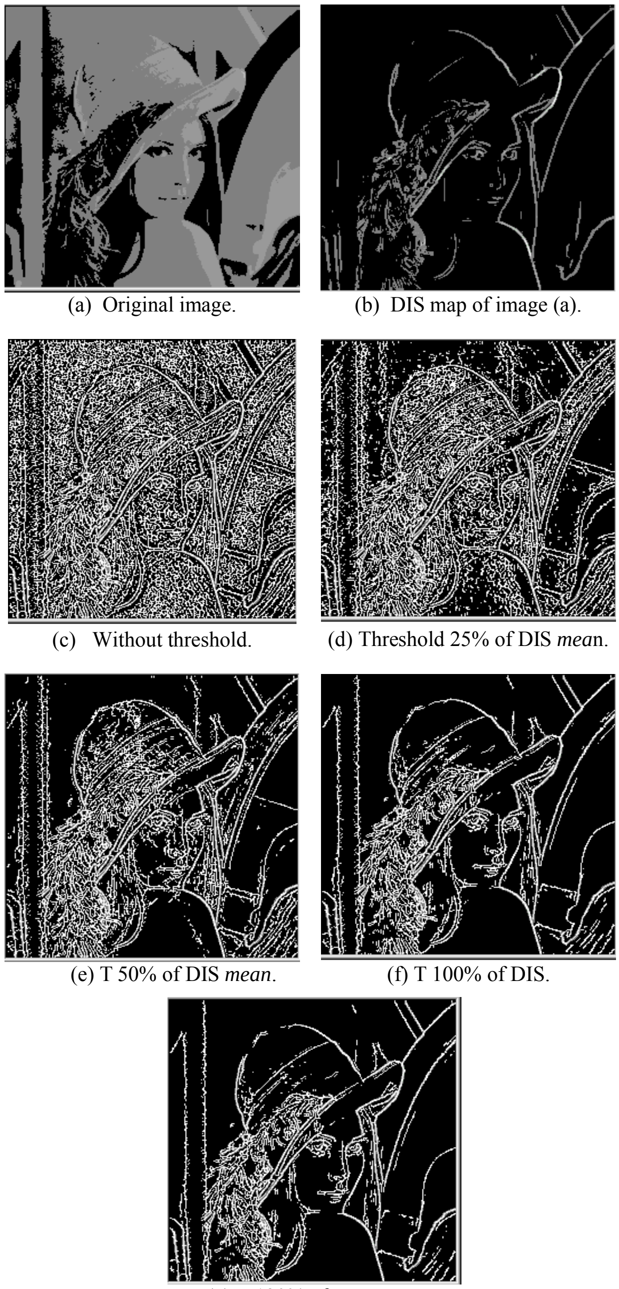


Figure 4. The DIS detecting windows.

Examples of DIS maps are shown in Figure 5-b. One can expect that the values of DIS should be small in the smooth regions obtained by k-means. The greater DIS value represents that the pertaining pixel is on the area that changes severely in gray levels. With the DIS map one can check with the result of image segmentation based on K-means. It is clear that the DIS map consists of all edge information about the input image even on the smooth regions.

Since the DIS of the smooth region is small (weak edge), one can use a threshold T to eliminate false edges and thus obtain larger regions. In this case, the DIS map provides the complete edge (strong and weak) information about the image. By exploiting these information, one can accurately locate the contour of an object. Now to find the effect of DIS, we used multithreshoding edge detection; first we calculated DIS for each pixel in the image then we calculated the mean value of DIS for the whole image. From the mean value we thresholded our image by different % of mean DIS. The threshold for discarding weak edges is set to the mean of DIS as in Figure 5 (d through g).

The threshold used for connected edges is set to the 50% of mean DIS. So, using multi-threshold is important to eliminate false edges and thus obtain larger regions as in Figure 5 (d through g). The region map without threshold is shown in Figure 5-c. As we can see from the Figure 5 (d-g) compared with the Figure 5-c, the concept that an object should have a closed contour help us to eliminate redundant edge pixels and connect the broken contour by using multi-threshold based on different values of mean DIS of the whole image under study. But if we take k-means and then DIS with 25% of mean DIS, we will get all the edges of our images as in the Figure 6 below and we don't need to use watershed technique.



(a) Original image. (b) DIS map of image (a).

(c) Without threshold. (d) Threshold 25% of DIS *mean*.

(e) T 50% of DIS *mean*. (f) T 100% of DIS.

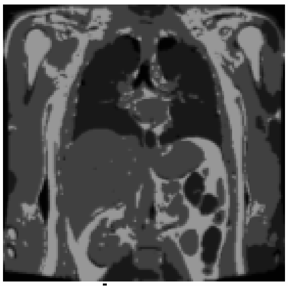
(g) T 120% of *DIS mean*.

Figure 5. DIS map of image and multithreshoding of DIS *mean* edge detection results.

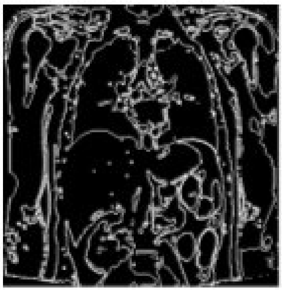
5. Experiment and Results

The experimental results are shown in Figures (3, 5, 6). Medical images as brain images are simple pattern images with the size of {156 x 156} and 256 gray levels images and other images to test our segmentation and edge detection methods. We obtained output images that consist of all edge information and regions about the input image.The region maps are shown in Figure3-c. As can be seen from the edge maps Figure 3-d, that there are no broken lines on the whole image regions. The output image was displayed as an edge map as in Figure 3-d, Figure 5(e through g); and Figure 6-c.

(a) Original image.



(b) After K-means process.



(c) Using threshold 25% of mean DIS.

Figure 6. Edge map using K-means process and thresholding 5% of mean DIS.

6. Conclusion

In our proposed method, the segmentation regions and their boundaries were defined well and all of the boundaries are accurately located at the true edge as shown clearly from Figure3-(c, d), Figure 5-g, and Figure 6-c. And if we take k-means first and then DIS with 25% of mean DIS, we will get all the edges of our images as shown in the Figure 6 above, so we don't need to use watershed technique.

Also, we concluded that using multi-threshold which is important to eliminate false edges and thus obtain larger regions, the DIS map consists of all edge information about the input image even on the smooth regions, and the combination of k-means, watershed segmentation method, DIS map are good techniques to perform image segmentation and edge detection tasks, where the final segmentation results are one closed boundary per actual region of the image under study, and the two *edge strengths* gradient values (T_1, T_2), T_1 is less than T_2, are very sensitive to get good results. Where the incorrect choice of these values gives us uncorrected image segmentation and edge detection results and this is a disadvantage. So we will develop this work in future with automatically determined the threshold values.

Finally, the disadvantages of these techniques depend mainly on k-means results, where if the clustering procedure is not implemented correctly, the results are incorrect by the other techniques we used. However, in this paper we solved the problem of undesirable oversegmentation results produced by the watershed algorithm, also the edge maps we obtained have no broken lines on entire image.

References

[1] Alvarez L., Lion P. L., and Morel J. M., "Image Selective Smoothing and Edge Detection by Non Linear Diffusion," *SIAM Journal*, vol. 29, no. 3, pp. 845-866, 1992.

[2] Chowdhury M. I. and Robinson J. A., "Improving Image Segmentation Using Edge Information," *in Proceedings of the 1st IEEE Conference on Electrical and Computer Engineering*, Halifax, Canada, vol.1, pp. 312-316, 2000.

[3] Gary R. M. and Linde Y., "Vector Quantizers and Predicative Quantizers for Gauss-Markov Sources," *IEEE Transactions on Communication*, vol. 30, no. 2, pp. 381-389, 1982.

[4] Salman N. and Liu C. Q., "Image Segmentation and Edge Detection Based on Watershed Techniques," *International Journal of Computers and Applications*, vol. 25, no. 4, pp. 258-263, 2003.

[5] Tang H., Wu E. X., Ma Q. Y., Gallagher D., Perera G. M., and Zhuang T., "MRI Brain Image Segmentation by Multi-Resolution Edge Detection and Region Selection," *Computerized Medical Imaging and Graphics*, vol. 24, no. 6, pp. 349-357, 2000.

[6] Thrasyvoulos N. P., "An Adaptive Clustering Algorithm for Image Segmentation," *IEEE Transaction on Signal Processing*, vol. 40, no. 4, pp. 901-914, 1992.

[7] Tou J. T. and Gonzalez R. C., *Pattern Recognition Principles*, Addison Wesley, USA, pp. 75-97, 1974.

[8] Vincent L. and Soille P. "Watershed in Digital Space: An Efficient Algorithm Based on Immersion Simulations," *IEEE Transactions on Pattern Analysis and Machine* Intelligence, vol. 13, no. 6, pp. 583-593, 1991.

[9] Yan M. X. H. and Karp J. S., "Segmentation of 3D Brain MR Using an Adaptive K-means Clustering Algorithm," *in Proceedings of the 4th IEEE Conference on Nuclear Science Symposium and Medical Imaging*, San Francisco, USA, vol.4., pp. 1529-1533, 1995.

[10] Yu Y. and Wang J., "Image Segmentation Based on Region Growing and Edge Detection," *in Proceedings of the 6th IEEE International Conference on Systems, Man and Cybernetics*, Tokyo, vol.6., pp. 798-803, 1999.

Nassir Salman received his BSc, and MSc degrees from Mustansyriah University, Iraq, in 1983 and 1989 respectively, and his PhD degree in pattern recognition and intelligence systems, image processing engineering from Shanghai Jiao Tong University, China. Currently, he is a member of the Computer Science Department, Zarqa Private University, Jordan. His research interests include remote sensing, image processing and image analysis based on image segmentation, and edge detection techniques.