Cursive English script recognition – A knowledge-based system

P. V. S. Rao and T. M. Ajitha

Script recognition by computers becomes extremely difficult due to variability at the input arising from a variety of reasons. Most existing recognizers are based on statistical pattern recognition techniques and require extensive training. This system uses domain knowledge to extract the broad shape features while filtering out the deformations and other variations in shape as 'noise'. This eliminates the need for any type of training. The performance of the system (even in a writer-independent task) is in the range of 95–100%, which compares very favourably with the current state of the art. This approach can be used for Indian language scripts too.

SCRIPT recognition – recognition of handwritten words and characters by computers – is receiving a lot of attention nowadays due to its practical applications in many areas: office automation, bank cheque processing, postal address and ZIP code recognition, signature verification, and, in general, document and text recognition. It is, in addition, a challenging area of research.

Script recognition can be accomplished using either of the following two approaches: recognizing words as integral units or visualizing each word as being composed of discrete characters. This distinction between word- and character-based recognition systems is particularly significant for cursive English script, where each word is not merely a string of discrete characters but a more or less continuous curve where character shapes may get modified to some extent. Recognition of words in terms of individual characters becomes difficult because decomposing a word into character length segments is a nontrivial problem. Segmentation into characters is not needed for a word level recognition system; on the other hand, the number of classes (each word will be a distinct class) increases very substantially in this case.

The different methodologies used for script recognition (character- or word-based) include:

- Point-by-point global comparison (comparison of all pixels in the image).
- Global transformations involving Fourier transformations, moment calculations and rotation according to the principal axis of inertia.
- Extraction of local properties such as lines, end points, T-junctions, corners and strokes.
- Analysis in terms of curvature which involves curve following, detection of concavities and geometrical analysis.
- Structural methods (decomposition of each character into its constituent elements, topological description and reduction of each character into a graph).

P. V. S. Rao and T. M. Ajitha are in the Computer Systems & Communications Group, Tata Institute of Fundamental Research, Homi Bhabha Road, Colaba, Bombay 400 005, India.

Overview

How can a computer-based recognition system 'learn', i.e. acquire knowledge about the characters and script of a language? How does it organize and utilize this acquired knowledge for actual recognition? The task of handwritten character and word recognition gets complicated due to variability arising from context effects, sloppy writing, subject-to-subject differences in writing style, etc. In fact, even the same person writes quite differently at different times. It is for such reasons that even the best trained optical readers – the human eyes – make some 4% mistakes when reading handwritten script in the absence of context information. According to Mantas[1]: 'Handwritten character recognition is not a simple task as it may appear. Errors in reading hand prints are caused by the infinite variations of shape resulting from the writing habit, mood, style, education, region of origin, health, social environment and other conditions of the writer, as well as other factors such as writing instrument, writing surface, scanning methods and finally, of course, the machine's character recognition algorithms.' The complexity of the task increases progressively as we go from fixed-font recognition of printed text (recognition of specific fonts) to recognition of single unconnected handwritten characters and finally to connected cursive (handwritten) scripts.

The underlying presupposition behind all recognition approaches is that there exist certain invariant properties which define each pattern in the task domain in a unique way and that these properties (parameters) can be determined in a quantitative sense.

In a general case of pattern recognition, each sample or pattern can be represented as a point in an n-dimensional parametric space where each dimension corresponds to one parameter (Figure 1). The set of parameters required for representing the individual patterns at the input would naturally depend on the complexity of the problem domain. Ideally, various instances of each class map into points which cluster into a compact neighbourhood in this space. Also, there would be no significant overlaps between regions defining the classes. Both these are necessary conditions for pattern recognition systems to perform well. These facilitate deriving a 'mean' representation for each input class, determined by the input instances (belonging to this class) encountered at the time of training.

For instance, in the case of the minimum-distance approach, the 'distance' between an input pattern and the 'mean' or 'class prototype' is a measure of the dissimilarity between them. The input pattern will be recognized as the class prototype 'nearest' to it. In the maximum-likelihood approach, the likelihood that a test sample may be a member of a cluster or class is determined on the basis of probabilistic distributions in the parametric space. The input pattern is put into the class with the maximum value for the probability of occurrence.

Many existing recognition systems use statistical pattern recognition techniques based on minimum distance, maximum likelihood, hidden Markov models[2], neural network models[3,4], etc. They are trained 'by exposure' to representative samples; the system tries to extract representative models for each of the script characters (or words) as encountered in real life. Understandably, these models would be distorted versions of the corresponding ideal character shapes.

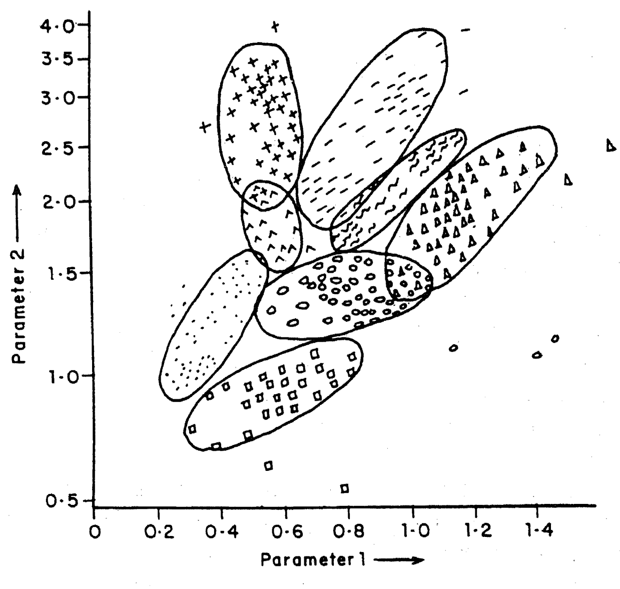

Figure 1. Clustering in parametric space.

Figure 2 shows some representative patterns 'learnt' by a neural network classifier proposed by Schomaker[4], for the characters */a/*, */b/*, */g/*, */k/*, */n/*, */o/*, */p/*, */r/*, */w/*, */x/*, */y/*, */z/*. It can be seen that the mean representation or class prototype is rather distorted and is quite different from the 'ideal shape' for each particular class (character). This is because while developing a 'statistical model' for the respective input classes, the classifier is, in fact, modelling not only the canonical shape behind it but also the variations (or 'noise') encountered at the input level.

In contrast to the statistical approaches mentioned above, our feature-based approach[5,6] is based on the fact that for each script character or word, there exists notionally an ideal pattern – the copy book version – which the writer is attempting to reproduce while writing. The variations or distortions that occur are viewed as added noise. Essentially, our system represents each individual character/word (input class) by the corresponding canonical shape (prototype). The task of the recognition system is thus to map a noisy real-life input (Figure 3 *a*) into the copy book counterpart (Figure 3 *e*). This is done in two steps.

Step 1: Filter out (local) shape-related distortions and retain only the broad (global) shape features of the input shapes by choosing an appropriate set of parameters which preserve only the very essential information regarding the shapes (Figure 3 *b*). It can be easily verified that there exists only one way of linking the feature points while preserving the order. The shape distortions in Figure 3 *a* have been eliminated in the reconstructed pattern (Figure 3 *c*).

Step 2: Filter out the distortions relating to size, shape and slant. Figure 3 *e* is such a reconstruction. (This is accomplished by quantization of parameter values as shown in Figure 3 *d* and is discussed in greater detail later.)

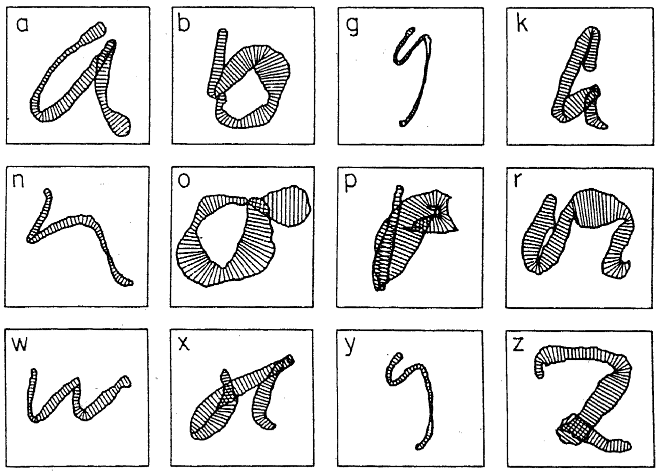

Figure 2. Abstract shapes derived by Kohonen net trained on samples written by a single subject (taken from ref. 4).

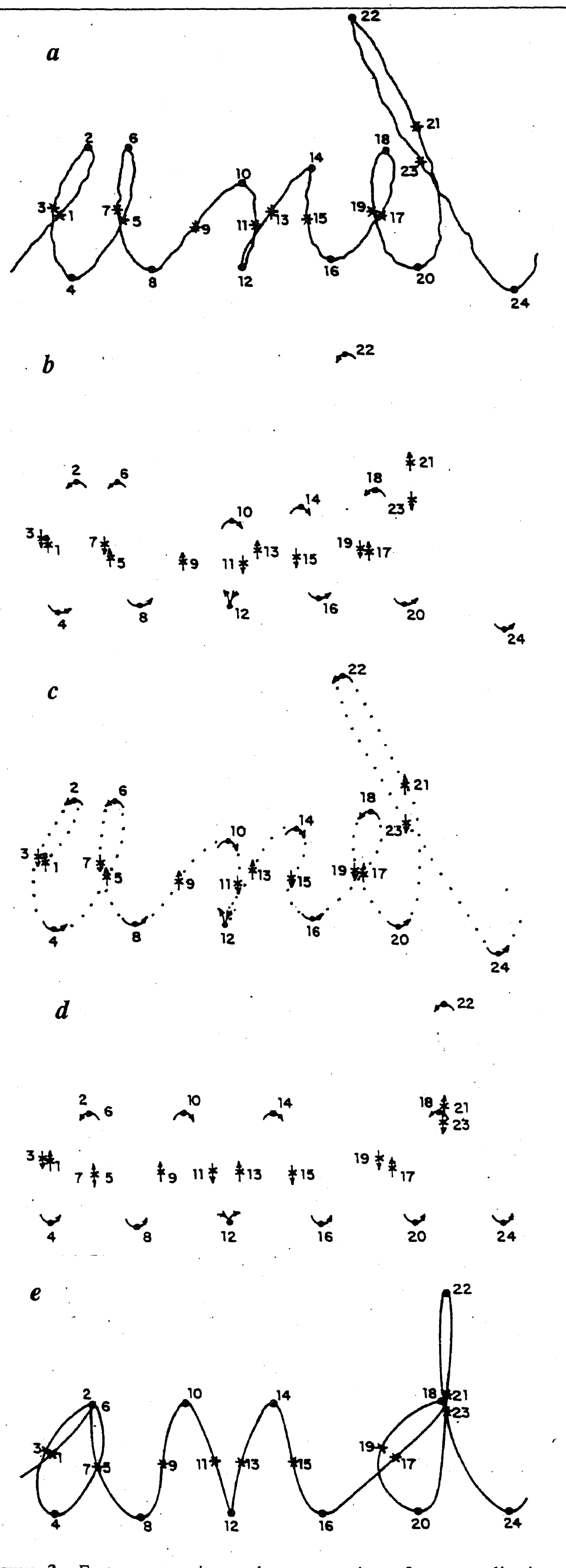

Figure 3. Feature extraction and reconstruction after normalization. *a*, Original pattern. *b*, Information captured by the feature set used. *c*, Reconstruction based on information in (*b*). *d*, Feature set after quantization. *e*, Reconstruction using quantized feature set.

Thus, this feature-based approach helps us to filter out the deformations, thereby ignoring the actual and detailed shapes of the character and to retain only broad shape features. Consequently, there is no need for analysing different samples corresponding to each individual character since each of them carries the same broad shape features. This makes training superfluous in this approach.

Selection of parameters

As mentioned earlier, recognition needs to be performed on the basis of the attributes of the patterns involved; these are represented as values of specific physical parameters of the signal. Selection of an appropriate minimum set of parameters is thus an important step for any script recognition system, and, in general, for all pattern recognition systems. For example, the selected set of parameters should not be sensitive to the style (and mood) of the writer (or to any of the other variations that occur in practice). It would be ideal if these parameter values remain invariant irrespective of contextual effects, variation in size, and placement of individual characters within a word. A character level recognition system should also be insensitive to all possible variations of individual character shapes when they are concatenated to form a word. This is a very important requirement for cursive script recognition in English.

In the current feature-based approach, the parameters selected permit the retention of broad shapes by capturing the archetypal characteristics of each individual character, and filtering out as noise all writer-specific, context-dependent and other distortions such as those relating to shape, size and slant. We accomplish this by heavy quantization of the parameter values (Figure 3).

Feature-based recognition system

In our parametric model, we use features concerned with points of minimum radius of curvature (minima) and maximum radius of curvature (maxima) – the feature points – for representing the characters. The feature vector representing an individual character consists of the points of minimum radius of curvature – the minima – of the character. The attributes we use are the location of the minima and the direction of curvature at these points.

Figure 4 shows the word /*andy*/ and the minima and maxima points; the minima bear even numbers and the maxima are the points with odd numbers.

The attributes of the minima and maxima are insensitive to local fluctuations in the shapes of the actual curves. A recognizer using these attributes would, there-

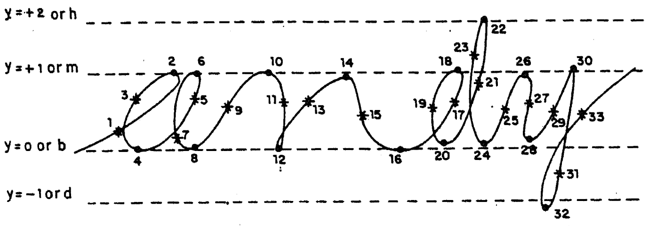

Figure 4. Feature points for the word /andy/.

fore, be reasonably robust. It is, in principle, possible to train a statistical recognizer to recognize script words using this information alone. Though insensitive to shape fluctuations, such a recognizer would still be sensitive to the fluctuations in slant, size and position. Such a recognizer would, therefore, need a large number of training samples. To obviate this, it would be desirable to 'standardize' the character size which manifests itself as horizontal and vertical distances between successive minima. This standardization can be conveniently done by warping the characters to fit into a uniform grid, i.e. by quantizing the X and Y coordinates of the minima points. This takes care of variation in character sizes. This has, in fact, been done in Figure 3 *d*. Similarly, quantizing the slopes of the maxima would eliminate the fluctuations in slant angle while writing.

Each minimum on the curve is characterized by position (X and Y coordinates) and curvature. The curvature is quantized to three values to carry the direction information: 'a' for anticlockwise, 'c' for clockwise and 's' for sharp turns (where the curvature could be either clockwise or anticlockwise). The Y coordinate value is quantized to four possible values: 'b' and 'm' (baseline and midline) for characters such as /c/ and /n/; 'h' and 'l' (highline and lowline) for characters with ascenders and descenders such as /h/ and /g/. X coordinate values are expressed as relative displacements as follows:

- for minima at the baseline and midline, with respect to the nearest left minimum at the same level.
- for minima at the highline and lowline, with respect to nearest left minimum at any level (because it is unlikely that there will be an adjacent left minima at the same level in these cases).

This helps us to distinguish between /d/ and the combinations such as /el/ or /il/. Figure 4 and Table 1 illustrate this scheme. Minima 2, 6 and 18 have the value 'a' (anticlockwise) for direction while minima 10, 14, 26 and 31 have the value 'c' (clockwise) for direction. Minima 12 and 30 have the value 's' (sharp) for direction. Minimum 12 has been traced in clockwise direction (as a loop). It could alternatively have been traced as a sharp turn as in the case of minimum 30. This would have made the curvature of minimum 12 anticlockwise; hence, the need for classifying it as

Table 1. Feature vectors for /a/ and /n/ in /andy/

		character a					character n				
Max.	Serial No.	1	3	5	7	9	9	11	13	15	17
	Slope	U	D	U	D	U	U	D	U	D	U
Min.	X	f	f	n	f		f	f	f	f	
	Y	m	b	m	b		m	m	b	b	
	Curvature	a	a	a	a		c	s	c	c	
	Serial No.	2	4	6	8		10	12	14	16	

'sharp' rather than as clockwise or anticlockwise. X has the value 'n' for minimum 6 because it is very near to minimum 2 at the same Y level and at its left, and 'f' for minimum 10 since it is far from its left neighbour. The X value helps to distinguish between /a/ and /u/, /g/ and /y/, /o/ and /v/, etc. The X coordinates are expressed not as absolute values but relative to each other, so that they remain unchanged even after concatenation of individual characters.

Maxima points on the curve are labelled by the slope angle quantized to the following values: up, down, right and left. The actual thresholds used for quantization are:

$0 \pm 20^\circ$:	right (R)
$180 \pm 20^\circ$:	left (L)
$90 \pm 70^\circ$:	up (U)
$270 \pm 70^\circ$:	down (D)

All the above parameters are extracted from single samples of individual letters. The sequence of values constitute the feature vector for the character.

The feature matrix for a word is obtained by concatenating the feature vectors for individual characters forming the word in the same sequence.

A lexicon consisting of feature matrices for all the words in the vocabulary is compiled from the feature vectors of individual characters. The test word written by the subject (which is to be recognized) is similarly analysed to yield the corresponding feature matrix. This is then compared with the feature matrices of all words in the lexicon. Recognition is accomplished on a best match basis, i.e. either the exact match or, failing this, the nearest match.

Details of implementation

Data acquisition and preprocessing

Both reference and test data are collected using a graphics tablet connected to the serial port of an IBM-PC-compatible system and feature vectors are extracted therefrom. Specimen samples for the 26 English lowercase letters are collected from a single subject. Feature vectors extracted from these letters constitute the reference data for the recognition system. The writer has to write the input pattern on the graphics tablet using the digitizer pen. The tablet digitizes the handwritten pattern and sends a stream of X and Y coordinates to the host computer through the serial port. These data are stored in the ASCII file format. To facilitate extraction of height information, it is necessary to ensure that the script is in a straight line and that the characters are of a reasonably standard size. To this end, the upper and lower reference lines are provided on the writing surface as in ruled copy books for children. The subject is required merely to write broadly within these lines. There is no need for anything like copy book precision.

Radius of curvature extrema – the feature points on the curve – can be located using one of several alternative methods. One of the easiest ways to locate minima (and maxima) is by using the fact that the stylus travels fast and almost in a straight line near the maxima but is slow and almost doubles back on itself as the curve turns nearly 180° around minima. The chord distance (not arc distance) between two sample points separated by constant time would, therefore, be minimum around minima and maximum in the neighbourhood of maxima. The relevant parameter values can be extracted for each minima and maxima and quantized easily.

Handwritten characters are usually not vertical in their orientation; they tilt to a greater or lesser extent, usually to the right. The tilt or slant remains more or less invariant for each person and is one of the features that characterize an individual's handwriting. It would seem very desirable to eliminate this slant so that we can deal with more standard – essentially vertical – characters. This can be done by means of a number of slant estimation and correction processes. It turns out that, except for extreme slants, this is not really essential, since our method of parameter representation and quantization for script characters is robust enough to tolerate even major slants in the writing.

Comparison and recognition

For recognizing the given input word, we narrow down the list of prospective candidates (dictionary words or reference patterns) on the basis of a rough match and perform a detailed matching only within this short list. The number of minima in a word is used as the criterion for this short-listing. All dictionary words where the mismatch in the number of minima exceeds 3 are rejected. If an exact match is found within the short list, recognition is successful. Failing this, the best match is identified from the short-listed set of words. We use a comparison procedure based on dynamic programming which permits efficient matching of feature vectors even in the presence of spurious minima and absence of genuine minima in the test pattern. The input word is recognized as the word in the short list which matches the test pattern best. Recognition fails if either the short list is empty or if, for all words in the short list, the mismatch during detailed matching turns out to be excessive.

Experimental results

As stated before, each character is represented by a feature vector, a sequence of labelled minima and maxima points. Each minimum is characterized by (a) location (quantized) and (b) direction of movement (clockwise, anticlockwise or sharp turn) while each maxima is labelled with the direction of movement. Recognition is successful if an exact match is found within the short list. Failing this, the best match is selected as the candidate.

Data sets were collected from 10 (male and female) subjects from different age groups and educational backgrounds. Each subject was asked to write the 63 words of the vocabulary (word length varied between 3 and 7 characters) twice on the graphics tablet. The vocabulary included words such as 'fill' and 'fell' which are very smilar to each other. The reference character set (26 lowercase alphabets) was taken from a different subject. Standard lexical entries for each word in the vocabulary were generated from these by appropriately concatenating the feature vectors of these individual letters. Recognition results (using this 'common' set of lexical entries as standard and the word sets written by each subject at the test set) are summarized in Table 2.

Each subject was also asked to write ten sets of the word /and/. These were pooled together and used in a separate recognition experiment. In this case the overall recognition score was 93%. Figure 5 includes some of the samples which were recognized correctly despite the significant shape deviations in many of them. This demonstrates the robustness of the approach followed.

Table 2. System performance

	Recognition scores (%)	
Subjects	Word set 1	Word set 2
Subject 1	95	97
Subject 2	97	97
Subject 3	98	98
Subject 4	97	97
Subject 5	97	98
Subject 6	97	100
Subject 7	100	100
Subject 8	100	97
Subject 9	100	100
Subject 10	100	100

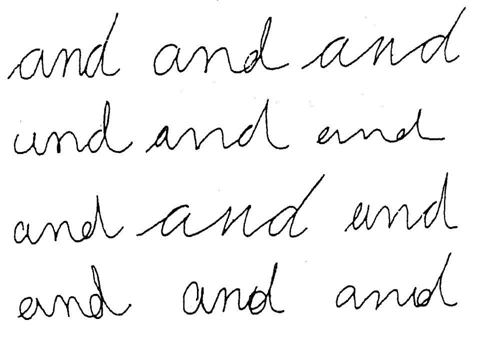

Figure 5. Samples of the word /and/.

Discussion and conclusions

In this feature-based approach for script recognition, we introduce an appropriate empirical parametric model for cursive English script. The minimal set of parameters used permit the retention of the shapes of the script characters, while ignoring the context-dependent, writer-specific and other distortions in shape, size and slant of characters. Resynthesis based on extracted parameters was successful in recreating the shapes of connected words and individual characters. This demonstrates the validity of our approach.

This is a knowledge-based system; knowledge about the task domain is contained implicitly in the set of parameters chosen for representation and the manner in which they are processed. Needless to say, any recognition system needs knowledge regarding the task domain at some stage before it can start 'recognizing' the input. Statistical systems start without any *a priori* knowledge. Consequently, such classifiers acquire this knowledge through elaborate training. In the case of hidden Markov models and neural networks, the topology chosen implicitly incorporates some knowledge about the task domain into the system. This, however, is not enough to dispense with the need for training. In the current feature-based approach, we use knowledge about the task domain to choose an appropriate parametric model which extracts and retains only the canonical or prototype shape information for specimen words in the lexicon on the one hand and for the test words on the other hand; in both cases, the model filters out real-life shape deformations as 'noise'. This obviates the need for training of the type needed for statistical or neural-network-based approaches. Only an 'initialization' with one sample for each of the 26 lowercase script characters is required prior to recognition. The performance of the system does not degrade significantly even when the initialization and test samples are written by different subjects. The recognition scores are in the range of 95–100%. This demonstrates the robustness of the approach.

The present system is able to handle vocabulary sizes of the order of a few hundreds of words. Though the parameter set selected here is ideally suited for the Roman script, the underlying principle is general enough to adopt for script recognition in other languages, say Indian languages. The set of features or parameters used for representation might, of course, vary across languages. We are developing feature-based recognition systems for some Indian languages and preliminary results are encouraging. Machine recognition of Indian language scripts becomes a particularly attractive avenue for research due to the complexity of the task of adapting Indian scripts to the qwerty keyboard.

1. Mantas, J., *Pattern Recogn.*, 1986, **19**, 425–430.
2. Kundu, A., He, Y. and Bahl, P., *Pattern Recogn.*, 1989, **22**, 283–297.
3. Rao, P. V. S., Proceedings of International Conference on Automation, Robotics and Computer Vision (ICARCV '90), Sept. 18–21, Singapore, 1990, pp. 1237–1241.
4. Schomaker, L., *Pattern Recogn.*, 1993, **26**, 443–450.
5. Rao, P. V. S., Proceedings of IAPR Workshop on Machine Vision Applications, Nov. 28–30, Tokyo, 1990.
6. Rao, P. V. S., *Sadhana*, 1994, **19**, 257–270.

ACKNOWLEDGEMENTS. We thank Mr Thomas C. Jose, Mr Yashowardhan R. Sowale and other student trainees for their help at various stages of experimentation and to other colleagues who helped in data collection. We are also grateful to Mr R. S. Thakur for his neat drawings and timely help.