

# Genetic-Based Multi Resolution Noisy Color Image Segmentation

Raghad Jawad Ahmed

**Abstract**—Segmentation of a color image composed of different kinds of regions can be a hard problem, namely to compute for an exact texture fields. The decision of the optimum number of segmentation areas in an image when it contains similar and/or unstationary texture fields. A novel neighborhood-based segmentation approach is proposed. A genetic algorithm is used in the proposed segment-pass optimization process. In this pass, an energy function, which is defined based on Markov Random Fields, is minimized. In this paper we use an adaptive threshold estimation method for image thresholding in the wavelet domain based on the generalized Gaussian distribution (GGD) modeling of sub band coefficients. This method called Normal Shrink is computationally more efficient and adaptive because the parameters required for estimating the threshold depend on sub band data energy that used in the pre-stage of segmentation. A quad tree is employed to implement the multi resolution framework, which enables the use of different strategies at different resolution levels, and hence, the computation can be accelerated. The experimental results using the proposed segmentation approach are very encouraging.

**Keywords**—Color image segmentation, Genetic algorithm, Markov random field, Scale space filter.

## I. INTRODUCTION

IMAGE segmentation [1], [2] plays an important role in scene analysis and image understanding. Many techniques, which have been proposed in this area, can be coarsely classified into the following categories:

- Histogram based segmentation
- Neighborhood based segmentation
- Surface fitting based segmentation
- Physically based segmentation

This paper addresses the problem of segmenting a noisy color image using energy statistical models. Each pixel in the observed image must be assigned membership to one of a finite number of classes depending on statistical properties of the pixel and its neighbors. The individual pixel classifications or labels form a matrix or two dimensional fields with the same dimensions as the observed image in which the value at a given spatial location reflects the class to which the corresponding pixel in the observed image belongs. This matrix containing the individual pixel classifications will be referred to as the label field. The label field is unknown and must be estimated from the observed image.

The segmentation problem will be formulated as an optimization problem. The optimization criterion which will be used is the minimization of the expected value of the number of misclassified pixels.

The technique discussed in this paper is a neighborhood-based approach. Basically, neighborhood based approaches use both the intensity information and the spatial knowledge, which means that the label of a pixel depends not only on its color but also on its neighbors' colors. Therefore, these approaches are more robust with respect to noise than histogram based approaches, where the clustering is done in the measurement space only. However, for neighborhood-based approaches, the clustering information of the image is required *a priori*.

We use in our paper a wavelet sated approach. A wavelet thresholding is that as the wavelet transform is good at energy compaction, the small coefficient is more likely due to noise and large coefficient due to important signal features [3]. These small coefficients can be threshold without affecting the significant features of the image. The thresholding technique is useful to reduce the number of the coefficients passing to quad tree therefore reduces the time splitting operation, and increase the efficiency of the segment results.

Thresholding is a simple non-linear technique, which operates on one wavelet coefficient at a time. In its most basic form, each coefficient is threshold by comparing with the threshold, if the coefficient is smaller than threshold, set to zero; otherwise it is kept or modified. Replacing the small coefficients by zero and inverse wavelet transform on the result may lead to reconstruction with the essential signal characteristics and with less noise.

Since the work of Donoho & Johnstone [4], [5], [6], [7], there has been much research on finding thresholds, however few are specifically designed for images. In this paper, a near optimal threshold estimation technique for image denoising is used which is sub band dependent i.e. the parameters for computing the threshold are estimated from the observed data, one set for each sub band, Markov Random Fields (MRF's) is a commonly used method in the neighborhood-based approach [8], [9].

Within this framework, the segmentation process is equivalent to finding the optimum state of the MRF. Because of the Gibbs' equivalence, the probability that the MRF is in a particular state can be calculated using local energies information. Consequently, a given segmentation result can be evaluated by modeling local interactions, and the problem of finding the best segmentation can be viewed as finding the solution to a combinatorial optimization problem [10].

To solve such a combinatorial optimization problem is not a trivial problem because of the large search space. Techniques such as simulated annealing and genetic algorithm are often employed. Simulated annealing is modeled based on the physical process of cooling while the genetic algorithm is based on evolution and natural selection. The latter one is used here because it is possible to tailor the genetic operators to special structures like a quad tree.

Recently, researchers have investigated the application of genetic algorithms (GA, [11], [3], [12]) into the image segmentation problem. Perhaps the most extensive and detailed work on GAs within image segmentation is that of *Bhanu and Lee* [13]. Many general pattern recognition applications of this particular paradigm can also be found in [8]. One reason (among others) for using this kind of approach is mainly related with the GA ability to deal with large, complex search spaces in situations where only minimum knowledge is available about the objective function.

For example, most existing image segmentation algorithms have many parameters that need to be adjusted. The corresponding search space is in many situations quite large and there are complex interactions among parameters, namely if we are seeking to solve color image segmentation problems.

For instance, this led *Bhanu et al.* [13] to adopt a GA to determine the parameter set that optimize the output of an existing segmentation algorithm under various conditions of image acquisition. That was the case for the optimization of the *Phoenix* segmentation algorithm [14], by genetic algorithms, implementation described also by *Bhanu* [5]. Another situation wherein GAs may be useful tools is illustrated by the work of *Yoshimura and Oe* [15].

In their work, the two authors formulated the segmentation problem upon textured images as an optimization problem, and adopt GAs for the clustering of small regions in a feature space, using also *Kohonen's* self-organizing maps (SOM).

They divided the original image into many small rectangular regions and extracted texture features from the data in each small region by using the two-dimensional autoregressive model (2D-AR), fractal dimension, mean and variance. In other example, *Bhandarkar et al.* [10] defined a multi-term cost function, which is minimized using a GA-evolved edge configuration.

The idea was to solve medical image problems, namely edge-detection. In their approach to image segmentation, edge detection is cast as the problem of minimizing an objective cost function over the space of all possible edge configurations and a population of edge images is evolved using specialized operators. Results comparable with those obtained using simulated annealing are reported. Fuzzy GA fitness functions were also considered by *Chun and Yang* [16], mapping a region based segmentation onto the binary string representing an individual, and evolving a population of possible segmentations.

In this paper, a Genetic-Based Multi resolution Color Image Segmentation algorithm is presented. The proposed approach consists of three steps, which are the preprocess phase, the minimization of an MRF-based energy function, and the optimization of segmentation with respect to a given evaluation criterion.

The second section discusses how the Preprocess Phase is computed first using the Scale-Space Filter based histogram thresholding and the preprocess stage, a quad tree representation of the image is constructed and the color information of each node in the quad tree is computed, the third section the GA is used to minimize the energy function, which is defined based on the MRF. Detailed issues are addressed first, which include the encoding mechanism for all possible segmentations, the formulation of the energy function, and the appropriate crossover and mutation operators to be used. An outline of the minimization process is given latter, the fourth section present the Experimental Results, evaluate the performance of the proposed algorithm. One of them, the "cylinder" is a synthesized image. Gaussian noise with a standard deviation 50 is added to it. Two others, the "flower" and the "girl" are noisy real images, the fifth section conclusion and acknowledgment.

## II. PREPROCESS PHASE

Since the clustering information of the input image is required *a priori* for the MRF-based approach, this information, e.g. the number of clusters and the mean color of each cluster, is computed first using the Scale-Space Filter based histogram thresholding [17].

In addition, in the preprocess stage, a quad tree representation of the image is constructed and the color information of each node in the quad tree is computed. There are two advantages to employing quad tree structure here. First, it gives a multi-resolution representation so that different strategies can be used at different resolution levels. Second, since each upper level node in a quad tree covers many pixels, the color errors introduced by assigning these pixels to different clusters can be efficiently computed using pre-calculated color information of the node.

### 2.1 Scale-space Filter based thresholding

#### 2.1.1 Wavelet Thresholding

The basic idea of using scale-space filter [17] is to analyze the histogram and energy distribution of the original image and to find the fingerprint of the histogram. Given an image, this approach uses Gaussian functions of different scale to smooth the histogram and then detects the locations of peaks and valleys of energy coefficients information. It is noteworthy that in the scale-space filter algorithm, the user can control the coarseness of clusters by selecting different maximum scale  $\sigma_{\max}$ . Generally, a larger value of  $\sigma_{\max}$  gives a smaller number of clusters while a smaller value of  $\sigma_{\max}$  generates a larger number, but smaller in size, clusters.

Let  $f = \{f_{ij}, i, j = 1, 2, \dots, M\}$  denote the  $M \times M$  matrix of the original image to be recovered and  $M$  is some integer power of 2. After analyzing the signal  $f$ , the signal can be distributed by independent or identically distributed (i.i.d) zero mean, Gaussian distribution  $n_{ij}$  with standard deviation  $\sigma$  i.e.  $n_{ij} \sim N(0, \sigma^2)$ , different noise. The noisy observations  $g_{ij} = f_{ij} + \sigma n_{ij}$  is obtained. The goal is to estimate energy level of the signal  $f$  in order to generate image clusters from observations  $g_{ij}$  such that Mean Squared error (MSE) [11] is minimum. Let  $W$  and

$W^{-1}$  denote the two dimensional orthogonal discrete wavelet transform (DWT) matrix and its inverse respectively. Then  $Y = Wg$  represents the matrix of wavelet coefficients of  $g$  having four sub bands (LL, LH, HL and HH) [16], [18]. The sub-bands HHk, HLk, LHk are called details, where  $k$  is the scale varying from 1, 2 ...  $J$  and  $J$  is the total number of decompositions.

The size of the sub band at scale  $k$  is  $N/2^k \times N/2^k$ . The sub band  $LL_J$  is the low-resolution residue. The wavelet thresholding segmentation method processes each coefficient of  $Y$  from the detail sub bands with a soft threshold function to obtain clusters. In the experiments, soft thresholding has been used over hard thresholding because it gives more visually pleasant images as compared to hard thresholding; reason being the latter is discontinuous and yields abrupt artifacts in the recovered images especially when the noise energy is significant.

### 2.1.2 Estimation of Parameters for clustering

This section describes the method for computing the various parameters used to calculate the threshold value (TN), which is adaptive to different sub band characteristics.

$$T_s = \frac{\sigma\beta^2}{\sigma_y} \quad (1)$$

Where, the scale parameter  $\beta$  is computed once for each scale using the following equation:

$$\beta = \text{SQRT}(\log(\frac{L_k}{J})) \quad (2)$$

$L_k$  is the length of the sub band at  $k^{\text{th}}$  scale and is the constant of average the sub band at  $k^{\text{th}}$ . And SQRT is the square root.

$\sigma^2$  is the variance, which is estimated from the sub band LL1, using the formula [16][18]:

$$\sigma^2 = \left( \frac{\text{median}(|Y_{ij}|)}{0.6745} \right)^2 \quad (3)$$

Where  $Y_{ij}$  is in the sub band LL1 and  $\sigma_y$  is the standard deviation of the sub band. To summarize, the proposed method is named as *Normal Shrink* which performs soft thresholding with the data driven sub band dependent threshold  $T_s$ .

The above process can handle gray scale images only. For color images, histogram of wavelet coefficients energy thresholding is performed on all three color channels(RGB), and then the thresholded results are combined together to partition the three-dimensional color space [19].

Assume that the scale-space filter algorithm partitions each of the red ( $R$ ), green ( $G$ ), and blue ( $B$ ) channels. Then the color space can be partitioned by  $U(ru,gv,bw) = ru \leq gv \leq bw$ ,

where  $\leq$  denotes the cross-product operator, and  $ru$ ,  $gv$  and  $bw$  denote a cluster in the  $R$ ,  $G$  and  $B$  space, respectively. However, not every partition corresponds to a prominent cluster in the color space. The commonly used approach [8] is to sort these partitions according to the number of coefficients in the clusters, and a parameter is used to pick the most dominant clusters.

After the cluster information is obtained, the color image can be segmented through assigning each coefficients (pixel) to the closest cluster based on the Euclidean distance between coefficients and threshold  $T_s$  in the color space. As shown in Fig. 1, the segmentation results obtained this way are sensitive to noise due to the lack of local spatial knowledge. In our approach, the scale-space filter is only used to provide the whole the coefficients energy of the clusters energy distribution compare with the threshold  $T_s$ . The labeling of pixels is determined by the genetic-based optimization process.

### 2.2 Quad tree Construction

Quad tree is a data structure that represents a two-dimensional image hierarchically. The root of the quad tree represents the whole image coefficients and each child represents one quarter of the area that is represented by its parent. For each node  $k$  of the quad tree, the number of coefficients (pixels) in the node,  $N_k$ , the mean of colors of  $k$ ,  $\mu_k$ , and the mean of squares of colors of  $k$ ,  $\omega_k$ , are computed as:

$$\mu_k = (\mu_k^r, \mu_k^g, \mu_k^b) = \left( \frac{1}{N_k} \sum_{i=1}^{N_k} C_i^r, \frac{1}{N_k} \sum_{i=1}^{N_k} C_i^g, \frac{1}{N_k} \sum_{i=1}^{N_k} C_i^b \right)$$

$$\omega_k = (\omega_k^r, \omega_k^g, \omega_k^b) = \left( \frac{1}{N_k} \sum_{i=1}^{N_k} C_i^{r^2}, \frac{1}{N_k} \sum_{i=1}^{N_k} C_i^{g^2}, \frac{1}{N_k} \sum_{i=1}^{N_k} C_i^{b^2} \right)$$

where  $C_i$ ,  $C_i^g$  and  $C_i^b$  are the values of the  $R$ ,  $G$  and  $B$  channel of coefficient (pixel)  $i$ , respectively.

The above parameters are independent of the segmentation results. Hence, they only need to be calculated once in the preprocess stage. With this complete quad tree is used so that we can encode different parameters, when we classify pixels of node  $k$  into cluster quad trees using a one-dimensional integer string. In such  $x$ , whose mean color is  $L_x$ , the color error  $e_k(L_x)$  can be a representation, the index of node  $k$  in the string can be efficiently computed by:

$$e_k(L_x) = \sum_{i=1}^{N_k} \|C_i - L_x\|^2$$

$$= \sum_{i=1}^{N_k} ((C_i^r - L_x^r)^2 + (C_i^g - L_x^g)^2 + (C_i^b - L_x^b)^2)$$

$$= N_k \times \left( \omega_k^r - L_x^r (2\mu_k^r - L_x^r) + \omega_k^g - L_x^g (2\mu_k^g - L_x^g) + \omega_k^b - L_x^b (2\mu_k^b - L_x^b) \right)$$

### MRF-based Energy Minimization

#### The First Pass

In this stage, the GA is used to minimize the energy function, which is defined based on the MRF. In the following

subsections, detailed issues are addressed first, which include the encoding mechanism for all possible segmentations, the formulation of the energy function, and the appropriate crossover and mutation operators to be used. An outline of the minimization process is given latter.

### 3.1 Encoding Scheme for Segmentations

Fundamental to all GA's is the encoding scheme for representing the solutions of the corresponding optimization problems. Normally, the method to encode the solutions depends not only on the applications to which the GA is applied but also on the genetic operations used. For example, in Chun and Yang's approach [20] the solution is encoded by the labeling status of all the small regions, and in Bhandarkar and Zhang's approach [10] the representation of each solution contains the membership label array, the edge image, and the region adjacency graph.

As discussed in section 2, the quad tree representation of an image is employed in our approach. Here, the segmentation results are also represented by quad trees and the solutions are encoded through encoding the corresponding quad trees. The quad trees used to represent the segmentation results must satisfy the following two constraints:

- Every leaf (i.e. a node with no child)  $k$  of the quad tree has an associated label  $x$ , which implies that all the pixels covered by  $k$  are assigned to cluster  $x$  in the segmentation result. A leaf  $k$  is said to cover pixel  $p$  if  $k$  contains  $p$ . [18]
- Any interior node in the quad tree cannot have all its descendants assigned to the same label; otherwise, all the descendants of the node should be removed and the node itself should be selected as the leaf. [19].

Now we need to find a way to encode all the possible quad trees. In this paper, an array representation of a complete quad tree is used so that we can encode different quad trees using a one-dimensional integer string. In such a representation, the index of node  $k$  in the string can be computed by:

$$O(k) = \sum_{i=0}^{h-1} 4^i + Yx2^h + X$$

Where  $h$ ,  $x$ , and  $y$  are the height and the  $x$ -,  $y$ -coordinates of node  $k$ , respectively. The content at location  $O(k)$  in the integer string is determined by:

$$S[O(k)] = \begin{cases} X & \text{if node } k \text{ is a leaf and its label is } x; \\ -1 & \text{otherwise} \end{cases}$$

### 3.2 Fitness Evaluation

The fitness of a given chromosome, which is represented as a string, controls the evolution process. The fitter the chromosome, the greater is its probability to survive from one generation to the next. In the context of image segmentation, we need a way to evaluate the results of different segmentations.

Here, in the first pass, we use an energy function, which is based on the MRF as the fitness evaluation tool. The neighborhood system we used for the MRF in this paper is similar to the one introduced by Liu and Yang [8]. Basically, the neighbor of node  $k$  is the following set:

$$Q_k = \left\{ d \left| \begin{array}{l} \text{node } d \text{ and } k \text{ have shared boundary, and} \\ \text{node } d \text{ and } k \text{ do not overlap each other} \end{array} \right. \right\}$$

Under this neighborhood system, the energy function we used is defined as follows:

$$F(S) = \sum_{K \in P} (e_k(L_{S[K]}) + \lambda T_K)$$

Where  $S$  denotes the segmentation string,  $P$  the set that contains all the leaves in the quad tree,  $S[k]$  the label of leaf  $k$ . The smaller the value of  $f(S)$ , the better the segmentation is.  $\lambda$  is the weight of the penalty term with a small value of  $\lambda$  favors a finer segmentation and a large value encourages a coarser result.  $e_k$  is the color error, which is computed using equation (1).  $T_k$  is the length of the boundary of leaf  $k$ , which is defined as:

$$T_K = \sum_{d \in P \cap Q_K} (1 - \delta(S[K] = S[d])) \times t_{k,d}$$

Where  $\delta(\text{true})=1$ ,  $\delta(\text{false})=0$ , and  $t_{k,d}$  the length of the shared boundary between leaf  $k$  and leaf  $d$ .

The energy function defined above is a local measure. This means that when we change the label of a node, only the energy of node  $k$  and its neighbors will be affected. This feature enables us to evaluate the crossover and mutation effects of GA efficiently.

### 3.3 Initial Population Generation

The GA needs a number of initial segmentations as the initial population to start with. The choice of the population size is very important. If the selected population size is too small, then the algorithm may result in premature convergence without finding an appropriate solution. On the other hand, a large population size will lead to long computation time. Our experiments show that premature convergence is likely to occur when the population size is smaller than 30. In this paper, the population size is set to 40, which appears to be appropriate to avoid the problem of premature convergence.

The initial segmentations are generated purely randomly. First, a recursive procedure is invoked using the root of the quad tree as the input parameter. Inside the procedure, whether or not the given parameter, the node  $k$ , is selected as a leaf depends on a random number. If the decision is to select it as a leaf then a label  $x$  is selected and assigned to node  $k$  such that the color error  $e_k(L_x)$  is minimized. Otherwise, the procedure invokes itself recursively using the four child nodes of  $k$  until the bottom level is reached.

### 3.4 Crossover Operator

In our scenario, it will be inefficient to apply the commonly used crossover operators, such as the two-point crossover

operator used by Chun and Yang [20], because they do not guarantee that the crossover results of the two quad trees will still be legal quad trees. To address this problem, a new crossover operation, called graft crossover, is proposed.

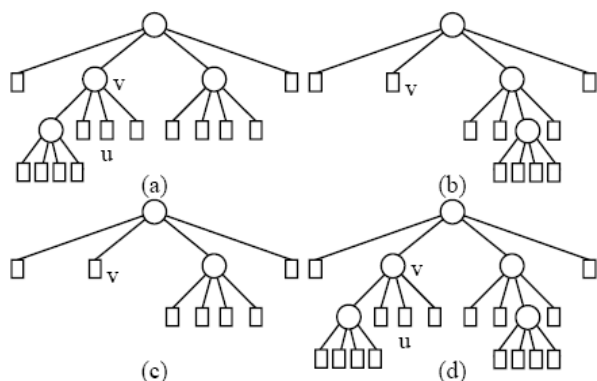


Fig. 1 Graft crossover for quad trees, (a) and (b) the parents; (c) and (d) the offspring

Given two strings, which represent two quad trees, we want to compare them and find out all the leaves that appear in only one of the two quad trees. The crossover process will be terminated if no such leaves are found. Otherwise, we randomly select one of these leaves as a *seed node*. For example, after comparing the two quad trees shown in Fig. 1(a) and (b), we can pick leaf *u* since it appears only in the left quad tree. Then the *cover node* that is the predecessor of the *seed node* and appears in both quad trees is determined (it is node *v* in our example). Finally, we swap all the nodes that are the descendants of the *cover node*. Fig. 1(c) and (d) show the results after we do the graft crossover at *cover node v*.

By construction, this algorithm guarantees that the results of crossover will still be legal quad trees. After crossover, the energies of the two quad trees may change and one of the offspring's strings may have a lower energy than either of its parents.

### 3.5 Mutation Operator

The mutation operation is important for the GA since the crossover operator cannot generate offspring that have genes that do not appear in the initial population. In our approach, we introduce three mutation operations, which are the splitting, the merging, and the alteration.

The mutation operator randomly selects a number of pixels from the original image. In our experiments, the number of pixels selected is equal to one eighth of the perimeter of the image. For each pixel, we search for leaf *k* in the quad tree that covers this pixel. How to mutate leaf *k* depends on its resolution level. If leaf *k* is in the bottom two levels of the quad tree, i.e. it contains at most 4 pixels; we will apply one of the splitting, merging, and altering operations with equal chance. Otherwise, we will pick one of the splitting and merging operations randomly. Therefore, by using different strategies at different resolution levels, we can prohibit

altering operations for leafs that have size larger than  $2 \times 2$  pixels.

When trying to merge leaf *k* with its siblings, we need to find out all of its siblings first. The merging operation will be inhibited if any of its siblings has children. Otherwise, the local energy is computed, and the merging operation will happen only if the energy is lower after merging than that before merging.

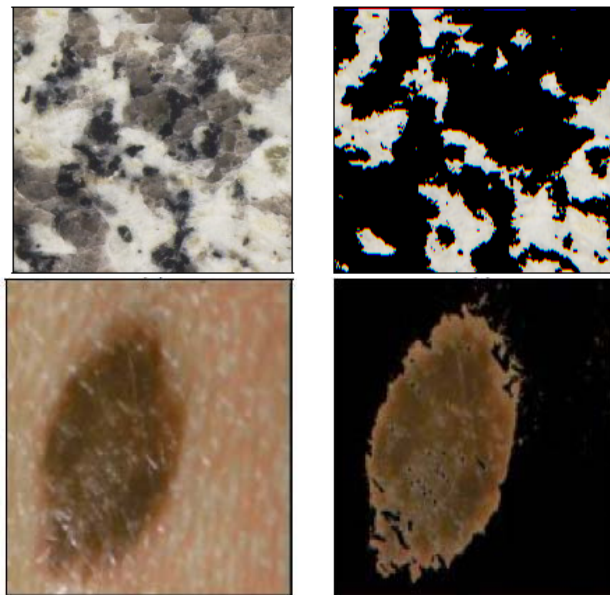
When trying to split leaf *k*, the process is similar. The local energies are computed for segmentations before and after splitting. The splitting operation will occur only when the energy decreases after splitting [20].

When trying to alter leaf *k*, all the neighbors of *k* are found first. If one of the neighbors has the same label as leaf *k* has, then the alteration operation will be inhibited. Otherwise, the alteration operation will change the label of leaf *k* to the label of one of the labels of its neighbors, which can reduce the energy.

## IV. EXPERIMENTAL RESULTS

Four images are used to evaluate the performance of the proposed algorithm. One of them, the "cylinder" is a synthesized image. Gaussian noise with a standard deviation 50 is added to it. Two others, the "flower" and the "girl" are noisy real images.

The last image is composite by putting four different textures, which are lawn, brick, wood, and cement, together. To avoid favoring the quad tree data structure, which has a bias for horizontal and vertical edges, the texture image are rotated by 45 degrees. The original images and the contours of the scale-space filter segmentation results are shown in Fig. 2



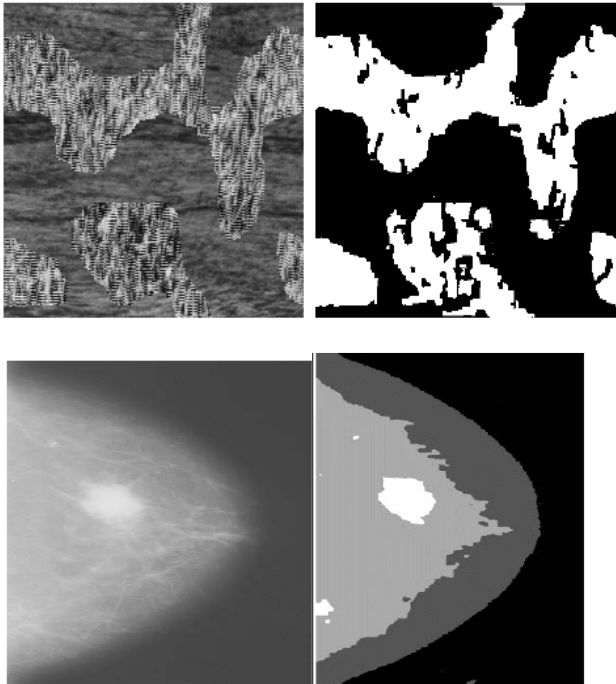


Fig. 2 The results of the proposed system

Fig. 2 shows the final result obtained through the genetic optimization process. Since the segmentation results generated from the quad tree splitting pass are optimized using the genetic crossover operator, the result is even better than the best solution available. This argument is supported by the experiment result. Indeed, as shown in Fig. 2, the evaluation values of the second pass results are smaller than the best results from the first pass. The improvement is as high as 11 percents for the image “cylinder” with respect to the best solution available from the first pass.

## V. CONCLUSION

In this paper, we introduce a new neighborhood-based algorithm for image segmentation. The segmentation algorithm is used in the two-pass optimization process. In the first pass, an energy function, which is defined based on MRF, is minimized. Through adjusting the number of pixels(coefficients) that will be segment depending on the distributed energy by using the wavelet thresholding technique. Also using the quad tree to splitting the coefficients depending on the distance and energy reduced the operation time and increase the efficiency of the proposed algorithm, these preprocessing operation can give a set of coarse to fine segmentation results. In the second pass, the segmentation results obtained in the first pass with different values are used as the initial population and are optimized with respect to the given evaluation criterion of genetic like fitness, crossover and mutation operators.

The quad tree structure is used to implement the multi resolution framework, which enables the use of different strategies at different resolution levels and accelerates the

computation. To apply the GA under the multi resolution framework, a new encoding mechanism for the quad tree structure is proposed.

Normally, the computational costs of genetic-based algorithms are very high. our approach takes about 500-800 generations to converge. However, since useful information is calculated in the preprocess stage and the energy function used in the first pass is locally defined, the computation time is reduced. To segment an image of size  $128 \times 128$  on our Pentium 4 2.8GHz computer with 1024MB RAM running Windows XP, our algorithm converges in less than 5 seconds.

If the user wants to search for a good segmentation in terms of a given criterion instead of specifying the value of  $\lambda$ , our approach will use 30 different segmentations obtained using different values of  $\lambda$  as initial segmentations to start the second pass. Since only the crossover operator is used in the second pass, the algorithm needs 300-400 generations to converge.

The second pass itself takes about 50 seconds on the same computer and the whole two-pass process can be finished in about 5 minutes, which includes the time to compute 30 segmentations in the first pass. However, the benefit we gain in using the two-pass approach is that better segmentation results are obtained in terms of the given criterion since the algorithm is optimizing the evaluation function directly and the genetic algorithm is good at searching for the global optimum.

In summary, our experimental results suggest that the proposed approach is flexible, efficient, and is able to generate an acceptable segmentation result automatically.

## ACKNOWLEDGMENT

I wish to my thank and grateful to my study supervisor Prof. Stefan Trausan-Matu Phd, Computer Science and Engineering Dept., University Politehnica of Bucharest, for his continuous support and significant advice as well as academic assistance, honest help. And for all the encouragement he gave me and for his valuable advice.

## REFERENCES

- [1] R. M. Haralick and L. G. Shapiro, "Image Segmentation Techniques," *CVGIP*, Vol. 29, No. 1, pp. 100-132, January, 1985.
- [2] N. R. Pal and S. K. Pal, "A Review on Image Segmentation Techniques," *Pattern Recognition*, Vol. 26, No. 9, pp. 1277-1294, September, 1993.
- [3] M. Celenk, "A Color Clustering Technique for Image Segmentation," *CVGIP*, Vol. 52, No. 2, pp. 145-170, November, 1990.
- [4] Bhandarkar, S.M., Zhang, Y. and Potter, W.D., 1994, "An Edge Detection Technique using Genetic Algorithm-based Optimisation", *Pattern Recognition* 27(9), pp. 1159-1180.
- [5] B. Bhanu, S. Lee, and J. Ming, "Adaptive Image Segmentation Using a Genetic Algorithm," *IEEE Trans on Systems, Man and Cybernetics*, Vol. 25, No. 12, pp. 1543-1567, December, 1995.
- [6] D. N. Chun and H. S. Yang, "Robust Image Segmentation Using Genetic Algorithm with a Fuzzy Measure," *Pattern Recognition*, Vol. 29, No. 7, pp. 1195-1211, July, 1996.
- [7] H. Derin and H. Elliott, "Modeling and Segmentation of Noisy and Textured Images Using Gibbs Random Fields," *IEEE Trans. on PAMI*, Vol. 9, No. 1, pp. 39-55, January, 1987.
- [8] J. Liu and Y. H. Yang, "Multi resolution Color Image Segmentation," *IEEE Trans. on PAMI*, Vol. 16, No. 7, pp. 689-700, July, 1994.
- [9] S. H. Park, I. D. Yun, and S. U. Lee, "Color Image Segmentation Based on 3-D Clustering," *Pattern Recognition*, Vol. 31, No. 8, pp. 1061-1076, August, 1998.

- [10] Andrey, P., 1999, "Selections Relaxation: Genetic Algorithms applied to Image Segmentation", *Image and Vision Computing* 17, pp. 175-187.
- [11] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Reading, MA, USA, 1989.
- [12] Chun, D.N. and Yang, H.S., 1996, "Robust Image Segmentation using Genetic Algorithm with a Fuzzy Measure", *Pattern Recognition* 29(7), pp. 1195-1211.
- [13] S. M. Bhandarkar and H. Zhang, "Image Segmentation Using Evolutionary Computation," *IEEE Trans. on Evolutionary Comp.*, Vol. 3, No. 1, pp. 1-21, April, 1999.
- [14] H. J. Kim, E. Y. Kim, J. W. Kim, and S. H. Park, "MRF Model Based Image Segmentation Using Hierarchical Distributed Genetic Algorithm," *Electronics Letters*, Vol. 34, No. 25, pp. 2394-2395, December 10, 1998.
- [15] G. J. Klinker, S. A. Shafer, and T. Kanade, "Physical Approach to Color Image Understanding," *Inter. J. of Computer Vision*, Vol. 4, No.1, pp. 7-38, January, 1990.
- [16] Cagnoni, S., Dobrzeniecki, A.B., Poli, R. and Yanch, J.C., 1999, "Genetic Algorithm-based Interactive Segmentation of 3D Medical Images", *Image and Vision Computing* 17, pp. 881-895.
- [17] Davis, L.S. and Rosenfeld, A., 1981, "Cooperating Processes for Low-Level Vision: A Survey", *Artificial Intelligence* 17, pp.245-263.
- [18] R.C.Dubes, A. K. Jain, S.G. Nadabar, and C. C. Chen, "MRF Model-Based Algorithms for Image Segmentation," In: *Proceedings of the 10th ICPR Vol. 1*, pp. 808-814, Atlantic City, NJ, USA, June 16-21, 1990.
- [19] P. J. Besl and R. C. Jain, "Segmentation Through Variable-Order Surface Fitting" *IEEE Trans. on PAMI*, Vol. 10, No. 2, pp.167-192, March, 1988.
- [20] M. Borsotti, P. Campadelli, and R. Schettini, "Quantitative Evaluation of Color Image Segmentation Results," *Pattern Recognition Letters*, Vol. 19, No. 8, pp. 741-747, June, 1998.