Pergamon

Pattern Recognition, Vol. 31, No. 3, pp. 315–325, 1998

0031-3203/98$17.00 + .00

PII: S0031-3203(97)00045-9

A GAUSSIAN-MIXTURE-BASED IMAGE SEGMENTATION ALGORITHM

LALIT GUPTA and THOTSAPON SORTRAKUL

Department of Electrical Engineering, Southern Illinois University, Carbondale, IL 62901, U.S.A.

(*Received* 12 *October* 1995; *accepted* 24 *May* 1996)

Abstract—This paper focuses on the formulation, development, and evaluation of an autonomous segmentation algorithm which can segment targets in a wide class of highly degraded images. A segmentation algorithm based on a Gaussian-mixture model of a two-class image is selected because it has the potential for effective segmentation provided that the histogram of the image approximates a Gaussian mixture and the parameters of the model can be estimated accurately. A selective sampling approach based on the Laplacian of the image is developed to transform the histogram of any image into an approximation of a Gaussian mixture and a new estimation method which uses information derived from the tails of the mixture density is formulated to estimate the model parameters. The resulting selective-sampling–Gaussian-mixture parameter-estimation segmentation algorithm is tested and evaluated on a set of real degraded target images and the results show that the algorithm is able to accurately segment diverse images. © 1997 Pattern Recognition Society. Published by Elsevier Science Ltd.

Segmentation Histogram Thresholding Gaussian mixture

1. INTRODUCTION

This paper focuses on developing an autonomous segmentation algorithm which can segment targets in a wide class of degraded images. The autonomous segmentation of targets is one of the most complex problems in automatic target recognition (ATR) as the environment of the target is not controllable. Segmentation in the context of ATR problems is the process of separating a target from the background as accurately as possible. Because the subsequent stages of the ATR system depend upon the information provided by the segmentation stage, reducing segmentation error clearly reduces the probability of target misclassification. A multitude of image segmentation algorithms have been developed based on grey level thresholding,[1–13] edge detection,[11–16] region growing,[11,12,17–19] region splitting and merging,[11,12,20–22] relaxation,[11,12,23–25] and fuzzy set theory.[13,26–28] It is well known in image segmentation research that there is no single algorithm that can effectively segment all types of images. Additionally, the application of different algorithms to the same image generally produce different segmentation results. The selection of an appropriate segmentation algorithm depends largely on the type of images and the application areas. One approach is to select/design an effective algorithm for a class of images generated by a particular sensor. This approach is especially suitable for segmenting images generated in problems such as industrial inspection where the environment is controllable. The drawback to this approach when applied to segmenting images in problems such as ATR is that there is no control of the environment. The images, therefore, have varying characteristics and the results of segmenting some images may be seriously incorrect. Since the performance of the segmentation stage has a major influence on the performance of the ATR system, the goal is to focus on the formulation of an autonomous segmentation algorithm capable of accurately segmenting a wide class of highly degraded images.

The segmentation algorithm developed in this paper is based on a parametric model in which the probability density function of the gray levels in the image is a mixture of two Gaussian density functions. This model has received considerable attention in the development of segmentation algorithms and it has been noted that the performance is influenced by the shape of the image histogram and the accuracy of the estimates of the model parameters. The model-based segmentation algorithms perform poorly if the histogram of an image is a poor approximation of a mixture of two Gaussian density functions. It is especially poor when the two modes of the mixture density are not well-separated. The application of this model in image segmentation is, therefore, highly limited as the histograms of the images encountered in practical problems are seldom approximations of Gaussian mixtures with well-defined modes. Additionally, no method has been proposed to broaden the application of Gaussian-mixture-based segmentation to images that have histograms which are poor approximations of Gaussian mixtures. In order to broaden the application, a selective sampling method is proposed to transform the histogram of any image into

a histogram which approximates a Gaussian mixture. A new method is also proposed to accurately estimate the model parameters by first estimating the two Gaussian components of the mixture density and then estimating the parameters of the model from the two components.

2. THE GAUSSIAN MIXTURE MODEL FOR IMAGE SEGMENTATION

Gaussian-mixture-based segmentation is based on histogram thresholding which is one of the most frequently used methods for segmenting images. In histogram thresholding, it is assumed that an image consists of two regions or classes: target and background, with each region having a uni-modal gray-level distribution. The segmentation problem, therefore, involves determining a suitable threshold to partition the image into the target and background regions. Gaussian-mixture segmentation algorithms assume a parametric model in which the probability density function (PDF) of the gray levels in the image are a mixture of two Gaussian density functions with given means, standard deviations, and proportions. That is, it is assumed that an image with the target and background intensities is combined with Gaussian noise. The mixture probability density function of such an image can be written as[12]

$$p(z) = (P_1)p(z/C_1) + (P_2)p(z/C_2),$$

where

$$p(z/C_i) = (2\pi\sigma_i^2)^{-1/2}\exp[-(z-\mu_i)^2/(2\sigma_i^2)],$$

mixture component i, $i = 1, 2$

μ_i = mean of intensity level i, $i = 1, 2$

σ_i = standard deviation of intensity level i, $i = 1, 2$

P_i = prior probability of intensity level i, $i = 1, 2$;

$$(P_1 + P_2 = 1).$$

For a given threshold T, the overall probability of error is

$$E(T) = P_2E_1(T) + P_1E_2(T),$$

where

$$E_1(T) = \int_{-\infty}^{T} p(z/C_2)\,dz$$

and

$$E_2(T) = \int_{T}^{\infty} p(z/C_1)\,dz.$$

The threshold value for which $E(T)$ is minimum is given by solving

$$AT^2 + BT + C = 0, \qquad (1)$$

where

$$A = \sigma_1^2 - \sigma_2^2,$$

$$B = 2(\mu_1\sigma_2^2 - \mu_2\sigma_1^2),$$

$$C = \mu_2^2\sigma_1^2 - \mu_1^2\sigma_2^2 + 2\sigma_1^2\sigma_2^2\ln(\sigma_2P_1/\sigma_1P_2).$$

If the five model parameters P_1 (or P_2), μ_1, μ_2, σ_1, and σ_2 are known, the optimum threshold can be determined analytically from equation (1), however, if the parameters are unknown, they have to be estimated from the gray-level histogram of the image.

Several image segmentation algorithms based on estimating the parameters of the Gaussian mixture using curve-fitting techniques,[1,3] truncated histograms,[6,7] and the E–M algorithm[10] have been proposed. The algorithms based on curve-fitting techniques use a goodness-of-fit criterion through a conjugate gradient hill-climbing procedure to fit histograms with bimodal Gaussian density curves. The drawback with the curve-fitting methods is that they are computationally involved. The drawback with the iterative methods which estimate the model parameters from truncated histograms is that the estimation of parameters is biased even if the optimal threshold is selected. An improvement in estimating the variances when a threshold in an iteration is away from the overlapping regions of the mixture components is proposed in reference (7), however, the parameters are otherwise estimated as described in reference (6). The drawback of the E–M-based algorithm is that it tends to give poor estimates for small sample sizes and the time for convergence can be extremely high. Additionally, the E–M algorithm fails to converge if one or both variances approach zero as in the cases of targets and backgrounds with uniform intensities.

The performance of Gaussian-mixture-based segmentation algorithms clearly depend on how well the histogram of an image approximates a Gaussian mixture and the accuracy of the estimates of the model parameters. The histograms of real images seldom approximate a Gaussian mixture and none of the algorithms described propose methods to broaden the application of Gaussian-mixture-based segmentation to images with histograms that are a poor approximation of a Gaussian mixture. The focus of the segmentation algorithm developed in this paper, therefore, is to broaden the application of Gaussian mixture segmentation algorithms through a histogram transformation approach and to accurately estimate the mixture density parameters through a method specifically designed to estimate the mixture density parameters from the transformed histogram. The formulations of the selective sampling algorithm for histogram transformation, the Gaussian-mixture parameter-estimation algorithm, and the selective-sampling–Gaussian-mixture parameter-estimation image segmentation algorithm are described in the following sections.

3. THE SELECTIVE SAMPLING (SS) ALGORITHM

Several histogram modification methods for image segmentation have been proposed to facilitate the detection of the bottom of the valley between two modes by producing a transformed gray-level histogram which is more bimodal, or has a deeper valley between the modes, or converting the valley between the modes into a peak.[5,12] The approach developed in this paper focuses on selecting a set of pixels which are typical of the target and the background. Only the selected pixels contribute to the transformed histogram. The pixels selected are the set of pixels which most likely fall on the two sides of the boundary of a target. The Laplacian $L[f(x, y)]$ of the image $f(x, y)$ is used to identify and select such pixels.[12] The Laplacian is a second-derivative operator which gives high values to pixels on both sides of the edge (modelled as a step or a ramp function) and low-values to all other pixels. Pixels on the bright side of an edge have a negative Laplacian whereas pixels on the dark side of an edge have a positive Laplacian. The histogram of the selected non-zero Laplacian pixels will, therefore, be bi-modal with the 2 modes representing typical target and background gray-levels and the valley between the two peaks would also be well-defined. Additionally, the two peaks will be approximately equal in size because the number of pixels on both sides of the boundary would be approximately equal even if the target and the background have unequal areas.

The Laplacian of a continuous function $g(x, y)$ is defined as

$$L[g(x, y)] = [(\delta^2 g)/\delta x^2) + (\delta^2 g)/\delta y^2)].$$

For a discrete image $f(x, y)$, the Laplacian at (x, y) can be approximated by

$$L[f(x, y)] = f(x-1, y) + f(x+1, y) + f(x, y-1) + f(x, y+1) - 4[f(x, y)].$$

The Laplacian appears to be ideal for selecting pixels to generate a well-defined bi-modal histogram as it produces values with differing signs at each edge, thus, facilitating the identification of target and background pixels. Rules based on the sign and magnitude of the Laplacian can be easily developed to identify the target and background pixels whether an edge is modeled as a step or a ramp.

The problem with the Laplacian is that as a second-order derivative, it is highly sensitive to noise and produces high values not only at the boundary of the target but also in any region where sharp gray-level transitions occur. Therefore, the applicability of this method to highly degraded images is questionable. To overcome this problem, the pixel selection rule developed is based on using the Laplacian to accurately locate edges in noise and using the connectivity of the pixels to exclude pixels that may be noise pixels. Without a loss of generality, it will be assumed that an image is composed of a bright target on a dark background. In the first step, the sign transition in the Laplacian is used to determine which type of edge is encountered when the image rows and columns are scanned. A transition from plus to minus indicates a step edge transition from a background pixel to a target pixel whereas a transition from minus to plus indicates a step edge transition from a target pixel to a background pixel. Ramp edge transitions have similar sign transitions in the Laplacian along with a zero (or zeros) between the sign changes. Figure 1(a) illustrates the Laplacian computed for step and ramp edge models. The gray level is denoted by $f(x)$, the Laplacian by $L(x)$, and the location of the pixels selected are underlined. Note that the average value of the Laplacian is zero at an edge transition. As it is impossible to characterize all possible distortions which may occur in edges, only edges which are neither steps nor ramps but are monotonically increasing or decreasing are considered as distorted edges. A set of rules are developed to compute a more accurate Laplacian should such distortions occur in the edges. By first detecting an edge and then combining adjacent Laplacians, the average value $L(x)$ of the Laplacian at an edge transition can be made close to zero as shown by the underlined values in Fig. 1(b). The next step involves excluding pixels with low Laplacian magnitudes by comparing them with an empirically determined threshold. Such pixels typically result from noise and are not edge pixels. In the final step, the connectivity of the remaining pixels is determined by examining the eight-neighborhood of each pixel in order to exclude isolated pixels as the goal is to retain only boundary edge pixels which are generally connected to each other. The detailed steps of the selective sampling algorithm are described below:

Step 1: *Computation of the Laplacian of the image.* Compute the Laplacian $L[f(x, y)]$ of the image $f(x, y)$. Let the symbols 0, +, and − represent the zero, positive and negative Laplacian values of $L[f(x, y)]$ and let $n0$, $n+$, and $n-$ represent a string of n, $(n \geq 2)$ continuous zeros, positive, and negative Laplacians.

Step 2: *Detection of all edges in the image.* Scan the rows and columns of $L[f(x, y)]$ and use the following rules to detect edges:

(a) Noise-free step edge detection
A noise-free step edge is characterized by the sequence:

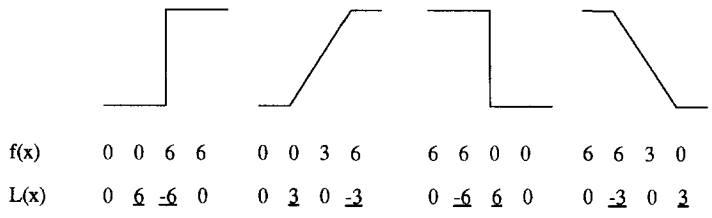
$$[\ldots 0, 0, (+, -), 0, 0 \ldots] \quad \text{or}$$

$$[\ldots 0, 0, (-, +), 0, 0 \ldots].$$

(b) Noise-free ramp edge detection
A noise-free ramp edge is characterized by the sequence:

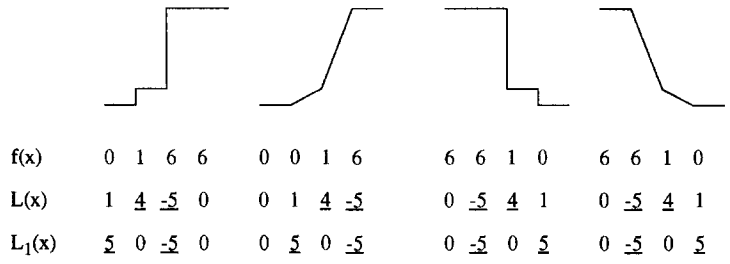
$$[\ldots 0, 0, (+, n0, -), 0, 0 \ldots] \quad \text{or}$$

$$[\ldots 0, 0, (-, n0, +), 0, 0 \ldots]$$

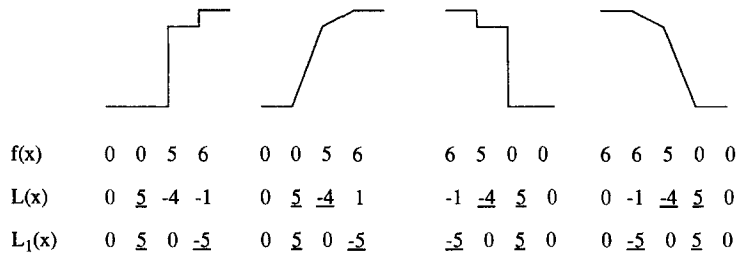


f(x)	0	0	6	6	0	0	3	6	6	6	0	0	6	6	3	0
L(x)	0	6	-6	0	0	3	0	-3	0	-6	6	0	0	-3	0	3

(a) Noise-free step and ramp edges.



f(x)	0	1	6	6	0	0	1	6	6	6	1	0	6	6	1	0
L(x)	1	4	-5	0	0	1	4	-5	0	-5	4	1	0	-5	4	1
$L_1(x)$	5	0	-5	0	0	5	0	-5	0	-5	0	5	0	-5	0	5



f(x)	0	0	5	6	0	0	5	6	6	5	0	0	6	6	5	0	0
L(x)	0	5	-4	-1	0	5	-4	1	-1	-4	5	0	0	-1	-4	5	0
$L_1(x)$	0	5	0	-5	0	5	0	-5	-5	0	5	0	0	-5	0	5	0

(b) Distorted edges.

Fig. 1. The Laplacian computed for noise-free and distorted edges.

(c) Detection of monotonically increasing/decreasing edges

A monotonically increasing edge is characterized by the sequence:

$$[\ldots 0,0,(n+,k0,m-),0,0,\ldots]$$

and a monotonically decreasing edge is characterized by

$$[\ldots,0,0,(m-,k0,n+),0,0,\ldots].$$

If a monotonically increasing edge is detected, the Laplacian is updated by summing the n positive and m negative Laplacians and storing them at the first occurrence and last occurrence of positive and negative values, respectively. All other Laplacians are set to zero. Similarly, for the monotonically decreasing edge, the Laplacian is updated by summing the m negative and n positive Laplacians and storing them at the first occurrence and last occurrence of negative and positive values, respectively. It is also possible to have sequences of zeros between a string of positive or negative Laplacians. For this case also, the strings of positive and negative Laplacians are summed and stored at the extreme locations.

Step 3: *Exclusion of small edges.* Generate an image $f_1(x, y)$ according to

$$f_1(x,y) = \begin{cases} 1 & \text{if } |L[f(x,y)]| > L_0, \\ 0 & \text{otherwise.} \end{cases}$$

L_0 is an empirically determined threshold.

Step 4: *Removal of isolated disconnected pixels.* Generate the transformed image $f_2(x, y)$ containing only the selected pixels according to

$$f_2(x,y) = \begin{cases} 0 & \text{if } [f_1(x,y) = 0] \\ & \text{or } [f_1(x,y) = 1 \\ & \text{and } \sum_{8N} f_1(x,y) < 2], \\ f(x,y) & \text{otherwise.} \end{cases}$$

$8N$ denotes the set of 8-connected pixels of the pixel at (x, y).

Step 5: *Histogram of* $f(x, y)$. Compute the histogram $h_2(z)$ of $f_2(x, y)$.

It is important to note that $f_2(x, y)$ is an image which contains the gray levels of the pixels selected from the original image $f(x, y)$. Given that, typically, only pixels on both sides of an edge transition are selected, the number of pixels used in computing the transformed histogram will be relatively small. The means of the target and background pixels can be expected to be well separated and the proportion of target and background pixels will be approximately equal irrespective of the target and background sizes.

4. THE GAUSSIAN-MIXTURE PARAMETER ESTIMATION (GMPE) ALGORITHM

The parameter-estimation algorithm focuses on the tails of the mixture density to first estimate the two Gaussian components of the mixture density and then estimate the parameters of the model from the two components. For a given threshold T_j, the notations used in the formulation are as follows:

$p(z)$ = mixture density

$p(z/C_i)$ = mixture component i, $i = 1, 2$

$P_i p(z/C_i)$ = weighted mixture component i, $i = 1, 2$

$p(z)_L | T_j$ = left truncated component

$p(z)_R | T_j$ = right truncated component

$L_i | T_j$ = left tail of weighted component

$P_i p(z/C_i)$, i, $i = 1, 2$

$R_i | T_j$ = right tail of weighted component

$P_i p(z/C_i)$, i, $i = 1, 2$.

The mixture density, the components, and the tails are illustrated in Fig. 2. The parameter estimation is done iteratively, and each iteration consists of three estimation steps. For a given threshold T_j, the parameters μ_1 and μ_2 are estimated initially using the truncated components $p(z)_L | T_j$ and $p(z)_R | T_j$. The next step involves estimating the weighted components $P_1 p(z/C_1)$ and $P_2 p(z/C_2)$ and the proportions P_1 and P_2. Finally, the threshold is estimated from the parameters of the estimated mixture components $p(z/C_1) | T_j$ and $p(z/C_2) | T_j$. In the formulation of the parameter-estimation algorithm, it is assumed that the contribution of tail $L_2 | T_j$ is negligible in tail $L_1 | T_j$ and that tail $R_1 | T_j$ is negligible in tail $R_2 | T_j$. Then, using the symmetrical property of the Gaussian function, the tails $R_1 | T_j$ and $L_2 | T_j$ can be estimated from the tails of the mixture $p(z)$. The detailed steps of the estimation algorithm are described next.

Step 0: *Selection of initial threshold* T_0. Let the initial estimate of the optimum threshold T_0 be μ_0, where μ_0 is the mean of $p(z)$.

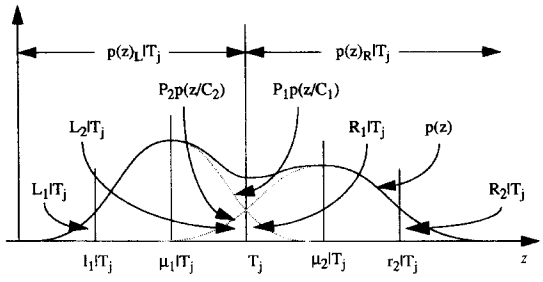


Fig. 2. Illustration of the mixture density, components, and tails.

Step 1: *Estimation of means.* At iteration j, $j = 1, 2, \ldots$, estimate the two means of the model from the left and right components $p(z)_L | T_{j-1}$ and $p(z)_R | T_{j-1}$ truncated by the threshold T_{j-1}. Let the estimates be $\mu_1 | T_{j-1}$ and $\mu_2 | T_{j-1}$.

Step 2: *Estimation of weighted mixture components.* Determine the positions of the mixture tails from

$$l_1 | T_{j-1} = 2(\mu_1 | T_{j-1}) - T_{j-1},$$

$$r_2 | T_{j-1} = 2(\mu_2 | T_{j-1}) - T_{j-1}.$$

Determine the tails L_1 and R_2 using

$$L_1 | T_{j-1} = p(z) U(-z + l_1 | T_{j-1}),$$

$$R_2 | T_{j-1} = p(-z + l_1 | T_{j-1} + T_{j-1}) U(z - T_{j-1}).$$

Estimate the tails L_2 and R_1 using

$$L_2 | T_{j-1} = p(-z + r_2 | T_{j-1} + T_{j-1}) U(-z + T_{j-1}),$$

$$R_1 | T_{j-1} = p(z) U(z - r_2 | T_{j-1}).$$

Estimates of the weighted components are given by

$$P_1 p(z/C_1) | T_{j-1} = p(z) U(-z + T_{j-1}) - L_2 | T_{j-1} + R_1 | T_{j-1},$$

$$P_2 p(z/C_2) | T_{j-1} = p(z) U(z - T_{j-1}) - R_1 | T_{j-1} + L_2 | T_{j-1}.$$

$U(z)$ denotes the step function.

Step 3: *Estimation of parameters.* Estimate the parameters P_1, μ_1, and σ_1 using

$$\hat{P} | T_{j-1} = \int_{-\infty}^{\infty} P_1 p(z/C_1) | T_{j-1} \, dz,$$

$$\hat{\mu}_1 | T_{j-1} = \int_{-\infty}^{\infty} z p(z/C_1) | T_{j-1} \, dz$$

$$\hat{\sigma}_1^2 | T_{j-1} = \int_{-\infty}^{\infty} (z - \hat{\mu}_1 | T_{j-1})^2 p(z/C_1) | T_{j-1} \, dz.$$

Similarly, estimate the parameters P_2, μ_2, and σ_2^2 using $P_2 p(z/C_2) | T_{j-1}$.

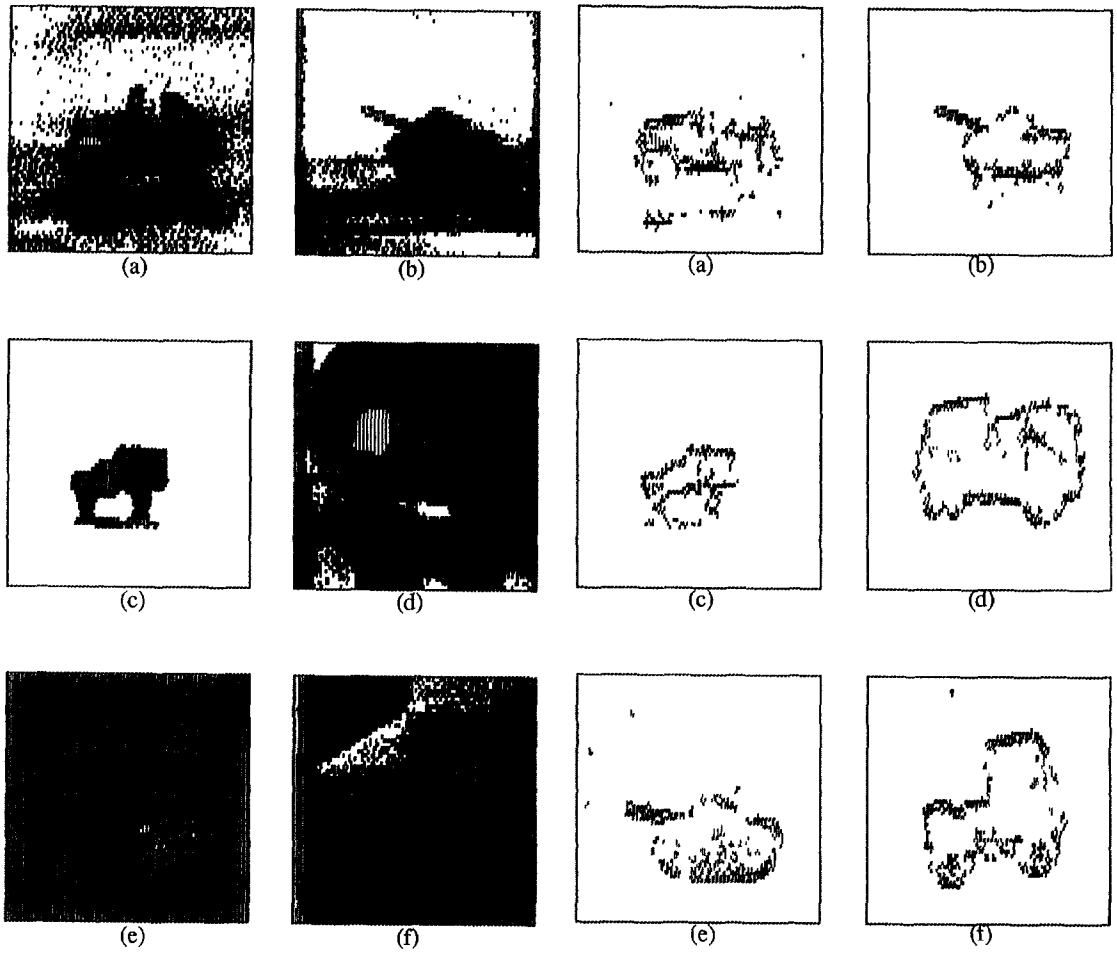


Fig. 3. The real infra-red images used in the experiments.

Fig. 4. The pixels selected by the SS algorithm.

Step 4: *Estimation of the threshold.* Use the parameter estimates from Step 3 in equation (1) to compute $\hat{T}_j$, the estimate of the optimal threshold.

Step 5: *Convergence test.* If $\hat{T}_j = T_{j-1}$, the algorithm has converged and the $\hat{T}_j$ is the final estimate of the optimum threshold. If not, let $T_{j-1} = \hat{T}_j$ and go to Step 1.

5. THE SELECTIVE SAMPLING–GAUSSIAN MIXTURE PARAMETER (SS–GMPE) SEGMENTATION ALGORITHM

The SS and GMPE algorithms are combined to formulate a segmentation algorithm which overcomes the general limitations of Gaussian-mixture-based image segmentation algorithms. The steps of the selective-sampling–Gaussian-mixture parameter-estimation algorithm are summarized below:

Step 1: *Histogram modification.* Transform image $f(x, y)$ to image $f_2(x, y)$ using the SS algorithm. Compute the histogram $h_2(z)$ of $f_2(x, y)$.

Step 2: *Parameter estimation.* Estimate the parameters of the mixture density from $h_2(z)$ using the GMPE algorithm. Estimate the optimal threshold using the estimated parameters.

Step 3: *Segmentation.* Use the estimate of the optimal threshold determined in Step 2 to segment image $f(x, y)$.

6. APPLICATION OF THE SS, GMPE, AND SS–GMPE ALGORITHMS

A series of experiments were designed to demonstrate and evaluate the performances of the SS algorithm in histogram transformation, the GMPE algorithm in estimating the parameters of a Gaussian mixture and the optimal threshold, and the SS–GMPE algorithm in image segmentation.

6.1. *Performance of the SS algorithm*

In order to demonstrate the performance of the SS algorithm, a set of experiments involved in computing the histograms of the 6 real 256-gray-level infra red (IR) images shown in Fig. 3 were designed. Through experimentation, the threshold L_0 used to retain prominent edges (or exclude small edges) was selected to be the 90th percentile of the values of $L[f(x, y)]$. The results of applying the SS algorithm to the images are shown in Fig. 4. The histograms of the original images and the histograms of the selected pixels

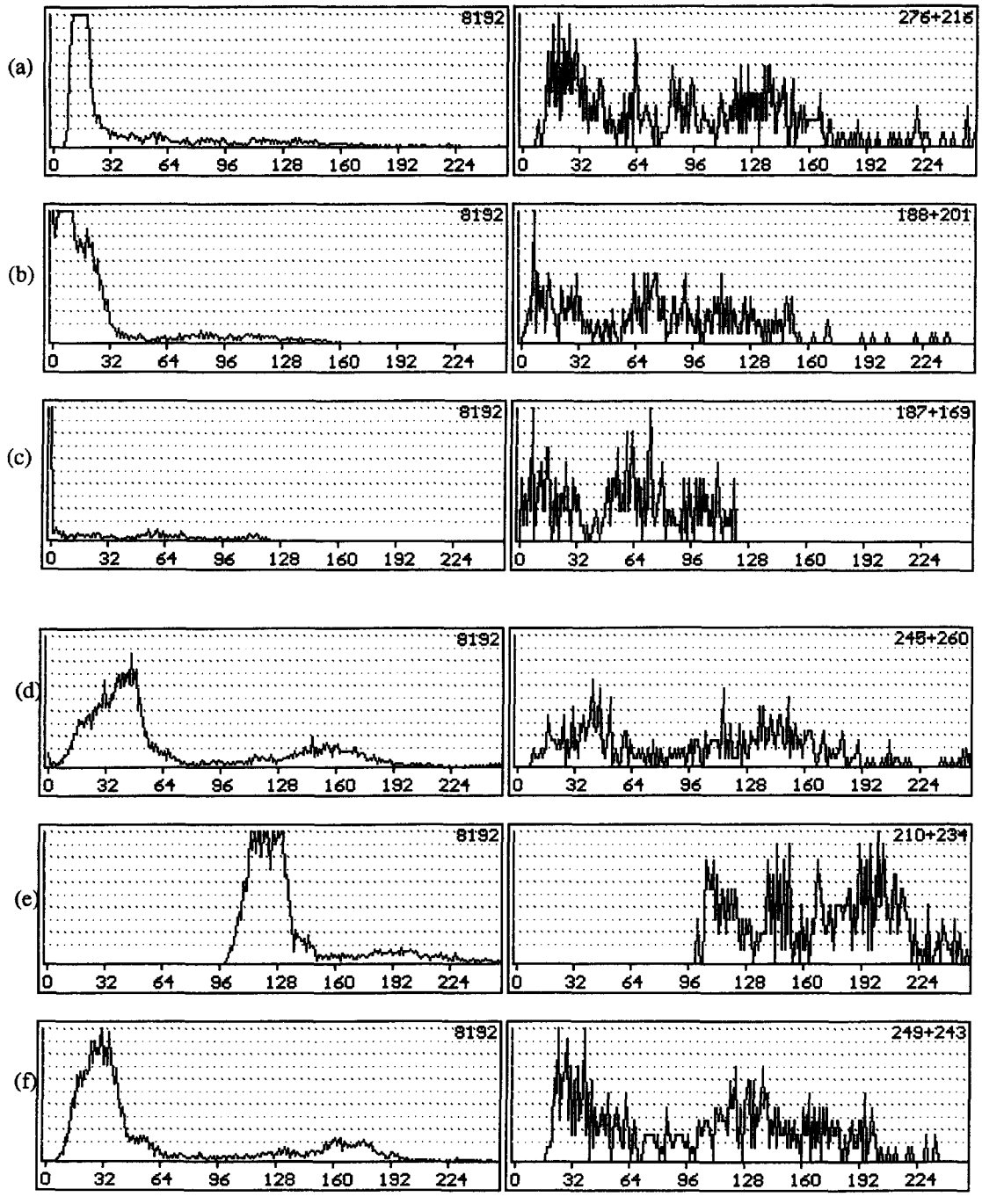


Fig. 5. Original histograms on the left and the transformed histograms on the right.

(transformed histograms) are shown in Fig. 5. The number of pixels used in computing the histograms are shown in the upper right corner of the plots. In the transformed histograms, the number of pixels is shown as $(x + y)$, where, x and y are the number of target and background pixels, respectively. By examining the results, the following can be concluded:

(a) The transformed histogram is computed from a small sample size.

(b) The majority of pixels selected are from the two sides of edge transitions.

(c) Irrespective of the target size, the number of target and background pixels are approximately equal.

(d) If smoothed, the transformed histogram could be considered to approximate a Gaussian mixture with well-separated modes.

6.2. *Performance of the GMPE algorithm*

The GMPE algorithm was tested on synthesized histograms approximating Gaussian-mixture probability

density functions. The means, variances, and proportions of the two components were specified and samples drawn from Gaussian distributions with the specified means and variances were used to generate the mixture histograms. Because the optimal threshold separating the target and background in a Gaussian-mixture density must be between the means of the components and the mean of mixture density also is between the means of the components, the initial threshold is selected to be the mean of the mixture density in Step 0 of the GMPE algorithm. This rule when compared to the random selection of an initial threshold as suggested in reference (6) or scanning the entire gray-level range in reference (7) was found to be efficient as it aided convergence by decreasing the number of iterations in the estimation of the parameters and optimal threshold.

The performance of the algorithm was measured in terms of the accuracy of estimating the known theoretically computed optimum threshold using the specified parameters in equation (1) and the accuracy of estimating the model parameters. The number of samples was varied in order to determine the dependence of the performance of the algorithm on the sample size. Numerous experiments with different combinations of parameters were conducted and typical results averaged over 100 repetitions are shown in Tables 1 and 2. The results in Table 1 is for the case of well-separated means and equal variances and the results in Table 2 is for the case of closer means and unequal variances. An estimate of a parameter θ is denoted by $\hat{\theta}$. By considering Table 1(a), the results can be interpreted as follows: The parameters μ_1, μ_2, σ_1^2, and σ_2^2 are set at 100, 160, 150, and 150, respectively, and the proportions P_1 and P_2 are varied as shown in columns 1 and 2. The estimates of the model parameters as computed by the GMPE algorithm are shown in columns 3 to 8. The last column contains the theoretical computed optimal threshold T and the estimate $\hat{T}$ of the optimal threshold as computed by the GMPE algorithm. From the results obtained it could be concluded that the GMPE algorithm gives accurate estimates of the model parameters and the optimal threshold for various combinations of model parameters even when the sample size is small which is crucial as the number of pixels selected by the SS algorithm is typically very small.

6.3. *Performance of the SS–GMPE segmentation algorithm*

In order to evaluate the segmentation accuracy, the SS–GMPE algorithm was tested on the 6 real infrared images shown in Fig. 3. Although the 6 images were captured using the same sensor, they represent a wide class of images. For example, the image in Fig. 3(c) has little detail and is relatively noise free. The images in Figs 3(a) and (b) are relatively noisy and have more detail. The image in Fig. 3(e) has poor contrast with little shadows. The images in Figs 3(d) and (f) have relatively well-defined targets. Additionally, the histograms of these images are quite diverse. The segmentation of these images are shown in Fig. 6. The initial threshold which is the mean of $h_2(z)$ is shown in the upper left and the optimal segmentation threshold as determined by the SS–GMPE algorithm is shown on the upper right of the segmentation results. From a careful examination of the results, it can be concluded that the SS–GMPE algorithm is able to accurately segment diverse images as the

Table 1. An example of model parameters and optimal threshold estimation

P_1	P_2	$\mu_1 = 100$	$\mu_2 = 160$	$\sigma_1^2 = 150$	$\sigma_2^2 = 150$	$\hat{P}_1$	$\hat{P}_2$	$T/\hat{T}$
		$\hat{\mu}_1$	$\hat{\mu}_2$	$\hat{\sigma}_1^2$	$\hat{\sigma}_2^2$			
(a) 250 samples								
0.30	0.70	100.36	160.36	152.65	149.92	0.30	0.70	128/128
0.40	0.60	99.90	159.96	141.09	145.19	0.40	0.60	129/129
0.50	0.50	99.75	159.83	148.75	148.58	0.50	0.50	130/130
0.60	0.40	100.31	160.11	146.95	135.58	0.60	0.40	131/131
0.70	0.30	99.53	159.48	148.84	137.67	0.70	0.30	132/132
(b) 500 samples								
0.30	0.70	100.22	160.20	148.70	145.97	0.30	0.70	128/128
0.40	0.60	100.35	160.31	151.32	146.95	0.40	0.60	129/129
0.50	0.50	99.75	159.86	147.77	148.86	0.50	0.50	130/130
0.60	0.40	100.22	160.30	149.95	148.22	0.60	0.40	131/131
0.70	0.30	100.09	159.99	149.53	147.20	0.70	0.30	132/132
(c) 1000 samples								
0.30	0.70	100.05	160.04	151.12	149.85	0.30	0.70	128/128
0.40	0.60	100.17	160.15	151.16	148.95	0.40	0.60	129/129
0.50	0.50	100.10	160.11	149.10	146.90	0.50	0.50	130/130
0.60	0.40	100.11	160.17	149.29	145.90	0.60	0.40	131/131
0.70	0.30	100.03	160.11	149.57	145.34	0.70	0.30	132/132

Table 2. Another example of model parameters and optimal threshold estimation

P_1	P_2	$\mu_1 = 100$	$\mu_2 = 140$	$\sigma_1^2 = 150$	$\sigma_2^2 = 200$	$\hat{P}_1$	$\hat{P}_2$	$T/\hat{T}$
		$\hat{\mu}_1$	$\hat{\mu}_2$	$\hat{\sigma}_1^2$	$\hat{\sigma}_2^2$			
(a) 250 samples								
0.30	0.70	105.38	142.75	148.55	165.73	0.35	0.65	115/121
0.40	0.60	101.07	140.14	148.00	149.43	0.43	0.57	117/117
0.50	0.50	100.50	140.42	147.72	148.27	0.53	0.47	119/120
0.60	0.40	99.96	139.97	127.40	174.61	0.60	0.40	121/121
0.70	0.30	100.24	141.17	147.94	178.50	0.70	0.30	123/123
(b) 500 samples								
0.30	0.70	104.17	141.73	153.57	134.33	0.37	0.63	115/118
0.40	0.60	101.45	141.12	152.62	160.08	0.44	0.56	117/120
0.50	0.50	100.12	140.16	140.00	179.58	0.51	0.49	119/120
0.60	0.40	100.08	140.11	146.72	157.90	0.60	0.40	121/121
0.70	0.30	100.32	141.00	149.23	155.05	0.70	0.30	123/123
(c) 1000 samples								
0.30	0.70	101.93	141.00	146.07	170.85	0.31	0.69	115/116
0.40	0.60	102.33	141.73	151.02	150.90	0.45	0.55	117/121
0.50	0.50	100.78	140.02	140.05	154.92	0.51	0.49	119/119
0.60	0.40	100.02	140.92	150.30	150.83	0.62	0.38	121/121
0.70	0.30	100.16	141.68	142.13	140.39	0.70	0.30	123/124

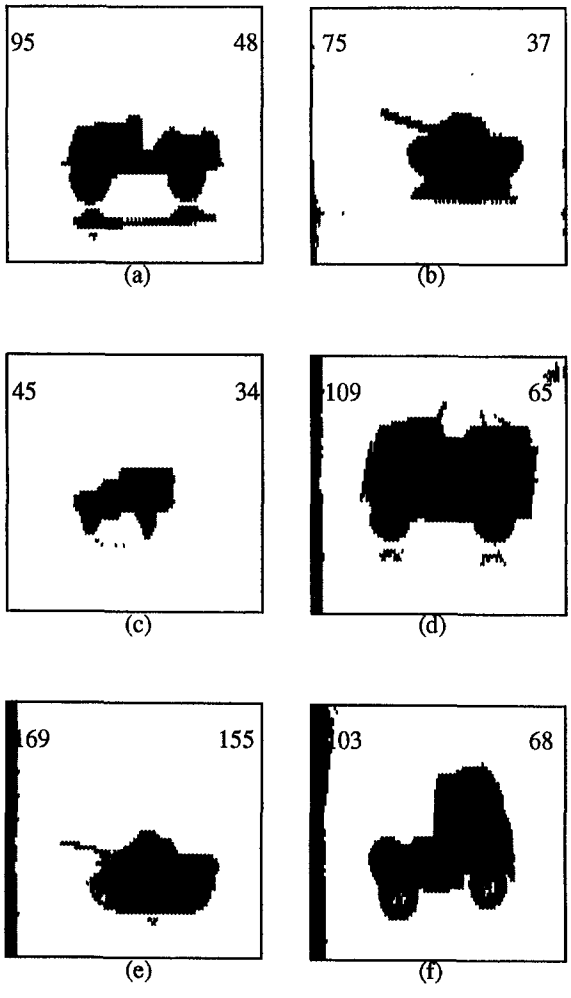


Fig. 6. SS–GMPE segmentation results.

targets are separated from the background while preserving the target features.

In order for the algorithm to be autonomous, the threshold L_0 used to retain prominent edges in the images has to be experimentally determined and preset from a collection of diverse reference images. The selection of one of the two possible solutions for the optimal threshold from equation (1) in Step 4 of the GMPE algorithm is also required for an autonomous implementation. Both solutions are considered in each iteration. The threshold which first satisfies the convergence test is selected as the optimal threshold. The experiments were conducted on a 486-DX 33 MHz machine and the average execution time for segmenting an image was 280 ms (230 ms for the selective sampling stage and 50 ms for the parameter estimation and thresholding stage).

7. CONCLUSIONS

This paper focused on formulating, developing, and evaluating an autonomous segmentation algorithm which could accurately segment a wide class of degraded images. The goal was to develop a segmentation algorithm which was invariant to target environmental changes. A Gaussian mixture was assumed for the image model. Although this model has received considerable attention for image segmentation, its application in practical situations was severely limited. The potential of this model for improved image segmentation was identified and a histogram transformation method in conjunction with a new parameter-estimation algorithm was developed so that the segmentation algorithm could be applied to a wide class of images. The histogram of any image

was transformed to better approximate a Gaussian mixture by using the Laplacian to select pixels on both sides of an edge transition in targets. It was shown that the selective sampling algorithm successfully selected the target and background pixels at edge transitions in real images. The parameter estimation algorithm focused on estimating the components of the mixture density by first identifying the tails of the two components and then estimating the model parameters from the mixture components. Estimation results on synthesized histograms showed that the estimates of the model parameters and the optimal threshold are accurate even when the sample size is small. Finally, from the segmentation results obtained from the real infra-red images, it can be concluded that the SS–GMPE algorithm is able to accurately segment diverse images.

REFERENCES

1. C. K. Chow and T. Kaneko, Automatic boundary detection of the left ventricle from cineangiograms, *Comput. Biomed. Res.* **5,** 388–410 (1972).
2. N. Ahuja and Rosenfeld, A note on the use of second-order gray level statistics for thresholding selection, *IEEE Trans. Systems Man Cybernet.* **SMC-8**(12), 895–898 (1978).
3. Y. Nagawa and A. Rosenfeld, Some experiments on variable thresholding, *Pattern Recognition* **11,** 191–204 (1979).
4. R. L. Kirby and A. Rosenfeld, A note on the use of (gray level, local average gray level) space as an aid in thresholding selection, *IEEE Trans. Systems Man Cybernet.* **SMC-9**(12), 860–864 (1979).
5. J. S. Weszka and A. Rosenfeld, Histogram modification for threshold selection, *IEEE Trans. Systems Man Cybernet.* **SMC-9**(1), 38–52 (1979).
6. J. Kittler and J. Illingworth, Minimum error thresholding, *Pattern Recognition*, **19,** 41–47 (1986).
7. S. Cho, R. Haralick, and S. Yi, Improvement of Kittler and Illingworth's minimum error thresholding, *Pattern Recognition* **22**(5), 609–617 (1989).
8. S. Lee and S. Y. Chung, A comparative performance study of several global thresholding techniques for segmentation, Comput. Vision Graphics Image Process. **52,** 171–190 (1990).
9. T. Kurita, N. Otsu, and N. Abdelmalek, Maximum likelihood thresholding based on population mixture models, *Pattern Recognition* **25** (10), 1231–1240 (1992).
10. R. C. Jansen, K. Reinink, and G. W. A. M. van der Heijden, Analysis of gray level histograms by using statistical methods for mixtures of distributions, *Pattern Recognition Lett.* 586–590 (1993).
11. A. Rosenfeld and A.V. Kak, *Digital Image Processing*, Vol. 2. Academic press, New York (1982).
12. R. C. Gonzalez and R. E. Woods, *Digital Image Processing.* Addison-Wesley, New York (1992).
13. N. R. Pal and S. K. Pal, A review on image segmentation techniques, *Pattern Recognition*, **26,** 1277–1294 (1993).
14. Y. T. Zhou, V. Venkateswar and R. Chellappa, Edge detection and linear feature extraction using 2-D random field model, *IEEE Trans. Pattern Anal. Machine Intell.* **PAMI-11,** 84–95 (1989).
15. L. Dron, The multiscale veto model: A two-stage analog network for edge detection and image reconstruction, *Int. J. Comput. Vision* **11,** 45–61 (1993).
16. I. J. Cox, J. M. Rehg, and S. Hingorani, A Bayesian multiple hypothesis approach to edge grouping and contour segmentation, *Int. J. Comput. Vision.* **11,** 5–24 (1993).
17. M. D. Levine and S. I. Shaheen, A modular computer vision system for picture segmentation and interpretation, *IEEE Trans. Pattern Anal. Machine Intell.* **PAMI-3**(5), (1981).
18. T. C. Pong, L. G. Shapiro, L. T. Watson and R. M. Haralick, Experiments in segmentation using a facet model region grower, *Comput. Vision Graphics Image Process.* **25,** (1984).
19. J. P. Gambotto, A new approach to combining region growing and edge detection, *Pattern Recognition Lett.* **14,** 869–875 (1993).
20. M. Pietikainen, A. Rosenfeld, and I. Walter, Split and link algorithm for image segmentation, *Pattern Recognition*, **15,** 287–298 (1982).
21. J. R. Parker, Gray level thresholding in badly illuminated images, *IEEE Trans. Pattern Anal. Machine Intell.* **PAMI-13**(8), 813–819 (1991).
22. X. Wu, Adaptive split-and-merge segmentation based on piecewise least-square approximation, *IEEE Trans. Pattern Anal. Machine Intell.* **PAMI-15,** 808–815 (1993).
23. A. Rosenfeld, R. A. Hummel and S. W. Sucker, Scene labeling by relaxation operations, *IEEE Trans. Pattern Anal. Machine Intell.* **PAMI-6,** 420–433 (1976).
24. S. Peleg, A new probabilistic relaxation scheme, *IEEE Trans. Pattern Anal. Machine Intell.* **PAMI-2,** 362–369 (1980).
25. A. Rosenfeld and R. C. Smith, Thresholding using relaxation, *IEEE Trans. Pattern Anal. Machine Intell.* **PAMI-3,** 598–606 (1981).
26. M. Trivedi and J. C. Bezdek, Low-level segmentation of aerial images with fuzzy clustering, *IEEE Trans. System Man Cybernet.* **16**(4), 589–598 (1986).
27. J. M. Keller, and C. L. Carpenter, Image segmentation in the presence of uncertainty, *Int. J. Intell. Systems* **5,** 193–208 (1990).
28. S. K. Pal and A. Ghosh, Index of area coverage of fuzzy image subset and object extraction, *Pattern Recognition Lett.* **11,** 831–841 (1990).

About the Author—LALIT GUPTA received the B.E. (Hons.) degree in electrical engineering from the Birla Institute of Technology and Science, Pilani, India (1976), the M.S. degree in digital systems from Brunel University, Middlesex, England (1981), and the Ph.D degree in electrical engineering from Southern Methodist University, Dallas, Texas (1986). Since 1986, he has been with the Department of Electrical Engineering, Southern Illinois University at Carbondale, where he is currently an Associate Professor. His research interests include computer vision, pattern recognition, digital signal processing, and neural networks. Dr Gupta serves as an associate editor for Pattern Recognition and is a member of the Pattern Recognition Society, the Institute of Electrical and Electronics Engineers, and the International Neural Network Society.

About the Author—THOTSAPON SORTRAKUL received the B.E. (Hons.) degree (1990) in electrical engineering from King Mongkut's Institute of Technology at Ladkrabang, Bangkok, Thailand and the M.S. (1992) and Ph.D (1995) degrees in electrical engineering and engineering science, respectively, from Southern Illinois University at Carbondale. He worked as a Processing Engineer at SubMicron Technology, California in 1995 and he is currently a Business Development Section Manager at Shinawatra Satellite Public Co., Ltd, Bangkok, Thailand. His research interests include digital image processing, pattern recognition, and digital signal processing.