pointed out for an ideal error-free measurement system, the sensitivity of Bayes' rule in feature selection is sometimes quite low and hence should be treated carefully for the nonideal case also. Nevertheless, any strategy for feature selection should include a component with minimum average measurement error.

Relaxation of the assumption that stage 1) is perfect may be treated as follows. If the number of prototypes is small but feature measurement is accurate, the estimated parameters may deviate from its true value. If the deviation is small, the tolerance of misclassification probability may be studied by sensitivity analysis. This problem along with nonparametric classification problem in a noisy environment will be reported in future.

## ACKNOWLEDGMENT

The authors acknowledge the helpful discussions with Mr. S. K. Parui of the E.C.S. Unit of the Indian Statistical Institute. One of the authors wishes to thank Leverhulme Trust for providing him with a Visiting Fellowship at Queen's University, Belfast. The help by Mrs. M. Simpson and Mrs. N. McAllister in preparing the manuscript is gratefully acknowledged.

## REFERENCES

[1] H. C. Andrews. *Introduction to Mathematical Techniques in Pattern Recognition.* New York: Wiley-Interscience, 1972.
[2] K. Fukunaga, *Introduction to Statistical Pattern Recognition.* New York: Academic, 1972.
[3] W. H. Tsai and K. S. Fu, "A pattern deformational model and Bayes' error-correcting recognition system," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-9, no. 12, pp. 745–756, Dec. 1979.
[4] M. Ben-Bessat, "On the sensitivity of the probability of error rule for feature selection," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-2, no. 1, pp. 57–60, Jan. 1980.
[5] M. Ben-Bessat, K. L. Kløve, and M. H. Weil, "Sensitivity analysis in Bayes' classification models: Multiplicative deviations," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-2, no. 3, pp. 261–266, May 1980.
[6] P. H. Swain, "Bayesian classification in time-varying environment," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-8, no. 12, pp. 879–883, Dec. 1978.
[7] G. R. Dattatreye and V. V. S. Sarma, "Bayesian and decision tree approaches for pattern recognition including feature measurement cost," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-3, no. 3, pp. 293–298, May 1981.
[8] D. G. Lainiotis, "A class of upper bounds on probability of error for multihypotheses pattern recognition," *IEEE Trans. Inform. Theory*, vol. IT-15, no. 6, pp. 730–731, Nov. 1969.
[9] R. M. Hawkes and J. B. Moore, "An upper bound on the mean square error for Bayesian parameter estimation," *IEEE Trans. Inform. Theory*, vol. IT-22, no. 5, pp. 610–615, Sep. 1976.

# Image Description and Primitive Extraction Using Fuzzy Sets

SANKAR K. PAL, MEMBER, IEEE, ROBERT A. KING, AND ABDULLAH A. HASHIM, MEMBER, IEEE

***Abstract*—An application of the theory of fuzzy sets in understanding an image is demonstrated. The task of understanding consists of three parts—1) representation of image contours by their respective chains of octal codes, 2) smoothing of the chain to remove spurious wiggles, and 3) segmentation and assignment of degree of "arcness" to each segmented smoothed chain in order to extract their primitives for description. Octal code is provided to a two-pixel or, even more length contour by taking maximum of its grades of membership corresponding to "vertical," "horizontal," and "oblique" lines. Four different smoothers have been used to eliminate the spurious wiggles occurring in the contours. Segmentation of edges into different curves and lines is made on the basis of constant increase / decrease in code values. The primitives for description and interpretation of images are extracted by assigning the appropriate membership value which provides a measure of curvature to the different arcs. The sense of curving is also determined. The effectiveness of the algorithm is demonstrated when a gray tone edge-detected X-ray image of wrist is considered as input.**

Manuscript received March 5, 1982; revised August 19, 1982.

S. K. Pal is with the Electrical Engineering Department, Imperial College of Science and Technology, London SW7 2BT, England, on leave from the Electronics and Communication Sciences Unit, Indian Statistical Institute, Calcutta 700 035, India.

R. A. King is with the Electrical Engineering Department, Imperial College of Science and Technology, London, England.

A. A. Hashim is with the School of Electronics and Electrical Engineering, Leicester Polytechnic, Leicester LE1 9BH, England.

## I. INTRODUCTION

Pictorial pattern recognition may be considered to be a twofold task, namely, image processing and image understanding. Image processing primarily involves enhancement, restoration, smoothing, sharpening, and other noise-reduction techniques in order to isolate the objects in the picture. Input and output are therefore both the images, with output an improved version of input. In image understanding the input is an image, but the output is an interpretation and description of the various figures characterizing the objects (contents) of the input image. This correspondence presents an algorithm for computer-based description and interpretation of contour outlines of objects in an image.

The various approaches developed so far for line representation, shape description, and figure interpretation [1]–[17] in an image are based on either the heuristic principles in artificial intelligence or on the topological information of patterns. The heuristic way of interpretation and recognition involves selection of an observation and then its evaluation with respect to the predictions made by the current hypothesis. Rubin [13] used an artificial intelligence search technique (called locus) to recognize the major buildings, rivers, and other objects from the photographs of the city of Pittsburgh, PA, given a knowledge base of over 50 objects. Other investigators [14] used edge-detection and line-finding techniques for linear feature extraction, which identifies the linear features like roads and airport runways from the aerial images. Topological properties of patterns (geometric configurations), on the other hand, have been successfully used in the cases of description of alphanumeric characters [4], recognition of handwritten characters [9], interpretation of finger prints [5], studying the connectedness and convexity of figures [2], and finding the convex hull of a digitized figure [15]. Recently, Tou [15] developed an algorithm for computer-based interpretation and description of different curves, single-loop, and multiple-loop figures using their topological information. This information was extracted from the corresponding octal-coded chains which were assumed to be the input to the system. In practice the conversion from a contour to its coded version poses many problems.

In this correspondence we present a model, Fig. 1, which demonstrates an application of the theory of fuzzy sets [18]–[22] in automatic interpretation and description of gray tone edge-detected images. The procedure involves the following operations.

1) Encoding of image contours in order to represent them by their respective one-dimensional octal-coded strings. Octal chain codes are used because of storage requirements, ease of manipulation, ease with which it can be adapted to standard display techniques, and simple topological properties [15], [23]. We use this code to describe multiple-pixel length contour by taking the maximum of its grades of membership corresponding to "vertical," "horizontal," and "oblique" lines.

2) Smoothing of the chains to remove the spurious wiggles in the contours. Here we have used four smoothing operations.

3) Segmentation and assignment of degree of "arcness" to each

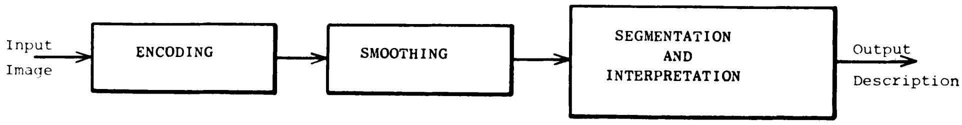



Fig. 1. Block diagram of description model.

segmented smoothed chain in order to extract the primitives of the contours for interpretation and description of image. Degree of "arcness" is provided by defining a suitable fuzzy membership function which measures the amount of curvature of an arc.

In this model the input is assumed to be an edge-detected image. It should be noted that there are numerous algorithms available in the texts [24]–[28] for detecting the edge contours of objects in an image. The effectiveness of our algorithm is demonstrated when a gray tone edge-detected image of an X-ray of the wrist is considered as input. Digital computer CDC-6400 was used for numerical analysis.

## II. Fuzzy Sets and Membership Function

A fuzzy set $A$ with its finite number of supports $x_1, x_2, \cdots, x_n$ in the universe of discourse $U$ is defined as

$$A = \{\mu_A(x_i)/x_i\},$$

where the membership function $\mu_A(x_i)$ having positive value in the interval [0, 1] denotes the degree to which an event $x_i$ may be a member of or belong to $A$.

Let us now define membership functions of some basic primitives which have been used here for interpretation and description of X-ray images. These primatives are often encountered in generation and recognition of geometrical figure, character, numeral, finger print, etc. The fuzzy set of straight lines labeled "vertical," "horizontal," and "oblique" are defined as

$$\text{vertical} = \int_x \mu_V(x)/x \tag{1a}$$

$$\text{horizontal} = \int_x \mu_H(x)/x \tag{1b}$$

$$\text{oblique} = \int_x \mu_{\text{Ob}}(x)/x, \tag{1c}$$

where

$$\mu_V(x) = \begin{cases} 1 - \left|\dfrac{1}{m_x}\right|^{F_e}, & |m_x| > 1 \\ 0, & \text{otherwise} \end{cases} \tag{2a}$$

$$\mu_H(x) = \begin{cases} 1 - |m_x|^{F_e}, & |m_x| < 1 \\ 0, & \text{otherwise} \end{cases} \tag{2b}$$

$$\mu_{\text{Ob}}(x) = \begin{cases} 1 - \left|\dfrac{\theta - 45}{45}\right|^{F_e}, & 0 < |m_x| < \infty \\ 0, & \text{otherwise.} \end{cases} \tag{2c}$$

$m_x$ ($= \tan\theta$) is the gradient of a straight line $x$ making an angle $\theta$ with the horizontal line $H$, as shown in Fig. 2(a). $\mu_V(x)$, $\mu_H(x)$, and $\mu_{\text{Ob}}(x)$ represent the membership function for vertical, horizontal, and oblique, respectively, of this line $x$ such that

$$\begin{aligned} \mu_V(x) &\to 1 \quad \text{as } |\theta| \to 90^\circ \\ \mu_H(x) &\to 1 \quad \text{as } |\theta| \to 0^\circ \\ \mu_{\text{Ob}}(x) &\to 1 \quad \text{as } |\theta| \to 45^\circ \end{aligned}$$

and

$$\mu_V(x) \gtrless \mu_H(x) \text{ as } |\theta| \gtrless 45^\circ.$$

The positive constant $F_e$ which controls the fuzziness in a set is known as the exponential fuzzifier [21], [29].

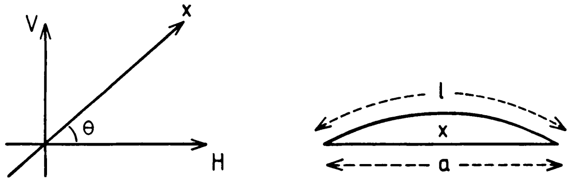



Fig. 2. Membership functions. (a) Line. (b) Arc.

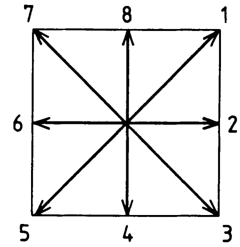



Fig. 3. Direction of octal codes.

Similarly the grade of membership of a line segment $x$ to be an "arc" may be expressed by the function

$$\mu_{\text{arc}}(x) = \left(1 - \frac{a}{l}\right)^{F_e}, \tag{3}$$

where $a$ is the length of the line joining the two extreme points of an arc $x$ as shown in Fig. 2(b) and $l$ is the arc length such that the lower the ratio $(a/l)$ is the higher is the degree of "arcness." In other words it provides a quantitative measure of curvature of an arc.

In the next section we will show how the previously described membership functions have been implemented in order to encode and interpret the image contours.

## III. Encoding

The contours of an $M \times N$ dimensional gray tone edge-detected image were encoded into one-dimensional symbol strings having "NC" octal codes ($\text{Code}_i$, $i = 1, 2, \cdots$, NC) by using the rectangular (octal) array method shown in Fig. 3. In this method an octal code is used to describe a $w$-pixel ($w > 1$) length contour by taking the maximum of its grades of membership corresponding to "vertical," "horizontal," and "oblique" lines (1). This approximation of using a $w$-pixel (instead of one pixel) length line saves computational time and storage requirement without affecting the system performance. The block diagram for encoding images is available from the author (also see Appendix A).

During scanning operation of the image the system first looks for a nonzero pixel intensity (starting point ($xsp$, $ysp$)). Its movement to the next pixel is then determined by the "max" operator on the values of the eight neighboring pixels. In case the max value is possessed by more than one pixel (i.e., at a multiple-crossing pixel, as shown in Fig. 4), the final decision is taken by choosing that direction which most closely matches that of the previous description of the path. Of course this matching procedure is not used until some codes (say NC = 5) have been generated. A flow chart of the matching algorithm (available from the authors) with respect to codes ($\text{Code}_j$, $j = 1, 2, \cdots$, NALT) shows that there are "NALT" number of possibilities of the resulting movement. In the case of ambiguity a maximum of "NMATCH" preceding codes are considered to decide the continuing direction. If this is still indecisive the first code is accepted.

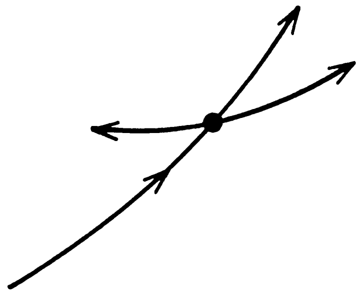


Fig. 4. Multiple-crossing pixel.

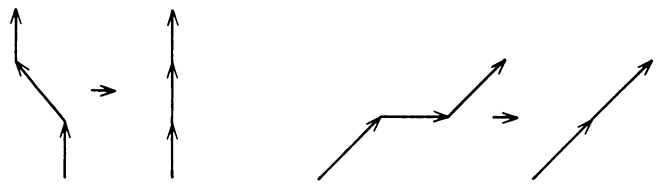



Fig. 5. Principles. (a) Replacement. (b) Deletion.

TABLE I
REPLACEMENT DIGITS FOR DIFFERENT COMBINATIONS OF CODES

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| 1 | | | 22 | 2 | | 8 | 88 | |
| 2 | | | | 3 | 4 | | 8 | 1 |
| 3 | 22 | | | | 44 | 4 | | 2 |
| 4 | 2 | 3 | | | | 5 | 6 | |
| 5 | | 4 | 44 | | | | 66 | 6 |
| 6 | 8 | | 4 | 5 | | | | 7 |
| 7 | 88 | 8 | | 6 | 66 | | | |
| 8 | | 1 | 2 | | 6 | 7 | | |

It is to be mentioned here that all but the multiple-crossing pixels are deleted after their consideration. Retention of the multiple-crossing pixels enables us to use these pixels on the return path, and hence to continue the description of other contours connected to them. Although the starting pixel for each contour has been initially deleted, we need to replace it after a certain length of string has been produced. This permits the system to decide if the contour is closed. Provision is also made in the algorithm for temporarily storing the coordinates $((xst, yst)_j,\ j = 1, 2, \cdots, 10)$ of the ten previous pixels. This allows the machine to be able to retrace its path in case it gets lost in undesirable side tracks (lost path) which may have been generated from the main one during the operation of edge detection of the image. The contour description corresponding to the lost path is not finally taken into consideration.

## IV. SMOOTHING

After a string of NC octal codes has been produced by the encoding algorithm (as described above), we use four different smoothers on it to eliminate those symbols which were generated due to spurious wiggles in the contours. The resulting smoothed chain then consists of NCM symbols where NCM $\leqslant$ NC. Let us now explain the principles of the four smoothing algorithms which have been tested in the smoothing block of Fig. 1.

### *A. Smoother-1*

A code which is preceded and followed by a specific number (e.g., four) of other identical codes is either replaced by the neighboring code if it represents an oblique line, or deleted, otherwise. For example consider the combination "* * * + *." If the asterisks and the plus sign represent horizontal/vertical and oblique lines, respectively, we simply replace the plus sign by an asterisk, keeping the number of codes constant. On the other hand if the plus sign denotes either a horizontal or a vertical line we delete it, thus making a reduction in codes by unity. The reduction technique is logical if we look at Figs. 5(a) and 5(b) corresponding to two different cases. For both cases the length of curves (the net displacement) after smoothing remains unaffected.

The same principle of replacement/deletion is also applied for the combination "* + + * * * * *" or its other forms where two identical codes + + are preceded and followed by, e.g., six other identical codes.

### *B. Smoother-2*

In Smoother-1 we see that the number of changes in a set of codes to be smoothed is two. The second smoother, on the other hand, smooths all those combinations having three or four such changes in code values, e.g., a) "* + .* * *" or other combinations where four identical codes preceded and followed two other codes and b) "* + .?* * * * *", "* + .+ * * * * *", "* + +.* * * * *", or any other combinations where six identical codes preceded and followed only three other codes. A flowchart for Smoother-2 (available from the authors) shows that the typical replacement/deletion procedure is the same as used in Smoother-1. $\Delta_j$ is equal to five and eight for categories a) and b), respectively.

### *C. Smoother-3*

Two adjacent inverse codes (like one and five, two and six, three and seven, and four and eight) are omitted in this algorithm. A simple way to find the inverse of a code is to add/subtract four if it is less/greater than four. Another way is by adding "four" modulo eight.

### *D. Smoother-4*

Smoother-4 is based on the principle that within a group of four (or three) codes having zero-total vector rotation, each pair (or only one pair) is replaced by a digit or a pair of digits depending upon its combination. Table I illustrates all such possible replacement digits when the absolute difference (subtract from eight if the difference is greater than four) in code values of a pair is either two or three. When the difference is unity we simply delete the code representing vertical or horizontal line.

Double-digit entries are shown in Table I when the pair is formed between two codes denoting oblique lines. Such a replacement does not permit reduction in the length of string.

## V. SEGMENTATION AND INTERPRETATION

The next task before extraction of primitives and interpretation of contours is the process of segmentation. The flowchart for segmenting (available from the authors) the smoothed chain into different curves and lines shows that splitting up of the chain is dependent on the constant increase/decrease in code values. For extracting an arc we segment the string at a position whenever we find a decrease/increase after constant increase/decrease in values of codes. Again, if the number of codes between two successive changes exceeds a prespecified limit (LARC), a straight line is said to exist between the two curves. In the case of a closed curve we have kept provision for increasing the length of the chain by adding the first two starting codes to the tail of the string. This enables one to take the continuity of the chain into account in order to reflect its proper segmentation.

After segmentation we need to provide a measure of curvature along with direction to the different arcs, and also to measure the length of lines in order to extract the primitives of contours. The degree of "arcness" is obtained by assigning the appropriate membership values $\mu_{arc}$ (3) to them. For measuring the length of a line we assume that "LRATE" number of codes make one unit of straight line where LARC = LRATE + 1. As soon as the final segment position of any segment is obtained, the system computes the number of lines (preceding) and then $\mu_{arc}$ of the following arc. This consecutive measurement process continues for each segment until the end of the chain is reached, when it finally computes the number of ending lines.

For measuring $\mu_{arc}$ using (3) we have used the codes between initial and final segment positions (inclusive) in order to compute the length $l$ and the diameter $a$ of the arc. If a code represents an oblique line, the corresponding increase in arc length would be by $\sqrt{2}\,w$, $w$ being the pixel-length of line denoting the code. Otherwise increase $l$ by $w$. Arc diameter is computed by measuring the resulting shifts $\Delta m$ and $\Delta n$ of spatial coordinates due to those codes in question. For example we consider a sequence of codes

5 6 6 7

denoting an arc where $\text{Code}_{SGMI} = 5$ and $\text{Code}_{SGMF} = 7$. Here we have for $w = 1$ and $F_e = 0.5$

$$\Delta m = 1 + 0 + 1 - 1 = 0$$

$$\Delta n = -1 - 1 - 1 - 1 = -4$$

$$a = \sqrt{(\Delta m^2 + \Delta n^2)} = 4$$

$$l = 1.414 + 1 + 1 + 1.414 = 4.828$$

and

$$\mu_{arc} = 0.414.$$

Since $\text{Code}_{SGMF}$ is greater than $\text{Code}_{SGMI}$ the sense of the curve is positive (clockwise).

## VI. Implementation and Results

Fig. 6 shows an 145 × 128 dimensional gray tone edge-detected image [22], [27] of a part of the wrist containing radius (with epiphysis and metaphysis) and a part of two carpal bones. To implement the abovementioned algorithm we have considered $w = 2$, LRATE = 3, NMATCH = 10 and $F_e = 0.5$. A computer-based description of the contours is explained in Fig. 7.

Here, $L$, $A$, and $\bar{A}$ denote the "straight line," "clockwise arc," and "anticlockwise arc," respectively. Suffices of $L$ and $A$ represent the number of line units and the degree of "arcness" of the arc $A$. For example $L_3$ denotes three units of straight line, and $\bar{A}_{0.541}$ indicates that the sense of the arc is negative and its degree of "arcness" measuring the amount of curvature is 0.541.

Now among these varying membership values of arc, one can use a threshold according to his problem in order to select a specific set of primitives. These primitives, which can describe the contours, would then be used to develop a grammar such that an unknown string representing some contour can automatically be recognized using the syntactic pattern-recognition technique.

As a typical illustrative case, the coded strings (before and after smoothing) of only contour ii) have been shown here. The arrows (↓) indicate the positions of segmentation. Application of the four smoothers has reduced the lengths of the strings (except for contour v) greatly.

Again we consider a part (shown by the arrowed lines in Fig. 6) of contour ii) as an example to demonstrate the encoding and smoothing algorithms, which have been described in Sections III and IV. The octal-code representation before and after smoothing operations (with $w = 2$) of that part of the contour is shown by the respective underlined strings. The corresponding pictorial representation of these strings is sketched in Figs. 8(a) and 8(b). Fig. 8(a) gives an approximate version of the part of the contour in question as obtained using algorithms based on the concept of fuzzy sets. Some of the undesirable small loops are also found to be eliminated. The four algorithms have further approximated the shape of that segment (Fig. 8(b)) by smoothing the direction of contour, thus simplifying the problem of extracting its primitives. It should be noted here that the first two smoothers may disturb the total bend around a contour (Fig. 8), but as an approximation it does not really affect much as far as its syntactic recognition is concerned.

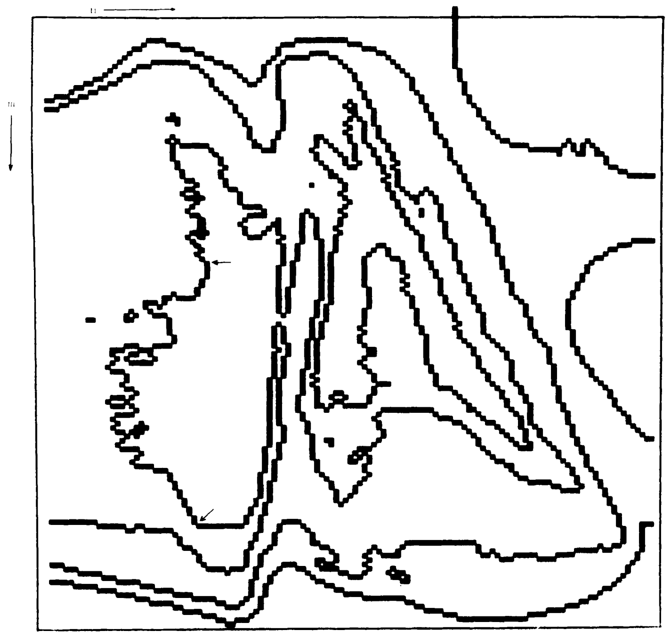


Fig. 6. Input image showing gray tone contour outlines of part of wrist.

## VII. Conclusion

The concept of fuzzy sets is found to be applied successfully to the problems of computer-based description and interpretation of gray tone contour outlines of an X-ray image. The ultimate aim is to recognize the images using the syntactic pattern-recognition technique, which needs a suitable grammar to be developed before classification of the image contours. The extracted primitives can therefore be used to program such a grammar.

Contours have been encoded using fuzzy membership functions corresponding to "vertical," "horizontal," and "oblique" lines. Although we have considered the line segments of two-pixel length it can be extended to longer line segments, making the encoding technique more approximate but yet effective. The problem of choosing final code at a multiple crossing point is solved by measuring the closeness of the possible code values with the previous codes.

Smoothing algorithms are seen to be effective in removing the spurious wiggles in a contour, and hence make the task of their primitive extraction more convenient. These also reduce the amount of data to be processed in the following segmentation algorithm. The four smoothers we have programmed here are sufficient to serve our purpose and are also expected (because of their flexibilities) to be encountered very much in any practical problem of the same line. Degree of "arcness" providing a measure of curvature is used to extract the primitives of contours for their final description.

i) Starting point of contour : (22,1)
End of contour is reached at (129,1)
Number of octal codes = 166
Number of smoothed octal codes = 138
Description of the contour is as follows

$L_3\ A_{0.465}\ L_2\ \bar{A}_{0.541}\ L\ A_{0.272}\ L\ A_{0.272}\ L_{15}\ A_{0.533}\ L_7\ A_{0.272}\ L_3\ \bar{A}_{0.465}\ L_2\ A_{0.541}\ \bar{A}_{0.272}\ L\ A_{0.272}$

ii) Starting point of contour : (24,1)
End of contour is reached at (119,1)
Number of octal codes = 355
Octal coded chain is as follows

2 1 1 1 1 1 1 1 1 1 1 2 2 2 3 3 3 3 3 3 3 3 3 3 3 2 8 1 1 8 8 7 8 8 8 1 1 2 1 2
3 3 3 3 3 3 3 3 4 3 3 3 3 4 3 4 3 3 1 2 3 3 4 3 4 3 4 3 4 3 3 3 3 3 4 4 3 3 3 3
4 3 3 3 3 4 3 3 3 3 3 3 3 3 5 6 7 7 6 7 7 7 7 7 7 7 7 7 7 7 7 6 6 6 6 5 4 3 5 5
5 5 1 1 3 5 3 5 5 5 6 8 7 7 7 7 8 8 7 8 7 7 8 1 8 8 1 8 8 1 8 8 1 8 1 8 8 8 8 8
8 7 8 7 6 4 4 4 5 4 5 4 4 5 4 4 6 8 8 8 8 8 1 8 7 8 8 6 5 5 7 7 1 2 1 7 6 7 7 7
7 1 7 7 7 6 6 7 4 4 3 3 3 3 4 4 5 3 3 3 6 5 3 4 3 4 5 5 5 6 7 5 5 7 5 3 2 3 4 3
5 6 5 7 5 5 4 4 2 3 4 3 5 5 7 1 3 3 5 6 4 4 3 5 3 2 4 4 2 3 2 3 3 3 3 3 3 3 2 2
2 2 1 1 1 1 8 8 1 8 8 1 8 8 8 8 8 8 8 1 8 8 8 8 8 2 4 3 4 5 4 5 4 4 4 4 4 3 4 5
4 5 4 5 4 4 5 5 4 5 5 5 4 5 6 6 6 7 7 7 7 7 7 6 6 7 5 7 6 6 7 6 6 6 6

Number of smoothed octal codes = 317
Smoothed octal coded chain is as follows

2 1↓1 1 1 1 1 1 1 1↓1 2 2 2 3↓3 3 3 3 3 3 3 3 3↓3 2 8↓8 8 8 8 8 8 8↓8 1 1 1 2 3↓
3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3↓3 5
6 7↓7 7 7 7 7 7 7 7 7 7 7 7↓7 6 6 6 6 5↓5 5 5 5 5 5 5 5 5 5 5 5↓5 6 ↓8 7 7 7↓7 8↓
8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8↓8 7 6 4↓4 4 4 4 4 4 4 4 4 4↓4 6
8↓8 8 8 8 8 8 8 8↓8 6 5↓5 7↓7 7 7 7 7 7 7 7 7 7 7 7↓7 4 4 3↓3 3 3 3 3 3 3 3 3 3
↓3 4 5↓5 5 5 5↓5 3↓3↓3 5 6↓6 5 5 4 4 ↓2 3 3 5↓5 7 ↓1 3 3 ↓5 ↓6 4 4 4 4 3↓3 3 3 3 3 3
3 3 3↓3 2 2 2 2 1 1 1 1 8↓8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8↓8 2 4↓4 4 4 4 4 4 4
4 4 4 4 4 4 4 4 4 4 4↓4 5↓5 5 5 5↓5 6 6 6 7↓7 7 7 7 7 7 7↓7 6↓6 6 6 6 6 6

(a)

Description of the contour is as follows

$\bar{A}_{0.272}\ L_2\ A_{0.377}\ L_3\ \bar{A}_{0.644}\ L_2\ A_{0.488}\ L_{12}\ A_{0.757}\ L_4\ \bar{A}_{0.348}\ L_4\ A_{0.644}\ \bar{A}_{0.272}\ A_{0.272}\ L_8\ \bar{A}_{0.702}\ L_3\ A_{0.816}\ L_2\ \bar{A}_{0.644}\ A_{0.541}\ L_4\ \bar{A}_{0.765}\ L_3\ A_{0.465}\ L\ \bar{A}_{0.541}\ A_{0.645}\ \bar{A}_{0.587}\ A_{0.559}\ \bar{A}_{0.816}\ A_{0.707}\ A_{0.272}\ \bar{A}_{0.47}\ L_3\ \bar{A}_{0.429}\ L_5\ \bar{A}_{0.816}\ L_6\ A_{0.272}\ L\ A_{0.377}\ L_2\ \bar{A}_{0.272}\ L_2$

iii) Starting point of contour : (133,1)
End of contour is reached at (120, 128)
Number of octal codes = 70
Number of smoothed octal codes = 58
Description of the contour is as follows

$A_{0.272}\ L_2\ \bar{A}_{0.541}\ L\ A_{0.465}\ L\ \bar{A}_{0.272}\ A_{0.272}\ L\ \bar{A}_{0.272}\ L_3\ \bar{A}_{0.272}\ L_2\ \bar{A}_{0.272}\ A_{0.541}$

iv) Starting point of contour : (39,128)
End of contour is reached at (1,86)
Number of octal codes = 32
Number of smoothed octal codes = 27
Description of the contour is as follows

$A_{0.272}\ L_5\ A_{0.272}\ L$

v) Starting point of contour : (54,128)
End of contour is reached at (100,128)
Number of octal codes = 27
Number of smoothed octal codes = 27
Description of the contour is as follows

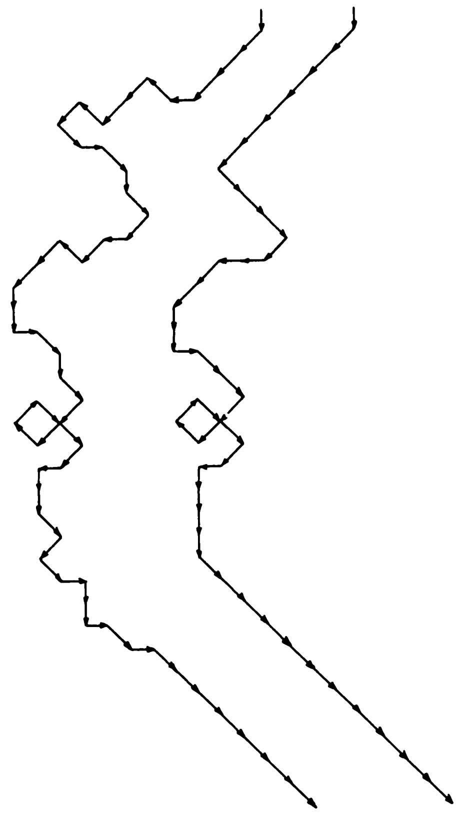
$\bar{A}_{0.272}\ L_2\ \bar{A}_{0.272}\ L\ \bar{A}_{0.272}\ L_3$

vi) Starting point of contour : (22,64)
The contour is closed
Number of octal codes = 132
Number of smoothed octal codes = 121
Description of the contour is as follows

$L_{11}\ A_{0.86}\ L_4\ A_{0.272}\ L\ \bar{A}_{0.662}\ L_4\ A_{0.598}\ L_7\ \bar{A}_{0.272}\ \bar{A}_{0.765}\ A_{0.816}\ A_{0.272}\ L\ A_{0.765}$

(b)

Fig. 7. Computer-based description of contours.

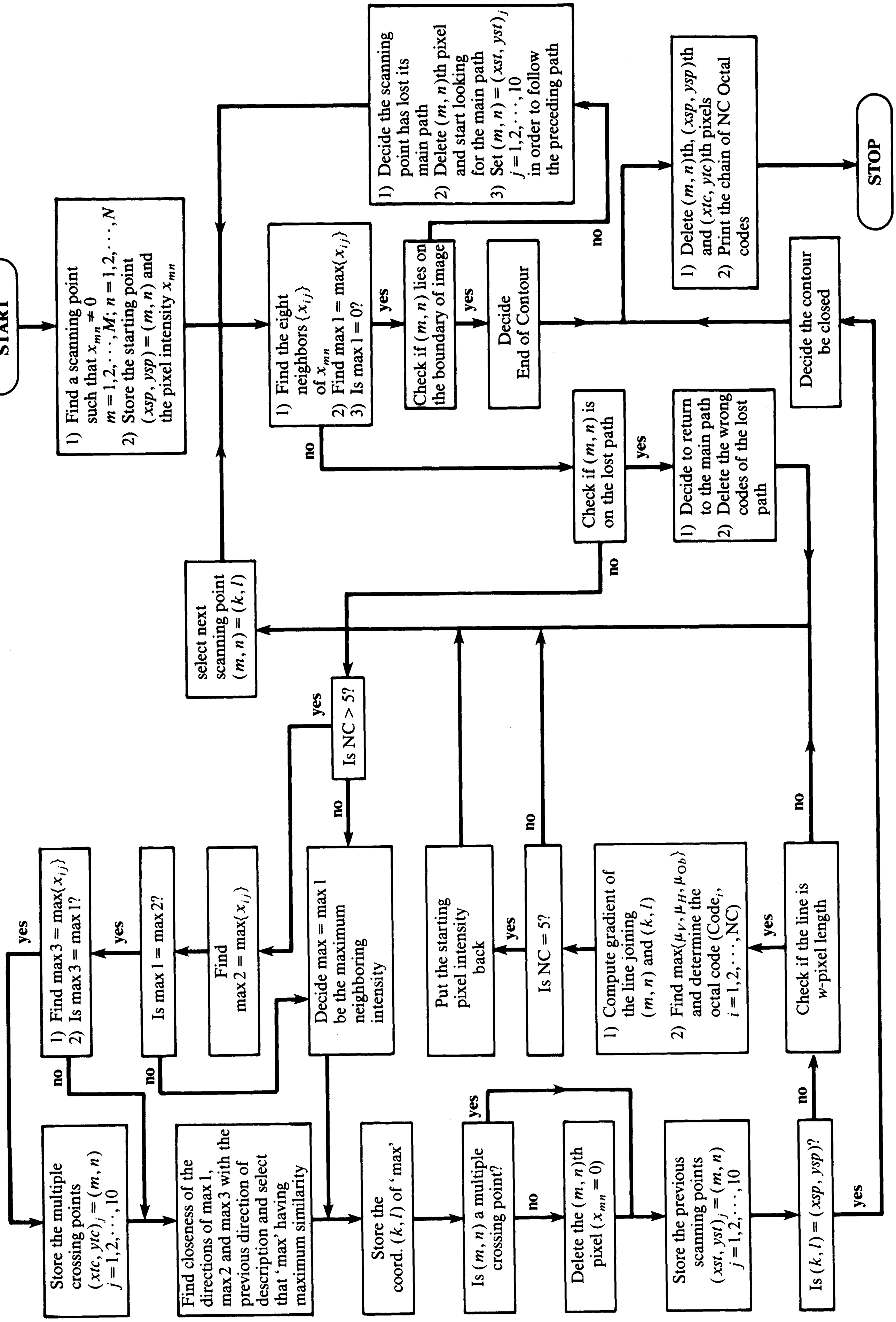


(a) (b)

Fig. 8. Pictorial representation of string. (a) Before smoothing. (b) After smoothing.

## APPENDIX I.

*Block Diagram for Encoding Images*



## ACKNOWLEDGMENT

Provision of the X-ray films by Professor J. M. Tanner is gratefully acknowledged by the authors.

## REFERENCES

[1] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis.* New York: Wiley, 1973.
[2] M. Minsky and S. Papert, *Perceptrons.* Cambridge, MA: MIT, 1969.
[3] J. K. Aggrawal, R. O. Duda, and A. Rosenfeld, Eds., *Computer Methods in Image Analysis.* New York: IEEE 1977.
[4] E. C. Greanias et al., "The recognition of handwritten numerals by contour analysis," *IBM J.* vol. 7, Jan. 1963.
[5] W. J. Hankley and J. T. Tou, "Automatic fingerprint interpretation and classification via contextual analysis and topological coding," in *Pictorial Pattern Recognition.* Washington, DC: Thompson Book Co., 1968.
[6] C. T. Zahn, "A formal description for two-dimensional patterns," in *Proc. Int. Joint Conf. on Artificial Intelligence*, pp. 621-628, May 1969.
[7] A. Rosenfeld and M. Thurston, "Edge and curve detection for visual scene analysis," *IEEE Trans. Comput.*, vol. C-20, pp. 562-569, May 1971.
[8] R. O. Duda and P. E. Hart, "Use of Hough transformation to detect lines and curves in pictures," *Commun. Ass. Comput. Mach.*, vol. 15, pp. 11-15, Jan. 1972.
[9] J. T. Tou and R. C. Gonzalez, "Automatic recognition of handwritten characters via feature extraction and multi-level decision," *Int. J. Comput. Sci.*, vol. 1, 1972.
[10] A. Martelli, "An application of heuristic search methods to edge detection," *Commun. Ass. Comput. Mach.*, vol. 19, pp. 73-83, Feb. 1976.
[11] P. H. Winston, *Artificial Intelligence.* Reading, MA: Addison-Wesley, 1977.
[12] K. Price and R. Reddy, "Matching segments of images," *IEEE Trans. Patt. Recog. Machine Intell.*, vol. PAMI-1, pp. 110-116, Jan. 1979.
[13] S. M. Rubin, "Natural scene recognition using locus search," *Comput. Graphics and Image Process.*, vol. 13, pp. 298-333, 1980.
[14] R. Nevatia and K. Ramesh Babu, "Linear feature extraction and description," *Comput. Graphics and Image Process.*, vol. 13, pp. 257-269, 1980.
[15] J. T. Tou, "An approach to understanding geometrical configurations by computer," *Int. J. Comput. and Inform. Sci.*, vol. 9, pp. 1-13, Feb. 1980.
[16] M. Brady, "Artificial intelligence approaches to image understanding," in *Proc. NATO ASI on Patt. Recog. Theory and Applications.*, Oxford, England, Apr. 1981.
[17] *Proc. IEEE*, special issue on image processing, vol. 69, May 1981.
[18] L. A. Zadeh, "Outline of a new approach to the analysis of complex systems and decision processes," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-3, pp. 28-44, Jan. 1973.
[19] A. Kaufmann, *Introduction to the Theory of Fuzzy Subsets—Fundamental Theoretical Elements*, vol. 1. New York: Academic, 1975.
[20] P. P. Wang and S. K. Chang, Eds., *Fuzzy Sets—Theory and Applications to Policy Analysis and Information Systems.* New York: Plenum, 1980.
[21] S. K. Pal, "Studies on the application of fuzzy set theoretic approach in some problems of pattern recognition and man-machine communication by voice," Ph.D. dissertation, Univ. Calcutta, Calcutta, India, 1978.
[22] ——, "Fuzzy set theory in grey tone image processing," Ph.D./D.I.C. dissertation, Univ. London (Imperial College of Sci. and Tech.) London, England, 1982.
[23] H. Freeman, "On the encoding of arbitrary geometrical configurations," *IRE Trans. Electron. Comput.*, vol. EC-10, pp. 260-268, June 1961.
[24] R. C. Gonzalez and P. Wintz, *Digital Image Processing.* Reading, MA: Addison-Wesley, 1977.
[25] A. Rosenfeld and A. C. Kak, *Digital Picture Processing.* New York: Academic, 1976.
[26] W. K. Pratt, *Digital Image Processing.* New York: John Wiley and Sons, 1978.
[27] S. K. Pal and R. A. King, "Application of fuzzy set theory in detecting X-ray edges," *Proc. IEEE ICASSP*, Apr. 1981, vol. 3, pp. 1125-1128.
[28] ——, "Histogram equalisation with $S$ and $\pi$ functions in detecting X-ray edges," *Electron. Lett.*, vol. 17, pp. 302-304, Apr. 16, 1981.
[29] D. Dutta Majumder and S. K. Pal, "On fuzzification, fuzzy language and multicategory fuzzy classifier," in *Proc. IEEE Seventh Int. Conf. on Cybern. and Soc.*, Washington, DC, pp. 591-595, Sep. 1977.

# The Absolutely Expedient Nonlinear Reinforcement Schemes Under the Unknown Multiteacher Environment

NORIO BABA, MEMBER, IEEE

***Abstract*—Learning behaviors of variable-structure stochastic automata under a multiteacher environment are considered. The concepts of absolute expediency and $\epsilon$-optimality in a single-teacher environment are extended by the introduction of an average weighted reward and are redefined for a multiteacher environment. As an extended form of the absolutely expedient learning algorithm, a general class of nonlinear learning algorithm, called the GAE scheme, is proposed as a reinforcement scheme in a multiteacher environment. It is shown that the GAE scheme is absolutely expedient and $\epsilon$-optimal in the general $n$-teacher environment. Learning behaviors of the GAE scheme in various multiteacher environments are simulated by computer and the results indicate the effectiveness of the GAE scheme.**

Manuscript received January 21, 1982; revised February 15, 1982, July 30, 1982, and August 19, 1982.
The author is with Information Science and Systems Engineering, Faculty of Engineering, Tokushima University, Josanjima, Tokushima City, 770, Japan.

## I. INTRODUCTION

The study of learning automata operating in an unknown random environment was started by Tsetlin [1]. He considered the learning behaviors of deterministic automata and showed that they are asymptotically optimal under some conditions. The learning behaviors of stochastic automata were investigated by Varshavskii and Vorontsova [2], and since then have been studied quite extensively by many researchers. Fu [10], Norman [7], [8], Flerov [13], Chandrasekaran and Shen [6], Shapiro and Narendra [9], Lakshmivarahan and Thathachar [15], etc., have contributed many fruitful results to the literature of learning automata. Survey papers written by Narendra and Thathachar [19] and Narendra and Lakshmivarahan [23] contain most of the recent work in this field along with the comments for future research.

However, almost all research so far has dealt with learning behaviors of a single automaton in a single-teacher environment. Recently Koditschek and Narendra [22] considered the learning behaviors of fixed-structure automata acting in a multiteacher environment. Thathachar and Bhakthavathsalam [24] then studied the learning behaviors of variable-structure stochastic automata operation in two distinct teacher environments. The behavior of a collective of interacting stochastic automata in a single-teacher environment was also considered by El-Fattah [26].

In this correspondence we consider the learning behaviors of variable-structure stochastic automata acting in the general $n$-teacher environment. In the first section brief explanations about the stochastic automaton acting in the general $n$-teacher environment are given. In the second section the new concept of an average weighted reward is introduced and several basic norms of the learning behaviors of stochastic automata in the general $n$–teacher environment are given. In the third section a class of nonlinear reinforcement scheme in the general $n$-teacher environment is proposed. It will be shown to be absolutely expedient and $\epsilon$-optimal in the general $n$-teacher environment. In the final section several numerical examples which indicate the effectiveness of our proposed learning algorithm are given.

## II. STATEMENT OF THE PROBLEM

The learning behaviors of stochastic automata have been extensively discussed under a single-teacher environment. Recently Koditschek and Narendra [22] introduced the concept of learning automata under a multiteacher environment. They considered the learning behaviors of fixed-structure automata under the multiteacher environment satisfying the condition (2). In this correspondence we will consider the learning behaviors of variable-structure stochastic automata under the $n$-teacher environment satisfying the more general condition (1).[1]

Let us briefly explain the learning mechanism of the stochastic automaton $A$ under the $n$-teacher environment (NTE) (Fig. 1).

The stochastic automaton $A$ is defined by the set $\langle S, Y, W, g, P(t), T\rangle$. $S$ denotes the set of $n$ inputs $(i_1, i_2, \cdots, i_n)$, where $i_j$ $(j = 1, \cdots, n)$ is the response from the $j$th teacher

[1]Koditschek and Narendra [22] considered the fixed-structure automata operating under the multiteacher environment in which there is an action $y_\gamma$ such that

$$C_i^\gamma < C_i^j \qquad \text{for all } i\ (1 \leqslant i \leqslant n), \text{ and all } j,\ (1 \leqslant j \leqslant r, j \neq \gamma). \qquad (2)$$

The condition (1) is more general than (2). Because it is assumed in (2) that all $n$-teachers agree that the $\gamma$th action, $y_\gamma$ is the best one. That is the reason why we call the environment satisfying (1) the general $n$-teacher environment.