



On-line signature verification based on split-and-merge matching mechanism

Quen-Zong Wu^{a,b}, Suh-Yin Lee^{a,*}, I-Chang Jou^c

^a Department of Computer Science and Information Engineering, National Chiao-Tung University, Hsin-Chu, Taiwan, ROC

^b Telecommunication Laboratories, P.O. Box 71, Chung-Li, Taiwan, ROC

^c Department of Electrical Engineering, Chang Gung College of Medicine and Technology, Tao-Yuan, Taiwan, ROC

Received 21 November 1996; revised 5 May 1997

Abstract

In this paper, an on-line signature verification scheme based on split-and-merge matching mechanism is proposed. Each word in the signature is specified with static and dynamic features: a sequence of (x, y) coordinates and a sequence of (x, y) velocities. We employ the split-and-merge matching mechanism for each input sequence of coordinates or velocities with the corresponding sequence of the reference template. Alignment and refinement are exploited in the matching to justify the data skew problem. The accumulated coordinate and velocity distances are compared with corresponding verification thresholds to determine the genuineness of the input signature. Algorithms to compute reference templates and verification thresholds are also proposed. The performed simulation results support the effectiveness of this signature verification scheme.

Keywords: Signature verification; Split-and-merge; Distance measurement; Dynamic feature; Static feature

1. Introduction

Signatures are the most popular validation tools for documents or commercial transactions. Signatures can be verified either on-line or off-line. Off-line verification determines the genuineness of the signature through examining the overall shapes of the signatures. On the other hand, on-line verification can also capture the dynamics of the signatures, such as the variation of the coordinates, to make decision. Hence, forged signatures are more likely to be detected through on-line approaches.

Imitating either overall shape or dynamics of a

signature is achievable, but to achieve both is difficult. The imitator is not likely to construct a similar overall shape of a signature without showing his hesitation in the waveform of the writing velocity. Therefore, signature verification systems adopting both overall shapes and dynamics of the signatures to judge their genuineness are more reliable.

So far, signatures are usually verified through visual examination. Neither “off-line” checking on already signed signatures nor “on-line” examining on signatures being signed is an easy task for human beings. Human eyes are not able to capture the dynamic characteristics of the signatures, and the signatures are verified through their overall shapes only. Thus, human eyes can only verify signatures

* Corresponding author.

roughly. Using computer-based approaches for signature verification appears to be the possible solution for better verification precision.

Recently, many methods (Plamondon, 1989) have been developed for computer-based signature verification, especially for on-line signature verification. Among these methods, several criteria are considered for building signature verification systems. The criteria include: static or dynamic features, matching scheme and thresholding. Static features, such as coordinates of sample points, as well as dynamic features, including writing velocities and acceleration, are useful characteristics for on-line signature verification. Matching schemes are responsible for calculating the similarity or distance between two signatures. Thresholding is to choose appropriate thresholds for reference templates such that the input signatures whose distances from the reference templates are less than the thresholds are regarded as genuine signatures.

Most researches of signature verification select trajectory-based (or time domain) features for measuring the similarity or distance of the signatures (Wu et al., 1994; Chen et al., 1991; Yasuhara et al., 1977; Lew, 1983; Herbst et al., 1977; Congedo et al., 1994; Suen et al., 1993). In addition, polar coordinates (Wu et al., 1995), which can be derived from (x, y) coordinates, of the trajectory can also be used to feature the signatures. Fig. 1 shows a Chinese signature of three words. The x and y coordinates of

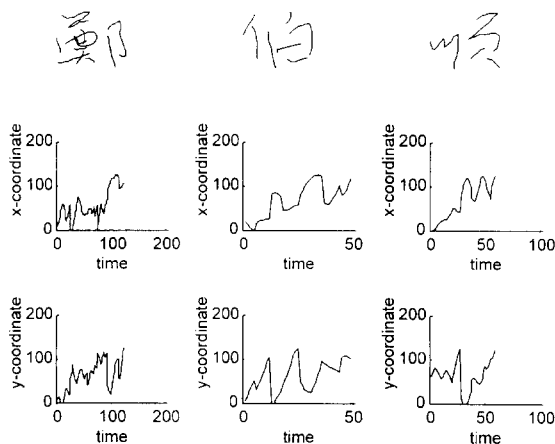


Fig. 1. A Chinese signature of three words and their x and y coordinates.

the sample points in each word are also demonstrated in this figure. Trajectory-based features, such as coordinates or velocities, are the basic characteristics of the sample points. In fact, the overall shape of a signature word can be specified with a sequence of (x, y) coordinates of all sample points, while the dynamics of the signature word can be the sequence of (x, y) velocities of all sample points. Thus, matching two signatures is to match their corresponding sequences of coordinates or velocities.

With respect to the reference sequences, the input sequences may be skewed or of different length. Hence, in order to match input signatures with the reference templates, the most important task is to align the input sequences of the input patterns with the reference sequences. We have proposed a split-and-merge algorithm for spatial/temporal pattern matching (Wu et al., 1996). First, the input pattern is refined to remove the non-uniform compression or spreading among its sub-patterns. Refinement can be recursive such that each sub-pattern is refined by removing the non-uniform compression or spreading among its sub-sub-patterns. It is guided to minimize the distance between input pattern and the reference template. After the input pattern is refined, we apply interpolation skills to the refined input pattern and the reference template to make them of the same length. Finally, we have one-to-one correspondences between the sample points of the refined input pattern and the sample points of the reference template. The distance between input pattern and the reference template can thus be computed by summing up the distances between corresponding input and reference sample points.

In this paper, we propose an on-line signature verification scheme based on split-and-merge matching mechanism. Signature words are represented as sequences of (x, y) coordinates and (x, y) velocities representing static and dynamic features. In matching the input signature words with the reference signature words, each input sequence is split into two sub-sequences and each input sub-sequence is matched with its corresponding reference sub-sequence. Recursively, for matching an input sub-sequence with a reference sub-sequence, the input sub-sequence is split into two sub-subsequences and each input sub-subsequence is matched with its corresponding reference sub-subsequence. The depth of

split-and-merge could be pre-set. We could also control the depth of split-and-merge by setting the threshold of some parameter measurement, such as the percentage of the reduction of matching distance. The split-and-merge matching process continues if the measurement is smaller (or larger) than the threshold. For simplicity, we pre-set the depth of split-and-merge in this paper. The final refined input sequence could be obtained through merging its sub-sequences after they are refined. Finally, we can compute the distance between the reference sequence and the refined input sequence. By packing the matching result of the signature words, we could obtain the overall coordinate distance and velocity distance between the input signature and the reference template. The genuineness of the input signature is determined by comparing these two distances with their corresponding verification thresholds. If both of them are smaller than the verification thresholds, then the input signature is genuine; otherwise, it is forged.

2. Signature specification

As we mentioned above, signatures can be characterized by their overall shapes and dynamics. The overall shapes of signatures can be specified with sequences of (x, y) coordinate pairs of the writing trajectory, while the dynamics of the signatures can be represented by sequences of (x, y) velocity pairs. Coordinates are all we have from the data capture device, i.e. the digitizer. Velocities, which are approximately derived from coordinates, are less accurate than coordinates although they can reflect the change of coordinates. Accelerations, which are approximately derived from velocities, are even less accurate than velocities although they can reflect the change of the change of coordinates. Therefore, we select only coordinates and velocities to feature the signatures.

A signature could be regarded as consisting of several words, depending on the skills used for signature segmentation. The easiest way is to regard the whole signature as a big word no matter how the signature is signed. Another convenient approach is to restrict the users to write signature words in different locations on the tablet to avoid segmenta-

tion problem. Each word in the signature is described by a sequence of coordinates and a sequence of velocities. Thus, for a signature of M words, $2M$ sequences are used to specify it.

If words of a signature are allowed to be connected, then the connected words will be treated as an entire entity. No segmentation is required. A single unconnected word is also an entity. Therefore, if the number of entities of the input signature is different from that of the reference signature, we determine that the input signature is a forgery and omit the matching process. An entity of connected words in the input signature is matched with its corresponding entity in the reference signature using the same split-and-merge process as the matching of an unconnected single word does.

Usually, we use a digitizer to capture the coordinates of the on-line signatures. Hence, the overall shape of a signature is directly provided by the digitizer. The sequence of velocities of each signature word needs to be derived through the following formulas:

$$U_{i+1} = \left. \frac{\Delta X}{\Delta t} \right|_{t=t_{i+1}} \cong \frac{(X_{i+1} - X_i)}{(t_{i+1} - t_i)} \quad (1)$$

and

$$V_{i+1} = \left. \frac{\Delta Y}{\Delta t} \right|_{t=t_{i+1}} \cong \frac{(Y_{i+1} - Y_i)}{(t_{i+1} - t_i)}, \quad (2)$$

where (X_{i+1}, Y_{i+1}) are the x - and y -coordinates at sampling time t_{i+1} and (U_{i+1}, V_{i+1}) are the derived x - and y -velocities.

3. Split-and-merge matching mechanism

Spatial/temporal patterns can be represented as sequences of outcomes from observation. The term ‘‘outcomes from observation’’ could be (x, y) coordinates for on-line handwritings, spectral magnitudes for speech signals and pixels for images. With respect to the reference sequences, the input sequences may be skewed or of different length. Hence, in order to match input patterns with the reference templates, the most important task is to align the spatial/temporal sequences of the input patterns with the spatial/temporal sequences of the reference templates.

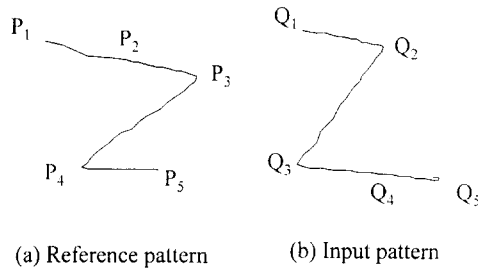


Fig. 2. Two handwritten patterns for illustrating skews.

Fig. 2 shows an example of non-uniform compression or spreading of input line segments $\overline{Q_1Q_2}$, $\overline{Q_2Q_3}$ and $\overline{Q_3Q_5}$ relative to reference line segments $\overline{P_1P_3}$, $\overline{P_3P_4}$ and $\overline{P_4P_5}$, respectively. Though the correspondences between $\overline{Q_1Q_2}$ and $\overline{P_1P_3}$ and the correspondence between $\overline{Q_3Q_5}$ and $\overline{P_4P_5}$ are not uniform in length, i.e. $\overline{Q_1Q_2} < \overline{P_1P_3}$ but $\overline{Q_3Q_5} > \overline{P_4P_5}$, these two correspondences seem reasonable. Therefore, successful alignments of $\overline{Q_1Q_2}$ to $\overline{P_1P_3}$ and $\overline{Q_3Q_5}$ to $\overline{P_4P_5}$ are critical in matching the input pattern with the reference pattern.

We use Fig. 3 as a simplified example to illustrate the resolution of the alignment problem. The reference sequence R of symbolic outcomes 1, 2, 3, 4 and 5 in Fig. 3(a) consists of five sub-sequences: sub-sequences R_1 , R_2 , R_3 , R_4 and R_5 of outcomes 1111, 2222, 3333, 4444 and 5555, respectively. On the other hand, the input sequence S , shown in Fig. 3(b), consists of five sub-sequences: sub-sequences S_1 , S_2 , S_3 , S_4 and S_5 of outcomes 1, 22, 333, 44 and 55, respectively. In the alignment of these two sequences, it is reasonable to relate R_1 , R_2 , R_3 , R_4 and R_5 to S_1 , S_2 , S_3 , S_4 and S_5 , respectively. In matching input sequence S with the reference se-

- (a) 11112222333344445555
 (b) 1223334455
 (c) 1112223333444555
 (d) 11112222333344445555

Fig. 3. An example of aligning two sequences: (a) the original reference sequence, (b) the original input sequence, (c) the refined input sequence, (d) the enlarged version of the refined input sequence.

quence R , we find that S_1 is relatively compressed, while S_3 is relatively spread because

$$\frac{|S_1|}{|R_1|} < \frac{|S_2|}{|R_2|} = \frac{|S_4|}{|R_4|} = \frac{|S_5|}{|R_5|} < \frac{|S_3|}{|R_3|},$$

where $|S_i|$ and $|R_i|$ means the respective length of S_i and R_i . If we spread S_1 by 2 outcomes and spread each of S_2 , S_4 and S_5 by 1 outcome, we could make

$$\frac{|S_1|}{|R_1|} = \frac{|S_2|}{|R_2|} = \frac{|S_4|}{|R_4|} = \frac{|S_5|}{|R_5|} = \frac{|S_3|}{|R_3|}.$$

We thus obtain the refined input sequence \hat{S} as shown in Fig. 3(c).

The above steps illustrate the process of removing non-uniform compression or spreading among sub-sequences of the input sequence with respect to the reference sequence. By now, there still remains the problem that the lengths of the reference sequence and the refined input sequence are not the same. To make one-to-one alignment of outcomes successful, we may enlarge the smaller of the reference sequence and the refined input sequence to make them of the same length. This is achievable by outcome duplication for symbolic outcomes or interpolation for numeric outcomes. We thus obtain the enlarged version of the refined input sequence as shown in Fig. 3(d). The exact match of the reference sequence with the enlarged input sequence implies that the input pattern belongs to the same category as the reference pattern. Due to the relatively faster writing speed at some portions of the signing, the sequence representing the signature may be compressed at these portions. Sequence refinement is to have the compressed portions of the sequence enlarged.

The above example is a simplified application of the split-and-merge matching algorithm. Four steps can be concluded from the above example:

- Step 1 Split input and reference sequences into sub-sequences.
- Step 2 The input sub-sequences are refined by removing non-uniform compression or spreading among sub-sequences.
- Step 3 Obtain the refined input sequence by merging its refined sub-sequences.
- Step 4 Enlarge the shorter of the refined input sequence and the reference sequence such that

their lengths are equal. Compute the similarity or the distance between two sequences.

Splitting a sequence into sub-sequences could be straightforward for a sequence of symbolic outcomes. But, for the sequences of numeric outcomes, such as the spectral magnitudes of the speech signal, this is not the case. Therefore, it is difficult to relate a reference sub-sequence to an input sub-sequence. Step 1 for splitting a sequence into sub-sequences needs to be enhanced. The sub-steps for enhanced Step 1 is listed as follows:

- Step 1.1 For the reference sequence, split it at the middle of the sequence to produce two sub-sequences of equal length. As for the input sequence, split it into two sub-sequences such that the refined input sequence, which is obtained from merging its sub-sequences after they are refined, best matches the reference sequence.
- Step 1.2 Recursively apply the split-and-merge algorithm for matching the first input sub-sequence with the first reference sub-sequence and matching the second input sub-sequence with the second reference sub-sequence.

The split-and-merge matching is achieved through these steps. The pseudo codes for split-and-merge matching is shown in Fig. 4 and the details can be found in (Wu et al., 1996). Unlike conventional matching algorithms, such as dynamic programming (Furui, 1989), which piecwisely advance the matching to accumulate the matching results, this algo-

```

Match(Z, J, dcpth) // Z: the reference sequence, J: the input sequence.
                  // depth: the splitting depth
{
  if depth>0
    Find the splitting point bp to split sequence J into two sub-sequences.
    J0=J[1..bp]; J1=J[bp+1..length(J)];
    Z0=Z[1..length(Z)/2]; Z1=Z[length(Z)/2+1..length(Z)];
    Match(Z0, J0, depth-1);
    Match(Z1, J1, depth-1);
    Enlarge the smaller sequence of J0 and J1.
    J is the concatenation of J0 and J1.
  end if
  Enlarge the smaller sequence of Z and J.
  Compute the Euclidean distance between Z and J.
  return distance;
}
    
```

Fig. 4. The recursive pseudo codes for split-and-merge matching.

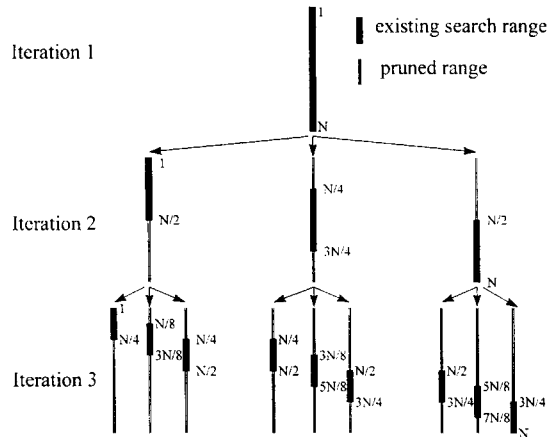


Fig. 5. The existing search range for each iteration.

rithm, which recursively splits patterns into sub-patterns, is a top-down approach.

Step 1.1 searches the optimal point to split the input sequence into two sub-sequences. In order to enhance the search speed, we need to iteratively prune away some sample points which are unlikely to be the optimal splitting point. Fig. 5 shows the tree structure of the variations of the existing search ranges and the pruned ranges. In searching the optimal point to split the input sequence of N sample points into two sub-sequences, we first select the candidates of the optimal point as follows:

- Candidate 1: the $(N/4)$ th sample point of the sequence.
- Candidate 2: the $(N/2)$ th sample point of the sequence.
- Candidate 3: the $(3N/4)$ th sample point of the sequence.

We define that two sequences best match with each other if they yield the highest similarity or the lowest distance. The words “best match” in distance is to choose the matching style which leads to minimum distance. Assume D_1 , D_2 and D_3 are the distances of matching based on splitting the input sequence at the $(N/4)$ th, the $(N/2)$ th and the $(3N/4)$ th sample points of the sequence. Then $\min(D_1, D_2, D_3)$ is the distance that best matches the input sequence with the reference sequence.

If the refined input sequence that can be obtained from splitting at the first candidate best matches with the reference sequence, then report the matching result or continue to search the splitting point be-

tween the *first* and the $(N/2)$ th sample point of the input sequence. If the refined input sequence obtained from splitting at the second candidate best matches with the reference sequence, then report the matching result or continue to search the splitting point between the $(N/4)$ th and the $(3N/4)$ th sample point of the input sequence. If the refined input sequence obtained from splitting at the third candidate best matches with the reference sequence, then report the matching result or continue to search the splitting point between the $(N/2)$ th and the N th sample point of the input sequence. This is an iterative search scheme. Each iteration prunes away one half of the sample points in the existing search range.

Each call of the split-and-merge algorithm breaks a spatial/temporal sequence into two sub-sequences. Therefore, recursive calls on this algorithm for *depth* times will break a sequence into 2^{depth} sub-sequences. The deeper the recursive call is, the more refined the input sequence is. We have to take computation time into consideration when the recursion is deep. Usually, the depth of the recursive call is pre-set.

In (Wu et al., 1996), the similarity between two sequences, whose lengths are adjusted through interpolation, is obtained by Pearson correlation formula (Flannery et al., 1988; Vald'es, 1993). In this paper, we use the following formula to measure the distance such that the computation time is reduced. Assume that patterns V and W are constructed by sequences of samples V_0, V_1, \dots, V_{S-1} and samples W_0, W_1, \dots, W_{S-1} with each sample being a vector $V_i = (v_i^0, v_i^1, \dots, v_i^{T-1})$ and $W_i = (w_i^0, w_i^1, \dots, w_i^{T-1})$, respectively. S is the number of samples. T is the dimension of V_i and W_i , $0 \leq i \leq S-1$. Then the distance between V and W is

$$\text{Distance}(V, W) = \sum_{i=0}^{S-1} \sum_{j=0}^{T-1} |(v_i^j - \bar{v}^j) - (w_i^j - \bar{w}^j)|, \quad (3)$$

where \bar{v}^j and \bar{w}^j are the respective mean values of $\{v_i^j | 0 \leq i \leq S-1\}$ and $\{w_i^j | 0 \leq i \leq S-1\}$.

In this paper, each word of the signature is represented as sequences of sample points. In matching the input signature words with the reference signature words, each input sequence is split into two sub-sequences and each input sub-sequence is

matched with its corresponding reference sub-sequence. We recursively call the split-and-merge algorithm until the pre-set recursive depth is met. Then the refined input sequence is obtained through merging its sub-sequences after they are refined. Finally, the distance between the reference sequence and the refined input sequence can be computed. As we mentioned in the previous section, each signature word is specified with a sequence of coordinates and a sequence of velocities. The matching results of the coordinate sequences of all words are summed up as the coordinate distance between the reference template and the input signature. The matching results of the velocity sequences of all words are summed up as the velocity distance between the reference template and the input signature.

4. Reference template and verification threshold

The reference templates and the verification thresholds of registered persons must be computed before the verification system can determine the genuineness of the input signature. Though the simplest method for obtaining the reference template is to choose one of the training signatures as the reference template, but this approach is risky because the chosen one may not be able to accurately stand for the usual signing style. For each registered person, we choose three training signatures to generate the reference template. One of these training signatures is chosen as the initial reference template. Our training facility then matches these three training signatures with the initial reference template and we receive the refined training signatures as return. We then average these three refined signatures as the new reference template. Again, we could match the three original training signatures with the reference template and revise the reference template. The reference template could further be revised for a pre-set number of iterations. Each reference template is represented as coordinate sequences as well as velocity sequences for signature words. Fig. 6 shows the examples of training signature words and the generated reference template.

We use another 7 training signatures for computing the verification thresholds. The coordinate distance and the velocity distance are acquired by

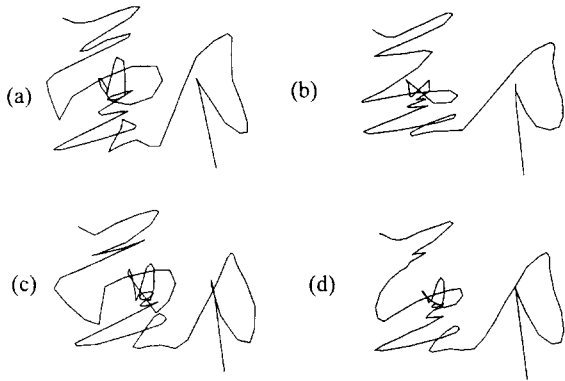


Fig. 6. Examples of training signature words (a), (b) and (c) and the generated reference template (d).

matching these 7 signatures with the reference template. Listed in the following are the formulas for the computation of verification thresholds:

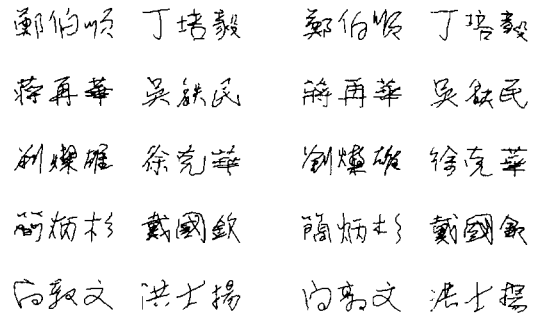
$$T_c = \min(\text{mean}\{C_i | 1 \leq i \leq 7\} + 3.0 * \text{standard_derivation}\{C_i | 1 \leq i \leq 7\}, 1.1 * \max\{C_i | 1 \leq i \leq 7\})$$

and

$$T_v = \min(\text{mean}\{V_i | 1 \leq i \leq 7\} + 3.0 * \text{standard_derivation}\{V_i | 1 \leq i \leq 7\}, 1.1 * \max\{V_i | 1 \leq i \leq 7\}),$$

where

- T_c : the threshold of coordinate distance,
- C_i : the coordinate distance between the i th training signature and the reference template,
- T_v : the threshold of velocity distance,
- V_i : the velocity distance between the i th training signature and the reference template.



(a) genuine signatures (b) forged signatures

Fig. 7. Examples of genuine and forged signatures of the simulation.

An input signature is verified as a genuine one if both of its coordinate and velocity distance from the reference template are less than the respective coordinate and velocity threshold; otherwise, it is a forgery.

5. Simulation

We collect genuine signatures from 10 registered persons and forged signatures from 4 imitators for simulation. Each signature contains three words. Fig. 7 shows the examples of genuine and forged signatures in the simulation. Each registered person signs his/her signatures for 30 times, ten of which are used to acquire the reference template and the verification threshold and 20 of which are used for the testing. Totally, there are $20 * 10 = 200$ genuine signatures of all registered persons for system testing.

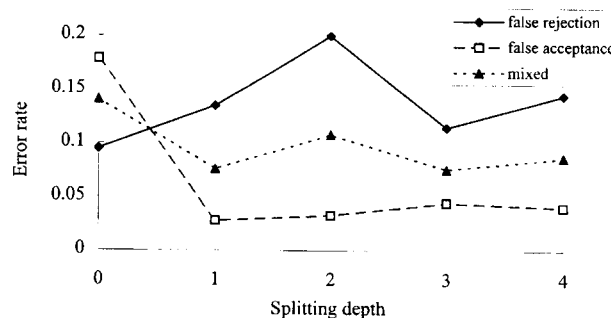


Fig. 8. Error rates of all registered persons for different splitting depth.

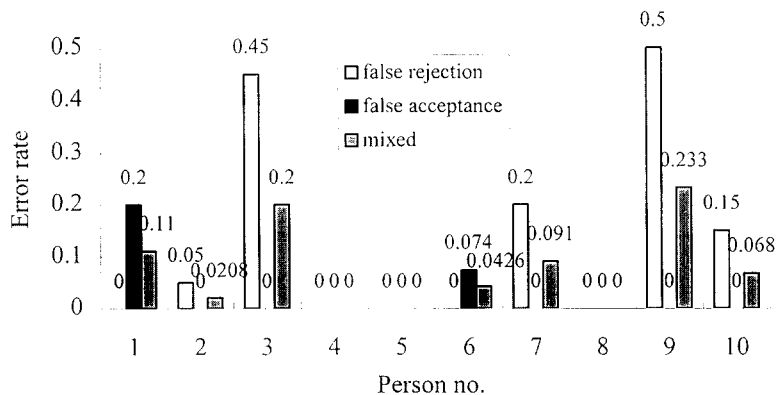


Fig. 9. The error rates of each registered person with splitting $depth = 1$.

The genuine signatures of all registered persons are shown to the imitators when they imitate. The imitators totally forge 246 signatures.

False rejection rate is defined to be the rate that the genuine signatures are classified as forgeries. False acceptance rate is the rate that the forgeries are classified as genuine ones. The mixed error rate is defined to be the ratio of the number of mis-verified genuine and forged signatures over the total number of genuine and forged signatures. Thus, if false rejection and false acceptance rates are $A/200$ and $B/246$, then the mixed error rate should be $(A + B)/(200 + 246)$.

Fig. 8 depicts the false rejection, false acceptance and mixed error rates of all registered persons with different splitting depth. It demonstrates that this verification scheme can verify signatures with high accuracy. Among these three types of error rates, false acceptance rate is the most important because the false acceptance of the forged signatures may cause fatal losses. There is a big decline in false acceptance rate from $depth = 0$ to $depth = 1$, which results from the fact that the refined training signatures can provide stricter verification thresholds and more accurate reference templates such that the forgeries are less likely to be accepted. As the splitting depth increases, the verification thresholds decrease and the input genuine and forged signatures are more refined such that the distance between the input signature, forged or genuine, and the reference template decreases. Under such circumstances, the false rejection and false acceptance rates are not guaranteed to decrease as the splitting depth increases. This

explains why the false rejection and the false acceptance rates do not decline as we raise the splitting depth from 1 to 2 or from 2 to 3. Setting the splitting depth to 1 seems optimal in this simulation.

Fig. 9 shows these three types of error rates for each registered person when splitting depth is 1. Many signatures are verified with zero false rejection and/or zero false acceptance. However, as the writing behaviors of some persons are not stable, a genuine signature written at time instance t_1 may differ greatly from that written at time instance t_2 with or without intention. Therefore, the error rates are more or less time-dependent. Other factors to affect the quality of signatures include the writing tablet and the mood of the writer. Most of the errors occur because signing on the tablet is not so smooth as signing on the paper. Under such circumstances, some genuine signatures may look like forgeries and some forgeries may look like genuine signatures.

6. Conclusions

In this paper, an on-line signature verification scheme based on split-and-merge matching algorithm is proposed. Both the overall shapes and the dynamics of the signatures are considered for the determination of genuineness of the signatures. A mechanism for generating the reference templates and the verification thresholds is developed such that the generated templates and thresholds can be as representative as possible. The input signature is regarded as a genuine one if both of its coordinate

and velocity distance from the reference template are less than the respective coordinate and velocity threshold; otherwise, it is a forgery. From the simulation on 10 registered persons, it is demonstrated that 86.5% of the genuine signatures can be successfully verified and 97.2% of the forgeries can be pointed out when splitting depth is 1.

Although the term “signature” is generally known in western countries to refer to a name handwritten in an alphabetic script, there is no doubt that this term could also refer to handwritten names in character form, such as Chinese characters. We use Chinese signatures, which are often written in character-by-character form, for simulation. In fact, this work can be easily adapted to other types of signatures. For many languages, word connection is usually the case. In the simulation of this paper, the split-and-merge process is for a single word or an entity. If the words of a signature are allowed to be connected, then the connected words will be treated as an entire entity. Since the connected words are treated as an entity, the split-and-merge process still works. Therefore, the proposed method works for a signature consisting of connecting letters or characters, as often seen in Latin languages.

References

- Chen, Y.-T., Z. Chen (1991). Automatic authentication of on-line signature data. In: *Proc. Nat. Computer Symp.*, 446–451.
- Congedo, G., G. Dimauro, S. Impedovo and G. Pirlo (1994). A new methodology for the measurement of local stability in dynamic signatures. In: *Proc. 4th Internat. Workshop on Frontiers in Handwriting Recognition*, 135–144.
- Flannery, B.P., et al. (1988). *Numeric Recipes in C*. Cambridge University Press, Cambridge, MA.
- Furui, S. (1989). *Digital Speech Processing, Synthesis, and Recognition*. Marcel Dekker, New York.
- Herbst, N.M. and C.N. Liu (1977). Automatic signature verification based on accelerometry. *IBM J. Res. Develop.* 21, 245–253.
- Law, J.S. (1983). An improved regional correlation algorithm for signature verification which permits small speed changes between handwriting segments. *IBM J. Res. Develop.* 27 (2), 181–185.
- Plamondon, R. and G. Lorette (1989). Automatic signature verification and writer identification – The state of the art. *Pattern Recognition* 22 (2), 107–131.
- Suen, H.-M., J.-F. Wang and H.-D. Chang (1993). On-line Chinese signature verification. In: *Proc. IPPR Conf. on Computer Vision, Graphics and Image Processing*, 29–36.
- Vald'es, R. (1993). DDJ handprinting recognition contest wrap-up. *Dr. Dobb's Journal*, 52–93.
- Wu, Q.-Z., et al. (1994). On the distortion measurement of on-line signature verification. In: *Proc. 4th Internat. Workshop on Frontiers in Handwriting Recognition*, 347–353.
- Wu, Q.-Z., I.-C. Jou and S.-Y. Lee (1995). On-line signature verification through statistical analysis. In: *Proc. Internat. Symp. on Communications*, 1050–1056.
- Wu, Q.-Z., S.-Y. Lee, I.-C. Jou (1996). A split-and-merge algorithm for spatial/temporal pattern matching. In: *Proc. IPPR Conf. on Computer Vision, Graphics and Image Processing*, 98–105.
- Yasuhara, M. and M. Oka (1977). Signature verification experiment based on nonlinear time alignment: a feasibility study. *IEEE Trans. Systems Man Cybernet.* 7 (3), 212–216.