ELSEVIER

Pattern Recognition Letters 20 (1999) 1371–1379

Pattern Recognition Letters

www.elsevier.nl/locate/patrec

Combining multiple matchers for a high security fingerprint verification system

Anil K. Jain *, Salil Prabhakar, Shaoyun Chen

Department of Computer Science and Engineering, Michigan State University, 3115 Engineering Building, East Lansing, MI 48824-1226, USA

Abstract

Integration of various fingerprint matching algorithms is a viable method to improve the performance of a fingerprint verification system. Different fingerprint matching algorithms are often based on different representations of the input fingerprints and hence complement each other. We use the logistic transform to integrate the output scores from three different fingerprint matching algorithms. Experiments conducted on a large fingerprint database confirm the effectiveness of the proposed integration scheme.

Keywords: Fingerprint matching; Verification; Combination of matchers; Logistic transform

1. Introduction

A fingerprint is the smoothly flowing pattern formed by ridges and valleys on the tip of a finger. Identification of people based on their fingerprints has a long history and it is the most prevalent technology in automatic verification and identification systems. Fingerprints are matched based on the local ridge features, called minutiae, i.e., ridge endings and bifurcations (see Fig. 1). An automatic authentication system first locates these features using a minutiae extraction algorithm (see Fig. 2) and then matches them with the template of the claimed identity stored in a database. With the recent availability of solid-state sensors for fingerprint capture, it is expected that fingerprints will be increasingly used in civilian applications such as Internet transactions, ATM and border control. The emerging applications based on automatic fingerprint identification demand stringent performance requirements. For example, a high security access control system requires an extremely low false accept rate ($< 0.01\%$). Because of the inexact nature of image acquisition and vulnerability of feature extractor to noise and distortion in the fingerprint images, it is very difficult to reduce the false reject rate of a matcher when the specified false accept rate is extremely low. Many researchers have combined multiple biometrics (e.g., fingerprint and face) to improve the overall system performance (Bigün et al., 1997; Kittler et al., 1998; Jain et al., 1998, 1999a), but this involves the additional cost of sensors and inconvenience to the user in providing multiple cues.

A number of fingerprint verification systems have been developed and tested on large databases

* Corresponding author. Tel.: +1-517-353-6484; fax: +1-517-432-1061.

E-mail addresses: jain@cse.msu.edu (A.K. Jain), prabhaka@cse.msu.edu (S. Prabhakar), chenshao@cse.msu.edu (S. Chen)

0167-8655/99/$ - see front matter
PII: S 0 1 6 7 - 8 6 5 5 (9 9) 0 0 1 0 8 - 7

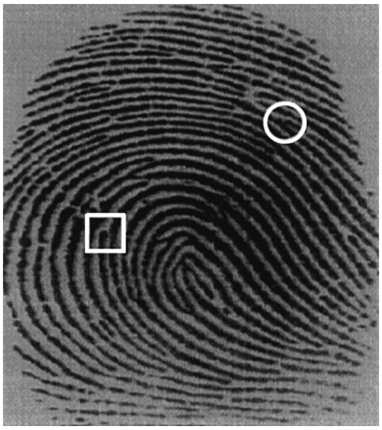

Fig. 1. A fingerprint image and minutiae features (a ridge ending (□) and a bifurcation (○)).

but none of them has been able to meet the rigid performance requirement in high security applications. Each fingerprint verification system uses different feature extraction and matching algorithms to generate a matching score which is used for authentication. It has been reported in various classifier design studies in pattern recognition literature that different classifiers often misclassify different patterns. This suggests that different classifiers offer complementary information about the classification task. A combination scheme which harnesses various information sources is likely to improve the overall system performance. The outputs of various classifiers can be combined to obtain a decision which is more accurate than the decisions made by any one of the individual classifiers. A combination of multiple classifiers to improve the recognition rate has been a common practice in building successful recognition systems (Kittler et al., 1998; Ho et al., 1994; Sinha and Mao, 1998; Lam and Suen, 1994, 1995). Approaches for combining classifier outputs include the Bayesian method (Bigün et al., 1997), the sum and the product rules (Kittler et al., 1998), the Borda count (Ho et al., 1994), logistic regression (Ho et al., 1994) to assign weights to the ranks produced by each classifier, the majority vote (Lam and Suen, 1994), the behavior-knowledge space method (Lam and Suen, 1995) and Dempster–Shafer theory for weighted voting (Xu et al., 1992).

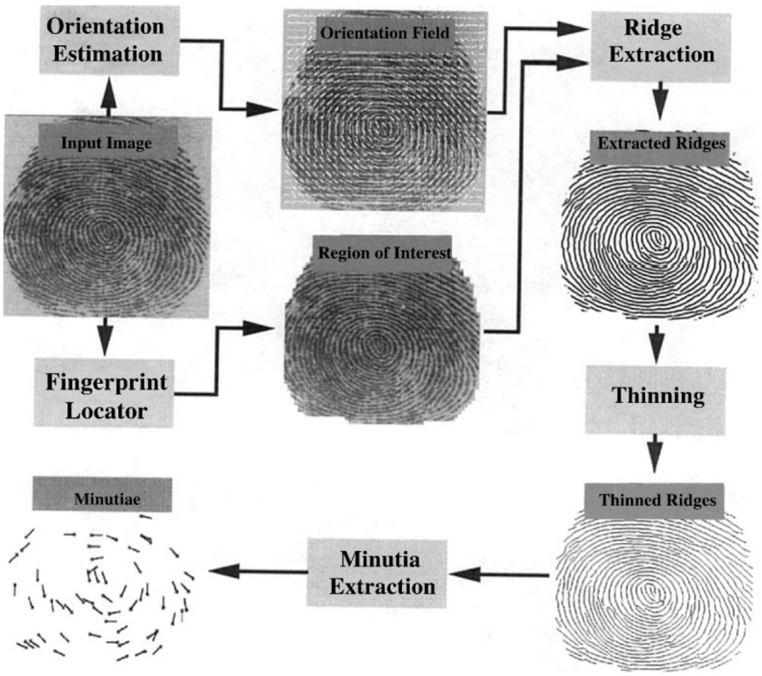

Fig. 2. Flowchart of the minutia extraction algorithm.

Logistic regression has been widely used in the analysis of binary data. It can be easily shown that when the class-conditional densities are Gaussian with the same covariance matrix, the a posteriori probability is a logistic function. Logistic regression has found its application in integration of multiple classifiers. Let $\pi(\boldsymbol{x}) = P(Y=1|\boldsymbol{x})$, where Y is the binary output response and $\boldsymbol{x} = (x_1, x_2)$, with x_1 and x_2 being the rank scores from the two classifiers. Then the *log-odds*, or $\mathrm{logit}(\pi(\boldsymbol{x})) = \log \pi(\boldsymbol{x})/(1-\pi(\boldsymbol{x}))$, is approximated by a linear regression, $\log \pi(\boldsymbol{x})/(1-\pi(\boldsymbol{x})) = \alpha + \beta x_1 + \gamma x_2$. Logistic regression assigns weights to the rank output of each classifier. Sinha and Mao (1998) used logistic function to transform the output scores of two classifiers into a single overall score. The parameters were learned with a multilayer perceptron using the gradient descent method.

In our experiments, logistic regression outperforms the Neyman–Pearson rule (Jain et al., 1999a). In the integration of multiple fingerprint matching algorithms, the two types of errors (False Acceptance Rate (FAR) and False Rejection Rate (FRR)) are not equal. The integration involves minimizing the FRR for a specified level of FAR and hence the sum and the product rules (Kittler et al., 1998) are not directly applicable. In this paper we use logistic regression to integrate multiple fingerprint matching algorithms. The three parameters α, β and γ are tuned so that the FRR is minimized for a given FAR. This method is different from the approach of Ho et al. (1994) and Sinha and Mao (1998). Ho et al. (1994) applied logistic regression to $\hat{\pi}(\boldsymbol{x})$, the estimate of $\pi(\boldsymbol{x})$: $\mathrm{logit}\{\hat{\pi}(\boldsymbol{x})\} = \alpha + \beta x_1 + \gamma x_2$, while Sinha and Mao (1998) used a known analytical cost function. Our optimization problem directly links the three unknown parameters α, β and γ to FAR and FRR and the function to be optimized has an unknown analytical form.

The rest of the paper is organized as follows. Section 2 presents our integration strategy and the problem formulation. Section 3 gives a brief description of the three fingerprint matching algorithms to be integrated. The integration algorithm is given in Section 4. The experimental results are reported in Section 5. Finally, Section 6 concludes the paper.

2. Integration strategy

Let $I_i(x_i)$ and $G_i(x_i)$ be the imposter and genuine distributions of the ith matcher, $i = 1, 2$. We use logistic function to map the output scores x_1 and x_2 of these two matching algorithms into a single overall score x. The integration scheme is shown in Fig. 3. The logistic transform can be expressed as

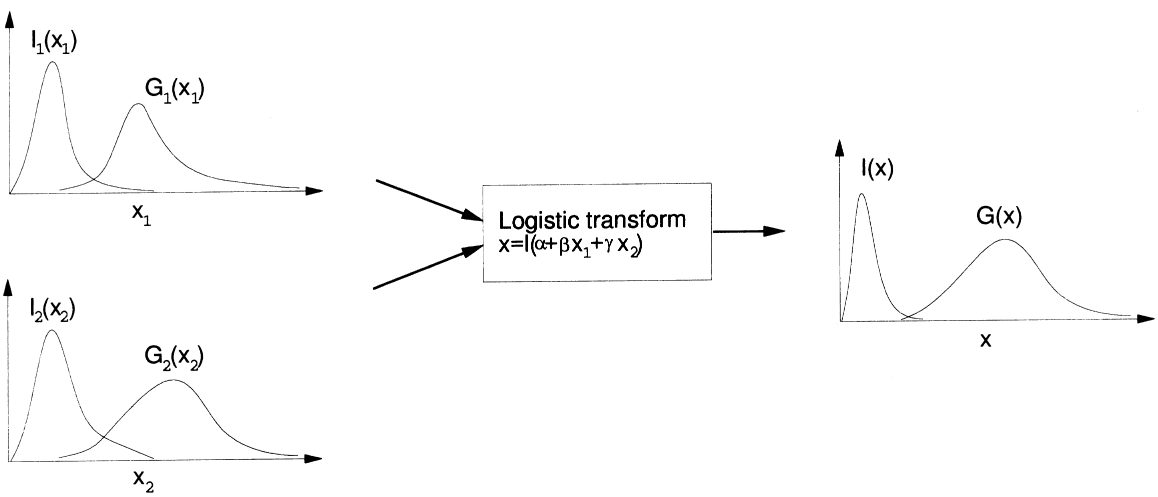

Fig. 3. Integration of two fingerprint matching algorithms using a logistic transform with tunable parameters α, β and γ.

$$x = l(\alpha + \beta x_1 + \gamma x_2) = \frac{\exp(\alpha + \beta x_1 + \gamma x_2)}{1 + \exp(\alpha + \beta x_1 + \gamma x_2)}, \quad (1)$$

where α, β and γ are the three parameters. The objective of the integration is to estimate the parameters α, β and γ such that the FRR is minimized for a specified level of FAR.

Assuming that x_1 and x_2 are independent, the joint imposter and genuine distributions $I(x_1, x_2)$ and $G(x_1, x_2)$, respectively, can be represented as

$$I(x_1, x_2) = \iint I_1(x_1) I_2(x_2)\, \mathrm{d}x_1\, \mathrm{d}x_2 \quad (2)$$

and

$$G(x_1, x_2) = \iint G_1(x_1) G_2(x_2)\, \mathrm{d}x_1\, \mathrm{d}x_2. \quad (3)$$

Therefore, the new probability distribution functions $I(x)$ and $G(x)$ of imposter and genuine individuals, respectively, after logistic transform, can be written as

$$I(x) = \iint I_1(x_1) I_2(x_2) \delta(\alpha + \beta x_1 + \gamma x_2 - l^{-1}(x))\, \mathrm{d}x_1\, \mathrm{d}x_2 \quad (4)$$

and

$$G(x) = \iint G_1(x_1) G_2(x_2) \delta(\alpha + \beta x_1 + \gamma x_2 - l^{-1}(x))\, \mathrm{d}x_1\, \mathrm{d}x_2, \quad (5)$$

where $\delta(\cdot)$ is the delta function. In other words, $I(x)$ and $G(x)$ are line integrals of $I(x_1, x_2)$ and $G(x_1, x_2)$, respectively, along the line $\alpha + \beta x_1 + \gamma x_2 = l^{-1}(x)$ on the (x_1, x_2) plane. The FRR, p_{frr}, for a given α, β, γ and FAR, t_{far}, is

$$p_{\mathrm{frr}}(\alpha, \beta, \gamma, t_{\mathrm{far}}) = p_{\mathrm{frr}}(t) = \int_{-\infty}^{t} G(x)\, \mathrm{d}x, \quad (6)$$

where $t = \arg_x \inf_x \{p_{\mathrm{far}}(x) \geqslant t_{\mathrm{far}}\} = \arg_x \inf_x \left\{\int_x^{+\infty} I(x)\, \mathrm{d}x \geqslant t_{\mathrm{far}}\right\}$, and $p_{\mathrm{frr}}(t)$ and $p_{\mathrm{far}}(t)$ are the FRR and FAR levels at the threshold t.

The integration of two fingerprint matching algorithms can be formulated as follows. For specified FAR levels $t_{\mathrm{far}}^{(i)}$, $i = 1, 2, \ldots, L$, compute the set of parameters $(\alpha_i, \beta_i, \gamma_i, t_i)$ which satisfy the following optimization criterion:

$$\{\alpha_i, \beta_i, \gamma_i\} = \arg \min_{\alpha,\beta,\gamma} \left\{p_{\mathrm{frr}}(\alpha, \beta, \gamma, t_{\mathrm{far}}^{(i)})\right\} \quad (7)$$

and

$$t_i = \arg_x \inf_x \left\{p_{\mathrm{far}}(x) \geqslant t_{\mathrm{far}}^{(i)}\right\} = \arg_x \inf_x \left\{\int_x^{+\infty} I(x)\, \mathrm{d}x \geqslant t_{\mathrm{far}}^{(i)}\right\}. \quad (8)$$

The minimization criterion estimates parameters $(\alpha_i, \beta_i, \gamma_i)$ such that the FRR is minimized at each given FAR level. Since we do not know the analytical form of $I(x)$ and $G(x)$, it is not possible to solve the minimization problem analytically. However, the minimization of Eq. (7) can be solved by efficient numerical algorithms.

3. Fingerprint matching algorithms

The features (minutiae) are extracted from a fingerprint image using a minutiae extraction algorithm. Each feature is characterized by its location and the direction of the ridge on which it resides. The features obtained from two fingerprint images can be matched using one of the three matching algorithms described in this section to yield a matching score. A brief description of the three algorithms used in our integration is given below.

3.1. Hough transform based matching (Algorithm A)

Hough transform and its variant, generalized Hough transform, are often used in line or curve detection. The equivalence between Hough transform and template matching was first stated by Stockman and Agrawala (1977). The fingerprint matching problem can be regarded as template matching (Ratha et al., 1996): 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, δ_θ and s, where δ_x and δ_y are translations along x- and y-directions, respectively, θ is the rotation angle, and s the scaling factor; (2) Align two sets of minutia points with the estimated parameters and count the matched pairs within a bounding box.

3.2. String distance based matching (Algorithm B)

Dynamic programming has been commonly used in speech processing to deal with the time warping problem. Similar ideas can be used to solve the elastic distortion problem in fingerprint matching (Jain et al., 1997). Each set of extracted minutia features is first converted into polar coordinates with respect to an anchor point. The 2D minutia features are, therefore, reduced to a 1D string. The string matching algorithm is applied to compute the edit distance between two strings. The edit distance can be easily normalized and converted into a matching score. This algorithm can be summarized as follows: (1) Estimate the rotation θ between the two sets of minutia features and the anchor minutia of each set A_1 and A_2. Rotation is estimated through a least-squares fitting. The values of parameters θ, A_1 and A_2 which result in the maximum number of matched minutia pairs within a bounding box are chosen as the best estimates; (2) Convert each set of minutia into a 1D string using polar coordinates anchored at A_1 and 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 as

$$\left\{\frac{\text{No. of matched-pairs}}{O_1+1}\times\frac{\text{No. of matched-pairs}}{O_2+1}\right\},$$

where O_1 and O_2 are the cardinalities of the two minutia sets, respectively.

3.3. 2D Dynamic programming based matching (Algorithm C)

A fingerprint and its extracted minutia features are 2D patterns. The transformation of a 2D pattern into a 1D pattern usually results in a loss of information. For example, if there is more than one minutia along the same direction, then usually only one minutia is retained in the transformation. Furthermore, the transformation of 2D minutia into 1D pattern causes another problem: due to elastic distortion, some minutiae could change their placement orders in the template minutia and the input minutia. This will create a problem in subsequent string editing operation which usually restricts the warping function so that the order of string components is preserved to ensure a good matching performance.

Chen and Jain (1998) 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 B. 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 B; (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 as

$$\mathrm{NMP}\times\left\{\frac{\mathrm{NMP}-0.20(O_1-\mathrm{NMP})}{O_1+1}\right\}\times\left\{\frac{\mathrm{NMP}-0.20(O_2-\mathrm{NMP})}{O_2+1}\right\}, \tag{9}$$

where NMP is the number of matched-pairs and O_1 and O_2 are the number of minutia features within the overlapped region of the two minutia sets.

Algorithm C uses a different normalized score than algorithm B because the matching score should be based on the minutiae that lie within the overlapping region, rather than the entire set of minutiae. A penalty term is added to deal with unmatched minutia features. Furthermore, the normalized score is the product of matched pairs and the goodness of match between the two minutia sets.

4. Integration algorithm

Given the discrete probability distribution functions of $I_1(x_1)$, and $G_1(x_1)$, $\{x_1 = 0, 1, \ldots, N-1\}$, and $I_2(x_2)$ and $G_2(x_2)$, $\{x_2 = 0, 1, \ldots, N-1\}$, the following algorithm computes the set of parameters $(\alpha_i, \beta_i, \gamma, t_i)$ for a set of specified FARs, $t_{\text{far}}^{(i)}$, $i = 1, 2, \ldots, L$.

for $i = 1$ to L
{
1. Compute $I(x)$ and $G(x)$
   for ($x = 0$; $x < N$; x++) $I[x] = G[x] = 0$;
   for ($x_1 = 0$; $x_1 < N$; x_1++)
   for ($x_2 = 0$; $x_2 < N$; x_2++)
   {
      $x = l(\alpha_i + \beta_i x_1 + \gamma_i x_2)$;
      $I[x] += I_1[x_1] \cdot I_2[x_2]$;
      $G[x] += G_1[x_1] \cdot G_2[x_2]$;
   }
   Normalize $I(x)$ and $G(x)$ so that $\sum_{k=0}^{N-1} I[k] = 1$ and $\sum_{k=0}^{N-1} G[k] = 1$.
2. Apply Brent's algorithm to obtain α_i, β_i, γ_i and t_i by solving the following minimization problem:
   $\{\alpha_i, \beta_i, \gamma_i\} = \arg\min_{\alpha,\beta,\gamma} \{p_{\text{frr}}(\alpha, \beta, \gamma, t_{\text{far}}^{(i)})\}$
   and
   $t_i = \arg_x \inf_x \left\{ \sum_{k=x}^{N-1} I[k] \geqslant t_{\text{far}}^{(i)} \right\}$
   which is the discrete version of the optimization functions in Eqs. (7) and (8),
   $p_{\text{frr}}(\alpha_i, \beta_i, \gamma_i, t_{\text{far}}^{(i)}) = \sum_{k=0}^{t_i} G[x]$.
}

Step 2 is computed using Brent's algorithm (Brent, 1973; Press et al., 1992) for minimization without the use of derivatives. Brent's algorithm is suited for minimization of a function f with a relatively small number of parameters (typically less than 10) and when the evaluation of f' is difficult or expensive. Brent's algorithm is well suited for our application here as the minimization function p_{frr} has only four parameters and its derivative is not available. Brent's algorithm usually needs a small number of function evaluations and converges in two to three steps. However, similar to other gradient descent methods, Brent's algorithm needs an initial guess of the solution and an initial searching direction. While there is no guarantee of convergence to a global minimum, the algorithm converges to good solutions after a few trials with different initial guesses of the solution. When we used $N = 200$, the CPU time to compute ten sets of parameters $(\alpha_i, \beta_i, \gamma_i, t_i)$, $i = 0, 1, \ldots, 9$, took about 1.5 minutes on a SUN Ultra-10. When more than one set of parameters needs to be computed and two consecutive specified FARs are close to each other, the solution of $(\alpha_i, \beta_i, \gamma_i)$ can be used as the initial guess for estimating $(\alpha_{i+1}, \beta_{i+1}, \gamma_{i+1})$.

5. Experimental results

Fingerprint images were collected in our laboratory from 167 subjects. The images were captured 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 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 six 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 partitioned the dataset into two subsets: the first subset consists of fingerprints from the first 83 subjects, while the second consists of fingerprints from the remaining 84 subjects. We used the first subset for training and the second subset for testing.

A Receiver Operating Characteristic (ROC) curve is a plot of genuine acceptance rate (1-FRR) against FAR for all possible system operating points (i.e., matching score threshold) and measures the overall performance of the system. Each point on the curve corresponds to a particular decision threshold. In the ideal case, both the error rates, i.e., FAR and FRR, should be zero and the genuine distribution and imposter distribution should be disjoint. In such a case, the "ideal" ROC curve is a step function at the zero false acceptance rate. On the other extreme, if the genuine and imposter distributions are exactly the same, then the ROC is a line segment with a slope of 45° with an end point at zero false acceptance rate. In practice, the ROC curve shape is in between these two extremes.

The integration results of all three different pairwise combinations of matching algorithms are reported as ROC curves in Figs. 4–6. In these figures, the dash-dot and the dash curves indicate the performance of two individual algorithms, while the solid curve indicates the performance after integration. From Figs. 4 and 5, we can see that the performance improvement by integrating algorithm A with either B or C is not significant. This is consistent with the nature of algorithm A, which is basically the linear pairing step in algorithms B and C, with very little capability of

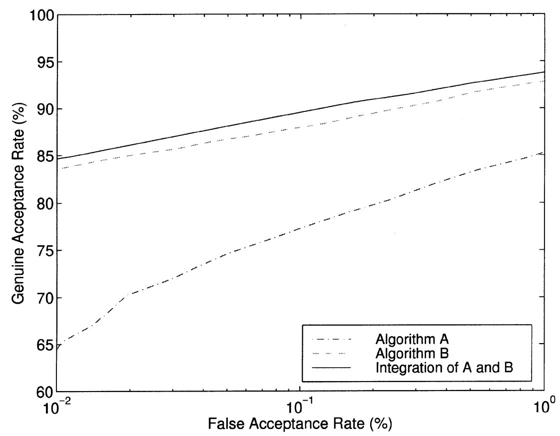

Fig. 4. ROC curves for integration of algorithms A and B.

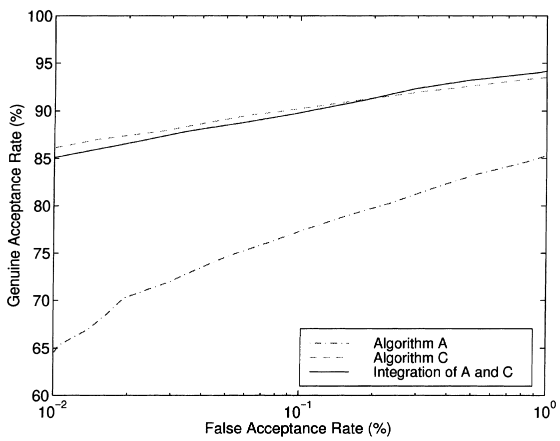

Fig. 5. ROC curves for integration of algorithms A and C.

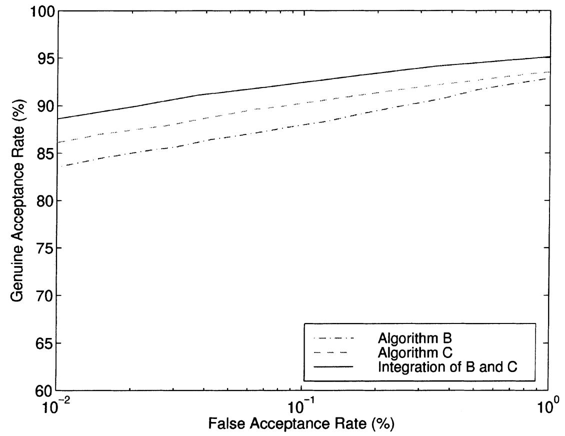

Fig. 6. ROC curves for integration of algorithms B and C.

dealing with elastic distortions. Fig. 6 shows that the integration of algorithms B and C results in a significant performance improvement, even though both the algorithms apply dynamic programming to deal with elastic distortions.

6. Conclusions and discussions

This paper presents a scheme to integrate the output scores of three different fingerprint matchers to improve the performance of a fingerprint verification system. Logistic regression makes use of the conditional probability of Y given $\boldsymbol{x}$, and can improve the accuracy of fingerprint verification by using more observations (output scores) from multiple matchers under the assumption that these observations are not totally redundant. Our experimental results show that the performance improvement by integrating algorithms B and C is significant, even though both the algorithms are minutia-based and make use of dynamic programming to deal with elastic distortion. Integration of all the three algorithms A, B and C involves solving a minimization problem with more number of parameters (four) but results in the same performance improvement as by integration of algorithms B and C alone. This is because algorithm A is substantially inferior to algorithms B and C and does not offer complementary information to algorithms B and C.

Integration of matching algorithms which utilize different types of information, e.g., minutia-based and non-minutia-based, could be an interesting future research. We are currently working on development of a non-minutiae-based matching algorithm (Jain et al., 1999b) and the preliminary results of matcher combination are very encouraging. We must point out that integration of two matching algorithms may not always result in a better performance. A poor matching algorithm may not improve the overall performance after the integration.

Discussion

Bunke: Could you comment on efficiency and scalability of the approach? You use a Hough transform with four parameters and 2D programming. How would this scale up in the case of thousands of users?

Prabhakar: It is important to point out that we are addressing the problem of fingerprint verification, and not fingerprint identification. So, a given fingerprint is not matched with a large number of fingerprints in the database; the given fingerprint is matched with only one (or a few) template(s) of the claimed identity stored in the database. Since we are doing one-to-one matching, we do not have to worry about the scalability issue in verification.

Bunke: How much time does the 4D Hough transform and the 2D dynamic programming take?

Prabhakar: All the matchers are very efficient and take less than two seconds on a Sun Ultra 1 to match two fingerprints.

Szirányi: How robust is your system against fingerprint defects?

Prabhakar: If there is a significant change in the fingerprint impression since the time of enrollment (due to cuts and bruises), it could pose a problem to the verification system and lead to false rejects. A possible solution is to update the template of a user in the database, each time a user is positively verified.

Gimel'farb: Did you check the system on real fingerprints? Given by FBI or minister of justice?

Prabhakar: Yes, we have tested our system on real fingerprints, but these did not come from the FBI. We collected our own database, using live-scan fingerprint capture devices. We have also tested our algorithms on a standard fingerprint database, called NIST-9, which is available in the public domain and contains several thousand images.

Kanal: In Caltech they use a very small chip to capture fingerprints.

Prabhakar: The Caltech sensor is based on ultrasound. Several commercial chip-based sensors that are very small are available for capturing fingerprints. These sensors are being embedded in devices such as laptop computers and cellular phones.

References

Bigün, E.S., Duc, B., Fisher, S., 1997. Expert conciliation for multi modal person authentication systems by Bayesian statistics. In: Proceedings of the First International Conference on Audio- and Video-based Biometric Person Authentication, Crans-Montana, Switzerland, pp. 291–300.

Brent, R.P., 1973. Algorithms for Minimization without Derivatives. Prentice-Hall, Engelwood Cliffs, NJ.

Chen, S., Jain, A.K., 1998. A fingerprint matching algorithm using dynamic programming. Technical report, Department of Computer Science and Engineering, Michigan State University.

Ho, T.K., Hull, J.J., Srihari, S.N., 1994. On multiple classifier systems for pattern recognition. IEEE Trans. Pattern Anal. Machine Intell. 16 (1), 66–75.

Jain, A.K., Hong, L., Bolle, R.M., 1997. Online fingerprint verification. IEEE Trans. Pattern Anal. Machine Intell. 19 (4), 302–314.

Jain, A.K., Bolle, R.M., Pankanti, S., 1998. Biometrics: Personal Identification in a Networked Society. Kluwer Academic Publishers, Dordrecht, MA.

Jain, A.K., Hong, L., Kulkarni Y., 1999a. A multimodal biometric system using fingerprint, face and speech. In: Proceedings of the Second International Conference on Audio- and Video-based Biometric Person Authentication, Washington DC, pp. 182–187.

Jain, A.K., Prabhakar, S., Hong, L., Pankanti, S., 1999b. FingerCode: A filterbank for fingerprint representation and matching. In: Proceedings of the IEEE Conference on

Computer Vision and Pattern Recognition 2, Ft. Collins, Colorado, pp. 187–193.

Kittler, J., Hatef, M., Duin, R.P.W., Matas, J., 1998. On combining classifiers. IEEE Trans. Pattern Anal. Machine Intell. 20 (3), 226–239.

Lam, L., Suen, C.Y., 1994. A theoretical analysis of the application of majority voting to pattern recognition. In: Proceedings of the 12th International Conference on Pattern Recognition, Jerusalem, pp. 418–420.

Lam, L., Suen, C.Y., 1995. Optimal combination of pattern classifiers. Pattern Recognition Letters 16, 945–954.

Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P., 1992. Numerical Recipes in C, second ed., Cambridge University Press, Cambridge, MA.

Ratha, N.K., Karu, K., Chen, S., Jain, A.K., 1996. Real-time matching system for large fingerprint database. IEEE Trans. Pattern Anal. Machine Intell. 18 (8), 799–813.

Sinha, P., Mao, J., 1998. Combining multiple OCRs for optimizing word recognition. In: Proceedings of the 14th International Conference on Pattern Recognition 1, Brisbane, pp. 436–438.

Stockman, G.C., Agrawala, A.K., 1977. Equivalence of Hough curve detection to template matching. Communications of the ACM 20, 820–822.

Xu, L., Krzyzak, A., Suen, C.Y., 1992. Methods for combining multiple classifiers and their applications to handwriting recognition. IEEE Trans. Systems, Man and Cybernetics 22 (3), 418–435.