



Title	An iterative image segmentation algorithm utilizing spatial information
Author(s)	Leung, CK; Lam, FK
Citation	IEEE Region 10 Digital Signal Processing Applications Proceedings, Perth, Western Australia, 26-29 November 1996, v. 1, p. 141-146
Issued Date	1996
URL	http://hdl.handle.net/10722/46009
Rights	Creative Commons: Attribution 3.0 Hong Kong License

An Iterative Image Segmentation Algorithm Utilizing Spatial Information

*LEUNG, C.K. and **LAM, F.K.

*Dept. of Electronic Engineering, The Hong Kong Polytechnic University, Hong Kong.

**Dept. of Electrical & Electronic Engineering, The University of Hong Kong, Hong Kong.

ABSTRACT: *An iterative image segmentation algorithm that segments an image on a pixel-by-pixel basis is described. The observation information to be utilized is the joint gray level values of the pixel to be segmented and those of its neighborhood pixels. The iterative process is initialized by thresholding the image with Otsu's method. Each pixel is segmented into a class when the a posteriori probability, conditioned on the observation information, that it belongs to this class is maximum. The newly segmented image is employed to re-estimate the a posteriori probabilities and the segmentation process is repeated until there is no further pixel classification change in a particular run. Among those segmented images generated in the iterative process, the best segmented image is chosen according to a maximum entropy criterion. Simulation studies demonstrate that the proposed algorithm can achieve very significant improvement in segmentation performance as compared to the more popular thresholding approach. Furthermore, the performance is neither sensitive to the initial threshold value nor the form of the probability density function of the image. Segmentation of practical images also demonstrates that the proposed algorithm is capable of good segmentation results for real-life images.*

1. INTRODUCTION

An image is a collection of spatially-ordered picture elements (pixels) representing a scene. For the consideration of an image segmentation process, the scene usually consists of several objects in a background and the segmentation process will classify, or segment, each pixel into a certain class. Such a classification process is an important step towards higher level image processing including image understanding, object identification and pattern recognition [1]. A very popular method of image segmentation is thresholding. In this method, a threshold value is determined by analyzing the image. A pixel is segmented into one class if its gray level value is greater than the threshold value;

otherwise it is segmented into another class. Multi-class thresholding is defined in a similar manner. There exist various algorithms for the determination of the thresholding value [2-4].

An important advantage of thresholding is that the threshold value can be derived in high speed if only simple statistical properties, such as pixels' probability density function (pdf), of the image are considered [5]. But since an image pdf does not contain sufficient amount of information to characterize the image faithfully, a pdf-based thresholding algorithm may not always arrive at a good threshold value. A possible improvement is to make use of more information, such as spatial information, to determine the threshold value. In recent years, several new thresholding methods that utilize spatial information in determining the threshold value have been reported [6-8]. Another limitation of thresholding lies in its fundamental characteristics and hence is more difficult to overcome. After a threshold value has been determined, all pixels having the same gray level value will invariably be segmented into the same class. The segmentation result is limited by the degree of overlap of the pdf's of the constituent sub-images irrespective of how the threshold value is derived. When the degree of overlap is small, good thresholding results are still possible; if the degree of overlap is substantial, it is impossible to obtain good thresholding results.

To overcome these limitations, it would be necessary to segment each image pixel individually according to some observation information. For this purpose, a pixel-by-pixel segmentation algorithm that utilizes spatial information is proposed and described in this paper. In Sections 2 to 4, the notations, theory and implementation of the proposed algorithm are described in details. In Section 5, the simulation and practical results are presented to illustrate the capability of the proposed algorithm. The paper is then ended with Section 6, the discussions and conclusions part.

2. NOTATIONS

Consider the general K -class segmentation problem. The image to be segmented is a representation of a scene having $K-1$ objects in a background, i.e. there are totally K constituent sub-images with each sub-image generating a class of image pixels. Each class of pixels is characterized by a probability density function (pdf). An image pixel is denoted as X . A class parameter x is defined to denote the fact that X belongs to class k by $x=k$, $k=1,2,\dots,K$. If the gray level value of X is g , it is denoted as $X=g$, $g=0,1,\dots,L$, i.e. the gray level is quantized to $L+1$ values uniformly spaced in unit steps. The pdf of the k^{th} sub-image is denoted as $f_k(g)$:

$$f_k(g) = P(X=g | x=k) \quad k = 1, 2, \dots, K \quad (1)$$

For each image pixel X , a collection of M pixels bearing a fixed spatial relationship to X are defined as "neighborhood pixels" of X . These neighborhood pixels are denoted as $X^{(1)}, X^{(2)}, \dots, X^{(M)}$. Such a fixed spatial relationship is usually defined in a neighborhood scheme. As an example, Fig. 1 below shows the nearest 4-neighbors scheme as defined in [9].

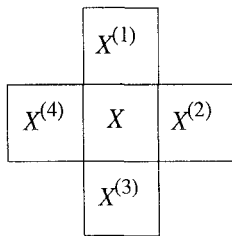


Fig.1 Nearest 4-neighbors neighborhood scheme

The class parameter of the neighborhood pixel $X^{(m)}$ is denoted as $x^{(m)}$. The M class parameters of the neighborhood pixels when taken jointly are defined as "neighborhood configuration" $N = (x^{(1)}, x^{(2)}, \dots, x^{(M)})$. In a similar manner, the gray level value of the neighborhood pixel $X^{(m)}$ is denoted as $g^{(m)}$ and the M gray level values of the neighborhood pixels are taken jointly as $G = (g^{(1)}, g^{(2)}, \dots, g^{(M)})$.

3. THEORY

To segment an image pixel X , the joint observation data (g, G) will be utilized. Based on this observation data, the *a posteriori* probability that pixel X belongs to class k is denoted as P_k and can be expressed as a conditional probability:

$$P_k = P(x=k | g, G) \quad k = 1, 2, \dots, K \quad (2)$$

Applying Bayes' rule [10] to Eq. (2), the *a posteriori* probability P_k can be written as:

$$P_k = \frac{P(g, G | x=k) \times P(x=k)}{P(g, G)} \quad k = 1, 2, \dots, K \quad (3)$$

The most probable class that X belongs to is class k^* where P_{k^*} is the maximum among all the P_k values described in Eq. (3). If pixel X is segmented into class k^* , the probability of segmentation error will be minimum. To determine the maximum P_k value, there is no need to consider the denominator $P(g, G)$ in Eq. (3) since this term is independent of k . Only the products of $P(g, G | x=k)$ and $P(x=k)$ need to be computed and compared.

On account of the extremely large dimension of $P(g, G | x=k)$, a confident estimation of their values could be quite difficult. For instance, with an image of 65,536 pixels, a 4-pixel neighborhood and 8 bits of gray level quantization, there are 2^{41} values of $P(g, G | x=k)$ to be estimated. It is impossible to make the estimation with only 65,536 image observation data available. Such a dimensionality problem has challenged many segmentation algorithms intending to incorporate spatial information. Possible solutions that have been reported earlier include simulated annealing, iterated conditional modes (ICM) or maximizer of posterior marginals (MPM) [11].

To overcome the dimensionality problem associated with the estimation of the $P(g, G | x=k)$ values, it is proposed to retain only the **class** inter-dependence feature among the image pixels and to neglect the gray level inter-dependence feature. In essence, the following two assumptions are made:

- (i) the probability that a pixel has a gray level value g only depends on its class parameter, and does not depend on the gray level values of its neighborhood pixels,
- (ii) the probability that a pixel has a neighborhood configuration N only depends on its class parameter, and does not depend on its gray level value.

For a K -class segmentation problem with M neighborhood pixels, there are K^M possible neighborhood configurations each of which may be denoted as N_i , $i=1,2,\dots,K^M$. Each N_i will be denoted as $(x_i^{(1)}, x_i^{(2)}, \dots, x_i^{(m)}, \dots, x_i^{(M)})$ where each $x_i^{(m)}$ is a class parameter which value ranges from 1, 2, \dots , K . Due to assumption (ii), if it is known that pixel X belongs to class k , then the probability of its neighborhood configuration being N_i will be given by the conditional probability $P(N_i | x=k)$. Since all the possible neighborhood configurations N_i are mutually exclusive, the total

probability theorem [10] may be applied and the value of $P(g, \mathbf{G} | x=k)$ is given as:

$$P(g, \mathbf{G} | x=k) = \sum_{i=1}^{K^M} P(N_i | x=k) \times P(g, \mathbf{G} | x=k, N_i) \quad (4)$$

Since the observation data is (g, \mathbf{G}) where \mathbf{G} is $(g^{(1)}, g^{(2)}, \dots, g^{(M)})$, with assumption (i) the term $P(g, \mathbf{G} | x=k, N_i)$ in Eq. (4) can be expressed as a product:

$$P(g, \mathbf{G} | x=k, N_i) = f_k(g) \times \prod_{m=1}^M f_{(m)}(g^{(m)}) \quad (5)$$

By substituting Eq. (5) into Eq. (4), $P(g, \mathbf{G} | x=k)$ can be written as:

$$P(g, \mathbf{G} | x=k) = \sum_{i=1}^{K^M} P(N_i | x=k) \times f_k(g) \prod_{m=1}^M f_{(m)}(g^{(m)}) \quad (6)$$

To calculate all the values of $P(g, \mathbf{G} | x=k)$ according to the right hand side of Eq. (6), the values of $P(N_i | x=k)$ and $f_k(g)$ need to be known for all i, k and g . There are K^{M+1} values of $P(N_i | x=k)$ and $k(L+1)$ values of $f_k(g)$ to be estimated. These numbers are usually not very large for most practical images. For instance, in the case of binary segmentation of an image of 65,536 pixels and 256 gray levels employing a 4-pixel neighborhood scheme, only 32 values of $P(N_i | x=k)$ and 512 values of $f_k(g)$ are to be estimated. The dimensionality problem is now solved. The specific estimation procedures for a practical segmentation algorithm will be described in Section 4.

It may be shown that the proposed segmentation method degenerates to a purely pdf-based thresholding method when the image does not have any spatial inter-dependence feature. In this case all the $P(N_i | x=k)$ terms in Eq. (6) are identical and the determination of the maximum $P(g, \mathbf{G} | x=k) \times P(x=k)$ value is equivalent to determining the maximum $f_k(g) \times P(x=k)$ value. This is equivalent to the purely pdf-based maximum likelihood thresholding scheme where no spatial information of the image is taken into account [12].

4. IMPLEMENTATION

A binary iterative segmentation algorithm, known as the ITSEG algorithm, is implemented with the same nearest 4-neighbor neighborhood scheme as that defined in Fig.1. The ITSEG algorithm is initialized by thresholding the gray-scale image with Otsu's method [14]. This method is chosen for its popularity and its high

speed of operation. It will be shown that the final segmentation result is fairly insensitive to the initial thresholding result and hence any good thresholding method may be used.

Based on the initial thresholded image, the values of $P(x=k)$, $f_k(g)$ and $P(N_i | x=k)$ are estimated. The value of $P(x=k)$ for $k=1, 2$ is estimated by counting the number of pixels segmented into class k and dividing it by the total number of image pixels. To estimate the values of $f_k(g)$ for $k=1, 2$, it is assumed that $f_k(g)$ has a quasi-Gaussian form parameterized by its mean μ_k and standard deviation σ_k [12, 13], as in the following:

$$f_k(g) = \frac{1}{\sigma_k \sqrt{2\pi}} \exp \frac{-(g-\mu_k)^2}{2\sigma_k^2} \quad k=1, 2 \quad (7)$$

With this assumption, the estimation of $f_k(g)$ will be replaced by estimating μ_k as the sample mean, and σ_k as the sample standard deviation, of the pixels that have been segmented into class k . It will be shown that even if the actual pdf's $f_k(g)$ are not quasi-Gaussian, this method of estimation is still applicable. Finally, consider all those pixels that have been segmented into class k . The value of $P(N_i | x=k)$ can be estimated as the portion of these pixels that have a neighborhood configuration N_i .

With all the values of $P(x=k)$, $f_k(g)$ and $P(N_i | x=k)$ estimated, for each pixel with observation data (g, \mathbf{G}) the maximum value of $P(g, \mathbf{G} | x=k) \times P(x=k)$ is determined and the pixel segmented accordingly. After all the pixels have been segmented, a new segmented image is obtained and the segmentation procedure is repeated. This iterative process will come to a termination when no pixel classification change is resulted in a particular run. A series of segmented images would have been produced in the course of iteration. To identify the best segmented images from the series, the entropy measure H is calculated for each segmented image [16], as in the following:

$$H = - \sum_{g=0}^L P(g|1) \log P(g|1) - \sum_{g=0}^L P(g|2) \log P(g|2) \quad (8)$$

where $P(g|k)$ is the pdf of the pixels that have been segmented into class k , $k=1, 2$. The segmented image that has the maximum H value is taken to be the best segmented image, and hence the final result. A flow-chart describing the ITSEG algorithm is as shown in Fig.2.

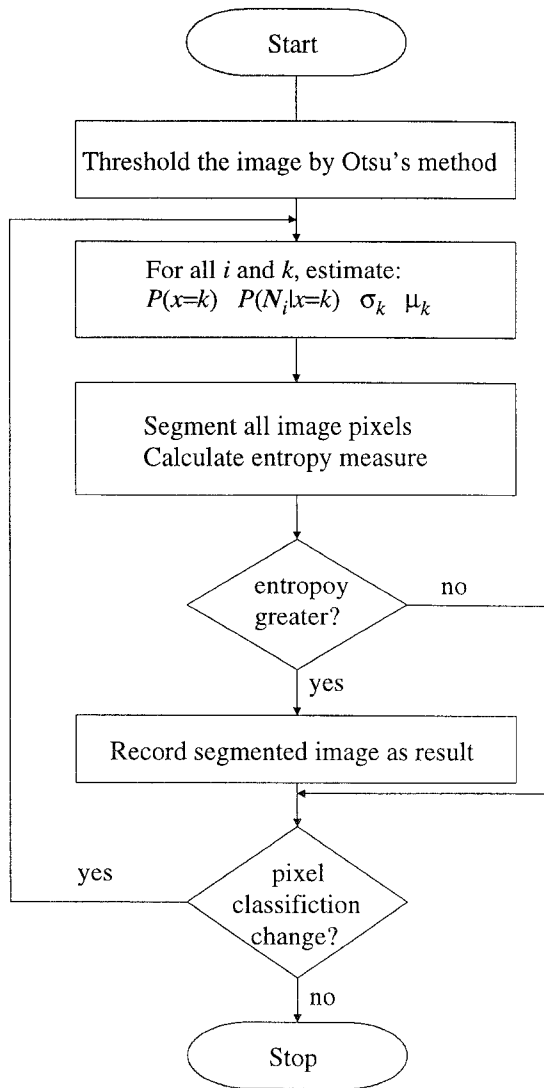


Fig.2 Flowchart of ITSEG algorithm

5. SIMULATION RESULTS

To evaluate the performance of the ITSEG algorithm, a simulation study is carried out. In this study, images of some scenes are generated and then segmented by the ITSEG algorithm. Each image has a size of 256×256 pixels and the gray level value of each pixel is quantized to 8 bits, i.e. from 0 to 255. The pixel gray level values are generated by a random number subroutine following some pdf, such as Gaussian or exponential [15]. For performance comparison purpose, each synthetic image is thresholded by the ideal minimum-error thresholding (IDTH) method. The IDTH method makes use of the individual pdf's and the relative sizes of the constituent sub-images to derive the minimum-error threshold value. This is the best thresholding result obtainable and hence is an upper bound of the performance of all, known or unknown, thresholding algorithms. All the investigations are carried out in an IBM compatible personal computer with a 75MHz Pentium CPU.

A case for illustration is the segmentation of a synthetic image of a scene that is a single circular object in a background. The circular object has a size of 0.2 relative to the entire image and its pdf is quasi-Gaussian with mean and standard deviation equal to 100 and 20, respectively. The background pdf is also quasi-Gaussian with mean and standard deviation equal to 160 and 20, respectively. When thresholded by the IDTH method at a threshold value of 120, the segmentation error rate is 4.72%. When thresholded by Otsu's method at a threshold value of 135, the segmentation error rate is 8.82%. The ITSEG algorithm is then performed on the image and each iteration takes 2 minutes to run. The segmented image that has the maximum entropy measure is obtained after the fifth iteration. The ITSEG segmentation error rate is 0.09%. Table 1 tabulates the change in the ITSEG segmentation error rate as the iteration progresses.

Table 1 Iterative performance

Step No.	ITSEG Err (%)	Step No.	ITSEG Err (%)
0 (Otsu)	8.82	3	0.21
1	2.95	4	0.1
2	0.81	5 (ME)	0.09

The gray-scale image, its pdf, the Otsu-thresholded and the ITSEG-segmented images are shown in Fig.3 for illustration. This example illustrates clearly the capability of the ITSEG algorithm.

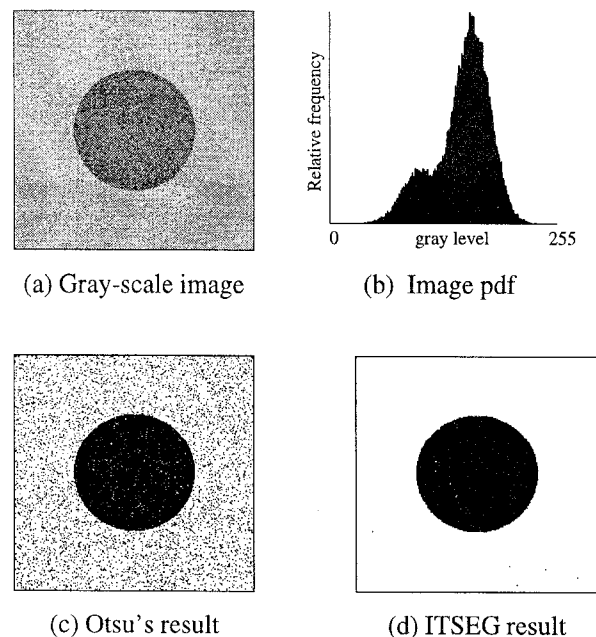


Fig.3 Segmentation results

To investigate the sensitivity of the ITSEG algorithm to the choice of initial threshold value, the same image is segmented once again with an initial threshold value of 150 instead of the Otsu threshold value of 135. This results in an initial segmentation error rate of 23.91%. After six iterations of the ITSEG algorithm, the best segmented image is obtained and the ITSEG segmentation error rate is 0.09%. This is the same value as when the initial threshold is set at the Otsu threshold value of 135. This case together with many other cases studied suggest that the ITSEG algorithm performance is fairly insensitive to the initial thresholding result.

To evaluate the ITSEG algorithm more thoroughly, a set of synthetic images are segmented. The synthetic images are characterized with a relative object size of 0.2, $\sigma_1=\sigma_2=20$ and $\mu_1=100$. By combining these parameters with a μ_2 value of 160, 140, 130 and 120 in turn, a set of images with means separation equal to 3, 2, 1.5 and 1 standard deviations have been generated. Images with Gaussian and exponential pdf's are generated. The scenes include single circle, double circles, triple circles in a background. The ITSEG results are compared against the IDTH results by plotting the ITSEG segmentation error rate against the IDTH segmentation error rate as shown in Fig.4.

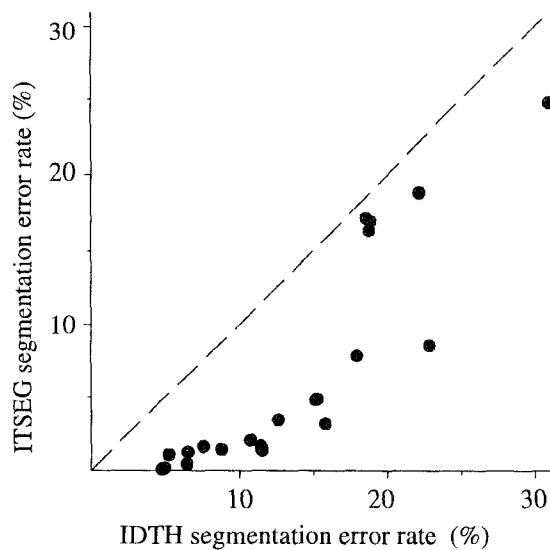


Fig.4 Overall performance of ITSEG

In Fig.4, each dot in the plot represents the segmentation results from one synthetic image. A dotted line representing the locus of ITSEG segmentation error rate being equal to IDTH segmentation error rate is also drawn. The region below the dotted line represents superior performance to the IDTH method while the region above it represents inferior performance, and hence it is in this upper region that the performance of

all practical thresholding methods, known or unknown, will lie. From the results in Fig.4, it is seen that the ITSEG algorithm performs much better than all thresholding methods.

To assess the performance of the ITSEG algorithm on real-life images that cannot be described by simple models, some practical images are segmented. A case for illustration is an image of a screw. The original gray scale image, its pdf, the Otsu-thresholded and the ITSEG-segmented images are shown in Fig.5. It can be seen that the ITSEG algorithm removes the “holes” in the Otsu-thresholded image, thus making the segmented image more suitable for object recognition, object identification or size determination.

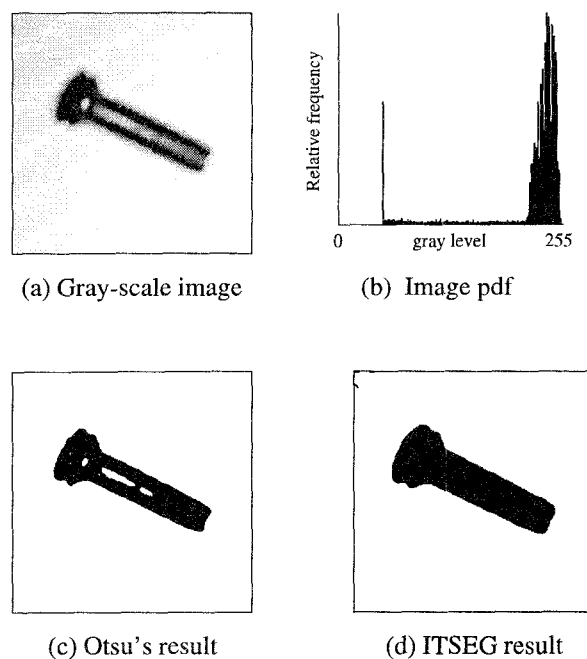


Fig.5 Real image segmentation results

7. DISCUSSIONS AND CONCLUSIONS

An iterative image segmentation algorithm that segments an image on a pixel-by-pixel basis utilizing spatial information has been implemented successfully. The spatial information to be utilized is the joint gray level values of the neighborhood pixels and the pixel to be segmented. The inter-dependence between the gray level values of the image pixels is neglected but the inter-dependence between their classes is retained. This results in a computationally feasible iterative segmentation algorithm. Segmentation results of synthetic images illustrate that the proposed algorithm performs substantially better than the thresholding

approach. Good segmentation results are also obtainable for real-life images. In general, this algorithm is worthy of consideration for segmenting images where long computation time can be tolerated.

REFERENCES

- [1] Rosenfeld, A. and Kak, A.: 'Digital Picture Processing', 2nd ed., (Academic Press, New York, USA, 1982)
- [2] Weszka, J.S.: 'A survey of threshold selection techniques', *Comp. Vis. Graph. Image Proc.*, 1978, **7**, pp. 259-265
- [3] Sahoo, P.K., Soltani, S., Wong, A.K.C. and Chen, Y.C.: 'A survey of thresholding techniques', *Comp. Vis. Graph. Image Proc.*, 1988, **41**, pp. 233-260
- [4] Lee, S.U. and Chung, S.Y.: 'A comparative performance study of several global thresholding techniques for segmentation', *Comp. Vis. Graph. Image Proc.*, 1990, **52**, pp. 171-190
- [5] Glasbey, C.A.: 'An analysis of histogram-based thresholding algorithms', *CVGIP: Graphical Models and Image Process.*, 1993, **55**, pp. 532-537
- [6] Beghdadi, A., Negrate, A.L. and Lesegno, P.V.D.: 'Entropic thresholding using a block source model', *CVGIP: Graph. Models and Image Proc.*, 1995, **57**, pp. 197-205
- [7] Brink, A.D., 'Minimum spatial entropy threshold selection', *IEE Proc. - Vis. Image Signal Process.*, 1995, **142**, (3), pp. 128-132
- [8] Leung, C.K. and Lam, F.K.: 'Maximum segmented-scene spatial entropy thresholding', Proceedings of the IEEE International Conference on Image Processing, 1996, Lausanne, Switzerland (to be published).
- [9] Besag, J., 'Spatial interaction and the statistical analysis of lattice systems', *Journal of the Royal Statistical Society*, series B, 1974, **36**, pp. 192-236
- [10] Papoulis, A.: 'Probability, random variables, and stochastic processes', 3rd ed., (McGraw Hill, New York, USA, 1991)
- [11] Dubes, R.C. and Jain, A.K.: 'Random field models in image analysis' in chapter 8 of 'Statistics and images: 1' ed. by Mardia, K.V. and Kanji, G.K. (Carfax Pub. Co., Abingdon, Oxfordshire, 1993)
- [12] Lam, F.K. and Leung, C.K.: 'Image segmentation using maximum entropy method'. Proceedings of the IEEE International Symposium on Speech, Image Processing and Neural Networks, 1994, Hong Kong, pp.29-32
- [13] Kurita, T., Otsu, N. and Abdelmalek, N.: 'Maximum likelihood thresholding based on population mixture models', *Pattern Recognition*, 1992, **25**, pp. 1231-1240
- [14] Otsu, N.: 'A threshold selection method from gray-level histograms', *IEEE Trans. Syst. Man Cybern.*, 1979, **SMC-9**, (1), pp. 62-66
- [15] Knuth, D.E.: 'The art of computer programming, vol. 2: seminumerical algorithms', 2nd ed., (Addison-Wesley, Massachusetts, USA, 1981)
- [16] Kapur, J.N., Sahoo, P.K. and Wong, A.K.C.: 'A new method for grey-level picture thresholding using the entropy of the histogram', *Comput. Vision Graphics Image Process.* 1985, **29**, pp. 273-285