

Pattern Recognition 35 (2002) 861-874



www.elsevier.com/locate/patcog

Decision-level fusion in fingerprint verification

Salil Prabhakar^{a, *}, Anil K. Jain^b

^a Algorithms Research Group, Digital Persona, Inc., Redwood City, CA 94063, USA ^b Department of Computer Science and Engineering, Michigan State University, East Lansing, MI 48824, USA

Received 4 January 2001; accepted 9 February 2001

Abstract

A scheme is proposed for classifier combination at decision level which stresses the importance of classifier selection during combination. The proposed scheme is optimal (in the Neyman–Pearson sense) when sufficient data are available to obtain reasonable estimates of the join densities of classifier outputs. Four different fingerprint matching algorithms are combined using the proposed scheme to improve the accuracy of a fingerprint verification system. Experiments conducted on a large fingerprint database (~ 2700 fingerprints) confirm the effectiveness of the proposed integration scheme. An overall matching performance increase of $\sim 3\%$ is achieved. We further show that a combination of multiple impressions or multiple fingers improves the verification performance by more than 4% and 5%, respectively. Analysis of the results provide some insight into the various decision-level classifier combination strategies. © 2002 Pattern Recognition Society. Published by Elsevier Science Ltd. All rights reserved.

Keywords: Classifier combination; Parzen density estimate; Feature selection; Biometrics; Verification; Combination of matchers; Neyman–Pearson; Fingerprint

1. Introduction

It is often observed that different classifiers with essentially the same overall accuracy misclassify different test patterns. In an attempt to harness the complementary decision boundaries constructed by different classifiers, a large number of information fusion strategies have been proposed that combine the available information at different levels (i.e., sensor level, representational level, and decision level). Successful "multiclassifier" recognition systems [1–7,28,29] have been built in different application domains demonstrating the usefulness of information fusion. A comprehensive list of classifier combination strategies can be found in Refs. [8,2]. However, a priori it is not known which combination strategy works better than the others and if so under what circumstances.

* Corresponding author.

E-mail addresses: salilp@cdigitalpersona.com (S. Prabhakar), jain@cse.msu.edu (A.K. Jain).

In this paper we will restrict ourselves to a particular decision-level integration scenario where each classifier may select its own representation scheme and produces a confidence value as its output. A theoretical framework for combining classifiers in such a scenario has been developed by Kittler et al. [2]. However, the product rule for combination suggested in Ref. [2] implicitly assumes an independence of classifiers. The sum rule further assumes that the aposteriori probabilities computed by the respective classifiers do not deviate dramatically from the prior probabilities. The max rule, min rule, median rule, and majority vote rule have been shown to be special cases of the sum and the product rules. Making these assumptions simplifies the combination rule but does not guarantee optimal results and hinders the combination performance. We follow Kittler et al.'s framework without making any assumptions about the independence of various classifiers.

The contributions of this paper are two fold. Firstly, we propose a general system design for decision-level

classifier fusion that uses the optimal Neyman–Pearson rule and outperforms the combination strategies based on the assumption of independence amount the classifiers. Secondly, we propose a multi-modal biometric system design based on multiple fingerprint matchers. The use of the proposed combination strategy in combining multiple matchers significantly improves the overall accuracy of the fingerprint-based verification system. The effectiveness of the proposed integration strategy is further demonstrated by building multi-modal biometric systems that combine two different impressions of the same finger or fingerprints of two different fingers.

The rest of the paper is organized as follows: Section 2 gives a brief overview of biometrics and multi-modal biometric systems. Section 3 presents the proposed integration design which includes classifier selection, non-parametric density estimation, and optimal integration strategy. Section 4 gives a brief description of the four different fingerprint verification systems used in our case study. The fingerprint database, experimental results, and analysis of the results are presented in Section 5. Finally, Section 6 concludes the paper.

2. Biometrics

A reliable automatic person identification is critical in a wide variety of forensic, civilian, and commercial applications such as criminal investigation, issuing driver's license, welfare disbursement, credit cards and cellular phone usage, and access control. Biometrics [31], which refers to identification of people based on their physical or behavioral characteristics is inherently more reliable than traditional knowledge-based (such as a password) or token-based (such as an access card) systems. A physical or behavioral characteristic that has universality, distinctiveness, permanence, and collectability (such as fingerprint, iris, voice, face, etc.) is a candidate biometric for designing an automatic authentication system. Biometric-based identification is preferred over traditional methods because a biometric cannot be forgotten or lost. A biometric system is essentially a pattern recognition system that may work in two different modes: (i) verification, and (ii) recognition. Verification refers to authenticating the claimed identity of a user while recognition refers to determining the identity of a user. Recognition is inherently a more difficult pattern recognition problem as it involves a large number of classes. Verification is a relatively easier problem that can be formulated as a simple hypothesis testing problem. We will focus on only the verification problem in this paper and will use the words verification, authentication, and recognition, interchangeably to refer to the two-class (accept or reject) verification problem.

The biometric verification problem can be formulated as follows. Let the stored biometric signal (template) of a person be represented as *S* and the acquired signal (input) for authentication be represented by *I*. Then the null and alternate hypotheses are:

- H_0 : $I \neq S$, input fingerprint does not come from the same finger as the template,
- H_1 : I = S, input fingerprint comes from the same finger as the template.

The associated decisions are as follows:

- D_0 : person is an imposter,
- D_1 : person is genuine.

The verification involves matching S and I using a similarity measure. If the matching score is less than some decision threshold T, then decide D_0 , else decide D_1 . The above terminology is borrowed from communication theory where we want to detect a message in the presence of noise. H_0 is the hypothesis that the received signal is noise alone and H_1 is the hypothesis that the received signal is message plus the noise. Such a hypothesis testing formulation inherently contains two types of errors: Type I: false acceptance $(D_1$ is decided when H_0 is true) and Type II: false rejection (D_0 is decided when H_1 is true). False acceptance rate (FAR) is the probability that the system makes type I error (also called significance level of the hypothesis test) and false rejection rate (FRR) is the probability that the system makes type II error. Note that (1-FRR) is also called the power of the test.

$$FAR = P(D_1|w_0),$$

$$FRR = P(D_0|w_1)$$

where w_0 is the class with $H_0 = true$ and w_1 is the class with $H_1 = true$. There is a trade-off between the two types of errors (FAR and FRR) in a biometric system. Different applications may have different requirements on the error rates. For example, high security access applications have more strict requirements on the FAR than, say, forensic applications. A system designer may not know in advance the particular application for which the system may be used (or a single system may be designed for a wide variety of applications). So, it is a common practice to report the system performance at all operating points (decision thresholds). This is done by plotting a receiver operating characteristic (ROC) curve. A ROC curve is a plot of FAR (significance level) with 1-FRR (power) for various decision thresholds. The system designer's challenge is to minimize the FRRs for various specified FARs.

Several biometric systems have been designed and tested on large databases. However, in some applications with stringent performance requirement, no single biometric can meet the requirements due to inexact nature



Fig. 1. Various multi-modal biometric systems.

of sensing, feature extraction, and matching processes. This has generated interest in designing multi-modal biometric systems [9]. Multi-modal biometric systems may work in one of the following five scenarios (see Fig. 1): (i) multiple sensors: for example, optical, ultrasound, and capacitance-based sensors are available to capture fingerprints; (ii) multiple biometric system: multiple biometrics such as fingerprint and face may be combined [1,2,10]; (iii) multiple units of the same biometric: one image each from both the iris, or both hands, or 10 fingerprints may be combined [11]; (iv) multiple instances of the same biometric: for example multiple impressions of the same finger [11], or multiple samples of the voice, or multiple images of the face may be combined; (v) multiple representation and matching algorithms for the same input biometric signal: for example, combining different approaches to feature extraction and matching of fingerprints [12]. The first two scenarios require several sensors and are not cost effective. Scenarios (iii) causes inconvenience to the user in providing multiple cues and has a longer acquisition time. In scenario (iv), only a single input is acquired during verification and matched with several stored templates acquired during

the one-time enrollment process. Thus, it is slightly better than scenario (iii). In our opinion, scenario (v) is the most cost-effective way to improve biometric system performance.

We propose to use a combination of four different fingerprint-based biometric systems where each system uses different feature extraction and/or matching algorithms to generate a matching score which can be interpreted as the confidence level of the matcher. These different matching scores are combined to obtain the lowest possible FRR for a given FAR.

We also compare the performance of our integration strategy with the sum and the product rules [2]. Even though we propose and report results in scenarios (iii)–(v), our combination strategy could be used for scenarios (i) and (ii) as well.

3. Optimal integration strategy

Let us suppose that pattern Z is to be assigned to one of the two possible classes, w_0 and w_1 . Let us assume that we have N classifiers, and the *i*th classifiers outputs a single confidence value θ_i about class w_1 (the confidence for the class w_0 will be $1 - \theta_i$), i = 1, 2, ..., N. Let us assume that the prior probabilities for the two classes are equal. The classifier combination task can now be posed as an independent (from the original N classifier designs) classifier design problem with two classes and N features $(\theta_i, i = 1, 2, ..., N)$.

3.1. Classifier selection

It is a common practice in classifier combination to perform an extensive analysis of various combination strategies involving all the N available classifiers. In feature selection, it is well known that the most informative d-element subset of N conditionally independent features is not necessarily the union of the d individually most informative features [13-16]. Cover [17] argues that no non-exhaustive sequential d-element selection procedure is optimal, even for jointly normal features. He further showed that all possible probability of error ordering can occur among subsets of features subject to a monotonicity constraint. The statistical dependence among features causes further uncertainty in the *d*-element subset composed of the individually best features. One could argue that the combination strategy itself should pick out the classifiers that should be combined. However, we know in practice that the "curse of dimensionality" makes it difficult for a classifier to automatically delete less discriminative features [18,19,30]. Therefore, we propose a classifier selection scheme prior to classifier combination. We propose to use the *class separation* statistic [20] as the feature effectiveness criterion. This statistic, CS, measures how well the two classes (imposter and genuine, in our case) are separated with respect to the feature vector, X^d , in a *d*-dimensional space, R_d .

$$CS(X^{d}) = \int_{R^{d}} |p(X^{d}|w_{0}) - p(X^{d}|w_{1})| \, dx, \qquad (1)$$

where $p(X^d|w_0)$ and $p(X^d|w_1)$ are the estimated distributions for the w_0 (imposter) and w_1 (genuine) classes, respectively. Note that $0 \le CS \le 2$.

We will use the class separation statistic to obtain the best feature subset using an exhaustive search of all possible $2^N - 1$ feature subsets.

3.2. Non-parametric density estimation

Once we have selected the subset containing d ($d \le N$) features, we develop our combination strategy. We do not make any assumptions about the form of the distributions for the two classes and use non-parametric methods to estimate the two distributions. We will later show that this method is superior to a parametric approach which approximates the form of the density.

The Parzen window density estimate of an d-dimensional density function based on n observations is given by [21]

$$P(X) = \frac{1}{nh^d} \sum_{j=1}^n \left\{ \frac{1}{(2\pi)^{d/2} |\Sigma|^{1/2}} \exp\left[-\frac{1}{2h^2} (X - X_j)^t \Sigma^{-1} (X - X_j) \right] \right\}, \quad (2)$$

where *n* is the number of training samples and *h* is the window width. The covariance matrix, Σ , is estimated from the *n* training samples and $h \propto n^{-1/d}$. The value of *h* is usually determined empirically. A large value of *h* means a large degree of smoothing and a small value of *h* means a small degree of smoothing. A rule of thumb states that for a small (large) number of training samples (*n*), window width should be large (small), and for a fixed *n*, the window width should be large (small) for large (small) number of features (*d*). When a large number of samples are available, the density estimated using Parzen window approach are very close to the true densities.

3.3. Decision strategy

We use the likelihood ratio $L = P(X^d | w_0)/P(X^d | w_1)$ to make the final decision for our two-class problem: Decide D_0 (person is an imposter) for high values of L; decide D_1 (person is genuine) for low values of L. If L is small, the data is more likely to come from class w_1 ; the likelihood ratio test rejects the null hypothesis for small values of the ratio. The Neyman–Pearson lemma states that this test is optimal, that is, among all the tests with a given significance level, α , the likelihood ratio test has the maximum power. For a specified α , λ is the smallest constant such that $P\{L \le \lambda\} \le \alpha$. The Type II error (β) is given by $P\{L > \lambda\}$.

4. Matching algorithms

We have developed four different fingerprint verification systems which can be broadly classified into two categories: (i) minutiae-based, and (ii) filter-based. The three minutiae-based and one filter-based algorithms are summarized in this section.

4.1. Minutiae-based fingerprint matching algorithms

In this type of matching algorithms, minutiae (fingerprint ridge bifurcations and endings) are used as features. Each feature is characterized by its location and the direction of the ridge on which it resides. For each of the three matchers considered here in this category, the minutiae are extracted using the same algorithm. The extraction algorithm has four main components (see Fig. 2): (i) orientation field estimation, (ii) ridge



Fig. 2. Flowchart of the minutia extraction algorithm and matching. Any of the three matching algorithms described in Sections 4.1.1–4.1.3 can be used to match the template minutiae set and the detected minutiae set.

detection, (iii) ridge thinning, and (iv) minutiae detection. The orientation field is estimated using the method in Ref. [22]. The second stage binarizes the fingerprint image by convolving the image with local filters oriented in the direction estimated in step (i). The ridges in the binary image are thinned using a standard thinning algorithm and minutiae are detected on the thinned ridges as those points which have either one or more than two neighbors. The minutiae features obtained from the two fingerprint images can be matched using one of the three matching algorithms briefly described below.

4.1.1. Hough transform-based matching (Algorithm Hough)

The fingerprint matching problem can be regarded as template matching [23]: given two sets of minutia features, compute their matching score. The two main steps of the algorithm are: (1) compute the transformation parameters δ_x , δ_y , θ , and *s* between the two images, where δ_x and δ_y are translations along *x*-and *y*-directions, respectively, θ is the rotation angle, and *s* is the scaling factor; (2) align two sets of minutia points with the estimated parameters and count the matched pairs within a bounding box; (3) repeat the previous two steps for the set of discretized allowed transformations. The transformation that results in the highest matching score is believed to be the correct one. The final matching score is scaled between 0 and 99. Details of the algorithm can be found in Ref. [23].

4.1.2. String distance-based matching (Algorithm String)

Each set of extracted minutia features is first converted into polar coordinates with respect to an anchor point. The two-dimensional (2D) minutia features are, therefore, reduced to a one-dimensional (1D) string by concatenating points in an increasing order of radial angel in polar coordinate. The string matching algorithm is applied to compute the edit distance between the two strings. The edit distance can be easily normalized and converted into a matching score. This algorithm [22] can be summarized as follows: (1) Rotation and translation are estimated by matching ridge segment (represented as planar curve) associated with each minutia in the input image with the ridge segment associated with each minutia in the template image. The rotation and translation that results in the maximum number of matched minutiae pairs within a bounding box is considered the correct transformation and the corresponding minutiae are labeled as anchor minutiae, \mathscr{A}_1 and \mathscr{A}_2 , respectively. (2) Convert each set of minutia into a 1D string using polar coordinates anchored at \mathcal{A}_1 and \mathcal{A}_2 , respectively; (3) Compute the edit distance between the two 1D strings. The matched pairs are retrieved based on the minimal edit distance between the two strings; (4) Output the normalized matching score which is the ratio of the number of matched-pairs and the number of minutiae points.

4.1.3. 2D dynamic programming-based matching (Algorithm Dynamic)

This matching algorithm is a generalization of the above-mentioned string algorithm. The transformation of a 2D pattern into a 1D pattern usually results in a loss of information. Chen and Jain [24] have shown that fingerprint matching using 2D dynamic time warping can be done as efficiently as 1D string editing while avoiding the above-mentioned problems with algorithm String. The 2D dynamic time warping algorithm can be characterized by the following steps: (1) Estimate the rotation between the two sets of minutia features as in Step 1 of algorithm String; (2) Align the two minutia sets using the estimated parameters from Step 1; (3) Compute the maximal matched minutia pairs of the two minutia sets using 2D dynamic programming technique. The intuitive interpretation of this step is to warp one set of minutia to align with the other so that the number of matched minutiae is maximized; (4) Output the normalized matching score which is based on only those minutiae that lie within the overlapping region. A penalty term is added to deal with unmatched minutia features.

4.2. Texture-based matching

The minutiae-based representation is widely used in fingerprint verification but does not utilize a significant component of the rich discriminatory information available in the ridge structures of the fingerprints. Local ridges cannot be completely characterized by minutiae. Further, minutiae-based matching has problems in efficiently matching two fingerprint images containing different numbers of unregistered minutiae points. The fingerprint image can be viewed as an oriented texture. Texture-based representation of fingerprint image overcomes some of the problems with minutiae-based representation and captures both the local and the global information in a fingerprint as a compact FingerCode [25].

4.2.1. Filterbank-based matching (Algorithm Filter)

The four mains steps in the filter-based feature extraction algorithm are (see Fig. 3): (i) determine a reference point and region of interest for the fingerprint image. The reference point is taken to be the center point in a fingerprint which is defined as the point of maximum curvature of the ridges in a fingerprint. The region of interest is a circular area around the reference point. The algorithm rejects the fingerprint images for which the reference point could not be established. (ii) tessellate the region of interest. The region of interest is divided into sectors and the gray values in each sector are normalized to a constant mean and variance. (iii) filter the region of interest in eight different directions using a bank of Gabor filters (eight directions are required to completely capture the local ridge characteristics in a fingerprint while only four directions are required to capture the global configuration). Filtering produces a set of eight filtered images. (iv) compute the average absolute deviation from the mean (AAD) of gray values in individual sectors in each filtered image. AAD value in each sector quantifies the underlying ridge structures and is defined as a feature. A feature vector, which we call FingerCode, is the collection of all the features (for every sector) in each filtered image. Thus, the feature elements capture the local information and the ordered enumeration of the tessellation captures the invariant global relationships among the local patterns. The representation is invariant to translation of the image. It is assumed that the fingerprint is captured in an upright position and the rotation invariance is achieved by storing 10 representations corresponding to the various rotations $(-45.0^{\circ}, -45^{\circ}, -33.75^{\circ}, -22.5^{\circ}, -11.25^{\circ}, 0^{\circ}, 11.25^{\circ},$ 22.5°, 33.75°, 45.0°) of the image. Euclidean distance is computed between the input representation and the 10 templates to generate 10 matching distances. Finally, the minimum of the 10 distances is computed and inverted to give a matching score. The matching score is scaled between 0 and 99 and can be regarded as a confidence value of the matcher.

5. Experimental results

Fingerprint images were collected in our laboratory from 167 subjects using an optical sensor manufactured by Digital Biometrics, Inc. (image size = $508 \times$ 480, resolution = 500 dpi). A single impression each of the right index, right middle, left index, and left middle



Fig. 3. Flowchart of the filterbank-based feature extraction and matching algorithm.

fingers for each subject was taken in that order. This process was then repeated to acquire a second impression. The fingerprint images were collected again from the same subjects after an interval of 6 weeks in a similar fashion. Thus, we have four impressions for each of the four fingers of a subject. This resulted in a total of 2672 (167 \times 4 \times 4) fingerprint images. We call this database MSU_DBI. A live feedback of the acquired image was provided and the subjects were guided in placing their fingers in the center of the sensor in an upright position. A total of 100 images (about 4% of the database) was removed from the MSU_DBI because the filter-based fingerprint matching algorithm rejected these images due to failure in locating the center or due to a poor quality of the images. We matched all the remaining 2572 fingerprint images with each other to obtain 3,306,306 ($2572 \times 2571/2$) matchings and called the matchings genuine only if the pair are different impressions of the same finger. Thus, we have a total of 3,298,834 (3,306,306–7472) imposter and 7472 genuine matchings per matcher from this database. For the multiple matcher combination, we randomly selected half the imposter matching scores and half the genuine matching scores for training and the remaining samples for test. This process was repeated 10 times to give 10 different training sets and 10 corresponding independent test sets. All performances will be reported in terms of ROC curves computed as an average from the 10 ROC curves corresponding to the 10 different training and test sets. For the multiple impression and multiple finger combinations, the same database of 3,298,834 imposter and 7472 genuine matchings computed using the *Dynamic* matcher was used.

The ROC curves computed from the test data for the four individual fingerprint matchers used in this study are shown in Fig. 4. The class separation statistic computed from the training data was 1.88, 1.87, 1.85 and 1.76 for the algorithms *Dynamic*, *String*, *Filter*, and



Fig. 4. Performance of individual fingerprint matchers.

Hough, respectively, and is found to be highly correlated to the matching performance on the independent test set. Fig. 4 shows that matcher *Filter* is better than the other three matchers at high FARs while it is worst at very low FARs. Matcher *Hough* is the worst at most operating points except at very low FARs. At an equal error rate of 3.5%, the matchers *Dynamic*, *String*, and *Filter* are equivalent while the matcher *Hough* has an equal error rate of about 6.4%.

In general, biometrics applications demand very low error rates. Small errors in estimation of the imposter and genuine distributions can significantly effect the performance of a system. Consider the empirical imposter density and a normal approximation to the imposter density for the algorithm Filter shown in Fig. 5(a). One would expect to get very accurate estimates of the parameters of a one-dimensional density from over 1.6 million data points. In fact, visually the normal approximation to the imposter density seems to fit the empirical density very well (see Fig. 5(a)). At equal error rate, using either the normal approximation or the nonparametric approximation of the imposter density give similar results. However, a significant decrease in performance is observed at low FARs when a normal approximation to the density is used in place of the nonparametric estimate of the density (see Fig. 5(b)). This is because the normal approximation to the imposter density has a heavier tail than the empirical density. To achieve the same low FAR, the system will operate at a higher threshold when the normal density is used than when the empirical density is used. The FRR, which is the area under the genuine density curve less than the threshold, increases significantly. So, we would like to stress that a parameterization of the density should be avoided.

Next, we combine the four available fingerprint matchers in pairs of two. It is well known in classifier combination studies that the independence of classifiers



Fig. 5. Normal approximation for the imposter distribution for the matcher *Filter*: (a) imposter and genuine distributions; (b) ROC curves. Visually, the normal approximation seems to be good, but causes significant decrease in the performance compared to nonparametric estimate.

plays an important role in performance improvement [26]. A plot of the scores in a 2D space from the training data for the String + Filter combination is shown in Fig. 6. The correlation coefficient, ρ , between the matching scores can be used as a measure of diversity between a pair of matchers [27]. A positive ρ is directly proportional to the measure of "dependence" between the scores from the two matchers. Table 1 lists the correlation coefficients for all possible pairing of the four available fingerprint matchers. It can be observed from this table that the minutiae-based fingerprint matchers have more dependence among themselves than with the filter-based fingerprint matcher. The ranking of combination by the amount of increase $(\Delta \text{ ROC})$ in performance with respect to the better of the two component matchers, listed in the last column in Table 1, is found to be coarsely related to ρ .





Fig. 6. Plot of joint scores from matchers String and Filter.

Table 1

Combining two fingerprint matchers. CS is the class separation statistic. CS and ρ are computed from the training data. Ranks by ROC and ranks by Δ ROC are computed from the independent test data

Combination	CS (rank)	Rank by ROC	ρ	Rank by Δ ROC
String + Filter	1.95 (1)	1	0.52	2
Dynamic + Filter	1.95 (1)	2	0.56	3
String + Dynamic	1.94 (3)	2	0.82	3
Hough + Dynamic	1.93 (4)	4	0.80	6
Hough + Filter	1.91 (4)	6	0.53	1
Hough + String	1.90 (6)	5	0.83	5

To combine two fingerprint matchers, we first estimate the 2D genuine and imposter densities from the training data. The 2D genuine density was computed using the Parzen density estimation method. The value of window width (h) was empirically determined to obtain a smooth density estimate and was set at 0.01. We used the same value of h for all the two-matcher combinations. As a comparison, the genuine density estimates obtained from the normalized histograms were extremely peaky due to unavailability of sufficient data (only about 3780 genuine matching scores were available in the training set to estimate a 2D distribution in 10,000 (100×100) bins). However, for estimation of the 2D imposter distribution, over 1.6 million matching scores were available. Hence, we estimated the 2D imposter distribution by computing a normalized histogram using the following formula:

$$p(X^{d}|w_{0}) = \frac{1}{n} \sum_{j=1}^{n} \delta(X, X_{j}),$$
(3)

where δ is the delta function that equals 1 if the raw matching score vectors *X* and *X_j* are equal, 0 otherwise. Here *n* is the number of imposter matchings from the



Fig. 7. Two-dimensional density estimates for the genuine and imposter classes for *String*+*Filter* combination. Genuine density was estimated using Parzen window (h = 0.01) estimator and the imposter density was estimated using normalized histograms.



Fig. 8. ROC curves for all possible two-matcher combinations.

training data. The computation time for Parzen window density estimate depends on n and so, it is considerably larger than the normalized histogram method for large n. The smooth estimates of the two-dimensional genuine and imposter densities thus computed for *String* + *Filter* combination are shown in Fig. 7. The class separation statistic for all pairs of matcher combination is shown in the second column of Table 1; the number in parenthesis is the predicted ranking of the combination performance based on CS. The actual ranking of performance obtained from the independent test set is listed in the third column marked ROC (see Fig. 8 for ROC curves). As can be seen, the predicted ranking is very close to the actual rankings on independent test data.



Fig. 9. Comparison of the proposed combination scheme with the sum and the product rules for the String + Filter combination.

The following observations can be made from the two-matcher combinations:

- Classifier combination improvement is directly related to the "independence" (lower values of ρ) of the classifiers.
- Combining two weak classifiers results in a large performance improvement.
- Combining two strong classifiers results in a small performance improvement.
- The two individually best classifier do not form the best pair.

The proposed combination scheme either outperforms or matches the performance of the sum rule and outperforms the product rule in all the two- three- and four-matcher combinations. However, we provide illustrations of the comparison in two-matcher combinations as it is easier to visualize the decision boundaries in two dimensions. We choose the String + Filter combination which involves a strong and a weak classifier. The results of this combination and a comparison with the sum and the product rules is shown in Fig. 9. By assuming that the errors in estimation of a posteriori probabilities (matching scores) are very small, Kittler et al. [2] mathematically showed that the sum rule is less sensitive to these errors than the product rule. In our case, instead of considering the scores from two classifiers as estimates of a posteriori probability, we consider them as features in a separate classification problem. In such a case, the decision boundaries corresponding to the sum and the product rules can be drawn and visualized. In Fig. 6, the decision boundaries corresponding to three different thresholds are shown for the sum and the product rules by solid and dotted lines, respectively. The product rule has a strong bias for low values of the two component classifier outputs. This is undesirable in most practical



Fig. 10. The performance of the best individual matcher *Dynamic* is compared with the various combinations. The *String* + *Filter* is the best two-matcher combination and *String* + *Dynamic* + *Filter* is the best overall combination. Note that addition of the classifier *Hough* to the combination *String* + *Filter* results in a degradation of the performance.

situations and the product rule is not expected to perform well in most cases. The sum rule decision boundary is very restrictive (a line at 135° slope) and sum rule performs well only when combining two classifiers of equal strength (two weak or two strong classifiers). When a weak and a strong classifier is combined, the decision boundary should bend towards the axis of the strong classifier. Weighted sum rule can adapt the slope of its decision boundary but the decision boundary is still linear. The proposed technique can produce a decision boundary that is non-linear and is expected to perform better than the sum and the product rules. However, the disadvantage of the proposed technique is that it requires sufficient training data to obtain reasonable estimates of the densities while the sum rule is a fixed rule and does not require any training. Weighted sum rule can perform better than the sum rule but it is difficult to determine the weights. In summary, the proposed scheme performs the best, followed by the sum rule and the product rule performs the worst when combining a weak and a strong classifier (Fig. 9).

Finally, we combine the matchers in groups of three and then combine all the four matchers together. From the tests conducted on the independent data set, we make following observations (see Fig. 10).

- Adding a classifier may actually degrade the performance of classifier combination. This degradation in performance is a consequence of lack of independent information provided by the classifier being added and finite size of the training and test database.
- Classifier selection based on a "goodness" statistic is a promising approach.

• Performance of combination is significantly better than the best individual matcher.

Among all the possible subsets of the four fingerprint matchers, the class separation statistic is maximum for String + Dvnamic + Filter combination. Hence, our feature selection scheme selects this subset for the final combination and rejects the matcher Hough. This is consistent with the nature of the Hough algorithm, which is basically the linear pairing step in algorithms String and Dynamic, without the capability of dealing with elastic distortions. Therefore, Hough does not provide "independent" information with respect to String and Dynamic. Fig. 11 shows the small overlap in the scores from the genuine and the imposter classes for the best combination involving fingerprint matchers String, Dynamic, and Filter. The performance of the various matcher combinations on an independent test supports the prediction that String + Dynamic + Filter is the best combination.

Our final multi-modal biometric system design is depicted in Fig. 12. The performance of the combined system is more than 3% better than the best individual matcher at low FARs (see Table 2). The equal error rate is



Fig. 11. Matching scores for the best combination involving *String, Dynamic*, and *Filter* matchers. Visually, one can see the small overlap between the genuine (\bigcirc) and the imposter (*) classes. The class separation statistic is 1.97 for the 3D genuine and imposter densities estimated from these scores.



Fig. 12. Proposed architecture of multi-modal biometrics system based on sever fingerprint matchers.

Table 2

Comparison of the performance of the best matcher combination with the best individual matcher. GAR refers to the genuine acceptance rate that is plotted on the ordinate of the ROC curves

FAR GAR Dynamic (%)		GAR String + Dynamic + Filter (%)	GAR Improvement (%)	
	Mean (Var)	Mean (Var)		
1.00%	95.53 (0.08)	98.23 (0.02)	2.70	
0.10%	92.96 (0.05)	96.16 (0.04)	3.20	
0.01%	90.25 (0.04)	93.72 (0.05)	3.47	

Table	3						
Equal	error	rate	improvement	due	to	combination	

	String	Dynamic	Filter	Combination
Equal error rate (%)	3.9	3.5	3.5	1.4

more than 2% better than the best individual matcher (see Table 3). The matcher combination takes about 0.02 s on an Sun Ultra 1 in the test phase. In an authentication system, this increase in time will have almost no effect on the verification time and the overall matching time is still bounded by the slowest individual matcher.

The performance improvement due to combination of two impressions of the same finger and the combination of two different fingers of the same person using the proposed strategy is shown in Fig. 13(a) and (b), respectively. The matcher *Dynamic* was used. The correlation coefficient between the two scores from two different impressions of the same finger is 0.42 and between two different fingers of the same person is 0.68 and is directly related to the improvement in the performance of combination. The CS for individual impressions is 1.84 and 1.87, respectively, and for the combination is 1.95. The CS for individual fingers is 1.87 and 1.86, respectively, and for the combination is 1.98. Combination of two impressions of the same finger or two fingers of the same person using the proposed combination strategy is extremely fast. Therefore, the overall verification time is same as the individual matcher *Dynamic*.

6. Summary and conclusions

We have presented a scheme for combining multiple matchers (classifiers) at decision level in an optimal fashion. Our design emphasis is on classifier selection before arriving at the final combination. It was shown that one of the fingerprint matchers in the given pool of matchers is redundant and no performance improvement is achieved by utilizing this matcher. This matcher was identified and rejected by the matcher selection scheme. In case of a larger number of classifiers and relatively small training data, a classifier may actually degrade the performance when combined with other classifiers, and hence classifier selection is essential. We demonstrate that our combination scheme improves the performance of a fingerprint verification system by more than 3%. We also show that combining multiple instances of the same biometric or multiple units of the same biometric characteristics is a viable way to improve the verification system performance. We observe that independence among various classifiers is directly related to the improvement in performance of the combination.



Fig. 13. (a) Combining two impressions of the same finger, and (b) combining two fingers of the same person.

Acknowledgements

We would like to thank Dr. R.P.W. Duin and M. Skurichina of Delft University of Technology and Dr. T. K. Ho of Bell Laboratories for their many useful suggestions.

References

- E.S. Bigün, B. Duc, S. Fisher, Expert conciliation for multi modal person authentication systems by Bayesian statistics, Proceedings of the First International Conference on Audio- and Video-based Biometric Person Authentication, Crans-Montana, Switzerland, 1997, pp. 291–300.
- [2] J. Kittler, M. Hatef, R.P.W. Duin, J. Matas, On combining classifiers, IEEE Trans. Pattern Anal. Mach. Intell. 20 (3) (1998) 226–239.
- [3] T.K. Ho, J.J. Hull, S.N. Srihari, Decision combination in multiple classifier systems, IEEE Trans. Pattern Anal. Mach. Intell. 16 (1) (1994) 66–75.
- [4] P. Sinha, J. Mao, Combining multiple OCRs for optimizing word recognition, Proceedings of the 14th International Conference on Pattern Recognition 1, Brisbane, 1998, pp. 436–438.
- [5] L. Lam, C.Y. Suen, A theoretical analysis of the application of majority voting to pattern recognition, Proceedings of the 12th International Conference on Pattern Recognition, Jerusalem, 1994, pp. 418–420.
- [6] L. Lam, C.Y. Suen, Optimal combination of pattern classifiers, Pattern Recognition Lett. 16 (1995) 945–954.
- [7] R. Cappelli, D. Maio, D. Maltoni, Combining fingerprint classifiers, First International Workshop on Multiple Classifier Systems (MCS2000), Cagliari, 2000, pp. 351–361.
- [8] A.K. Jain, R.P.W. Duin, J. Mao, Statistical pattern recognition: a review, IEEE Trans. Pattern Anal. Mach. Intell. 22 (1) (2000) 4–37.
- [9] L. Hong, A.K. Jain, S. Pankanti, Can multibiometrics improve performance? Proceedings AutoID'99, Summit, NJ, October 1999, pp. 59–64.
- [10] A.K. Jain, L. Hong, Y. Kulkarni, A multimodal biometric system using fingerprint, face, and speech, Proceedings of the Second International Conference on Audio- and Video-based Biometric Person Authentication, Washington DC, 1999, pp. 182–187.
- [11] A.K. Jain, S. Prabhakar, A. Ross, Fingerprint matching: data acquisition and performance evaluation, MSU Technical Report TR99-14, 1999.
- [12] A.K. Jain, S. Prabhakar, S. Chen, Combining multiple matchers for a high security fingerprint verification system, Pattern Recognition Lett. 20 (11–13) (1999) 1371–1379.
- [13] J.D. Elashoff, R.M. Elashoff, G.E. Goldman, On the choice of variables in classification problems with dichotomous variables, Biometrika 54 (1967) 668–670.

- [14] G.T. Toussaint, Note on optimal selection of independent binary-valued features for pattern recognition, IEEE Trans. Inform. Theory IT-17 (1971) 618.
- [15] T.M. Cover, The best two independent measurements are not the two best, IEEE Trans. Systems, Man, Cybern. SMC-4 (1) (1974) 116–117.
- [16] G.S. Fang, A note on optimal selection of independent observables, IEEE Trans. Systems, Man, Cybern. SMC-9 (5) (1979) 309–311.
- [17] T.M. Cover, On the possible ordering in the measurement selection problem, IEEE Trans. Systems, Man, Cybern. SMC-7 (9) (1977) 657–661.
- [18] A.K. Jain, B. Chandrasekaran, Dimensionality and sample size considerations in pattern recognition practice, in: P.R. Krishnaiah, L.N. Kanal (Eds.), Handbook of Statistics, Vol. II, North-Holland, Amsterdam, 1982, pp. 835–855.
- [19] S. Raudys, A.K. Jain, Small sample size effects in statistical pattern recognition: recommendations for practitioners, IEEE Trans. Pattern Anal. Mach. Intell. 13 (3) (1991) 252–264.
- [20] I.-S. Oh, J.-S. Lee, C.Y. Suen, Analysis of class separation and combination of class-dependent features for handwriting recognition, IEEE Trans. Pattern Anal. Mach. Intell. 21 (10) (1999) 1089–1094.
- [21] R.O. Duda, P.E. Hart, D.G. Stork, Pattern Classification, 2nd Edition, Wiley, New York, 2000.
- [22] A.K. Jain, L. Hong, S. Pankanti, R. Bolle, An identity authentication system using fingerprints, Proc. IEEE 85 (9) (1997) 1365–1388.
- [23] N.K. Ratha, K. Karu, S. Chen, A.K. Jain, Real-time matching system for large fingerprint database, IEEE Trans. Pattern Anal. Mach. Intell. 18 (8) (1996) 799–813.
- [24] S. Chen, A.K. Jain, A fingerprint matching algorithm using dynamic programming, Technical Report, Department of Computer Science and Engineering, Michigan State University.
- [25] A.K. Jain, S. Prabhakar, L. Hong, S. Pankanti, Filterbank-based fingerprint matching, IEEE Trans. Image Process. 9 (5) (2000) 846–859.
- [26] L.I. Kuncheva, C.J. Whitaker, C.A. Shipp, R.P.W. Duin, Is independence good for combining classifiers, IEEE International Conference on Pattern Recognition (ICPR), Barcelona, Spain, 2 (2000) 168–171.
- [27] L.I. Kuncheva, C.J. Whitaker, Measures of diversity in classifier ensembles, Mach. Learning, 2000, submitted.
- [28] L. Xu, A. Krzyzak, C.Y. Suen, Methods for combining multiple classifiers and their applications to handwriting recognition, IEEE Trans. Systems, Man, Cybern. 22 (3) (1992) 418–435.
- [29] A.K. Jain, S. Prabhakar, L. Hong, A multichannel approach to fingerprint classification, IEEE Trans. Pattern Anal. Mach. Intell. 21 (4) (1999) 348–359.
- [30] A.K. Jain, D. Zongker, Feature selection: evaluation, application and small sample performance, IEEE Trans. Pattern Anal. Mach. Intell. 19 (2) (1997) 153–158.
- [31] A.K. Jain, R.M. Bolle, S. Pankanti (Eds.), Biometrics: Personal Identification in a Network Society, Kluwer Academic Publishers, MA, 1999.

About the Author—SALIL PRABHAKAR was born in Pilani, Rajasthan, India, in 1974. He received his B.Tech. degree in Computer Science and Engineering from Institute of Technology, Banaras Hindu University, Varanasi, India, in 1996. During 1996–1997 he worked with Tata Information Systems Ltd. (now IBM Global Services India Pvt. Ltd.), Bangalore, India, as a software engineer. He earned his Ph.D. degree from the Department of Computer Science and Engineering, Michigan State University, East Lansing, MI 48823, in 2001. He is currently with the Algorithms Research Group, DigitalPersona Inc., Redwood City, CA 94063. His research interests include pattern recognition, image processing, machine learning, biometrics, data mining, and multimedia applications.

About the Author—ANIL JAIN is a University Distinguished Professor in the Department of Computer Science and Engineering at Michigan State University. His research interests include statistical pattern recognition, Markov random fields, texture analysis, neural networks, document image analysis, fingerprint matching and 3D object recognition. He received the best paper awards in 1987 and 1991 and certificates for outstanding contributions in 1976, 1979, 1992, and 1997 from the Pattern Recognition Society. He also received the 1996 IEEE Trans. Neural Networks Outstanding Paper Award. He was the Editor-in-Chief of the IEEE Trans. on Pattern Analysis and Machine Intelligence (1990–94). He is the co-author of Algorithms for Clustering Data, Prentice-Hall, 1988, has edited the book Real-Time Object Measurement and Classification, Springer-Verlag, 1988, and co-edited the books, Analysis and Interpretation of Range Images, Springer-Verlag, 1989, Markov Random Fields, Academic Press, 1992, Artificial Neural Networks and Pattern Recognition, Elsevier, 1993, 3D Object Recognition, Elsevier, 1993, and BIOMETRICS: Personal Identification in Networked Society, Kluwer in 1999. He is a Fellow of the IEEE and IAPR. He received a Fulbright research award in 1998.