PERGAMON

Pattern Recognition 32 (1999) 217–233

PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY

Off-line signature verification using genetically optimized weighted features

V.E. Ramesh, M. Narasimha Murty*

Department of Computer Science and Automation, Indian Institute of Science, Bangalore 560 012, India

Received 6 April 1998; received in revised form 14 September 1998; accepted 14 September 1998

Abstract

This paper is concerned with off-line signature verification. Four different types of pattern representation schemes have been implemented, viz., geometric features, moment-based representations, envelope characteristics and tree-structured Wavelet features. The individual feature components in a representation are weighed by their pattern characterization capability using Genetic Algorithms. The conclusions of the four subsystems (each depending on a representation scheme) are combined to form a final decision on the validity of signature. Threshold-based classifiers (including the traditional confidence-interval classifier), neighbourhood classifiers and their combinations were studied. Benefits of using forged signatures for training purposes have been assessed. Experimental results show that combination of the feature-based classifiers increases verification accuracy. © 1999 Pattern Recognition Society. Published by Elsevier Science Ltd. All rights reserved.

Keywords: Off-line signature verification; Genetic algorithms; Tree-structured wavelets; Threshold-based classifiers; Neighbourhood classifiers; Hybrid classifier; Combination of classifiers

1. Introduction

Human handwritings are among the most complicated objects to recognize. They comprise some of the areas where humans possess monopoly and computers have treaded little on. Signatures form a special class of handwriting in which legible letters or words may not be exhibited. They provide secure means for authentication, attestation and authorization in legal, banking or other high security environments. Signature verification problem pertains to determining where a particular signature is verily written by a person so that forgeries can be detected. Based on the hardware front-end, a signature verification system can be classified as either online or offline. On-line system [1] employs an electronic pen and pad and a host of dynamic information like speed of writing, pressure applied, number of strokes, etc., can be extracted. In off-line signature verification system, signatures written on paper as has been done traditionally will suffice. The signatures are converted to electronic form with the help of scanner or camera. Financial constraints dictate most of the applications requiring signature analysis not to be equipped with hardware necessary for on-line technique. Moreover, it is not feasible to meet the needs of a big mass of people with such finite systems and authorization is rendered impracticable. So off-line technique appears to be more pragmatic. However, signature analysis using off-line technique is relatively more difficult as only static information is available.

*Corresponding author. Tel.: 0091 80 309 2779; Fax: 0091 80 3341683; E-mail: mnm@csa.iisc.ernet.in

PII: S0031-3203(98)00141-1

The static information derived in an off-line signature verification system can be either at a gross or detailed scale. Features that grossly characterize signature can be quite helpful to detect skilled forgeries, yet treating the genuine signatures of a person at par inspite of intra-personal variations. Features that capture details in signatures are capable of detecting simple and decent forgeries at the cost of the power to be insensitive to intra-personal variations in genuine signatures of the same person.

Static features may be global, structural, geometric or statistical. Use of transform-based representations [2] and critical points from off-line tracing of signatures [3] have been reported. Combination of global geometric and grid features has been attempted [4]. A connectionist scheme [5] of combining classifiers based on moment measures and envelope characteristics has been proposed by Bajaj et al. Geometric feature extraction with neural network classification has been used [6].

In this paper, we present an off-line signature verification scheme based on four different types of pattern representations, viz., geometric features, moment-based representations, envelope characteristics and tree-structured wavelet features. The individual feature components in each of the first three representations are weighed by their pattern characterization capability using Genetic Algorithms. The objective function of the Genetic Algorithm (GA) module depends only on the training samples obviating the need for validation patterns. Several threshold-based verifiers, neighbourhood classifiers and their combinations have been studied in this work. The neighbourhood classifier requires both genuine and forgery samples for training. The conclusions of the four subsystems (each depending on a pattern representation scheme) are combined to form a final decision on the validity of signature.

Section 2 describes the size of the database of signatures collected and the preprocessing work done for any signature image before it makes its way into the system. In Section 3, the different ways in which signature patterns are represented in this work are discussed. Each representation scheme is described in detail until the stage of obtaining the feature vector. Section 4 throws light on the distance measures used, GA implementation to determine the optimal weightages for the feature components and the several verification techniques employed in this work. In Section 5, system implementation and experimental results are presented. Finally, conclusions of the study are noted in Section 6.

2. Data preparation

2.1. Database

Most of the literature that dealt with signature verification systems had reported the use of only genuine signatures in the training patterns. All the forgery patterns collected were used only for testing purposes. They made a reasonable assumption that the genuine signatures fall within some boundary and anything out of the boundary is a forgery. Theoretically, if we can collect all possible forgeries for a given signature, then we can use the forgery patterns also for training and the verification process boils down to a two-class classification problem. However, it is infeasible to collect all types of forgeries of a given signature exhaustively. In our work, some of the forgery patterns have been used for training purposes also. This idea is based on the fact that if we know certainly some pattern to be a forgery, then any other pattern close to it is also a forgery. Our scheme is likely to catch some of the forgeries and hence obviates the need to decide based on the imprecise boundary threshold for such forgery cases.

The database collected comprised about 650 signatures. Genuine signatures were collected from 15 people. A total of 20 signatures were collected from each person over a period of one month to accommodate intra-personal variations. Fifteen of these signatures were used for system training and the remaining five for system testing. Five people were asked to produce a set of forgeries. Twenty-three forged signatures for each of the 15 persons were collected. Thirteen of these were used for system training and the remaining ten for system testing. Number of test forgeries is more than that of test genuines to give room for the variety of forgeries likely to occur in real life. The forgery patterns included simple, decent and skilled forgeries. Simple forgeries were made by asking the forgers to produce forgery given only the printed name of the persons. Decent forgeries were prepared by giving the forgers a glance of the genuine signatures for a minute or so. In order to produce skilled forgeries, the forgers were given a copy of the genuine signatures and were allowed to practice as many times as they wished before writing the forgeries. A sample set of genuine signatures and their simple, decent and skilled forgeries is shown in Fig. 1a, b, c and d, respectively. The size of this database is comparable to that of the data sets reported in the literature.

Ten non-overlapping signatures were obtained on every page. The signatures were scanned into the computer using an 8-bit scanner at 72 dpi resolution. The scanned signatures were separated out as such from the scan page using an image editor. The signatures obtained were all grey-level images. No attention was paid to remove dirt or any other mild stain on the papers to be scanned. Signatures were obtained intentionally on both sides of the paper, so that the mark of the signatures on one side of the paper will be visible on the other side as shown by Fig. 1. Such noisy signature patterns reflect the real-life conditions. For example, the cheques in banks have background colour, which should be removed while processing the signature images.

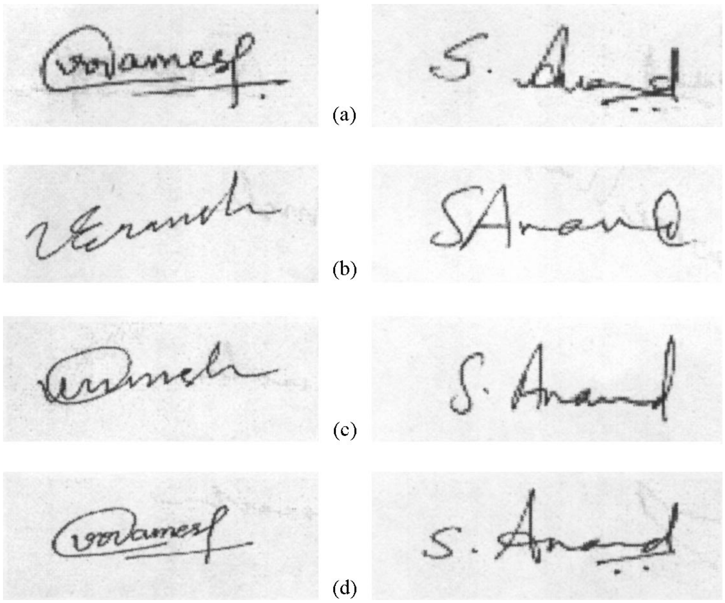

Fig. 1. Sample signatures.

2.2. Preprocessing of signatures

The gray-level signature images were linearly normalized with respect to the highest gray value in the image. This gives a uniform representation of the images even if signatures are written with different pens. The gray level images were converted to binary images by *thresholding* with the help of the Laplacian operator as illustrated in [7]. Conversion of a raw signature image to binary image should not only identify the signature portions in the image, but also eliminate the non-signature portions in the image like white space, background color, any other stain, etc.

The angle of inclination (or orientation with respect to paper) may, at first, be thought of as a good characteristic feature of a person's signature. However, even for a straight and horizontal signature, people cannot be asked to align the page perfectly with their signature's direction. Hence, angle of inclination of a signature cannot be taken as a characteristic feature. We should, instead, make the signatures invariant of the angle of inclination. This forces all the signatures to be aligned in the horizontal direction for enabling good comparison. Signature image is considered as a bivariate distribution of points and the angle of the eigenvector corresponding to the maximum eigenvalue of the covariance matrix of the distribution is an indicator of the angle of inclination. This method was employed for critical point normalization in [3]. In the present work, it is found to estimate the angle of inclination of a binary signature image with a correction requirement of 2.5°.

In order to achieve translation invariance, all the blank edges around signature were removed by sequentially eliminating rows (and columns) on the edge of the signature if the total number of pixels in a row (or column) was less than 2 (an arbitrary limit). High signal-to-noise ratio in the thresholded binary images made the above operation reliable.

Size-normalization of signature images is left to the individual feature extraction stage. This is because some of the features' values can be normalized with respect to some other measure (e.g. area of black pixels can be normalized with respect to the area of the image). Such features do not require the image size to be truncated/expanded to a standard size. Wherever it is difficult for the features to be normalized, the image is resized suitably. Resizing an image, like any other geometric operation, requires the points in the image to be mapped to a new set of points. After the mapping is done, the new image is left with two types of spurious points: (1) points which have not been mapped into by any of the points in the original image and (2) points which have been mapped into by more than one point in the original image. These points can be dealt with by some averaging or interpolation methods. Yet, distortion of the image is inevitable. Too much of averaging may make the image too smooth and hence lose sharp edges. Too little averaging may give rise to some "holes" in the transformed image due to the points of the first type. Though the distortion may not appear significant for images of photographs or shaded diagrams, it will be very explicit in images of handwriting and signatures where there are innumerable delicate strokes. The following two–phase algorithm has been proposed to minimize the amount of these "holes" without any requirement of averaging.

1. *For each row i of the original image do*:
2. *For every line l* of **black** *pixels in the original image in the row, let the columns corresponding to the end points of the line be s_{il} and e_{il}.*
3. *Let the transformed columns according the desired width of the new image be S_{il} and E_{il}. Set all the pixels between columns S_{il} and E_{il} in row i of the transformed image as* **black**.
4. *If no more lines in row i, goto step* 5. *Else, goto step* 2.
5. *If $i \leqslant M$ (where M is the original height), goto step* 1.
6. *The above operation changes the width of the image. Repeat it for every column of the resultant image for a change in height.*

The final image may still have some breaks between the lines formed. However, the shape of the signature will be more faithfully retained than by applying blind resizing of the image. Fig. 2a shows a signature image. The resizing results of two popular interpolation algorithms — nearest neighbour and bicubic, are shown in Fig. 2b and c, respectively. The former has many gaps and the latter appears patchy due to oversmoothing. Fig. 2d shows the result of the proposed algorithm. It can be found to be certainly better than other resizing methods.

3. Pattern representation

The following features comprise the entire feature set used in the present work to represent a signature:

- *Global geometric features*: Geometric characteristics of a signature such as size and slant angle.
- *Moment-based features*: Certain measures like kurtosis and skewness based on the vertical and horizontal projection images.
  Envelope features: Relative positions of the extremal points on the envelopes of a signature.
  Wavelet Features: The energy maps obtained from a Tree-structured Wavelet Transform of a signature.

3.1. Global geometric features

The global geometric features of a signature image included:

- *Aspect-ratio of a signature image.* Height normalized with respect to width after removing blank edges. This can be very helpful in detecting skilled forgeries.
- *Width without blanks.* Width with blank columns within signature removed.
- *Slant angle of the signature*
- *Number of short vertical connections.* Number of vertical lines of width, say, 3 pixels.
- *Vertical center of gravity* (COG).
- *Maximum horizontal projection.*

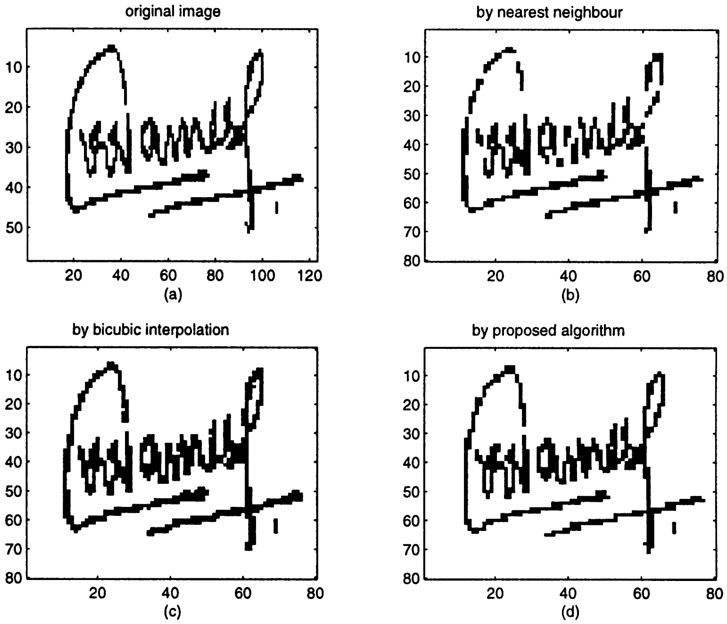

Fig. 2. Image resizing methods.

- *Area of black pixels.*
- *Baseline shift.* Shift in vertical COGs between left and right halves of the image.

Implementation details of the above geometric features can be found in [4].

3.2. Moment-based representations

Consideration of different types of global features can increase reliability of the identification decisions. Hence, attempt has been made to extract features from projection moments. “Projection moments [8] provide a statistical measure of the distribution of the signature pixels and unlike other global features, is relatively insensitive to distortions and style variations” [5]. Horizontal and vertical projection images of the signatures can be defined in the following way:

1. Horizontal projection image:

$$X_i = \sum_j im(i,j).$$

2. Vertical projection image:

$$Y_j = \sum_i im(i,j),$$

where $im(i,j)$ is either 1 or 0 and i is the row index and j refers to the columns. An rth-order moment measure μ_r for the projection image is defined as

$$\mu_r = \sum_i (x_i - x^c)^r G(x_i)$$

where x^c is the centroid of the corresponding projection image and $G(.)$ can be either $X()$ or $Y()$. In the present work, following moment measures, as suggested in [5] for signatures, have been used (V indicates moments computed from the vertical image, and H from the horizontal image).

1. Kurtosis measures:

(a) $K_V = \dfrac{\mu_4^V}{(\mu_2^V)^2}$,

(b) $K_H = \dfrac{\mu_4^H}{(\mu_2^H)^2}$.

2. Skewness measures:

(a) $S_V = \dfrac{\mu_3^V}{(\mu_2^V)^{1.5}}$,

(b) $S_H = \dfrac{\mu_3^H}{(\mu_2^H)^{1.5}}$.

3. Relative kurtosis and skewness measures:

(a) $R_V = \dfrac{\mu_3^V}{(\mu_4^V)^{0.75}}$,

(b) $R_H = \dfrac{\mu_3^H}{(\mu_4^H)^{0.75}}$.

4. Relative vertical and horizontal projection measure

$$VH = \frac{\mu_2^V}{\mu_2^H}.$$

Kurtosis is a measure of flatness and skewness is a measure of asymmetry of distribution. It was observed in [5] that if a signature is not properly aligned, erroneous measures will be obtained, because projection images will be different. But alignment problem has been taken care of by the method outlined in Section 2.2.

3.3. Envelope characteristics

Envelope characteristics were developed by Bajaj and Chaudhury [5] for signatures. The curve connecting the external points lying on the upper side of the principal axis of signature was referred to as the *upper envelope* and that on the lower side was called *lower envelope*. By focusing on one side of the principal axis, some of the external points may be lost. For example, the upper envelope, defined as above, will not include the external points on the lower side of the principal axis. So, envelopes in the present work, were allowed to take points from both sides of principal axis. The locus of the points corresponding to the first black pixel in each column when traversing from top is referred to as the *upper envelope* and that when traversing from bottom of the signature image is called the *lower envelope*.

3.3.1. Grid features

For extracting the shape information from these curves a suitably sized grid is to be overlaid on the corresponding envelopes (see Fig. 3). Several sizes of the grid have been experimented on. Finally, a grid of size 3 by 2 was arrived at. This grid was found to capture best the local shape properties of the signature, yet being insensitive to the variations due to noise. The nature of the portion of envelope caught within a grid element determines the numerical value assigned to it:

- 0: if envelope does not pass through the region, or if envelope passes through the region and any of significant peak or valley, upslope or downslope is not present.
- 1: if there is a prominent upslope only lying in the region.
- 2: if there is a prominent downslope only lying in the region.

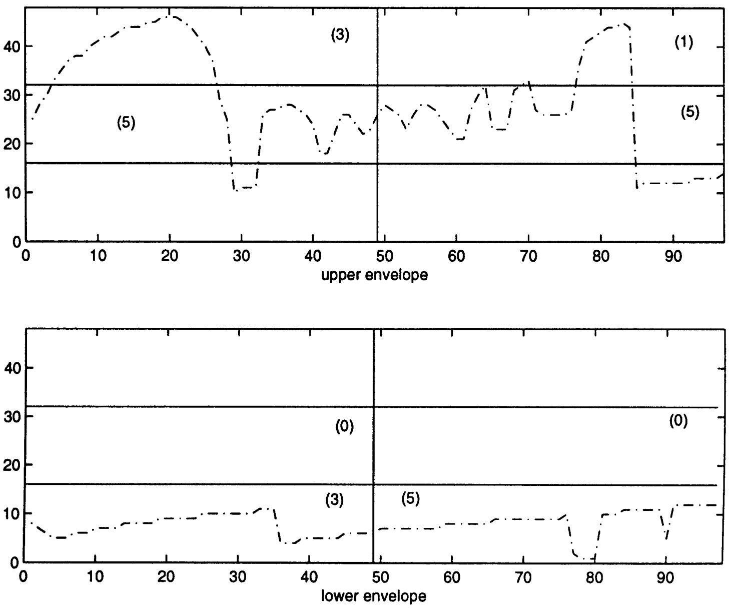

Fig. 3. Envelopes for signature in Fig. 2a.

- 3: if there is a prominent peak (and hence upslope and downslope) only lying in the region.
- 4: if there is a prominent valley (and hence upslope and downslope) only lying in the region.
- 5: if both prominent upslope and prominent downslope, but no peak/valley (or) if both prominent peak and prominent valley (and hence upslope and downslope) are lying in the region.

Upslopes and downslopes were not considered in [5]. A valley, peak, upslope or downslope is deemed prominent if its height exceeds 25% of the height of the grid cell. We found that all grid elements were not useful for characteristic feature extraction. In the case of lower envelope, the top 2 grid elements were found to be assigned the same values for all signatures. So, they were eliminated in the construction of feature-vector. Similarly, in the case of upper envelope, the bottom 2 grid elements were ignored. Thus, we obtain two four-dimensional feature vectors, one for each type of envelope. The numbers within parenthesis indicated in Fig. 3 on each of the grid elements denote the feature values assigned to the respective grid cell. For example, the entry corresponding to the first grid cell in the first row of the upper envelope is 3, because there is a significant peak in the region. Again, the entry corresponding to the second grid cell in the same row is 1, as there is only a prominent upslope. Note that there is no gradual downslope for the curve to be taken as a peak, but only a sudden fall on the right side. The entries in all the cells in the second row are 5 as there are both peaks and valleys. It is clear from the given example that the shape information of the envelope curve is adequately captured in the encoded feature vector.

3.3.2. Degree of convolutions

The degree to which the envelopes are convoluted and connected is well exhibited by the following features:

- Number of turns in the upper envelope.
- Number of turns in the lower envelope.
- Number of gaps in the envelopes. This will be the same for both the envelopes.

These features are quite capable of detecting many types of simple forgeries, and to some extent, decent forgeries. One may be tempted to consider the left and right envelopes of signatures. But, there will not be considerable shape variations on the left and right sides of typical signatures as there are on the upper and lower sides.

3.4. Wavelet-based representations

3.4.1. Wavelet transform

Wavelet transform is a way of decomposition of a signal $f(x)$ into wavelet coefficients computed as

$$c_{m,n} = \int_{-\infty}^{+\infty} f(x)\psi_{m,n}(x)\,\mathrm{d}x,$$

where $\psi_{m,n}(x)$ is a family of real orthonormal bases obtained through dilation (of m) and translation (by n) of a kernel function $\psi(x)$ known as the *mother wavelet*, i.e.

$$\psi_{m,n}(x) = 2^{-m/2}\psi(2^{-m}x - n), \tag{1}$$

where m and n are integers. The original signal can be reconstructed from its wavelet coefficients by the synthesis formula

$$f(x) = \sum_{m,n} c_{m,n}\psi_{m,n}(x)$$

To construct the mother wavelet $\psi(x)$, we may first determine a scaling function $\phi(x)$, which satisfies the two-scale difference equation

$$\phi(x) = \sqrt{2}\sum_k h(k)\phi(2x - k). \tag{2}$$

Then, the wavelet kernel $\psi(x)$ is related to the scaling function as

$$\psi(x) = \sqrt{2}\sum_k g(k)\phi(2x - k),$$

where

$$g(k) = (-1)^k h(1 - k).$$

We do not require the explicit forms of $\phi(x)$ and $\psi(x)$, but only the coefficients $h(x)$ and $g(x)$ to perform the wavelet transform.

Any discrete function $f(x)$ can be decomposed into two functions called an *averaged function* and a *detail function*. We achieve compression of information by considering only the *averaged function* and ignoring the *detail function*. The *averaged function* can be further decomposed to two such functions and so on. The decomposition at the jth level is given by

$$c_{j+1,n} = \sum_k c_{j,k}h(k - 2n), \tag{3}$$

$$d_{j+1,n} = \sum_k c_{j,k}g(k - 2n). \tag{4}$$

It is often convenient to view the decomposition (3), (4) as passing a signal $c_{j,k}$ through a pair of filters H and G with impulse responses $h(-n)$ and $g(-n)$ and down-sampling the filtered signal by two (dropping every alternate sample). The pair of filters H and G correspond to the lowpass and highpass filters, respectively, and are called *quadrature mirror filters* in the signal processing literature. Since signature image is a two-dimensional signal, we have to perform filtering and down-sampling along rows and then along columns for one level of decomposition. Fig. 4 shows how this decomposition is done. LL and LH are derived from the averaged image and HL and HH from the detail image.

3.4.2. Tree-structured wavelet transform

The traditional pyramid-type wavelet transform recursively decomposes subsignals in the low frequency

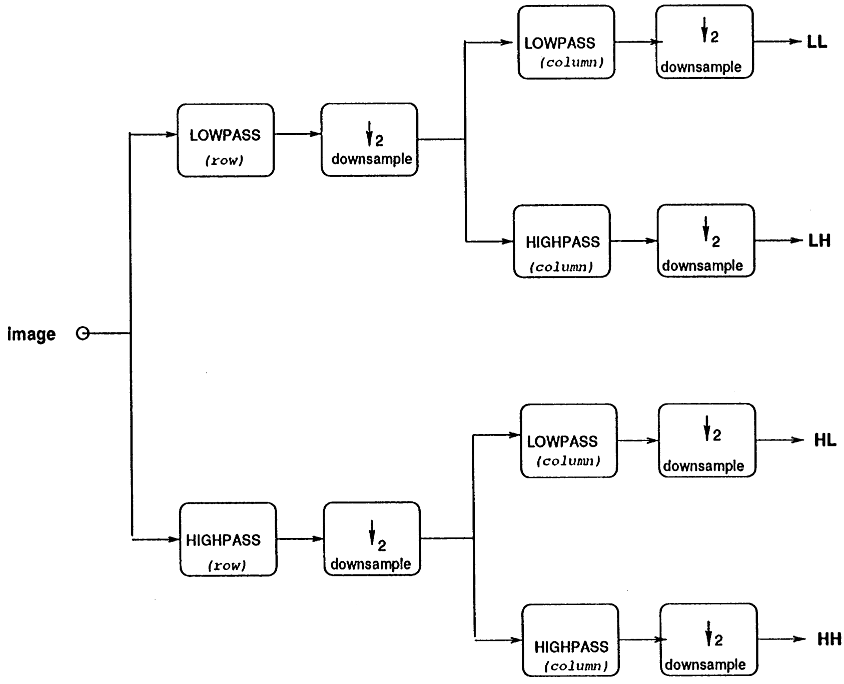

Fig. 4. One level of wavelet decomposition.

channels. However, since a signature consists of many irregular stokes in different directions, there will be some information in the high-frequency channels also. Thus, an appropriate way to perform the wavelet transform for signatures is to detect the significant frequency channels and then to decompose them further. The above idea leads to a tree-structured wavelet transform. The algorithm for performing a tree-structured wavelet transform is given below [9]:

1. *Decompose a given signature image with 2-D wavelet transform (as in Fig. 4) into four subimages, which can be viewed as the parent and children nodes in a tree.*
2. Calculate the energy of each decomposed image (child node). That is, if the decomposed image is $x(m, n)$, *with* $1 \leqslant m \leqslant M$ *and* $1 \leqslant n \leqslant N$, *the energy is is defined as*

$$e = \frac{1}{MN} \sum_{m=1}^{M} \sum_{n=1}^{N} |x(m, n)|.$$

3. *If the energy of a subimage is significantly smaller than others, we stop the decomposition in this region since it contains less information. This step can be achieved by comparing the energy with the largest energy value in the same scale. That is, if* $e < Ce_{max}$, *stop decomposing this region where* C *is a constant less than* 1.
4. *If the energy of a subimage is significantly larger, we apply the above decomposition procedure to the subimage.*

With respect to each tree-structured wavelet transform, we calculate the energy at its leaves, and obtain an energy function defined on the spatial/frequency domain called the *energy map*, which will be used for verification. Fig. 5 shows a sample signature and its energy maps with $C = 0.01$. A tree branch without an arrow head implies that there cannot be any more decomposition. Each leaf is associated with an energy map comprising of the path to the leaf and the value of the energy function at the leaf. One such path is shown darkened in the Fig. 5. Similar signatures tend to have similar sets of prominent energy paths. We can take p strongest energy maps for each signature and do the comparison. Table 1 shows the energy maps with energy values for $p = 10$ with a maximum of three levels of decomposition.

4. Verification

A typical representation of a pattern consists of a set

$$\mathbf{f} = (f_1, f_2, \ldots, f_n) \tag{5}$$

of real numbers, i.e. $f_i \in \Re$ called feature vector. A pattern representation scheme is meaningful only if more characteristic features are given greater importance. We can define a weighted feature vector

$$\mathbf{f}' = \mathbf{w} * \mathbf{f}$$

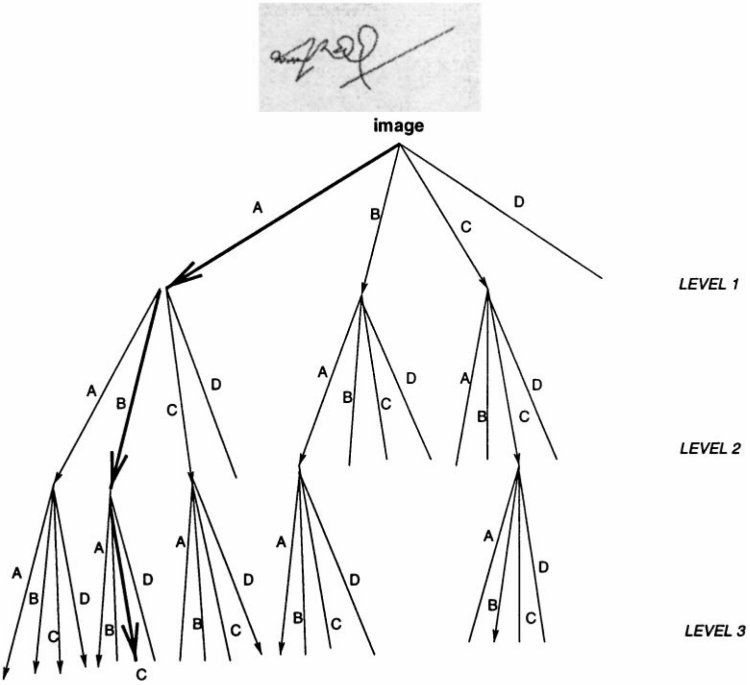

Fig. 5. Quadtree representation of tree-structured wavelet transform.

Table 1
Prominent energy maps

Energy path	Energy value at leaf
AAA	59.88
AAC	48.10
BAA	46.17
AAD	44.99
AAB	43.19
ABA	37.64
ACD	35.07
ACB	32.99
ABC	32.50
CCB	31.43

where '*' denotes array multiplication of vectors, **f** is the raw feature vector and **w** ($w_i \geqslant 0\ \forall i$, $\|\mathbf{w}\| = 1$) is the weight vector, Global geometric features, Moment measures and Envelope characteristics were all represented as weighted feature vectors.

4.1. Choice of weights — Genetic Algorithms

The weighted feature vector can be considered to be better than the raw feature vector only if the weights are appropriately assigned. An improper distribution of weights to the features would be disastrous as it may reduce the weightage to a good feature. A "good" weightage vector can be determined by either (1) analyzing the individual feature values of some test patterns, and assessing how representative each feature is with respect to the patterns or (2) finding an optimal or near-optimal weight vector based on a good representation measure as the objective function. The first method may only be a rough estimation of the weight vector. The second method requires searching the N-dimensional hyper space in order to determine the optimal weight vector. Obviously, an exhaustive search through the weight vector space is not preferable. A Hill-Climbing approach can be applied, but may get trapped in local optima. This is where a genetic optimization process can help.

Genetic algorithms are robust and are not likely to be affected by the presence of spurious local optima in the solution space. They can concentrate on a set of promising solutions that may converge to a set of globally optimal solutions. The operation cycle involves a number of adjustable parameters (like the elements of weight vector), all of which could take part in an optimization process simultaneously. Natural environment enables the fittest individuals to survive and reproduce in a competitive ecology. GAs simulate this *survival of the fittest* mechanism by targeting at the optimal or near optimal solutions in complex search spaces. A good overview is provided by Goldberg [10]. So, GAs involve chromosomes, or individual solutions which go through a process of evolution. The chromosomes which are fitter (in terms of the acceptability of the solution) will survive and move on to the next generation passing on their genetic characteristics through offspring. The children forming the new generation are created through *selection* of two parents with high relative fitnesses, *crossover* between the selected individuals and *mutation*. *Crossover* is performed by randomly exchanging parts of the selected chromosomes and *mutation* represents the phenomenon of rare chance in the evolution process. The traditional GA chromosome representation is that of a bit string. In our case, chromosome is modeled as a string of floating-point numbers derived from the weight vector. Each gene in the chromosome can take real numbers in the range (0, 1) as its allele. Length of a chromosome is equal to the dimension of the weight vector.

Fitness function of a GA is the objective function of the optimization problem. It is dependent on the chromosome (terms chromosome and weight vector will be used interchangeably). It should be chosen so that maximizing it will help us in achieving better recognition of patterns. In our verification problem, we have to choose a fitness function which will increase the verification accuracy. One way is to earmark certain number of genuine and forgery patterns (called validation patterns) for this purpose. With each weightage vector, we have to go through the verification process, calculate the verification accuracy and return it as fitness to the GA module. This process has to be repeated for each of the chromosomes present in the population. However, this scheme is inelegant as it requires some extra patterns set aside for its own sake. Also, such patterns cannot be fewer in number as the verification accuracy may be under-estimated. It would be appropriate if we set the fitness based on training samples alone. We have used the following heuristic measure as the fitness function:

$$\textit{fitness} = \frac{\sum_{i=1}^{G}\sum_{j=1}^{F} d(g_i, f_j)}{\sum_{i=1}^{G}\sum_{j=1}^{G} d(g_i, g_j)},$$

where G and F are the number of genuine and forgery training samples respectively, g_i and f_j are the ith genuine sample and jth forgery sample respectively, and $d(.,.)$ is the distance measure. In the absence of forgery samples for training purposes,

$$\textit{fitness} = \frac{1}{\sum_{i=1}^{G}\sum_{j=1}^{G} d(g_i, g_j)}$$

can be used as the the fitness function. These measures are clearly not the same as verification accuracy. But, a feature is good or bad depending on how far it distinguishes forgery from genuine and to what extent it treats all the genuines at par. Hence, with the above fitness functions, the verification accuracy is likely to improve. Note that the fitness functions depend only on the training patterns.

The initial population consists of a reasonable number of different weight vectors. The vectors are chosen in such a way that they almost span the whole weight vector space. In each iteration, fitness values are calculated for all the chromosomes. Two chromosomes are chosen randomly (the probability depends directly on the fitness value) and are allowed to cross-over. During cross-over, the chromosome pair is split at some randomly chosen position along its length, say, at the second gene from right, and the four resulting segments are recombined by joining the head of one chromosome to the tail of the other. Fig. 6 illustrates how this is done. This is a simple single-point crossover. The mutation operation, which consists of selecting a random gene (element of weight vector) and altering its value, is applied very rarely (with a very low probability). The resultant chromosomes are members of the new population. The cycle of Selection, Crossover and Mutation is repeated until the new population is filled up.

The new population consists of weight vectors whose average fitness value is likely to be higher than that of the previous population. This optimization cycle has to be carried out over, say, r generations (fixed, for simplicity, at 20).

In the initial stages of GA runs, it is common to have a few extraordinary individuals amidst mediocre members. Employing a fitness-based selection rule would make the extraordinary individuals to attain majority in a single generation and lead to premature convergence. Another kind of problem likely to prop up at later generations is the significant diversity within the population with the population average fitness staying close to the population best fitness. This will lead to a random walk among the mediocre in place of survival of the fittest. Fitness scaling methods, as outlined by Goldberg [10], have been applied to allay the above-mentioned problems. If there is some little diversity left at the end of r generations, the chromosome having maximum fitness has been chosen as the final optimal weight vector.

4.1.1. Distance measure for wavelet-based representations

On employing a tree-structured wavelet transform, we get a set of energy maps for a pattern as shown in Table 1. The first three elements in each map correspond to the path in the tree and the last element is the energy at the leaf end of the path. Obviously, simple L_1 norm cannot be applied to this feature set. The following algorithm was used to calculate the distance in Wavelet domain between two signatures:

```
initialize distance to zero
for each energy map of pattern A
    if the path is not contained in energy maps of pattern B
        increment distance
end
for each energy map of pattern B
    if the path is not contained in energy maps of pattern A
        increment distance
end
```

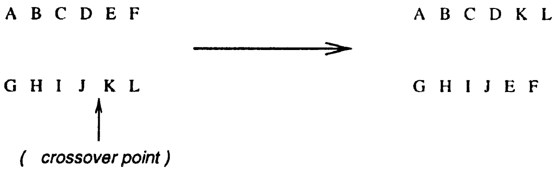

Fig. 6. Genetic crossover between strings.

4.2. Verification criteria

Several verification criteria have been attempted in this work. A verification criterion may depend on the genuine training patterns alone or both genuine and forgery training patterns. In the former case, it is assumed that the feature vectors derived from genuine signatures tend to cluster and the classification procedure relies on a threshold on the distance to the cluster. In the latter case, we may opt for a classification problem into either genuine or forgery.

4.2.1. Confidence interval approach

This addresses the verification problem with no forgery samples for training. Let NF be the number of features to be used; μ_I be the mean of the Ith feature, computed on a set of known genuine signatures; σ_I the standard deviation of the Ith feature, computed on the same set. The distance for an unknown signature is defined as in [11] :

$$dist = \left[\sum_{I=1}^{NF} \left(\frac{FV_I - \mu_I}{\sigma_I} \right)^2 \right]^{1/2}.$$

where FV_I is the measured value of the Ith feature on the unknown signature. To classify unknown signatures, we choose a threshold value T. If the measured distance for a given unknown signature is greater than T, the signature is labeled a forgery; otherwise, it is declared to be genuine. We can choose T to correspond to a given value of z for the normal distribution. If we constrain $(FV_I - \mu_I)/\sigma_I)$ to be atmost z for each I for a signature to be authentic, *dist* must be atmost

$$\left(\sum_{1}^{NF} z^2 \right)^{1/2} = (NFz^2)^{1/2} = \sqrt{NF}z$$

so that the corresponding value of T is $z\sqrt{NF}$. For example, let $z = 2.575$, which corresponds to the case where each feature value FV_I has a probability of 99% of belonging to the normal distribution of mean μ_I and standard deviation σ_I [11]. Then the corresponding value of T is $2.575\sqrt{NF}$. This choice of T is to minimize the number of genuines being concluded as forgery.

4.2.2. Min–Max approach

This approach requires both genuine and forgery patterns to be available for training. Let the maximum distance between a forged training sample i and all the genuine samples be d^i_{max}. Let the minimum of all d^i_{max}'s be d_m. The idea is to identify the forgery sample that is nearest to the genuine training samples. The maximum distance between this forgery and the genuines, d_m is a measure of the boundary of the genuines beyond which all other forged training patterns lie. We took a factor of this measure as the threshold T.

$$T = k_m d_m,$$

where $k_m \in \Re$ is a constant. If the maximum of the distances between a test signature and all the genuine training signature is greater than T we declare it to be forgery, else accept it as genuine.

4.2.3. N-dimensional boundary

This approach can be employed when forgery patterns are not available for training. The above two approaches have a scalar T as the threshold for verification. One can always find a forgery which is reasonably far from the genuines with respect to one of the features and too close to the genuines with respect to all other features. Such a forgery will escape the distance threshold test. Suppose we have a set of thresholds $\{T_1, T_2, \dots, T_{NF}\}$ (where NF is the number of features), instead of a single threshold T, the boundary formed will be a N-Dimensional hyper-cuboid. The projection of the hyper-cuboid along any one axis has as limits,

$$f_{min} - k_v v \quad \text{and} \quad f_{max} + k_v v,$$

where f_{min} and f_{max} are the minimum and maximum feature values, respectively, of all the genuine training patterns in that projection, v is a measure of probable deviation of a test genuine from the training genuine samples given by $E|f - \mu|$ (where f is the vector containing feature values of all the training patterns for that feature and μ is their mean) and $k_v \in \Re$ is a factor of deviation. We used this set of limits (both higher and lower) as the set of thresholds.

4.2.4. Neighbourhood approach

This method can be applied only if both genuine and forgery patterns are available for training. The decision is based on which class is closer to the test signature. Many strategies are possible using this idea like nearest neighbour, K-nearest neighbour and modified K-nearest neighbour. The nearest-neighbour classifier (NNC) decides the validity of a signature on the class label of the training sample which is closest to it. Given a genuine test signature, there is a chance that it will get classified incorrectly as one skilled forgery may be nearest to it. A forged test signature, if found closest to a forgery sample, will be correctly classified. However, since the forgery training samples are not exhaustive, a forged signature may be nearer to a genuine than to any of the provided forged patterns. K-nearest-neighbour classifier (KNNC) considers the "K" nearest training patterns ("K" is a constant) to a test signature. The class label which forms a majority in the "K" neighbours is concluded as that of test signature. Since the genuine training patterns are all almost alike, K-nearest-neighbour classifier (KNNC) almost always classifies a genuine signature correctly when "K" is high. However, a forged test signature may be wrongly classified even if few training forgeries are near, but many genuines are nearer to it than other forgeries. Though KNNC has better performance than NNC for genuine signatures, it is worse for forgeries.

NNC has the drawback that it considers only one nearest neighbour. KNNC has a pitfall that it does majority voting without taking distances into consideration. A blend of NNC and KNNC into a modified KNNC can be attempted as follows: Let the "K" nearest samples be $\{X_1, X_2, \dots, X_k\}$ and d_j be the distance between test signature and jth nearest neighbour, we associate with jth nearest neighbour an importance measure given by

$$w_j = \frac{d_k - d_j}{d_k - d_1}$$

$w_j = 0$ for $j = k$ and $w_j = 1$ for $j = 1$. So, the importance measure is directly related to the nearness to the test signature. Let $\{X_{i_1}, X_{i_2}, \dots, X_{i_p}\}$ be the genuine samples and $\{X_{i_{p+1}}, X_{i_{p+2}}, \dots, X_{i_k}\}$ be the forgery samples. Then, if

$$(w_{i_1} + w_{i_2} + \cdots + w_{i_p}) > (w_{i_{p+1}} + w_{i_{p+2}} + \cdots + w_{i_k})$$

then the test signature is concluded as genuine, else as forgery. If the sum of the importance measures for the genuine and forgery classes among the "K" neighbours are equal, either NNC or KNNC can be applied.

4.3. Hybrid approach

Though the Modified KNNC is able to capture the advantages of NNC and KNNC, the problem of a forgery lying too far from the genuines, yet nearer to the genuines than to the forgeries is not handled. We attempted to combine the qualities of the neighbourhood approach and the other approaches treated earlier into a hybrid approach. Since modified KNNC is likely to be incorrect when it concludes a signature as genuine, we can handle this case by passing such patterns to any of the other threshold-based classifiers discussed in Sections 4.2.1–4.2.3. But when the Modified KNNC concludes a test signature as forgery, we accept it and no more scrutinization is done.

5. Results

Signature verification is a problem which cannot be approached with a deterministic framework of formulae alone or with sheer set of well-defined rules. It relies primarily on experimentation and careful analysis of requirements of the application. The particular value of the threshold T to be fixed for verification will determine the probable number of false acceptance (forgery concluded as genuine) and false rejection (genuine concluded as forgery) cases. The choice of T should therefore be based on the cost incurred on overseeing these two types of errors by the system, availability of human document examiners to verify the decision and a priori probability of freehand forgeries.

Four types of signature representations were discussed in Section 3, viz., geometric features, moment-based representations, envelope characteristics and wavelet-based features. Four types of classification techniques were described in Section 4, viz., confidence-interval technique, Min–Max method, N-dimensional boundary technique and the neighbourhood methods. Each of the feature extraction methods has been considered individually with every classification technique. The hybrid approach (Section 4.3) of integrating threshold-based classification methods with neighbourhood techniques has also been attempted. The modified KNNC has been used uniformly as the Neighbourhood classifier for the purpose of experiments. As an attempt to build more reliability into the system, a combination of the individual verification systems (with different feature extraction techniques and same classifier) has been performed. Such a combination operation is shown in Fig. 7 for the N-dimensional boundary classifier. Given a test signature, each of the subsystem is allowed to pronounce its decision (genuine or forgery). Combination is achieved by declaring the decision of the majority of the subsystems as the final decision. If two subsystems conclude as genuine and the remaining two as forgery, then the test signature is *rejected*. By rejecting a test signature, it is implied that it has to be passed for further scrutinization – either to any auxiliary verification system or to human examiners. The *reject* option will prevent many wrong decisions when there is ambiguity or little certainty. The decision (as either genuine or forgery) made by the combination system is likely to be closer to reality.

Any verification system will normally have its boundary threshold pre-determined before installation. Given a particular threshold, a specific system may be very optimal under genuine test cases and may perform poorly under forgeries or vice versa. A system is considered good only if it exhibits optimal performance under both genuine and forged test cases.

Fig. 8 shows the results obtained when the classifier is based only on the Confidence-Interval (CI) approach discussed in Section 4.2.1. Fig. 8a shows the percentage error in the decisions made by the verification system when the test signatures were authentic. The plots were made with respect to percentage confidence, which amounts to the confidence we have for the genuine signatures to fall within the distance threshold. Wavelet-based features are incompatible to taking mean and other

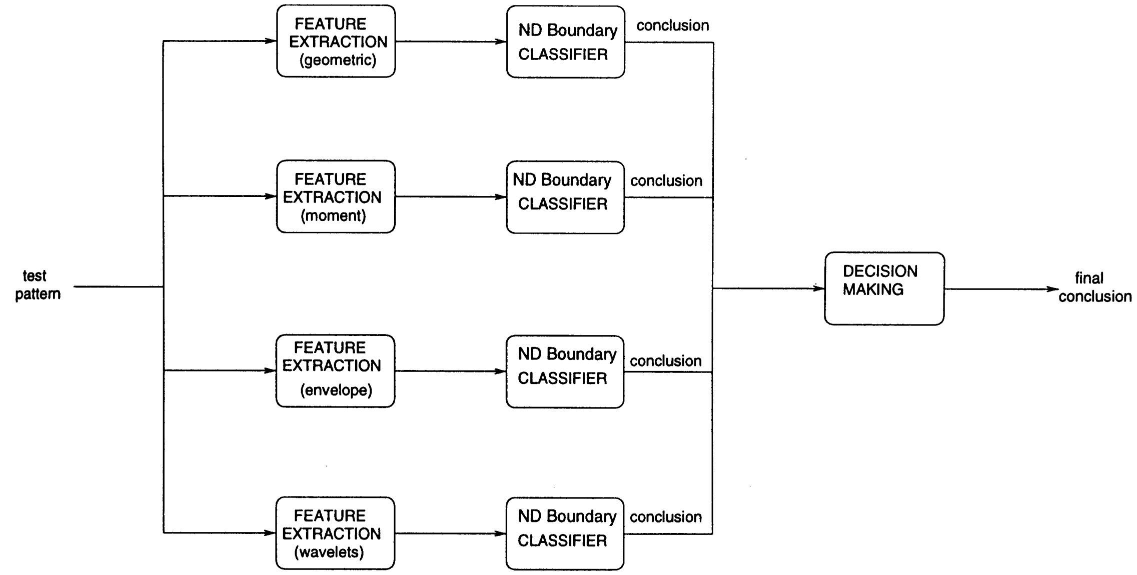

Fig. 7. Combination of conclusions.

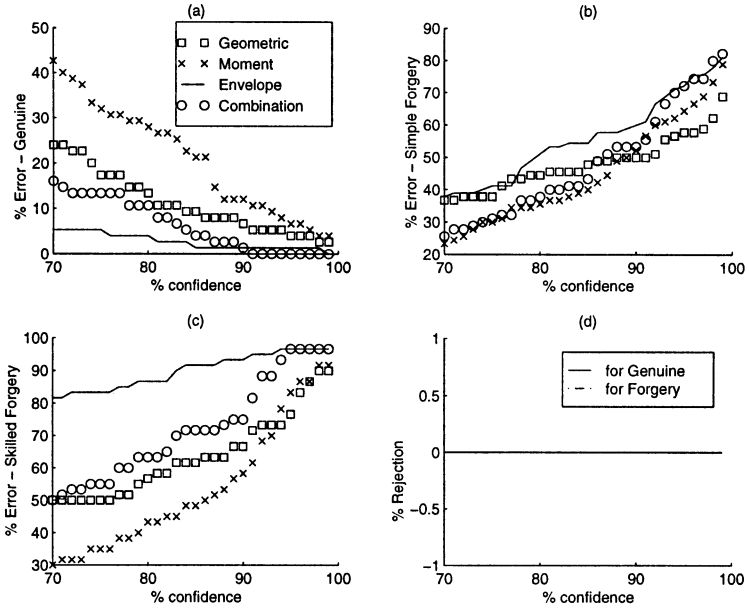

Fig. 8. Confidence-interval classifier.

statistics, and hence do not figure in the plots for this classifier. Note that with all the feature extraction techniques and for the combination system, the percentage error decreases as the percentage confidence rises. This trend can be observed in all subsequent plots also. It is natural because as the confidence increases, the distance threshold increases (or in this case, confidence interval gets broader) and there is a likelihood for more and more test signatures to fall within the distance threshold and hence the verification accuracy shoots up. Fig. 8a shows that the system with envelope characteristics alone for pattern representation produces the least error. This shows that envelope characteristics are able to capture the similarity among genuine signatures much more than other features. The moment features produce the maximum error. It can be found that the combination system falls closer to envelope characteristics in performance. Fig. 8b[1] shows the performance when the test signatures were all simple forgeries. It can be found from the figure that the envelope characteristics have the worst performance. This can be quite critical as applications may require most of the forgeries to be caught even at the cost of some genuine signatures being pronounced as forgeries. All the individual feature extraction methods exhibit complimentary performance under genuine and forged test signatures. But, the combination system tends to fall on the low verification error region under both conditions. Fig. 8c shows the percentage error when the test signatures were all skilled forgeries. The systems based on geometric and moment features (being global features) are able to detect more forgeries, even the skilled ones. The combination system exhibits an average performance. Since the combination system involves only three subsystems (as wavelet-based subsystem is excluded) and there are only two conclusions possible (genuine or forgery), there cannot be any ambiguity and simple majority always exists. So the are no rejection cases as shown in Fig. 8d.

Fig. 9 shows the performance characteristics for the Min–Max classifier. The plots are versus the factor of relaxation k_m in the boundary threshold d_m (see Section 4.2.2). For the combination system, in the region where the confidence-interval classifier produced less than 15% errors under genuine cases, the error characteristics vary from 25 to 80% under forgeries. But, in the corresponding region for the Min–Max classifier (i.e. for factor of relaxation greater than 1.25), the minimum error is 40% under forgeries. This shows that the former classifier has better performance. Even under skilled forgeries, we can see that Confidence–Interval classifier achieved similar or better performance than the Min–Max classifier with nil rejections.

Fig. 10 shows the performance characteristics for the N-dimensional boundary classifier. The plots are versus the factor of deviation v (see Section 4.2.3) in threshold. In the low threshold region (a desirable region where most of the forgeries are caught), we can see that the system with envelope characteristics and the combination

[1] For better clarity of plots, the legend box indicating the linestyles of sub-plots (a), (b) and (c) of all the result graphs is shown only on one of them in the respective graphs

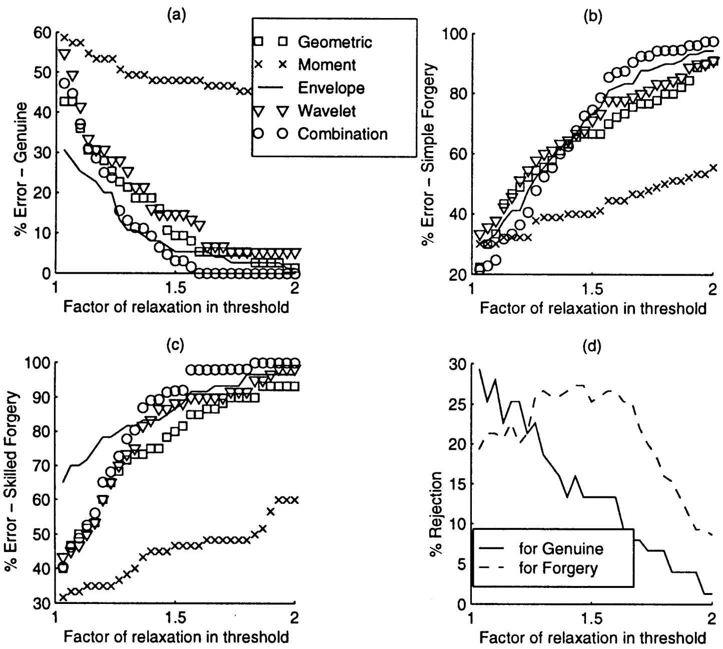

Fig. 9. Min–Max classifier.

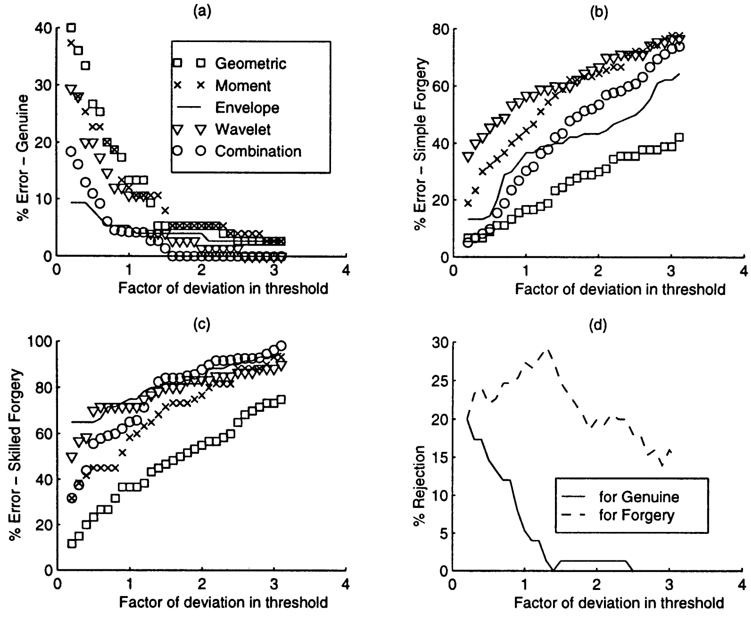

Fig. 10. N-dimensional boundary classifier.

system perform reasonably well under both genuine and simple forgery cases. All other systems behave in antithetic fashion. Also, for the combination system, in the region where this classifier produces less than 15% error under genuine cases, the minimum error is only 10% under simple forgeries. The minimum error under skilled forgeries in the same region is 40% whereas the corresponding measure is over 50% for the earlier two classifiers. The rejection characteristics seem to be similar to that of Min–Max classifier.

An experiment was conducted using the Modified KNNC as the system classifier. A good verification system should not depend on a neighbourhood classifier alone. Such an experiment was carried out in order to estimate the capability of the classifier. Verification accuracies of 95.6% for genuine signatures and 86.7% for forged signatures with the combination system were achieved.

Fig. 11 shows the performance characteristics of the classifier based on applying the hybrid approach (see Section 4.3) to the modified KNNC and the confidence-interval classifier. Wavelet features figure in these plots as the Neighbourhood classifier can make use of them, but they do not change with percentage confidence, as they are not compatible with the Confidence-Interval classifier. One can also find a more even characteristics than for CI classifier as some of the conclusions are made by the Neighbourhood classifier. The combination system seems to give the best performance as the error does not exceed 15% under both genuine and forged test cases. Focusing on the low threshold region, one can achieve more than 95% accuracy under simple forgeries, more than 90% under genuine cases and about 70% for skilled forgeries. But, this improvement in performance is achieved at the cost of about 20% rejection in the low threshold region. The percentage error seems to have increased under genuine test cases for the hybrid classifier when compared to that for the CI classifier. This might have been due to the neighbourhood classifier concluding some of the genuines as forgeries (and hence the test pattern did not get passed on to the CI classifier). The hybrid classifier based on Min–Max approach was found to be inferior to the one based on CI classifier.

Fig. 12 shows the performance characteristics by combining N-dimensional boundary and neighbourhood classifiers. Here too, as in the earlier hybrid classifiers, the percentage error has marginally increased for genuine cases when compared to the stand-alone N-dimensional boundary classifier. Though the verification accuracy achieved under genuine and simple forgery cases is almost similar to the first hybrid classifier discussed (with CI and neighbourhood classifiers), the performance under skilled forgeries is better for the former than the latter. Wavelet features can be found to exhibit even characteristics and low error under skilled forgeries with both the hybrid classifiers. The rejection characteristics are also similar for the two hybrid classifiers. Thus, with this hybrid classifier, a verification accuracy of 90% under genuine cases, over 98% under simple forgeries and around 70–80% against skilled forgeries with about 15–20% rejections can be achieved. The verification system was implemented in Java Programming Language on Sun's Ultra Sparc System.

The prevailing process of signature verification is managed by human experts. Hence, this research work would be incomplete without taking into consideration the efficacy of human document analysts. We provided the same data sets to two teams of examiners for verification. Table 2 summarizes the results obtained from this exercise. One can observe that all the simple forgeries are

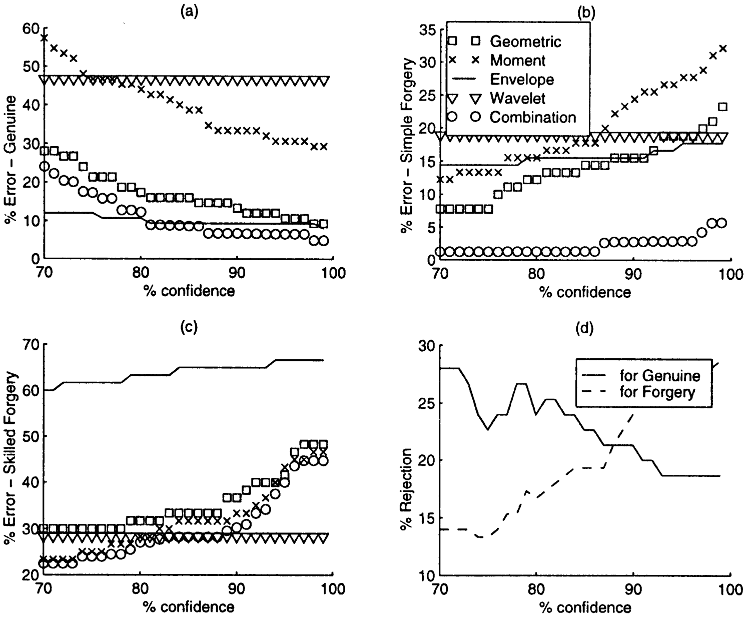

Fig. 11. Hybrid classifier – CI + neighbourhood.

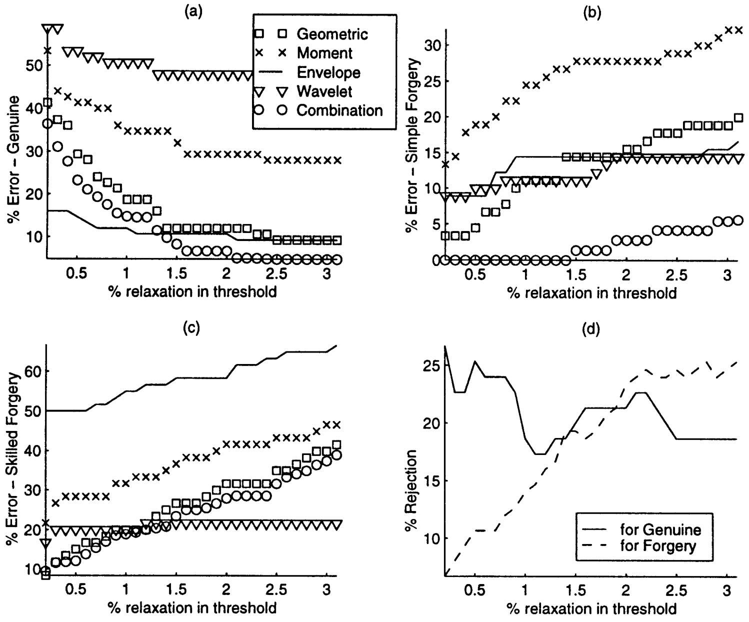

Fig. 12. Hybrid classifier – N-dimensional boundary + neighbourhood.

Table 2
Verification results of human experts

Genuine	Simple forgeries	Skilled forgeries
82%	100%	75%

successfully identified by the analysts whereas machine fell short of perfection. This behaviour may be attributed to the fact that machine could not comprehend the arbitrariness of simple forgeries. The performance is comparable to that of machine on skilled forgeries. Humans tend to play safe in the absence of any data associated with an individual but for the signature. This could be the reason for a higher number of false rejections as observed in Table 2.

6. Conclusions

Signature verification is an extremely complex problem and the philosophy that combination of different representations alone can make way for a reliable system has been emphasized all through the paper. The whims and fancies of the signatures produced by humans can never be handled by a single representation scheme, however "good" it may appear. Experimental results establish the effectiveness of the philosophy. Also, integration of the Neighbourhood method and the threshold-based techniques seem to work well, particularly for forgeries. The weighted feature vector (with the weights determined by Genetic Algorithms) was found to represent the patterns better than the raw feature vector. The above-mentioned ways of combining conclusions or different pattern representations are likely to be beneficial irrespective of the representativeness of the features.

A simple majority voting scheme was used in deciding on several conclusions. One can attempt to give a weightage factor to the conclusion from each of the subsystems for better decision making. Here too, the usefulness of GA can be exploited.

Image rotations were required in many places like making signatures rotation-invariant and in finding the slant angle. Signature image scaling was effectively done by a new algorithm (see Section 2.2). But, it cannot be directly applied to image rotations. An efficient and viable algorithm needs to be devised for rotation operations suitable for signatures.

Tree-structured Wavelet features are not compatible with CI classifier. The set of 10 thresholds required for the N-Dimensional boundary classifier with the wavelet features were set based on the *energy values* of the 10 dominant energy paths whereas the paths are the elements of feature vector. These may not form a characteristic set of thresholds. The distance measure algorithm of Section 4.1.1 was used with Min–Max classifier, which according to experimental results, is an inferior classifier. Better classifiers need to be suggested for these wavelet features.

The results obtained from human verifiers reveal that the performance of machine is not adequate under simple forgeries. Hence, further study is required in the direction of making machines fail-proof when subjected to patently naive inputs.

Acknowledgements

The authors thank the Branch Managers of State Bank of India, Indian Institute of Science Campus, Bangalore, India and Canara Bank, Jayanagar 9th block branch, Bangalore, India for providing conclusions of human experts. The comments of the anonymous referee have helped in enhancing the quality of the paper.

References

[1] I-C. Jou, Q.-Z. Wu, S.-Y. Lee. On-line signature verification using LPC Cepstrum and neural networks. IEEE Trans. Systems Man Cybernet. 27(1) February (1997) 148–153.

[2] W. Namcek, W. Lin, Experimental investigation of automatic signature verification, IEEE Trans. Systems Man Cybernet. 4 (1974) 121–126.

[3] S. Lee, J.C. Pan, Offline tracing and representation of signatures, IEEE Trans. Systems Man Cybernet. SMC–22 (1992) 755–771.

[4] Y. Qi, B.R. Hunt, Signature Verification using global and grid features, Pattern Recognition 27(12) (1994) 1621–1629.

[5] R. Bajaj, S. Chaudhury, Signature verification using multiple neural classifiers, Pattern Recognition 30(1) (1997) 1–7.

[6] K. Huang, H. Yan, Off-line signature verification based on geometric feature extraction and neural network classification, Pattern Recognition 30(1) (1997) 9–17.

[7] R.N. Nagel, J.S. Weszka, A. Rosenfeld, A threshold selection technique, IEEE Trans. Comput. C-23(12) (1974) 1322–1326.

[8] H. Al-Yousefi, S.S. Udupa, Recognition of arabic characters. IEEE Trans. Pattern Anal. Machine Intell. PAMI-14 (1992) 853–857.

[9] T. Chang, C.C. Jay Kuo, Texture analysis and classification with tree–structured wavelet transform, IEEE Trans. Image Process. 2(4) (1993) 429–441.

[10] D.E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, Reading MA. 1989.

[11] R.N. Nagel, A. Rosenfeld, Computer detection of freehand forgeries, IEEE Trans. Comput. C–26(9) (1977) 895–905.

About the Author—V.E. RAMESH received his B.E. degree from Anna University, Madras, India in Electrical and Electronics Engineering and M.E. degree from Indian Institute of Science, Bangalore, India in the Department of Computer Science and Automation. His research interests include Pattern Recognition, Artificial Intelligence and Fuzzy systems.

About the Author—M. NARASIMHA MURTY received his B.E. and M.E. degrees in Electrical Engineering from the Indian Institute of Science, Bangalore. He received his Ph.D. degree from Indian Institute of Science, Bangalore in Pattern Recognition in 1982. He is a Professor in the Department of Computer Science and Automation. His research interests are in Pattern Recognition and Genetic Algorithms.