Automatic Multilevel Histogram Thresholding

Ren-You Huang[#1], Lan-Rong Dung[*2], Yin-Yi Wu[+3]

[#]*Institute of Electrical Control Engineering, National Chiao Tung University,*
1001 Ta Hsueh Rd., Hsinchu, Taiwan.
[1]hry76519@gmail.com

[*]*Department of Electrical and Computer Engineering, National Chiao Tung University,*
1001 Ta Hsueh Rd., Hsinchu, Taiwan.
[2]lennon@faculty.nctu.edu.tw

[+]*National Chung-Shan Institute of Science and Technology, Tao-Yuan, Taiwan*
[3]davidwu0801@gmail.com

Abstract— **This paper proposed a novel approach which aims to automatically obtain the multilevel thresholds of histogram for image segmentation purposes. The proposed algorithm is unsupervised, i.e. we don't make any assumption about the input images, the number of thresholds, and the distributions of histograms. In contrast to the conventional algorithms which aim to minimize the within-class variance or maximize the entropy function, our method aims to find the "significant peaks" to represent the major clusters of histogram and select the thresholds between every pair of successive peaks. The significant peaks are determined by the proposed histogram local least square regression (LLSR), which is robust to noisy histogram and false peaks. There are only two parameters which are all independent to input image and fixed, thus the proposed method is adaptive to different cases of histograms. The experiments show that our algorithm is able to obtain appropriate thresholds without any prior knowledge and gives visually satisfactory thresholding results. We also proposed a novel quantitative index which is suitable for evaluating the thresholded histogram.**

Keywords—**Automatic multi-level histogram thresholding, least square regression, image segmentation.**

1. Introduction

Image segmentation is an important low-level image processing technique which is widely used as a pre-processing step in many computer vision applications. The objects with different colors or intensities will be extracted in image segmentation step for further processing.

There are many algorithms for image segmentation task. For example, mean shift [10][11] are most well-known algorithms for their satisfactory segmentation results. Comaniciu et al. [10] proposed the first mean-shift based image segmentation algorithm which clusters every pixel to the local mode in high dimensional feature space. However, the segmentation results depend on the parameter settings, i.e. the color distance and spatial distance. Moreover, the computational complexity is expensive since every pixel is clustered in 3-D/5-D feature space for gray/color images.

Image thresholding is another popular approach for image segmentation purposes. The thresholding can be classified into spatial based [12] and histogram based [1]-[8] algorithms. Bradley et al. [12] proposed an adaptive thresholding algorithm using the integral image to achieve real-time performance which is robust to extremely non-uniform illumination condition. However, this method is only able to binarize the image, hence this method is more suitable for extracting the text on documents images.

Histogram based algorithms first compute the histogram and find thresholds to classify all the pixels into a few intensity values. According to the requirement of number of thresholds, the histogram based algorithms can be further classified into automatic [1]-[4] and semi-automatic [5]-[8]. Automatic algorithms [1]-[4] make no assumption about the number of thresholds and find the thresholds to minimize some cost functions while semi-automatic methods require the number of thresholds as an input parameter such that the algorithms can start to find the optimal thresholds. The computational complexity of histogram based algorithms is usually low in contrast to other algorithms since these algorithms only process 256 bins rather than all pixels in image. In this paper, we focus on the histogram based algorithms.

We proposed a histogram based algorithm which aims to find the "significant peaks" based on histogram information. To make the algorithm robust to noisy histogram, we proposed a local least square regression to re-estimate every bin in the histogram and obtain the significant peaks. The thresholds are located between every two successive peaks. The main advantage of our method is that there are only two parameters independent to the input images, hence the proposed algorithm is adaptive to most cases of images.

The remainder of this paper is organized as follows. The conventional histogram based algorithms are briefly reviewed in Section II. The proposed algorithms are described in Section III. The experiments and comparisons are shown in Section IV. And finally, conclusions are summarized in Section V.

2. RELATED WORKS

Image thresholding techniques have been widely used in many low level computer vision applications. Otsu [5] proposed an optimal threshold-selection method which selects the thresholds to maximizing between-class variance. Original Otsu's method is an exhaustive searching algorithm which considers all the possible combinations of k (desired number) thresholds and selects the optimal ones which maximize the between-class variance, hence the computational complexity is extremely expensive as the number k increase. Liao et al. [6] proposed a fast version of Otsu's method which is able to significantly speed up the algorithms. However, the complexity still exponentially increases as the k increases since the nature of exhaustive searching. Dong et al. [7] proposed an algorithm which is mathematically equivalent to Otsu's method with only linear complexity of k rather than exponential complexity. Lopes et al. [8] applied the fuzzy set theory and proposed a fuzzy-based algorithm which selects the optimal thresholds according to the index of fuzziness. The above methods require the number of thresholds as input parameter which affects the thresholding results and quality. If the number of thresholds is equal to the number of modes in the histogram, the thresholding results will be satisfactory. Otherwise, the image will be over-segmented or under-segmented.

To make the thresholding task more adaptive to different histogram distribution without any prior knowledge, automatic thresholding algorithms were proposed in many works, e.g. [1]-[4]. Yen et al. [2] proposed an automatic method which selects the optimal thresholds based on maximizing total correlation. The histogram is first separated into two partitions and the one of all partitions with largest variance will be separated into another two partitions. The process repeats until the cost function of discrepancy and number of clusters reaches its minimum. Sezgin et al. [3] proposed an improved version of [2], which selects the optimal thresholds based on not only maximizing the total correlation but also minimizing the within-class variance. The overall separation process is similar to the algorithm in [2], except the threshold selection criterion. Sahaphong et al. [4] proposed an automatic thresholding method based on fuzzy c-means clustering algorithm which starts from Otsu's method to find the initial two partitions of the histogram and increases the threshold number until the change of energy function is below a pre-defined tolerance. Bruzzese et al. [1] proposed an algorithm which selects the thresholds by minimizing the fuzzy entropy. The "divide and conquer" strategy adopted in [1] is quite similar to the algorithms proposed in [2][3]. The histogram partitioning process repeats until the changing ratio of fuzzy entropy is below a pre-defined value.

In general, the automatic methods are more suitable for the unsupervised histogram thresholding with unknown input images. For more comprehensive thresholding algorithms of earlier researches, please refer to Segzin's survey of image thresholding [9].

3. PROPOSED ALGORITHM

3.1. Overview

Histograms of images provide statistical information about the intensity or color distributions. The ideal histogram is that one intensity level represents one object. However, there are many factors which affect the captured image in real world, e.g., non-uniform illumination, noise, blurring, nonhomogeneous object textures, etc. Fig. 1 shows the comparison of ideal and real-world image histogram. Although the histogram is influenced by the factors mentioned above, we are still able to separate different object or intensity level by locating the significant peaks and determine the thresholds between two successive peaks to segment the object in image. Our goal is to develop an algorithm which is automatic and adaptive to different histogram distribution without any prior knowledge. The overall flow of the proposed method is shown in Fig. 2.

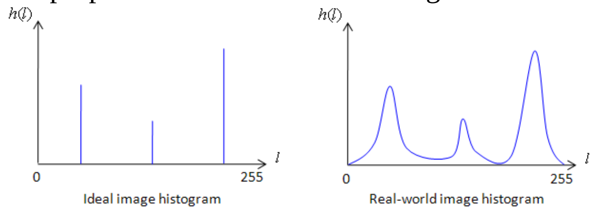


Fig. 1. The comparison of ideal and real-world image histogram.

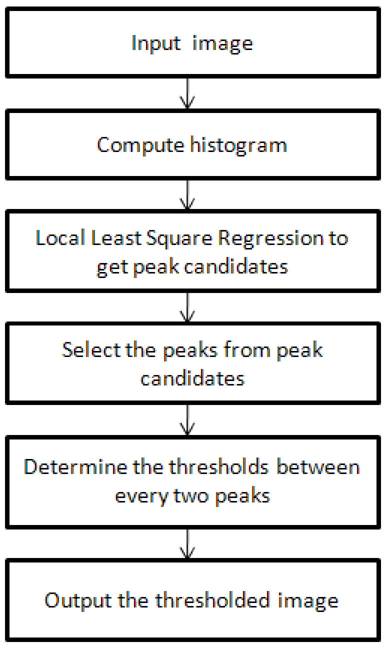

Input image
Compute histogram
Local Least Square Regression to get peak candidates
Select the peaks from peak candidates
Determine the thresholds between every two peaks
Output the thresholded image

Fig. 2. The flowchart of the proposed algorithm.

First, we compute the histogram of input image I as follows:

$$h(g)=\sum_{\mathbf{x}\in\Omega}\delta\left[I(\mathbf{x})-g\right] \tag{1}$$

where h(g) is g-th bin of histogram, g ∈ [0, 255]. We normalize the histogram by dividing each bin by the maximal bin:

$$H(g)=\frac{h(g)}{\max_{g} h(g)} \tag{2}$$

where H(g) is the g-th bin of normalized histogram, g ∈ [0, 255]. Note that the normalization of our method is different from traditional one since we want to fix the scale of histogram in the range of minimum 0 and maximum 1, such that the parameter setting can be fixed. Next, we smooth the histogram and obtain peak candidates by applying 2ndorder polynomial regression to every bin of histogram. The regression is perform in least square sense, hence the proposed histogram smoothing scheme is named “local least square regression” (LLSR). Next, we select the true peaks according to the location of peak candidates and the original histogram information. Finally, we determine the thresholds between every two successive peaks and segment the image based on the thresholds.

3.2. Local least square regression (LLSR)

Histograms of real-world images may contain many noises and “false peaks”. For example, Fig. 3 shows a desktop image of windows XP and the corresponding histogram. It is clear that there are many noise and false peaks in the histogram. To make the histogram smoother, the intuitive method is filtering, e.g. Gaussian filter or mean filter. However, the smoothness of filtered histogram depends on the window size, hence the filtering cannot be adaptive to different histogram distributions. The noisier the histogram is, the larger windows size is needed. Moreover, these linear filters are not able to filter out the false peaks clearly. Thus, our goal is to develop an algorithm which is able to smooth different histograms adaptively without tuning any parameter.

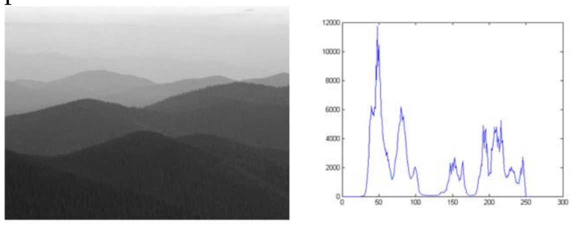


Fig. 3. Example image (left) and the intensity histogram (right).

Motivated by the data analysis, we apply a second order linear regression to find the “trend” of data in a window centred at i-th bin and re-estimate the value of i-th bin according to the regression solution. The optimal solution of polynomial coefficients is obtained by least square method, hence the proposed algorithm is named “local least square regression” (LLSR).

Fig. 4 depicts the LLSR algorithm. To re-estimate every bin of histogram, the information of histogram bins within the window of size 2*w+1 centred at i-th bin are exploited to least square regression. We denote the polynomial coefficients of i-th bin as $\mathbf{a}_i=\left(a_{i2},a_{i1},a_{i0}\right)^T$, the data vector $\mathbf{x}_j = \left(j^2, j, 1\right)^T$ where j ∈ {-w, -w +1, ..., 0, ..., w-1, w}, and histogram information of i-th bin y_i . The least square regression can be written as follows:

$$\mathbf{a}_i = \arg\min \sum_{j=-w}^{w} \left|\mathbf{x}_j^T \mathbf{a} - y_{i+j}\right|^2 \qquad (3)$$

Let $\mathbf{X} = \left(\mathbf{x}_{-w}, ..., \mathbf{x}_w\right)^T$ be a (2w+1)-by-3 data matrix, and $\mathbf{y}_i = \left(y_{i-1}, ..., y_{i+w}\right)^T$, the optimal solution of the least square problem can be obtained as follows:

$$\mathbf{a}_i = \left(\mathbf{X}^T \mathbf{X}\right)^{-1} \mathbf{X}^T \mathbf{y}_i \qquad (4)$$

Consider the boundary problem, i.e. the range of the window is out of [0, 255], we reflect the histogram information at boundaries 0 and 255. In other words, the elements y_{i+j} in (3) should be re-written as follows:

$$y_{i+j} = \begin{cases} h\left(-(i+j)\right) & , if\ (i+j) < 0. \\ h\left(510-(i+j)\right) & , if\ (i+j) > 255. \\ h(i+j) & , otherwise. \end{cases} \qquad (5)$$

The LLSR processes every bin of histogram, as shown in Fig. 5. It is obvious that the re-estimate histogram is much smoother than the original one.

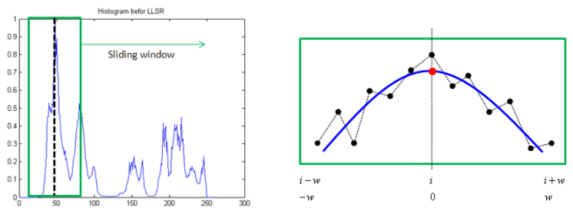


Fig. 4. Illustration of LLSR algorithm. All the histogram information in the window centred at i-th bin are exploited to re-estimate the i-th bin. The sliding window will process all the bins of histogram.

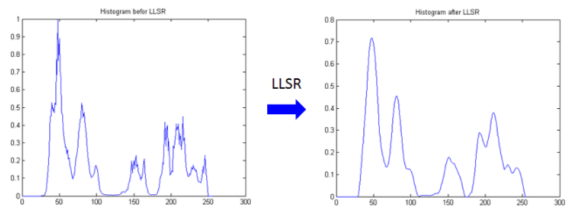


Fig. 5. Left: original histogram. Right: histogram re-estimated and smoothed by local least square regression. The peaks are much clearer than original ones.

3.3 Peak candidates

The estimated coefficients of a second order polynomial (parabola) can be further exploited to determine whether the i-th bin is a possible peak or not. A peak can be described as a "downward parabola", hence we only have to consider the coefficient of second order term, namely a_{i2} . Fig. 6 shows different a_{i2} and the corresponding curves of second order terms. $a_{i2} > 0$ describes an upward parabola, which is obviously not a peak. However, if a_{i2} approaches to 0, the second order polynomial function degrades to linear or constant function. Here we set a pre-defined a positive value θ for a_{i2} to select possible peaks. In other words, the i-th bin is a possible peak if $a_{i2} < -\theta$. The constraint of a_{i2} only describes a downward parabola. However, as shown in Fig. 7, the peak of second order polynomial may not locate at i-th bin (the center of window), hence we defined the second condition to eliminate inappropriate peak candidates. Let $\hat{\mathbf{y}}_i = \left(\hat{y}_{i-w}, ..., \hat{y}_i, ..., \hat{y}_{i+w}\right)^T$ be the histogram bins estimated by $\mathbf{a}_i$ and $\mathbf{X}$,

$$\hat{\mathbf{y}}_i = \mathbf{X}\mathbf{a}_i \qquad (6)$$

As shown in Fig. 7, an appropriate peak is at least greater than the two boundary bins in the window, i.e. $\hat{y}_{i-w}$ and $\hat{y}_{i+w}$. Hence the i-th bin with coefficient $a_{i2} < -\theta$ which satisfies $\hat{y}_i > \hat{y}_{i-w}$ and $\hat{y}_i > \hat{y}_{i+w}$ is defined as a peak candidate.

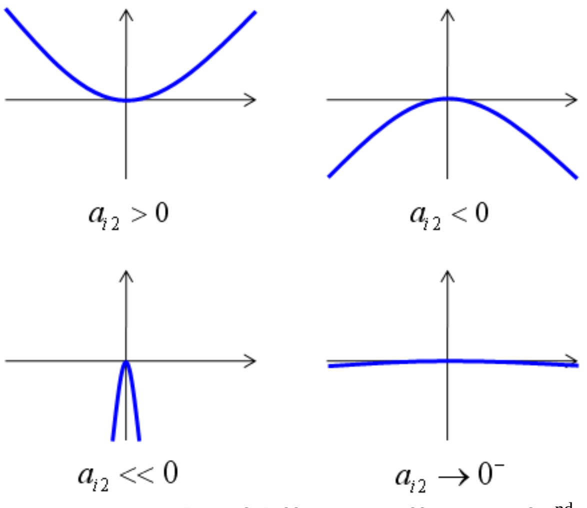


Fig. 6. Examples of different coefficients of 2nd order term.

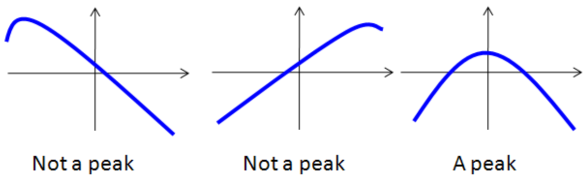


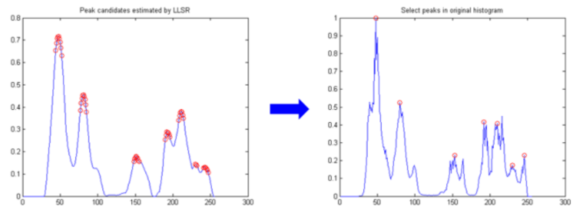
Fig. 7. Examples of different downward parabolas.

3.4. Thresholds selection

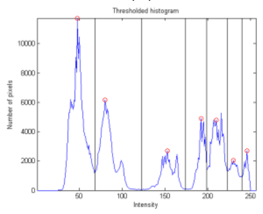
After LLSR re-estimation and selection of peak candidates, we refine the peaks locations from the peak candidates. Since the candidates are selected coarsely, one real peak may be surrounded by many candidates. Hence a simple rule for truth peaks selection is defined as follows:

$$p_k = \arg\max_{g \in S_k} \{h(g)\} \qquad (7)$$

Where p_k is the k-th peak, g is the gray level which belongs to the range of k-th successive candidates S_k. Note that here we select the peaks according to corresponding original histogram bins instead of the re-estimated ones since the local peaks in original histogram are more representative than peaks in the re-estimated histogram. Fig. 8(a) shows an example of peaks selection.



(a)



(b)

Fig. 8. Peaks and thresholds Selection. (a) Peaks selection (b) thresholds selection.

After all the truth peaks are located, we determine the thresholds by choosing the "deepest valley" between every two successive peaks since the valleys usually minimize the misclassification error. The thresholds selection is defined as follows:

$$t_k = \arg\min_{g \in [p_k, p_{k+1}]} \{h(g)\} \qquad (8)$$

where t_k is the k-th threshold. Fig. 8(b) shows an example of thresholds selection. The locations of thresholds are marked by black vertical lines. The thresholded image is shown in Fig. 9.

Let N_p be the number of peaks and N_t be the number of thresholds (valleys), N_t is always equal to $N_p - 1$ in our method. If $N_p \le 1$, i.e. there is only one peak or even no peak to determine one threshold. For this situation, we apply the Otsu's method [5][6] to obtain one threshold which optimally divides the histogram into two partition. Note that the computational complexity of Otsu's method for binarizing histogram is still low.

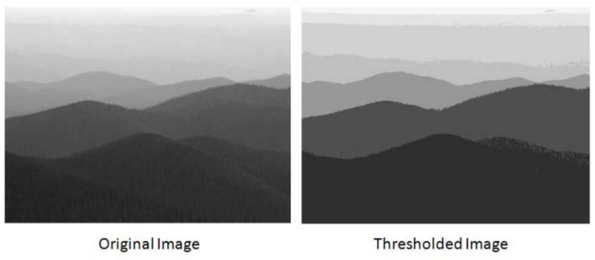


Fig. 9. Example of image thresholding according to the thresholds obtained by our method.

4. EXPERIMENTS

This section shows the experimental results and compares our method with Bruzzese's method [1] and Sezgin's method [3]. We denote the fuzzy entropy thresholding [1] as "FET" and Sezgin's method [3] as "SEZ" in the rest of this paper. We tried our best to implement the FET and SEZ algorithms according to the steps described in the papers for fair comparisons. In our method, there are only two parameters, which are window size w and the constraints value θ of a_{i2}. The window size affects the result of least square regression. Smaller w leads to noisier estimation while larger w leads to smoother estimation. Fortunately, the least square regression is not very sensitive to the number of data, hence we only need to give a reasonable window size, e.g. 10 times the order of polynomial model. In this paper, we set w = 10 and θ = 0.0001, and fixed these two parameters in all experiments.

4.1. Quantitative evaluation

Since the multilevel thresholding is application-dependent, it is difficult to determine ground truths for all applications. In [1][3], the authors applied the "non-uniformity" (NU) or "uniformity" (U) to evaluate the thresholded histogram. The uniformity is given by the following equation:

$$U = 1 - NU = 1 - \frac{\sum_{k=1}^{N_t+1} w_k \sigma_k^2}{\sigma_{\max}^2} = 1 - \frac{\sigma_W^2}{\sigma_{\max}^2} \quad (9)$$

where w_k is sum of the probability density function within k-th histogram interval, σ_k^2 is the corresponding within-class variance, $\sigma_{\max}^2$ is the variance of overall distribution used for normalizing the non-uniformity value to [0, 1], and σ_W^2 is denoted as the total within-class variance. The non-uniformity has the same meaning as the "discrepancy" defined in [2], which describes the difference between thresholded histogram and original histogram. However, these measurements will monotonic decrease as the number of thresholds increases. Here we modify the uniformity formula and define a novel quantitative index as follows:

$$\eta = 1 - \frac{\sigma_W^2}{\sigma_B'^2} \quad (10)$$

Where the σ_W^2 is the same within-class variance in (9) and $\sigma_B'^2$ is a novel between-class variance defined as follows:

$$\sigma_B'^2 = \sum_{k=1}^{N_t} \frac{w_k (\mu_k - t_k)^2 + w_{k+1} (\mu_{k+1} - t_k)^2}{w_k + w_{k+1}} \quad (11)$$

where μ_k is the weighted mean of k-th histogram interval. The novel between-class variance describes the sum of between-class variance with respect to each threshold. In contrast to the original between-class variance defined in [5] which increases as the number of thresholds increases, our version of between-class variance will not always increase as the number of thresholds increases, hence the situation that over-segmented histogram leads to high index value can be avoided.

4.2. Results

In the following experiments, we select some test images in the UC Berkeley segmentation dataset [13][14], four desktop images of Windows XP, and some classical test images in image processing area, e.g. *Lena*, *peppers*, *house*, *cameraman*.

Fig. 10 shows the thresholded images of Windows XP desktop. By visual observation, the SEZ method gives more details but the thresholded image is noisier; the FET method gives less number of thresholds and leads to more homogeneous results but loses some details; our method gives clearer results than SEZ and more details than FET. Since our algorithm aims to find the significant peaks and there is only one peak in the image winter, hence our algorithm switches to Otsu's method and find one optimal threshold to separate the histogram. Table 1 lists the evaluation of thresholded histograms using (10), the values marked by black bold style are the highest index values corresponding to each test image.

TABLE 1
Evaluations of Windows XP Desktop Images

Images	**SEZ[3]**	**FET[1]**	**Ours**
Blue hills	0.9487	0.9053	**0.9765**
Water lilies	**0.9439**	0.6971	0.9409
Sunset	0.9017	0.8070	**0.9110**
Winter	**0.9525**	0.8847	0.8044
Average	**0.9367**	0.8235	0.9082

Fig. 11 shows the thresholded images of classical test images. Our method still gives more homogeneous images than images thresholded by SEZ, and preserves more details than images thresholded by FET. The quantitative index values are shown in Table 2. Note that our method obtained the lowest quantitative index value for thresholding *Sailboat* image, however, the result is still acceptable.

TABLE 2
Evaluations of Classical Test Images

Images	**SEZ[3]**	**FET[1]**	**Ours**
Lena	**0.9397**	0.7376	0.9220
Peppers	0.9338	**0.9719**	0.9702
House	0.9240	0.6032	**0.9252**
Cameraman	0.9419	0.9046	**0.9420**
Sailboat	**0.9393**	0.9327	0.8814
Average	**0.9357**	0.8300	0.9282

Fig. 12 and Fig. 13 show 14 images we selected from UC Berkeley segmentation dataset [14] for visual comparisons. By observation, our algorithm gives clearer thresholded images than SEZ and preserves more details than FET. Table 3 lists the quantitative index evaluated by (10). Our algorithm is slightly higher than SEZ and greatly higher than FET in most cases. In some cases, the SEZ obtained over-segmented histograms, the quantitative index values decrease since the between-class variance defined in (11) become smaller as the number of thresholds increases. The FET usually produces images with fewer thresholds then other methods, hence in some cases with significant multi-modal distributions, the images thresholded by FET obtained poor index values.

TABLE 3
Evaluations of UC Berkeley Dataset

Images	**SEZ[3]**	**FET[1]**	**Ours**
UCB-1	**0.9577**	0.5606	0.9547
UCB-2	0.9709	**0.9732**	0.9454
UCB-3	**0.9676**	0.8594	0.9368
UCB-4	0.9613	0.9059	**0.9790**
UCB-5	0.8901	0.8904	**0.9058**
UCB-6	0.9176	0.7931	**0.9342**
UCB-7	0.9064	0.7000	**0.9067**
UCB-8	0.8843	0.8925	**0.9098**
UCB-9	0.9403	0.3397	**0.9479**
UCB-10	0.9380	0.9022	**0.9472**
UCB-11	0.9336	0.9065	**0.9430**
UCB-12	0.9191	0.8286	**0.9548**
UCB-13	**0.9267**	0.7422	0.9224
UCB-14	0.9137	0.6275	**0.9543**
Average	0.9305	0.7801	**0.9387**

In most cases of the experiments, the SEZ gave more details but noisier images; the FET gave much more homogeneous images but lost many details; on the contrary, our algorithm obtained homogeneous thresholded images and preserved important intensity levels at the same time. However, according to visual observation and quantitative evaluations, our method is not suitable for thresholding uni-modal distribution. For the cases with uni-modal or near uni-modal distributions, the SEZ and FET are more suitable than our method to threshold the histograms.

5. CONCLUSIONS

This paper proposed a novel histogram thresholding algorithm which aims to find the significant peaks and determine the thresholds between every two successive peaks. In order to make the algorithm robust to noisy histogram and false peaks, we proposed the local least square regression (LLSR) scheme to re-estimate each bin and select peak candidates simultaneously. The proposed method is fully automatic, and suitable for thresholding the images with significant intensity levels without any prior knowledge. The experiments show that our method gives the results which are homogeneous with appropriate number of thresholds. The quantitative index values show that our algorithm slightly outperforms the SEZ algorithm [3] and FET algorithm [1].

In the future, we will try to modify our algorithm for more general images. For example, develop a strategy for the unimodal histogram instead of applying the Otsu's method [5], or determine appropriate thresholds for extremely noisy (spiky) histogram.

REFERENCES

[1] D. Bruzzese and U. Giani, "Automatic multilevel thresholding based on a fuzzy entropy measure", In Classification and Multivariate Analysis for Complex Data Structures, pp.125-133, 2011.

[2] J. C. Yen, F. J. Chang, and S. Chang, "A new criterion for automatic multilevel thresholding", IEEE Trans. Image Process., vol. 4, no. 3, pp. 370-378, 1995.

[3] M. Sezgin and R. Tasaltin, "A new dichotomization technique to multilevel thresholding devoted to inspection applications", Pattern Recognition Letters, 21, pp. 151-161, 2000.

[4] S. Sahaphong and N. Hiransakolwong, "Unsupervised image segmentation using automated fuzzy c-means", in Proc. IEEE Int. Conf. Computer and Information Technology, pp. 690-694, Oct. 2007.

[5] N. Otsu, "A threshold selection method from gray level histograms", IEEE Trans. Syst., Man, Cybern., vol. SMC-9, pp. 62-66, 1979.

[6] P. S. Liao, T. S. Chen, and P. C. Chung, "A fast algorithm for multilevel thresholding", J. Inf. Sci. Eng., 17, pp. 713-727, 2001.

[7] L. Dong, G. Yu, P. Ogunbona, and W. Li, "An efficient iterative algorithm for image thresholding", Pattern Recognition Letters, 29, pp. 1311-1316, 2008.

[8] N. V. Lopes, P. A. Mogadouro do Couto, H. Bustince, and P. Melo-Pinto, "Automatic histogram threshold using fuzzy measures", IEEE Trans. Image Process., vol. 19, no. 1, pp. 199-204, 2010.

[9] M. Sezgin and B. Sankur, "Survey over image thresholding techniques and quantitative performance", J. Electron. Imag., vol. 13, no. 1, pp. 146-165, Jan. 2004.

[10] D. Comaniciu and P. Meer, “Mean shift: a robust approach toward feature space analysis”, IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 24, no.5, pp. 603-619, 2002.

[11] Z. Qian, C. Zhu, and R. Wang, “An Improved Fast Mean Shift Algorithm for Segmentation”, International Conference on Computer Application and System Modeling, pp. 116-120, 2010.

[12] D. Bradley and G. Roth, “Adaptive thresholding using the integral image”, journal of graphics, gpu, and game tools, 12, pp. 13-21, 2007.

[13] D. Martin, C. Fowlkes, D. Tal, and J. Malik, “A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics”, Proc. 8th Int’l Conf. Computer Vision, vol. 2, pp. 416-423, July, 2001.

[14] UC Berkeley segmentation dataset: http://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/

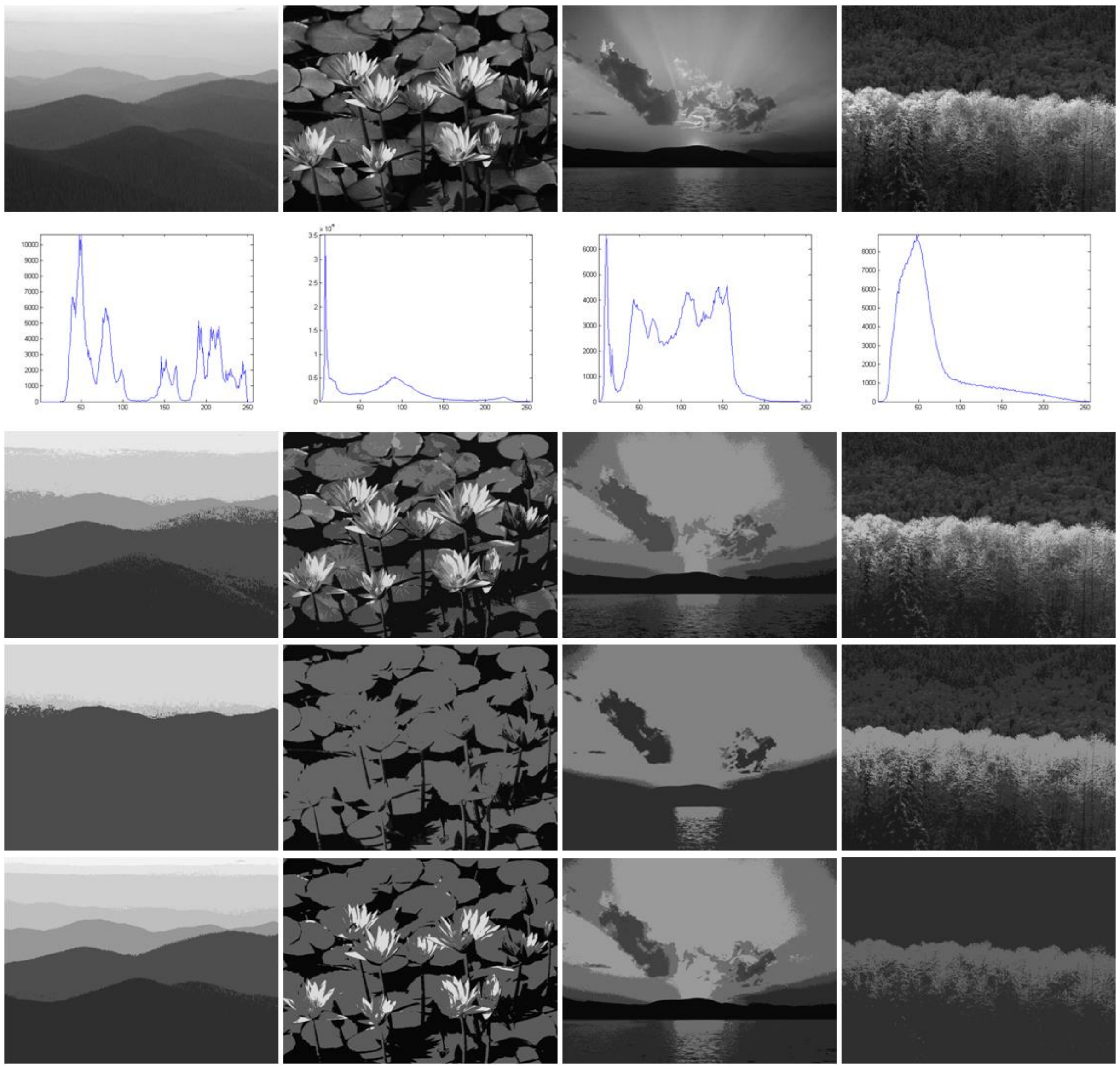


Fig. 10. Thresholding results of the Windows XP desktop images. The 1st row: input gray images, from left to right, which are blue hills, water lillies, sunset, winter, respectively. The 2nd row: histograms corresponding to the input images. The 3rd row: images thresholded by SEZ [3]. The 4th row: images thresholded by FET [1]. The 5th row: images thresholded by the proposed method.

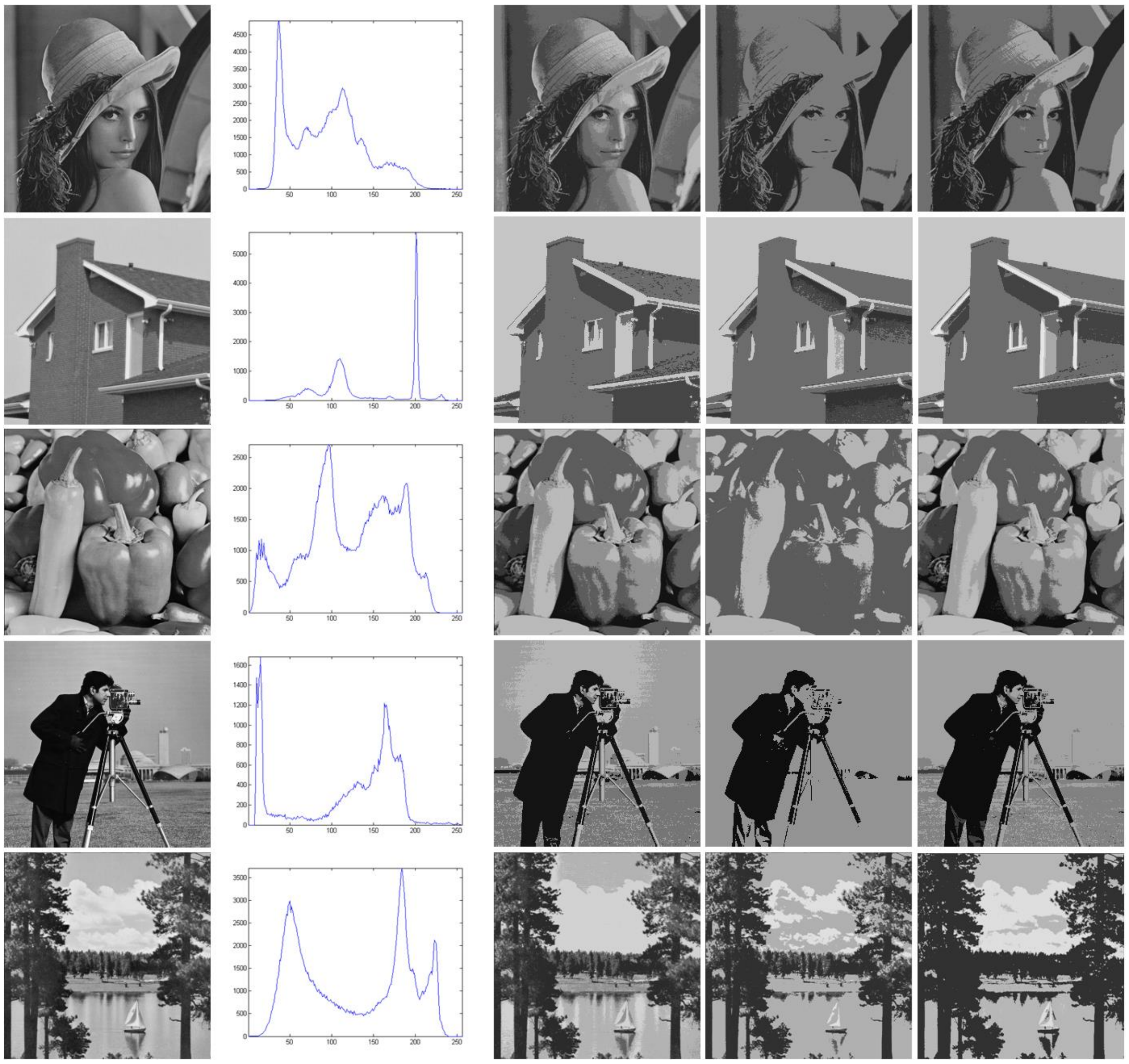


Fig. 11. Thresholding results of the classical test images. The 1st column: original images, from top to bottom, which are Lena, House, Peppers, Cameraman, and Sailboat, respectively. The 2nd column: histograms corresponding to the input images. The 3rd column: images thresholded by SEZ [3]. The 4th column: images thresholded by FET [1]. The 5th column: images thresholded by the proposed method.

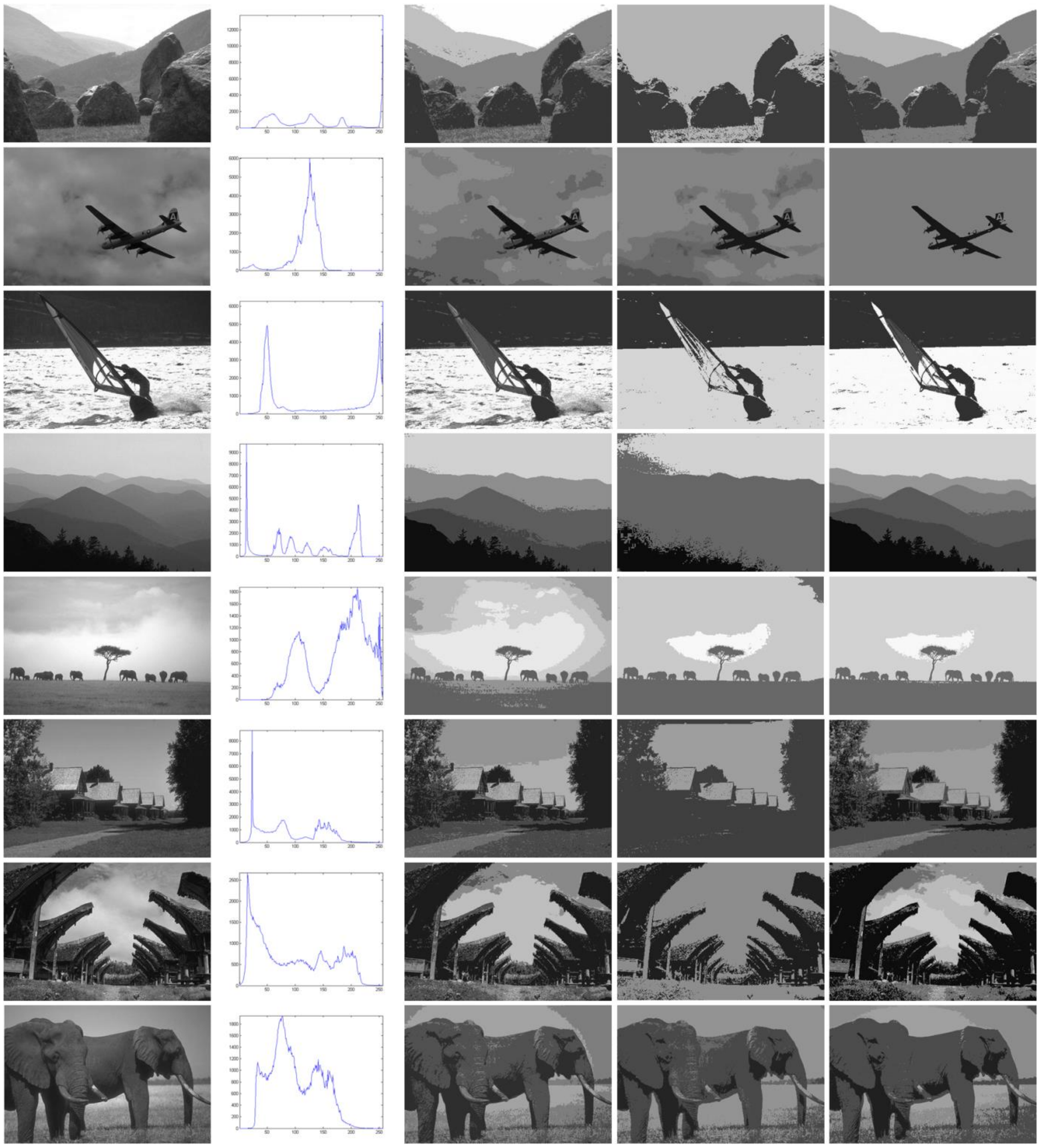


Fig. 12. Thresholding results of the UC Berkeley dataset [14]. The 1st column: original images, from top to bottom, which are UCB-1, UCB-2, UCB-3, UCB-4, UCB-5, UCB-6, UCB-7, and UCB-8, respectively. The 2nd column: histograms corresponding to the input images. The 3rd column: images thresholded by SEZ [3]. The 4th column: images thresholded by FET [1]. The 5th column: images thresholded by the proposed method.

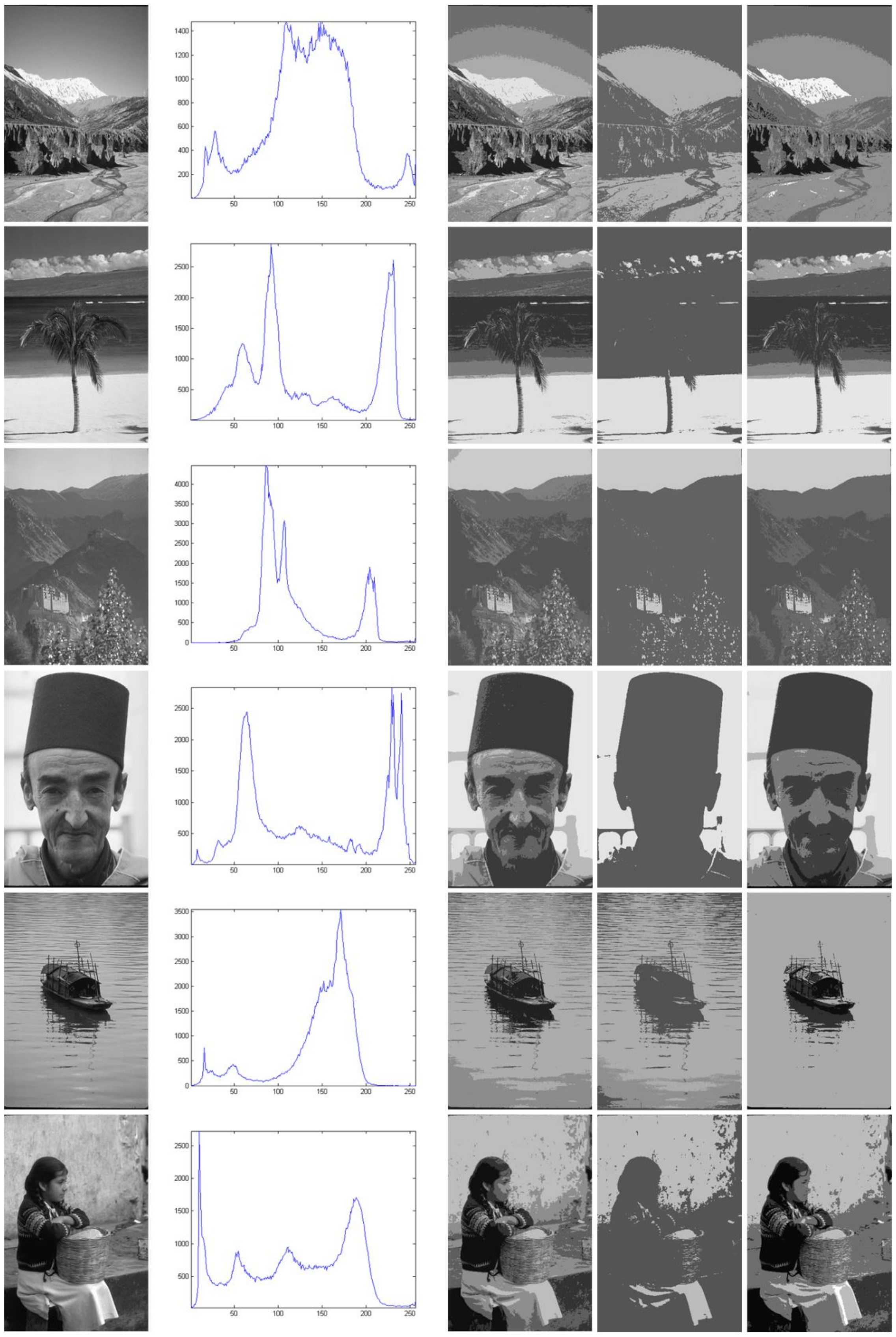


Fig. 13. Thresholding results of the UC Berkeley dataset [14]. The 1st column: original images, from top to bottom, which are UCB-9, UCB-10, UCB-11, UCB-12, UCB-13, and UCB-14, respectively. The 2nd column: histograms corresponding to the input images. The 3rd column: images thresholded by SEZ [3]. The 4th column: images thresholded by FET [1]. The 5th column: images thresholded by the proposed method.