WCCI 2012 IEEE World Congress on Computational Intelligence
June, 10-15, 2012 - Brisbane, Australia
IEEE CEC

A NOVEL AUTO-PARAMETERS SELECTION PROCESS FOR IMAGE SEGMENTATION

Yunzhi Jiang[a,b], Pohsiang Tsai[*,d], Zhifeng Hao[a,c], Longbing Cao[b]

[a] School of Computer Science and Engineering, South China University of Technology, China
[b] Centre for Quantum Computation and Intelligent Systems, Advanced Analytics Institute, University of Technology, Sydney, Australia;
[c] Faculty of Computer, Guangdong University of Technology, China
[d] Department of Computer Science and Information Engineering, National Formosa University, Taiwan

ABSTRACT

Segmentation is a process to obtain the desirable features in image processing. However, the existing techniques that use the multilevel thresholding method in image segmentation are computationally demanding due to the lack of an automatic parameter selection process. This paper proposes an automatic parameter selection technique called an automatic multilevel thresholding algorithm using stratified sampling and Tabu Search (AMTSSTS) to remedy the limitations. It automatically determines the appropriate threshold number and values by (1) dividing an image into even strata (blocks) to extract samples; (2) applying a Tabu Search-based optimization technique on these samples to maximize the ratios of their means and variances; (3) preliminarily determining the threshold number and values based on the optimized samples; and (4) further optimizing these samples using a novel local criterion function that combines with the property of local continuity of an image. Experiments on Berkeley datasets show that AMTSSTS is an efficient and effective technique which can provide smoother results than several developed methods in recent years.

Keywords: Image Segmentation; Multilevel Thresholding; Stratified Sampling; Tabu Search

1. INTRODUCTION

Many environments, such as public train stations or airports, require visual surveillance systems to protect the public's safety. This security issue requires techniques such as image segmentation, image processing and computer vision to segment the targeted region of interest (ROI) for further suspicious patterns analysis and recognition. Image segmentation is the process of assigning a label to every pixel in an image; the pixels with the same label are called a region and share certain visual characteristics. Each region is homogeneous and connected. The union of any two spatially adjacent regions is not homogenous.

Four popular image segmentation approaches: threshold techniques, edge-based methods, region-based techniques, and connectivity-preserving relaxation methods, have been used for intensity images. Of these, thresholding methods have gained most of the attention. The thresholding techniques involve bi-level thresholding and multilevel thresholding. Bi-level thresholding classifies the pixels into two groups and multilevel thresholding divides the pixels into several classes. These techniques can be also categorized into parametric and nonparametric approaches, and a rigorous survey can be found in [1-2]. 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. Nonparametric approaches 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, and other criteria. They are easy to extend to multilevel thresholding; however, the amount of thresholding computation significantly increases with this extension. Consequently, multilevel thresholding techniques using Intelligent Optimization Algorithms (IOAs) have been proposed to overcome this problem. Chen et al. [3] proposed two variants of Fuzzy C-means Clustering with spatial constraints for image segmentation, which were robust to noise and outliers. Yin [4] developed a recursive programming technique to reduce the order of magnitude of computing the multilevel thresholds, and further used the Particle Swarm Optimization (PSO) algorithm to minimize cross entropy. Ye et al. [5] proposed a particle swarm optimization algorithm to optimize Otsu’s criterion. Hammouche et al. [6] proposed a multilevel thresholding technique based on the genetic algorithm (GA), which was combined with a wavelet transform-based technique to fast determine the appropriate threshold number and values, but the automatic thresholding criteria used here was still time-consuming. Tao et al. [7] proposed a fuzzy entropy method that incorporated ant colony optimization (ACO). Gao et al. [8] proposed the quantum-behaved PSO, which employed the cooperative method to save computation time and to conquer the curse of dimensionality. Horng [9] proposed a new algorithm based on the honey bee mating optimization, using the maximum entropy criterion. Bhattacharyya et al. [10] proposed a multilevel method using a self-supervised

Supported by Natural Science Foundation of China (61070033), Natural Science Foundation of Guangdong province (9251009001000005).

multilayer self-organizing neural network and a supervised pyramidal neural network. However, how to automatically decide the threshold number in these works [3-10] was not mentioned. Although many segmentation methods based on IOAs have been used to reduce the computational time in thresholding, they still suffer from the following: (a) the number of classes is predetermined, which implies that users must identify the number of regions beforehand; (b) they need preprocessing stages to reduce or to remove the noise; (c) they enhance the convergence rate or improve the stability by using special techniques [3,6,8]; and (d) the IOA is employed to optimize the global criterion function of an image; however, no consideration is given to local continuity of an image.

To rectify these limitations, an automatic multilevel thresholding algorithm using stratified sampling and Tabu Search (AMTSSTS) is proposed. This method was based on model research into the visual attention of primates[11-12], showing that higher visual processes appear to select a subset of available sensory information before further processing.

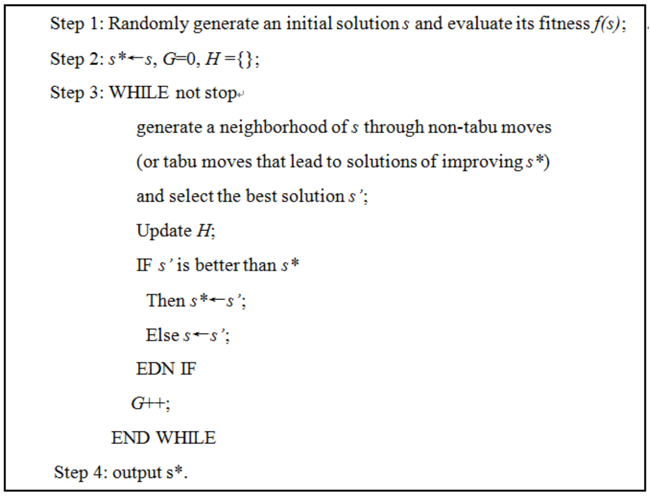
2. METHODOLOGY

In this section, we first described the Tabu Search algorithm, whose purpose is to optimize the sampling procedure, and we then elaborate our proposed algorithm in more detail. The aim of our algorithm is to predict the appropriate threshold number and values using various combined procedures – the prediction method, stratified sampling with Tabu search, and an optimization algorithm, with a newly employed local criteria function.

2.1 Tabu Search

The Tabu Search [13-15] algorithm proposed by Glover was successfully applied to various combinatorial problems, including planning and scheduling problems. It was an iterative heuristic optimization method based on (1) the local search techniques started from a feasible solution, (2) unexplored solutions space searching at each iteration, and (3) avoidance of repetitions with the help of the tabu list. This heuristic method followed the general guidelines stated in [15]. While this list was a short-term memory mechanism containing solutions that had been visited in the recent past, it raised the problem that might miss some excellent solutions which might not have been visited. Nonetheless, an aspiration criterion was introduced to mitigate this problem by overriding the tabu state in solutions space.

Let s and s^* denoted the current and best known solutions respectively. G is the iteration counter and H is the tabu list. Note that G, the number of generations, is used to control the loop in the following figure. Figure 1 depicts this 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
            generate a neighborhood of $s$ through non-tabu moves
            (or tabu moves that lead to solutions of improving $s^*$)
            and select the best solution $s'$;
            Update $H$;
            IF $s'$ is better than $s^*$
              Then $s^* \leftarrow s'$;
              Else $s \leftarrow s'$;
            EDN IF
           $G$++;
        END WHILE
Step 4: output s*.
```

Figure 1: The Tabu Search algorithm.

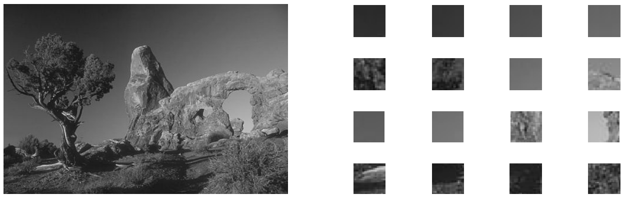
2.2 The Proposed Algorithm

An image is treated as a population that contains the gray values of pixels. The proposed AMTSSTS method consists of four main steps: 1) an image is evenly divided into several blocks (strata), and a sample is taken from each stratum; (2) every sample is optimized by Tabu Search heuristics whose fitness function is the ratio of its mean and variance; (3) the threshold number and values are forecasted according to the optimized samples, and (4) the threshold number and values are optimized by an optimization algorithm. We illustrate this algorithm in more detail in the following subsections.

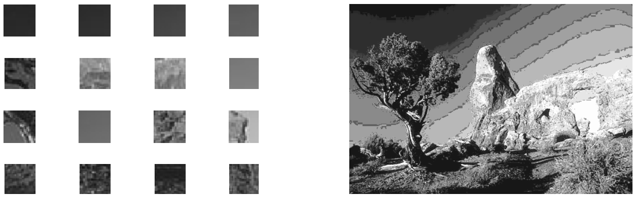
A) Stratified Sampling

Stratified Sampling is a method of sampling from a population which often improves the representativeness of the sample by reducing its sampling error. Stratification always achieves great precision provided that the strata have been chosen so that members of the same stratum are closely similar with regard to the characteristics of interests.

An intensity image I is treated as a population containing $m \times n$ gray pixel values. The number of objects in the image can be predicted by the stratified sampling technique. The sampling method is as follows: an image I is evenly divided into ω strata (as shown in Figure 2 (ω=16)). A 20×20 sample is then drawn from the middle part of each stratum.



2.1 The original image 2.2 The initial sample



2.3 The optimized sample 2.4 The segmented image

Figure 2. An example of our segmentation method (ω =16).

B) The optimization method of samples using Tabu Search

This section described the steps to separately optimize each sample using Tabu Search presented as follows:

a). Fitness function

All the samples in the image are arranged from left to right and top to bottom orientations. They are denoted as $\omega_1,\omega_2,...,\omega_{16}$.The mean and variance of ω_t are m_i and s_i; respectively, where i=1,2,…,16. The fitness function of each sample is formulated as $f_i = m_i / s_i$. Note, in this case, that we prefer a larger f_i . (In the program, the variance s_i adds 0.0001; or else an exception is raised.)

b). Initialization

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

c). 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". The new solution *s'* at each iteration is generated when the current solution *s* randomly selects a move. The move in every direction follows uniform distribution.

d). Tabu list, tatu tenure, aspiration criteria

The tabu list of each stratum is shown in Figure 3. The status of each direction is initially set to 0 (if applicable). If *s'* is better than *s*, then *s*←*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 tatu tenure strategy adopted in our algorithm.

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

Fig. 3. The initialization of tabu list

e). Stopping criteria

The search stops after a fixed number of generations have been 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.

C) The forecasting method of the number of thresholds and threshold values

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

i). m_1=m_2=…=m_{16}, and s_1=s_2=…=s_{16}=0, the threshold number is equal to 1. This is the lower limit of the threshold number in our experiments described in Section 3.

ii). ∀ *i,j*, $m_i \neq m_j$, where $i \neq j$, *i,j*=1,…,16; and s_1=s_2=…=s_{16}=0, the number of thresholds is equal to 16. This is the upper limit of the threshold number in our experiments.

iii). ∃ *i,j*, $m_i \neq m_j$, where $i \neq j$ or ∃ *u,v*, $s_u \neq s_v$, where $u \neq v$, *u,v*=1,…,16, the number of thresholds is greater than 1 and smaller than 16.

The procedure of our forecasting method hence can be described as:

i). ω_1 is treated as a benchmark threshold value of the first category, if the mean of ω_i is between m_1×(1-*rangeMean*) and m_1×(1+*rangeMean*), and the variance is between s_1×(1-*rangeStd*) and s_1×(1+*rangeStd*), then ω_i belongs to the first category. The values of *rangeMean* and *rangeStd* are between 0 and 1.

ii). In the same way, we treat the first sample not yet categorized as another benchmark threshold value to date; if the mean of ω_i (*i*=*j* , …, 16) is between m_j×(1-*rangeMean*) and m_j×(1+*rangeMean*); and the variance is between s_j×(1-*rangeStd*) and s_j×(1+*rangeStd*); then ω_i belongs to the second category.

iii). 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% and *rangeStd* to 20% in the proposed algorithm.

D) The deterministic optimization algorithm for the number of thresholds and threshold values

The fourth step is to further optimize the threshold number and values obtained from the above by implementing the deterministic method described below. In our experiments, when two thresholds satisfy the condition that the absolute value of their difference is less than 5, we will unify them. The step is as follows:

Let the gray levels of a given image ranged over [0, *L*] and *h(i)* be denoted as the occurrence of gray level *i*; and also let

$$N=h(0)+\cdots+h(i)+\cdots+h(L)$$
$$p_i= h(i) / N \qquad (1)$$

Assuming that there are *M* thresholds: {t_1, t_2,…, t_M}, where (1≤*M*≤*L*-1), which divide the original image into *M* +1 classes that are represented in the following notations: C_0 = {0,1,…,t_1} ,…, C_1 = { t_2+1, t_2+2,…, t_3} ,…, C_M = { t_M + 1, t_M + 2,…,*L*}.

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

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

, where

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

The optimal thresholds $t_1^*, t_2^*,\ldots, t_M^*$ are the gray- level values that maximize (2); that is,

$$t_1^*, t_2^*,\ldots, t_M^*=Arg\max D(t_1, t_2,\ldots, t_M). \tag{3}$$

In 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 ($k \le n$) by minimizing the within-cluster sum of squares. They both adopt the Euclidean distance measure as the similarity metric and the squared error metric as the criterion function. Otsu's method is extensively employed as the global criterion function of an image [8]. Because an image has the property of local continuity, we believe that a local criterion function will be better for image thresholding. Thus, a new local criterion function Eq. (4) is designed as:

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

Assuming that the set of the forecasting threshold values $FC = \{t_1, t_2,\ldots, t_M\}$, where $t_1\le t_2,\ldots, \le t_M$, is arranged on a line segment (as shown in Figure 4). Let t_r denote an arbitrary forecasting threshold value and t_r^* as the corresponding optimal threshold value. We obtain t_r^* by Eq. (5).

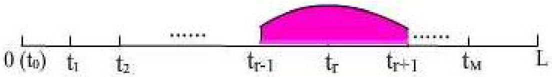


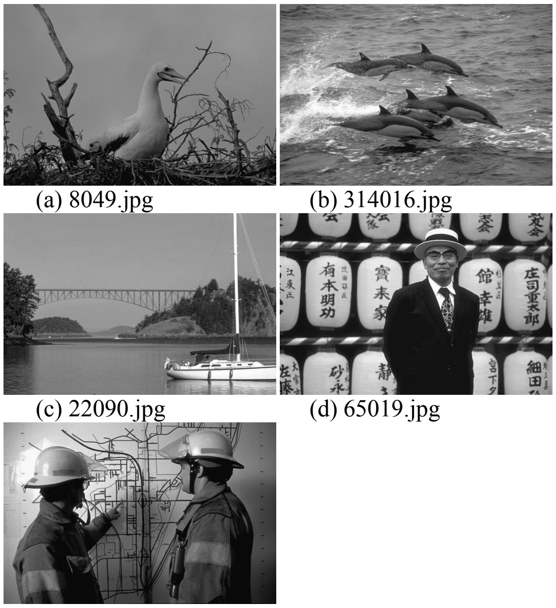
Fig. 4 The location of threshold values

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

When we obtain the first optimal threshold value t_1^*, FC is immediately upgraded to $FC'=\{t_1^*, t_2,\ldots, t_M\}$. Consequently, the second optimal threshold value t_2^* is obtained according to FC', …, until $FC'=\{t_1^*, t_2^*,\ldots, t_M^*\}$ which are the optimized threshold values.

3. EXPERIMENTAL RESULTS

The performance of the proposed AMTSSTS algorithm is evaluated by comparing its results with other algorithms. They are (1) the clustering based approach using hierarchical evolutionary algorithm (CHEA) [16], (2) the quantum-behaved PSO algorithm employing the cooperative method (CQPSO) [8], and (3) the maximum entropy-based honey bee mating optimization thresholding (MEHBMOT) method [9]. CHEA can automatically determine the threshold number, and the threshold number for the CQPSO and MEHBMOT algorithms are predetermined and set to different numbers in our experiments. These methods are then implemented on the Berkeley segmentation image database [17]. Figure 5 displays the five original images. The parameters of AMTSSTS are given as above; the parameters of CHEA and other comparative algorithms are set the same as described in their corresponding paper, except that the generation numbers in CQPSO and MEHBMOT are set to 100. The experiments are implemented on a Compaq notebook PC with 512MB RAM and 1.6GHz CPU. Some aspects are taken into consideration in the experiments: 1) The comparison of segmentation results; 2) The comparison of threshold values, *PSNR*, and computation time, and 3) The ability to conquer "the curse of dimensionality".



(a) 8049.jpg (b) 314016.jpg

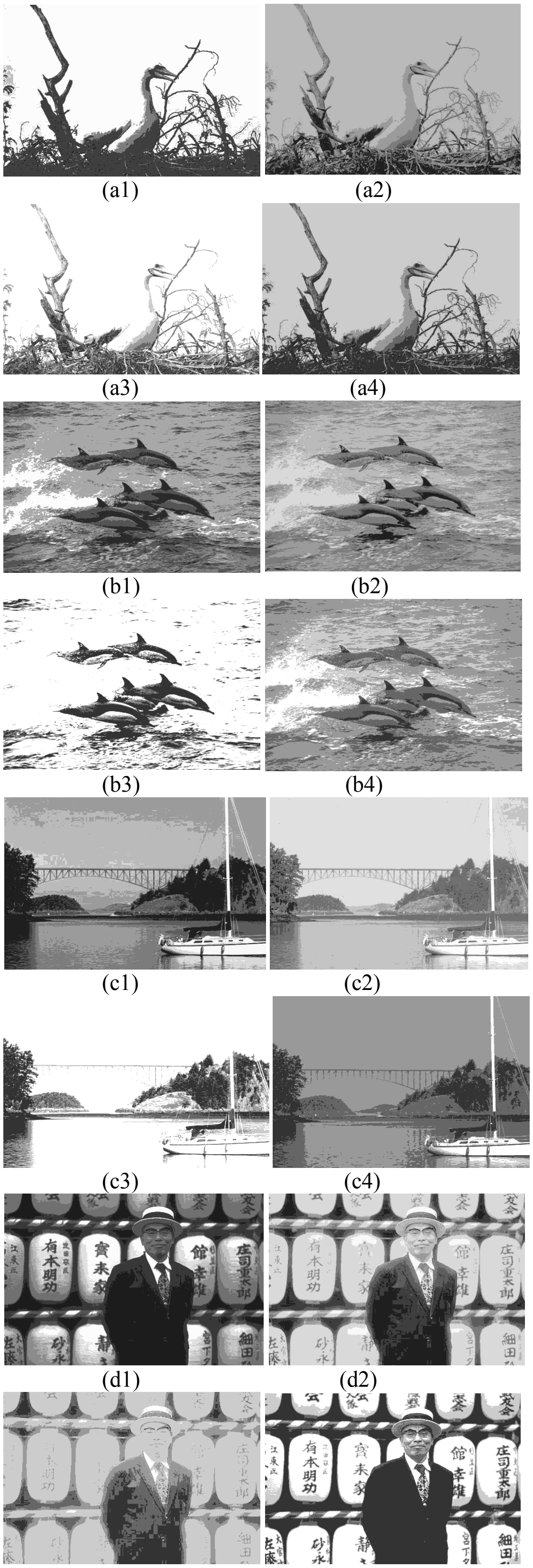
(c) 22090.jpg (d) 65019.jpg

(e) 23084.jpg

Figure 5. The five original images.

A. The comparison of segmentation results

The CHEA and our proposed algorithm automatically determine the threshold number. For the CQPSO and MEHBMOT methods, the threshold number was predetermined and set to 4.



(a1) (a2)

(a3) (a4)

(b1) (b2)

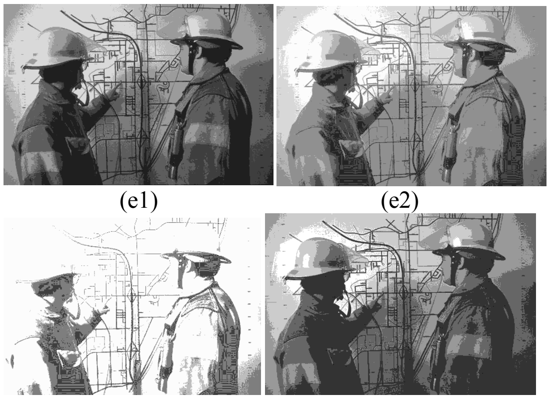
(b3) (b4)

(c1) (c2)

(c3) (c4)

(d1) (d2)

(d3) (d4)



(e1) (e2)

(e3) (e4)

Figure 6. The comparison of segmentation results: (a1)-(e1) the segmentation results of our method; (a2-e2) the segmentation results of CHEA; (a3-e3) the segmentation results of CQPSO; (a4-e4) the segmentation results of MEHBMOT.

Figure 6 displays the visual interpretation of the segmentation results. Some details about them are given in Table 1. The results show that the quality of our segmentation method is generally better than others; especially the images of a1, b1, and d1 (as shown in Figure 6). When we enlarged the segmented images in Figure 6, the following observations could be made: a) our method yielded the clearer result of 'a' (with only 3 thresholds) than that of CHEA (with 7 thresholds); b) our segmented result of 'c' was absolutely clear, except that the upper part of the sky was dim. Although the CHEA method removed the blur of the sky, its threshold number was large; and the bridge and the boat were blurred. The CQPSO method obtained even worse results than CHEA. Moreover, the MEHBMOT method could clearly separate the sky and, but the scene was a bit obscured; c) the CHEA result of 'd' was more blurred than ours, the CQPSO result was too vague to read the writing, and the MEHBMOT result was similar to ours; and d) the CHEA and CQPSO results of 'e' on the armbands were not clear and recognizable; however, the MEHBMOT result seemed much better.

B. The comparison of threshold values, PSNR, computation time

The quality of segmented images can be evaluated through the peak signal to noise ratio (*PSNR*), which is used to compare the segmentation results by using the multilevel image threshold techniques [9] [18] as follows.

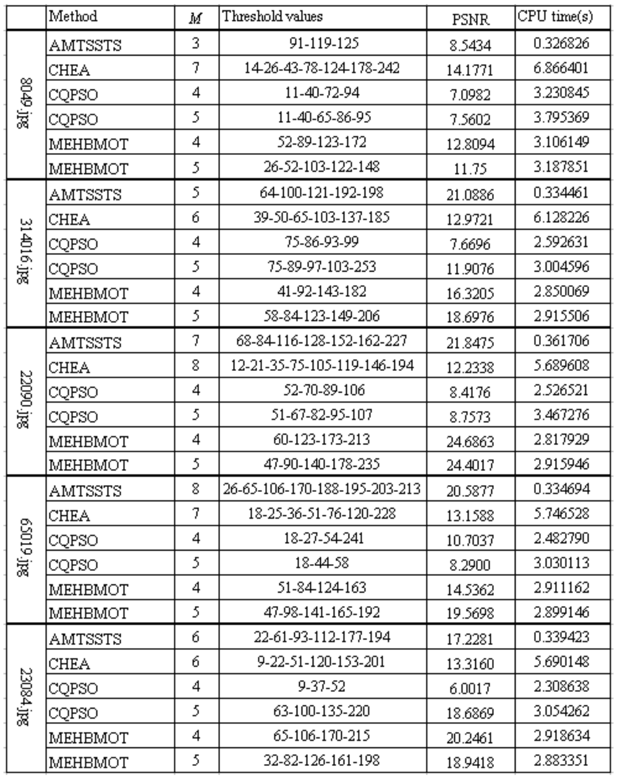
$$PSNR = 20 \times \log_{10}(255/RMSE) \qquad (6)$$

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

$$RMSE = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} (I(i,j) - I^{'}(i,j)) \Big/ (m \times n)} \qquad (7)$$

, where *I* and *I'* are the original and segmented images of size $m\times n$; respectively. A larger value of *PSNR* means the quality of the segmented image is better. Furthermore, as the threshold number increases, the *PSNR* tends to be larger. The results from the four methods over the testing images are summarized in Table 1. From the table, we can make the following observations. CHEA was the slowest of the four methods. The computation time for AMTSSTS, was insensitive to the number of thresholds and it was faster than other algorithms. The *PSNR* of images 'b1' and 'd1' (as shown in figure 6) were the largest among the four methods. The *PSNR* of image 'a1' in Figure 6 was smaller because its threshold number was only 3. Moreover, it obtained a clearer and more complete segmentation image for video image compression. The *PSNR* values of the images 'c1' and 'e1' in Figure 6 were close to the best.

Table 1. The comparison of threshold values, *PSNR*, computation time



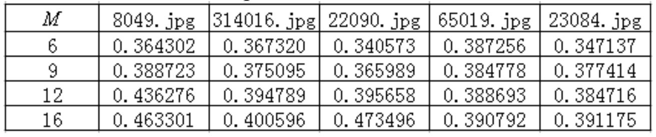
	Method	*M*	Threshold values	PSNR	CPU time(s)
8049.jpg	AMTSSTS	3	91-119-125	8.5434	0.326826
	CHEA	7	14-26-43-78-124-178-242	14.1771	6.866401
	CQPSO	4	11-40-72-94	7.0982	3.230845
	CQPSO	5	11-40-65-86-95	7.5602	3.795369
	MEHBMOT	4	52-89-123-172	12.8094	3.106149
	MEHBMOT	5	26-52-103-122-148	11.75	3.187851
314016.jpg	AMTSSTS	5	64-100-121-192-198	21.0886	0.334461
	CHEA	6	39-50-65-103-137-185	12.9721	6.128226
	CQPSO	4	75-86-93-99	7.6696	2.592631
	CQPSO	5	75-89-97-103-253	11.9076	3.004596
	MEHBMOT	4	41-92-143-182	16.3205	2.850069
	MEHBMOT	5	58-84-123-149-206	18.6976	2.915506
22090.jpg	AMTSSTS	7	68-84-116-128-152-162-227	21.8475	0.361706
	CHEA	8	12-21-35-75-105-119-146-194	12.2338	5.689608
	CQPSO	4	52-70-89-106	8.4176	2.526521
	CQPSO	5	51-67-82-95-107	8.7573	3.467276
	MEHBMOT	4	60-123-173-213	24.6863	2.817929
	MEHBMOT	5	47-90-140-178-235	24.4017	2.915946
65019.jpg	AMTSSTS	8	26-65-106-170-188-195-203-213	20.5877	0.334694
	CHEA	7	18-25-36-51-76-120-228	13.1588	5.746528
	CQPSO	4	18-27-54-241	10.7037	2.482790
	CQPSO	5	18-44-58	8.2900	3.030113
	MEHBMOT	4	51-84-124-163	14.5362	2.911162
	MEHBMOT	5	47-98-141-165-192	19.5698	2.899146
23084.jpg	AMTSSTS	6	22-61-93-112-177-194	17.2281	0.339423
	CHEA	6	9-22-51-120-153-201	13.3160	5.690148
	CQPSO	4	9-37-52	6.0017	2.308638
	CQPSO	5	63-100-135-220	18.6869	3.054262
	MEHBMOT	4	65-106-170-215	20.2461	2.918634
	MEHBMOT	5	32-82-126-161-198	18.9418	2.883351

C. The Curse of Dimensionality

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

To further verify the searching ability of our proposed method on high dimensionality, we tested each image, with small values of *rangeMean* and *rangeStd*. Table 2 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.

Table 2. The computation time with *M*=6, 9, 12, 16.



M	8049.jpg	314016.jpg	22090.jpg	65019.jpg	23084.jpg
6	0.364302	0.367320	0.340573	0.387256	0.347137
9	0.388723	0.375095	0.365989	0.384778	0.377414
12	0.436276	0.394789	0.395658	0.388693	0.384716
16	0.463301	0.400596	0.473496	0.390792	0.391175

4. CONCLUSION

In this paper, we proposed a method called AMTSSTS algorithm for automatic multilevel thresholds selection using a new local criterion function. The AMTSSTS method obtained the initial threshold number and values using the prediction method which combined stratified sampling with Tabu Search, and which were then optimized by an optimization algorithm. Our previous sections have shown that, with the help of results obtained for five benchmark images, the proposed method can rapidly converge and produce better results than several new and developed methods from recent years. In our future research, we will try to find a better method to forecast the threshold values and threshold number, with the help of statistical methods and the theory of computer vision. Furthermore, we will make a more detailed analysis of this algorithm, and apply it to complex and real-time image analysis problems, such as automatic target recognition, medical image analysis, and other applications.

6. REFERENCES

[1] S.U. Lee, S.Y. Chung, R.H. Park, "A comparative study of several global thresholding techniques for segmentation," Computer Vision Graphics and Image Processing, 1990, 52(2), pp. 171-190.

[2] M. Sezgin, B. Sankur, "Survey over image thresholding techniques and quantitative performance evaluation," Journal of Electronic Imaging, 2004, 13(1), pp. 146-165.

[3] S. Chen, D. Zhang, "Robust image segmentation using FCM with spatial constraints based on new kernel-induced distance measure," IEEE Transactions on System Man and Cybernetics B, 2004, 34(4), pp. 1907-1916.

[4] P.Y. Yin, "Multilevel minimum cross entropy threshold selection based on particle swarm optimization," Applied Mathematics and Computation, 2007, 184(2) pp. 503-513.

[5] Z.W. Ye, H.W. Chen, W. Li, J.P. Zhang, "Automatic threshold selection based on particle swarm optimization algorithm," in: International Conference on Intelligent Computation Technology and Automation, 2008, pp. 36-39.

[6] K. Hammouche, M. Diaf, P. Siarry, "A multilevel automatic thresholding method based on a genetic algorithm for a fast image segmentation. Computer Vision and Image Understanding," 2008, 109(2), pp. 163-175.

[7] W. B. Tao, H. Jin, and L. M. Liu, "Object segmentation using ant colony optimization algorithm and fuzzy entropy," Pattern Recognition Letters, 2008, 28(7), pp. 788-796.

[8] H. Gao, W. Xu, J. Sun, and Y. Tang, "Multilevel thresholding for image segmentation through an improved quantum-behaved particle

swarm algorithm," IEEE Transactions On Instrumentation And Measurement, 2010, 59(4), pp. 934-946.

[9] Ming-Huwi Horng. A multilevel image thresholding using the honey bee mating optimization. Applied Mathematics and Computation, 2010, 215(9), pp. 3302-3310.

[10] S. Bhattacharyya et al. "Multilevel image segmentation with adaptive image context based thresholding," Applied Soft Computing 2011, 11(1) , pp. 946-962.

[11] J.K. Tsotsos, S.M. Culhane, W.Y.K. Wai, Y.H. Lai, N. Davis, and F. Nuflo, "Modelling visual attention via selective tuning," Artificial Intelligence, 1995, 78, 1-2, pp. 507-545.

[12] L. Itti, C. Koch, E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," IEEE Transactions On Pattern Analysis And Machine Intelligence, 1998, 20(11), pp. 1254-1259.

[13] F. Glover, Tabu Search - Part I, ORSA Journal on Computing 1989 1: 3, 190-206.

[14] F. Glover, Tabu Search - Part II, ORSA Journal on Computing 1990 2: 1, 4-32.

[15] F. Glover, and M. Laguna. Tabu Search, 1997, Boston MA: Kluwer.

[16] C.-C. Lai, C.-Y. Chang. "A hierarchical evolutionary algorithm for automatic medical image segmentation," Expert Systems with Applications, 2009, 36(1), pp.248-259.

[17] P. Arbelaez, C. Fowlkes, D. Martin, "The Berkeley, Segmentation Dataset and Benchmark," http://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/, 2009-06-22.

[18] S. Arora, J. Acharya, A. Verma, Prasanta K. Panigrahi, "Multilevel thresholding for image segmentation through a fast statistical recursive algorithm," Pattern Recognition Letters, 2008, 29(2), pp.1 19-125.