



PII: S0031-3203(97)00089-7

## MULTI-MODAL IMAGE SEGMENTATION USING A MODIFIED HOPFIELD NEURAL NETWORK

SAROJ ROUT,<sup>†</sup> SEETHALAKSHMY, PRAMOD SRIVASTAVA<sup>‡</sup>  
 and JHARNA MAJUMDAR<sup>\*,§</sup>

<sup>†</sup> BITS, Pilani

<sup>‡</sup> Centre for Artificial Intelligence and Robotics, Raj Bhavan Circle, Bangalore 560 001, India

<sup>§</sup> Aeronautical Development Establishment, C.V. Raman Nagar, Bangalore 560 093, India

(Received 22 August 1995, in revised form 9 September 1996)

**Abstract**—In most computer vision applications, it is required to segment objects from a background. In case of a multi-modal image the segmentation is an involved problem in comparison to a bi-modal image. This paper deals with an adaptive technique for choosing local threshold values for faithful image segmentation. The image segmentation is done by generating a threshold surface which is determined by interpolating the image gray levels at points where the gradient is high, indicating probable object edges. The interpolation of edge points is done using a modified Hopfield neural network and the results are compared with that of a potential surface interpolation method. © 1998 Pattern Recognition Society. Published by Elsevier Science Ltd. All rights reserved

Segmentation    Bi-modal image    Multi-modal image    Threshold surface  
 Potential surface method    Hopfield neural network

### 1. INTRODUCTION

Image segmentation is an important component of any computer vision application. It is often necessary for images to be transformed into a binary image in order to be used by high-level image understanding and interpretation algorithms. The solution to this problem is very complex when the illumination is non-uniform and the values of the objects are of heterogenous nature (i.e. multimodal). For a bimodal image the histogram indicates that the gray levels corresponding to one mode are populated with pixels that form the objects within the image and the gray levels corresponding to the other mode are populated with pixels that form the background. Therefore, a global threshold can be chosen in order to segment the objects from the background. But in case of multi-modal image, a fixed threshold, no matter how well chosen, can never be appropriate. In the case of a multi-modal image, some local information has to be used in order to segment the image faithfully. We need here a threshold level that varies over different image regions so as to fit the spatially changing background and lighting conditions. In other words, a threshold surface is needed. A good survey of thresholding techniques were suggested by Sahoo and Soltani.<sup>(1)</sup> Weszka, Panda and Rosenfeld<sup>(2, 3)</sup> suggested several variations of histogram transformation methods by using a gray level versus edge value

scatter plot. Some new algorithms are presented in references (4, 5) for multi-modal image segmentation. Yanowitz and Bruckstein<sup>(6)</sup> presented a method for finding a threshold surface using edge information. The gray levels at the boundary points are chosen as the local thresholds. Now these points serve as the interpolating points of the threshold surface. Yanowitz and Bruckstein use a successive overrelaxation method (SORM) to interpolate the surface. This method suffers from slow convergence. In this paper we use a modified Hopfield network, subjected to some constraints which has been used to interpolate the surface. The constraints are expressed as the energy function of the network which it tries to minimize. The interpolation problem will be optimally solved if the energy function can be minimized. The iterations will stop when the energy function has been minimized. The subsequent operation is to segment the image with this threshold surface pixel by pixel. The resulting image is smoothed by median filtering to remove isolated points.

### 2. SEGMENTATION USING THE THRESHOLD SURFACE METHOD

Most of the thresholding techniques, global or local, are based only on gray-level distribution in the image. This method does not suffice if the image histogram has a complicated distribution. To avoid this kind of difficulty, gray-level and gradient information can be used in a more direct way in the original image plane. For clean images, sharply defined edges are enough for segmenting (flood fill technique). But

\* Author to whom correspondence should be addressed.  
 e-mail: jharna@ade.ernet.in.

for poorly illuminated objects edges are usually broken.

Given an image to be segmented, first the image is smoothed with a median filter or an averaging filter. This is done in order to smoothen the sharp edges, so that a threshold value chosen at a high gradient point will lie between a background pixel and a object pixel as shown in Fig. 1. Thus, it is easier to separate objects when image is smoother, see references (7, 8, 3). Then a gradient image of the original image is obtained. This gradient image is thresholded using a statistics of 90% of maximum gradient value in order to get the high gradient points. The gray values at these points are chosen as the appropriate candidates for interpolating. The thresholded image is median filtered to remove discontinuities in the segmented image.

### 3. INTERPOLATION WITH POTENTIAL SURFACES

Surface interpolation is done to recover the full surface representation when only partial information of the surface is available (in this case it is the gray values at high gradient points in the image). These interpolating points form the constraints of the surface being interpolated. The problem is usually formulated in the following form

$$\varepsilon(f) = S(f) + \beta C(f). \quad (1)$$

The solution is the one that minimizes the objective functional  $\varepsilon(f)$ . The objective functional  $\varepsilon(f)$  is the combination of a stabilizing functional  $S(f)$  and a cost functional  $C(f)$ . The parameter  $\beta$  is a non-negative number and is used to adjust the weighting between the two considerations or known as the relaxation factor.

In this algorithm an exact interpolation procedure is based on potential surfaces. Laplace equation is solved using Southwell's successive overrelaxation method. The surface starts with the original pixel values of the image. Scanning the surface sequentially we compute the discrete Laplacian operation for every pixel except the edge pixels. The condition of the interpolation is to have the discrete Laplacian of the threshold surface to zero. That is

$$LAP(i, j) = \sum_{k=-1}^1 \sum_{l=-1}^1 L(k+1, l+1) V_{i+k, j+l}, \quad (2)$$

where

$$L = \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}.$$

Initially, the surface does not satisfy the equation, therefore the operator will give a residual value  $LAP(i, j)$  other than zero. Each pixel on the threshold surface is corrected by using the values of the pixel as follows:

$$V_{ij}(t+1) = V_{ij}(t) + \frac{\beta LAP(i, j)}{4}. \quad (3)$$

Therefore,  $LAP(i, j)$  is the cost function which has to be minimized. In this  $\beta$  is taken as one for the above equation to converge.

### 4. INTERPOLATION WITH HOPFIELD NEURAL NETWORK

Hopfield neural network has succeeded in solving optimization problems such as the travelling salesman problem.<sup>(9)</sup> In this paper the surface interpolation is modelled as a NP-complete problem, subjected to some constraints, that can be solved by the network.

The hopfield network is modelled as neuron-like units with symmetric connections between units ( $T_{ij} = T_{ji}$ ), continuous values, sigmoidal input-output transfer function. The motional equation of the neuron  $i$  is described as

$$C_i \frac{dU_i}{dt} = -\frac{U_i}{R_i} + \sum_{j=1}^N T_{ij} V_j + I_i, \quad (4)$$

$$V_i = g(U_i), \quad (5)$$

$$g(x) = \frac{1}{2} \frac{1}{1 + e^{-\beta x}}, \quad (6)$$

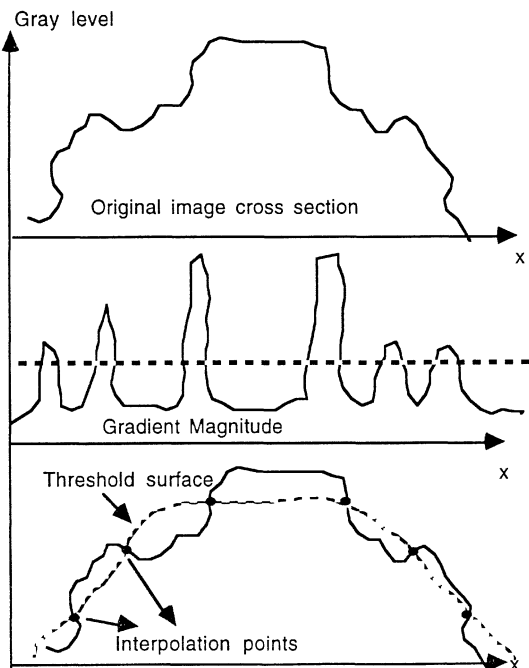


Fig. 1. Description of the adaptive threshold surface: (top) cross section in the original image, showing objects in an uneven background, (middle) cross section of the gradient magnitude image, (bottom) the peak gradient points whose gray values are interpolation values, that determine the threshold surface.

where  $g(x)$  is the sigmoidal monotonic transfer function,  $U_i$  is the total input of the  $i$ th neuron and  $V_i$  is the output of that neuron.  $T_{ij}$  is conductance between the output of the neuron  $j$  and the input of neuron  $i$ .  $C_i$  represents the capacitance of each node  $i$  and  $N$  represents the number input synapses.

The suitable energy function for the Hopfield network has the following form:

$$E = -0.5 \sum_{i=1}^N \sum_{j=1}^N T_{ij} V_i V_j - \sum_{i=1}^N V_i I_i + \sum_{i=1}^N \int_0^{V_i} g^{-1}(V) dV. \quad (7)$$

In equation (8)  $dE/dt \leq 0$  when  $C_i > 0$  and  $T_{ij} = T_{ji}$ .  $dE/dt = 0$  if and only if  $dV_i/dt = 0$  ( $i = 1, 2, 3, \dots, N$ ). It shows that the motional equation with symmetric connections ( $T_{ij} = T_{ji}$ ) always lead to convergence to the local minimum of the energy function. If the threshold problem is represented as suitable energy function then Hopfield network can be used for interpolation.

According to the requisite described in the equation (3), the error function  $E_1$  for interpolation can be formulated as follows:

$$E_1 = 0.5 \sum_{(i,j)} [LAP(i,j)]^2 = 0.5 \sum_{(i,j)} \left[ \sum_{k=-1}^1 \sum_{l=-1}^1 L(k+1, l+1) V_{i+k, j+l} \right]^2. \quad (8)$$

After rearranging the right-hand side, the error expression yields

$$E_1 = 0.5 \sum_{(i,j)} \left[ \sum_{k=-2}^2 \sum_{l=-2}^2 M_S(k+2, l+2) \times V_{i+k, j+l} V_{i,j} \right], \quad (9)$$

where

$$M_S = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 2 & -8 & 2 & 0 \\ 1 & -8 & 20 & -8 & 1 \\ 0 & 2 & -8 & 2 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix}.$$

In order to smooth the threshold surface, error function  $E_2$  is defined as follows.

$$E_2 = 0.5 \sum_{i,j} \sum_{k=-1}^1 \sum_{l=-1}^1 [V_{i+k, j+l} - V_{i,j}]^2 = 0.5 \sum_{i,j} \sum_{k=-1}^1 \sum_{l=-1}^1 S(k+1, l+1) V_{i+k, j+l} V_{i,j}, \quad (10)$$

where

$$S = \begin{pmatrix} -2 & -2 & -2 \\ -2 & 16 & -2 \\ -2 & -2 & -2 \end{pmatrix}.$$

The total error function is

$$E = E_1 + \alpha E_2 = 0.5 \sum_{(i,j)} \left[ \sum_{k=-2}^2 \sum_{l=-2}^2 M_S 1(k+2, l+2) \times V_{i+k, j+l} V_{i,j} \right] \quad (11)$$

where

$$M_S 1 = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 2-2\alpha & -8-2\alpha & 2-2\alpha & 0 \\ 1 & -8-2\alpha & 20+16\alpha & -8-2\alpha & 1 \\ 0 & 2-2\alpha & -8-2\alpha & 2-2\alpha & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix}.$$

Comparing the like coefficients between energy function (8) and (12) results in the following network parameter:

$$T_{ij, (i+k) (j+l)} = -M_S 1(k+2, l+2), \quad -2 \leq k, l \leq 2. \quad (12)$$

The resulting network is the special case of the general hopfield network and the motional equation of the neuron  $i$  can be obtained as follows.

$$C_{ij} \frac{dU_{ij}}{dt} = -\frac{U_{ij}}{R_{ij}} - \sum_{k=-2}^2 \sum_{l=-2}^2 M_S 1(k+2, l+2) \times V_{i+k, j+l} + U_{i0j0}, \quad (13)$$

$$V_{ij} = g(U_{ij}). \quad (14)$$

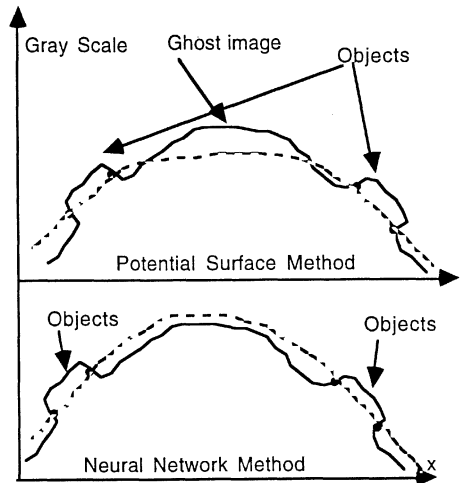


Fig. 2. (Top) Figure shows the threshold surface fitted by potential method. (Bottom) Threshold surface fitted by neural network method.

It can be seen from the above equation that though the whole image is treated as Hopfield network, while finding the motion equation for each pixel  $U_{ij}$  only a small neighbourhood ( $5 \times 5$ ) is taken into account. From the above equation the left-hand term  $dU_{ij}/dt$  is calculated and the surface values are updated using

the following equation:

$$U_{ij}(t+1) = U_{ij}(t) + \eta \frac{dU_{ij}}{dt}. \quad (15)$$

In equation (15)  $\eta$  controls the rate of convergence.

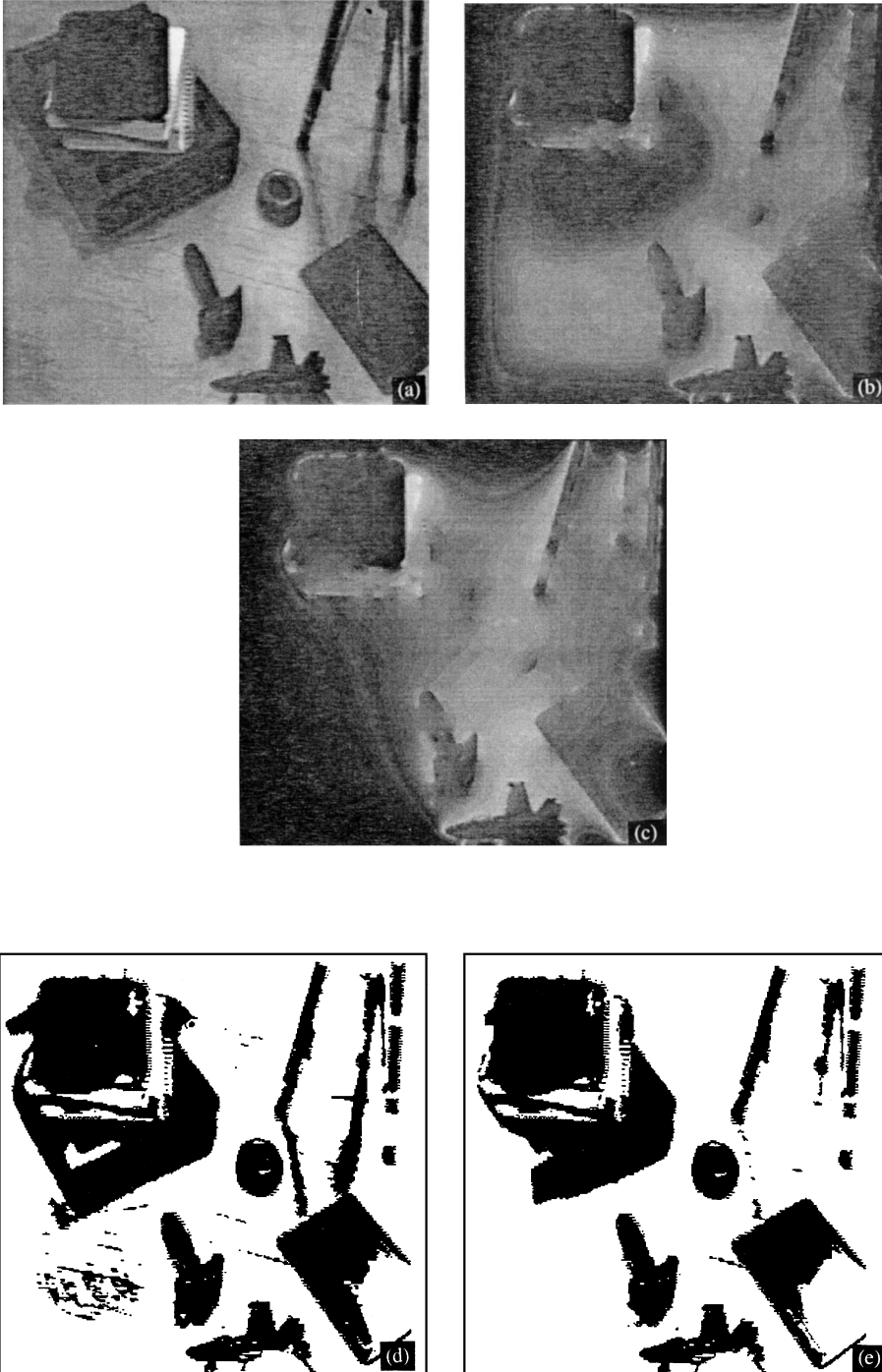


Fig. 3. (a) Original image. (b) Threshold surface by neural network. (c) Threshold surface by potential method. (d) Segmented image using neural network method. (e) Segmented image using potential method.

## 5. IMPLEMENTATION ISSUES AND EXPERIMENTAL RESULTS

The thresholding method described was implemented in C language using a Silicon Graphics workstation running UNIX. A summary of the algorithm is as shown below.

*Step 1:* For all pixels in the image ( $M \times N$ ) do the following steps for preprocessing.

- Process the image with a averaging filter using a  $3 \times 3$  window or  $5 \times 5$  window.
- Find the gradient magnitude image using any one of the derivative masks (Robert, Sobel, etc.).

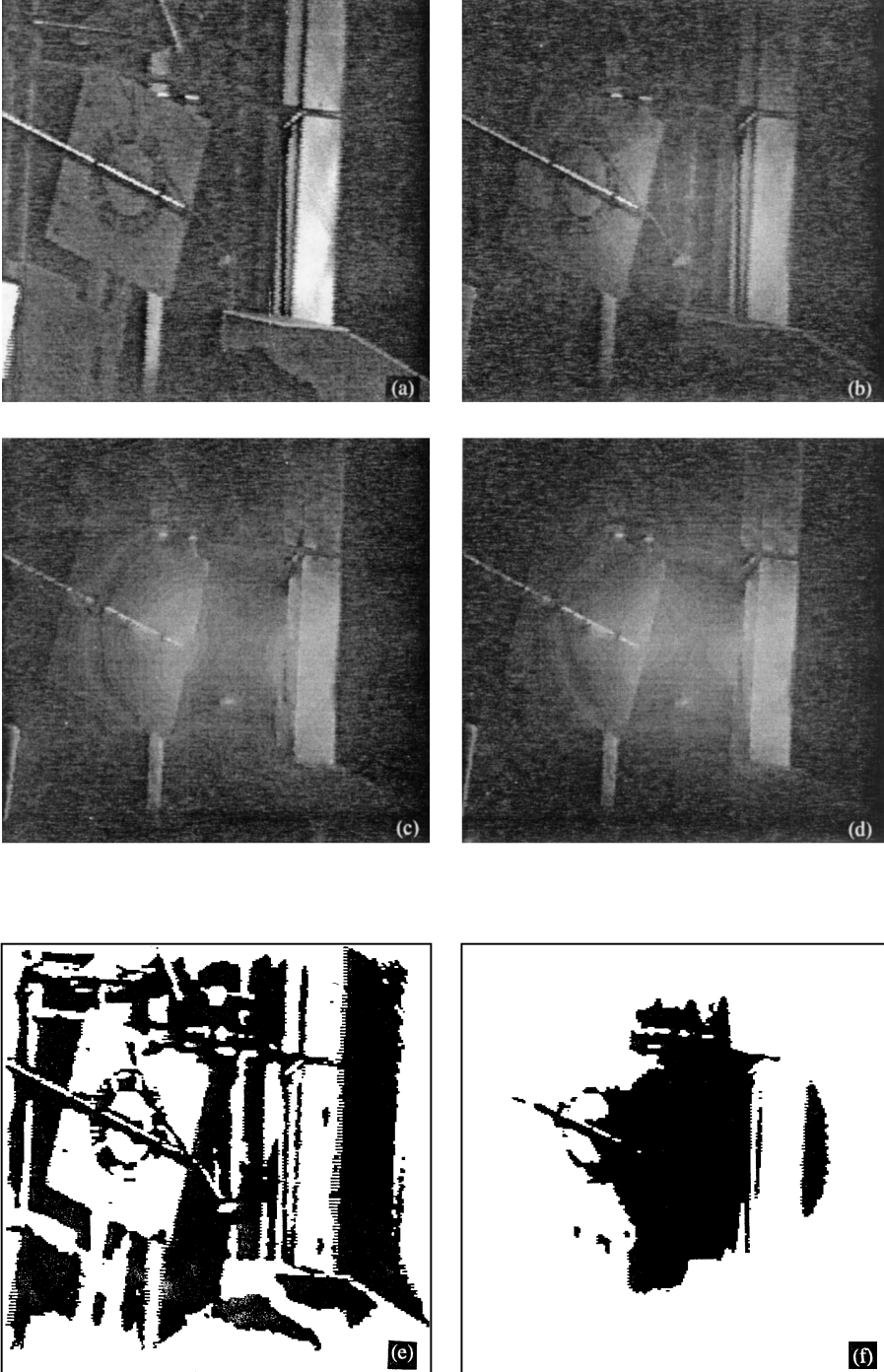


Fig. 4. (a) Original image. (b) Image superimposed with a Gaussian lighting. (c) Threshold surface by neural network. (d) Threshold surface by potential method. (e) Segmented image using neural network method. (f) Segmented image using potential method.

—Take a statistics of 90% of the maximum gradient value and any pixel lying within this range is marked as an edge point.

*Step 2:* Starting from row 2 to  $M - 2$  and column 2 to  $N - 2$ , for every pixel  $(i, j)$  take  $5 \times 5$  window.

*Step 3:* For every element in the  $5 \times 5$  window find their sigmoidal outputs.

*Step 4:* Operate the mask MS1 over the whole window and find the *sum*.

*Step 5:* Find the change in the pixel values  $dU_{ij}/dt$  by using equation (14).

*Step 6:* Modify the pixel values by using the equation (15).

*Step 7:* Continue executing from Steps 2 to 6 till the change in the pixel value is less than some small number  $\epsilon$ .

*Step 8:* After the threshold surface has converged, for every pixel  $U_{ij}$  in the original image (IMG1), compare the value of the threshold surface  $V_{ij}$  in (IMG2) and segment the image according to the following procedure.

$$I(i, j) = \begin{cases} 0 & \text{if } IMG1(i, j) \leq IMG2(i, j), \\ 1 & \text{if } IMG1(i, j) > IMG2(i, j). \end{cases}$$

*Step 9:* The thresholded image is then median filtered to remove the isolated points.

Figure 2 shows the object surface superimposed with the thresholded surface using the potential surface method and the neural network method. For implementing the algorithm the gray-scale values were scaled down to 0–10. The  $\beta$  value was chosen to



Fig. 5. (a) Original image. (b) Segmented image by neural network.

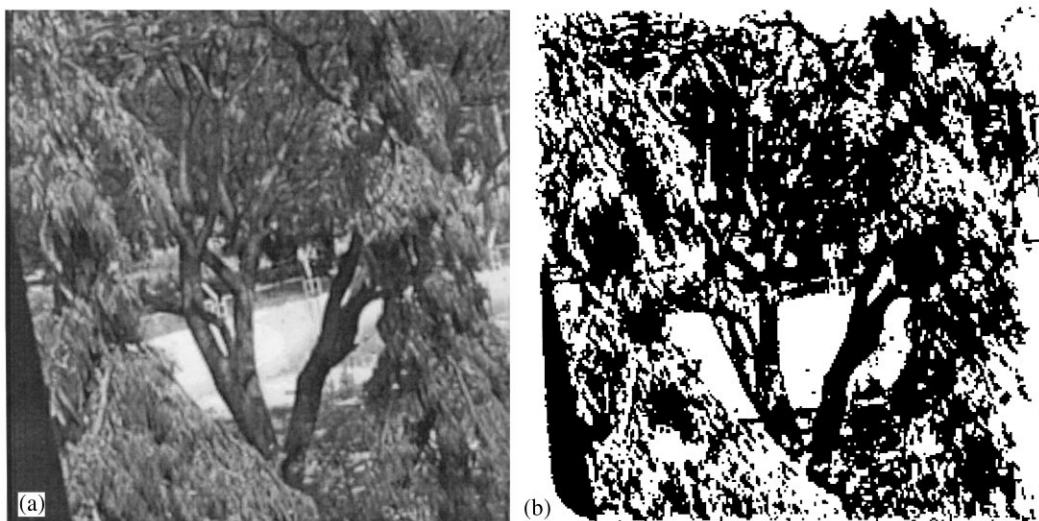


Fig. 6. (a) Original image. (b) Segmented image by neural network.

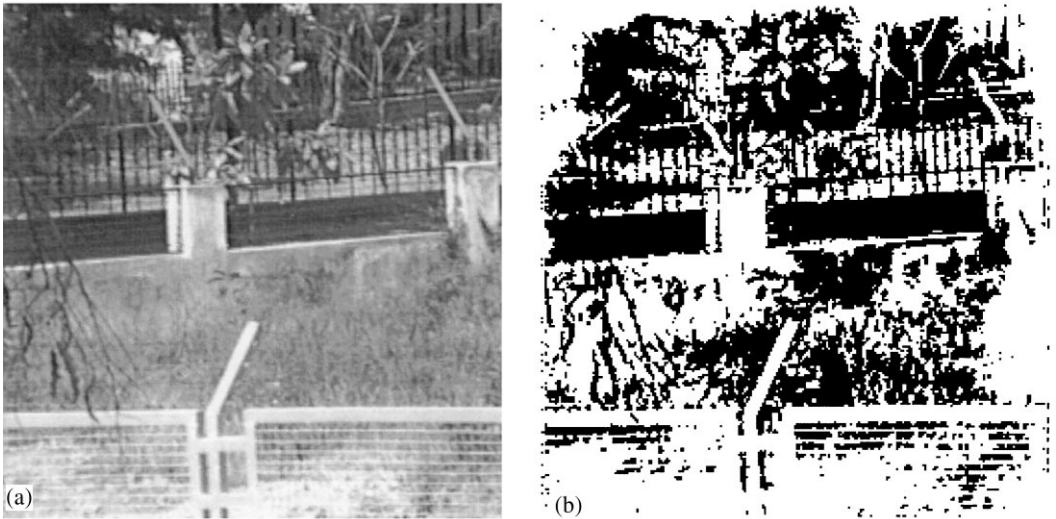


Fig. 7. (a) Original image. (b) Segmented image by neural network.

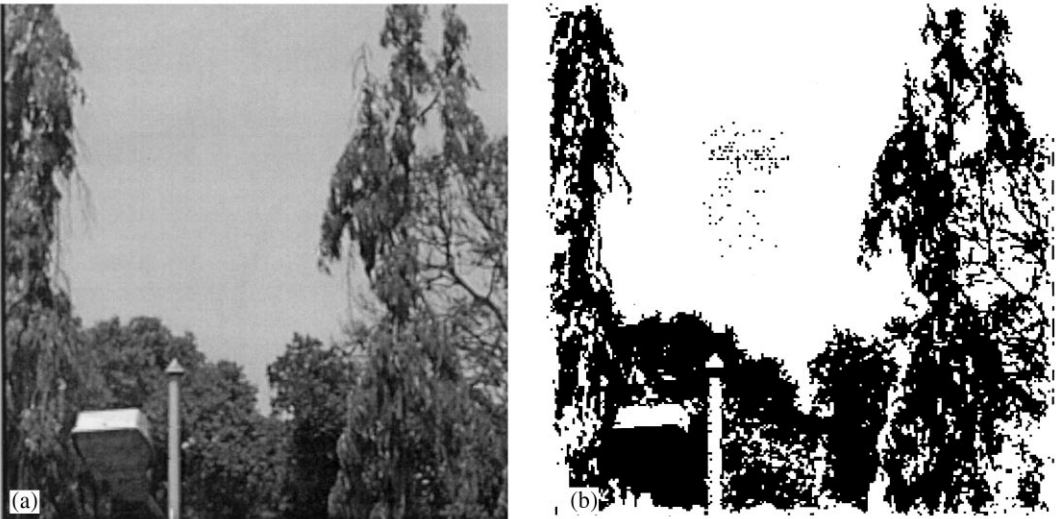


Fig. 8. (a) Original image. (b) Segmented image by neural network.

be 0.02 for the sigmoidal function  $g(x)$  in equation (6). The values of  $R_{ij}$  and  $C_{ij}$  are taken as same for all neurons. The choice of values for  $R_{ij}$  and  $C_{ij}$  are very crucial for the convergence of the surface. Typical values chosen for  $R_{ij}$  and  $C_{ij}$  are 1 and 1.414, respectively. The pixels corresponding to the edge values are not modified.

Some test images were taken and both the algorithms (potential surface method and neural network method) were tried on the images. The results are shown. The first test image is shown in Fig. 3(a). Figure 3(b) is the threshold surface by neural network method and Fig. 3(c) is the threshold surface fitted by potential method. It can be seen that the surface fitted by the neural network follows the object surface, due to the bias term  $U_{i,j_0}$  which is added to each of the

neuron. The no. of iterations taken are 4500 and 880 by potential method and neural method, respectively. This is due to the property of the network to optimize. Figures 3(d) and (e) is the corresponding segmented images. It can be seen from Fig. 3(d) that the edge of the left hand side box is clearly segmented, whereas in case of the potential method as shown in Fig. 3(e), it has disappeared. Figure 4(a) shows a second test image and Fig. 4(b) is the image superimposed with a Gaussian lighting. Figures 4(c) and (d) is the threshold surface fitted by neural and potential method, respectively. Figures 4(e) and (f) is the segmented image by neural and potential method, respectively. It can be seen clearly that many objects, fully or partially, has disappeared completely in case of the potential method of surface interpolation. The neural

network method has succeeded in this case because it tries to follow the image surface as shown in Fig. 2. Figures 5–8 show some more test runs with input images and corresponding segmented output images obtained using neural network approach.

*Acknowledgements*—The authors thank the Director, CAIR for his kind permission to publish this paper.

#### REFERENCES

1. P. K. Sahoo, S. Soltani and A. K. C. Wong, A survey of thresholding techniques, *CVGIP-41*, pp. 233–260 (1988).
2. Panda and A. Rosenfeld, Image segmentation by pixel classification in (gray level, edge value Space), *IEEE Trans. Comput.* **C-27**, 875–879 (1978).
3. S. W. Joan and A. Rosenfeld, Histogram modification for threshold selection, *IEEE Trans. Systems Man Cybernet.* **SMC-9**(1), 38–51 (1979).
4. J. R. Parker, Gray level thresholding in badly illuminated images, *IEEE Trans. Pattern Analy. Mach. Intell.* **PAMI-13**(8), 813–819 (1991).
5. N. Papamarkos and B. Gatos, A new approach for multilevel threshold selection, *Graphical Models Image Process.* **CVGIP-56**(5) 357–370 (1994).
6. S. D. Yanowitz and A. M. Bruckstein, A new method for image segmentation, *CVGIP-46*, pp. 82–95 (1989).
7. A. Rosenfeld and A.C. Kak, *Digital Picture Processing*. Academic Press, New York (1982).
8. L. S. Davis, A. Rosenfeld and J. S. Weszka, Region extraction by averaging and Thresholding, *IEEE Trans. Systems Man Cybernet.* **SMC-5**, 383–388 (1975).
9. J. J. Hopfield and D. W. Tank, Neural computation of detection in optimization problems, *Biol. Cybernet.* **52**, 141–152 (1985).
10. S. Dinggang and F. Qi, Adaptive image segmentation by neural network, *Proc. 3rd Int. Conf. on Automation, Robotics and Computer Vision, ICARCV '94*, Singapore, pp. 9–11, November (1994).

**About the Author**—SAROJ ROUT, born in 1974 in Cuttack India, received B.E. degree in Instrumentation from Birla Institute of Technology and Science, Pilani, India in 1991. Currently he is doing M.E. in Microelectronics with a teaching assistantship from Birla Institute of Technology, Pilani. His areas of interest are Digital/Analog VLSI Design, Fast Computer Architectures, High Level Synthesis and development of Hardware for Computer.

**About the Author**—JHARNA MAJUMDAR received B. Tech in Electronics and Electrical Communication Engineering and Post graduate diploma in Computer Technology from Indian Institute of Technology, Kharagpur in 1969 and 1970, respectively. From 1971 to 1983, Dr Majumdar worked as a Research Scientist in the Central Mechanical Engineering Research Institute (CSIR), Durgapur, India. She received Ph.D. degree in Electrical Engineering in 1980. From 1983 to 1989, Dr Majumdar has worked as Research Scientist in the Robotics and Computer Science Research Institute, University of Karlsruhe, Germany. Dr Majumdar joined DRDO at the Centre for Artificial intelligence and Robotics, Bangalore, India in 1990. During the period 1990–1995, she has worked in various research areas including Robot Vision and Machine Vision for industrial automation. Currently, Dr Majumdar is working in the Aeronautical Development Establishment, Bangalore, India in the area of analysis and interpretation of airborne digital images and satellite images for defence application. Dr Majumdar has published about 45 reviewed technical papers in national and international journals and conference proceeding. She is a member of IEEE and Computer Society of India.