TABLE II
TYPICAL RESULTS DEMONSTRATING THE PROPERTIES OF THE $DL_{RI}$, $ADL_{IP}$, AND THE $ADL_{RP}$ AUTOMATA[a]

| | $c_1$ | $DL_{RI}$ Scheme $\hat{E}[p_1(\infty)]$ | $DL_{RI}$ Scheme M.T.C. | $ADL_{IP}$ Scheme $\hat{E}[p_1(\infty)]$ | $ADL_{IP}$ Scheme M.T.C. | $ADL_{RP}$ Scheme $\hat{E}[p_1(\infty)]$ | $ADL_{RP}$ Scheme M.T.C. |
|---|---|---|---|---|---|---|---|
| $N = 4$ | 0.2 | 0.896 | 4.61 | 0.960 | 9.87 | 0.95 | 2.965 |
| | 0.4 | 0.843 | 6.04 | 0.825 | 10.65 | 0.86 | 3.730 |
| | 0.6 | 0.741 | 8.52 | 0.655 | 10.57 | 0.72 | 4.830 |
| $N = 10$ | 0.2 | 0.980 | 11.46 | 1.00 | 70.28 | 1.00 | 8.220 |
| | 0.4 | 0.951 | 16.15 | 1.00 | 196.78 | 0.98 | 14.88 |
| | 0.6 | 0.855 | 25.58 | 0.93 | 499.11 | 0.93 | 32.45 |

[a] In all the experiments $c_2 = 0.8$.

environments, the $DL_{RP}$ will perform better and will give a lower value for the average asymptotic expected loss.

## VI. CONCLUSION

In this paper we have stated and proved asymptotic results concerning various variable structure stochastic automata. These automata however, unlike most automata discussed in the literature, change the action probabilities in discrete jumps. The automata are called linear because these jumps are all of equal length. We have proved that the $DL_{RP}$ scheme is ergodic and is $\epsilon$-optimal in all environments wherever the minimum penalty probability is less than 0.5. By artificially making the end states of the latter automaton absorbing, we have designed the $ADL_{RP}$ automaton and proven its $\epsilon$-optimality. This is the only known symmetric reward–penalty scheme that is linear and yet $\epsilon$-optimal. We conjecture that there is none other. Also by stochastically filtering the inputs to the $DL_{RP}$ automaton we have designed the modified $DL_{RP}$ ($MDL_{RP}$) scheme that is the only known ergodic linear reward-penalty scheme $\epsilon$-optimal in all environments.

We are currently investigating (and have some preliminary results) on the use of these automata to adaptively control a robot manipulator operating in a noisy workspace. The problems of studying nonlinear [14] and multiaction discretized reward-penalty schemes remain open.

## ACKNOWLEDGMENT

The authors are grateful to David Ng who did much of the simulations and to Robert Cheetham who painstakingly prepared the document from a poorly written manuscript. The first author is also grateful to Prof. Thathachar from the Indian Institute of Science, Bangalore, India, who is a pioneer in the area of learning automata. The first author worked jointly with him on some of the initial results involving discretized automata and he was a constant source of motivation for the subsequent results.

## REFERENCES

[1] Y. A. Flerov, "Some classes of multi-input automata," *J. Cybern.*, vol. 2, no. 3, pp. 112-122, 1972.

[2] D. L. Isaacson, and R. W. Madson, *Markov Chains: Theory and Applications*. New York: Wiley, 1976.

[3] S. Lakshmivarahan, *Learning Algorithms Theory and Applications*. New York: Springer-Verlag, 1981.

[4] ____, "$\epsilon$-optimal learning algorithms—Nonabsorbing barrier type," Tech. Rep. EECS 7901, Feb. 1979, School of Electrical Engineering and Computing Sciences, University of Oklahoma, Norman, OK.

[5] ____, "A learning approach to the two-person decentralizing team problem with incomplete information," *Applied Mathematics and Computation*, vol. 8, pp. 51–78, 1981.

[6] S. Lakshmivarahan and M. A. L. Thathachar, "Absolutely expedient algorithms for stochastic automata," *IEEE Trans. Syst. Man Cybern.*, vol. SMC-3, no 3, pp. 281–286, May 1973.

[7] M. R. Meybodi, "Learning automata and its application to priority assignment in a queueing system with unknown characteristics," Ph.D. thesis, School of Elect. Eng. Computing Sciences, University of Oklahoma, Norman, OK.

[8] K. S. Narendra and M. A. L. Thathachar, forthcoming book on learning automata.

[9] ____, "Learning automata—A survey," *IEEE Trans. Syst. Man Cybern.*, vol. SMC-4, no. 4, pp. 323–334, July 1974.

[10] ____, "On the behaviour of a learning automaton in a changing environment with routing applications," *IEEE Trans. Syst. Man Cybern.*, vol. SMC-10, 1980, pp. 262–269.

[11] K. S. Narendra, E. Wright, and L. G. Mason, "Applications of learning automata to telephone traffic routing and control," *IEEE Trans. Syst. Man Cybern.*, vol. SMC-7, no. 11, pp. 785–792, Nov. 1977.

[12] B. J. Oommen and E. R. Hansen, "The asymptotic optimality of discretized linear reward- inaction learning automata," *IEEE Trans. Syst. Man Cybern.*, May/June 1984, pp. 542–545.

[13] B. J. Oommen and M. A. L. Thathachar, "Multi action learning automata possessing ergodicity of the mean," in *Proc. 1983 IASTED Symp. Measurement and Contr.*, MECO 83, pp. 61–64.

[14] B. J. Oommen, "Absorbing and ergodic discretized two-action learning automata," *IEEE Trans. Syst. Man Cybern.*, vol. SMC-16, no. 2, pp. 282–296, Mar./Apr. 1986.

[15] ____, "A learning automaton solution to the stochastic minimum spanning circle problem," *IEEE Trans. Syst. Man Cybern.*, vol. SMC-16, no. 4, pp. 598–603, July/Aug. 1986.

[16] B. J. Oommen and D. C. Y. Ma, "Deterministic learning automata solutions to the equi-partitioning problem," *IEEE Trans. Comput.*, vol. 37, no. 1, pp. 2–14, Jan. 1988.

[17] ____, "Fast object partitioning using stochastic learning automata," in *Proc. 1987 Int. Conf. Res. and Develop. in Inform. Retrieval*, New Orleans, June 1987, pp. 111–120.

[18] M. A. L. Thathachar and B. J. Oommen, "Discretized reward inaction learning automata," *J. Cybern. Inform. Sci.*, pp. 24–29, Spring 1979.

[19] M. L. Tsetlin, "On the behaviour of finite automata in random media," *Automat. Telemekh.* (USSR), vol. 22, pp. 1345–1354, 1961.

[20] ____, *Automation Theory and the Modelling of Biological Systems*. New York: Academic, 1973.

[21] Y. Z. Tsypkin and A. S. Poznyak, "Finite learning automata," *Eng Cybern.*, vol. 10, pp. 478–490, 1972.

[22] V. I. Varshavskii and I. P. Vorontsova, "On the behaviour of stochastic automata with variable structure," *Automat. Telemekh.* (USSR), vol. 24, pp. 327–333, 1963.

# A Modified Scheme for Segmenting the Noisy Images

B. CHANDA, MEMBER, IEEE, B. B. CHAUDHURI,
AND D. DUTTA MAJUMDER

***Abstract*—An image segmentation scheme based on gray level thresholding is presented. To reduce errors in misclassification, gray level histograms are sharpened before thresholding using a gray level transformation function that also leads to an expression for computing the expected threshold. Three new thresholding methods are proposed that reduce the noise, smooth region boundaries, and preserve connectedness among different parts of objects, and are not expensive.**

Manuscript received June 11, 1986; revised April 25, 1987.

The authors are with the Electronics and Communication Sciences Unit, Indian Statistical Institute, Calcutta 700035, India.

IEEE Log Number 8717370.

## I. Introduction

Usually image analysis requires images to be segmented into different compact and meaningful regions. By the term "meaningful regions," are meant regions having a shape that human observers can readily recognize as a particular object or a part of an object. Segmentation is basically a process of pixel classification that extracts the regions by assigning the individual pixels to classes. Most of the image segmentation techniques are *ad hoc* in nature. There are no general algorithms that will work for all images. The experimenter has to choose or develop their own method depending on the problem in hand. Fu *et al.* [1] have published a good survey on the topic. In the following sections only the gray level thresholding that is popular for extracting the entire object will be considered. The gray level thresholding technique segments an image into different regions by classifying pixels based on their gray levels and/or average gray levels. The scheme described here is based on the assumption that the images have only two sets of regions, namely objects and background. Let $O_r$ denote a rectangular set of pixels $\{(j,k) \mid j = 0,1,2,\cdots,M-1 \text{ and } k = 0,1,2,\cdots,N-1)$ of an image $g$ with $g(j,k)$ representing the brightness value at the pixel $(j,k)$. Let $S_i$ be disjoint, nonempty, path-connected subset of $O_r$ obtained by thresholding $g(j,k)$, at $x_t$, such that

$$\bigcup_i S_i = O_r.$$

Then, either

$$\min\{g(j,k) \mid (j,k) \in S_i\} > x_t,$$

or

$$\max\{g(j,k) \mid (j,k) \in S_i\} \leqslant x_t.$$

Also, if $S_i$ and $S_2$ are adjacent, then

$$\min\{g(j,k) \mid (j,k) \in S_i\} > x_t = $$
$$> \max\{g(j,k) \mid (j,k) \in S_i\} \leqslant x_t$$

and vice versa. The main problem with this technique is the selection of the threshold $x_t$. Weszka [2] has surveyed a number of threshold selection techniques.

Thresholds can be selected using the information contained by the gray level histograms [3]. Doyle [4] has suggested the $p$-tile method that chooses as a threshold the gray level that most closely corresponds to mapping (100-$p$) percent of total pixels into the object. This method is not applicable if the object is unknown or varies from picture to picture. Prewitt *et al.* [5] suggested the mode method that selects the threshold at the bottom of valley between the peaks (or modes). The automatic selection scheme involves smoothing the histogram, searching for modes, and placing the threshold at the minima between them [6]. The concavity analysis technique can also be applied to a histogram as an aid in threshold selection [7]. However in many cases, even when peaks in the histograms are detected, it may be difficult to locate accurately the valley bottom since this may be flat and broad. To surmount this problem, obtaining the histograms with deep valley are proposed [8], [9]. Such a histogram can be obtained by computing the gray levels of only those pixels whose laplacian [8] or gradients [9] are in the $p$-tile range, which again is a heuristic approach.

Rosenfeld *et al.* [10] and Peleg [11] proposed using iterative histogram modification to sharpen the peaks and, consequently, to deepen the valleys. So the histogram sharpening technique is used to smooth the image as well as to facilitate the choice of threshold. Both of them use only global information obtained from the histogram. As a result, sometimes they create some spikes representing unwanted regions. In another interesting algorithm [12], local information is used to sharpen the histogram. However, it cannot smooth all possible noise and, even worse, enhances it in some cases.

After sharpening the histogram and selecting the threshold, the image is finally segmened by thresholding the gray level. Hence, at this stage the image segmentation can be described as a thresholding operation that transforms the input image onto another image, i.e.,

$$T_{th}: \{\hat{g}(j,k)\} \longrightarrow \{b(j,k)\}.$$

So $\{b(j,k)\}$ is a binary image where image subset $S$ represents the object and $\bar{S}$ the background. Here $\hat{g}$ denotes the input image that may be either the observed image or the processed image. The operator $T_{th}$ satisfies the following two conditions:

a) $T_{th}$ is not invertible since the same output is produced by more than one input;
b) $T_{th}$ can use any value $t_x(j,k)$ as threshold from the interval $[\min_{(j,k)}\{g(j,k)\}, \max_{(j,k)}\{g(j,k)\}]$.

Therefore, $t_x(j,k)$ is the gray level threshold at the pixel $(j,k)$ and can be found out using the relation

$$t_x(j,k) = \phi_{th}(j,k,Q_p(j,k))$$

$Q_p(j,k)$ denotes the set of local properties of the pixel $(j,k)$. For each pixel $(j,k)$ in the image, if $\hat{g}(j,k) \leqslant t_x(j,k)$, then $(j,k)$ is labeled as an object point. Otherwise, $(j,k)$ is labeled as a background point. Depending on the variables required to compute $t_x(j,k)$ the threshold will be called global, local, and dynamic.

When $t_x(j,k)$ depends on none of the variables and remains constant throughout the image, then $t_x(j,k)$ is called global (or constant) threshold.

When $t_x(j,k)$ depends on $Q_p(j,k)$ only, i.e., nature of the function $\phi_{th}$ is same at every pixel in the image, then $t_x(j,k)$ is called local or space-variant threshold.

The variable $t_x(j,k)$ is called dynamic threshold if it depends on $j,k$ and $Q_p(j,k)$. It may depend on $g(j,k)$ also.

Due to presence of noise in the input image, regions extracted through gray level thresholding technique may have coarse boundaries and holes in them. So the outcome of the segmentation process may require postprocessing for better recognition score. Shrinking and expansion (sometimes also called contraction and dilation) operations are popularly used as the postprocessing techniques in such cases [13].

In Section II, a new measure for bimodality of a histogram is proposed. The determination of threshold is facilitated through histogram sharpening by using a gray level transformation as described in Section III. This transformation helps us to compute rather than to select the expected threshold. Applicability of this method depends on the detection of peaks and valleys in the histogram, so a new method of computing the histograms with deep valleys is also described here. Finally, to have smooth boundaries, to avoid creation of holes and to preserve connectedness among the weakly connected parts of the extracted objects, three modified thresholding procedures are proposed in Section IV. These procedures are not expensive. Section V describes the experimental results. Concluding remarks are cited in Section VI.

## II. Measure of Bimodality of a Histogram

In this scheme some modifications of mode method are suggested for easy selection of threshold and for achieving better segmentation of regions. Mode method selects, as the threshold, the gray level corresponding to the bottom of valley between the peaks of a histogram, so the applicability of mode method in selecting the threshold depends on the bimodality of histogram.

Locating peaks and valleys and determining bimodality are nontrivial tasks in themselves. Let $x_i (i = 1,2,3,\cdots,L)$ are possible gray levels in the image and let $h_g(x_i)$ represents the bar height (or frequency) corresponding to the gray level $x_i$. To reduce the possibility of selecting the improper local modes or bottom of valley due to presence of noise and digitization errors, first the histogram heights are smoothed with a one dimensional window of length $(2q+1)$ to obtain new value of barheights $h_g(x_i)$, i.e.,

$$\bar{h}_g(x_i) = \frac{1}{2q+1} \sum_{j=-q}^{q} h_g(x_{i-j}) \tag{1}$$

for all $i$ except the end points. Now let $\bar{h}_g(x_i) = 0$ for $x_i < GL$ and $x_i > GH$. Then define

$$s = \left[ \frac{GH - GL}{K} \right] \tag{2}$$

where, $K$ is a constant chosen *a priori* and $[\beta]$ is greatest integer not exceeding $\beta$. Now select local maxima at

$$x_l\,(l = l_1, l_2, l_3, \cdots, l_f),$$

such that

$$\bar{h}_g(x_l) \geqslant h_g(x_{l-i}) \qquad \text{for } -s \leqslant i \leqslant s$$

and the bottom of valley is minimum between two extreme modes, i.e.,

$$\bar{h}_g(x_v) \leqslant \bar{h}_g(x_i) \qquad \text{for } l_1 < i < l_f$$

Hence, two distinct modes $x_{m_1}$ and $x_{m_2}$ are found as

$$\bar{h}_g(x_{m_1}) \geqslant \bar{h}_g(x_i) \qquad \text{for } l_1 \leqslant i < v$$

and

$$\bar{h}_g(x_{m_2}) \geqslant h_g(x_i) \qquad \text{for } v < i \leqslant l_f$$

such that $|x_{m_1} - x_{m_2}|$ attains maximum possible value.

The last condition abolishes the possibility of selecting more than one gray level as modes $x_{m_1}$ or $x_{m_2}$ due to equality condition. It is evident that the value of the parameters $q$ and $K$ can be selected from a wide range of values (e.g., $q = 1, 2, 3, 4$).

Now, the slope of lines joining the points $(x_v, \bar{h}_g(x_v))$, $(x_{m_1}, \bar{h}_g(x_{m_1}))$ and $(x_v, \bar{h}_g(x_v))$, $(x_{m_2}, \bar{h}_g(x_{m_2}))$, respectively, are given by

$$SL_1 = \left| \frac{h_g(x_{m_1}) - h_g(x_v)}{x_{m_1} - x_v} \right| \tag{3}$$

and

$$SL_2 = \left| \frac{h_g(x_{m_2}) - h_g(x_v)}{x_{m_2} - x_v} \right|. \tag{4}$$

Based on the values of $SL_1$ and $SL_2$ about the bimodality of a histogram is inferred in the following way.

Let $S_t$ is a threshold chosen *a priori*. Now if $SL_1 < S_t$ and $SL_2 < S_t$, then the histogram is considered to be nearly flat. If $SL_1 < S_t$ and $SL_2 \geqslant S_t$, or vice versa, then it is inferred that the histogram has a peak and shoulder. Finally if $SL_1 \geqslant S_t$ and $SL_2 \geqslant S_t$, then histogram is said to be strongly bimodal. In our experiment we have taken $S_t = \tan 30°$. This measure of bimodality has a disadvantage in that it depends on the rate of sampling. To avoid this problem, we should normalize the histogram bar heights or the values of $SL_1$ and $SL_2$ before drawing any inference about the bimodality.

## III. Histogram Sharpening

It is evident that even in the case of fairly bimodal gray level histogram, selection of threshold is still a perplexing task due to overlapping of gray level distributions for different regions. So we need some algorithms that will separate the gray level distributions for different regions sufficiently, and increase the strength of bimodality. The algorithms should use both the global and local informations so that noise is removed to some extent and so that a deep valley between the peaks will emerge, thereby facilitating the choice of threshold.

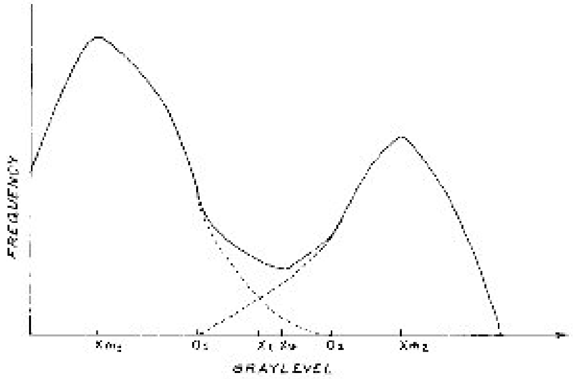


Fig. 1. Gray level histogram of image showing different parameters. Dotted lines represent individual gray level distributions corresponding to each class of regions.

### A. A Piecewise Continuous Graylevel Transformation Function

Suppose an image $(g)$ contains two classes of distinct regions, for which there are two significant peaks at $x_{m_1}$ and $x_{m_2}$ in histogram $h_g$. $x_v$, is the gray level corresponding to the bottom of valley between the peaks. For the time being, let us consider the gray level of an image as a continuous variable and that the histogram is approximated by the probability density function $P_x(x)$ of gray level $x$ in the image. Suppose $P(x|Z_1)$ and $P(x|Z_2)$ represent gray level density function for class 1 (i.e., object) and class 2 (i.e., background) regions, respectively and possess the following properties:

(1) $P(x|Z_1)$ has only one maxima at $x = x_{m_1}$;
(2) $P(x|Z_2)$ has only one maxima at $x = x_{m_2}$;
(3) $P(x|Z_1) = 0$ for $x < x_1$ or $x > a_2$;
(4) $P(x|Z_2) = 0$, for $x < a_1$ or $x > x_L$;

where, $a_1$ may be greater than, equal to or less than $a_2$. For first two situations, the gray level threshold can be selected most easily and accurately. But if $a_1$ is less than $a_2$ (Fig. 1) as is, common in practice, a problem due to the overlapping of gray level distributions occurs. Here, we have tried to solve this problem by applying a gray level transformation function that maps the gray level $x$ onto another gray level $u$ so that the threshold can easily be selected from $p_u(u)$, the probability density function (PDF) of gray level in the transformed image.

Now using the previous assumptions the PDF for the entire input image is given by

$$P_x(x) = P_x(x|Z_1) + P_x(x|Z_2) \tag{5}$$

where, $p_x(x|Z_1) = p(Z_1)p(x|Z_1)$ and $p_x(x|Z_2) = P(Z_2)p(x|Z_2)$. Since either the slope of the tangent to $P_x(x)$ changes abruptly at $a_1$ and $a_2$, as shown in Fig. 1, or $p_x(x)$ may be nondifferentiable at $a_1$ and $a_2$, the gray levels may be defined as $a_1$ and $a_2$ in the discrete domain such that

$$\text{diff}^2 h_g(a_1) = \max_i \left\{ \text{diff}^2 \bar{h}_g(x_i) \right\} \qquad \text{for } x_{m_1} < x_i < x_v$$

and

$$\text{diff}^2 \bar{h}_g(a_2) = \max_i \left\{ \text{diff}^2 \bar{h}_g(x_i) \right\} \qquad \text{for } x_v < x_i < x_{m_2}.$$

Here, $\text{diff}^2 \bar{h}_g(x_i)$ denotes the second difference of $\bar{h}_g(x_i)$ in discrete domain, i.e.

$$\text{diff}^2 \bar{h}_g(x_i) = \bar{h}_g(x_{i+1}) - 2\bar{h}_g(x_i) - \bar{h}_g(x_{i-1}). \tag{6}$$

Then $[a_1, a_2]$ is said to be within the range of overlapping, which means that the pixels having gray levels within this interval may belong to either of the classes of regions. While sharpening the histogram, only the pixels having gray level $x_i$ [where, $a_1 \leqslant x_i \leqslant a_2$] will be affected.

To sharpen the histogram, we modify the gray level of every pixel $(j, k)$, if $a_1 \leqslant g(j,k) \leqslant a_2$, in the following way. We define a set $Q(j,k)$ that contains the gray levels of all the neighbors $(m,n)$ lying within a $I \times I$ window centered at $(j,k)$. Then we find a new value $\bar{f}(j,k)$ as

$$\bar{f}(j,k) = T_{sm}(g(j,k), Q(j,k)) \tag{7}$$

where $T_{sm}$ is a smoothing (i.e., mean filtering over $k$-nearest neighbors or median filtering) operator. It is evident that if a pixel $(j,k)$ having gray level $g(j,k)$ (where, $a_i \leqslant g(j,k) \leqslant a_2$) actually belongs to $Z_2$, then the probability that $g(m,n)$ is greater than $g(j,k)$ will be more than 0.5. As a result, $\bar{f}(j,k)$ will be greater than $g(j,k)$. So if $f(j,k) > g(j,k)$ then it may be predicted that $(j,k)$ belongs to $Z_2$. Similarly, if $f(j,k) < g(j,k)$ then it may be predicted that $(j,k)$ belongs to $Z_1$. This leads us to compute the gray levels of the pixels of the image with sharpened gray level histogram as follows

$$f(j,k) = \begin{cases} g(j,k) - \alpha\left[x_{m_1} - g(j,k)\right] & \text{if } \bar{f}(j,k) < g(j,k) \\ g(j,k) + \alpha\left[x_{m_2} - g(j,k)\right] & \text{if } \bar{f}(j,k) > g(j,k) \end{cases}, \tag{8a}$$

There are some situations when $f(j,k) = g(j,k)$. Then $f(j,k)$ is computed as

$$f(j,k) = \begin{cases} g(j,k) + \alpha\left[x_{m_1} - g(j,k)\right] & \text{if } g(j,k) \leqslant x_t \\ g(j,k) + \alpha\left[x_{m_2} - g(j,k)\right] & \text{if } g(j,k) > x_t \end{cases}. \tag{8b}$$

Equation (8) makes the overlapping region of the gray level histogram rarefied and produces deeper valleys. $x_t$ is the expected threshold where the bar height of the histogram will be zero after the algorithm converges. The parameter $\alpha$ controls the rate of convergence and for quick convergence the value of $\alpha$ is computed in the following way.

If a pixel $(j,k)$ actually belongs to $Z_2$ and if its gray level is $a_1$, then after one iteration its gray level should be modified to at least $x_t + 1$. On the other hand, if the pixel $(j,k)$ having gray level $a_2$ really belong to $Z_1$, then its gray level should be modified after one iteration to at most $x_t - 1$. This gives

$$\alpha = \frac{a_2 - a_1 + 2}{(x_{m_2} - x_{m_1}) + (a_2 - a_1)} \tag{9}$$

and

$$x_t = a_1 - \alpha[x_{m_2} - a_1] - 1 \tag{10}$$

or

$$x_t = a_2 + \alpha[x_{m_1} - a_2] + 1.$$

If we use the value of $\alpha$ obtained from (9), theoretically, only one iteration is sufficient to separate the gray level distributions for different classes of regions. However, for discreteness of data and also due to presence of quantization noise, we take a lower value for $\alpha$ and iterate the algorithm accordingly.

### *B. Shape of the sharpened histogram*

In this subsection, we examine the shape of the gray level histogram $h_f(u_i)$ of the image $[f]$ resulted from $[g]$ through the gray level transformation function as described by (8). Firstly, we find the range of gray level of pixels belonging to each of the classes of regions in the sharpened image. To find this range, let us rewrite (8) in terms of $x$ and $u$, i.e.,

$$u = T_1(x) = x + \alpha[x_{m_1} - x] \tag{11}$$

for

$$a_1 \leqslant x = g(j,k) \leqslant a_2 \text{ and } (j,k) \in Z_1.$$

and

$$u = T_2(x) = x + \alpha[x_{m_2} - x] \tag{12}$$

for

$$a_1 \leqslant x = g(j,k) \leqslant a_2 \text{ and } (j,k) \in Z_2.$$

So using (9), (11), and (12) we can calculate the ranges of gray level in $Z_1$ and $Z_2$ of $[f]$ as follows. From (11) we can say that if $x = a_1$, then $u = a_1 + \alpha[x_{m_1} - a_1] = a_3$ (say) $> 0$, and if $x = a_2$, then $u = a_2 + u[x_{m_1} - a_2] = x_t - 1$.

From (12) we can say that if $xa_1$, then $u = a_1 + \alpha[x_{m_2} - a_1] = x_t + 1$, and if $x = a_2$, then $u = a_2 + \alpha[x_{m_2} - a_2] = a_4$ (say) L − 1. So, it is evident that the gray levels of the pixels belonging to $Z_1$ will lie in the interval $[0, x_t - 1]$ and those of the pixels belonging to $Z_2$ will be in the interval $[x_t + 1, L - 1]$. The frequency of pixels in $[f]$ having gray level $x_t$ will be zero, i.e.,

$$h_f(x_t) = 0. \tag{13}$$

This implies that the gray level distributions for $Z_1$ and $Z_2$ are separated at $x_t$ and the gray level $X_i$ [where, $a_1 \leqslant x_i \leqslant a_2$] are mapped to either the interval $[a_3, x_t - 1]$ or the interval $[x_t + 1, a_4]$. As a result, $h_f(u_i)$ satisfies the following conditions

$$h_f(\alpha_i) \geqslant h_g(\alpha_i) \qquad \text{for } a_3 \leqslant \alpha_i \leqslant a_1 \tag{14a}$$

$$h_f(\alpha_i) < h_g(\alpha_i) \qquad \text{for } a_1 < \alpha_i \leqslant x_t - 1 \tag{14b}$$

$$h_f(\alpha_i) < h_g(\alpha_i) \qquad \text{for } x_t - 1 \leqslant \alpha_i < a_2 \tag{14c}$$

$$h_f(\alpha_i) \geqslant h_g(\alpha_i) \qquad \text{for } a_2 \leqslant \alpha_i \leqslant a_4 \tag{14d}$$

where, $\alpha_i$ is a dummy variable. Conditions (15) and (16) reveal that the histogram is sharpened. For more sharpness (8) is to be implemented iteratively.

### *C. Method to Obtain Improved Histogram*

This histogram sharpening algorithm is applicable to those cases where the gray level histograms are fairly bimodal, or where at least two significant peaks and the bottom of valley can be detected. In many cases, $h_g$ do not take the "bimodal shape," though there exist two distinct sets of regions in the original image. So to find out these hidden peaks, the gray level histogram can be computed in a different fashion. Algorithm 1, here, represents an improved method for computing histogram where peaks and bottom of valley can be detected more easily.

*Algorithm I*

*Step 1:* Initialize $h_g(x_i)$ as

$$h_g(x_i) \leftarrow 0 \text{ for all } x_i.$$

*Step 2:* If $g(j,k) = x_i$ then count the number of pixels having gray level to $x_i$ in $Q(j,k)$. Let the number be $\Omega$.

*Step 3:* $h_g(x_i) \leftarrow h_g(x_i) + \Omega$.

Hence, Algorithm 1 counts the frequencies of the gray levels of pixels lying interior to the objects or to the background heavily, whereas the frequencies of the gray levels of pixels lying in the vicinity of border will be counted at a lower rate. This increases the possibility of acquiring the bimodal shape by the gray level histogram.

## IV. Thresholding with Noise Cleaning

Finally, segmentation is achieved by transforming the processed image $\{f(j,k)\}$ on to a binary image $\{b(j,k)\}$ based on the threshold $x_t$. Now presence of some spurious noise in $\{f(j,k)\}$ may make $\{b(j,k)\}$ busy. We can get rid of this problem by applying median filtering to $\{f(j,k)\}$ before thresholding. Determination of median over a $3\times3$ window by a straightforward method requires $9\times8/2=36$ comparison. Chaudhuri [14] has developed an efficient algorithm that requires, on an average, only 11.25 comparisons for the same purpose. Gray level thresholding requires one comparison. Hence, approximately $11.25+1=12.25$ comparisons per pixel are required for desired segmentation. We can do this work with less computations as described in Method I.

*Method I*

Fig. 2 shows a $3\times3$ window around the candidate pixel $(j,k)$. For brevity, let $f(j,k)$ be written as $f$ and the gray levels of its neighboring pixels $f(j-1,k)$, $f(j-1,k-1)$, $f(j,k-1)$, $f(j+1,k-1)$, $f(j+1,k)$, $f(j+1,k+1)$, $f(j,k+1)$ and $f(j\ \ 1,k+1)$ be written as $f_1, f_2, f_3,\cdots,f_8$, respectively as shown in the figure. Now we label the pixel $(j,k)$ as an object point or as a background point depending on $f_i\,(i=0,1,2,\cdots,8,)$ so that the resulted binary image be equivalent to that obtained through median filtering/thresholding operations. The labeling procedure is described in Algorithm 2.

*Algorithm 2*

```
Step 0:  Consider a 3×3 window for a (j, k) as shown in Fig. 2.
Step 1:  $F1 \leftarrow 0$
Step 2:  for $i = 0$–$8$ do
         if $(f_i < = x_t)$ then $F1 \leftarrow F1 + 1$
Step 3:  if $(F1 > = 5)$ then $b(j,k) \leftarrow 1$
                          else $b(j,k) \leftarrow 0$
```

Repeat Step 0 to Step 3 until all $(j,k)$'s have been considered.

The experimental results of applying the Method I show that the binary image may still have some defects. These defects are

1) the border of the segmented regions may be coarse enough and recognition process should be preceded by smoothing the border for better results, and
2) some weakly connected portions of the object may be torn away and this will introduce a lot of difficulties in recognition of the object.

In Method II, it is attempted to solve both of the aforesaid problems.

*Method II*

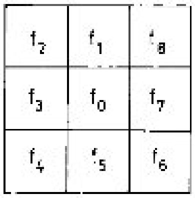
The approaches we adopt are neighbors counting and their connectedness in terms of the threshold $x_t$. Consider again Fig. 2. If at least five neighboring pixels of $(j,k)$ have gray level less than or equal to $x_t$, then $(j,k)$ will be assigned to object. This operation eliminates sharp concave turns on the object-boundary.

If at most three neighboring pixels of $(j,k)$ have gray level less than or equal to $x_t$, then $(j,k)$ will be assigned to background. This operation eliminates sharp convex turn on the object boundary.

Now if the number of neighboring pixels that have gray levels less than or equal to $x_t$ (or greater than $x_t$) is four, then any of the following three different situations may occur:

1) if $f_0 \leqslant x_t$, then $(j,k)$ will be assigned to the object;
2) if $f_0 > x_t$, then it evidently appears that $(j,k)$ should be assigned to the background.

However, this assignment may disconnect the pair of neighboring pixels that lie in the same object. It is assumed, here, that the thickness of the background is greater than one. In particular, consider the case where there are at least two sets of neighboring pixels (having gray level $\leqslant x_t$) around $(j,k)$. Pixels in each set are connected to each other, but the sets are mutually disconnected (e.g., ($f_3$ $f_3$ and $f_7$ $f_8$) and ($f_2$ and $f_4$ $f_5$ $f_6$) as shown in Figs. 3(a) and 3(b), respectively). For preservation of connectedness in the object we must look for the strongest among all possible paths (eg., for above two cases, the paths are ($f_4$ $f_5$, $f_0$ and $f_1$ $f_8$) and ($f_3$, $f_0$ and $f_1$ $f_8$ $f_7$) as shown by dotted lines in Figs. 3(a) and 3(b), respectively) connecting these sets of pixels. By the term "strongest path" we mean the path that consists of the darkest set of pixels. If $(j,k)$ represents the strongest path (i.e., $f_0 \leqslant f_i$, for all $f_i$ $(>x_t)$ lying on other paths), then $(j,k)$ will be assigned to the object, otherwise to the background.



|  |  |  |
|---|---|---|
| $f_2$ | $f_1$ | $f_8$ |
| $f_3$ | $f_0$ | $f_7$ |
| $f_4$ | $f_5$ | $f_6$ |

Fig. 2. A $3\times3$ window centered at candidate pixel $(j,k)$ having gray level $f_0$.

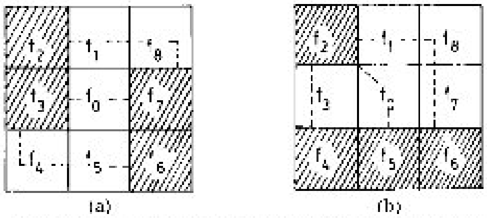


Fig. 3. Examples of disconnected sets of pixels of same object, and paths connecting them.

If $f_0 > x_t$ and all the pixels having a gray level less than or equal to $x_t$ are connected, that means, the questions of preservation of connectedness does not arise, then it is evident that $(j,k)$ should be labeled as the background point. However, here, we still assign $(j,k)$ to the object or to the background according to the condition that $f_0 \leqslant f_i$ for all $i$ such that $f_i > x_t$ is true or false, respectively. This is done to reduce computational complexity without introducing much error.

These operations are described in a more compact form in Algorithm 3 as given in the following

*Algorithm 3*

```
Step 0:  Consider a 3×3 window for a (j, k) as shown in Fig. 2,
Step 1:  $Fl \leftarrow 0$, $l \leftarrow 0$,
Step 2:  for $i = 1$ to $8$ do
         if $(f_i < = x_t)$ then $F_l \leftarrow Fl + 1$
             else
                 begin
                     $l \leftarrow l + 1$
                     $Y_l \leftarrow f_i$
                 end,
Step 3:  if $(Fl < 4)$ then $b(j,k) \leftarrow 0$
         else if $(Fl = 4)$ then
                 begin
                     $Y_{\min} = (Y_l | l = 1,2,3,4)$;
                     if $(f_0 < = Y_{\min})$ then $b(j,k) \leftarrow 1$,
                                          else $b(j,k) \leftarrow 0$,
                 end;
             else $b(j,k) \leftarrow 1$,
```

Repeat Step 0 to Step 3 until all $(j,k)$'s have been considered.

This algorithm requires eight comparisons in Step 2 and at most five comparisons in Step 3. Now flag $Fl$ may take any integer value from the interval $[0,8]$. If we assume that $Fl$ attains

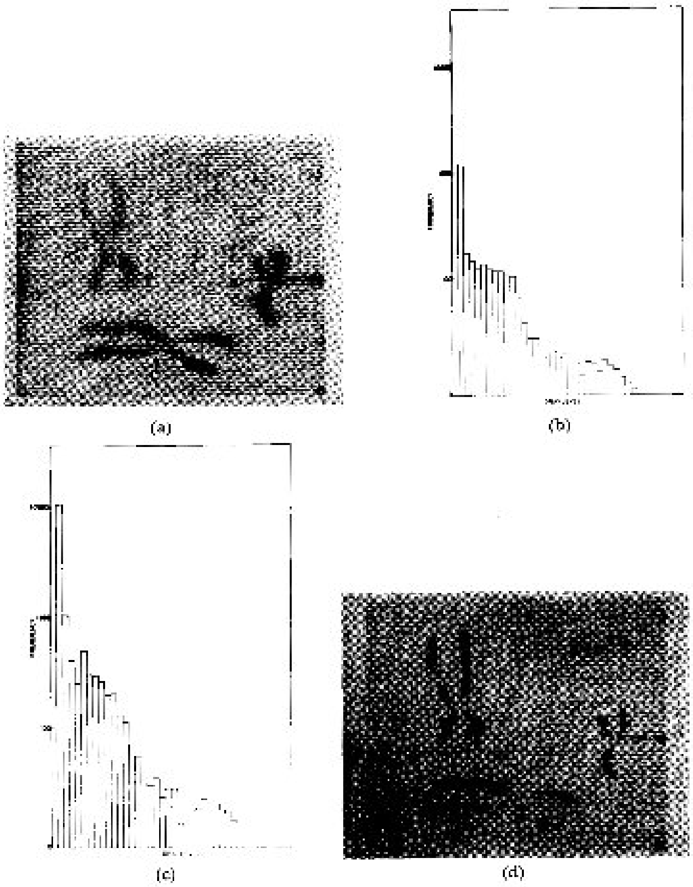


Fig. 4. Result of ordinary thresholding at bottom of valley of gray level histogram. (a) Original image. (b) Gray level histogram of (a) obtained through ordinary method. (c) Gray level histogram of (a) obtained through modified method (Algorithm 1). (d) Segmented image obtained by ordinary thresholding at $x_t$.

any value from the interval with equal probability, then Pr $(Fl = i) = 1/9$ for $i = 0, 1, 2, \cdots, 8$. In practice, it is seen that probability $(F1 = 4 < 1/9$. So on an average, less than 9.5 comparisons are required to implement Algorithm 3.

Both the Method I and the Method II i.e., Algorithm 2 and Algorithm 3, respectively have a common shortcoming. The object extracted by these methods is torn away at the places where they are of one pixel thick. Moreover, the lines and streaks will also disappear. However, to solve this problem Method III is suggested, which emphasizes preserving the connectedness in the object of any thickness. We assume here, again, the thickness of background is greater than one.

### *Method III*

In this method, the neighboring pixels, as well as the mutually disconnected sets of neighboring pixels belonging to the same object, are counted. Now if the number of disconnected sets of pixels be greater than one, the candidate pixel $(j, k)$ will be mapped to the object since it represents the common and the shortest path connecting all possible mutually disconnected sets.

If all the neighboring pixels of $(j, k)$ belonging to an object are connected to each other, then $(j, k)$ will be mapped to an object or background according to the number of $(m, n)$, the neighboring pixel of $(j, k)$ (such that $f(m, n) \leqslant x_t$) be greater than or less than four, respectively. But if this number is equal to four, $(j, k)$ will be mapped either to background or to object according to $f(j, k)$ is greater than $x_t$ or not. Algorithm 4 describes this procedure in the following concise form.

*Algorithm 4*

```
Step 0:  Consider a 3×3 window for a $(j, k)$ as shown in Fig. 2.
Step 1:  $Fl_1 \leftarrow 0$, $Fl_2 \leftarrow 0$, $i \leftarrow 0$, $R \leftarrow$ False
Step 2:  if $(f_i <= x_t)$ then
                    begin
                        $Fl_1 \leftarrow Fl_1 + 1$
                        $R \leftarrow$ True
                    end
Step 3:  while $i < 8$ do
         if $(R)$ then
                  if $(f_{i+1} <= x_t)$ then $Fl_1 \leftarrow Fl_1 + 1$
                  else
                      begin
                          $Fl_2 \leftarrow Fl_2 + 1$
                          $R \leftarrow$ False
                      end
              else
                  if $(f_{i+1} <= x_t)$ then
                      begin
                          $Fl_1 \leftarrow Fl_1 + 1$
                          $Fl_2 \leftarrow Fl_2 + 1$
                          $R \leftarrow$ True
                      end
Step 4:  if $(Fl_2 <= 1)$ then
                    begin
                    if $(Fl_1 > 4)$ then $b(j, k) \leftarrow 1$
                    else if $(Fl_1 = 4)$ then
                                 begin
                                 if  $(f_0 <= x)$  then
```

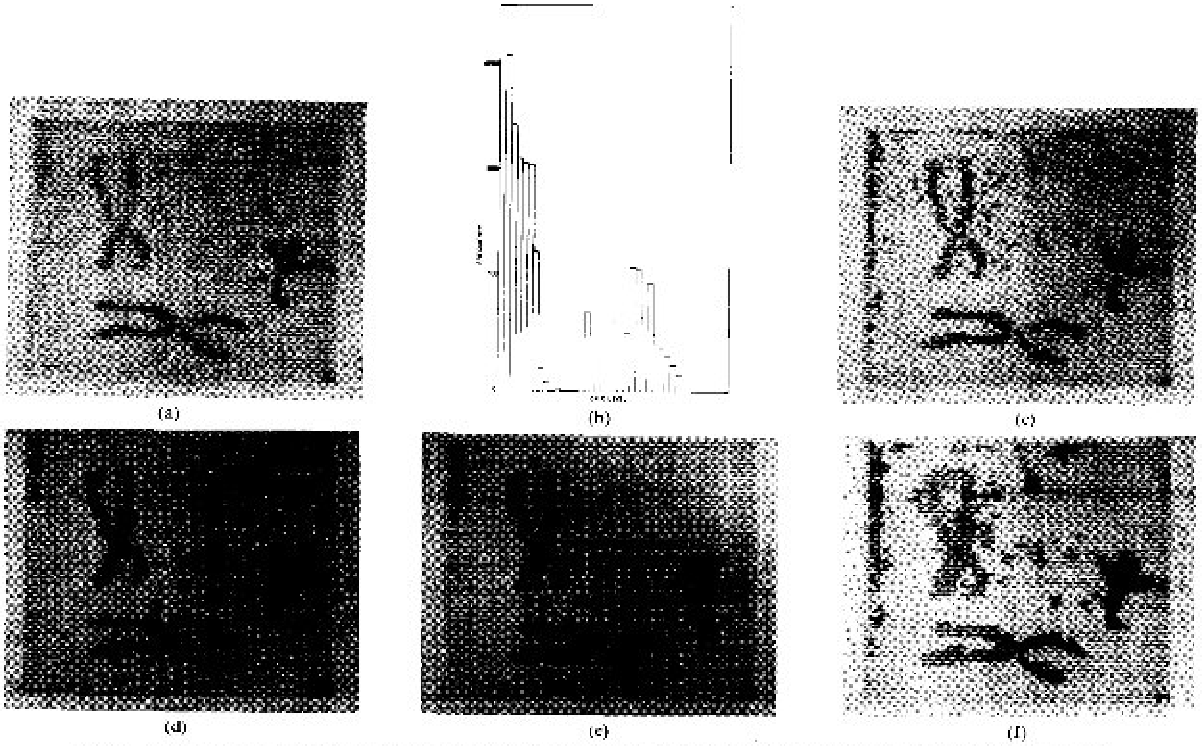


Fig. 5. Result of segmentation by proposed scheme for $\alpha = 0.5$ (a) Processed image after 1-iteration. (b) Gray level histogram of (a). Segmented image obtained by the following using $x_t$ as threshold. (c) Ordinary thresholding. (d) Applying Algorithm 2. (e) Applying Algorithm 3. (f) Applying Algorithm 4.

```
$b(j,k) \leftarrow 1$
                                                  else
$b(j,k) \leftarrow 0$
                                       end
                                 else $b(j,k) \leftarrow 0$
                            end
                       else $b(j,k) \leftarrow 1$
```

Repeat Step 0 to Step 4 until all $(j,k)$s' have been considered.

So the Algorithm 4 requires one comparison in Step 2, seven comparisons as a whole in Step 3, 1.5 comparison on an average in Step 4. Hence, in aggregate, 9.5 comparisons are required to implement Algorithm 4 for every pixel in the image. However, with this method a region may grow up or the area of objects may increase by connecting the densely spread noisy pixels. This is undesirable.

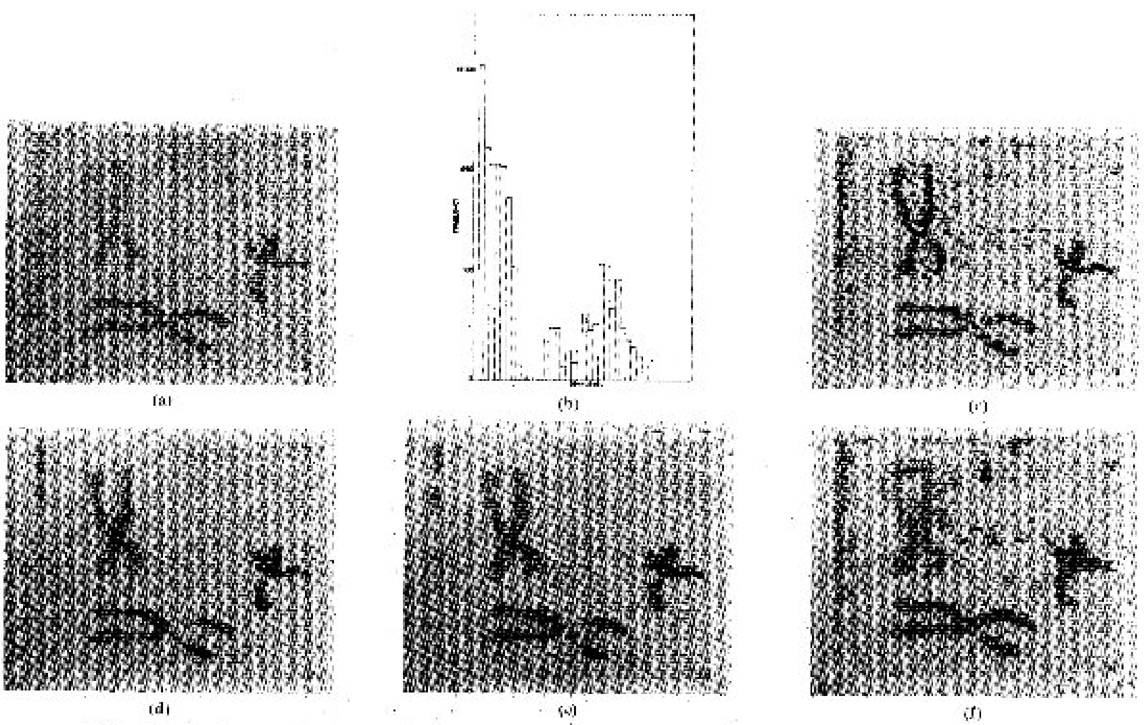
Finally, it should be noted that the number of computations required by Algorithm 2, Algorithm 3, and Algorithm 4 can still be reduced by deleting redundant computations common to adjacent neighboring pixels as is discussed in [14].

## V. Experimental Results and Discussion

The algorithms discussed in the previous sections are implemented on a chromosome image. Fig. 4(a) shows the original input image to be segmented. Figs. 4(b) and 4(c) exhibit its gray level histogram[1] obtained by applying ordinary and modified (Algorithm 1) methods, respectively. It is revealed that the Algorithm 1 produces a histogram that is more bimodal compared to that shown in Fig. 4(b). From Fig. 4(c) modes and bottom of valley $x_0$ can be detected more easily. Fig. 4(d) gives the segmented image of Fig. 4(a) by ordinary thresholding at $X_0$. For ease in discussion we have marked the chromosomes in Fig. 4(a) with A, B, and C. It is seen that the arms of the extracted chromosomes (as shown in the binary image Fig. 4(d)) are torn off at the centromeres, since the centromeres are less dark than the arms of chromosomes in case of uniform staining. Moreover, comparison between Fig. 4(a) and Fig. 4(d) reveals that a considerable number of object pixels are mapped to the background due to improper choice of threshold. As a result, there appear some discontinuities in the arms and the thickness of the arms becomes less also.

This leads us to enhance Fig. 4(a) so that its gray level histogram becomes sharp and also to compute the expected threshold $x_t$. This makes the gray level distributions for objects and background are separated at $x_t$, i.e., $h_g(x_t) = 0$. The value of $\alpha$ and $x_t$ are computed from (9) and (10), respectively. The value of $\alpha$, here, is 0.5 and only $l$-iteration is required by the algorithm to be converged. Figs. 5(a) and 5(b) show the enhanced image and its gray-level histogram, respectively. The segmented images are shown in Figs. 5(c)–5(f) obtained by ordinary thresholding, applying Algorithm 2, applying Algorithm 3, and applying Algorithm 4, respectively, with $x_t$ as threshold. In Fig. 5(c) the boundaries of the objects are coarse, the arms are still detached at the centromere except chromosome A and densely spread noisy pixels that emerge out at the top-lefthand corner of the figure. Algorithm 2 cleans out the noises and makes the boundaries of the object smooth as shown in Fig. 5(d). However, arms of chromosomes B and C are still detached at their respective centromere. Algorithm 3 connects the arms of chromosome B at the centromere due to its capability of connecting thick limbs of an object separated by single pixel gap. But boundaries of the objects in Fig. 5(e) are coarse compared to those in Fig. 5(d). Fig. 5(f) reveals that Algorithm 4 is best regarding the capability of connecting different parts of the same object. In Fig. 5(f) arms of all the chromosomes are connected, and consequently, noisy

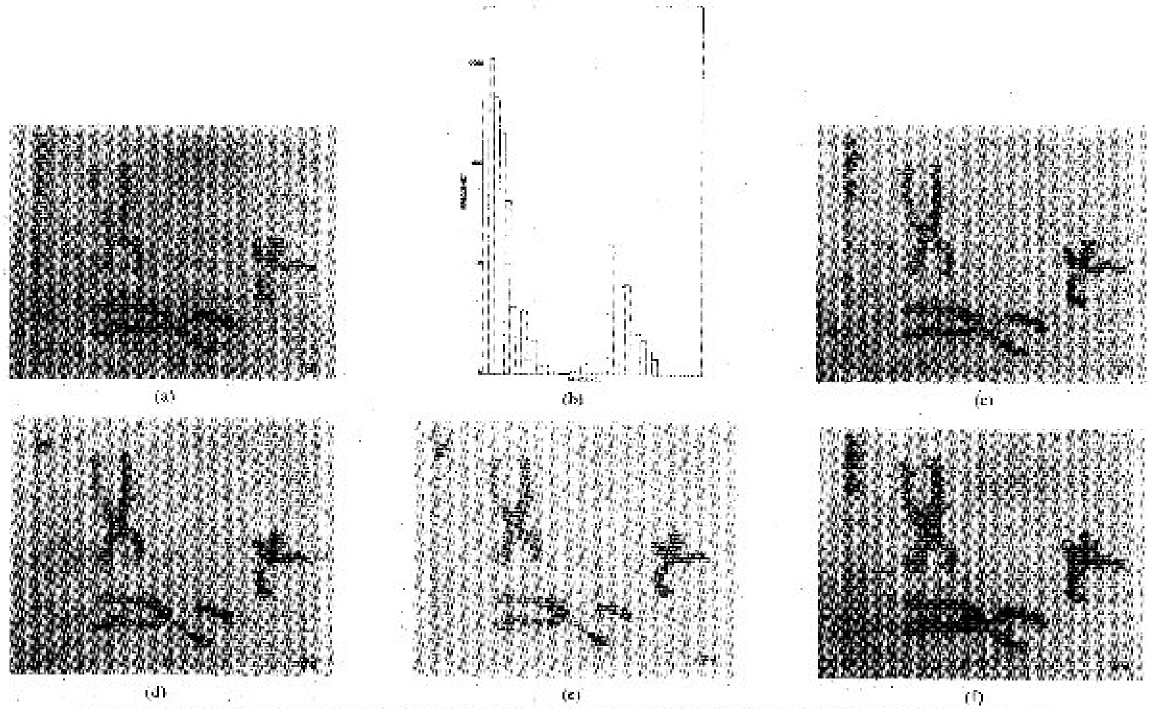
[1] In plotting the gray level histograms, here the gray scale is reversed, that means, higher values in gray scale represent lower intensities and lower gray levels represent higher intensities.



(a) (b) (c)

(d) (e) (f)

Fig. 6. Result of segmentation by proposed scheme after 1-iteration for $\alpha = 0.47$. (a) Processed image. (b) Gray level histogram of (a). Segmented image obtained by the following using $x_t$ as threshold. (c) Ordinary thresholding. (d) Applying Algorithm 2. (e) Applying Algorithm 3. (f) Applying Algorithm 4.



(a) (b) (c)

(d) (e) (f)

Fig. 7. Result of segmentation by proposed scheme after 2-iteration for $\alpha = 0.47$. (a) Processed image. (b) Gray level histogram of (a). Segmented image obtained by the following using $x_t$ as threshold. (c) Ordinary thresholding. (d) Applying Algorithm 2. (e) Applying Algorithm 3. (f) Applying Algorithm 4.

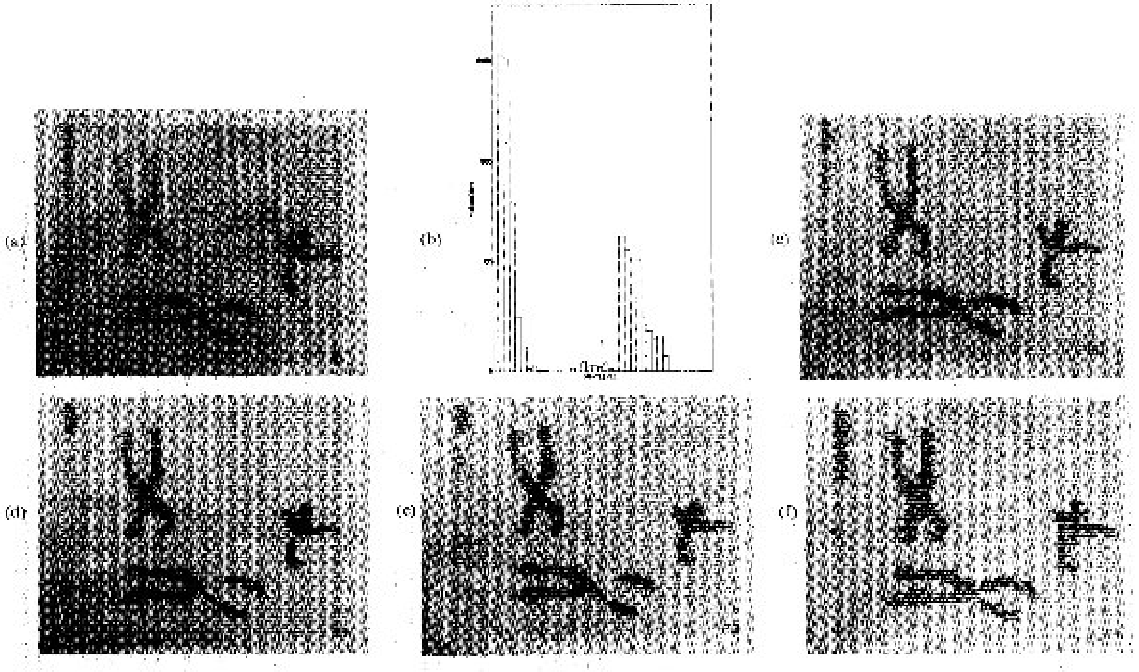


Fig. 8. Result of segmentation by proposed scheme after 3-iteration for $\alpha = 0.47$. (a) Processed image. (b) Gray level histogram of Fig. (a). Finally segmented image obtained by the following using $t_1$ as threshold. (c) Ordinary thresholding. (d) Applying Algorithm 2. (e) Applying Algorithm 3. (f) Applying Algorithm 4.

pixels also become connected to each other to grow regions having noticeable area rather than being removed.

This happened due to the overenhancement of Fig. 4(a) before applying the thresholding technique. So we take the value of $\alpha$ less than 0.5 for example, here, we have taken $\alpha = 0.47$. The algorithm is converged after 3-iteration. Results are shown in Fig. 6, Fig. 7, and Fig. 8 after 1-, 2-, and 3-iteration, respectively. Parts (a) and (b) in Figs. 6–8 show the enhanced images and its gray level histogram. Segmented images due to ordinary thresholding, Algorithm 2, Algorithm 3, and Algorithm 4 are presented in parts (c), (d), (e), and (f) of Figs. 6–8, respectively using same $X_t$ (as in Fig. 5) as the value of threshold. It is seen that the results shown in Fig. 8 are better than those shown in Fig. 5.

## VI. Conclusion

In this paper, a simple but effective scheme for image segmentation is described. This is a modified form of mode method. However, here, the value of the threshold is computed, unlike the conventional method where this value is heuristically selected. The expression for computing this value is derived from gray level transformation function used for histogram sharpening. This transformation, as a result, reduces error of misclassification during thresholding. The implementation of this transformation requires the histogram to be bimodal. A powerful measure for bimodality and a new Algorithm 1 for computing histogram-embedding higher bimodality are also proposed. Finally, three different thresholding algorithms are suggested. The first one is most efficient in noise removing and the third one most effectively preserves the connectedness between weakly connected parts of the same object. The second algorithm removes noise and smooths object boundaries better than the third algorithm, and also preserves the connectedness. The methods are not expensive, either.

The scheme is developed for the class of image that contains only two types of regions. However, it can be extended to images where more than two types of regions are present. In other words, the scheme can be modified to make it applicable on multimodal histogram.

## Acknowledgment

The authors express their thanks to Mrs. S. De. Bhowmick for typing the manuscript.

## References

[1] K. S. Fu and J. K. Mui, "A survey on image segmentation," *Pattern recognition*, vol. 13, pp. 3–16, 1981.

[2] J. S. Weszka, A survey of threshold selection techniques, *CGIP*, vol. 7, pp. 259–265, 1978.

[3] A. Rosenfeld and A. C. Kak, *Digital picture processing*. New York: Academic 1976.

[4] W. Doyle, "Operations useful for similarity invariant pattern recognition," *J. ACM*, vol. 9, pp. 259–267, 1962.

[5] J. M. S. Prewitt and M. L. Mendelsohn, The analysis of cell images, *Trans. N.Y. Acad. Sc.*, vol. 128, pp. 1035–1053, 1966.

[6] B. Chandra, B. B. Chaudhuri and D. Dutta Majumder, "Noise removing image segmentation through gray level histogram sharpening," Tech. Rep. ECSU/1/84, Electronics and Comm. Sc. Unit, Indian Statistical Institute, Calcutta, India, 1984.

[7] A. Rosenfeld and P. Torre, Histogram concavity analysis as an aid in threshold selection, *IEEE Trans. Syst. Man Cybern.*, vol. SMC-13, no. 2, pp. 231–235, Mar./Apr. 1983.

[8] J. S. Weszka, R. N. Nagel and A. Rosenfeld, "A threshold selection technique," *IEEE Trans. Comput.*, vol. C-23, pp. 1322–1326, 1974.

[9] J. S. Weszka and A. Rosenfeld, "Threshold selection techniques 5," TR-349, Computer Science Center, Univ. of Maryland, 1975.

[10] A. Rosenfeld and L. S. Davis, Iterative histogram modification, *IEEE Trans. Syst. Man Cybern.*, vol. SMC-8, pp. 300–302, 1978.

[11] S. Peleg, Iterative histogram modification 2, *IEEE Trans. Syst. Man Cybern.*, vol. SMC-8, pp. 555–556, 1987.

[12] K. A. Narayanan and A. Rosenfeld, Image smoothing by local use of global information, *IEEE Trans. Syst. Man Cybern.*, vol. SMC-11, no. 12, pp. 826–831, Dec. 1981.

[13] A. Rosenfeld and J. S. Weszka, "Pattern recognition and scene analysis," *Computer*, pp. 23–28, 1978.

[14] B. B. Chaudhuri, "Efficient algorithm for image enhancement," in *Proc. IEEE*, vol. 130, pp. 95–97, 1983.