Pattern Recognition Letters 13 (1992) 245–252
North-Holland
April 1992

A fast histogram-clustering approach for multi-level thresholding

Du-Ming Tsai and Ying-Hsiung Chen

Department of Industrial Engineering, Yuan-Ze Institute of Technology, Tao-Yuan, Taiwan, ROC

Received 6 August 1991

Abstract

Tsai, D.-M. and Y.-H. Chen, A fast histogram-clustering approach for multi-level thresholding, Pattern Recognition Letters 13 (1992) 245–252.

Traditional thresholding methods are computationally expensive in multi-level thresholding. In this paper a histogram-based approach, called hill-clustering, is developed. It is computationally fast and efficient. The variation of computational time of this proposed approach is insignificant regardless of the number of thresholds to be identified.

Keywords. Multi-level thresholding, histogram-clustering.

1. Introduction

Thresholding has been a popular tool used in image segmentation. It is useful in separating objects from background, or discriminating objects from objects that have distinct gray levels. Sahoo et al. [5] have presented a thorough survey of a variety of thresholding techniques. Among those techniques, histogram (gray-level distribution) is the most common method to detect the threshold. Abutaleb [1] has classified two main approaches, parametric and nonparametric, to locate the thresholds. In the parametric approach [6,8], each group is assumed to have the probability density function of a gray-level distribution; it is usually assumed to be a Gaussian distribution. One then attempts to find an estimate of the parameters of the distribution that will best fit the given histogram data. The result is typically a nonlinear optimization problem that is computationally expensive. In the nonparametric approach, one is to find the threshold value that separates the two gray-level segments in an optimum manner according to some criterion such as variance [3,4], or entropy [1,2]. The nonparametric approaches have been proven to be robuster and more efficient than the parametric techniques.

In bilevel thresholding, the nonparametric approach is computationally fast for time-critical applications. However, its computational time is in an order of magnitude increase when the number of thresholds to be identified increases. The traditional nonparametric approach becomes untolerable in multi-level thresholding.

In this paper, a fast nonparametric histogram-based approach is developed. It assumes each desired segment of an image can be represented by a hill in gray-level histogram. By varying the cell size (interval) of histogram bars, the hill can be smoothed out, and the number of peaks identified in the histogram will be recorded. The desired number of peaks is the only user-specified parameter to this approach. When the number of peaks reaches some user-specified value, the bottom of valley of the hills will be identified as the threshold value. The proposed approach is computationally simple and efficient, and most important, the variation of computational time is insignificant

0167-8655/92/$05.00

regardless of the number of thresholds to be identified. Actually, the computational time will decrease while the desired number of thresholds increases. Therefore, this proposed approach is appropriate for multi-level thresholding, especially in time-critical applications.

In Section 2, the proposed approach is described. We call it hill-clustering. The result of applying the proposed approach to a number of images is presented in Section 3. The performance of the hill-clustering approach is compared to the variance-based, and the entropy-based approaches. Summary and conclusions are given in Section 4.

2. The hill-clustering approach

The proposed approach assumes each group in a gray-level histogram can be represented by a hill. Also, the gray-level histogram is assumed to have one valley between two adjacent hills. The challenge is to locate the bottom of the valley that best separates the two groups. A hill in a gray-level histogram is defined as follows. Consider the gray-level histogram of an image as shown in Figure 1. The histogram in Figure 1 consists of 15 cells. Each cell contains an equal interval of distinct gray levels. Let f_i represent the frequency count of cell i. Compare f_i with its two adjacent neighbors, f_{i-1} and f_{i+1}, and make an arrow pointing to the cell that has larger frequency, i.e.,

If $(f_{i+1} > f_{i-1})$ and $(f_{i+1} \geqslant f_i)$,
then put an arrow pointing to the right.

If $(f_{i-1} > f_{i+1})$ and $(f_{i-1} \geqslant f_i)$,
then put an arrow pointing to the left.

Otherwise, do nothing.

Observing the arrow directions in Figure 1, we find that the first hill climbs up from cell 1 to cell 4 (consecutive right-arrows), and then goes down from cell 4 to cell 8 (consecutive left-arrows). Similarly, the second hill ascends from cell 9 to cell 12, and then descends from cell 12 to cell 15. Cells 4 and 12 are identified as peaks since they are pointed by both the left- and the right-arrow. The bottom of valley is located between cells 8 and 9 since they have opposite arrow directions; one points to the left and the other points to the right. The gray-level threshold can be set at valley point between two adjacent hills to form gray-level segmentation. Therefore, the arrow directions can be employed to cluster histogram bars to represent a hill and detect its peak and valley. Note that a hill can only have a peak and it is pointed by both the left- and the right-arrow.

In addition, consider the gray-level histogram shown in Figure 2. The frequency of cell 3 is larger than its adjacent neighbors, cells 2 and 4. However, cell 4 points to cell 5 and establishes a right-arrow since $f_5 > f_3$. Therefore, cell 3 is not identified as a peak. The right-arrow link is disconnected between cells 3 and 4, yet the arrow directions are consistent from cell 1 to cell 6. In such case, cell 1 through cell 10 are still clustered as a single hill, and cell 6 is the only peak.

To use the proposed approach, the user needs to specify the desired number of peaks (segments). Due to the fluctuation of gray-level histogram, the hill-clustering approach may generate more peaks than actually needed. This can be adjusted by automatically increasing the cell size (interval of a bar) and smoothing out the local peaks of a hill. The cell size is acted as a detection parameter; it

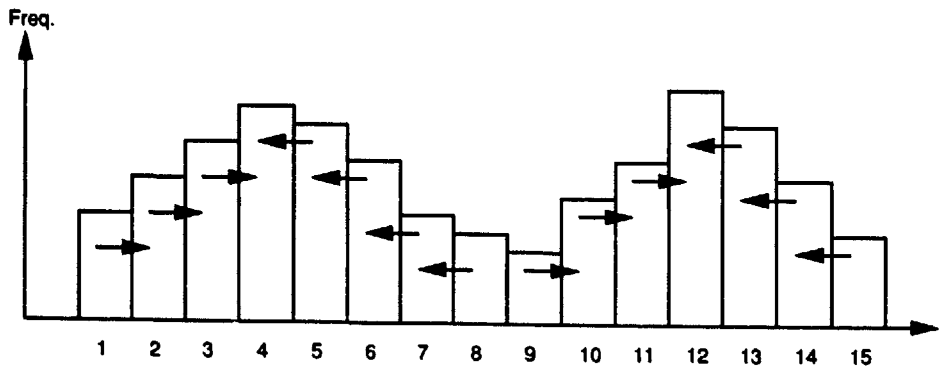


Figure 1. A histogram with two hills.

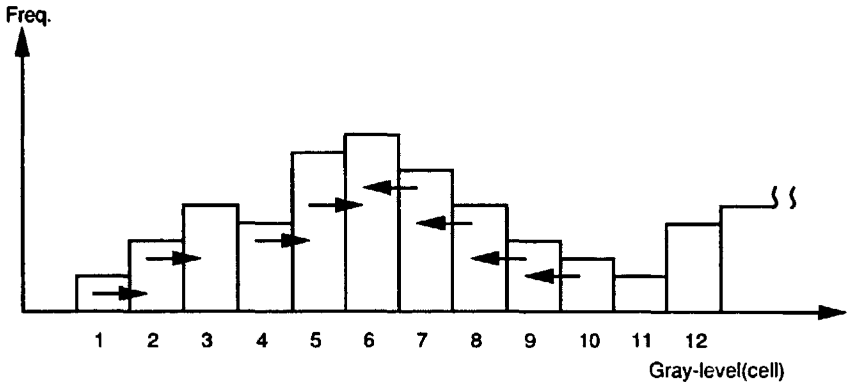


Figure 2. A hill with disconnected link.

controls the sensitivity of the peak detection and, hence, the number of detected peaks.

There are two ways to calculate the threshold value from the bottom of valley. Let the bottom of valley be located between cells i and j, where $f_i \neq 0$, $f_j \neq 0$ and $j > i$. Let a_i and b_j represent the left-end gray level of cell i and the right-end gray level of cell j, respectively. The threshold value can be approximately estimated as the average gray level of these two cells, i.e.,

$$\text{threshold} = a_i + (b_j - a_i)/2.$$

When the final cell size selected is large, the above estimated threshold value could be rough. In this case, the hill-clustering procedure can be employed again to identify the accurate location of the valley for gray levels in the range of a_i and b_j. Hence, for large cell size and two adjacent cells (i.e., $j = i + 1$) that have the opposite arrow-directions, restart the cell size from 1, and increase the cell size by 1 in each hill-clustering iteration until there is only one valley located between gray levels a_i and b_j.

For instance, the threshold value in Figure 1 is somewhere between cells 8 and 9. By using the averaging method, the threshold value is determined as follows:

$$\text{threshold} = a_8 + (b_9 - a_8)/2.$$

Or, we can perform the hill-clustering procedure again for the gray levels in the range of a_8 and b_9, by starting the cell size from 1. Suppose the histogram of gray levels between a_8 and b_9 is as shown in Figure 3. The valley is now much accurately located within cells 5′ and 6′ because they have opposite arrow-directions. The threshold is, therefore, determined to be

$$\text{threshold} = a_{5'} + (b_{6'} - a_{5'})/2.$$

Based on the above discussion, the detailed procedure of the hill-clustering approach is described as follows.

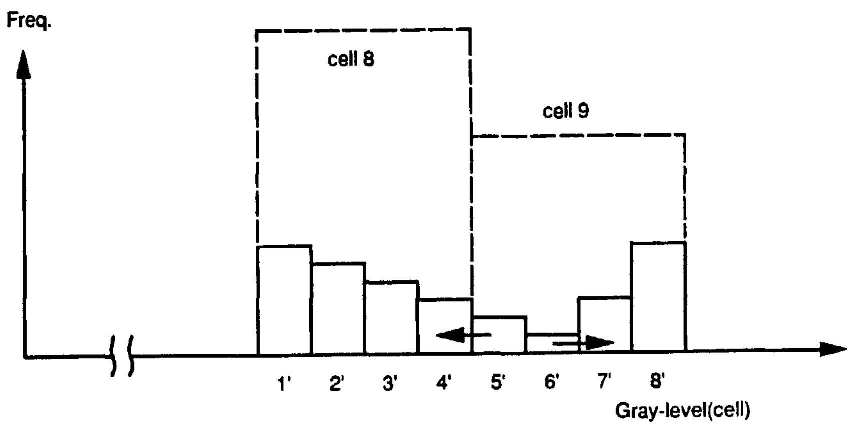


Figure 3. Valley location under smaller cell size.

The algorithm

Denote

$g(x,y)$ = the gray level at pixel location (x,y),
N = total number of gray levels,
$[t]$ = the nearest integer to t such that $[t] \geqslant t$.

Step 1. Assign $c=1$.

Step 2. For a given cell size c,
compute the frequency count of gray levels within each cell;
let f_k = the frequency of cell k containing gray level $g(x,y)$ such that

$$(k-1)c \leqslant g(x,y) < kc, \quad k=1,2,\ldots,[N/c].$$

Step 3. Establish the arrow-direction.
Let d_k represent the arrow-direction at cell k. Then

$$d_k = \begin{cases} +1, & \text{if } (f_{k-1} > f_{k+1}) \wedge (f_{k-1} \geqslant f_k) \wedge f_k \neq 0 \quad \text{(the left-arrow)}, \\ -1, & \text{if } (f_{k+1} > f_{k-1}) \wedge (f_{k+1} \geqslant f_k) \wedge f_k \neq 0 \quad \text{(the right-arrow)}, \\ 0, & \text{otherwise} \quad \text{(do-nothing)}. \end{cases}$$

In the case of tie (i.e., $f_{k-1} = f_{k+1}$), set $d_k = d_{k-1}$.

Step 4. Identify the peak.
(a) If $(d_k=0) \wedge (d_{k-1}=-1) \wedge (d_{k+1}=+1)$, then cell k is the peak of a hill. Or,
(b) If $(d_k=-1) \wedge (d_{k+1}=+1)$, then a peak is also identified between cells k and $k+1$.
This occurs only when two or more cells have equal frequency counts.
Repeat Step 4 to identify all peaks in the gray-level histogram.

Step 5. If number of peaks identified in Step 4 equals the user-specified number of peaks, then go to Step 6. Otherwise, increase the cell size. Set $c=c+1$ and go to Step 2.

Step 6. Locate the valley.
If $(d_k=+1) \wedge (d_j=-1)$ where $j=\min\{i \mid f_i \neq 0, i>k\}$, then a valley is located between cells k and j.
Repeat Step 6 to locate all valleys.

Step 7. Determine the threshold value.
Let a_k be the left-end gray level of cell k;
b_j be the right-end gray level of cell j.
Cells k and j are those determined in Step 6.
(a) By the averaging method,

threshold $= a_i + (b_j - a_i)/2$.

Or,
(b) by the hill-clustering method (only if $j=k+1$), repeat Steps 1 through 7(a) above (skip Steps 4 and 5) for gray levels between a_k and b_j until there is only one valley located.

3. Experimental results

The performance of the proposed hill-clustering approach is compared to two most popular automatic thresholding methods used in commercial image processing software [7]. Both are non-parametric histogram-based approaches; one is based on between-class variance, and the other is based on entropy criterion. These two approaches are described briefly as follows.

Let p_i be the probability of gray level i

$$p_i = \frac{\text{frequency of gray level } i}{\text{total number of pixels}}, \quad a \leqslant i \leqslant b.$$

Variance-based approach

$$w_0 = \sum p_i, \quad m_0 = -\sum i \frac{p_i}{w_0} \quad \text{for } a \leqslant i \leqslant T_1,$$

$$w_1 = \sum p_i, \quad m_1 = -\sum i \frac{p_i}{w_1} \quad \text{for } T_1 < i \leqslant T_2,$$

$$w_2 = \sum p_i, \quad m_2 = -\sum i \frac{p_i}{w_2} \quad \text{for } T_2 < i \leqslant b.$$

The optimal criterion used in bilevel thresholding:

$$V_2 = w_0 w_1 (m_1 - m_0)^2.$$

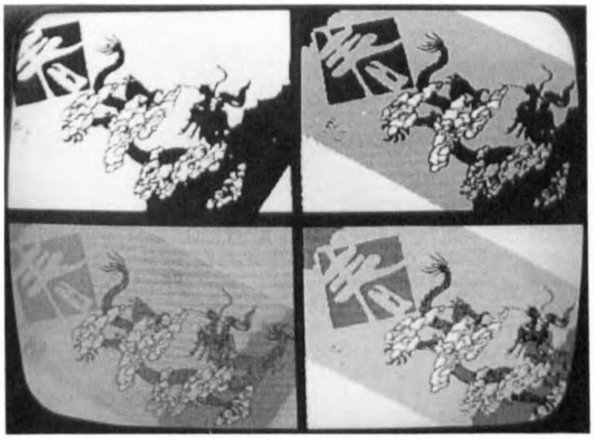
The threshold value T_1 is the gray level that maximizes the between-class variance V_2 ($T_2 = b$ in this case).

The optimal criterion used in three-level thresholding:

$$V_3 = w_0 w_1 w_2 (m_1 - m_0)^2 (m_2 - m_1)^2 (m_2 - m_0)^2.$$

T_1 and T_2 are the threshold values that maximize V_3.

(b) (c)



(a) (d)

	(b)	(c)	(d)
time(sec)	0.03	0.02	0.01
cell size	7	6	3
threshold value(s)	101	99 129	61 96 130

Figure 4. Hill-clustering approach: (a) the original image, (b) bilevel thresholding, (c) three-level thresholding, (d) four-level thresholding.

Entropy-based approach

$$w_0 = \sum p_i, \quad H_0 = -\sum \frac{p_i}{w_0} \ln \frac{p_i}{w_0} \quad \text{for } a \leqslant i \leqslant T_1,$$

$$w_1 = \sum p_i, \quad H_1 = -\sum \frac{p_i}{w_1} \ln \frac{p_i}{w_1} \quad \text{for } T_1 < i \leqslant T_2,$$

$$w_2 = \sum p_i, \quad H_2 = -\sum \frac{p_i}{w_2} \ln \frac{p_i}{w_2} \quad \text{for } T_2 < i \leqslant b.$$

Optimal criterion used:

$E_2 = H_0 + H_1$ in bilevel thresholding ($T_2 = b$ in this case),

$E_3 = H_0 + H_1 + H_2$ in the three-level thresholding.

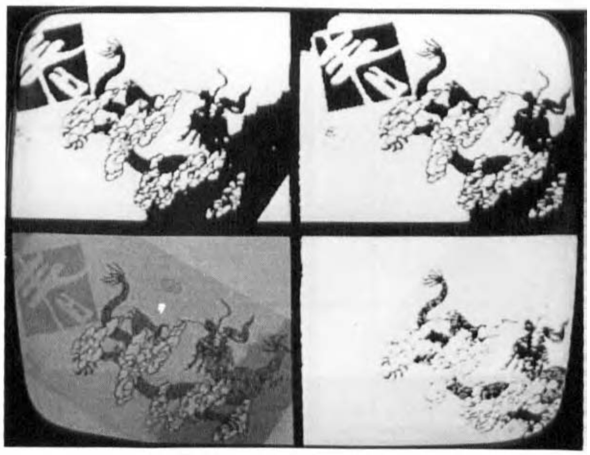
Since there is no guarantee that V_2, V_3, E_2 or E_3 does not have local maxima, the exhaustive search is the only possible method to find the best values T_1 and T_2.

Performance

The images under test are of size 256×240 with 256 gray levels. All images are taken under natural room lighting without the support of any special light sources. Table 1 shows the average computational time of twenty image samples using the three different approaches.

With the variance-based or entropy-based approach, the computational time is radically increased

(b) (c)

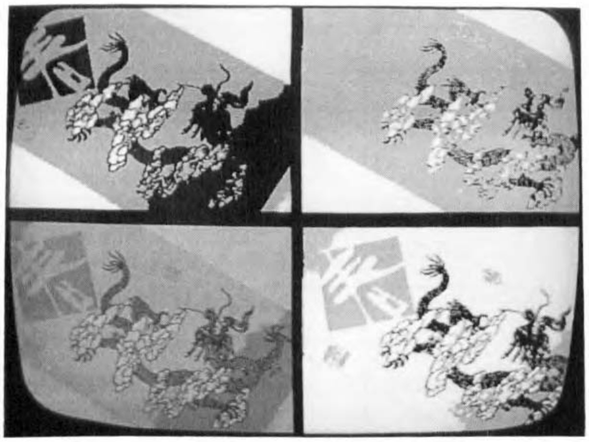


(a) (d)

	(b)	(c)	(d)
time(sec)	0.03	0.01	0.02
threshold value	101	100	80

Figure 5. Bilevel thresholding: (a) the original image, (b) hill-clustering approach, (c) variance-based approach, (d) entropy-based approach.

(b) (c)



(a) (d)

	(b)	(c)	(d)
time(sec)	0.02	21	14
threshold values	99 129	71 126	78 111

Figure 6. Three-level thresholding: (a) the original image, (b) hill-clustering approach, (c) variance-based approach, (d) entropy-based approach.

Table 1
Comparison of computational time

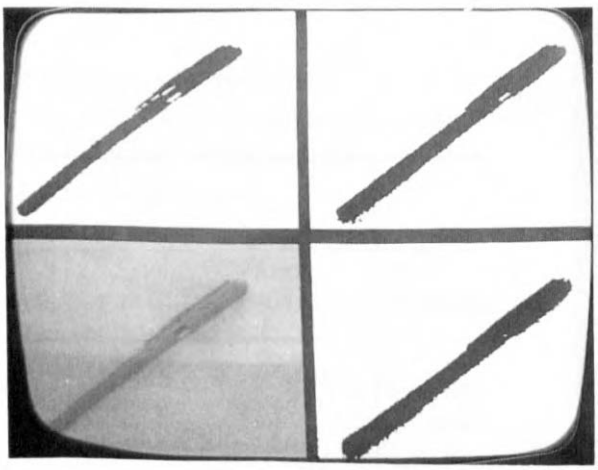
Methods	Time (seconds)* bilevel	three-level
Hill-clustering	0.026	0.018
Variance	0.016	20
Entropy	0.020	19

*Based on PC 386

from 0.02 seconds in bilevel thresholding to 20 seconds in three-level thresholding. With the hill-clustering approach, the difference of computational time between bilevel and three-level thresholding is not significant. The average final cell sizes used to detect two and three peaks are 6.2 and 3.6, respectively. For multi-level thresholding, 0.026 seconds can be considered to be the worst performance since the computational time will decrease while the user-specified number of threshold levels is increased. The cell size controls the sensitivity of the peak detection. The larger the number of peaks is specified, the smaller cell size will be employed. This reduces the iterations of the hill-clustering procedure and, hence, reduces the computational time.

In bilevel thresholding, these three approaches generate similar threshold values for 15 images out of a total of 20 test samples. Hill-clustering approach tends to locate a lower threshold value,

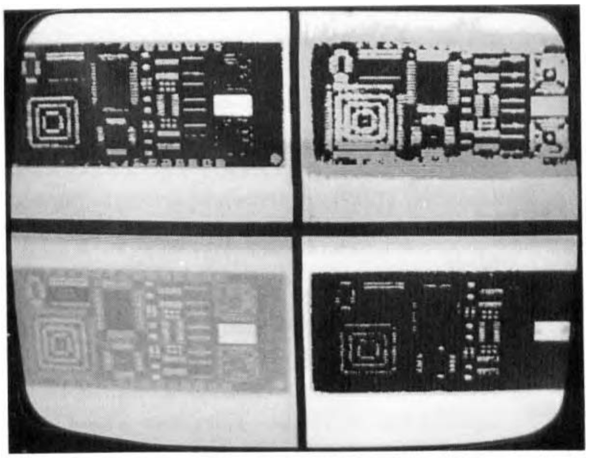
(b) (c)



(a) (d)

	(b)	(c)	(d)
time(sec)	0.01	0.02	0.02
threshold value	97	104	109

Figure 7. Bilevel thresholding of a pen: (a) the original image, (b) hill-clustering approach, (c) variance-based approach, (d) entropy-based approach.



	(b)	(c)	(d)
time(sec)	0.02	18	22
threshold values	106 133	90 174	118 155

Figure 8. Three-level thresholding of a PCB: (a) the original image, (b) hill-clustering approach, (c) variance-based approach, (d) entropy-based approach.

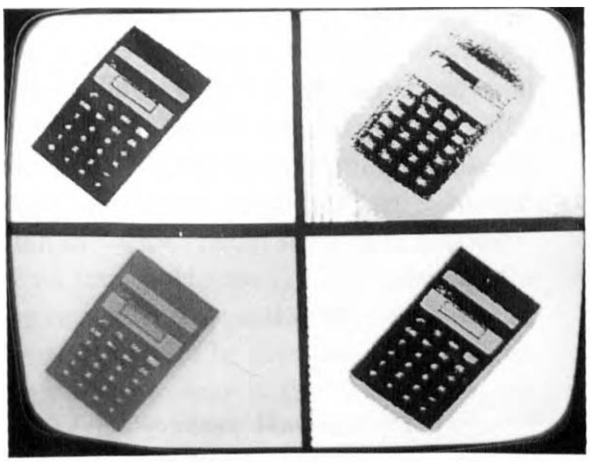
compared with that of variance-based and entropy-based approaches. This results in sharper segmentation since it is less sensitive to object's shadow.

In three-level thresholding, the distinction of threshold values among these three approaches are much significant. Variance-based approach has performed poorly because it is very sensitive to noise and object's shadow. The segmentation quality of hill-clustering and entropy approaches is much competitive to each other.

Figure 4 shows bilevel, three-level and four-level thresholding results of a New Year Card using the hill-clustering approach. The corresponding computational time and threshold values are also listed right beside each figure. Figures 5 and 6 show, respectively, the bilevel and three-level thresholding results of the same New Year Card using those three approaches. Figure 7 gives the binary image of a pen. Figure 8 illustrates the three-level thresholding result of a printed circuit board. Figure 9 also demonstrates the three-level thresholding result of a calculator.

4. Conclusion

A simple and efficient hill-clustering approach has been developed for multi-level thresholding. It requires only one user-specified parameter, namely the desired number of peaks. The variation of



	(b)	(c)	(d)
time(sec)	0.02	20	25
threshold values	90 126	70 170	86 153

Figure 9. Three-level thresholding of a calculator: (a) the original image, (b) hill-clustering approach, (c) variance-based approach, (d) entropy-based approach.

computational time is insignificant regardless of the number of peaks to be identified. The segmented image is sharper than or as good as that of variance-based and entropy-based approaches. It becomes more attractive when the desired levels of thresholding is greater than two. This makes the hill-clustering approach appropriate for time-critical applications.

Currently extensions are being investigated to access the inclusion of both the gray-level of each pixel and the average gray level at its neighbors. This makes the 1-D hill-clustering procedure a 2-D one. The payback of this approach should estimate noise and make the segmentation more accurate.

References

[1] Abutaleb, A.S. (1989). Automatic thresholding of gray-level pictures using two-dimensional entropy. *Computer Vision, Graphics, and Image Processing* 47, 22–32.

[2] Kapur, J., P. Sahoo and A. Wong (1985). A new method for gray-level picture thresholding using the entropy of the histogram. *Computer Vision, Graphics, and Image Processing* 29, 273–285.

[3] Otsu, N. (1979). A threshold selection method for gray-level histogram. *IEEE Trans. Syst. Man Cybernet.* 9, 62–66.

[4] Reddi, S., S. Rudin and H. Keshavan (1984). An optimal threshold scheme for image segmentation. *IEEE Trans. Syst. Man Cybernet.* 14, 661–665.

[5] Sahoo, P., S. Soltani and A. Wong (1988). A survey of thresholding techniques. *Computer Vision, Graphics, and Image Processing* 41, 233–260.

[6] Snyder, W. (1990). Optimal thresholding—a new approach. *Pattern Recognition Letters* 11, 803–810.

[7] VISILOG Programmer's Guide (1988). Noesis S.A.R.L., France.

[8] Weszka, J. and A. Rosenfeld (1979). Histogram modifications for threshold selection. *IEEE Trans. Syst. Man Cybernet.* 9, 38–52.