The SecurePhone PDA Database, Experimental Protocol and Automatic Test Procedure for Multimodal User Authentication

A.C. Morris[1], J. Koreman[1], H. Sellahewa[2], J. Ehlers[2], S. Jassim[2], L. Allano[3], S. Garcia-Salicetti[3]

[1]*Saarland University*
{amorris, jkoreman}@coli.uni-saarland.de

[2]*Buckingham University*
{harin.sellahewa, johan-hendrik.ehlers, sabah.jassim}@buckingham.ac.uk

[3]*GET Institut National des Télécommunications*
{lorene.allano, sonia.salicetti, }@int-evry.fr

Version 2.1, 21 February, 2006

1. Introduction

The SecurePhone[1] PDA will provide client authentication on the Qtek2020 PDA[2], using biometric features from voice, face, speaking face and handwritten signature. This document describes both the "SecurePhone PDA database" and the automatic test procedure which comes with it.

For security reasons it is necessary that all client profiles, as well as any world models used during verification, are stored on the PDA SIM card. This imposes a strong limitation on the memory available for client profile storage. The SecurePhone also requires fast enrolment. This does not permit the use of text independent models, which require a large amount of model storage and training data. The test procedure therefore tests only the use of fixed-text prompts. For systems which do not have such constraints, tests based on client-selected passwords may be enabled in future by supplying a text independent speech UBM which has been pretrained on a large external database[3].

The design of this database combines features from both the CSLU database [1] and the BANCA database [2]. The CSLU database is speech only and text dependent. The BANCA database is for speech, face and speaking face, with 6 languages[4] and 3 noise levels. BANCA is quasi text dependent[5]. Video data was recorded at Buckingham University, UK[6]. Chimera[7] signatures were recorded at GET-INT, France[8].

2. Database description

Although it would not be realistic to ask users to attend more than one enrolment session, it is known that training client models with data from separate recording sessions is better able to capture a realistic level of intra-client biometrics variation. Data were therefore recorded in three separate sessions. If tests show a strong advantage for recording in separate sessions then it will be necessary to design the single enrolment session used by the service provider with great care so that the effect of recording in separate sessions can be simulated within one session.

The video database has 60 speakers, 30 male and 30 female, of which 80% are native speakers. There are 3 age groups with 10 males and 10 females in each group. Each speaker was recorded in 3 recording sessions separated by at least one week. Each session comprised 2 indoor recordings and 2 outdoor. The 2 indoor recording (voice, face) conditions were ("light_clean") and ("dark_noisy"). The 2 outside recordings were ("light_noisy") and ("dark_noisy"). Handwritten signature conditions were always good. In order to test the effect of prompt length and prompt type, video recordings were made for 3 types of prompt (5-digit, 10 digit and short phrase), with 6 examples from each prompt type (see Table 1). Video data therefore consists of: (2 genders) x (3 age groups) x (10 subjects per gender per

age group) x (3 well separated sessions) x (2 recording locations) x (2 combinations of voice and face recording conditions) x (3 prompt types) x (6 examples per prompt type); a total of 2x3x10x3x2x2x3x6 = 12,960 recordings.

Session 1 was recorded for use in tests with low grade data, with hand synchronised audio at 8 kHz and visual data at 10 fps. In sessions 2 and 3 the audio and visual data was automatically synchronised[9] and recorded at 44 kHz audio and 19.6 fps visual data. Only the 2 sessions in which data was automatically synchronised are used in the tests described in this document. For voice and face forgery tests, impostorisations are taken as utterances of the same prompt by other speakers (no attempt is made to imitate the client's voice quality).

Twenty chimera signatures were recorded in one session from each of 30 male and 30 female subjects. Forgeries were made by dedicated impostors who were not clients of the signature (or video) database. Each of 4 dedicated impostors forged 20 signatures for each of 15 subjects (of the same gender as the impostor). In this way, every subject in the database has 20 signatures and 20 forged signatures from the same impostor. 100 points were recorded per second, with time data but no pressure or angle data.

All data was recorded at the maximum data rates which the Qtek2020 PDA would permit, i.e. an audio sampling rate of 44 kHz and a video frame rate of 19.6 fps. **The test protocol, however, specifies that all speech should be downsampled to 22 kHz** as this reduces speech feature processing time without loss of speaker verification accuracy[10] and the computational cost of audio data preprocessing is the main load in the SecurePhone multimodal verification process.

It is assumed that in real life the subject is cooperative and that the variation in face size, orientation and pose are therefore limited, with the face being positioned to fit inside a box area on the PDA display. The user is obliged to use all three modalities. If data from any modality is detected as being invalid, the user is requested to move into a more favourable environment.

Tables 1 to 3 show the fields used in video file naming. Voice and face feature data files must use a parallel directory structure with the same filenames, but can have a different extension.

Index	5-digit strings
01	5 3 8 2 4
02	6 2 1 9 7
03	4 2 7 1 3
04	2 8 3 7 6
05	1 9 8 5 4
06	4 5 2 3 9
Index	**10-digit strings**
07	4 3 1 3 8 7 4 6 1 5
08	2 9 2 8 7 3 7 9 3 8
09	5 7 9 2 4 7 9 1 2 6
10	3 9 6 4 6 3 7 6 3 1
11	6 4 2 1 4 7 1 5 3 4
12	1 2 6 1 6 9 2 9 8 1
Index	**phrases**
01	Stop each car if its little
02	Play in the street up ahead
03	A fifth wheel caught speeding
04	Charlie, did you think to measure the tree?
05	Tina got cued to make a quicker escape
06	Here I was in Miami and Illinois

Table 1. The three prompt types, with six examples of each, made in each recording

Gender/age group	Subject index
m_under_30	001-099
m_30_to_45	101-299
m_over_45	201-399
f_under_30	501-599
f_30_to_45	601-699
f_over_45	701-799

Table 2. Subject index used in filenames is unique to each subject

Index	Field name	valid arguments
1	Subject	nnn (see Table 1)
2	Gender	(male, female)
3	Location	(inside, outside)
4	Image condition	(light, dark)
5	Voice condition	(clean, noisy)
6	Session	(02, 03)
7	Prompt type	(numbers, phrases)
8	Prompt index	nn (see Table 1)
9	Image frame dimensions	nRows x nCols (both nnn)
10	Number of image frames	From 0001

Table 3. Fields used in video avi data file names, together with their possible values

Video files are in AVI format[11]. The file name comprises a value for each of the fields specified in Table 3, separated by "_", with extension ".avi". For example, the file path for a male, under 30, session 2, second 5-digit string "62197" in a dark room with no background noise could be:

```
session_1/male_under_30/009_male_inside_dark_clean_01_numbers_02_240x329_0020.avi
```

Raw signature data is in text format. Client signature raw data is in a single directory with file names USmCLn for subject m and client signature n. Imposterised signature raw data is in the same directory with file names USmIMn for imposterised signature n, where mm is 1..60 and n is 1..20. Any derived signature feature data (e.g. including time derivatives, curvature, etc.) must be in a parallel directory structure with the same filenames but can have a different extension.

3. Test procedure design

With BANCA seven standard tests are specified in terms of different selections of training and test data. This is convenient because it encourages different sites to publish comparable results. However, with BANCA the user is left to interpret the somewhat complex test design and implement the test procedure themselves according to this interpretation, which is time consuming and provides great scope for errors. By contrast, the PDA database does not specify a small number of standard tests, but comes complete with its own automated and flexible test procedure (test options are summarised in Table 6). While the selection of training and test data is hidden from the user, the user still has unlimited control over the core test functions of client model training, world model training, scores generation and scores fusion[12]. In this section we describe the design of the automatic test procedure.

3.1 Division of subject data into three disjoint sets

As with BANCA, and whether or not a UBM is used by each modality[13], for every test the data is divided into 3 disjoint sets of subjects (24 for UBM, 18 for g1 and 18 for g2), in which each set has the same number of subjects from each gender, age and recording condition groups and, as far as possible, the same proportion of native to non native speakers. During testing, decision thresholds are tuned to give best test performance for g1. These a-posteriori thresholds (tuned to the test data, which of course is not possible in a working system) are then used as a-priori thresholds (i.e. fixed values, as would be

used in a real application) for tests on g2. This procedure is then repeated with g1 and g2 reversed. Scores using a-priori (i.e. realistic) thresholds are then reported for the average over these two tests.

Gender	UBM	g1	g2
male	(0,1,2)01-04	(0,1,2)05-07	(0,1,2)08-10
female	(5,6,7)01-04	(5,6,7)05-07	(5,6,7)08-10

Table 4. Division of subjects into groups for UBM training, for g1 and for g2.
Data is specified by subject index as defined in Table 2.

3.2 Universal background models

UBMs (Universal Background Models)[14] are widely used in voice authentication, but less often, if at all, in face or signature authentication. A UBM can be used for client model initialisation and/or for score normalisation. These are two separate choices which are both permitted in the test protocol.

The UBM for each of these purposes may be gender dependent or independent. To limit the number of possible tests to be considered we assume that: the UBM, if used, is age independent and is trained on data from one test prompt only (the UBM is text dependent). All of the data from the prompt being tested, from 24 of the 60 subjects, is used for UBM training (see Table 4).

3.3 Choice of data for training and testing

Direct comparison of results between tests is only possible when they use the same test data. As shown in Table 5, all tests use 4 recordings from each subject but data divisions 1 and 2 both use 4 tests from session 3 while divisions 3 and 4 use 2 tests from each of sessions 2 and 3. Comparison is therefore only possible between tests using divisions 1 and 2, or 3 and 4. Video impostor tests for each client are from 3 subjects of the same gender (2 in the same age group and the first from the age group below, or above if there is none below) and the first 2 subjects of the opposite gender and same age group. This results in 5*4=20 impostor tests from the prompt being tested. Voice and face model training uses data from one prompt only. Signatures do not use prompts.

Data division D	Training		Testing	
	Session 2	Session 3	Session 2	Session 3
1	I1 I2	none	none	I1 I2 O1 O2
2	I1 I2 O1 O2	none	none	I1 I2 O1 O2
3	I1 I2	none	O1 O2	I1 I2
4	I1 I2	O1 O2	O1 O2	I1 I2
5	I1 I2	none	O1 O2	O1 O2
6	I1 I2	I1 I2	O1 O2	O1 O2

Table 5. Choice of data divisions for training and testing models for voice and face.
"I1, I2" signifies recording under condition (inside, light, clean), (inside, dark noisy).
"O1, O2" signifies recording under condition (outside, light, noisy), (outside, dark noisy).
Of the 20 chimera signatures for each subject, 8 are used for training with D1 and D3, 16 with D2 and D4.
Tests 1 & 2 look at effect of increasing amount of training data within one enrolment session.
Tests 3 & 4 look at effect of increasing amount of training data by using two enrolment sessions.
Tests 5 & 6 look at effect of increasing amount of training data by using two inside enrolment sessions

3.4 Fixed-prompt tests

For the SecurePhone PDA application, in which client biometric profiles must be stored on the PDA SIM card, the memory available for profile storage is very limited. Because of this, the variety of data variation which it can model (due for example to different phonemes and noise conditions for voice, or different face-camera angle and lighting conditions for face) is also very limited. In this context text dependent systems, having to model little phonetic variation, are more accurate than text independent systems, which must model every different sound in the language. For a given size of GMM, verification accuracy increases as test data variation (not due to differences in subject biometrics)

decreases. However, very short prompts are not optimal because different phonemes serve to differentiate different speakers under different conditions, so that some phonetic redundancy is beneficial. This database has three prompt types but for fixed-prompt tests each test will use data from just one prompt. Fixed prompt systems have two important advantages for the PDA. Firstly, the pre-recorded UBM and the client profile, which must both be stored on the PDA SIM, are very small[15]. Secondly, enrolment requires only a small number of repetitions of the same short phrase.

3.5 Fusion tests

In scores-generation mode the automated test procedure (see Section 4) will create a set of match-scores for one biometric modality at a time. One scores file is generated for each subtest (one for each prompt example and, if gender dependent thresholds are used, one for each gender). Once match-scores have been generated for more than one biometrics mode, scores-fusion mode can be used to fuse these scores into a single scores set. In fusion mode the user supplies one routine to model the client and impostor scores distributions (optional), and another to combine the scores from each modality into a single scores set. Performance evaluation from the fused score set can then be evaluated using the same procedure which is used to evaluate unimodal scores.

3.6 Signature data coupling

Some believe that with a chimera database it is good practice to repeat fusion tests using a large number of different couplings between the chimera subjects and the subjects with which they are being matched. This calls for fusion experiments to be repeated a large number of times by everyone who uses the database. Although fusion processing is generally computationally light compared to the processing required to produce the scores to be fused, the position taken by the present automatic test procedure is that, as for all practical purposes there is no correlation between a person's signature and other biometrics (here, their facial appearance or the quality of their voice), there is no more theoretical reason for testing fusion with multiple couplings than there would be had the chimera data not been chimera but had been recorded from the subjects whose other biometrics were recorded. The present test procedure therefore only tests one coupling of signature with voice/face subjects.

3.7 Results reported

Once a set of match-scores have been generated (using scores-generation mode), the test procedure can be used in performance-evaluation mode (with thresholds set either a-posteriori or a-priori) to create the following files.

- A summary of all test results, comprising EER as well as (WER, FAR and FRR) for the CFA/CFR cost ratio R = (0.1, 1.0, 10.0). Separate results are reported for each prompt. Results are also reported for the average over all 6 prompts. Tests with gender dependent thresholds will report results for each gender. **See example results file in Appendix C** for tests using the setup.scr file shown in Appendix A. Results are in text file:

  $PDA_WRKDIR/TName/RESULTS_A_POSTERIORI or RESULTS_A_PRIORI (TName is test name, as specified in captions to Tables 6 and 7).

- A DET plot for each subtest (in eps format). **See example in Appendix B**

  $PDA_WRKDIR/TName/g*/results/det_plot_[n].eps (n = 1...number of subtests)

- A text scores file for each subtest. Scores are in text files:

  $PDA_WRKDIR/TName/g*/results/scores_[n] (n = 1...number of subtests)

  Example voice mode score line (followed by field descriptions):

  ```
  005_male    005_male_inside_light_clean_03_numbers_07_240x320_0069    0.9687000
  MODEL ID    TEST ID                                                   TEST SCORE
  ```

  Example signatures mode score line (followed by field descriptions):

  ```
  US27        US27CL14    2.8567200
  MODEL ID    TEST ID     TEST SCORE
  ```

  These scores can be used later to run multimodal fusion.

4. Automatic test procedure

Bash scripts [3, 4] are supplied that will run tests automatically under Linux or Unix operating systems. For the purpose of practical implementation on the PDA it is necessary to optimise a number of separate design choices. Each of these is listed in Table 6.

4.1 Steps involved in running PDA database tests

1. Create a full set of feature data for the modality being tested in root directory DIR, with a parallel directory structure to that in which the avi data is installed. Export PDA_DATDIR=DIR in the setup script.
2. If frame weighting is required (PDA_W=1) then also create, in a separate directory, a parallel set of frame weights data (one file per feature file, with one weight per frame).
3. Install the test scripts in the directory which you then export as PDA_SCRDIR.
4. Scripts use bash, octave & C. C programs can be compiled using "gcc -o prog prog.c".
5. Create the following script files for model training and example testing for each modality used ("mode" below is one of "voice", "face" or "signature"). All script files must be in the same directory PDA_SCRDIR in which the test procedure is installed.
   - `setup.scr` : *script to set up required PDA_ and user specific USR_ test parameters.*
   - `user_train_client_model_[mode].scr` : *perform client model training.*
   - `user_train_world_model_[mode].scr` : *do world model training (if UBM used).*
   - `user_test_example_[mode].scr` : *output score for given model and test example.*
   - `user_model_and_normalise_scores.scr:` *train client & impostor scores models and apply any required scores normalisation.*
   - `user_combine_all_scores.scr:` *input one scores file from each modality and output a single combined scores file.*

   The user scripts can be modelled on the voice mode user scripts which come with the test procedure. You can put what you like in these scripts and make use of unlimited "user" parameters, such as USR_NUMGSN. If frame weighting is selected (PDA_W=1) then add "_wts" above before the .scr extension for the user_train or user_test scripts.
6. The test procedure is then run as follows.
   - `source $PDA_SCRDIR/setup.scr`
   - `$PDA_SCRDIR/pda_test.scr`

   This will generate all of the results reported in Section 3.7 above.

Tag	value range	Meaning
M	0,1,2,3	Biometric mode = fusion, voice, face, signature
T	1,2,3	Prompt type = 5-digit, 10-digit, phrase
D	1,2,3,4,5,6	Train/test data selection (see Table 5)
G	0,1	UBM for model initialisation is gender indep./dep.
S	0,1	UBM for score normalisation is gender indep./dep.
H	0,1	Thresholds are gender indep./dep.
U	0,1,2	No UBM, text dep. UBM, (text indep. UBM not implem.)
W	0,1	Frame weights not used/used

Table 6. Key to switches used for scores generation

Each of these flags, and the value assigned to it, is used to construct a test name identifier, e.g. TName = M1.T2.D4.S0.H1.U1.W0. The bash variable corresponding to flag x is PDA_x

PDA_A	Meaning
0	Run tests to generate a (match-)scores file for each subtest
1	Get performance statistics from existing scores set, using a-posteriori thresholds
2	Get performance statistics from existing scores set, using a-priori thresholds
3	Fuse up to 3 sets of existing scores files into 1 set

Table 7. Key to actions specified by the PDA_A flag

PDA_A=3 is equivalent to PDA_M=0. TName can be specified using PDA_TSTNAM=TName. Otherwise the test name will be as for scores generation. In fusion mode fused scores are output to TName and up to 3 scores input subdirectories can be specified using PDA_FUS1, PDA_FUS2, PDA_FUS3.

All of the switches in Tables 6 & 7 must be specified (exported) before the test starts. The bash variable name for switch x is PDA_x. **See example set up script in Appendix A**. As well as these values, a user can specify any number of user parameters for use by the user scripts which must be provided (see below). It is recommended that all exported user parameters should start with USR_ because the values assigned to these will all be listed in the RESULTS file for later reference.

4.2 Example run of multiple tests

If you don't export PDA_TSTNAM to specify the test subdirectory name then the test name TName will be constructed from the values given to each of the test PDA_ switches in Table 6. If the bash variable PDA_TSTEXT is set then its value will be appended as an extra field at the end of TName. Output for scores-fusion mode, and all input and output for scores-generation and performance-evaluation modes, will be stored within a directory named $PDA_WRKDIR/TName. For scores-generation mode, if this directory already exists when the test starts then it will first be deleted. A single test can be run as follows.

```
export PDA_SCRDIR=/proj/SecurePhone/PDA_SpkrRec/scripts  # export scripts directory name
source $PDA_SCRDIR/setup.scr     # export required and user defined variables
$PDA_SCRDIR/pda_test.scr         # run test
```

The following runs multiple tests, storing the results for each test in a separate test directory.

```
export PDA_SCRDIR=/proj/SecurePhone/PDA_SpkrRec/scripts
source $PDA_SCRDIR/setup.scr

for NG in 32 64 128 256; do
   export USR_NUMGSN=$NG         # number of Gaussians in every GMM
   for G in 0 1; do
      export PDA_G=$G            # gender dependence of UBM for client model initialisation
      for T in 1 2 3; do
         export PDA_T=$T         # prompt type
         for D in 3 4; do
            export PDA_D=$D      # training data selection
            for W in 0 1; do
               export PDA_W=$W   # use frame weights?
               export PDA_TSTEXT=Ng$USR_NUMGSN  # ensures that all results files are kept
               export PDA_A=0    # generate match-scores
               $PDA_SCRDIR/pda_test.scr
               export PDA_A=2    # evaluate performance using a-posteriori thresholds
               $PDA_SCRDIR/pda_test.scr
            done
         done
      done
   done
done
```

5. Discussion

The SecurePhone PDA database and test procedure provides a tool for the automated and efficient development of multimodal user authentication applications on the Qtek2020 PDA. It allows the user freedom to use any model training, scores generation and scores fusion techniques, while automating the selection of training and test data and results reporting so that results obtained are strictly

standardised and therefore comparable with those obtained by other users. It is therefore well suited to both professional and academic use.

The supplied test script enables all of the fixed-prompt tests described above to be run automatically. User selected password tests will not be available until the proposed text independent voice UBM (trained on external data) is released.

As can be seen from the example in Section 4.2, once a full PDA database feature set has been created it is easy to run multiple tests in order to find the optimal configuration of test switches as well as the optimum value for any number of user specified model parameters. Exhaustive search is, of course, not generally a recommended optimisation technique. One could instead first hold all user parameters constant and optimise the test configuration, then hold the test configuration constant and optimise one or more of the user parameters as a time.

Test identifier	Num Gaussians		WER			FAR			FRR		
		EER	R=0.1	1.0	10.0	R=0.1	1.0	10.0	R=0.1	1.0	10.0
M1.T1.D4.G0.S0.H1.U1.C1.W0	128	4.14	3.31	4.53	2.25	13.24	3.50	1.11	2.31	5.56	13.66
M1.T2.D4.G0.S0.H1.U1.C1.W0	128	2.51	3.32	3.51	1.47	6.39	1.81	0.60	3.01	5.21	10.19
M1.T3.D4.G0.S0.H1.U1.C1.W1	256	4.37	3.69	5.10	2.23	12.80	3.73	0.76	2.78	6.48	16.90

Table 8. Summary of best performing test configurations for each of the 3 prompt types (5-digit, 10-digit, phrase).
User parameters were USR_NUMGSN=128, USR_MXFLOR=0.03, USR_VAFLOR=0.8, USR_PRIOWT=0.0 (tuned on BANCA test G)

Test identifier	Num Gaussians		WER			FAR			FRR		
		EER	R=0.1	1.0	10.0	R=0.1	1.0	10.0	R=0.1	1.0	10.0
M1.T2.D4.G0.S0.H1.U1.C1.W0	128	2.16	3.34	3.06	1.65	5.44	1.83	0.95	3.12	4.28	8.91

Table 9. Summary of best performing test configurations prompt type 2 (10-digits).
Tuned user parameters are USR_NUMGSN=128, USR_MXFLOR=0.06, USR_VAFLOR=0.08, USR_PRIOWT=0.2

Results obtained by this approach are shown in Tables 8 and 9. Table 8 shows results for the best performing test configurations for voice mode verification for each of the three prompt types (top row is for 5-digit prompts, etc.) when the user parameters are fixed at values optimised for BANCA.

Table 9 shows corresponding results for prompt type T2 (10-digits) after further optimisation of the USR_ parameters with the above optimised PDA_ parameters held constant.

Test identifier	Num Gaussians	Fusion weight		WER			FAR			FRR		
			EER	R=0.1	1.0	10.0	R=0.1	1.0	10.0	R=0.1	1.0	10.0
M1.T1.D4.G0.S0.H0.U1.W0	100	0.41	4.00	3.41	4.71	1.88	12.01	3.87	0.67	2.55	5.56	14.00
M2.T1.D4.G0.S0.H0.U1.W1	1	0.25	27.82	10.76	28.54	9.31	87.13	28.38	1.60	3.12	28.70	86.46
M3.T1.D4.G0.S0.H0.U1.W0	100	0.34	6.19	3.76	5.90	8.71	13.61	6.94	4.31	2.78	4.86	52.78
Fused_wts_0.41_0.25_0.34	-	-	0.83	1.99	0.97	1.20	15.00	1.25	1.25	0.69	0.69	0.69

Table 10. Performance per modality and after scores fusion by posteriors normalisation followed by weighted sum

Table 10 shows the effect of scores fusion by a simple method of scores fusion. In this case log-likelihood ratio scores from each modality were first converted to posterior client probabilities and then fused in a weighted sum. Optimal weights were found by using the test script to evaluate performance for every possible combination of 3 positive weights which sum to 1, and selecting the optimum weights as those which result in the minimum average HTER score (WER for R=1.0) using a-priori acceptance thresholds. Another more powerful fusion technique is to model the joint (normalised-) score distribution using a GMM and defining the fused score as the client posterior probability given the scores from each modality.

Appendix A. Example user setup script file

File: $PDA_SCRDIR/setup.scr

```
#!/bin/bash -eu
#
# BASH SCRIPT TO SET UP PDA TEST PROTOCOL FOR USE AT SARLAND UNIVERSITY
#-----------------------------------------------------------------------

# REQUIRED PARAMETERS
export PDA_M=1      # biometrics mode
export PDA_T=2      # prompt type
export PDA_D=4      # training data selection
export PDA_G=0      # gender dependent UBM for client model initialisation
export PDA_S=0      # gender dependent UBM for score normalisation
export PDA_H=1      # gender dependent thresholds
export PDA_U=1      # UBM type
export PDA_W=0      # use frame weights?
Export PDA_A=0      # action = scores-generation
export PDA_SCRDIR=/proj/SecurePhone/PDA_SpkrRec/scripts  # scripts directory
export PDA_WRKDIR=/proj/SecurePhone/PDA_SpkrRec          # working directory
export PDA_DATDIR=/speech_22_khz_MFCC_Z_D_38_nocutoff    # feature data directory
export PDA_WTSDIR=/speech_22_khz_MFCC_0_20_nocutoff_weights # weights data directory
export PDA_EXTFTR=ftr          # feature data extension
export PDA_EXTWTS=spw          # weights data extension
export PDA_TSTEXT=none         # test name extension
export PDA_TSTNAM=none         # default test name constructed from test switches used

# USER PARAMETERS (NO DEFAULT VALUES)
export USR_NUMGSN=128          # number of Gaussians
export USR_NUMINP=38           # number of feature dimensions
export USR_VAFLOR=0.8          # variance floor
export USR_MXFLOR=0.03         # minimum Gaussian weight
export USR_PRIOWT=0.0          # UBM prior weight with adaptive training
export USR_SEED=123            # random number seed
export USR_COMENT=none         # comment for results report
export USR_NORMAL=nonorm       # apply feature mu/sd normalisation?
```

Appendix B. Example DET plot

The prompt number “numbers_08” arises from this plot being for the 2nd of the 6 10-digit prompts, which have names “numbers_[nn]”, where [nn] is 07..12 (as specified in Table 1). The name “det_plot_4” arises because this is the 4th of 2 x 6 gender dependent subtests for each of the 6 10-digit prompts which are tested when PDA_T=2.

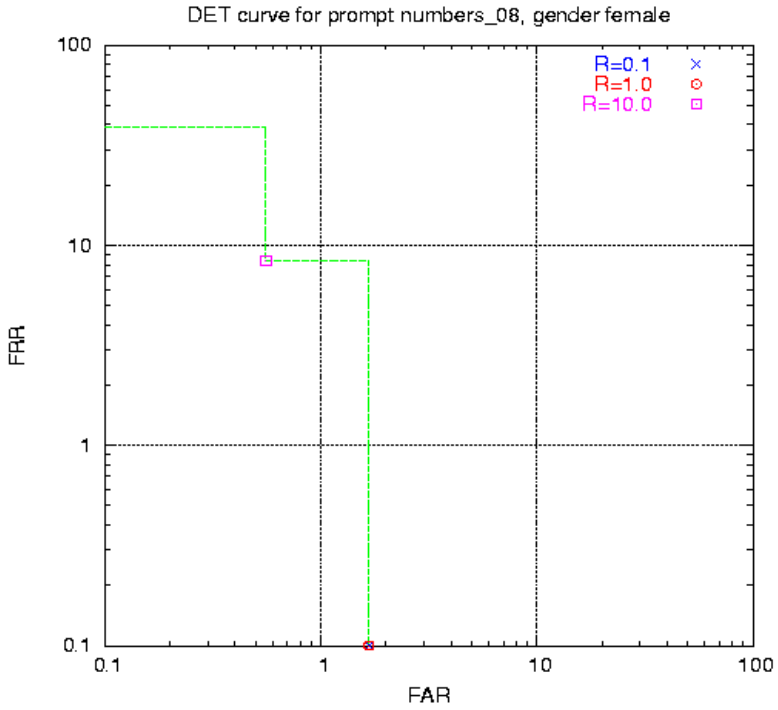

Fig D.1 Example DET plot $PDA_WRKDIR/TName/g1/results/det_plot_4.eps

Appendix C. Example results from scores-generation and performance-evaluation mode

File: $PDA_WRKDIR/TName/LOG, produced by scores-generation mode

```
REQUIRED PARAMETERS
-------------------
PDA_A=2
PDA_C=1
PDA_COMENT=none
PDA_D=4
PDA_DATDIR=/proj/PhonLSV/tmp/PDA/speech_22_khz_MFCC_Z_D_38_nocutoff
PDA_EXTFTR=ftr
PDA_EXTWTS=spw
PDA_G=0
PDA_H=1
PDA_LSTDIR=/proj/SecurePhone/PDA_SpkrRec/Prm.M1.T2.D4.G0.S0.H1.U1.C1.W0/lists
PDA_M=1
PDA_MDLDIR=/proj/SecurePhone/PDA_SpkrRec/Prm.M1.T2.D4.G0.S0.H1.U1.C1.W0/models
PDA_TSTEXT=none
PDA_RESULT=/proj/SecurePhone/PDA_SpkrRec/Prm.M1.T2.D4.G0.S0.H1.U1.C1.W0/RESULTS
PDA_S=0
PDA_SCRDIR=/proj/SecurePhone/PDA_SpkrRec/scripts
PDA_T=2
PDA_TSTDIR=/proj/SecurePhone/PDA_SpkrRec/Prm.M1.T2.D4.G0.S0.H1.U1.C1.W0
PDA_TSTNAM=Prm.M1.T2.D4.G0.S0.H1.U1.C1.W0
PDA_U=1
PDA_W=0
PDA_WAVDIR=/export/blue/speech_22_khz
PDA_WRKDIR=/proj/SecurePhone/PDA_SpkrRec
PDA_WTSDIR=/proj/PhonLSV/tmp/PDA/speech_22_khz_MFCC_0_20_nocutoff_weights

USER PARAMETERS
---------------
USR_MXFLOR=0.03
USR_NORMAL=nonorm
USR_NUMGSN=128
USR_NUMINP=38
USR_PRIOWT=0.0
USR_SEED=123
USR_VAFLOR=0.8
```

File: $PDA_WRKDIR/TName/RESULTS_A_PRIORI, produced by scores-generation mode

```
-----------------------------------------------------------------------------------------------
Results for PDADB verification test: Prm.M1.T2.D4.G0.S0.H1.U1.C1.W0
-----------------------------------------------------------------------------------------------
             WER                    FAR                    FRR                    Threshold
 G    EER    0.10   1.00  10.00     0.10   1.00  10.00     0.10   1.00  10.00     0.10   1.00  10.00
prompt numbers_07, gender = male
 1   1.11    0.20   1.11   1.52     2.22   2.22   0.00     0.00   0.00  16.67    -0.65  -0.65  -0.21
 2   2.78    5.10   3.06   1.01     0.56   0.56   0.56     5.56   5.56   5.56    -0.42  -0.42  -0.42
 0   1.94    2.65   2.08   1.26     1.39   1.39   0.28     2.78   2.78  11.11
prompt numbers_07, gender = female
 1   2.78    2.78   2.78   3.28     2.78   2.78   2.78     2.78   2.78   8.33    -0.89  -0.89  -0.68
 2   1.19    1.01   1.67   1.26    11.11   0.56   0.00     0.00   2.78  13.89    -1.19  -0.81  -0.28
 0   1.98    1.89   2.22   2.27     6.94   1.67   1.39     1.39   2.78  11.11
prompt numbers_08, gender = male
 1   0.00    5.05   2.78   1.52     0.00   0.00   0.00     5.56   5.56  16.67    -0.37  -0.37   0.03
 2   1.11    0.10   0.56   1.01     1.11   1.11   1.11     0.00   0.00   0.00    -0.42  -0.42  -0.42
 0   0.56    2.58   1.67   1.26     0.56   0.56   0.56     2.78   2.78   8.33
prompt numbers_08, gender = female
 1   3.33    3.13   9.17   3.03     6.67   1.67   0.56     2.78  16.67  27.78    -1.08  -0.73  -0.22
 2   2.78    1.06   5.28   1.77    11.67   7.78   0.56     0.00   2.78  13.89    -1.09  -1.01  -0.46
 0   3.06    2.10   7.22   2.40     9.17   4.72   0.56     1.39   9.72  20.83
prompt numbers_09, gender = male
 1   3.33    7.58   4.17   0.76     0.00   0.00   0.00     8.33   8.33   8.33    -0.39  -0.39  -0.39
 2   0.00    0.30   1.67   0.00     3.33   3.33   0.00     0.00   0.00   0.00    -0.76  -0.76  -0.42
 0   1.67    3.94   2.92   0.38     1.67   1.67   0.00     4.17   4.17   4.17
prompt numbers_09, gender = female
 1   0.56    0.35   1.39   0.25     3.89   0.00   0.00     0.00   2.78   2.78    -1.20  -0.54  -0.54
 2   2.78    2.68   2.22   1.01     1.67   1.67   0.00     2.78   2.78  11.11    -0.62  -0.62  -0.44
 0   1.67    1.52   1.81   0.63     2.78   0.83   0.00     1.39   2.78   6.94
prompt numbers_10, gender = male
 1   5.56    2.98   8.33   2.53    32.78   0.00   0.00     0.00  16.67  27.78    -1.52  -0.36   0.00
 2   4.44    4.19   4.17   2.02    18.33   2.78   1.67     2.78   5.56   5.56    -1.22  -0.62  -0.37
 0   5.00    3.59   6.25   2.27    25.56   1.39   0.83     1.39  11.11  16.67
prompt numbers_10, gender = female
 1   2.78    1.21   2.22   1.77    13.33   1.67   1.67     0.00   2.78   2.78    -1.45  -0.61  -0.61
 2   5.56    5.25   3.89   1.01     2.22   2.22   0.00     5.56   5.56  11.11    -0.77  -0.77  -0.49
 0   4.17    3.23   3.06   1.39     7.78   1.94   0.83     2.78   4.17   6.94
prompt numbers_11, gender = male
 1   5.56   22.78  12.78   2.78     0.56   0.56   0.56    25.00  25.00  25.00    -0.33  -0.33  -0.33
 2   0.00    1.77   0.83   1.52    19.44   1.67   1.67     0.00   0.00   0.00    -1.19  -0.49  -0.49
 0   2.78   12.27   6.81   2.15    10.00   1.11   1.11    12.50  12.50  12.50
prompt numbers_11, gender = female
 1   5.56    5.35   4.44   2.27     3.33   3.33   1.11     5.56   5.56  13.89    -0.88  -0.88  -0.55
 2   2.43    1.06   2.50   2.27    11.67   5.00   1.11     0.00   0.00  13.89    -1.15  -0.97  -0.43
 0   3.99    3.21   3.47   2.27     7.50   4.17   1.11     2.78   2.78  13.89
prompt numbers_12, gender = male
 1   1.11    2.53   1.39   0.25     0.00   0.00   0.00     2.78   2.78   2.78    -0.42  -0.42  -0.42
 2   0.56    0.10   0.56   1.01     1.11   1.11   0.56     0.00   0.00   5.56    -0.52  -0.52  -0.33
 0   0.83    1.31   0.97   0.63     0.56   0.56   0.28     1.39   1.39   4.17
prompt numbers_12, gender = female
 1   2.78    2.78   2.78   0.25     2.78   2.78   0.00     2.78   2.78   2.78    -0.95  -0.95  -0.53
 2   2.22    0.25   4.44   1.26     2.78   0.56   0.56     0.00   8.33   8.33    -0.99  -0.53  -0.53
 0   2.50    1.52   3.61   0.76     2.78   1.67   0.28     1.39   5.56   5.56
average over all prompts and both genders
 1   2.87    4.73   4.44   1.68     5.69   1.25   0.56     4.63   7.64  12.96
 2   2.15    1.91   2.57   1.26     7.08   2.36   0.65     1.39   2.78   7.41
 0   2.51    3.32   3.51   1.47     6.39   1.81   0.60     3.01   5.21  10.19
-----------------------------------------------------------------------------------------------
```

Appendix D. Example user training and test script files

All of the bash variables in the following example user_ files, besides the USR_ variables, are available to every user. The user can export as many USR_ variables as required.

File: user_train_client_model_voice.scr

```
#!/bin/bash -eu
#
# bash flags above signify:
# -e: exit on unsuccessful function return
# -u: exit on attempt to use undefined variable
# -x: show all command lines
#
# File: train_client_model.scr

# TRAIN SPEAKER MODEL (NO FRAME WEIGHTS)
#
# Method
# - GMM model
# - map adaptation from UBM (world model)
# - frame weights used to weight frame log likelihoods

PROG=/proj/SecurePhone/Progs/Torch/Torch3_nergal/examples/generatives/Linux_OPT_FLOAT/nergal_gmm_wts
if (test $USR_NORMAL == "norm") then NORMAL=-norm; else NORMAL=""; fi

if (test $PDA_U == 1) then
   nice -19 $PROG --retrain $WLDMOD_INIT \
    $PDA_LSTDIR/list_client_train $PDA_LSTDIR/list_client_train_wts \
    -weight_on_prior $USR_PRIOWT -e 1e-5 -learn_means -prior $USR_MXFLOR \
    -threshold $USR_VAFLOR -save $SPKMOD -dir $JOBDIR/convergence -bin \
    -n_inputs $USR_NUMINP -n_gaussians $USR_NUMGSN -iterg 100 $NORMAL
else
   nice -19 $PROG $PDA_LSTDIR/list_client_train  $PDA_LSTDIR/list_client_train_wts \
    -prior $USR_MXFLOR -threshold $USR_VAFLOR -n_inputs $USR_NUMINP -n_gaussians $USR_NUMGSN \
    -save $SPKMOD -seed 123 -bin -iterk 100 -iterg 100 -dir $JOBDIR/convergence $NORMAL
fi

exit
```

File: user_train_world_model_voice.scr

```
#!/bin/bash -eu
#
# bash flags above signify:
# -e: exit on unsuccessful function return
# -u: exit on attempt to use undefined variable
# -x: show all command lines
#
# TRAIN WORLD MODEL (NO FRAME WEIGTS)
#
# Method
# - GMM model
# - k-means clustering followed by EM iteration
# - frame weights used to weight frame log likelihoods

PROG=/proj/SecurePhone/Progs/Torch/Torch3_nergal/examples/generatives/Linux_OPT_FLOAT/nergal_gmm
if (test $USR_NORMAL == "norm") then NORMAL=-norm; else NORMAL=""; fi

nice -19 $PROG $PDA_LIST_UBM_FTR -prior $USR_MXFLOR -threshold $USR_VAFLOR \
 -n_inputs $USR_NUMINP -n_gaussians $USR_NUMGSN -save $WLDMOD -seed 123 -bin -iterk 100 \
 -iterg 100 -dir $JOBDIR/convergence $NORMAL

exit
```

File: user_test_example_voice.scr

```
#!/bin/bash -eu
#
# User-supplied script file to use given test data and given test model
# to generate and print out a test score.

PROG=/proj/SecurePhone/Progs/Torch/Torch3_nergal/examples/generatives/Linux_OPT_FLOAT/nergal_gmm

FNAME=$1
TEST_MODEL=$2

if (test $USR_NORMAL == "norm") then NORMAL=-norm; else NORMAL=""; fi

# append score to new line in $SCORE_MATRIX for each test file
nice -19 $PROG --test $TEST_MODEL $FNAME -bin -n_inputs $USR_NUMINP -n_gaussians $USR_NUMGSN $NORMAL

exit
```

When frame weights are in use the corresponding script filenames required are the same as those above but with "_wts" added to the end (before the extension). For training or testing with frame weights, within the script file each input feature file list is complemented by a similar frame-weights file list by the same name but with "_wts" appended.

Appendix E. Example user scores modelling and fusion script files

File: user_model_and_normalise_scores.scr

```
#!/bin/bash -eu
#!/bin/bash -eu
#
###############################################################################################################
#
# COMMAND FILE TO MODEL AND NORMALISE COMPLETE SCORES SET
#
# There is one input scores file from each expert for one (prompt, gender) subtest.
#
# Some fusion rules require no scores modelling, in which case this routine is blank.
#
# A typical scores modelling would be to train a GMM probability density distribution for the
# client test scores, and one for the impostor test scores (possibly using a UBM for
# initialisation).
#
# Score normalisation (such a Z-norm , T-norm, Min-Max or conversion to posterior client
# probabilities) would be applied before scores modelling.
#
###############################################################################################################

MDLDIR=$1       # directory to store output scores models
NFUS=$2         # number of modalities
SL1=$3          # scores list 1
SL2=$4          # scores list 2
SL3=$5          # scores list 3

# octave can read scores file using this routine (get_scores.m is used by use_ltholds_text.scr)
# [score_client,score_impost,nTests_client,nTests_impost] = get_scores(scoNam);

# C program can read scores file using this (read_scores() is a subroutine in getDet.c)
# bool read_scores(char *scoFil, float *score_client, float *score_impost,int nTests_client,int nTests_impost)

exit
```

File: user_combine_all_scores.scr

```
Exit
#!/bin/bash -eu
#
###############################################################################################################
#
# COMMAND FILE TO GENERATE FUSED SCORES
#
# There is a separate scores file for each subtest (prompt example and gender (if gender dependent
# thresholds were used). This routine combines one input scores file from each expert for just one
# of these subtests.
#
# In this version, for fusion by weighted sum, the user must export one weight for each expert.
#
# The user fusion routine is passed whole scores files, rather than one line from each at a time,
# in order to greatly increase fusion processing.
#
# The fusion routine outputs fused scores to stdout, which is redirected by the calling routine.
#
###############################################################################################################

MDLDIR=$1         # directory to find scores models
NFUS=$2           # number of modalities
SL1=$3            # scores list 1
SL2=$4            # scores list 2, or "none" if no second modality
SL3=$5            # scores list 3, or "none" if no third modality

WT1=${USR_WT1:-0} # user defined weight for modality 1
WT2=${USR_WT2:-0} # user defined weight for modality 1 (or 0 if not set)
WT3=${USR_WT3:-0} # user defined weight for modality 1 (or 0 if not set)

# write header line to fused scores output
echo -n "Fused scores from"
if (test ${PDA_FUS1:-none} != none) then echo -n "  $PDA_FUS1"; fi
if (test ${PDA_FUS2:-none} != none) then echo -n ", $PDA_FUS2"; fi
if (test ${PDA_FUS3:-none} != none) then echo -n ", $PDA_FUS3"; fi
echo ""
if (test $PDA_H == 1) then NUMTESTS=216; else NUMTESTS=432; fi

# fuse list of scores from each modality into single list of scores
$PDA_SCRDIR/combine_scores_wsum $NFUS $NUMTESTS $SL1 $SL2 $SL3 $WT1 $WT2 $WT3

exit
```

References

1. Cole, R., Noel, M. & Noel V., "The CSLU speaker recognition corpus", ICSLP'98, pp.3167-3170, 1998
2. Porée, F., Mariéthoz, J., Bengio, S. & Bimbot, F., " The BANCA database and experimental protocol for speaker verification", IDIAP-RR 02-13, 2002
3. GNU Bash Reference Manual, http:://www.network-theory.co.uk/docs/bashref/index.html
4. Advanced Bash Scripting Guide, http://www.tldp.org/LPD/abs/html

---

[1] EC IST-2002-506883 project, "Secure Contracts Signed by Mobile Phone".

[2] The trade name of the PDA selected for the SecurePhone project varies from country to country. In the UK it is called the XDA II, while in France, Germany and Italy it is called the Qtek2020. It has a writing tablet and a camera as well as the usual voice input and graphic display. At the time it was required this was considered to be the most suitable device in the market, although the camera is in the back, which is not suitable for recording the user's face while speaking a prompt read from the screen, and the memory capacity of the SIM card was insufficient to permit storage of the highest quality client biometric profiles. Both of these problems are overcome in new PDA models already nearing production.

[3] This model has not yet been provided because there are some difficulties in finding a suitable large vocabulary database which could be adapted to the recording conditions of the PDA and for which the resulting trained model would be legally open for circulation.

[4] Data for the original BANCA video database was recorded on digital tapes and not all of the original video image frames were provided with the database that was released. However, the English part of the database was later extended (within the SecurePhone project) to provide the complete original video data at 25 fps. This permits tests with "speaking face" mode, as well as important liveness tests which ensure that the audio and visual data correctly correspond.

[5] BANCA is not fully text dependent because, as well as the subjects (fictitious) name and address, each phrase also contains an arbitrary 12 digit number which is not the same number when spoken by impostors.

[6] The person responsible for the SecurePhone work at Buckingham University was Professor Sabah Jassim, email sabah.jassim@buckingham.ac.uk

[7] The people whose signatures were recorded were not the same as any of the people used for video recording.

[8] The person responsible for the SecurePhone work at GET INT was Dr. Sonia Salicetti, email sonia.salicatti@int-evry.fr

[9] Common use of the term "video" refers to audio-visual media. To avoid confusion we refer in this document to audio data as "audio" or "voice" data and visual data not as "video" but as "visual" or "face" data.

[10] Tests on the 32 kHz BANCA database showed that verification accuracy falls significantly for sampling frequency below 22 kHz.

[11] Audio and visual streams can be separated using a number of open source software tools. For example, for linux there is mplayer (e.g. to extract audio data from .avi: mplayer avifile -dumpaudio -dumpfile audiofile), and for Windows, VirtualDub (http://www.virtualdub.org).

[12] Multimodal fusion can take place at one or more different stages of processing. The scores generated separately for each modality by the test procedure here are suitable only for "late fusion". Late fusion has been shown to be very effective for many applications. Early fusion by frame-wise feature concatenation is also possible with the test procedure provided because there is no limit on what features are present in the feature data for each of the modalities being tested. Intermediate model level fusion is not possible using the test procedure as it stands because this would require the model training and testing procedures to be passed features from multiple modalities. The test procedure could be extended to allow this without too much difficulty, but this was not within the scope of the SecurePhone project.

[13] It is possible, but not certain, that face or signature verification will not benefit from any kind of UBM.

[14] A UBM has two main purposes. One is to provide initial model from which to train a client model by adaptation when the amount of client training data is limited. The other is to act as an impostor model for every client (hence it is essential that no client data is used in UBM training). UBMs are often beneficial when used with voice verification. With face and signature they may or may not be beneficial. However, with limited training data it is never a forgone conclusion that it will or will not improve verification performance.

[15] A text dependent GMM client profile typically requires 100 to 200 Gaussians, while a client-selected password profile typically requires 1000 to 2000 Gaussians.