Fast Recursive Segmentation Algorithm Based on Kapur's Entropy

Hamed Kiani[1], Reza Safabakhsh[2], Ehsan Khadangi[3]

[1,2,3] *Department of Computer Engineering and Information Technology, Amirkabir University of Technology*
Tehran, Iran.

[1]h_kiani@aut.ac.ir
[2]safa@aut.ac.ir
[3]khadangi@gmail.com

***Abstract*— Thresholding is an important operation in image analysis which is used in many applications. One of the most efficient techniques for image segmentation is the entropy-based thresholding technique. A popular entropy-based thresholding method is the Kapur thresholding method. In this method, the criterion to select a suitable threshold is the maximization of the Kapur's entropies based on gray-level histogram. Kapur's original method is very time-consuming due to the inefficient formulation of the Kapur entropy and the exhaustive search in multilevel thresholding. In order to reduce the computational time of the entropy function, a fast recurring thresholding algorithm based on two look-up tables and new form of Kapur entropy is proposed. Our analysis of the new algorithm clearly shows that it takes less computation to compute both the probability distribution of gray levels and the Kapur entropy, and that determining the Kapur entropy by accessing a look-up table based on new form of entropy is quicker then that based on performing arithmetic operations. This algorithm yields the same set of thresholds as the original Kapur method. For example, the experimental result of a five-level threshold selection in LENA image shows that the proposed algorithm can reduce the processing time from more than two hours by the conventional Kapur method to less then 170 seconds.**

***Keywords*— Image thresholding, Image segmentation, Kapur entropy, look-up table, Multilevel thresholding, Recursive algorithm.**

I. Introduction

Thresholding is an important technique for image segmentation that tries to identify and extract a target from its background on the basis of the distribution of gray levels or texture in the image. Most thresholding techniques are based on the statistics of the one-dimensional (1D) histogram of gray levels or on the two-dimensional (2D) co-occurrence matrix of an image. Many 1D thresholding methods have been investigated [1-6]. Locating thresholds can be achieved through parametric or nonparametric approaches [4, 9]. In parametric approaches, the gray level distribution of an object class leads to finding the thresholds. For instance, in Wang and Haralick's study [4], the pixels of an image are first classified as either edge or non-edge pixels. According to their local neighborhoods, edge pixels are then classified as being relatively dark or relatively bright. Next, one histogram is obtained for those edge pixels which are dark and another for those edge pixels which are bright. The highest peaks of these two histograms are chosen as the thresholds. Moment preserving thresholding is a parametric method which segments the image based on the condition that the thresholded image has the same moments as the original image [3]. In nonparametric approaches, the thresholds are obtained in an optimal manner according to some criteria. For instance, Otsu's method chooses the optimal thresholds by maximizing the between-class variance with an exhaustive search [2]. In Pun's method [6], as modified by Kapur et al. [1], the picture threshold is found by maximizing the entropy of the histogram of gray levels of the resulting classes. Other 1D thresholding techniques extend from bi-level threshold selection to multilevel threshold selection [2,4-5]. In contrast to 1D thresholding methods, 2D methods essentially do image segmentation by using spatial information in an image [8-10]. Kirby and Rosenfeld proposed a 2D thresholding method that simultaneously considers both the pixel gray level and the local statistics of its neighboring pixels [8]. One particular 2D method is entropic thresholding, which makes use of spatial entropy to find the optimal thresholds. Abutaleb [9], and Pal and Pal [10] proposed that optimal thresholds can be selected by maximizing the sum of the posterior entropies of two classes.

Kapur's method uses an exhaustive search to evaluate the criterion for maximizing the between-class entropy. As the number in classes of an image increases, Kapur's method takes too much time to be practical for multilevel threshold selection. To determine the 1D threshold of an image efficiently, we propose a recursive form of Kapur's original method to decrease the computation complexity of this algorithm. Then, a look-up table is derived from the recursive formulations in order to find the required parameters quickly. Compared to the conventional Kapur's method, the proposed algorithm can increase the speed of computation by 70 times in five-level thresholding based on experimental results. In Section 2, the Kapur's method for image thresholding is briefly reviewed. In Section 3, a fast 1D multilevel thresholding algorithm based on the recursive forms of parameters and the look-up tables are described in detail. Section 4 shows the experimental results. Finally, Section 5 gives a brief conclusion.

II. KAPUR'S METHOD FOR IMAGE SEGMENTATION

An image is a 2D grayscale intensity function, containing N pixels with gray levels from 0 to L-1. If the number of pixels with gray level *i* is denoted by *h (i)*, then the probability of gray level *i* in the image is defined as

$$p_i = h(i) / N \quad (1)$$

In the case of bi-level thresholding of an image, the pixels are divided into two classes, C_0 with gray levels [$0,...,t-1$] and C_1 with gray levels [$t,...,L-1$], where t is the threshold value and C_0 and C_1 normally correspond to foreground (objects) and background, respectively.

Using discriminate analysis, Kapur [1] defined the between-class entropy of the thresholded image as

$$f_1(t) = H(0,t) + H(t,L) \quad (2)$$

for bi-level thresholding, the optimal threshold, t^*, is chosen so that the between-class entropy $f_1(t)$ is maximized [1]; that is,

$$t^* = \underset{0 \le t \le L-1}{Arg \;\; Max} \;\; \{f_1(t)\} \quad (3)$$

and verified that

$$H(0,t) = -\sum_{i=0}^{t-1} \frac{p_i}{w_0} \ln \frac{p_i}{w_0}, \qquad w_0 = \sum_{i=0}^{t-1} p_i,$$
$$H(t,L) = -\sum_{i=t}^{L-1} \frac{p_i}{w_1} \ln \frac{p_i}{w_1}, \qquad w_1 = \sum_{i=t}^{L-1} p_i \quad (4)$$

The w_0 and w_1 in Eq. (4) are regarded as the zeroth-order cumulative moment of the C_0 and C_1 classes.

This formulation can be easily extended to multilevel thresholding of an image [1]. Assuming that there are M-1 thresholds, $\{t_1, t_2, ..., t_{M-1}\}$, which divide the original image into M classes: C_0 for [$0,...,t_1-1$], C_1 for [$t_1, ..., t_2-1$], … , C_{i-1} for [$t_{i-1}, ..., t_i-1$], … and C_{M-1} for [$t_{M-1}, ..., L-1$], the optimal thresholds $\{t_1^*, t_2^*, ..., t_{M-1}^*\}$ are chosen by maximizing $f_{M-1}(t_1, t_2, ..., t_{M-1})$ as follows [1]:

$$\{t_1^*, t_2^*, ..., t_{M-1}^*\} = \underset{0 \le t_1, ..., t_{M-1} \le L-1}{Arg \;\; Max} \;\; \{f_{M-1}(t_1, t_2, ..., t_{M-1})\} \quad (5)$$

where

$$f_{M-1}(t_1, t_2, ..., t_{M-1}) = H(0,t_1) + H(t_1,t_2) + ...$$
$$+ H(t_{M-2}, t_{M-1}) + H(t_{M-1}, L-1) = \sum_{i=0}^{M-1} H(t_i, t_{i+1}) \quad (6)$$

and $t_M = L-1, \;\; t_0 = 0$.

In this case $H(t_i, t_{i+1})$ is the Kapur's entropy of $[t_i, ..., t_{i-1}]$ interval which is calculated as

$$H(t_i, t_{i+1}) = -\sum_{j=t_i}^{t_{i+1}-1} \frac{p_j}{w_i} \ln \frac{p_j}{w_i} \quad (7)$$

where p_i is the probability of gray level i in the image and w_i is the zeroth-order cumulative moment of the *ith* class C_i defied as

$$w_i = \sum_{j=t_i}^{t_{i+1}-1} p_j \quad (8)$$

According to the criteria of Eq. (5), to find the optimal thresholds, the search ranges for the maximal f_{M-1} are $0 \le t_1 \le L-M+1$, $t_1+1 \le t_2 \le L-M+2$,… and $t_{M-2} \le t_{M-1} < L$, so based on multiple rule of different combinations, this exhaustive search involves $(L-M+1)^{M-1}$ possible combinations.

In original method by Kapur, finding optimal thresholds is bounded by $O(M(L-M)^M)$; we will prove this as follow. According to the summation of Eq. (8), calculating each w_i value can be done in $O(L-M)$ time and because for each possible combination $m-1$ different values of w_i must be calculated, therefore the total calculation of w_i values for each possible combination can be performed in $O(M(L-M))$ time. Considering the assumption that there are $(L-M)^{M-1}$ possible combinations to obtain $M-1$ optimal thresholds, as mentioned above, the Kapur method takes totally $O(M(L-M)^M)$ time to calculate all w_i values for all possible combinations in the algorithm. Similarly, we can prove that calculating the $H(t_i, t_{i+1})$ values by Eqs. (7,8) and finding final optimal thresholds by Eqs. (5,6) take respectively $O(M(L-M)^M)$ and $O(M(L-M)^{M-1})$ time to be calculated in all possible combinations. From the above description, it is clear that the final time complexity of Kapur original method, the total time for performing Eqs. (5-8), is bounded by $O(M(L-M)^M)$.

III. THE NEW METHOD

Optimal thresholds by the Kapur's original method can be found in $O(M(L-M)^M)$ time, making the method inefficient particularly when the number of thresholds increases.

In order to reduce computation complexity of the Kapur's original method, two major problems should be solved: 1) too much iteration due to the summation in Eqs. (7,8), and 2) the large number of pre-computing of $H(t_i, t_{i+1})$ and w_i parameters in Eqs. (6,7). To solve these problems for w_i, we first rewrite Eq. (8) in recursive form as:

$$w(1, t_i) = \sum_{j=1}^{t_i-1} p_j \quad (9)$$

$$w(1, t_i+1) = \sum_{j=1}^{t_i} p_j = \sum_{j=1}^{t_i-1} p_j + p_{t_i}$$
$$= w(1, t_i) + p_{t_i} \quad (10)$$

Where p_j is the probability of the gray level j in a L level image where $j \in [1, ..., L]$.

From Eqs. (9, 10), it follows that

$$w(t_{i-1},t_i)=\sum_{j=t_{i-1}}^{t_i-1} p_j = \sum_{j=1}^{t_i-1} p_j - \sum_{j=1}^{t_{i-1}-1} p_j = w(1,t_i)-w(1,t_{i-1}) \quad (11)$$

Note that in Eqs. (10) and (11) t_1 and t_{M-1} are defined as $t_1=1$ and $t_{M-1}=L$ in M level thresholding and $w(U,V)=0$ if $U \geq V$. For all possible intensities from u to v, the u-v interval zeroth-order moment $w(u,v)$ can be calculated recursively for once and stored in the look-up table W, as shown in Figure 1. The values in the first rows of the tables are determined by using the recursive forms of $w(1,t_i)$ and $w(1,t_i+1)$ given in Eqs. (9) and (11). Then the values in the remaining rows are determined from Eqs. (11-12) and the first row values. We can see from Tables 1 that the values of w_i , $w(t_{i-1},t_i)$, can be obtained directly on $O(1)$ time.

Unlike Eq. (8) which is potentially recursive, defining Eq. (7) in recursive form is not possible. Therefore, we first rewrite the Shannon entropy in recursive form and then define the new form of Eq. (7) based on the recursive form of Shannon entropy.

The Shannon entropy of $[t_i-1,t_i]$ interval is defined as

$$h_s(t_{i-1},t_i)=-\sum_{j=t_{i-1}}^{t_i-1} p_j \ln(p_j) \quad (12)$$

So the recursive form of the Shannon entropy can be written as

$$h_s(1,t_i)=-\sum_{j=1}^{t_i-1} p_j \ln(p_j) \quad (13)$$

and

$$h_s(1,t_i+1)=-\sum_{j=1}^{t_i} p_j \ln(p_j) = -\sum_{j=1}^{t_i-1} p_j \ln(p_j) - p_{t_i} \ln(p_{t_i}) = h_s(1,t_i) - p_{t_i} \ln(p_{t_i}) \quad (14)$$

where p_j is the probability of the gray level j in image and $j \in [1,...,L]$.

From Eqs. (13) and (14), it can be concluded that

$$h_s(t_{i-1},t_i)=-\sum_{j=t_{i-1}}^{t_i-1} p_j \ln(p_j) = -\sum_{j=1}^{t_i-1} p_j \ln(p_j) + \sum_{j=1}^{t_{i-1}-1} p_j \ln(p_j) = h_s(1,t_i)-h_s(1,t_{i-1}) \quad (15)$$

where $h_s(U,V)=0$ if $U \geq V$.

For all possible gray levels from u to v, the u-v interval Shannon entropy $h_s(U,V)$ can be calculated recursively once and stored in look-up table Hs, as shown in Figure 2. The values in the first rows of the tables are determined by using the recursive forms of Eqs. (13) and (14) then the values in remaining rows are determined from Eqs. (14) and (15) and the first row values. Now, using the recursive forms of (11) and (15), we can define the new form of $H(t_i,t_{i+1})$ without iterative summation as:

$$\begin{aligned} H(t_{i-1},t_i) &= -\sum_{j=t_{i-1}}^{t_i-1} \frac{p_j}{w(t_{i-1},t_i)} \ln(\frac{p_j}{w(t_{i-1},t_i)}) \\ &= -\frac{1}{w(t_{i-1},t_i)}[\sum_{j=t_{i-1}}^{t_i-1} p_j(\ln(p_j) - \ln(w(t_{i-1},t_i)))] \\ &= -\frac{1}{w(t_{i-1},t_i)}[\sum_{j=t_{i-1}}^{t_i-1} p_j \ln(p_j) - \sum_{j=t_{i-1}}^{t_i-1} p_j \ln(w(t_{i-1},t_i))] \\ &= -\frac{1}{w(t_{i-1},t_i)}[h_s(t_{i-1},t_i) - w(t_{i-1},t_i)\ln(w(t_{i-1},t_i))] \end{aligned} \quad (16)$$

U \ V	1	2	...	i	...	L
1	W(1,1)	W(1,2)	...	W(1,i)	...	W(1,L)
2	W(2,1)	W(2,2)		W(2,i)	...	W(2,L)
...	...	...	...	...	...	...
i	W(i,1)	W(i,2)		W(i,i)	...	W(i,L)
...	...	...	...	...	...	...
L	W(L,1)	W(L,2)	...	W(L,i)	...	W(L,L)

Fig. 1. Look-up table W for interval U-V

U \ V	1	2	...	i	...	L
1	hs(1,1)	hs(1,2)	...	hs(1,i)	...	hs(1,L)
2	hs(2,1)	hs(2,2)	...	hs(2,i)	...	hs(2,L)
...	...	...	...	...	...	...
i	hs(i,1)	hs(i,2)		hs(i,i)		hs(i,L)
...	...	...	...	...	...	...
L	hs(L,1)	hs(L,2)	...	hs(L,i)	...	hs(L,L)

Fig. 2. Look-up table Hs for interval U-V

In the original Kapur method, the computations of w_i and $H(t_i,t_{i+1})$ are performed for each threshold $\{t_1, t_2,..., t_{M-1}\}$ in $O(M(L-M))$ time. Therefore, the total computation complexity of these parameters is bounded by $O(M(L-M)^M)$ for all possible combinations.

On the other hand, in our method by using look-up Table W and Hs, the required time for all w_i and $H(t_i,t_{i+1})$ values are significantly reduced and bounded by $O(L^2)$ that is the required time to calculate and save the W and Hs tables. This occurs just once in the segmentation algorithm.

Moreover, according to Eq. (16), each value of $H(t_{i-1}, t_i)$ can be found in *O(1)* time, because of W and Hs tables, so the computation complexity to process each possible combination is equal to *O(m)*, Eq. (6). Consequently, the optimal thresholds can be find in $O(M(L-M)^{M-1})$ time because there are $(L-M)^{M-1}$ possible combinations and each possible combination take *O(m)* time to be processed.

The proposed method can reduce the upper bound for obtaining optimal thresholds by Eq. (5) from $O(M(L-M)^{M})$ in the original Kapur method to $O(M(L-M)^{M-1})$ time. Furthermore, finding partial values, w_i and $H(t_i, t_{i+1})$, can be done in $O(L^2)$ time in our method rather than $O(M(L-M)^{M-1})$ in the Kapur's method because of omitting the iterative pre-computing and using look-up tables and recursive formulations that makes this method faster. The computation times for Kapur's original method and the new method are listed in Table 5 in detail.

IV. EXPERIMENTAL RESULTS

For evaluating the performance of the proposed method versus the conventional Kapur's method, the four images (Lena, Baboon, Peppers and House) shown in Fig. 3 are chosen. These images have 256 × 256 pixels and 256 gray levels. Fig. 4 shows their corresponding histograms. To implement Kapur's method, we programmed Eqs. (5-8), while our method was implemented using Eqs. (5-6) and (9-16). The algorithms are coded in MATLAB version 7.1 and are run on a 2 GHz Pentium 4 personal computer with 1G B RAM, in Microsoft Windows XP Operating system.

Because time is measured in seconds, the true difference between our method and Kapur's method is indistinguishable for bi-level and tri-level cases. However, for the four-level selection, using the recursive forms of partial parameters reduces the processing time from more than 123 seconds to less than 2 seconds in the Peppers image.

In addition, decreasing computation time from 8619 to 169 seconds in Lenna image is considerable. The threshold selection values and computation time for the tested images are listed in Table 5.

Fig. 4 to 7 shows some of resulting segmented images by the bi-level, tri-level, four-level and five-level thresholding.

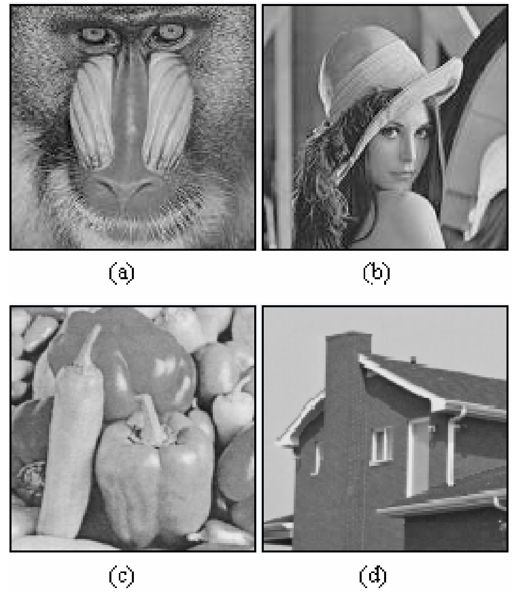

Fig. 3. Test images: (a) Baboon, (b) Lena, (c) Peppers and (d) House

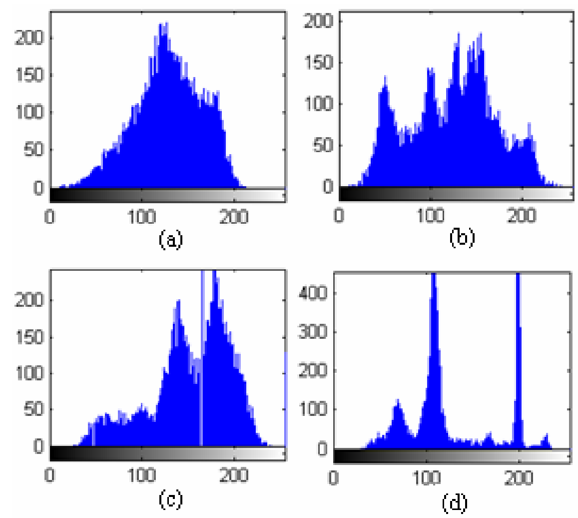

Fig. 4. Histograms of test images: (a) His. of Baboon image, (b) His. of Lena image, (c) His. of Peppers image and (d) His. of House image.

TABLE 1. THE COMPUTATION TIMES FOR KAPUR'S ORIGINAL METHOD AND THE NEW METHOD

	Single combination		***(L-M)$^{M-1}$* possible combinations**	
	Proposed Method	**Kapur's Method**	**Proposed Method**	**Kapur's Method**
w_i	$O(L^2)$	$O(M(L-M))$	$O(L^2)$	$O(M(L-M)^M)$
$H(t_i,t_{i-1})$	$O(L^2)$	$O(M(L-M))$	$O(L^2)$	$O(M(L-M)^M)$
Find $\{t_1^*, t_2^*, \ldots, t_{M-1}^*\}$	$O(M)$	$O(M)$	$O(M(L-M)^{M-1})$	$O(M(L-M)^{M-1})$
Total computation	$O(M)+2O(L^2)$	$O(M)+2O(M(L-M))$	$2O(L^2)+O(M(L-M)^{M-1})$	$2O(M(L-M)^M)+O(M(L-M)^{M-1})$
Upper bound	$O(L^2)$	$O(M(L-M))$	$O(M(L-M)^{M-1})$	$O(M(L-M)^M)$

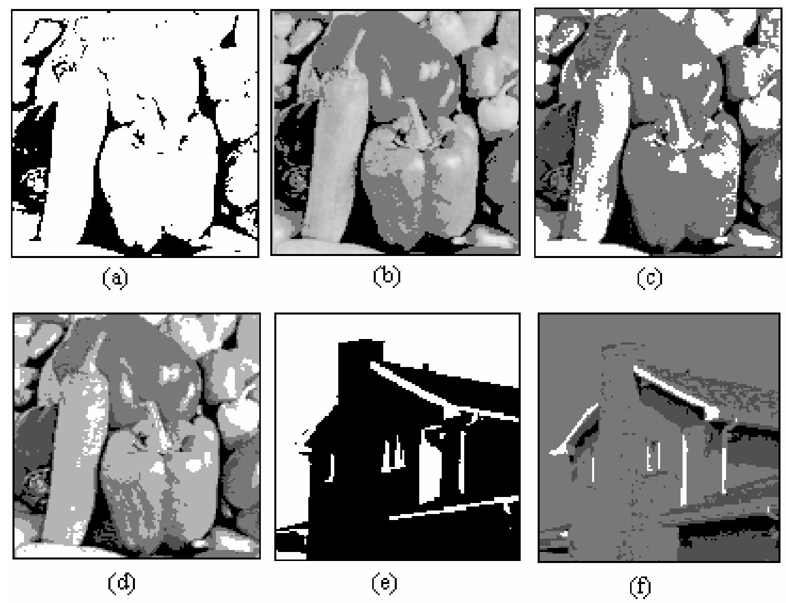

Fig. 5. Thresholded images. Pepper image: (a) bi-level thresholding (T=112), (b) tri-level thresholding (T=111,162), (c) four-level thresholding (T=88,125,184), (d) five-level thresholding (T=77, 120, 160, 200). House image: (e) bi-level thresholding (T=144), (f) four-level thresholding (T=86, 130, 172).

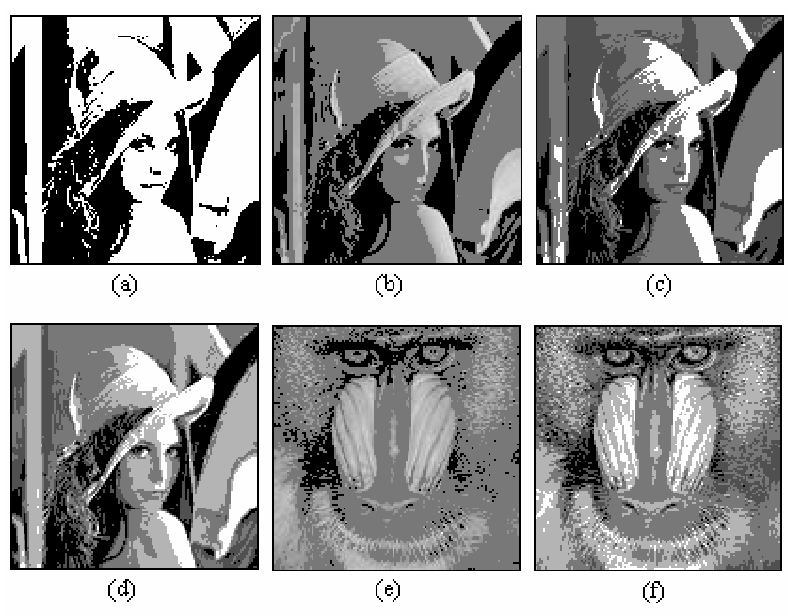

Fig. 6. Thresholded images. Lena image: (a) bi-level thresholding (T=119), (b) tri-level thresholding (T=96, 167), (c) four-level thresholding (T=77, 124, 176), (d) five-level thresholding (T=64, 98,139,182). Baboon image: (e) tri-level thresholding (T=72, 143), (f) five-level thresholding (T=45, 80, 118, 160).

TABLE 2 . THRESHOLDS AND COMPUTATION TIMES FOR THE TEST IMAGES.

Images	Thresholds				Computation Time (In second)							
					Kapur's Original method				Proposed Method			
	2	3	4	5	2	3	4	5	2	3	4	5
Lena	119	96 167	77 124 176	64 98 139 182	<1	2	153	8619	<1	<1	2.5	169
Peppers	122	111 162	88 125 184	77 120 160 200	<1	<1	123	8425	<1	<1	2	137
Baboon	100	79 143	58 104 153	45 80 118 160	<1	2	116	7440	<1	<1	2	128
House	114	82 156	86 130 172	61 89 133 175	<1	2	148	8469	<1	<1	2	156

V. CONCLUSION

In this paper, a fast and efficient recursive algorithm is developed for one-dimensional multilevel Kapur thresholding.

The formulas for some parameters for the between-class entropy are written in recursive form, which reduces the complexity of computation significantly. Moreover, determining these parameters by indexing in look-up tables is more efficient than that by applying arithmetic operations.

The proposed method can reduce the upper bound for obtaining optimal thresholds from $O(M(L-M)^M)$ in the original Kapur method to $O(M(L-M)^{M-1})$ time.

Furthermore, finding partial values, w_i and $H(t_i, t_{i+1})$, can be done in $O(L^2)$ time in our method rather than $O(M(L-M)^{M-1})$ in the Kapur's. For five-level threshold selection, the experimental results show that the processing time of the recursive formulation is less than 169 seconds, while the of conventional Kapur's method needs more than 8619 seconds.

REFERENCES

[1] J. N. Kapur, P. K. Sahoo, A. K. C. Wong, "A new method for gray-level picture thresholding using the entropy of the histogram,", Computer Vision Graphics Image Processing, 29, pp. 273-285, 1985.

[2] N. Otsu, "A threshold selection method from gray-level histogram," IEEE Transactions on System Man Cybernetics, Vol. SMC-9, No. 1, 1979, pp. 62-66.

[3] W. H. Tsai, "Moment-preserving thresholding: a new approach," Computer Vision, Graphics, and Image Processing, Vol. 29, 1985, pp. 377-393.

[4] S. Wang and R. Haralick, "Automatic multithreshold selection," Computer Vision, Graphics, and Image Processing, Vol. 25, 1984, pp. 46-67.

[5] P. K. Sahoo, S. Soltani, A. K. C. Wong, and Y. Chen, "A survey of thresholding techniques," Computer Vision Graphics Image Processing, Vol. 41, 1988, pp. 233-260.

[6] T. Pun, "A new method for gray-level picture thresholding using the entropy of the histogram," Signal Processing, Vol. 2, 1980, pp. 223-237.

[7] J. Gong, L. Li, and W. Chen, "Fast recursive algorithms for two-dimensional thresholding," Pattern Recognition, Vol. 31, No. 3, 1998, pp. 295-300.

[8] R. L. Kirby and A. Rosenfeld, "A note on the use of (gray, local average gray level) space as an aid in thresholding selection," IEEE Transactions on System Man Cyber- netics Vol. SMC-9, No. 12, 1979, pp. 860-864.

[9] A. S. Abutaleb, "Automatic thresholding of gray-level pictures using two-entropy," Computer Vision Graphics Image Processing, Vol. 47, 1989, pp. 22-32.

[10] N. R. Pal and S. K. Pal, "Entropic thresholding," Signal Processing, Vol. 16, 1989, pp. 97-108.