Detailed Instructions for Participants

SVC 2004: First International Signature Verification Competition

Signature Database Design

This competition consists of two separate signature verification tasks, each of which is based on a different signature database. The signature data for the first task contain coordinate information only, but the signature data for the second task also contain additional information including pen orientation and pressure. The first task is suitable for on-line signature verification on small pen-based input devices such as personal digital assistants (PDA) and the second task on digitizing tablets.

Each database has 100 sets of signature data. Each set contains 20 genuine signatures from one signature contributor and 20 skilled forgeries from five other contributors. Only 40 sets of signature data are released to participants for developing and evaluating their systems before submission. Although both genuine signatures and skilled forgeries are made available to participants, it should be noted that user enrollment during system evaluation will accept only five genuine signatures from each user, although multiple sets of five genuine signatures each will be used in multiple runs. Skilled forgeries will not be available during the enrollment process. They will only be used in the matching process for system performance evaluation. However, evaluation of signature verification performance for each user will only start after all users have finished their enrollment. Thus, participants may make use of genuine signatures from other users in improving the verification accuracy for a user.

Signature Files

Each signature is stored in a separate text file (with `.TXT` file extension). The naming convention of the signature files is `UxSy`, where `x` is the user ID and `y` is the signature ID. The signature files released to participants have `x` values ranging from 1 to 40. Genuine signatures correspond to `y` values from 1 to 20 and skilled forgeries from 21 to 40.

In each signature file, the signature is simply represented as a sequence of points. The first line stores a single integer which is the total number of points in the signature. Each of the following lines corresponds to one point characterized by features listed in the following order (the last three features are missing in signature files for the first task):

X-coordinate	-	scaled cursor position along the x-axis
Y-coordinate	-	scaled cursor position along the y-axis
Time stamp	-	system time at which the event was posted
Button status	-	current button status (0 for pen-up and 1 for pen-down)
Azimuth	-	clockwise rotation of cursor about the z-axis
Altitude	-	angle upward toward the positive z-axis
Pressure	-	adjusted state of the normal pressure

Signature Data Collection

Each data contributor was asked to produce 20 genuine signatures and 20 skilled forgeries in two separate sessions. For privacy reasons, the contributors were advised not to use their real signatures in daily use. Instead, they were suggested to design a new signature and to practice the writing of it sufficiently so that it remained relatively consistent over different signature instances, just like real signatures. Contributors were also reminded that consistency should not be limited to spatial consistency in the signature shape but should also include temporal consistency due to dynamic features.

In the first session, each contributor was asked to contribute 10 genuine signatures. Contributors were advised to write naturally on the digitizing tablet (WACOM Intuos tablet) as if they were enrolling themselves to a real signature verification system. They were also suggested to practice thoroughly before the actual data collection started. Moreover, contributors were provided the option of not accepting a signature instance if they were not satisfied with it.

In the second session, which was at least one week after the first one, each contributor came again to contribute another 10 genuine signatures. In addition, he/she also contributed four skilled forgeries for each of five other contributors. Skilled forgeries were collected in the following fashion. Using a viewer, a contributor could see genuine signatures of other contributors that he/she would attempt to forge. The viewer could replay the writing sequence of the signatures on the computer screen. Contributors were also advised to practice the skilled forgeries for a few times until they were confident to proceed to the actual data collection.

The signatures are mostly in either English or Chinese.

Signature Data Download

The 40 sets of signature data for system development and evaluation prior to submission are made available to registered participants only. On the "Download" page of our web site, there is a link for downloading the signature database for each of the two tasks. Registered participants should use the login information sent to them for authentication before they can download the data. The signature data should be used exclusively for SVC 2004 but not for other purposes without prior permission.

System Performance Evaluation

For each task, a submitted system will be evaluated on three different data sets. The first set contains signature data from the 40 users released to participants. The second set contains signature data from 60 other users. The third set is just the two sets combined.

Each test will go through the enrollment and matching stages for all users in the data set.

1. Enrollment

Enrollment of each user is done by running the corresponding enrollment program using the following command-line syntax:

```
ENROLL_<team_id> <user_id>.SIG <user_id>.TEM
```

where <user_id>.SIG is a signature index file and <user_id>.TEM is a template file.

For example,

```
ENROLL_210 U4.SIG U4.TEM
```

will enroll five genuine signatures of user 4 to the system submitted by team 10 of signature verification task 2. A template file for user 4 will be generated as a result of this enrollment process.

The signature index file contains filenames for genuine signatures that will be enrolled. For example, U4.SIG may contain the following six lines:

```
5
U4S1.TXT
U4S5.TXT
U4S9.TXT
U4S13.TXT
U4S17.TXT
```

The first line tells us the number of genuine signatures and the following lines list the corresponding filenames. For this competition, five genuine signatures will always be used.

The content format of template files is entirely up to the participants, as long as the corresponding matching program can read the template files correctly.

Notice that all users in the data set will be enrolled into a system before we will move on to the matching stage.

2. Matching

Signatures, including genuine signatures, skilled forgeries, and random forgeries, will be verified by running the matching program in the same way. The command-line syntax is:

```
MATCH_<team_id> <sign_id>.TXT <user_id>.TEM USERS.LOG <user_id>.RES
```

where <sign_id>.TXT is a signature file, <user_id>.TEM is a template file, USERS.LOG is a user enrollment log file, and <user_id>.RES is a result file.

For example,

```
MATCH_210 U4S24.TXT U4.TEM USERS.LOG U4.RES
```

will verify the system submitted by team 10 of task 2 by claiming the identity of user 4 with skilled forgery stored in U4S24.TXT.

USERS.LOG is a file that will be provided to you. It shows a list of users that have been enrolled into the system. The format is similar to that of a signature index file. For example, USERS.LOG may contain the following 41 lines (with some lines omitted):

```
40
U1
U2
...
U40
```

As for the result file, note that all matching results for one user should be kept in a single file. That means new matching results for a user should be appended to existing ones in the result file. For example, U4.RES may contain the following (with some lines omitted):

```
U4S11 0.95
U4S14 0.84
...
U4S24 0.57
U4S36 0.63
...
U3S15 0.14
U9S18 0.17
```

Each line contains the name of a signature followed by a similarity score, between 0 and 1, which indicates the similarity between the signature and the corresponding template. The larger the value is, the more likely the signature tested will be accepted as a genuine signature. Based on these similarity scores, we will compute false rejection rates (FRR) and false acceptance rates (FAR) for different threshold values. Equal error rates (ERR) and ROC curves will also be computed.

Note that the protocol above has been modified from the tentative one described in the Call for Participation announcement.

Code Submission

Both tasks use the same code submission scheme. However, submission for each task should be done independently. Each team may participate in either one or both tasks. The deadline for code submission is 31 December 2003.

To enter a submission, you should go to the "Submission" page of our web site. The same user authentication procedure as for data download is also required here. For each task, each team is required to submit two executable files, one for performing enrollment and the other for matching. Executable files must be for the Windows platform and can run in command-line mode without any graphical user interface.

Under no circumstances will the code be used for purposes other than the SVC 2004 competition. After the competition, all the executable files collected will be destroyed. If a team chooses to use some expiration mechanism, the expiration date should be set to 18 July 2004.