Genetic-Based Multiresolution Color Image Segmentation

Minglun Gong Yee-Hong Yang

Department of Computer Science
University of Saskatchewan

Abstract

The goal of image segmentation is to partition an image into some homogeneous regions. In this paper, a novel neighborhood-based segmentation approach is proposed. Genetic algorithm is used in the proposed two-pass process. In the first pass, an energy function, which is defined based on Markov Random Fields, is minimized. In the second pass, which is optional, segmentation results obtained from the first pass with different values of a parameter are used as initial population and are optimized with respect to a given evaluation criterion. Different evaluation criteria can be used in our algorithm and a new criterion is also proposed. A quadtree is employed to implement the multiresolution framework, which enables the use of different strategies at different resolution levels, and hence, the computation can be accelerated. In addition, a new encoding mechanism for the quadtree structure is proposed. New crossover and mutation operators, namely graft crossover, splitting, merging, and alteration mutation, which are suitable for the quadtree structure, are introduced. These operators guarantee that the resulting tree is a quadtree. The experimental results using the proposed segmentation approach are very encouraging.

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

1 Introduction

Image segmentation [10, 17] 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 [18, 21],
- Neighborhood based segmentation [7, 8, 13, 16],
- Surface fitting based segmentation [1].
- Physically based segmentation [11, 15]

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*.

Markov Random Fields (MRF's) is a commonly used method in the neighborhood-based approach [16, 18]. 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 [2]. 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 quadtree.

The genetic algorithm (GA) proposed by John Holland [12] is a general-purpose global optimization technique based on randomized search and incorporates some aspects of iterative algorithms. It is regarded as an alternative method for solving complex optimization problems. The research in GA and its applications are a topic of active research [19].

Some recent approaches have applied GA to image segmentation [2, 3, 6, 14, 20]. The genetic learning system proposed by Bhanu et al. [3] allows the segmentation process to adapt to image characteristics, which are affected by varying environmental factors such as the time of day, condition on cloudiness, etc.

Chun and Yang [6] present a region-based segmentation approach on grayscale images using a GA with a fuzzy measure. The fuzzy-c-means algorithm is used to generate a fine segmentation. Each individual in the GA is encoded by n integers (n equals to the number of regions in the fine segmentation), each of which indicates the labeling status of the corresponding region. Two-point crossover and dynamic mutation operators are used to optimize the result of fuzzy-c-means algorithm through a split-and-merge process.

Bhandarkar and Zhang [2] use the GA to minimize the cost function that is used to evaluate the segmentation results. The initial population is generated from the initial preprocessed and segmented image via a series of random mutations. The representation of each individual contains the membership label array, the edge image, and the region adjacency graph. Two-point crossover on the membership label array and mutation for the edge pixels are employed.

Tseng and Lai [20] notice that results obtained from the iterated conditional modes algorithm heavily depends on the initial state. Therefore, in their approach, the GA is used to provide a good initialization and the iterated conditional modes algorithm is then used to label pixels.

In this paper, a Genetic-Based Multiresolution 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 rest of the paper is organized as follows. In section 2, 3, and 4, the above three steps are discussed in detail. The experimental results are given in section 5. Finally, conclusions are described in section 6.

2 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 [21].

In addition, in the preprocess stage, a quadtree representation of the image is constructed and the color information of each node in the quadtree is computed. There are two advantages to employing quadtree 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 quadtree 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

The basic idea of using scale-space filter [21] is to analyze the histogram 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 to find cluster 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.

The above process can handle gray scale images only. For color images, histogram thresholding is performed on all three channels, and then the thresholded results are combined together to partition the three-dimensional color space [5]. In this paper, we use the RGB color space to represent color, although other color spaces can also be applied.

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 $U(r_u, g_v, b_w) = r_u \otimes g_v \otimes b_w$, where $\otimes$ denotes the cross-product operator, and r_u, g_v and b_w 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 [16] is to sort these partitions according to the number of pixels 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 pixel to the closest cluster based on the Euclidean distance in the color space [21]. As shown in Figure 2, 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 initial clusters information. The labeling of pixels is determined by the genetic-based optimization process.

2.2 Quadtree Construction

Quadtree is a data structure that represents a two-dimensional image hierarchically. The root of the quadtree represents the whole image and each child represents one quarter of the area that is represented by its parent. For each node k of the quadtree, the number of pixels in the node, N_k, the mean of colors of k, μ_k, and the mean of squares of colors of k, ω_k, are computed as:

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

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

where $\mathbf{C}_i^r$, $\mathbf{C}_i^g$ and $\mathbf{C}_i^b$ are the values of the R, G and B channel of 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 these parameters, when we classify pixels of node k into cluster x, whose mean color is $\mathbf{L}_x$, the color error $e_k(\mathbf{L}_x)$ can be efficiently computed by:

$$
\begin{aligned}
e_k(\mathbf{L}_x) &= \sum_{i=1}^{N_k} \left\| \mathbf{C}_i - \mathbf{L}_x \right\|^2 \\
&= \sum_{i=1}^{N_k} \left((\mathbf{C}_i^r - \mathbf{L}_x^r)^2 + (\mathbf{C}_i^g - \mathbf{L}_x^g)^2 + (\mathbf{C}_i^b - \mathbf{L}_x^b)^2 \right) \quad (1) \\
&= N_k \times \begin{pmatrix} \omega_k^r - \mathbf{L}_x^r (2\mu_k^r - \mathbf{L}_x^r) + \\ \omega_k^g - \mathbf{L}_x^g (2\mu_k^g - \mathbf{L}_x^g) + \omega_k^b - \mathbf{L}_x^b (2\mu_k^b - \mathbf{L}_x^b) \end{pmatrix}
\end{aligned}
$$

3 MRF-based Energy Minimization: the 1st 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 [6] the solution is encoded by the labeling status of all the small regions, and in Bhandarkar and Zhang's approach [2] the representation of each solution contains the membership label array, the edge image, and the region adjacency graph.

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

- Every leaf (i.e. a node with no child) k of the quadtree 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.
- Any interior node in the quadtree cannot have all its descendents assigned to the same label; otherwise, all the descendents of the node should be removed and the node itself should be selected as the leaf.

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

$$O(k) = \left(\sum_{i=0}^{h-1} 4^i \right) + y \times 2^h + x$$

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:

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

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 [16]. Basically, the neighbors of node k is the following set:

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

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

$$f(\mathbf{S}) = \sum_{k \in P} \left(e_k(\mathbf{L}_{\mathbf{S}[k]}) + \lambda T_k \right)$$

where $\mathbf{S}$ denotes the segmentation string, P the set that contains all the leaves in the quadtree, $\mathbf{S}[k]$ the label of leaf k. The smaller the value of $f(\mathbf{S})$, the better the segmentation is. λ is the weight of the penalty term with a small value of λ 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:

$$T_k = \sum_{d \in P \cap Q_k} \left(1 - \delta(\mathbf{S}[k] == \mathbf{S}[d])\right) \times t_{k,d}$$

where δ(true)=1, δ(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 quadtree 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 $e_k(\mathbf{L}_x)$ 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 [6], because they do not guarantee that the crossover results of the two quadtrees will still be legal quadtrees. To address this problem, a new crossover operation, called graft crossover, is proposed.

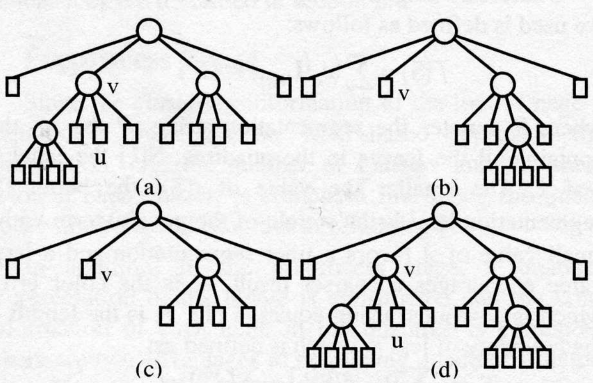


Figure 1: Graft crossover for quadtrees, (a) and (b) the parents; (c) and (d) the offspring.

Given two strings, which represent two quadtrees, we want to compare them and find out all the leaves that appear in only one of the two quadtrees. 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 quadtrees shown in Figure 1(a) and (b), we can pick leaf u since it appears only in the left quadtree. Then the *cover node* that is the predecessor of the *seed node* and appears in both quadtrees is determined (it is node v in our example). Finally, we swap all the nodes that are the descendents of the *cover node*. Figure 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 quadtrees. After crossover, the energies of the two quadtrees 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 quadtree 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 quadtree, 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 2×2 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.

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.

3.6 Minimization Process

After the above issues are addressed, the GA can be implemented as an iterative procedure. When the population evolves from one generation to the next, two strings are picked randomly each time to do the crossover and mutation until all the strings in the population are processed.

The *elitist strategy* [9] is applied when selecting strings for the next generation. The energy value $f(\mathbf{S}_i^m)$ of the best string $\mathbf{S}_i^m$ of generation m is compared with the energy value $f(\mathbf{S}_j^{m+1})$ of the worst string $\mathbf{S}_j^{m+1}$ of generation m+1. If $f(\mathbf{S}_i^m) < f(\mathbf{S}_j^{m+1})$, then string $\mathbf{S}_i^m$ is substituted for $\mathbf{S}_j^{m+1}$. By means of this strategy, the minimum energy value of the population will not increase during the process of evolution.

The above process is repeated until the termination condition is satisfied. In our approach, the evaluation process will be terminated if the energy difference between the best string and the worst string in the population is smaller than 0.1 percent of the average energy value.

4 Segmentation Optimization: the 2nd Pass

Given different values of λ, the algorithm discussed in the above section will give different segmentation results. On the one hand, this feature gives the user the flexibility to fine tune the segmentation results. On the other hand, the user has to face with the problem of choosing a suitable value of λ for a given image.

To make it possible to finish the segmentation without human interactions, we add a segmentation optimization stage to the above segmentation process. The GA and the quadtree structure are still used in the new stage. The encoding scheme for segmentations and the crossover operator that are used in this section are the same as that discussed above.

4.1 Fitness Evaluation

In order to segment an image without human interactions, we need an evaluation criterion that does not contain any parameter or threshold value. Liu and Yang [16] propose a parameter-free measure in 1994. Based on it, Borsotti et al. [4] give two enhanced functions that correspond more closely to visual judgment. Zhang [22] presents a survey on different evaluation methods for image segmentation.

Our approach gives the users the flexibility to choose which criterion to use. The algorithm tries to search for the best segmentation in terms of the given criterion. A new criterion that is modified based on Liu and Yang's approach is proposed in the paper and is used to generate the results that are shown in Figure 4.

Our motivation to propose a new measure is to increase the penalties to regions that have a large color error since we notice that the measures proposed in [16] and [4] do not penalize those regions sufficiently. This is done through modifying "the sum of the Euclidean distances" term in Liu and Yang's approach to "the sum of the squares of the Euclidean distances" term. The definition of the new criterion is:

$$F(\mathbf{S}) = \frac{10^{-6} \times \sqrt{R}}{N \times M} \times \sum_{i=1}^{R} \frac{Z_i^2}{\sqrt{A_i}}$$

where R is the number of regions of the segmented image, $N \times M$ the image size, A_i the area of region i, and Z_i the sum of the squares of the Euclidean distances between the color vectors of the pixels in region i and the color vector assigned to region i in the segmented image. The smaller the value of $F(\mathbf{S})$, the better the segmentation result is.

4.2 Initial Population Generation

In this stage, we use the segmentation results obtained from the first pass with different values of λ as the initial population. Experimentally, we found that a suitable range of λ is [0.01, 0.2]. When the value of λ is smaller than 0.01 the algorithm is too sensitive to noise, and when it is larger than 0.2 the algorithm will ignore all the small regions. Hence, the value of λ is varied from 0.01 to 0.2 with equal intervals to generate 30 different segmentation results, which are used as the new initial population in this stage.

4.3 Optimization Process

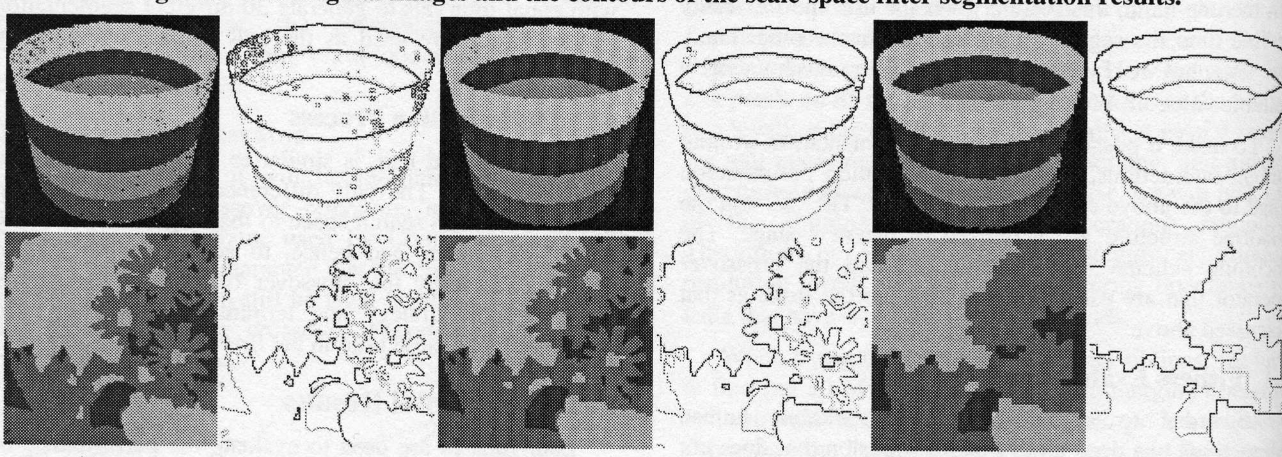
The GA used here is similar to the one described in the last section. The only difference is that: in order to limit the solution search space to the neighborhood of the initial population, and hence, to accelerate the rate of convergence, only the crossover operator is used here. The selection strategy and termination condition used here is same as those used in the first pass.

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

Figure 3 demonstrates the effect of the penalty term λ when using the first pass optimization only. We can see that using different values of λ give different level of details. With a higher value of λ, the segmentation result is very robust to noise. However, the drawback of using a higher value of λ is that the result may not be able to follow the detail of the image very well. Which value of λ can yield the best segmentation in terms of visual judgment depends on how noisy the original image is.

Figure 2: The original images and the contours of the scale-space filter segmentation results.



λ=0.01 λ=0.04 λ=0.16

Figure 3: The effect of changing the parameter λ.

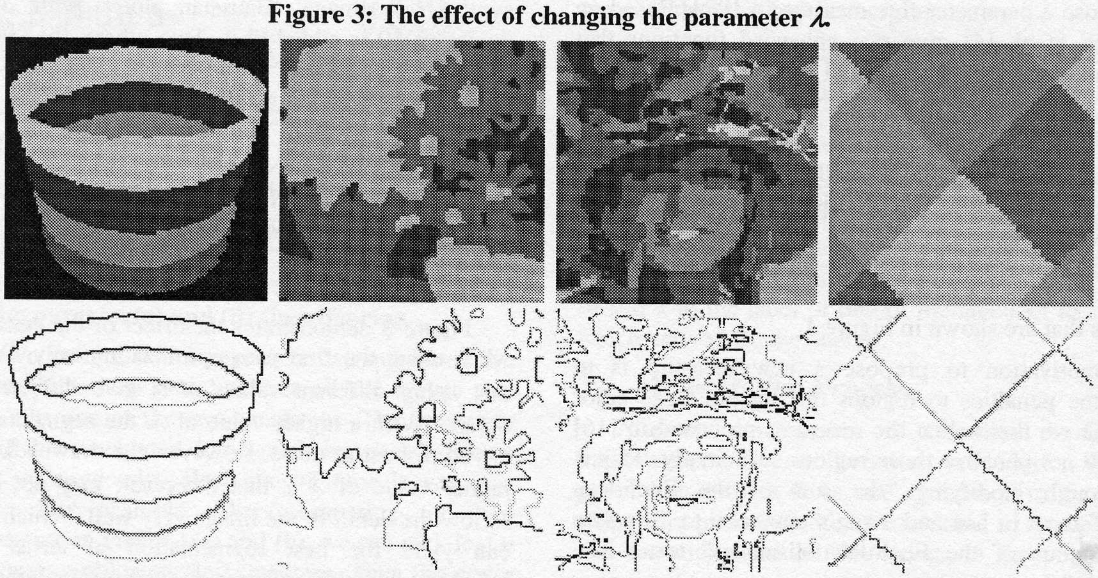


Figure 4: The final results and their contours obtained from the two-pass process.

Table 1 lists evaluation values of the segmentation results of image "flower" under different criteria. It shows that the criteria proposed in [16] and [4] favor the segmentation result obtained when λ=0.16. Nevertheless, our new criterion favors the segmentation result obtained when λ=0.04, which appears to preserve more details.

Table 1: Evaluation values for the three segmentation results of image "flower" under different measures.

	λ=0.01	λ=0.04	λ=0.16
Liu and Yang's measure (F)	78.277	61.1879	49.8886
Borsotti et al.'s measure (F')	29895.8	4774.41	1932.18
Borsotti et al.'s measure (Q)	5095.18	4013.69	3428.35
Ours measure	2594.92	2243.73	2894.77

Figure 4 shows the final result obtained through the two-pass optimization process using our measure as the evaluation criterion. Since the segmentation results generated from the first pass are optimized using the graft crossover operator, the result is even better than the best solution available from the first pass. This argument is supported by the experiment result. Indeed, as shown in Table 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.

Table 2: Evaluation values for the first pass results and the second pass result.

	Statistical information of the first pass results			Optimiza-tion result	Improve-ment
	Best	Worst	Mean		
cylinder	6842.6	29685.4	11641.2	6076.42	11.20 %
flower	2208.4	3036.2	2610.4	2051.77	7.09 %
girl	998.7	6435.7	3128.4	936.05	6.27 %
texture	1207.6	3407.9	1554.6	1202.42	0.43 %

Experiments are also made to compare the segmentation results of our approach with that of the Multiresolution Color Image Segmentation approach [16]. Two of the test images, "cylinder" and "girl", are also used by Liu and Yang. For comparison reason, the F measure as described in [16] is used as the evaluation criterion in the second pass. This also demonstrates the flexibility of this approach in using different evaluation criteria.

The measures for results obtained from the two approaches are listed in Table 3 and Table 4, where the first three columns are measures reported by Liu and Yang, and the fourth column is the results using our new measure (the corresponding images are not shown in this paper). From the tables, we notice that our approach can find the segmentation that has a lower F measure in both cases. In addition, for the "cylinder" image, the best segmentation obtained using the Multiresolution Color Image Segmentation has 10 regions and our segmentation result has 8 regions, where visual judgment indicates that the ideal segmentation should have 7 regions.

Table 3: Measures comparison for image "cylinder."

Measure	Result (a)	Result (b)	Result (c)	Our results
Average color error	67.5	69.9	72.6	66.1469
# of regions	14	10	14	8
F	96.6	87.1	110.6	71.5066

Table 4: Measures comparison for image "girl."

Measure	Result (a)	Result (b)	Result (c)	Our results
Average color error	31.8	35.5	42.1	36.3449
# of regions	2032	145	89	118
F	103.0	48.7	53.8	44.8834

6 Conclusions

In this paper, we introduce a new neighborhood-based algorithm for image segmentation. The genetic 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 penalty term λ, this algorithm can give a set of coarse to fine segmentation results. In the second pass, which is optional, the segmentation results obtained in the first pass with different values of λ are used as the initial population and are optimized with respect to the given evaluation criterion.

The quadtree structure is used to implement the multiresolution framework, which enables the use of different strategies at different resolution levels and accelerates the computation. To apply the GA under the multiresolution framework, a new encoding mechanism for the quadtree structure is proposed. New crossover and mutation operators, namely graft crossover, splitting mutation, merging mutation, and alteration mutation, which are suitable for the quadtree structure, are introduced.

Our new proposed measure can be used to determine the best segmentation result. It gives more penalties to regions that have a large color error comparing with measures proposed in [16] and [4]. Therefore, it favors segmentation results that preserve the detail of the original images, and prohibits noisy segmentation results as well.

Normally, the computational costs of genetic-based algorithms are very high. For example, Bhandarkar and Zhang [2] report that their genetic-based approach needs 700-850 minutes to segment a grayscale image (they did not mention the image size and the computer type). If the value of λ is given, which means that only the first pass is needed, 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 128×128 on our Pentium III 733MHz computer with 256MB RAM running Windows 2000, our algorithm converges in less than 10 seconds.

If the user wants to search for a good segmentation in terms of a given criterion instead of specifying the value of λ, our approach will use 30 different segmentations obtained using different values of λ 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.

Acknowledgement

The authors would like to thank financial support provided by NSERC and the University of Saskatchewan. The comments by the reviewers helped to improve the paper, and were much appreciated.

References

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] M. Celenk, "A Color Clustering Technique for Image Segmentation," *CVGIP*, Vol. 52, No. 2, pp. 145-170, November, 1990.

[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] 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.

[9] D. E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, Reading, MA, USA, 1989.

[10] R. M. Haralick and L. G. Shapiro, "Image Segmentation Techniques," *CVGIP*, Vol. 29, No. 1, pp. 100-132, January, 1985.

[11] G. Healey, "Segmenting Images Using Normalized Color," *IEEE Trans. on Systems, Man and Cybernetics*, Vol. 22, No. 1, pp. 64-73, January-February, 1992.

[12] J. H. Holland, Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor, MI, USA, 1975.

[13] C. L. Huang, T. Y. Cheng, and C. C. Chen, "Color Images' Segmentation Using Scale Space Filter and Markov Random Field," *Pattern Recognition*, Vol. 25, No. 10, pp. 1217-1229, October, 1992.

[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] J. Liu and Y. H. Yang, "Multiresolution Color Image Segmentation," *IEEE Trans. on PAMI*, Vol. 16, No. 7, pp. 689-700, July, 1994.

[17] N. R. Pal and S. K. Pal, "A Review on Image Segmentation Techniques," *Pattern Recognition*, Vol. 26, No. 9, pp. 1277-1294, September, 1993.

[18] 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.

[19] M. Srinivas and L. M. Patnaik, "Genetic Algorithms: A Survey," *Computer*, Vol. 27, No. 6, pp. 17-26, June, 1994.

[20] D. C. Tseng and C. C. Lai, "A Genetic Algorithm for MRF-Based Segmentation of Multi-Spectral Textured Images," *Pattern Recognition Letters*, Vol. 20, No. 14, pp. 1499-1510, 1999.

[21] A. P. Witkin, "Scale-Space Filtering," In: *Proceedings of the 8th International Joint Conference on Artificial Intelligence Vol. 2*, pp. 1019-1022, Karlsruhe, West Germany, August 8-12, 1983.

[22] Y. J. Zhang, "Survey on Evaluation Methods for Image Segmentation," *Pattern Recognition*, Vol. 29, No. 8, pp. 1335-1346, August, 1996.