

Automatic multilevel thresholding for image segmentation using stratified sampling and Tabu Search

Yunzhi Jiang · Pohsiang Tsai · Zhifeng Hao · Longbing Cao

Published online: 24 August 2014
© Springer-Verlag Berlin Heidelberg 2014

Abstract Image segmentation techniques have been widely applied in many fields such as pattern recognition and feature extraction. For the primate visual attention model, the perceptual organization is an important process to automatically extract the desirable features. In this article, we propose a new method called an automatic multilevel thresholding algorithm using the stratified sampling and Tabu Search (AMTSSTS) by imitating the primate visual perceptual behaviors. In the AMTSSTS algorithm, a gray image is treated as a population with the gray values of pixels as the individuals. First, the image is evenly divided into several strata (blocks), and a sample is drawn from each stratum. Second, a Tabu Search-based optimization is applied to each sample to maximize the ratio between mean and variance for each sample. The threshold number and threshold values are preliminarily determined based on the optimized sam-

ples, and are further optimized by a deterministic method which includes a new local criterion function with property of local continuity of an image. Results of extensive simulations on Berkeley datasets indicate that AMTSSTS can obtain more effective, efficient and smooth segmentation, and can be applied to complex and real-time environments.

Keywords Image segmentation · Multilevel thresholding · Intelligent optimization algorithms · Stratified sampling · Tabu Search

1 Introduction

Image segmentation is an essential and important operation for feature extraction, pattern recognition and classification. It is the process of assigning a label to every pixel in an image. Pixels with the same annotated label constitute a region, and the pixels inside this region share similar visual characteristics. Each region is homogeneous and connected. The union of any two spatially adjacent regions is not homogenous.

For intensity images, there are four popular approaches: threshold techniques, edge-based methods, region-based techniques, and connectivity-preserving relaxation methods. Of these, the image thresholding methods are the most widely used in image segmentation. The thresholding techniques involve bi-level thresholding and multilevel thresholding. Bi-level thresholding techniques classify the pixels into two groups; the first group includes the pixel values that are above a certain threshold value and the second group includes the pixel values that are below the threshold value. Multilevel thresholding techniques divide the pixels into several classes; that is, for the pixels that can be grouped as the same class, their pixel values should be within a certain range of threshold values.

Communicated by V. Loia.

Y. Jiang
School of Software, Jiangxi Agricultural University,
Nanchang, China
e-mail: jyzscut@gmail.com

Y. Jiang · Z. Hao
School of Computer Science and Engineering, South China
University of Technology, Guangzhou, China
e-mail: mazfhao@scut.edu.cn

P. Tsai (✉)
Department of Computer Science and Information Engineering,
National Formosa University, Huwei, Taiwan
e-mail: ptsai@nfu.edu.tw

Y. Jiang · L. Cao
Centre for Quantum Computation and Intelligent System,
Advanced Analytics Institute, University of Technology, Sydney,
Ultimo, Australia
e-mail: lbcao@it.uts.edu.au

Many image thresholding methods can be found in [Ma et al. \(2010\)](#), [Sezgin and Sankur \(2004\)](#). In general, the thresholding methods can be also classified into parametric and nonparametric approaches. In the parametric approaches, the gray-level distribution of each class has a probability density function that is generally assumed to obey a Gaussian distribution and will attempt to find an estimate of the parameters of distribution that will best fit the given histogram data. In the nonparametric approaches, they find the thresholds that separate the gray-level regions of an image in an optimal manner based on discriminating criteria such as the between-class variance, entropy, etc. They are easy to extend to multilevel thresholding; however, the amount of thresholding computation significantly increases with this extension. To overcome this problem, some recent multilevel thresholding techniques based on intelligent optimization algorithms (IOAs) have been proposed and shown promising results. However, they still suffer from many problems. We will discuss these problems in Sect. 2.

The structure of this paper is as follows. Section 2 introduces the related work. Section 3 formulates our problem, and presents the related methods and our proposed algorithm. Section 4 provides rigorous discussion regarding our proposed algorithm from the three different aspects: the optimization method, the sampling method, and algorithmic complexity. Section 5 discusses the experimental results. Finally, Sect. 6 concludes our findings and further work in the future research.

2 Related work

IOAs that were proposed for the multilevel thresholding are divided into several categories, genetic algorithms (GAs) ([Hammouche et al. 2008](#); [Bosco 2001](#); [Lai and Chang 2009](#); [Lin et al. 2010](#); [Yin 1999](#); [Yang et al. 2003](#); [Lai and Tseng 2004](#); [Cao et al. 2008](#)), particle swarm optimization algorithms (PSOs) ([Chander et al. 2011](#); [Yin 2007](#); [Zahara et al. 2005](#); [Gao et al. 2010](#); [Fan and Lin 2007](#)), ant colony optimization algorithms (ACOs) ([Malisia and Tizhoosh 2006](#); [Tao et al. 2008](#)), artificial neural networks (ANNs) ([Yang et al. 2003](#); [Bhattacharyya et al 2011](#)), fuzzy computing algorithms (FCs) ([Chen and Zhang 2004](#); [Li and Staunton 2007](#); [Tan and Isa 2011](#)), simulated annealing algorithms (SAs) ([Hou et al. 2006](#)), and honey bee mating optimization algorithms (HBMOs) ([Hornig 2010](#)).

[Hammouche et al. \(2008\)](#) proposed a fast multilevel thresholding technique based on GA. This technique employed wavelet transform to reduce the length of the histogram, but it was time consuming to choose the appropriate number of thresholds and threshold values by optimizing the automatic thresholding criterion (ATC) proposed by [Yen et al. \(1995\)](#) cited in [Hammouche et al. \(2008\)](#). [Lai and Chang](#)

(2009) proposed a clustering-based approach using a hierarchical evolutionary algorithm (HEA) for automatic medical image segmentation with the help of the particular property of HEA, but this approach initialized the number of regions randomly. [Chander et al. \(2011\)](#) proposed a new variant of PSO for image segmentation, of which an iterative procedure was proposed to obtain initial thresholds, however, its computational time was still somewhat expensive. [Yin \(2007\)](#) developed a recursive programming technique for computing the minimum cross entropy thresholding (MCET) objective function which stores the results of preceding tries as the basis for the computation of succeeding ones; it can reduce the order of magnitude of computing the multilevel thresholds to some extent. [Gao et al. \(2010\)](#) proposed the quantum-behaved PSO method, which employed the cooperative method (CQPSO) to conquer the curse of dimensionality. Although the cooperative method ensured that the search space was searched more thoroughly and the chances of finding better solutions were improved, it increased computational complexity. [Tao et al. \(2008\)](#) proposed a fuzzy entropy method that incorporated the ACO method for the segmentation of infrared images. The ACO method was used to obtain the optimal combination of the fuzzy parameters; however, how to automatically decide the threshold number in these works ([Yin 2007](#); [Zahara et al. 2005](#); [Gao et al. 2010](#); [Fan and Lin 2007](#); [Malisia and Tizhoosh 2006](#); [Tao et al. 2008](#); [Chang and Chung 2001](#); [Bhattacharyya et al 2011](#)) was not mentioned.

[Chen and Zhang \(2004\)](#) proposed two variants of fuzzy C-means (FCM) clustering method with spatial constraints for image segmentation, which were robust to noise and outliers. [Chuang et al. \(2006\)](#) presented a fuzzy C-means algorithm that incorporated spatial information into the membership function for image segmentation, which was also a powerful method for noisy image segmentation. [Li and Staunton \(2007\)](#) proposed a modified fuzzy C-means segmentation method for uneven illumination images, which was based on both pixel intensity and the average intensity of its neighborhood that was within a segment of the biased illumination field. The FCM method has been a popular algorithm in computer vision and pattern recognition due to its clustering validity and simplicity of implementation, but it often encounters two unavoidable difficulties of deciding the threshold number and obtaining the initial thresholds that are properly distributed. In addition, the papers in [Chen and Zhang \(2004\)](#), [Chuang et al. \(2006\)](#), [Li and Staunton \(2007\)](#) did not cover the method of automatically deciding the threshold number, either. Later, [Tan and Isa \(2011\)](#) presented a novel histogram thresholding fuzzy C-means hybrid (HTFCM) approach for color image segmentation, which consists of the histogram thresholding module and the FCM module. The histogram thresholding module was a good solution to overcome the two problems of FCM

that were mentioned above, but it still had higher computational complexity in comparison with the technique that uses the stratified sampling technique in this paper. In Hou et al. (2006), Horng (2010), the authors did not discuss the problem of automatically choosing the threshold number, either.

Although many segmentation methods based on IOAs have been used to reduce the computational time in thresholding, they are limited in the following three aspects. First, the threshold number is predetermined, which implies that users must identify the number of regions beforehand; however, this demand is strict and costly in practice. Second, they need preprocessing to reduce or remove the noise to enhance the convergence rate or to improve the stability using some special techniques (Hammouche et al. 2008; Gao et al. 2010; Chen and Zhang 2004). Third, the IOA is employed to optimize the global criterion function based on between-class variance (Otsu 1979), entropy (Li et al. 1995; Cheng and Lui 1997; Huang and Wang 1995) and minimum bayes error (Kittler and Illingworth 1986; Ye and Danielsson 1988); however, no consideration is given to local continuity of an image.

To rectify these limitations, an automatic multilevel thresholding algorithm (AMTSSTS) using stratified sampling and Tabu Search is proposed. This method is based on the study that higher visual processes of primate visual system appear to select a subset of available sensory information before further processing (Tsotsos et al. 1995; Itti et al. 1998). The fundamental idea of AMTSSTS is “forecasting + optimization”. “Forecasting” corresponds to “select a subset of available sensory information”, and “optimization” corresponds to “further processing”. Hence, AMTSSTS is very different from aforementioned algorithms. In AMTSSTS, the threshold number and threshold values are automatically forecasted by the prediction method combining stratified sampling with Tabu Search, and then they are optimized by an optimization algorithm. In addition, instead of the global criterion function based on between-class variance or entropy method, we did propose a new local criterion function, because each of the pixels in a region is similar with respect to some characteristics or computed properties such as intensity and texture.

3 The proposed method

3.1 Problem formulation

A formal definition of segmentation is given below (1). Let I be an image and H be a homogeneity predicate defined over connected pixel groups. The image I is then partitioned into a set of different regions $\{R_1, R_2, \dots, R_M\}$, which is formulated as follows:

$$\begin{aligned} \bigcup_{i=1}^M R_i &= I \text{ with } R_i \cap R_j = \emptyset, i \neq j. \\ \forall i, H(R_i) &= \text{true}. \\ \forall R_i \text{ and } R_j \text{ adjacent, } H(R_i \cup R_j) &= \text{false}. \end{aligned} \quad (1)$$

In the following subsections, we will describe our proposed algorithm in detail.

3.2 Tabu Search

The Tabu search algorithm (Glover 1989, 1990) proposed by Glover was successfully applied to different combinatorial problems such as planning and scheduling problems (Glover and Laguna 1997). It was an iterative heuristic optimization method based on the local search techniques that started from a feasible solution. This method moved to another solution at each iteration by trying to search unexplored regions of the solutions space, and it also attempted to avoid repetitions with the help of a tabu list. This list is a short-term memory mechanism containing solutions that have been visited in the recent past (less than tt iterations ago, where tt called the tabu tenure is the number of previous solutions to be stored). Note that the Tabu Search algorithm excludes the solutions in the tabu list. Moreover, a variation of a tabu list prohibits solutions that have certain attributes or prevents certain moves. The new solution selected at each iteration is the best one belonging to $N(s)$, the neighborhood of the current solution s . In general, the neighborhood contained all the solutions that could be obtained from s using non-tabu moves. However, the tabu list raised the problem that might hinder some excellent solutions, which might not have been visited. That is to say, when a single attribute was marked as a tabu, this typically resulted in more than one solution being tabu. Some of these solutions that must now be avoided could be of excellent quality and might not have been visited. To mitigate this problem, “aspiration criteria” are introduced to override a solution’s tabu state.

The Tabu Search heuristic follows the general guidelines (Glover and Laguna 1997). Let s and s^* denote the current and best known solutions, respectively. G is the iteration counter and is used to control the loop in the following pseudo code (as shown in Fig. 1). H is the tabu list.

In the following section, we will present our AMTSSTS framework.

3.3 Framework of AMTSSTS

In our method, we treat any pixel value of a gray image as an individual over the population. The proposed AMTSSTS method consists of four main steps: first, an image is evenly divided into some image blocks (strata), and a sample containing a group of pixels is taken from each stratum; second, every sample is optimized by the Tabu Search algorithm

```

step 1. randomly generate an initial solution  $s$  and evaluate its fitness  $f(s)$ ;
step 2.  $s^* \leftarrow s, G=0, H=\{\}$ ;
step 3. WHILE not stop
    To generate a neighborhood of  $s$  through non tabu moves
    (or tabu moves that lead to solutions that improve  $s^*$ ) and
    select the best solution  $s'$ ;
    Update  $H$ ;
    IF  $s'$  is better than  $s^*$  then  $s^* \leftarrow s'$ ;
         $s \leftarrow s'$ ;
    END IF
     $G++$ ;
END WHILE
step 4. output  $s^*$ .
    
```

Fig. 1 The Tabu Search algorithm

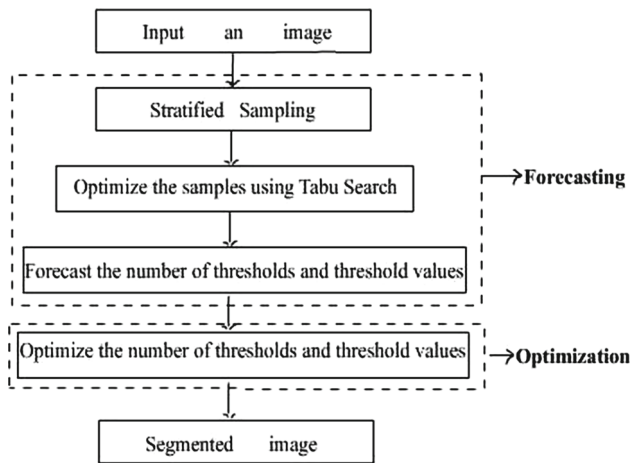


Fig. 2 The process of AMTSSTS

whose fitness function is the ratio of its mean and variance; third, the threshold number and threshold values are forecasted according to the optimized samples; and finally, the threshold number and threshold values are optimized by an optimization algorithm. The process of AMTSSTS is shown in Fig. 2. An example of using our segmentation method is further displayed in Fig. 3.

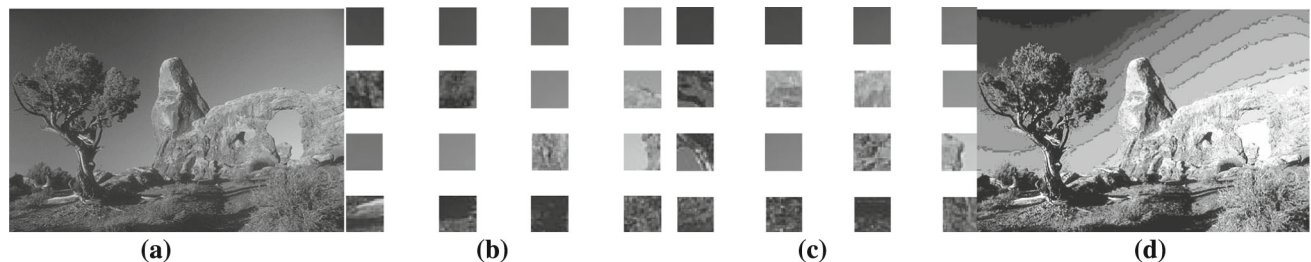


Fig. 3 An example of our segmentation method: **a** the original image, **b** the initial sample, **c** the optimized sample, and **d** the segmented image

3.3.1 Stratified sampling

Stratified sampling is a method of sampling from a population. In a stratified sample, the sampling frame is divided into non-overlapping groups or strata, e.g., geographical areas, age groups, or genders. A random or systematic sampling technique is then applied within each stratum. This often improves the representativeness of the sample by reducing sampling error. Stratification always achieves great precision provided that the strata have been chosen so that members of the same stratum are as similar as possible with regard to the characteristic of interest. The bigger the differences are between the strata, the greater the gain is in precision. The fitness function in our Tabu Search method is designed according to this principle.

An intensity image I is treated as a population which contains $m \times n$ gray values. We can predict the number of objects in an image by stratified sampling. The sampling method is as follows: an image I is evenly divided into ω strata (as shown in Fig. 4 ($\omega = 16$)). A sample is then drawn from the middle part of each stratum. The choice of selecting a sample size is discussed in Sect. 4.

3.3.2 The optimization method of samples using Tabu Search

This step is to optimize each sample separately using Tabu Search. The optimized samples can contain salient information about the threshold number and threshold values, and it will be more efficient if the parallelism among samples is developed. The steps and features of our Tabu Search method are described as follows:

1. Fitness function

All the samples of an image are arranged from left to right and top to bottom orientation. They are denoted as $\omega_1, \omega_2, \dots, \omega_{16}$. The mean and variance of ω_i are m_i and s_i , respectively, where $i = 1, 2, \dots, 16$. The fitness function of each sample designed according to Sect. 3.3.1, is then formulated as $f_i = m_i/s_i$. Note that we prefer a larger f_i, m_i , and the smaller s_i (In the program, the variance s_i adds 0.0001 temporarily when estimating f_i so as to avoid the exception of dividing by zero raised.)

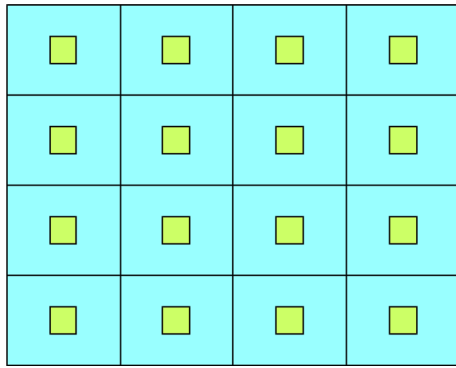


Fig. 4 The initial position of stratified sampling

2. Initialization

The initial solutions are generated by Stratified Sampling. Only one solution (sample) is drawn from each stratum.

3. Neighborhood

An initial solution s in each stratum has eight types of moves according to the direction of movement in the corresponding stratum. These are classified as “up move”, “top-right move”, “right move”, “bottom-right”, “down move”, “bottom-left”, “left move”, and “top-left move”. After the current solution s selects a move randomly, the new solution s' at each iteration is generated. Note that the move in every direction follows uniform distribution. More types of moves are needed to yield better solution; however, this increase causes more computational burden.

4. Tabu list, Tabu tenure, aspiration criteria

The Tabu list of each stratum is shown in Table 1. The status of each direction is initially set to 0 (if applicable). If s' is better than s , then $s \leftarrow s'$, or else the move is forbidden and the corresponding direction is set to 1 (if not applicable). When all kinds of moves are forbidden, the Tabu list will be reinitialized. This is the aspiration criterion and the tabu tenure strategy in our algorithm.

5. Stopping criteria

The search stops after a fixed number of generations is performed or after a maximum number of consecutive generations with no improvement in the best solution. In our experiment, we adopt the former, and G is set to 20.

3.3.3 The forecasting method for the number of threshold and threshold values

The third step is to forecast the threshold number and threshold values based on the optimized samples. The forecasting results fall into two categories:

Table 1 The initialization of Tabu list

Up	Top-right	Right	Bottom-right	Down	Bottom-left	Left	Top-left
0	0	0	0	0	0	0	0

1. $m_1 = m_2 = \dots = m_{16}$, the threshold number is equal to 1. This is the lower limit of the threshold number in our experiments given in Sect. 5.
2. $\exists_{i,j}, m_i \neq m_j$, where $i \neq j$ or $\exists_{u,v}, s_u \neq s_v$, where $u \neq v, u, v = 1, \dots, 16$, the threshold number is greater than 1 and not greater than 16. The upper limit of the threshold number is 16.

The procedure of our forecasting method is described as follows:

1. ω_1 is treated as a benchmark threshold value of the first category, if the mean of ω_i is between $m_1 \times (1 - rangeMean)$ and $m_1 \times (1 + rangeMean)$, and the variance is between $s_1 \times (1 - rangeStd)$ and $s_1 \times (1 + rangeStd)$, then ω_i belongs to the first category. The *rangeMean* and *rangeStd* are measured in percentage, and their ranges are between 0 and 100%. In our experiment, the gray levels of an image range over $[0, 255]$.
2. In the same way, we treat the first sample ω_j that is not yet categorized so far as another benchmark threshold value; if the mean of ω_i ($i = j, \dots, 16$) is between $m_j \times (1 - rangeMean)$ and $m_j \times (1 + rangeMean)$; and the variance is between $s_j \times (1 - rangeStd)$ and $s_j \times (1 + rangeStd)$, then ω_i belongs to the second category.
3. All the benchmark threshold values are the forecasting threshold values, the number of which is the threshold number. Empirically, *rangeMean* can be set to 30% in $m_i \times (1 \pm rangeMean)$ and *rangeStd* can be set to 20% in $s_i \times (1 \pm rangeStd)$.

3.3.4 The deterministic optimization algorithm

The fourth step is to further optimize the threshold number and threshold values obtained from the above by implementing a deterministic method which is described in Sect. 4.

4 Algorithm theoretical analysis

4.1 Design and analysis of the deterministic optimization algorithm

Let the gray levels of an image range over $[0, L]$ and $h(i)$ denotes the occurrence of gray level i . Let:

$$\begin{aligned} N &= h(0) + \dots + h(i) + \dots + h(L) \\ P_i &= h(i)/N \end{aligned} \quad (2)$$

Assuming that there are M thresholds: $\{t_1, t_2, \dots, t_M\}$, ($1 \leq M \leq L - 1$), which divide the original image into $M + 1$ classes represented as:

$$C_0 = \{0, 1, \dots, t_1\}, \dots, C_1 = \{t_1 + 1, t_1 + 2, \dots, t_2\}, \dots, C_M = \{t_M + 1, t_M + 2, \dots, L\}.$$

For multilevel thresholding, the formula based on Otsu's (Otsu 1979) method is computed as follows:

$$D(t_1, t_2, \dots, t_M) = \sum_{j=0}^{M-1} \sum_{k=j+1}^M \omega_j \omega_k (u_j - u_k)^2 \tag{3}$$

$$\omega_{j-1} = \sum_{i=t_{j-1}+1}^{t_j} P_i, u_{j-1} = \sum_{i=t_{j-1}+1}^{t_j} i \times P_i / \omega_{j-1},$$

where $t_0 = 0$. The optimal thresholds $t_1^*, t_2^*, \dots, t_M^*$ are the gray levels that maximize (3), i.e.,

$$t_1^*, t_2^*, \dots, t_M^* = \arg_{\max} D(t_1, t_2, \dots, t_M)$$

In the Otsu's method, we exhaustively search for the optimal threshold by maximizing inter-class variance. The k-means clustering aims to partition the n observations into k sets ($k \leq n$) by minimizing the within-cluster sum of squares. They both adopt the Euclidean distance measure as the similarity metric and squared error metric as the criterion function. The Otsu's method is extensively employed as the global criterion function of an image (Gao et al. 2010). Because an image has the property of local continuity, we believe that a local criterion function will be better for image thresholding. Therefore, a new local criterion function (4) is designed in the proposed algorithm.

$$f(v_r^*) = |v_r^* \times P_{v_r^*} - \sum_{i=t_{r-1}}^{t_{r+1}} i \times P_i / (t_{r+1} - t_{r-1} + 1)|, v_r^* \in [t_{r-1}, t_{r+1}] \tag{4}$$

Assuming that the set of the forecasting threshold values $F_C = \{t_1, t_2, \dots, t_M\}$, $t_1 \leq t_2, \dots, \leq t_M$, is arranged on a line segment (see Fig. 5). Let t_r denote an arbitrary forecasting threshold value and t_r^* be the corresponding optimal threshold value. We obtain t_r^* by Eq. (5).

$$t_r^* = \arg \min |v_r^* \times P_{v_r^*} - \sum_{i=t_{r-1}}^{t_{r+1}} i \times P_i / (t_{r+1} - t_{r-1} + 1)|, v_r^* \in [t_{r-1}, t_{r+1}] \tag{5}$$

When we obtain the first optimal threshold value t_1^* , F_C is immediately upgraded to $F'_C = \{t_1^*, t_2, \dots, t_M\}$. Consequently, the second optimal threshold value t_2^* is obtained according to F'_C , until $F'_C = \{t_1^*, t_2^*, \dots, t_M^*\}$, which is the set of the optimized threshold values.

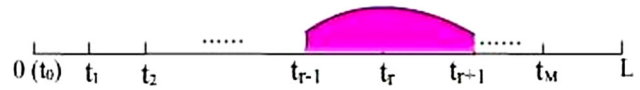


Fig. 5 The location of threshold values

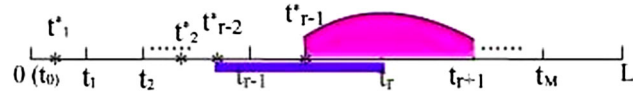


Fig. 6 The optimization of threshold values

Moreover, the threshold number will sometimes decrease after it is optimized: that is, when two optimized threshold values are the same or similar, the threshold number will decrease (see Fig. 6).

Assuming that t_{r-1}^* is obtained, a possible case that t_r^* will be equal to t_{r-1}^* is presented as follows:

$$t_{r-1}^* = \arg \min |v_{r-1}^* \times P_{v_{r-1}^*} - \frac{\sum_{i=t_{r-2}^*}^{t_r} i \times P_i}{t_r - t_{r-2}^* + 1}|, v_{r-1}^* \in [t_{r-2}^*, t_r] \tag{6}$$

$$t_r^* = \arg \min |v_r^* \times P_{v_r^*} - \frac{\sum_{i=t_{r-1}^*}^{t_{r+1}} i \times P_i}{t_{r+1} - t_{r-1}^* + 1}|, v_r^* \in [t_{r-1}^*, t_{r+1}] \tag{7}$$

Theorem 1 Let $A = \frac{\sum_{i=t_{r-2}^*}^{t_r} i \times P_i}{t_r - t_{r-2}^* + 1} = \frac{\sum_{i=t_{r-1}^*}^{t_{r+1}} i \times P_i}{t_{r+1} - t_{r-1}^* + 1}$, $A \leq t_{r-1}^* \times P_{t_{r-1}^*}$ (or $A \geq t_{r-1}^* \times P_{t_{r-1}^*}$) and $f(i) = i \times P_i$ is a strictly monotone increasing (or decreasing) function in $[t_r, t_{r+1}]$, then $t_r^* = t_{r-1}^*$.

Proof See Appendix □

4.2 Analysis of the stratified sampling method

The stratified sampling technique is similar to cluster analysis, which is the assignment of a set of observations into subsets (called clusters), so that observations in the same cluster are similar. This sampling is a method of sampling from a population, so it has lower computational complexity. However, Clustering needs to analyze all the pixels, and many clustering algorithms require the specification of the number of clusters, prior to the execution of the algorithm. What important is that an image has the property of local continuity and similarity, so it seems that the stratified sampling method is much suitable for automatic multilevel thresholding.

When we open our eyes on a scene, our visual system can automatically segment it into a set of objects in the scene. In computer vision, how many samples should we draw from

an image and how large should each sample be to represent the scene (image) better? We think it must be related to the content, size, and grayscale of an intensity image. Therefore, a thorough analysis is given in the following subsection.

4.2.1 Analysis on the number of samples

The segmented results of the two images are shown in Table 2. “ $a \times b$ ” denotes the number of rows of the samples is a and the number of columns is b . The sample size is $20 \times 20 = 400$. “Forecasted” denotes the forecasted thresholds by the forecasting method above, and “Optimized” denotes the optimized thresholds by the optimization algorithm above. From Table 2, the following observations can be made: the greater the number of samples is, the more the same/similar elements will sit between the set of “Forecasted” threshold values and the set of “Optimized” threshold values, but the higher the algorithmic complexity. The experimental results of other images show the same conclusion.

4.2.2 Analysis on the sample size

The experimental results of the four images are shown in Table 3. “ $c \times d$ ” denotes the sample size. From Table 3, we can conclude that the smaller the sample size is, the better its corresponding fitness is in general. The experimental results of other images show the same conclusion. But if the sample size is too small, it cannot represent the image better.

4.2.3 Analysis of time complexity

The computational time of the thresholding methods based on the Otsu’s method is very expensive when the exhaustive search is applied. For M thresholds, the complexity is $O(L^M)$, which grows exponentially with the threshold number. For AMTSSTS, the complexity of the Tabu Search method is $O(G * \omega * tt)$, the complexity of the forecasting method is $O(\omega * (\omega - 1)/2)$, the complexity of the deterministic optimization algorithm is $O(M * L)$; where G is the generation number, ω is the number of samples, tt is the Tabu tenure. A simple calculation shows that AMTSSTS has lower complexity and is insensitive to the threshold number, which is also shown in Table 5.

5 Experiments

5.1 Setup

We compare the performance of the proposed AMTSSTS algorithm with other algorithms. They are (1) the clustering-

Table 2 The experiment on the number of samples

22090.jpg	4×5	4×6	4×7	4×8
Forecasted	59,77,83,145,152,154,155,174	54,72,112,144,152,153,157,159,171	72,121,146,151,152,155,158,167	74,79,99,111,123,126,139,147,162,192
Optimized	59,81,137,152,159	54,72,112,144,157,171	72,128,151,158,167	74,99,111,126,139,147,162,189
22090.jpg	5×5	5×6	5×7	5×8
Forecasted	54,77,112,123,126,134,145,149,151,163,191	47,82,86,110,123,125,152,158,160,189	40,56,70,79,91,107,134,146,149,150,194	40,68,71,101,107,125,152,156,160,210
Optimized	54,81,119,138,163,195	47,77,84,110,128,155,189	40,56,70,79,91,107,134,155,194	40,68,107,125,156,210
65019.jpg	4×5	4×6	4×7	4×8
Forecasted	9,18,24,113,147,159,162,182,192,206,213	21,28,44,61,76,106,110,154,211,216	18,19,20,25,56,73,91,102,131,134,139,188	18,20,35,71,98,107,153,160,174,198,208
Optimized	20,109,159,187,203,213	18,44,61,81,108,154,196,216	18,25,56,73,91,102,134,188	20,35,71,107,160,177,198,211
65019.jpg	5×5	5×6	5×7	5×8
Forecasted	9,20,32,100,114,141,176,182,188,201,204	18,58,82,127,168,185,188,189,191,209	18,28,44,53,66,79,93,118,119,159,169,184	20,31,50,61,77,114,168,170,194,204,219
Optimized	28,104,114,131,173,182,204	19,28,58,82,140,168,185,212	18,28,44,53,66,79,93,118,159,184	20,31,50,63,77,114,170,194,204,219

Table 3 The experiment on the sample size

	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{12}	f_{13}	f_{14}	f_{15}	f_{15}	f_{16}
376001.jpg																
30×30	18.23	15.47	16.39	11.91	18.31	11.33	12.27	14.95	10.67	17.53	17.79	16.69	13.23	24.47	14.98	12.95
20×20	23.32	16.47	16.69	15.86	18.75	20.20	14.75	16.51	13.40	22.63	17.32	19.77	16.93	27.63	17.97	12.85
10×10	29.62	26.49	20.58	18.29	18.34	21.02	22.18	18.03	37.31	73.20	20.68	21.47	25.92	51.15	70.97	44.18
170057.jpg																
30×30	31.45	38.88	25.34	16.11	28.29	27.43	16.01	14.61	21.75	19.95	17.46	11.87	16.15	20.02	22.76	13.30
20×20	45.14	49.48	29.45	18.95	36.95	32.30	19.50	15.35	28.44	25.09	18.07	12.99	16.24	20.43	21.13	13.44
10×10	54.34	62.86	55.0.8	19.42	49.22	15.57	29.75	25.64	42.65	27.40	26.92	16.33	27.84	29.12	30.53	17.48
35008.jpg																
30×30	13.23	23.64	19.18	42.08	24.59	18.31	18.13	45.93	16.78	14.22	17.08	33.90	22.64	15.45	13.68	17.13
20×20	23.91	41.56	27.90	89.47	33.82	46.16	21.60	47.68	27.44	13.28	21.16	59.70	34.33	20.22	33.54	29.06
10×10	141.64	63.68	54.06	144.73	130.75	133.29	86.63	170000	62.28	17.56	80.65	131.71	262.63	59.31	137.74	44.02
253055.jpg																
30×30	2550000	258.09	179.04	94.34	42.90	35.52	37.94	308.89	35.95	19.58	18.79	20.48	29.75	19.53	22.26	23.80
20×20	2550000	820.96	567.42	123.99	134.71	53.27	48.99	311.89	57.59	23.58	22.45	30.22	36.20	19.85	24.51	27.67
10×10	2550000	1948.70	975.07	203.23	418.59	258.37	120.78	654.99	171.48	106.37	70.86	76.10	41.20	30.23	30.80	26.63

based approach using a hierarchical evolutionary algorithm (CHEA) (Lai and Chang 2009), (2) the quantum-behaved PSO algorithm employing the cooperative method (CQPSO) (Gao et al. 2010), and (3) the maximum entropy-based honey bee mating optimization thresholding (MEHBMOT) method (Hornig 2010). CHEA can automatically determine the threshold number. For CQPSO and MEHBMOT, their threshold numbers are preset to values estimated by the proposed method.

The experiments are implemented on the Berkeley segmentation data set (Research 0000). The segmented results of nine images are shown in Fig. 9. Figures 7 and 8 display these nine original images with their corresponding histograms. In AMTSSTS, the image is 8-bit grayscale and the size is 481×321 . The sample size is 10×10 and the number of samples is 16; that is, the upper limit of the threshold number is 16 and it is also easy to distinguish between the levels of brightness by human eyes. The other parameters of AMTSSTS are given in Sect. 3.3 and the parameters of other algorithms are set the same as described in their corresponding paper, except that the generation numbers in CQPSO and MEHBMOT are set to 100. We then conduct our experiments on a Compad notebook, with 512 MB RAM and a single core 1.6 GHz CPU, Microsoft Windows XP, and Matlab 7.2. Moreover, three aspects of the performance evaluation are taken into consideration in the experiments: (1) the comparison of segmentation results, (2) the comparison of threshold values, peak signal to noise ratio (PSNR), and computation time, and (3) the ability to conquer “the curse of dimensionality”. In our experiments, when two thresholds satisfy the condition that the absolute value of their difference is not

more than 5, we will unify them. Consequently, the threshold number obtained through the experiments may be less than M ; i.e., the initial number of thresholds in CQPSO and MEHBMOT.

5.2 Results

5.2.1 Comparison of segmentation results

Figure 9 displays the visual interpretation of the segmentation results. The findings show that our method outperformed others in terms of segmentation quality; in particular, those images of Fig. 9a–f, h images. When the segmented images are enlarged, the following findings are observed: in Fig. 9a, our method obtained the clearer results with 7 thresholds than those of CHEA with 6 thresholds. In Fig. 9c, the segmented result of our method was absolutely clear, except that the upper of the sky was dim. Although CHEA removed the blur of the sky, the bridge and boat were blurred. CQPSO obtained even worse results than those of CHEA. MEHBMOT could clearly separate the sky and water, but the scene was a bit obscured and gloomy. In Fig. 9d, the result of CHEA was more blurred than that of our method and the writing was illegible; the result of CQPSO was also a bit vague. The results of MEHBMOT and our method were similar. In Fig. 9f, the three faces of the results of CHEA and CQPSO were vague and the profiles were not very clear. However, our method and MEHBMOT showed promising results and the segmentation result of our method was much clear and precise. In Fig. 9h, our method also produced a more continuous and smoother segmentation result than others.



Fig. 7 The nine original images

5.2.2 Comparison of threshold values, PSNR, computation time

In this section, we use peak signal to noise ratio (PSNR) (Horng 2010; Arora et al. 2008) to evaluate the quality of segmented images. *PSNR* is defined and measured in decibel (dB), and as follows:

$$\text{PSNR} = 20 \log_{10} \left(\frac{255}{\text{RMSE}} \right), \quad (8)$$

where *RMSE* is the root mean-squared error defined as:

$$\text{RMSE} = \sqrt{\sum_{i=1}^m \sum_{j=1}^n (I(i, j) - I'(i, j))^2 / (m \times n)}, \quad (9)$$

where *I* and *I'* are the original and segmented images of size $m \times n$, respectively. A larger value of *PSNR* means that the quality of the segmented image is better. Furthermore, as the threshold number increases, the *PSNR* tends to be larger. Table 4 shows the comparative analyses of AMTSSTS, CHEA, CQPSO, and MEHBMOT in terms of threshold values, *PSNR*, and computational time over different images. CHEA was the slowest among the four methods. For AMTSSTS, the computation time was insensitive to the threshold number and was faster than other algo-

rithms. The *PSNR* of images 'a1', 'b1', 'e1', 'f1' and 'h1' (as shown in Fig. 9) was the largest among the four methods. The *PSNR* of image 'i1' in Fig. 9 was smallest and the *PSNR* values of the images 'c1', 'd1' and 'g1' were close to the best. Moreover, the visual inspection of the segmented images shows that AMTSSTS can obtain clearer and more complete segmentation images for video image compression.

Note that in Table 4, *M* was predetermined in CQPSO and MEHBMOT according to that estimated by AMTSSTS; *M* was automatically determined in CHEA and AMTSSTS.

5.2.3 Curse of dimensionality

To verify the searching ability of our proposed method on high dimensionality, we tested each image, with small values of *rangeMean* and *rangeStd*. Table 5 shows the mean computation time (seconds) for 100 runs. This is also consistent with earlier findings suggesting that our proposed method is robust under high dimensionality.

6 Conclusion and future work

This article proposed the AMTSSTS algorithm for automating the multilevel thresholding selection procedure using stratified sampling and a new local criterion function, which

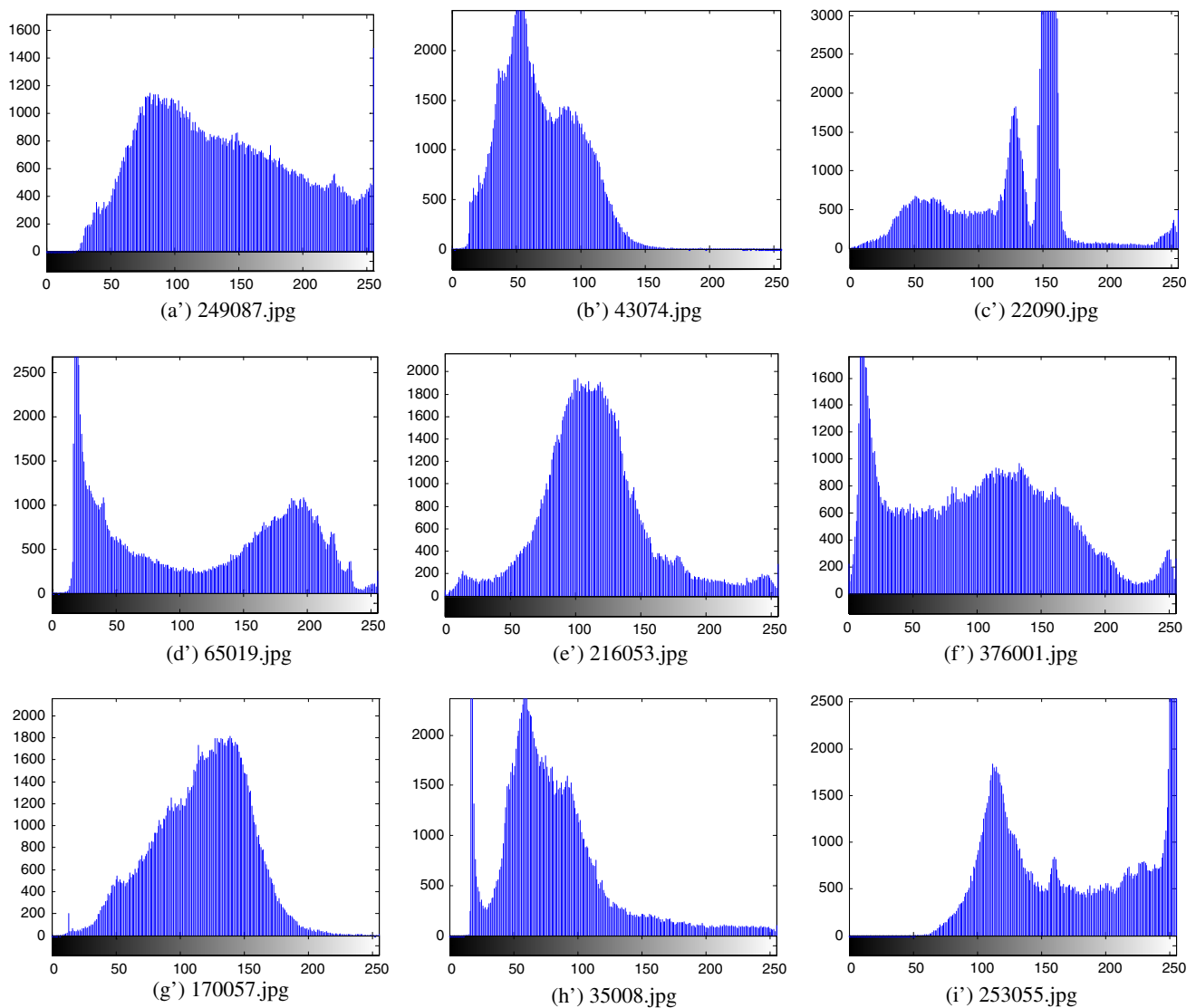


Fig. 8 The nine corresponding histograms of the original images

extends our preliminary research (Jiang et al. 2012). It has been shown in the previous sections, with the help of results obtained for nine benchmark images, that the proposed method was able to converge rapidly and to produce better segmentation results than several new and developed methods from recent years. In addition, the algorithm has the following characteristics: (a) it does not need to consider any auxiliary or extra image information such as contextual or textual properties. The threshold number does not need to be known in advance, (b) AMTSSTS avoids analyzing histograms and has a lower computing complexity that is almost independent from the threshold number, and (c) the proposed algorithm can produce effective, efficient, and smoother results. We also presented the detailed analysis from three aspects: its optimization method, sampling method and algorithmic complexity.

In our future research, we will further attempt to find a better method to forecast the threshold values and its numbers, with the help of statistical methods and the theory of computer vision. Furthermore, we will apply it to complex and real-time image analysis problem, such as automatic target recognition and medical image analysis.

Appendix: Proofs for Sect. 4

Proof of Theorem 1

Proof Let $\beta = |t_{r-1}^* \times P_{t_{r-1}^*} - A|$.

According to Eqs. (6) and (7), we have,

$$\min |v_r^* \times P_{v_r^*} - \frac{\sum_{i=t_{r-1}^*}^{t_{r+1}} i \times P_i}{t_{r+1} - t_{r-1}^* + 1}| = \min$$

Fig. 9 Comparison of segmentation results: (a1)–(i1) the segmentation results of our method, (a2–i2) the segmentation results of CHEA, (a3–i3) the segmentation results of CQPSO, and (a4–i4) the segmentation results of MEHBMOT



Table 4 Comparison of threshold values, uniformity, computation

	Method	M	Threshold value	PSNR	CPU time (s)
249087.jpg	AMTSSTS	7	64-102-123-141-170-189-232	23.5496	0.346206
	CHEA	6	34-97-137-167-236-252	22.7643	6.866401
	CQPSO	6	20-28-97-103-111-249	13.6371	4.070220
	MEHBMOT	7	39-107-123-166-178-186-249	21.6123	3.231057
43074.jpg	AMTSSTS	7	64-88-113-129-166-179-218	24.2997	0.351059
	CHEA	7	22-39-47-58-68-181-196	10.4783	6.128226
	CQPSO	5	28-49-57-141-147	11.1121	3.541111
	MEHBMOT	7	15-36-54-83-93-104-117	11.8353	3.434535
22090.jpg	AMTSSTS	7	70-107-124-153-218	18.3303	0.419265
	CHEA	8	12-21-35-75-105-119-146-194	12.2338	5.689608
	CQPSO	5	51-67-82-95-107	8.7573	3.467276
	MEHBMOT	5	47-90-140-178-235	24.4017	2.915946
65019.jpg	AMTSSTS	8	18-143-150-161-177-197-203-217	18.4390	0.353223
	CHEA	7	18-25-36-51-76-120-228	13.1588	5.746528
	CQPSO	7	57-68-83-90-130-147-156	15.7061	3.387223
	MEHBMOT	7	27-58-86-138-148-187-212	22.5091	3.448367
216053.jpg	AMTSSTS	8	60-83-118-149-157-176-200-230	22.7950	0.346077
	CHEA	7	16-22-85-97-120-134-149	13.9575	6.058972
	CQPSO	8	85-94-103-109-120-134-153-173	16.5322	4.083998
	MEHBMOT	8	57-94-122-148-161-185-200-236	21.2126	3.360095
376001.jpg	AMTSSTS	7	43-62-126-145-164-193-227	24.8587	0.335823
	CHEA	6	16-30-40-54-118-226	11.0668	5.798094
	CQPSO	5	18-51-57-100-107	9.4820	3.019550
	MEHBMOT	7	100-112-127-152-171-185-237	19.2043	3.440645
170057.jpg	AMTSSTS	5	101-128-153-160-172	17.9138	0.380617
	CHEA	9	17-36-42-53-88-105-154-206-235	14.7946	5.678565
	CQPSO	5	47-90-208-216-227	21.0741	3.180481
	MEHBMOT	5	40-69-119-155-188	17.6656	3.250963
35008.jpg	AMTSSTS	8	29-52-85-111-149-176-199-234	23.8800	0.361907
	CHEA	8	12-20-34-48-72-84-130-249	10.1670	5.626506
	CQPSO	6	44-51-117-140-178-237	18.6445	3.023091
	MEHBMOT	8	43-58-79-93-130-164-186-216	21.5595	3.563338
253055.jpg	AMTSSTS	6	116-124-138-198-227-249	16.3590	0.387195
	CHEA	6	13-75-94-128-158-246	19.1991	6.101050
	CQPSO	5	95-123-137-159-171	18.2997	3.061366
	MEHBMOT	6	96-113-133-158-167-223	21.3524	3.015020

Table 5 Computation time with $M = 6, 9, 12, 16$

M	249087.jpg	43074.jpg	22090.jpg	65019.jpg	216053.jpg	376001.jpg	170057.jpg	35008.jpg	253055.jpg
6	0.347164	0.352324	0.340573	0.387256	0.32609	0.369940	0.366950	0.356532	0.365436
9	0.418990	0.383675	0.365989	0.384778	0.364941	0.372522	0.393571	0.372593	0.380177
12	0.431369	0.393772	0.395658	0.388693	0.377185	0.385911	0.387471	0.375669	0.385629
16	0.470666	0.416313	0.473496	0.390792	0.381030	0.446923	0.414797	0.388894	0.415822

$$|v_{r-1}^* \times P_{v_{r-1}^*} - \frac{\sum_{i=t_{r-2}^*}^{t_r^*} i \times P_i}{t_r - t_{r-2}^* + 1}| = \beta, \quad v_r^* \in [t_{r-1}^*, t_r^*).$$

When $v_r^* \in [t_r, t_{r+1}]$, $f(v_r^*)$ is a strictly monotone increasing function, hence $|v_r^* \times P_{v_r^*} - A| > \beta$. So, we can obtain the function

$$|v_r^* \times P_{v_r^*} - \frac{\sum_{i=t_{r-1}^*}^{t_{r+1}} i \times P_i}{t_{r+1} - t_{r-1}^* + 1}|, \quad v_r^* \in [t_{r-1}^*, t_{r+1}],$$

which can get the minimum value when $v_r^* = t_{r-1}^*$ in Eq. (7); i.e., $t_r^* = t_{r-1}^*$. The similar method for the case $A \geq t_{r-1}^* \times P_{t_{r-1}^*}$.

References

- Arora S, Acharya J, Verma A, Panigrahi PK (2008) Multilevel thresholding for image segmentation through a fast statistical recursive algorithm. *Pattern Recognit Lett* 29(2):119–125
- Bhattacharyya S et al (2011) Multilevel image segmentation with adaptive image context based thresholding. *Appl Soft Comput* 11:946–962
- Bosco GL (2001) A genetic algorithm for image segmentation. In: *Proceedings of IEEE international conference on image analysis and processing*, Palermo, pp. 262–266
- Cao L, Bao P, Shi ZK (May 2008) The strongest schema learning GA and its application to multilevel thresholding. *Image Vis Comput* 26(5):716–724
- Chander A, Chatterjee A, Siarry P (2011) A new social and momentum component adaptive PSO algorithm for image segmentation. *Exp Syst Appl* 38:4998–5004
- Chang CY, Chung PC (2001) Medical image segmentation using a contextual-constraint-based Hopfield neural cube. *Image Vis Comput* 19:669–678
- Chen S, Zhang D (2004) Robust image segmentation using FCM with spatial constraints based on new kernel-induced distance measure. *IEEE Trans Syst Man Cybern B* 34:1907–1916
- Cheng HD, Lui YM (1997) Automatic bandwidth selection of fuzzy membership function. *Inf Sci* 103:1–21
- Chuang KS, Tzeng HL, Chen S et al (2006) Fuzzy C-means clustering with spatial information for image segmentation. *Comput Med Imaging Graph* 30:9–15
- Fan SKS, Lin Y (2007) A multi-level thresholding approach using a hybrid optimal estimation algorithm. *Pattern Recognit Lett* 28:662–669
- Gao H, Xu W, Sun J, Tang Y (2010) Multilevel thresholding for image segmentation through an improved quantum-behaved particle swarm algorithm. *IEEE Trans Instrum Meas* 59(4):934–946
- Glover F (1989) Tabu search-part I. *ORSA J Comput* 1(3):190–206
- Glover F (1990) Tabu search-part II. *ORSA J Comput* 2(1):4–32
- Glover F, Laguna M (1997) *Tabu Search*. Kluwer, Boston
- Hammouche K, Diaf M, Siarry P (2008) A multilevel automatic thresholding method based on a genetic algorithm for a fast image segmentation. *Comput Vis Image Understand* 109(2):163–175
- Hornig MH (2010) A multilevel image thresholding using the honey bee mating optimization. *Appl Math Comput* 215(9):3302–3310
- Hou Z, Hu Q, Nowinski WL (2006) On minimum variance thresholding. *Pattern Recognit. Lett.* 27:1732–1743
- <http://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/>
- Huang LK, Wang MJ (1995) Image thresholding by minimizing the measure of fuzziness. *Pattern Recognit* 28:41–51
- Itti L, Koch C, Niebur E (1998) A model of saliency-based visual attention for rapid scene analysis. *IEEE Trans Pattern Anal Mach Intell* 20(11):1254–1259
- Jiang Y, Tsai P, Hao Z, Cao L (2012) A novel auto-parameters selection process for image segmentation, WCCI 2012. *IEEE World Congress on Computational Intelligence*, Brisbane, Australia, 10–15 June 2012
- Kittler J, Illingworth J (1986) Minimum error thresholding. *Pattern Recognit* 19:41–47
- Lai CC, Tseng DC (2004) A hybrid approach using Gaussian smoothing and genetic algorithm for multilevel thresholding. *Int J Hybrid Intell Syst* 1(3):143–152
- Lai CC, Chang CY (2009) A hierarchical evolutionary algorithm for automatic medical image segmentation. *Exp Syst Appl* 36:248–259
- Li X, Zhao Z, Cheng HD (1995) Fuzzy entropy threshold approach to breast cancer detection. *Inf Sci* 4:49–56
- Lin Z, Wang Z, Zhang Y (2010) Optimal evolution algorithm for image thresholding[J]. *J Comput-Aided Des Comput Graph* 22(7):1201–1206 (In Chinese)
- Li M, Staunton RC (Nov. 2007) A modified fuzzy C-means image segmentation algorithm for use with uneven illumination patterns. *Pattern Recognit* 40(11):3005–3011
- Ma Z, Tavares JMRS, Jorge RN, Mascarenhas T (2010) A review of algorithms for medical image segmentation and their applications to the female pelvic cavity. *Comput Methods Biomech Biomed Eng* 13(2):235–246
- Malisia AR, Tizhoosh HR (2006) Image thresholding using ant colony optimization. In: *Proceedings of the 3rd Canadian conference on computer and robot vision*, Waterloo, pp 26–31
- Otsu N (1979) A threshold selection method from gray-level histograms. *IEEE Trans Syst Man Cybern SMC-9*:62–66 (1979)
- Sezgin M, Sankur B (2004) Survey over image thresholding techniques and quantitative performance evaluation. *J Electron Imaging* 13(1):146–165
- Tan KS, Isa NAM (2011) Color image segmentation using histogram thresholding-fuzzy C-means hybrid approach. *Pattern Recognit* 44:1–15
- Tao WB, Jin H, Liu LM (May 2008) Object segmentation using ant colony optimization algorithm and fuzzy entropy. *Pattern Recognit Lett* 28(7):788–796
- Tsotsos JK, Culhane SM, Wai WYK, Lai YH, Davis N, Nuflo F (1995) Modelling visual attention via selective tuning. *Artif Intell* 78(1–2):507–545
- Yang ZH, Pu ZB, Qi ZQ (2003) Relative entropy multilevel thresholding method based on genetic algorithm. In: *IEEE international conference on neural networks and signal processing*, Nanjing, China, pp 583–586
- Ye Q, Danielsson P (1988) On minimum error thresholding and its implementations. *Pattern Recognit Lett* 7:201–206
- Yen JC, Chang FJ, Chang S (1995) A new criterion for automatic multilevel thresholding. *IEEE Trans Image Process* IP-4:370–378 (1995)
- Yin PY (1999) A fast scheme for optimal thresholding using genetic algorithms. *Signal Process* 72:85–95
- Yin PY (2007) Multilevel minimum cross entropy threshold selection based on particle swarm optimization. *Appl Math Comput* 184:503–513
- Zahara E, Fan SKS, Tsai DM (2005) Optimal multi-thresholding using a hybrid optimization approach. *Pattern Recognit Lett* 26:1082–1095