Available online at www.sciencedirect.com
ScienceDirect
Pattern Recognition Letters xxx (2007) xxx–xxx
Pattern Recognition Letters
www.elsevier.com/locate/patrec

Multilevel thresholding for image segmentation through a fast statistical recursive algorithm

S. Arora [a], J. Acharya [b], A. Verma [c], Prasanta K. Panigrahi [c,*]

[a] *Dhirubhai Ambani Institute of Information and Communication Technology, Gandhinagar 382 009, India*
[b] *Indian Institute of Technology, Kharagpur 721 302, India*
[c] *Physical Research Laboratory, Navrangpura, Ahmedabad 380 009, India*

Received 12 February 2006; received in revised form 1 July 2007

Communicated by Y.J. Zhang

Abstract

A novel algorithm is proposed for segmenting an image into multiple levels using its mean and variance. Starting from the extreme pixel values at both ends of the histogram plot, the algorithm is applied recursively on sub-ranges computed from the previous step, so as to find a threshold level and a new sub-range for the next step, until no significant improvement in image quality can be achieved. The method makes use of the fact that a number of distributions tend towards Dirac delta function, peaking at the mean, in the limiting condition of vanishing variance. The procedure naturally provides for variable size segmentation with bigger blocks near the extreme pixel values and finer divisions around the mean or other chosen value for better visualization. Experiments on a variety of images show that the new algorithm effectively segments the image in computationally very less time.

Keywords: Multilevel thresholding; Image segmentation; Histogram; Recursion; Sub-range

1. Introduction

Thresholding is an important technique for image segmentation. Because the segmented image obtained from thresholding has the advantage of smaller storage space, fast processing speed and ease in manipulation, compared with a gray level image containing 256 levels, thresholding techniques have drawn a lot of attention during the last few years. The aim of an effective segmentation is to separate objects from the background and to differentiate pixels having nearby values for improving the contrast. In many applications of image processing, image regions are expected to have homogeneous characteristics (e.g., gray level, or color), indicating that they belong to the same object or are facets of an object, implying the possibility of effective segmentation.

Thresholding techniques can be divided into bi-level and multi-level category, depending on number of image segments. In bi-level thresholding, image is segmented into two different regions. The pixels with gray values greater than a certain value T are classified as object pixels, and the others with gray values lesser than T are classified as background pixels. Several methods have been proposed to binarize an image (Sezgin and Sankur, 2004). Otsu's method (1979) chooses optimal thresholds by maximizing the between class variance. Sahoo et al. (1988) found that in global thresholding, Otsu's method is one of the better threshold selection methods for general real world images with regard to uniformity and shape measures. However, inefficient formulation of between class variance makes the method very time consuming. Abutaleb (1989) used two-dimensional entropy to calculate the threshold. In (Pun's method, 1980), as modified by Kapur et al. (1985)

[*] Corresponding author. Tel.: +91 79 26314762; fax: +91 79 26301502.
E-mail addresses: prasanta@prl.res.in, prasanta@prl.ernet.in (P.K. Panigrahi).

0167-8655/$ - see front matter © 2007 Published by Elsevier B.V.
doi:10.1016/j.patrec.2007.09.005

the picture threshold is found by maximizing the entropy of the histogram of gray levels of the resulting classes. Wang et al. (2002) proposed an image thresholding approach based on the index of nonfuzziness maximization of 2D grayscale histogram. Kittler and Illingworth (1986) suggested a minimum error thresholding method. Niblack's method (1986) is a local approach which builds a threshold surface, based on the local mean, m, and local standard deviation, s, computed in a small neighborhood of each pixel in the form of $T = m + k \cdot s$, where k is a negative constant. This algorithm, however, produces a large amount of binarization noise in those areas that contain no text objects. Wu and Amin (2003) use a multi stage thresholding, first at global level, and then proceed locally over the image. Binarization for non-uniformly illuminated document images has been considered by Feng and Tan (2004).

Multilevel thresholding is a process that segments a gray level image into several distinct regions. This technique determines more than one threshold for the given image and segments the image into certain brightness regions, which correspond to one background and several objects. The method works very well for objects with colored or complex backgrounds, on which bi-level thresholding fails to produce satisfactory results. Reddi et al. (1984) proposed an iterative form of Otsu's method, so as to generalize it to multilevel thresholding. Ridler and Calward algorithm (1978) uses an iterative clustering approach. An initial estimate of the threshold is made (e.g., mean image intensity); pixels above and below are assigned to the white and black classes, respectively. The threshold is then iteratively re-estimated as the mean of two class means. The most difficult task is to determine the appropriate number of thresholds automatically. Unfortunately, many thresholding algorithms are not able to automatically determine the required number of thresholds, as has been noted by Whatmough (1991). Chang and Wang (1977) uses a lowpass/highpass filter repeatedly to adjust (decrease/increase) the number of peaks or valleys to a desired number of classes and then the valleys in the filtered histogram are used as thresholds. Boukharouba et al. (1985) define the zeros of a curvature function as multithreshold values by using a distribution function. Papamarkos and Gatos (1994) specify the multithreshold values as the global minima of the rational functions which approximate the histogram segments by using hill clustering technique to determine the peak locations of image histogram. Huang et al. (2005) proposed a multilevel thresholding for unevenly lighted image using Lorentz information measure. Tseng and Huang (1993) used an automatic thresholding method based on aspect of human visual system for edge detection and segmentation.

Keeping in mind human visual perception, extreme pixel values need not be finely quantized. By suitable coarse graining these can be progressively removed from the rest of the pixel values, which need to be finely segmented. A recursive implementation yields a non-uniform segmentation which naturally allows finer quantization around mean. This procedure zooms in to the mean in a manner similar to the approach of a variety of distributions towards Dirac delta function

$$\lim_{\sigma\to 0} f(x) = \lim_{\sigma\to 0} \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right] = \delta(x-\mu).$$

2. Approach

In the present approach, we use mean and the variance of the image to find optimum thresholds for segmenting the image into multiple levels. The algorithm is applied recursively on sub-ranges computed from the previous step so as to find a threshold and a new sub-range for the next step. The following points have been considered while designing the proposed algorithm:

(1) A large class of images have histograms having high intensity values for pixels near a certain value (generally the mean), or they have many structures at intensity values near the mean and less number of structures farther from mean. A rough estimate of such a histogram is a Gaussian distribution.
(2) The human eye is not very sensitive to the features present at both the extreme pixel intensity values, but is sensitive to distinguish features present at the mid-range values of intensities. Hence, it is useful to concentrate about the middle region of a gray scale image, i.e., about mean.
(3) Many algorithms suffer from the fact that there is no natural method to determine the number of optimum thresholds. After applying the present algorithm recursively a few times, PSNR of the thresholded image is found to saturate. This property can be used to obtain the appropriate number of thresholds. For the sake of completeness we define PSNR (peak signal to noise ratio), measured in decibel (dB) as

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

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

$$\mathrm{RMSE} = \sqrt{\frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}\left[I(i,j) - \widehat{I}(i,j)\right]^2}.$$

Here I and $\widehat{I}$ are the original and thresholded images, of size $M \times N$, respectively.

3. Algorithm

Following steps describe the proposed algorithm for image segmentation:

1. Repeat steps 2–6, $n/2-1$ times; where n is the number of thresholds.
2. Range $\boldsymbol{R}=[a,b]$; initially $a=0$ and $b=255$.
3. Find mean (μ) and standard deviation (σ) of all the pixels in $\boldsymbol{R}$.
4. Sub-ranges' boundaries T_1 and T_2 are calculated as $T_1=\mu-\kappa_1\cdot\sigma$ and $T_2=\mu+\kappa_2\cdot\sigma$; where κ_1 and κ_2 are free parameters.
5. Pixels with intensity values in the interval $[a,T_1]$and $[T_2,b]$ are assigned threshold values equal to the respective weighted means of their values.
6. $a=T_1+1, b=T_2-1$.
7. Finally, repeat step 5 with $T_1=\mu$ and with $T_2=\mu+1$.

The number of thresholds n can be chosen depending on the application under consideration. Optimum value of n for an image can be found out using the PSNR vs. n plot of the image as the rate of increase in the PSNR decreases with n and tends to saturate. Finer variation in the PSNR have no significant effect on quality of thresholded image. We stop iterating when the increase in PSNR is <0.1 dB.

Table 1
Thresholds, sub-ranges' boundaries, computational times and PSNR values for the test images

Image	n	Thresholds (sub-ranges' boundaries)	Time (ms)	PSNR (dB)	Otsu's method
Lena (512 × 512)	2	***80*** (124) ***161***	94	19.65	153 ms,
	4	***53*** (77) ***105*** (128) ***148*** (171) ***194***	172	25.90	52 min
	6	***53*** (77) ***91*** (104) ***116*** (129) ***140*** (152) ***159*** (171) ***194***	219	28.46	
	8	***53*** (77) ***91*** (104) ***108*** (116) ***122*** (129) ***135*** (142) ***146*** (152) ***159*** (171) ***194***	281	29.20	
Baboon (512 × 512)	2	***94*** (130) ***164***	94	20.57	161 ms,
	4	***63*** (88) ***111*** (130) ***149*** (171) ***185***	156	26.08	49 min
	6	***63*** (88) ***97*** (108) ***118*** (128) ***138*** (151) ***161*** (171) ***185***	219	28.46	
	8	***63*** (88) ***97*** (108) ***111*** (117) ***121*** (126) ***132*** (139) ***144*** (151) ***161*** (171) ***185***	297	28.98	
Peppers (256 × 256)	2	***72*** (117) ***162***	16	19.59	154 ms,
	4	***37*** (66) ***90*** (116) ***145*** (169) ***186***	20	25.30	51 min
	6	***37*** (66) ***77*** (87) ***96*** (111) ***130*** (146) ***157*** (169) ***186***	25	27.50	
	8	***37*** (66) ***77*** (87) ***89*** (94) ***98*** (107) ***117*** (129) ***138*** (146) ***157*** (169) ***186***	31	28.24	
Jet (512 × 512)	2	***116*** (179) ***206***	94	21.34	168 ms,
	4	***95*** (134) ***180*** (200) ***210*** (225) ***226***	156	24.04	48 min
	6	***95*** (134) ***159*** (182) ***196*** (204) ***210*** (218) ***220*** (225) ***226***	203	25.86	
	8	***95*** (134) ***159*** (182) ***189*** (196) ***199*** (204) ***207*** (211) ***214*** (218) ***220*** (225) ***226***	265	26.14	
Jet (512 × 512) with $\kappa_1=0.6$ $\&\kappa_2=1.7$	2	***114*** (178) ***205***	94	21.34	
	4	***74*** (100) ***131*** (172) ***194*** (205) ***212***	160	26.04	
	6	***74*** (100) ***106*** (116) ***133*** (158) ***178*** (191) ***198*** (205) ***212***	207	28.79	
	8	***74*** (100) ***106*** (116) ***117*** (118) ***127*** (142) ***157*** (172) ***183*** (191) ***198*** (205) ***212***	267	29.70	
Ariel (256 × 256)	2	***105*** (141) ***184***	16	20.90	140 ms,
	4	***85*** (96) ***112*** (133) ***157*** (185) ***209***	21	25.79	51 min
	6	***85*** (96) ***100*** (108) ***117*** (129) ***143*** (158) ***171*** (185) ***209***	26	28.16	
	8	***85*** (96) ***100*** (108) ***110*** (115) ***119*** (126) ***134*** (143) ***150*** (158) ***171*** (185) ***209***	32	28.79	
House (256 × 256)	2	***104*** (138) ***186***	16	21.39	165 ms,
	4	***69*** (93) ***112*** (123) ***135*** (183) ***191***	20	26.46	50 min
	6	***69*** (93) ***99*** (107) ***116*** (122) ***127*** (138) ***156*** (183) ***191***	24	28.43	
	8	***69*** (93) ***99*** (107) ***111*** (116) ***118*** (121) ***123*** (127) ***130*** (138) ***156*** (183) ***191***	30	28.61	
Moon (256 × 256)	2	***105*** (128) ***146***	16	22.90	170 ms,
	4	***75*** (101) ***116*** (129) ***141*** (154) ***165***	21	26.89	50 min
	6	***75*** (101) ***108*** (116) ***122*** (129) ***136*** (143) ***148*** (154) ***165***	27	28.04	
	8	***75*** (101) ***108*** (116) ***118*** (122) ***124*** (128) ***132*** (136) ***139*** (143) ***148*** (154) ***165***	32	28.26	
Lake (512 × 512)	2	***66*** (125) ***184***	94	18.77	156 ms,
	4	***47*** (62) ***86*** (129) ***169*** (187) ***207***	156	23.33	52 min
	6	***47*** (62) ***71*** (86) ***104*** (130) ***155*** (173) ***180*** (187) ***207***	219	24.53	
	8	***47*** (62) ***71*** (86) ***92*** (112) ***113*** (128) ***143*** (157) ***165*** (173) ***180*** (187) ***207***	282	24.79	

The above algorithm thus ensures that the whole image is segmented effectively based on different thresholds found at each stage using simple parameters like mean and standard deviation. Replacing the pixels within a sub-range by a single value leads to enhanced contrast. On the other hand, choosing the weighted mean of a class as the replacement value ensures that intra-class variance of sub-ranges is minimum leading to increased PSNR and quality of image. We also find that for some images, structures can be better extracted if we take the middle value of the sub-range in place of the mean; however this reduces the PSNR value. We see that the sub-range size is large at greater distance from the mean and reduces as we approach the mean. By doing this we are able to zoom in to the region of interest very fast and are able to extract features around mean efficiently. To ensure that a sub-range does not span two different structures or a single structure does not extend beyond a sub-range, the sub-range span is varied by changing the control parameter κ at each step. This leads to detailed feature enhancement by preventing clustering of different major structures within a sub-range. Skew in the asymmetric distributions can also be taken care of by using κ_1 and κ_2 parameters in each step.

4. Analysis

We theoretically evaluate the performance of the proposed algorithm and compare it with conventional Otsu's method for multilevel thresholding. For segmenting an image with G gray levels using n thresholds, Otsu's exhaustive search method searches ${}^{G-1}C_n$ different combinations of thresholds, which can be approximated to $\mathrm{O}(G^n)$ for $n \ll G$. However, to find thresholds using our recursive multilevel algorithm, we first calculate the mean of G gray levels which consumes time $\mathrm{O}(G)$ this is followed by calculation of variance which again takes time $\mathrm{O}(G)$. Due to this the time consumed at each level is also of $\mathrm{O}(G)$; note that two thresholds are calculated at each step. As we calculate newer thresholds, the number of gray levels considered at each step becomes progressively smaller. Thus the time taken for finding n thresholds is always $< \mathrm{O}(nG)$ and hence the computation is done in polynomial time, which is much less than Otsu's exhaustive search method. A significant advantage in computation time becomes evident from Table 1.

5. Results and observations

For evaluating the performance of the proposed algorithm, we have implemented the method on a wide variety of images. The performance metrics for checking the effectiveness of the method are chosen as computational time so as to get an idea of the complexity, and PSNR which is used to determine the quality of the thresholded image. The test images were chosen so as to rigorously test the algorithm for different histogram distributions. Fig. 1 shows some of the histograms used for experiments, starting from near Gaussian to asymmetric and ones having multiple mode. Apart from showing the efficacy of our algorithm, the latter ones illustrate the usability of κ-

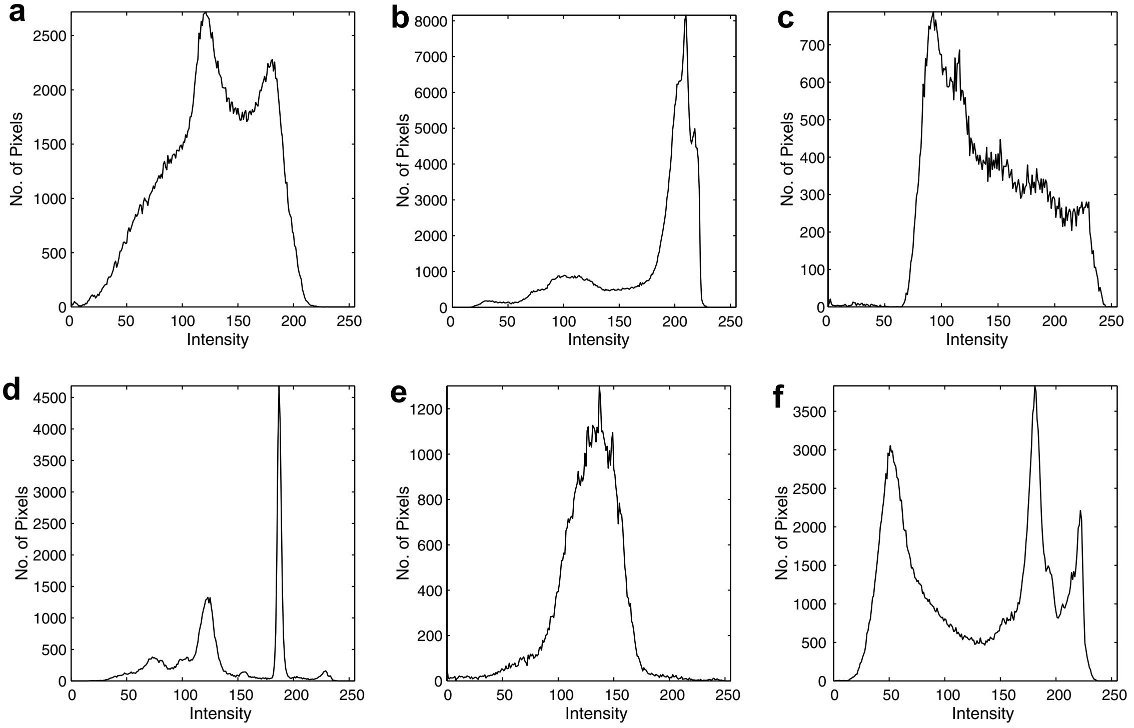

Fig. 1. Histogram distributions: (a) baboon, (b) jet, (c) ariel, (d) house, (e) moon and (f) lake.

parameters. First of all asymmetric distributions are assigned different segmentation windows around mean through κ parameter. Multiple-peak histogram domains are then naturally segmented into separate ones, since weighted average is considered for determining the threshold.

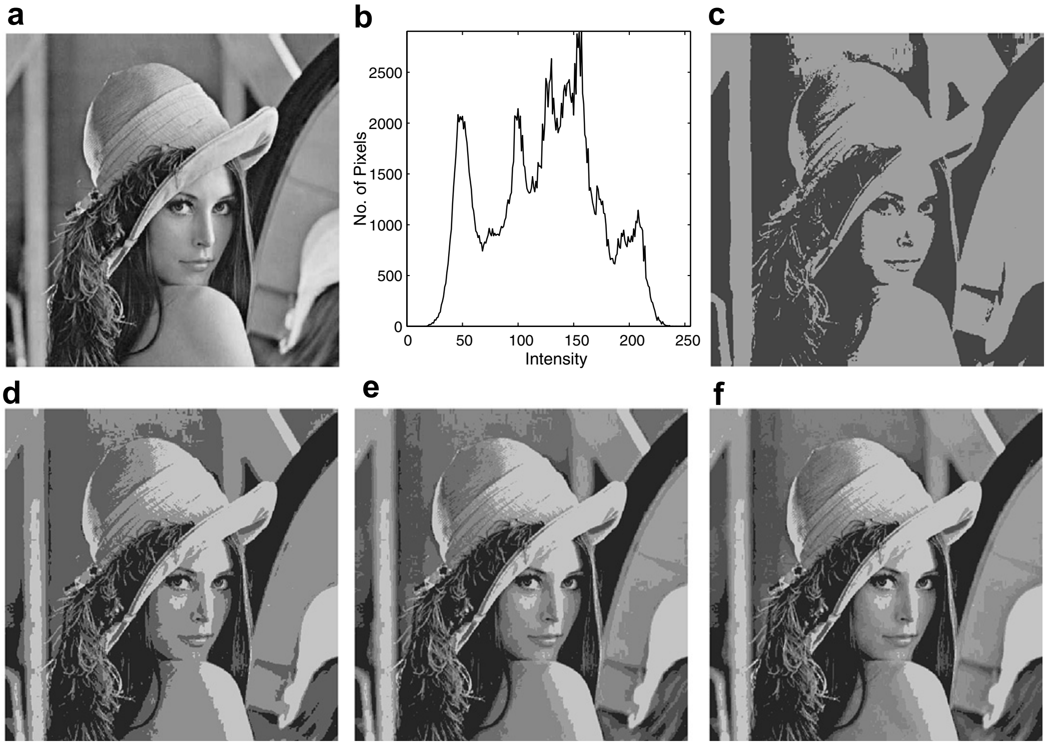

Fig. 2. Results: Lena (a) Lena gray, (b) histogram, (c) 2 level thresholding, (d) 4 level, (e) 6 level and (f) 8 level.

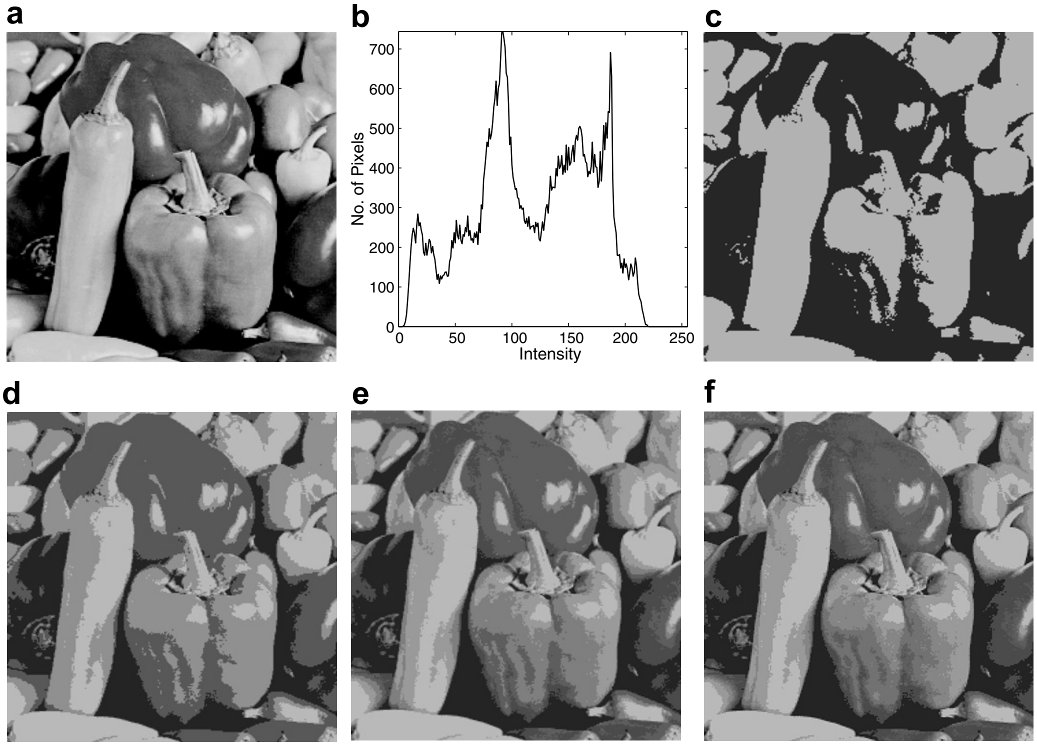

Fig. 3. Results: peppers (a) peppers gray, (b) histogram, (c) 2 level thresholding, (d) 4 level, (e) 6 level and (f) 8 level.

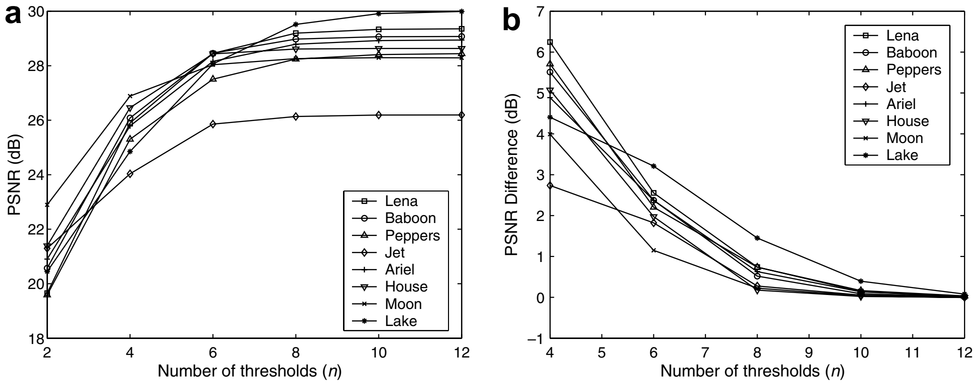

Fig. 4. Plots of (a) PSNR and (b) PSNR difference, of various test image vs. n, illustrating saturation of PSNR within a few iterations providing a criterion for optimum number of thresholds.

The results of two test images popular in image processing literature are shown in Figs. 2 and 3. As we increase the number of thresholds, the thresholded image rapidly tends towards the original image, from the visual point of view.

Thresholds, sub-ranges' boundaries, computational times and PSNR values for the test images at different number of thresholds are shown in Table 1. Also the table provides the comparison in computational time of the proposed algorithm with Otsu's exhaustive search method.

For simplicity, we have chosen the value of κ equal to 1 for all the cases, apart from the one where κ values are explicitly mentioned. κ parameter can be used to effectively threshold images with asymmetric or skewed histograms, as in the case of image 'Jet', whose histogram is shown in Fig. 1b. Using $\kappa_1 = 0.6$ and $\kappa_2 = 1.7$, significant increase in the PSNR is observed (as listed in Table 1). κ parameter shifts the sub-range to handle the skewness and improves the quality of thresholded image with insignificant overhead in terms of computational time.

For most of the images, we observed that (as shown in Fig. 4) the PSNR rapidly saturates after a few iterations, providing a criterion for selecting the number of iterations.

It has been observed, Liao et al. (2001), that Otsu's recursive method takes much larger time to calculate multi-thresholds, while our method gives same number of thresholds in relatively less time. From the above results we observe that the algorithm not only segments the image effectively, but also is computationally very fast. We observe that the blocking is non-uniform and the block size reduces near the mean. This gives rise to sharp boundaries and an increase in the contrast.

6. Conclusion

A method has been proposed that uses mean and variance of pixel distribution to naturally provide a non-uniform multi-segmentation scheme, ideally suited for human perception. The extreme pixel values are coarse grained in a broader interval as compared to the pixel value distribution around the mean. The procedure naturally adapts to distributions having non-zero higher moments like skew and is quite fast to implement. The recursive procedure converges rapidly as is seen from the quick saturation of the PSNR in variety of images.

References

Abutaleb, A.S., 1989. Automatic thresholding of gray level pictures using two-dimensional entropy. Comput. Vision Graphics Image Process. 47, 22–32.

Boukharouba, S., Rebordao, J.M., Wendel, P.L., 1985. An amplitude segmentation method based on the distribution function of an image. Comput. Vision Graphics Image Process. 29, 47–59.

Chang, C.C., Wang, L.L., 1977. A fast multilevel thresholding method based on lowpass and highpass filtering. Pattern Recognition Lett. 18, 1469–1478.

Feng, M.L., Tan, Y.P., 2004. Adaptive Binarization method for document image analysis. In: IEEE Internat. Conf. on Multimedia and Expo (ICME).

Huang, Q., Gai, W., Cai, W., 2005. Thresholding technique with adaptive window selection for uneven lighting image. Pattern Recognition Lett. 26, 801–808.

Kapur, J.N., Sahoo, P.K., Wong, A.K., 1985. A new method for gray-level picture thresholding using the entropy of the histogram. Comput. Vision Graphics Image Process. 29, 273–285.

Kittler, J., Illingworth, J., 1986. Minimum Error Thresholding. Pattern Recognition 19, 41–47.

Liao, P.S., Chen, T.S., Chung, P.C., 2001. A fast algorithm for multilevel thresholding. J. Inform. Sci. Eng. 17, 713–727.

Niblack, W., 1986. An Introduction to Digital Image Processing. Prentice Hall, Englewood Cliffs, NJ, pp. 115–116.

Otsu, N., 1979. A threshold selection using gray level histograms. IEEE Trans. Systems Man Cybernet. 9, 62–69.

Papamarkos, N., Gatos, B., 1994. A new approach for multilevel threshold selection. Graphics Models Image Process. 56, 357–370.

Pun, T., 1980. A new method gray-level picture thresholding using the entropy of the histogram. Signal Process. 2, 223–237.

Reddi, S.S., Rudin, S.F., Keshavan, H.R., 1984. An optical multiple threshold scheme for image segmentation. IEEE Trans. System Man Cybernet. 14, 661–665.

Ridler, T.W., Calward, S., 1978. Picture thresholding using an iterative selection method. IEEE Trans. Systems Man Cybernet. 8, 630–632.

Sahoo, P.K., Soltani, S., Wong, A.K.C., 1988. Survey: A survey of thresholding techniques. Comput. Vision Graphics Image Process. 41, 233–260.

Sezgin, M., Sankur, B., 2004. Survey over image thresholding techniques and quantitative performance evaluation. J. Electron. Imaging 13 (1), 146–165.

Tseng, D.A., Huang, M.Y., 1993. Automatic thresholding based on human visual perception. Image Vision Comput. 11, 539–548.

Wang, Q., Chi, Z., Zhao, R., 2002. Image thresholding by maximizing of non-fuzziness of the 2D grayscale histogram. Computer Image and Vision Understanding 85, 100–116.

Whatmough, R.J., 1991. Automatic threshold selection from a histogram using the exponential hull. Graphics Models Image Process. 53, 592–600.

Wu, S., Amin, A., 2003. Automatic thresholding of gray-level using multi-stage approach. ICDAR 38, 493–497.