Modification of Intersession Variability in On-Line Signature Verifier

Yasunori Hongo, Daigo Muramatsu, and Takashi Matsumoto

Department of Electrical Engineering and Bioscience, Waseda University,
3-4-1 Ohkubo, Shinjuku-ku, Tokyo, Japan
{hongo03,daigo}@matsumoto.elec.waseda.ac.jp
takashi@mse.waseda.ac.jp
http://www.matsumoto.elec.waseda.ac.jp/

Abstract. For Pen-input on-line signature verification algorithms, the influence of intersession variability is a considerable problem because hand-written signatures change with time, causing performance degradation. In our previous work, we proposed a user-generic model using AdaBoost. However, this model did not allow for the fact that features of signatures change over time. In this paper, we propose a template renewal method to reduce the performance degradation caused by signature changes over time. In our proposed method, the oldest template is replaced with a new one if the new signature data gives rise to an index which exceeds a threshold value. No further learning is necessary. A preliminary experiment was conducted on a subset of the MCYT database.

1 Introduction

Personal identity verification has a variety of applications including electronic commerce, access control for buildings and computer terminals, and credit card verification. The algorithms used to verify personal identity can be classified into the four groups described in Fig. 1, depending on whether they are static, dynamic, biometric, or physical/knowledge-based.

For example, algorithms for fingerprints, the iris, the retina, DNA, palm prints, the face, and the blood vessels are static and biometric. Algorithms classified as biometric and dynamic involve lip movements, body movements, the voice, and on-line signatures. Schemes that use passwords are static and knowledge-based, whereas methods using IC cards, magnetic cards, or keys are physical. Due to the rapidly increasing use of Tablet PCs and PDAs, on-line signature verification is a promising technique for personal identity verification.

A variety of algorithms have been proposed for on-line signature verification. Research results continue to be reported, indicating that this problem is difficult and challenging.

2 The Algorithm

2.1 Feature Extraction

The raw data from our readily available tablet (WACOM INTUOS A6 USB) consists of the five-dimensional time series data set:

$$(x(j), y(j), p(j), \gamma(j), \varphi(j)) \in R^2 \times \{0,1,...,1023\} \times R^2 \quad j = 1,2,...,J \tag{1}$$

T. Kanade, A. Jain, and N.K. Ratha (Eds.): AVBPA 2005, LNCS 3546, pp. 455–463, 2005.

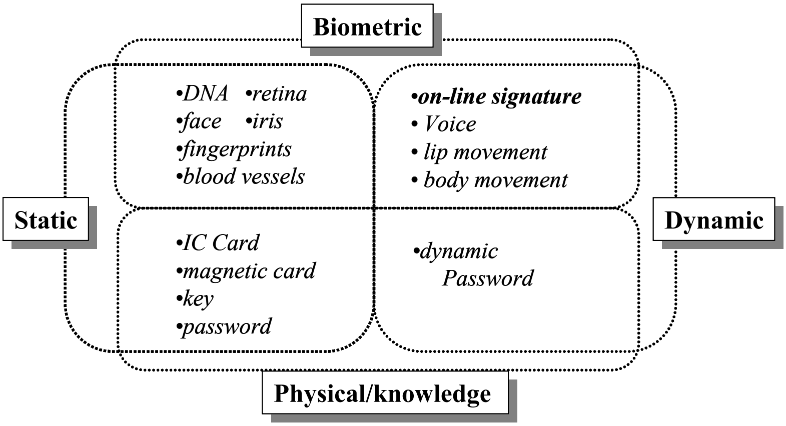

Fig. 1. Authentication methods

where $(x(j), y(j)) \in R^2$ is the pen position at time j, $p(j) \in \{0,1,\ldots,1023\}$ represents the pen pressure, $\gamma(j)$ is pen azimuth angle and $\varphi(j)$is pen altitude angle.

Define

$$X_g = \frac{\sum_{j=1}^{J} x(j)}{J} \tag{2}$$

$$Y_g = \frac{\sum_{j=1}^{J} y(j)}{J} \tag{3}$$

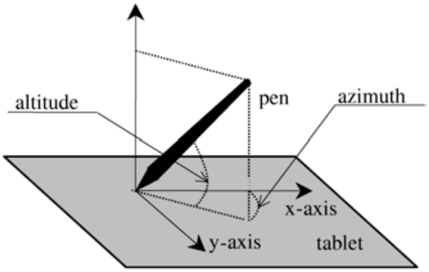

Fig. 2. Raw data from tablet

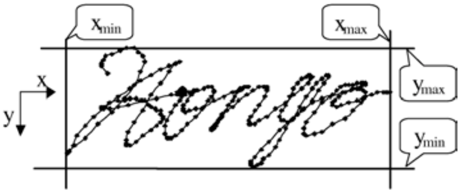

Fig. 3. $x_{\min}$, $x_{\max}$, $y_{\min}$ and $y_{\max}$ of signature

Let

$$(dx(j), dy(j)) := \left(\frac{x(j) - X_g}{x_{\max} - x_{\min}} \times L, \frac{y(j) - Y_g}{y_{\max} - y_{\min}} \times L\right) \qquad j = 1,2,\ldots,J \tag{4}$$

be the relative pen position with respect to (2) and (3) where L is a scaling parameter.

The length $f(j)$ and the angle $\theta(j)$ of each pen position are given by

$$f(j) = \sqrt{dx(j)^2 + dy(j)^2} \qquad j = 1,2,\ldots,J \tag{5}$$

$$\theta(j)=\begin{cases}\tan^{-1}\dfrac{dy(j)}{dx(j)} & (dx(j)>0)\\ sign(dy(j))\times\dfrac{\pi}{2} & (dx(j)=0)\\ \tan^{-1}\dfrac{dy(j)}{dx(j)}+\pi & (dx(j)<0,dy(j)\geq 0)\\ \tan^{-1}\dfrac{dy(j)}{dx(j)}-\pi & (dx(j)<0,dy(j)<0)\end{cases}\qquad j=1,2,...,J \tag{6}$$

Feature vectors that we use consist of the following five-dimensional data elements:

$$(\theta(j),f(j),p(j),\gamma(j),\varphi(j))\in R^2\times\{0,1,...,1023\}\times R^2 \quad j=1,2,...,J \tag{7}$$

where J is the number of sample points.

A typical original signature trajectory given by Fig. 4(a) is converted into the relative trajectory given by Fig. 4(b).

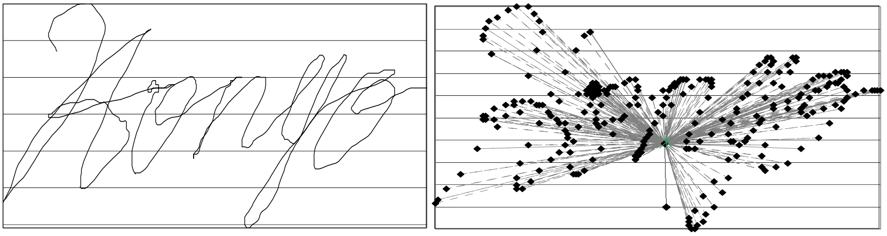
(a) (b)

Fig. 4. (a) Original signature trajectories, (b) Relative trajectories

2.2 Distance Calculation

Let

$$(\eta(k),g(k,)q(k),\delta(k),\phi(k))\in R^2\times\{0,1,...,1023\}\times R^2 \quad k=1,2,...,K \tag{8}$$

be the feature trajectory of a template signature.

We calculate the following six kinds of distance using each feature value along the number of sample points:

$$d_1 := \min_{\substack{i_s\leq i_{s+1}\leq i_s+1\\ k_s\leq k_{s+1}\leq k_s+1}} \sum_{s=1}^{S} \left|\theta(j_s)-\eta(k_s)\right| \tag{9}$$

$$d_2 := \min_{\substack{i_s\leq i_{s+1}\leq i_s+1\\ k_s\leq k_{s+1}\leq k_s+1}} \sum_{s=1}^{S} \left|p(j_s)-q(k_s)\right| \tag{10}$$

$$d_3 := \min_{\substack{i_s\leq i_{s+1}\leq i_s+1\\ k_s\leq k_{s+1}\leq k_s+1}} \sum_{s=1}^{S} \left|f(j_s)-g(k_s)\right| \tag{11}$$

$$d_4 := \min_{\substack{i_{s'}\leq i_{s'+1}\leq i_{s'}+1\\ k_{s'}\leq k_{s'+1}\leq k_{s'}+1}} \sum_{s'=1}^{S'} \left|\gamma(j_s)-\delta(k_s)\right| \tag{12}$$

$$d_5 := \min_{\substack{i_{s''}\leq i_{s''+1}\leq i_{s''}+1\\ k_{s''}\leq k_{s''+1}\leq k_{s''}+1}} \sum_{s''=1}^{S''} \left|\varphi(j_s)-\phi(k_s)\right| \tag{13}$$

$$d_6 := \left| J - K \right| \tag{14}$$

where j_1=k_1=1, j_s=J, k_s=K. J and K denote the size of the data (the number of sampled points).

Dynamic Programming can be used for computing d_1,…,d_5 because of the sequential nature of the distance function.

$$D_1(0,0) = 0$$

$$D_1(j_{s+1}, k_{s+1}) = \left| \theta(j_s) - \eta(k_s) \right| + \min \begin{cases} D_1(j_s - 1, k_s - 1) \\ D_1(j_s - 1, k_s) \\ D_1(j_s, k_s - 1) \end{cases} \tag{15}$$

2.3 Authentication Method

To distinguish genuine signatures from forged signatures, we use the six-feature vectors. We choose the Boosting algorithm for separation because its generalization error is small, and it has no free parameter affecting the threshold values when used for signature verification. AdaBoost can thus provide a good classifier.

2.4 AdaBoost

AdaBoost, originally proposed by Freund and Schapire [2], is a methodology which provides a highly accurate classifier by combining many weak classifiers.

We begin with training data, (u_1, v_1),…, (u_N, v_N), where u_i is a vector-valued feature and v_i ={-1,+1}. The training data has distribution $D_t(i)$, and $D_1(i)$ is uniform. At round t, a weak classifier defines a weak hypothesis $h_t(u_i)$ by a learning scheme that has moderate accuracy. When the classifier defines $h_t(u_i)$, we calculate the error.

$$\varepsilon_t = \sum_i D_t(i) I(h_t(u_i), v_i) \tag{16}$$

$$I(h_t(u_i), v_i) = \begin{cases} 1 & if \quad sign(h_t(u_i)) \neq sign(v_i) \\ 0 & otherwise \end{cases} \tag{17}$$

Using ε_t, we define the classifier's confidence.

$$\alpha_t = \ln \sqrt{\frac{1-\varepsilon_t}{\varepsilon_t}} \tag{18}$$

After defining the classifier's confidence, we change the distribution D using the following update rule:

$$D_{t+1}(i) = \frac{D_t(i) \exp(-\alpha_t v_i h_t(u_i))}{Z_t} \tag{19}$$

where Z_t is a normalization factor.

After finishing the learning stage, we are ready to calculate $F(u)$ and final hypothesis $H(u)$.

$$F(u) = \sum_t \alpha_t h_t(u) \tag{20}$$

$$H(u) = sign(F(u)) \tag{21}$$

2.5 Weak Classifier

In this paper, we use a two-layer perceptron as a weak classifier described by

$$h(u_m;\mathbf{w}_{weak}) := \sum_{j=1}^{U}(\mathrm{b}_j u_m + \mathrm{c}) \tag{22}$$

where

$$\mathbf{w}_{weak} = (\{b_j\},\{c\}) \tag{23}$$

U is the number of dimensions of input data. We use this as a weak classifier because it is simple and easy to calculate. Model parameter w is drawn from a Gaussian distribution.

2.6 Algorithm for Signature Verification

To effectively employ AdaBoost, many signatures belonging to both classes are necessary for training. However, we could use only a few genuine signatures, and there were no skilled forgeries available, so we could not generate a good user-specific model. Therefore, we will propose a user-generic model $Model(w_s)$ which is created by using available database (MCYT database [1] in the present study) where w_s is a parameter vector. The model does not use signature data from the person to be tested. Overall algorithm is described in Fig.5.

In the learning phase, we compute the parameter vector w_s. In the testing phase, we first calculate $u(t,sig_{test})$ defined by (24) below where D_i is defined in (9)-(15). And we use $I_1,\ldots,I_6$ defined by (24) in addition to $D_1,\ldots,D_6$ so in this paper, we use twelve feature vectors (U=12). $temp_i$ is the ith template signature.

$$\begin{aligned} &u(t,s) = (D_1, D_2, \ldots, D_6, I_1, I_2, \ldots, I_6) \\ &D_{i,j} = D_i(temp_j, sig_{test}) \quad i = 1,\ldots,6,\ j = 1,\ldots,M \\ &I_i = \frac{1}{M^2}\sum_{j=1}^{M}\sum_{k=1}^{M} D_i(temp_j, temp_k) \end{aligned} \tag{24}$$

M is the number of the template signatures.

Secondly we calculate the score as described in (25) and make decision using (26)

$$Score(sig_{test}) = \frac{\sum_{i=1}^{M} F(u(temp_i, sig_{test}))}{\frac{1}{M}\sum_{j=1}^{M}\sum_{k=1}^{M} F(u(temp_j, temp_k))} \tag{25}$$

$$sig_{test} \text{ is } \begin{cases} \text{genuine} & \text{if } c_{verf} \le Score(sig_{test}) \\ \text{forgery} & \text{if } c_{verf} > Score(sig_{test}) \end{cases} \tag{26}$$

where c_{verf} is a threshold value.

2.7 Template Renewal Method

The intersession variability of hand-written signature causes performance degradation. We propose an algorithm that solves the problem by changing template signatures.

We use the following scheme.

$$\begin{array}{ll} \text{if } c_{renew} \le Score(sig_{test}) & \text{One of the template signatures is replaced by the } sig_{test} \\ \text{if } c_{renew} > Score(sig_{test}) & \text{All the template signatures are hold} \end{array} \quad (27)$$

c_{renew} is the threshold value for changing the template signature.

Reference [3] also proposed a template renewal method, where all signatures that accepted as genuine signature were added to template signatures.

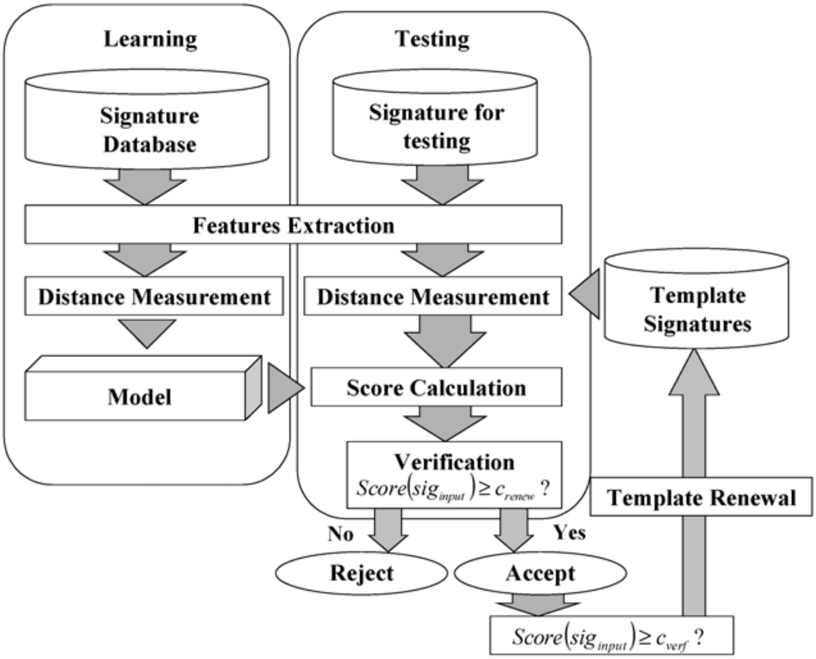

Fig. 5. Overall algorithm

3 Experiment

3.1 An Experiment with the Proposed Algorithm on Data Set 1

This section reports on an experiment using our algorithm for a data set consisting of 1000 signatures from 50 Europeans, with ten genuine signatures and ten skilled forged signatures associated with each individual. We used five genuine signatures as a template (M=5).

In this experiment, we applied a 50-fold cross-validation method. For each experiment, we used 980 signatures for training data (representing 49 individuals, excluding one test person), 5 signatures as template and 15 signatures for test data (5 genuine signatures and 10 skilled forgery signatures).

In this experiment, we continue 1,000 rounds for learning (i.e., T=1000) and set c_{renew}=0.7. Table 3.2 shows the error rates for our algorithm. In order to report an Error tradeoff curve, we tested several values of c_{verf}, although c_{verf}=0 is selected in original AdaBoost. Fig.6 displays the Error tradeoff curve.

Table 3.1. Data Set 1 (for one experiment)

Signatures for Learning		Template Signatures	Signatures for Testing	
Genuine	Forgery	Genuine	Genuine	Forgery
490	490	5	5	10

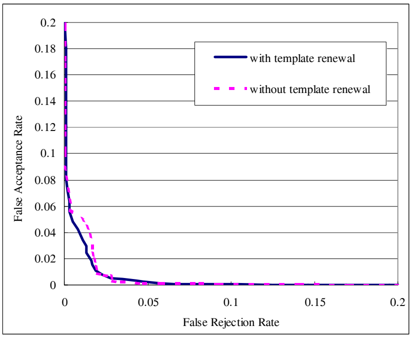

Fig. 6. Error tradeoff curve for Data Set 1

Table 3.2. Verification Error Rate on Data Set 1

	With Template Renewal	Without Template Renewal
EER	1.66%	1.82%
FR(@c_{verf} =0)	1.70%	1.90%
FA(@c_{verf} =0)	1.54%	1.80%
FR(@FA=1%)	1.85%	2.10%

3.2 Experiment Using the Proposed Algorithm on Data Set 2

This section reports on the experimental results of our algorithm for the second data set consisting of 5000 signatures from 100 Europeans, with 25 genuine signatures and 25 skilled forged signatures associated with each individual. About 70% of Data Set 1 is included in Data Set 2. It corresponds to 14% of Data Set 2.

Table 3.3. Data Set 2 (for one experiment)

Signatures for Learning		Template Signatures	Signatures for Testing	
Genuine	Forgery	Genuine	Genuine	Forgery
2475	2475	5	20	25

To show that our algorithm reduces the influence of intersession variability, we divided the genuine signatures used for testing into four groups. Each group consists of five signatures as follows:

Group 0 (template signatures): 1st-5th genuine signatures
Group 1: 6th-10th genuine signatures
Group 2: 11th-15th genuine signatures
Group 3: 16th-20th genuine signatures
Group 4: 21st-25th genuine signatures

In this experiment, we applied a 100-fold cross-validation method. For each experiment, we used 4900 signatures for training data (representing 99 individuals, excluding one test person), 5 signatures as template and 45 signatures for test data (20 genuine signatures and 25 skilled forgery signatures). We continued for 2,000 rounds for learning (i.e., $T = 2000$) and set $c_{renew} = 0.7$.

Table 3.4 shows the error rates for our algorithm. Figure.7 displays the Error tradeoff curve.

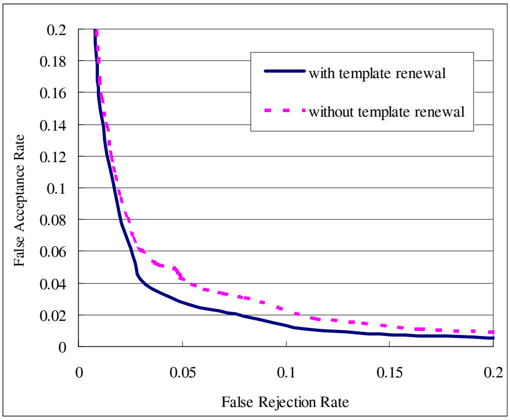

Fig. 7. Error tradeoff curve on Data Set 2

Table 3.4. Verification Error Rate on data set 2

		Total	Group 1	Group 2	Group 3	Group 4
With Template Renewal	EER	3.60%	3.14%	3.43%	3.22%	4.64%
	FR(@c_{verf}=0)	3.55%	2.40%	3.20%	2.60%	6.00%
	FA(@c_{verf} =0)	3.63%	3.63%	3.63%	3.63%	3.63%
	FR(@FA=1%)	11.85%	7.2%	12.7%	11.8%	15.7%
Without Template Renewal	EER	4.72%	3.44%	4.31%	5.06%	5.32%
	FR(@c_{verf} =0)	3.70%	1.60%	3.60%	4.20%	5.40%
	FA(@c_{verf} =0)	5.28%	5.28%	5.28%	5.28%	5.28%
	FR(@FA=1%)	16.15%	12.10%	19.40%	17.40%	23.70%

4 Conclusion

We proposed a verification algorithm using template renewal to reduce the influence of intersession variability. We improved the verification rate by using this algorithm. Considering that no fine tuning was done, this algorithm looks very promising.

References

1. J.Ortega-Garcia and J. Fierrez-Aguilar and D.Simon, J. Gonzalez and M. Faundez-Zanuy, V. Espinosa and A. Satue and I. Hemaez: MCTY Baseline corpus: a bimodal biometric database, IEE Proceeding Vision, Image and Signal Processing, vol. 150, No. 6, 2003

2. Yoav Freund and Robert E. Schapire: A decision-theoretic generalization of on-line learning and an applicationto boosting. In Computational Learning Theory: Eurocolt '95, pages 23–37. Springer-Verlag, 1995.
3. S.Yamanaka, Masato Kawamoto, T.Hamamono, and S.Hangai,: Signature Verification Adapting to Intersession Variability: IEEE International Conference on Multimedia and Expo 2001, Tokyo Japan (2001) 88-91.
4. Y. Hongo, D. Muramatsu, and T. Matsumoto: AdaBoost-based on-line signature verifier: SPIE Defense and Security Symposium, Orlando, Florida USA (2005) Proceeding vol. 5779 Biometric Technology for Human Identification II 373-380.