An improved scheme for minimum cross entropy threshold selection based on genetic algorithm

Kezong Tang\textsuperscript{a,}\textsuperscript{*}, Xiaojing Yuan\textsuperscript{b}, Tingkai Sun\textsuperscript{a}, Jingyu Yang\textsuperscript{a}, Shang Gao\textsuperscript{c}

\textsuperscript{a}School of Computer Science and Engineering, Jiangsu University of Science and Technology, Nanjing Jiangsu 210094, China
\textsuperscript{b}Engineering Technology Department, College of Technology, University of Houston, Houston Texas 77204, United States
\textsuperscript{c}School of Computer Science and Engineering, Jiangsu University of Science and Technology, Zhenjiang Jiangsu 212003, China

\textbf{A R T I C L E   I N F O}

Article history:
Received 18 July 2010
Received in revised form 18 February 2011
Accepted 20 February 2011
Available online 24 February 2011

Keywords:
Image segmentation
Minimum cross entropy
Thresholding
Recursive programming
Genetic algorithms

\textbf{A B S T R A C T}

Image segmentation is one of the most critical tasks in image analysis. Thresholding is definitely one of the most popular segmentation approaches. Among thresholding methods, minimum cross entropy thresholding (MCET) has been widely adopted for its simplicity and the measurement accuracy of the threshold. Although MCET is efficient in the case of bilevel thresholding, it encounters expensive computation when involving multilevel thresholding for exhaustive search on multiple thresholds. In this paper, an improved scheme based on genetic algorithm is presented for fastening threshold selection in multilevel MCET. This scheme uses a recursive programming technique to reduce computational complexity of objective function in multilevel MCET. Then, a genetic algorithm is proposed to search several near-optimal multilevel thresholds. Empirically, the multiple thresholds obtained by our scheme are very close to the optimal ones via exhaustive search. The proposed method was evaluated on various types of images, and the experimental results show the efficiency and the feasibility of the proposed method on the real images.

\textcopyright 2011 Elsevier B.V. All rights reserved.

1. Introduction

Image segmentation, a central issue of computer vision, is a fundamental pre-processing step for many image analysis tasks such as feature extraction, model matching and object recognition. Its goal is to divide an image into meaningful regions that are homogeneous according to some characteristic such as gray value, edge information and texture structure. In the past years, many segmentation methods on real images have been presented [1–7]. These methods include region-based methods, threshold-based methods and so on. Region-based methods employ either region growing or region splitting and merging to segment the image. Then, the application-dependent homogeneity criterion is applied and may be dynamic variation within a given image. On the other hand, threshold-based method is a common segmentation scheme, which can be regarded as the pixel classification. A feature value such as gray level is associated with each pixel. The value is compared to the threshold to classify a pixel into an object or the background. Generally image segmentation techniques can be classified as bilevel thresholding and multilevel thresholding in accordance with the number of thresholds. The former selects only one threshold which classifies the pixels into two classes, while the latter determines multiple thresholds which divide the pixels into several classes. In some cases, the threshold can be obtained automatically on the basis of the histogram of images for some simple images without noise or with low noise. For example, the midpoint or a valley between two distinct peaks is often used. However, this method is undesirable for two reasons. Firstly, the nature of the image histogram may be unimodal or multimodal, such that selection of the appropriate threshold value is less obvious. Under these circumstances, the threshold selection is unrepeatable due to the subjectivity inherent in the decision. Secondly, the fundamental limitation of this method is that it fails to exploit all the useful information provided by the image. Therefore, the auto selection of robust optimum threshold has remained a challenge for complex image segmentation.

Over the years many thresholding techniques have been proposed. Comprehensive surveys can be found in recently published literatures [1–3]. Among them, one of the most efficient techniques for image segmentation is entropy-based approaches such as Shannon entropy [4], Renyi entropy [5], Tsallis entropy [6] and cross entropy [7]. Of these approaches, cross entropy proposed by Kullback [23] has attracted considerable attention as a global optimization technique for complex optimization problems [8–10]. It is formulated as an information theoretic distance between the two probability distributions on the same set. The optimal threshold is obtained by minimizing the cross entropy between the object and
background. Now this method has been widely applied to image processing. Horng [9] proposed a new multilevel MCET algorithm based on the technology of the honey bee mating optimization. Zimmer et al. [11] presented a multivariate extension of MCET in which the segmented variable (gray level) is replaced by a weighted combination of several image parameters. Pang et al. [12] developed an image segmentation method on local minimum cross entropy that is combined with the quad-tree model. Ma et al. [13] put forward a kind of novel algorithm of image segmentation setting on cycle iteration automatically by incorporating the minimum cross entropy criterion into an improved segmentation mechanism of Pulse Coupled neural network. Gao et al. [14] applied the cross entropy measure of information theoretic to evaluating the divergence between the distribution of object and background. Vaz et al. [15] used a local minimum distance cross entropy principle to reconstruct the noisy image signal. Brink and Pendock [16] deliberately showed the relationship between the MCET technique and other methods. Despite the large amount of researches have been done in the field of image segmentation by using cross entropy, it is worth noting that the computational time needed using MCET grows exponentially with the increasing of the number of required thresholds. To overcome this deficiency, the several algorithms have so far been proposed in literatures [17–19] to reduce computational time of searching thresholds. Among them, the deployment of meta-heuristic computing has been flourishing in recent years. Many different population-based stochastic optimization methods such as genetic algorithms (GA), evolutionary strategies (ES) and particle swarm optimization (PSO) have been successfully developed and applied to solve complex optimization problems in various fields such as image processing [20], pattern recognition [21], and machine learning [22]. It has been shown that these biologically inspired heuristics offer better performances than the classical optimization approaches in complex optimization problems due to their domain independent nature and the capability of finding optimal or near optimal solutions in a large search space. Therefore, the attempt to search for the optimal threshold by using the population-based heuristic methods may be considered as a promising approach in image segmentation. Encouraged by these successful applications, we further investigate the feasibility of solving image segmentation by using GA.

The aim of this paper is focused on multilevel thresholding by using the minimum cross entropy criterion. It has been theoretically proven that GAs provide robust search even if the search space is not continuous. Since GAs perform parallel search, it is feasible to apply GAs to accelerating the optimal threshold selection in the case of multilevel thresholding. On the other hand, the main difference between GAs and the other classical search algorithms is that GAs can adaptively adjust their search speed while avoiding entrapment in local optimal. In the existing algorithms, GAs have received considerable attention regarding their potential to solve global optimization problems. Therefore, the application of GAs for solving certain problems of image processing (which need optimization of computation requirements, and robust, fast and close approximate solution) appears to be appropriate and natural. In this paper, our method can be summarized as follows: first of all, image segmentation is treated as an optimization problem to be solved by the recursive method; then a recursive programming technique is presented to store the results of proceeding tries as the basis for the computation of succeeding ones; finally, based on this recursive programming technique, a genetic algorithm is presented for searching the optimal MCET thresholds to optimize objective function. The proposed method is tested on several gray-level images, and the experimental results indicate that our method is efficient and feasible on real images.

The remainder of this paper is organized as follows. Section 2 introduces the mathematical model used to describe the concepts of minimum cross entropy. Section 3 describes the proposed method in details. The experimental results and analysis are given in Section 4. Conclusion and the future work are drawn in Section 6.

2. Minimum cross entropy

Entropy is a measure of the amount of information produced by a random process, or a measure of uncertainty in a random process. A larger value of entropy corresponds to more information (uncertainty) in the process. The entropy $I(P) = -\sum p(w) \log p(w)$ of a system (where the sum is taken over all elementary events $w$, using the convention $\log 0 = 0$) with the probability density $p$ first appeared as a physical quantity in statistical mechanics and was later interpreted by Shannon as an information-theoretic measure of the uncertainty inherent to $p$. In the context of prior and posterior probabilities, Kullback [23] introduced the concept of the cross entropy. It is a positive, additive, unsymmetrical, and convex function of probabilities. Cross entropy is also called directed divergence (or probabilistic distance) for it lacks symmetry. The cross entropy distance, subject to data consistency, between a posteriori probability distribution $q(x)$ and a priori distribution $p(x)$ is defined as

$$I_{p,q} = \int q(x) \log \frac{q(x)}{p(x)} dx$$

(1)

Eq. (1) states that the total amount of information produced by a process equals the sum of the amount of information gained by the posterior (current) density $q$ and the information already acquired by $p$. The priors must be strictly positive, i.e. $p > 0$.

For discrete (digital) probability distributions we substitute summation for the integration above. Suppose $P =\{p_1, p_2, \ldots, p_n\}$ and $Q =\{q_1, q_2, \ldots, q_n\}$ be two probability distributions, then the cross entropy $I_{p|q}$ is defined by

$$I_{p|q} = \sum_{i=1}^{n} p_i \log \frac{p_i}{q_i}$$

(2)

where $\sum_{i=1}^{n} p_i =\sum_{i=1}^{n} q_i = 1$, and $I_{p|q} \geq 0$. $I_{p|q}$ measures the statistical difference in uncertainty about the outcome of the experiment in which observation data is transmitted from $Q$ to $P$. It is well known that $I_{p|q}$ is zero when $p_i = q_i$. That means that $I_{p|q}$ would achieve less value responding to less event difference. $Q$ and $P$ in Eq. (2) are called the prior distribution and the posterior distribution respectively. Since $I_{p|q}$ is not symmetric, i.e. $I_{p|q} \neq I_{q|p}$, it is not a metric distance measure. Symmetry can, however, be imposed by adding these together, giving a metric distance measure:

$$I_{SPQ} = I_{p|q} + I_{q|p}$$

(3)

Several other mathematical properties of entropy and cross-entropy measures were discussed by Kapur and Kesavan [30]. In applications such as spectrum analysis, pattern classification and image segmentation the prior distribution, $P$, represents either such knowledge as we may have regarding the “correct” solution, desiderata such as “smoothness” or the original grey-level image itself (in the case of segmentation). The posterior distribution, $Q$, is the result of our processing of the data. Simply stated, the principle of minimum cross entropy, as with any discrepancy measure, is that we wish to find the solution closest to the desired or expected result by minimizing the difference (cross entropy) between them.

In image segmentation, the minimum cross entropy thresholding method selects several thresholds by minimizing the cross entropy between the original image and the resulting image. We assume a histogram $h(t)$ be defined on the gray level range $[1, L]$. $t$ is an assumed threshold, namely, $t$ partitions an image into two
regions (the object and the background). The cross entropy is then calculated by
\[ I(t) = \sum_{i=1}^{l} i h(i) \log \left( \frac{i}{u(t, 1)} \right) + \sum_{i=1}^{l} h(i) \log \left( \frac{i}{u(t, L+1)} \right), \]  
(4)

where \( u(a, b) = \sum_{i=a}^{b} i h(i) / \sum_{i=1}^{l} h(i) \). We determine optimal threshold value \( t^* \) by minimizing the cross entropy, i.e., \( t^* = \arg \min_t (I(t)) \).

The computational complexity for determining \( t^* \) is \( O(L^2) \). However, it could be time-consuming under that multilevel thresholding scenario. For the n-thresholding problem, it requires \( O(L^{n+1}) \).

3. Proposed method

3.1. Recursive programming

To determine the threshold value \( t \), the Eq. (4) can be rewritten as follows:
\[ I(t) = \sum_{i=1}^{l} i h(i) \log(i) - \sum_{i=1}^{l-1} i h(i) \log \left( \frac{\sum_{j=i}^{l} j h(j)}{\sum_{j=1}^{l} h(j)} \right) - \sum_{i=1}^{l} h(i) \log \left( \frac{\sum_{j=i}^{l} j h(j)}{\sum_{j=1}^{l} h(j)} \right), \]  
(5)

where \( \sum_{j=i}^{l} j h(j) / \sum_{j=1}^{l} h(j) \) is constant for a given image, the objective function can be redefined by
\[ \eta(t) = \sum_{i=1}^{l} i h(i) \log(i) - \sum_{i=1}^{l-1} i h(i) \log \left( \frac{\sum_{j=i}^{l} j h(j)}{\sum_{j=1}^{l} h(j)} \right) - \sum_{i=1}^{l} h(i) \log \left( \frac{\sum_{j=i}^{l} j h(j)}{\sum_{j=1}^{l} h(j)} \right), \]  
(6)

where a necessary condition for the \( \eta(t) \) to obtain the optimal value is given by setting the derivative of \( \eta(t) \) to zero. The derivative of \( \eta(t) \) is
\[ \eta'(t) = h(t) \left( \log \frac{u(t, 1)}{u(t, L+1)} - (u(1) - u(t, L+1)) \right). \]  
(7)

If \( \eta'(t) \) is zero, it means that either the first factor \( h(t) \) is zero or the second factor in Eq. (7) is zero. Clearly \( h(t) = 0 \); is satisfied only by those thresholds of the gray value which the image does not contain, and these solutions do not have a direct influence on the thresholding problem. Thus, determine the optimal threshold \( t' \) by setting the second term to zero, we get
\[ t' = \frac{u(t, L+1) - u(1)}{\log(u(t, L+1)) - \log(u(t, 1))}. \]  
(8)

If we set some equivalent forms for the zero-moment and first-moment on partial rang of the image histogram, i.e. \( m^n(a, b) = \sum_{i=a}^{b} i h(i) \), \( m^n(a, b) = \sum_{i=2}^{b} i h(i) \), the objective function (8) can be rewritten as follows:
\[ t' = \frac{m^n(L+1)}{\log(m^n(L+1))} - \frac{m^n(t)}{\log(m^n(t))} = f(t). \]  
(9)

Suppose an initial value \( t_0 \in [1, L] \), and let \( t_{i+1} = t_i + 1 \) or \( t_{i+1} = t_i - 1 \). \( \varepsilon \) is a given small constant. The condition of convergence is that when \( |f(t_i) - f(t_{i+1})| < \varepsilon \), then \( t_i \) is the optimal threshold sought. The calculation of the optimal threshold \( t_i \) probably involves the evaluation of all possible thresholds from the range \([1, L]\). Obviously, the optimal MCET threshold derived by exhaustive method is usually time-consuming for not only multilevel thresholding but also bilevel thresholding. Here we propose a recursive programming technique for expediting the computing process of the function (9) with different trial thresholds. Suppose that the initial trial threshold is \( t \) and the corresponding objective value \( f(t) \) has been calculated. By computing the intermediate moments \( m^n(t, 1), m^n(t, L+1) \), the other moments can be calculated recursively by the following equation:
\[ m^n(t, L+1) = m^n(t, t+1) + h(t), \quad m^n(t, t+1) = m^n(t, t) + th(t), \]
\[ m^n(t, L+1) = m^n(t, L) - h(t), \quad m^n(t, L) = m^n(t, L+1) - th(t) \]  
(10)

Hence, after getting \( f(t) \) of an initial threshold \( t \), we calculate the corresponding objection value \( f(t+1) \) for next threshold \( t+1 \), then its computational time is only a constant. Obviously, computational complexity for the optimal threshold \( t^* \) is reduced to some extent. It has been verified in our experiments on various types of images.

The recursive programming technique is easily extended to multilevel thresholding case. Assume that it is required to select \( n \) thresholds denoted by \( t_1, t_2, \ldots, t_n \). For the convenience of illustration, we add two dummy thresholds \( t_0 = 1 \) and \( t_{n+1} = L+1 \), and \( t_0 < t_1 < \cdots < t_n < t_{n+1} \). The objective function (6) then becomes
\[ \eta(t_1, t_2, \ldots, t_n) = -\sum_{i=1}^{n} m^n(t_i, t_{i+1}) \frac{m^n(t_{i+1}, t_i)}{m^n(t_{i+1}, t_i)} \]  
(11)

The derivative of \( \eta(t_1, t_2, \ldots, t_n) \) is
\[ \frac{d\eta}{dt_i} = h(t_i) \left( \log \frac{u(t_i, t_{i+1}) - (u(t_i, t_{i+1}) - u(t_i, t_{i+1}))}{u(t_i, t_{i+1})} \right). \]  
(12)

Similarly, where we set the second factor to zero, then we get
\[ t'_i = \frac{u(t_i, t_{i+1}) - u(t_i, t_{i+1})}{\log(u(t_i, t_{i+1}) - \log(u(t_i, t_{i+1}))} = f(t_i) \]  
(13)

where \( t_{i+1}, t_i \) are known constants. Given an initial guess \( t_{0,i} \in [t_{i+1}, t_i] \), and \( t_{n+1} = t_0 + 1 \) or \( t_{n+1} = t_n - 1 \), \( t_{n+1} \in [t_{i+1}, t_i] \). The condition of convergence is satisfied when \( |f(t_{i+1}) - f(t_i)| < \varepsilon \), then \( t_i \) is the ith component of \( n \)-dimensional MCET vector. To reduce the computational time, we still adopt the same recursive programming technique with the case of bilevel thresholding. Suppose the intermediate moments \( m^n(t_{i-1}, t_i), m^n(t_i, t_{i+1}), m^n(t_{i+1}, t_{i+1}) \) and \( m^n(t_{i+1}, t_{i+1}) \) have already been calculated. Then, the other moments can also be calculated recursively by the following equation:
\[ m^n(t_{i-1}, t_{i+1}) = m^n(t_{i-1}, t_i) + h(t_i), \quad m^n(t_i, t_{i+1} + 1) = m^n(t_i, t_{i+1}) + t_i h(t_{i+1}), \]
\[ m^n(t_{i+1}, t_{i+1} + 1) = m^n(t_{i+1}, t_{i+1}) + t_i h(t_{i+1}). \]  
(14)

Therefore, the solution to Eq. (13) always exist and computational complexity is reduced by \( O(nL^n) \). In spite of reducing the computational complexity to some extent, \( O(nL^n) \) is still computationally expensive when \( n > 3 \). In this paper, we proposed a genetic algorithm based on the recursive programming technique for solving the optimal threshold \( t'_i \) efficiently and effectively.

![Fig. 1. Genetic algorithm flow chart.](image-url)
3.2. Description of GAs

Genetic algorithm proposed by Holland [24] has attracted considerable attention as a global method for complex function optimization since Jong considered GAs in a function optimization setting [25]. Since 1962, complex optimization problems in many domains have been solved by using GAs. GAs perform search in complex, large and multimodal landscapes, and provide near-optimal solutions for objective or fitness function of an optimization problem. Encouraged by their successful applications, recently, GAs have found a variety of applications such as neural network [26], combinatorial optimization [27], and adaptive control [28]. The convergence and parameterization setting of the GAs have also been discussed [29]. The standard GA can be expressed as follows.

Fig. 2. The tested images and corresponding histograms: (a) Coin, (b) Cell, (c) Lena, (d) Pepper.
As a heuristic population-based method, GA is really like a “black box”, completely independent from the characteristic of the problem. Fig. 1 presents the classical GA flow chart. An initial population of individuals is generated randomly, which represents different points in the search space. Each of these individuals is evaluated in terms of a certain “fitness function” that can “guide” GA to the desired region of the search space. The parameters of the search space are encoded in the form of strings (called chromosomes in genetic language). Holland’s three genetic operators (selection, crossover and mutation) are the main components to improve the GA’s behavior. Selection is the process that mimics the “survival of the fittest” principle in the biological theory of evolution. Firstly, the selection operator assures that individuals are copied to the next generation with a probability associated to their fitness values. Although selection is implemented in GA as a policy for determining the best candidate individuals that will be presented in the next generation with a higher probability, it does not search the space further, because it just copies the previous candidate individuals. The search results from the creation of new individuals from old ones. Secondly, the crossover operator is implemented in GA by exchanging chromosome segments between two randomly selected chromosomes. Crossover process provides a mechanism to allow new chromosomes to inherit the properties from old ones. Thirdly, mutation is a random perturbation to one or more genes in the chromosomes during evolutionary process. The purpose of the mutation operator is to provide a mechanism to avoid local optima by exploring the new regions of the search space, which selection and crossover could not fully guarantee. The searching process terminates when the pre-defined criterion is satisfied.

3.3. The proposed algorithm

In proposed algorithm, we adopt the real-coded GA. Let the gray levels of a given image range over [1, L], and n thresholds are required. Each feasible solution can be coded as a string \( x \), namely, \( x = (x_1, x_2, \ldots, x_n) \). Obviously, \( x_i \neq x_j \) for \( i \neq j \), and \( x_i \) represents the ith component value in n-dimensional MCET vector. The discrete crossover operator is most often used in GA. In each generation, a new individual is randomly decided from one of the two pre-selected parents. If two parents are \( x = (x_1, x_2, \ldots, x_n) \) and \( y = (y_1, y_2, \ldots, y_n) \), respectively, then we generate offspring \( z = (z_1, z_2, \ldots, z_n) \), where \( z_i = \begin{cases} x_i & \text{if } \text{rand} \leq P_c \\ y_i & \text{otherwise} \end{cases} \). Selection and crossover could be performed recursively in the evolutionary process. The purpose of the mutation operator is to introduce new segments between two randomly selected chromosomes. A random perturbation to one or more genes in the chromosomes is obtained. The evaluation of \( f(\mathbf{z}) \) is computed by Eq. (13). Where we add two dummy thresholds \( z_0 = 1 \) and \( z_{n+1} = L + 1 \), and suppose \( z_0 < z_1 < \cdots < z_n < z_{n+1} \). Let \( t_0 = z_i \) and \( t_{n+1} = z_i + 1 \) or \( t_{n+1} = t_i - 1 \), \( t_i \) are equal to \( z_i \). The condition of convergence is satisfied when \( |f(t_i) - f(t_j)| < \epsilon \), then \( t_i \) is the ith component of \( n \)-dimensional MCET vector. We carry out a statistical analysis of the number of threshold components derived. More number of components derived means the fitness of the variable \( z \) is better. The user is required to specify the population size \( P \), crossover probability \( P_c \), and the maximum number of generations \( G \). The proposed algorithm with the real-coded is detailed as follows:

Step 1. Calculate \( m_i^P(a, b) \) and \( m_i^L(a, b) \) for \( 1 < a < b < L \) using the recursive programming technique. Where \( a, b \) are stochastic integer numbers.

Step 2. Generate \( P \) individuals randomly.

Step 3. Evaluate the fitness of each individual by using Eq. (13) on the pre-computed moments.

Step 4. Collect the best \( u \) individuals from population in selection.

Step 5. Generate \( u \times v \) offspring by performing crossover and mutation operators. Where \( v \) is the ratio of offspring and parents.

Step 6. Compare these offspring with the old population. If an individual \( s^* \) has a better fitness than an individual \( s \) in old pop-

![Fig. 3. The optimal bilevel MCET thresholds and the segmented images.](image-url)
ulation, then substitute $s'$ for $s$. Or if an individual $s'$ have the same fitness as the individual in old population, that randomly select an individual between $s'$ and $s$ into the next generation.

Step 7. Go to Step 3 if the termination criteria are not met.

4. Experimental results and comparative performances

In this section, we implement the proposed GA-based algorithm in Matlab language with a 2.27 GHz Intel Core i5 CPU. Four images named “Coin”, “Cell”, “Lena”, and “Pepper”, with image size of $256 \times 256$, $265 \times 272$, $512 \times 512$, $512 \times 512$, respectively, are used for conducting our experiments. In our experiments, the proposed algorithm is called GA-based method. The initial population size $P = 30$. The best $u$ individuals and crossover rate $P_c$ are set to 10 and 0.5, respectively. The ratio of offspring and parents, $v$ is 2. The algorithms terminates when the number of generations are 100.

In order to analyze the segmentation technique by the GA-based method, we simulate different histograms describing the “object” and “background” by different peaks. Fig. 2 shows tested images and corresponding histograms. It can be seen that both image histograms on Coin and Cell are of well defined Gaussian peaks. To segment the two images, we select only one threshold which classifies the pixels into two classes, and can obtain better result of segmentation. In the bilevel thresholding scenario, we implement the GA-based method with recursive programming technique for images Coin and Cell. It can be seen in Fig. 3 that the proposed recursive programming technique is feasible and the objects are well segmented from background. However, for images Lena and Pepper, the image histograms are more complex. To achieve better results of segmentation, a sophisticated segmentation based on multilevel thresholding is needed.

For the multilevel thresholding on images Lena and Pepper, the multiple MCET thresholds are obtained from both of the GA-based method and exhaustive search method. To measure the difference of performances between the GA-based method and exhaustive search method, we evaluate the quality of the thresholding images by the uniformity measure which is broadly used in the literature [22,23].

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>C Thresholds</th>
<th>CPU time</th>
<th>Uniformity</th>
</tr>
</thead>
<tbody>
<tr>
<td>GA-based method</td>
<td>1 107</td>
<td>0.0135</td>
<td>0.998637</td>
</tr>
<tr>
<td></td>
<td>2 93, 147</td>
<td>0.0166</td>
<td>0.996083</td>
</tr>
<tr>
<td></td>
<td>3 85, 126, 170</td>
<td>0.0201</td>
<td>0.994096</td>
</tr>
<tr>
<td></td>
<td>4 81, 115, 148, 179</td>
<td>0.0269</td>
<td>0.993859</td>
</tr>
<tr>
<td>Exhaustive search method</td>
<td>1 107</td>
<td>0.1035</td>
<td>0.998637</td>
</tr>
<tr>
<td></td>
<td>2 93, 147</td>
<td>0.2107</td>
<td>0.996083</td>
</tr>
<tr>
<td></td>
<td>3 85, 127, 169</td>
<td>15.7051</td>
<td>0.994045</td>
</tr>
<tr>
<td></td>
<td>4 81, 116, 146, 179</td>
<td>850.2065</td>
<td>0.993971</td>
</tr>
</tbody>
</table>

Table 1: The experimental results by using GA-based method and exhaustive search method for Lena.

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>C Thresholds</th>
<th>CPU time</th>
<th>Uniformity</th>
</tr>
</thead>
<tbody>
<tr>
<td>GA-based method</td>
<td>1 108</td>
<td>0.0125</td>
<td>0.998259</td>
</tr>
<tr>
<td></td>
<td>2 68, 134</td>
<td>0.1035</td>
<td>0.997359</td>
</tr>
<tr>
<td></td>
<td>3 64, 118, 165</td>
<td>0.2107</td>
<td>0.995215</td>
</tr>
<tr>
<td></td>
<td>4 51, 86, 125, 169</td>
<td>15.7051</td>
<td>0.995277</td>
</tr>
<tr>
<td>Exhaustive search method</td>
<td>1 108</td>
<td>0.1317</td>
<td>0.994383</td>
</tr>
<tr>
<td></td>
<td>2 68, 134</td>
<td>0.2864</td>
<td>0.9864</td>
</tr>
<tr>
<td></td>
<td>3 64, 117, 164</td>
<td>19.2183</td>
<td>0.995215</td>
</tr>
<tr>
<td></td>
<td>4 51, 87, 126, 167</td>
<td>901.1162</td>
<td>0.994749</td>
</tr>
</tbody>
</table>

Table 2: The experimental results by GA-based method and exhaustive search method for Pepper.

Fig. 4. The derived thresholded images by using GA-based method and exhaustive search method for Lena. (a) 3-Level thresholded image by using the GA-based method. (b) 3-Level thresholded image by using the exhaustive search method. (c) 4-Level thresholded image by using the GA-based method. (d) 4-Level thresholded image by using the exhaustive search method.

Fig. 5. The derived thresholded images by using GA-based method and exhaustive search method for Pepper. (a) 3-Level thresholded image by using the GA-based method. (b) 3-Level thresholded image by using the exhaustive search method. (c) 4-Level thresholded image by using the GA-based method. (d) 4-Level thresholded image by using the exhaustive search method.
Future work will focus on comparing the proposed method with other segmentation methods. In order to obtain better segmentation results, we will apply other meta-heuristics to the proposed method for further testing segmentation effect on real images. We are in intensive work for this research.

Acknowledgement

This paper is supported by the National Natural Science Foundation of China (Nos. 60803049 and 60632050).

References


5. Conclusion

In thresholding techniques of image segmentation, we wish the used method can not only obtain better segmentation effect, but also find the exact optimal threshold within reasonable time. In this paper, an improved scheme based on genetic algorithm was presented for threshold selection in multilevel MCET. To solve the segmentation problem, we introduce a recursive programming technique, and incorporate it into a genetic algorithm. Then we search optimal thresholds (including the case of bilevel thresholding and multilevel thresholding) by using the genetic algorithm. The testing results on various types of images show that the proposed method is efficient and feasible for real images.