FROM GMM TO HMM FOR EMBEDDED PASSWORD-BASED SPEAKER RECOGNITION

Anthony Larcher, Jean-François Bonastre, John S. D. Mason

FROM GMM TO HMM FOR EMBEDDED PASSWORD-BASED SPEAKER RECOGNITION

AnthonyLarcher[1,2], *Jean-François Bonastre*[1], *JohnS.D.Mason*[2]

1. Laboratoire d'Informatique d'Avignon (LIA), UAPV, France
2. Speech and Image group, Swansea University, Wales, UK

{anthony.larcher, jean-francois.bonastre} @univ-avignon.fr
J.S.D.Mason@swansea.ac.uk

ABSTRACT

Embedded speaker recognition in mobile devices involves a limited amount of computing resource but also is linked with several ergonomic constraints. For example both the enrolment and the test have to be done using short audio sequences. Even if they proved their efficiency in more classical situations, GMM/UBM based systems show their limits in this context.

This paper deals with this problem and proposes to take into account the linguistic nature of the speech material inside the GMM/UBM framework. The proposed solution mixes the text-independent aspects of the GMM/UBM with a semi-continuous like approach in order to deal with the text-dependent information. This system respects both the resource and the ergonomic constraints of the considered application field.

The preliminary experiments are done on the publicly available database ValidDB and show the potential of the proposed approach. Particularly, when compared to the GMM/UBM, our approach decreases drastically both the computational cost and the equal error rates when impostors don't know the user passwords. For other situations the performance remains comparable between both approaches.

1. INTRODUCTION

An embedded speaker recognition system, in cellphones for example, has to respect several constraints. Computational and memory resources are limited and the ergonomic aspects of a realistic application impose to train the models with few data and to perform on short test audio sequences. Classical speaker recognition engines work on text-independent inputs and follow usually the GMM/UBM (Gaussian Mixture Model/ Universal Background Model) paradigm [1]. This solution allows a high level of performance as shown during NIST evaluations [2]. Unfortunately, the UBM/GMM performance depends strongly on the quantity of training data available to enrol a speaker when the embedded context involves short duration speech material. A solution to this problem is to increase the amount of information taken into account by the system by including text dependencies, like in user-customized password scenario [3]. In this case, the Temporal Structure Information (TSI) gathered from the password helps to compensate the short duration of the audio sequences. In order to model the TSI of speech while achieving statistical modelling, a word recognition system could be combined with a speaker recognition system [4].

To satisfy the targeted application constraint, the system should accept all kinds of passwords and also be language-independent. Adding language options when using a phoneme-based word recognition system would seem viable with a good set of phonemes covering languages. However this solution would be expensive in terms of storage.

A cellphone-embedded system is confronted with strongly variable environments. Due to this constraint the acoustic modelling used in the recognition system has to be adapted to the environment and the computational cost of the adaptation has to follow the targeted context resource constraints. HMM modelling doesn't seem well suited as it requires a large amount of training data and a significant resource consumption.

The solution proposed in this paper tries to associate the well known advantages of a GMM based statistical acoustic modelling with an original architecture able to deal with the application context constraints. It uses the GMM/UBM paradigm for the general acoustic space modelling and the text-independent speaker recognition abilities. It also involves an HMM/Viterbi approach in order to incorporate the text-dependent and TSI aspects. Such a combined system was originally proposed in [5] for speaker recognition and extended to word recognition in [6].

The specific three stage architecture is described in Section 2. The method using this structure to reduce memory and computational costs and the enrolment algorithms are also described in this section. The experimental protocol and preliminary results are described in Section 3 as well as the Valid database [7]. Section 4 summarises the benefit of this approach and presents different future work directions.

2. DESCRIPTION OF THE APPROACH

The proposed system combines a statistical representation of the acoustic space and a precise modelling of the TSI. Based on a semi-continuous hidden Markov model (SCHMM) [8], it operates a three stage acoustic modelling architecture. In order to involve the TSI with the respect of training data and resources limits, a GMM which represents the acoustic space is derived to obtain the SCHMM state probability functions.

2.1 EBD Hierarchical Architecture

Figure 1 illustrates the proposed three stage hierarchical architecture, denoted EBD, for Embedded LIA_SpkDET [9] system in this paper. All the nodes in this architecture are a GMM. The upper layer is the least specialised one and is a classical UBM. It aims to model the general acoustic space. The middle layer contains the text-independent specific characteristics of each speaker. These text-independent speaker models are obtained by a classical GMM/UBM adaptation method: each speaker model is derived from the UBM following the *Maximum A Posteriori* (MAP) criterion [10] and using the EM algorithm. Only the mean parameters are adapted and the other parameters are taken from the UBM. The bottom layer uses the ability of a left-right SCHMM in order to capture the text-dependent information. Particularly the SCHMM takes into account the TSI of the user-customised passwords. Each of the SCHMM states is a GMM derived from the corresponding middle level model. The transformation function works only on the weights of the GMMs, the other parameters are directly taken from

the middle level model.

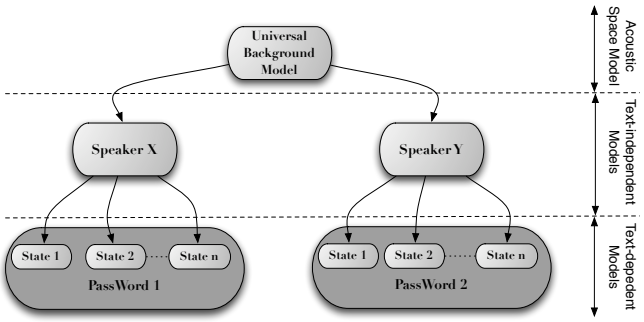

Figure 1: General view of the EBD model architecture.

2.2 Training Step

The training algorithm is decomposed into three steps, corresponding to each level of the EBD architecture:

- The upper level UBM is firstly trained to model the largest part of the acoustic space. It is built off-line using a large amount of data. The UBM is trained with a classical EM/ML algorithm as described in [1], [11].
- The text-independent speaker model training consists in adapting the UBM/GMM with the whole available data pronounced by this client. An energy labelling is performed on the signal and only the speech frames are kept. Using the selected frames, the final model is obtained by adapting the means of the UBM thanks to an EM algorithm with the MAP criterion.
- In order to catch the TSI of the password, a SCHMM is learnt. Contrary to the speaker dependent GMMs of the second stage, all frames of the password - and not only speech frames - are used to adapt the S states of the SCHMMs. The password sequence is cut in S segments $\{seg_i\}$ of the same length. Each state i of the SCHMM is adapted from the speaker text-independent model using seg_i to modellise a subpart of the password. An EM/MAP algorithm is applied on only the weight parameters. Then the SCHMM is optimised thanks to a classical Viterbi algorithm (a new segmentation is achieved by Viterbi and is used to adapt the state models). The number of states, S, of the SCHMM is experimentally determined. Finally, the transitions of the SCHMM are computed using the relative length of each segment.

2.3 Testing Step

During a test phase, the score between the input signal and a password SCHMM model is issued from the corresponding Viterbi path. All the input frames are used during this Viterbi decoding phase. The log-likelihood of a frame sequence could be expressed as a sum of two log-likelihood accumulations, one using speech-labelled frames and the other using non-speech-labelled frames, as shown in expression 1.

$$log\ p(X|\lambda) = log\ p(X_{speech}|\lambda) + log\ p(X_{non-Speech}|\lambda) \quad (1)$$

The final speaker matching score corresponds to the log-likelihood computed with only the $log\ p(X_{speech}|\lambda)$. This score is normalised by the log-likelihood computed using the UBM model on the same speech-labelled frames.

2.4 Respect of the Resource Constraints

In order to evaluate our approach in terms of memory constraints and computational cost, the EBD system is compared to baseline systems. The memory occupation of the EBD system is evaluated by considering the number of stored parameters. Indeed this paper does not consider data compression and coding which are both mandatory for a such embedded application. In this context the memory estimation in terms of megabytes does not make sense. The computational cost is estimated from the number of operations associated with the function $f(y, G)$, where $f(y, G)$ is the log-likelihood computation function for an acoustic feature y and a monovariate Gaussian G.
The EBD system lies at the limit between speaker and speech recognition systems and could be compared with two systems working in parallel. We use this decomposition and we compare our system to both a classical speech recognition engine and a classical speaker recognition system.

2.4.1 Speaker Recognition

As described in 2.1 the EBD system has a hierarchical architecture. The first and the second layers of this architecture present the same structure as a classical text-independent speaker recognition system. This part of the EBD system has then the same characteristics in terms of computational cost and memory resources as this standard GMM/UBM system. As in classical approaches the memory and computational resources are already minimized by tying parameters between the UBM and the speaker models and by computing only the n-top Gaussian. Considering a system with 128 Gaussian components per model, 32 acoustic coefficients and 5 speaker models, the number of stored parameters is 28 800 and for each input frame, 24 576 log-likelihood computations are done. We adapt all the mean parameters in this work, an additional gain is possible in both memory and computing by adapting only a subpart of the parameters or with a Gaussian selection [12].

2.4.2 Speech Recognition

The main benefit comes from the speech recognition part of the EBD system. As the two first layers are equivalent to a UBM/GMM speaker recognition system, the third layer should be compared to a speech recognition system. The main constraint of this application comes from the possibility of the users to choose their own password without any limitation. The system used should also be vocabulary-unconstrained. State-of-the-art speech recognition systems are mainly related to statistical methods like Hidden Markov Model and phonemic models [6]. In this sense, two approaches are compared with the EBD system.
The first one is a phonemic based approach using non-contextual models which requires significant memory resources. For an acoustic model with non-contextual phonemes there are 128 distributions per state, each with 32 acoustic coefficients. The acoustic model contains 108 emitting states. The number of trained passwords has a negligible effect and the memory occupation in terms of parameter numbers could be approximated by only the acoustic model size:

$$nbes \times nbg \times \underbrace{(2 \times vectsize + 1)}_{oneGaussian} \quad (2)$$

where *nbes* is the number of emitting states, *nbg* is the number of Gaussian components and *vectsize* is the number of acoustic features. The computational cost of this part is a supplement to the speaker recognition task. The likelihood of each distribution of the acoustic space model is computed for each acoustic feature of the test data. For the system previously described, the computational supplement consist in computing 13, 824 log-likelihood per acoustic feature.
The second approach is a global HMM for each password. For one password trained by one speaker, we estimate the acoustic model size (in terms of number of parameters) as

follows:

$$stnb \times nbg \times \underbrace{(2 \times vectsize + 1)}_{oneGaussian} \quad (3)$$

where *stnb* is the number of states of the HMM. The computational cost (in terms of log-likelihood number) is estimated as follows:

$$stnb \times nbg \times vectsize \quad (4)$$

When linking speaker recognition and password recognition with SCHMM in the EBD system, only the weight parameters are needed for a given state, the other parameters are tied to the upper level GMM. For one password trained by one speaker, this system only requires the storage of very few parameters:

$$stnb \times nbg \times 1 \quad (5)$$

The computation of the Gaussian component log-likelihoods is done in the speaker recognition part of the architecture. By adapting only the weights of the Gaussian distributions, the speech recognition task only requires weighted combinations which makes the computational cost of this system negligible compared to the baseline systems. The memory occupation for the EBD system is proportional to the number of passwords but would be very small compared to the baseline systems for the targeted application.
Table 1 shows the required resources of the 4 systems for 5 speakers, 2 passwords per speaker, 15 states per password (for the HMM and EBD system), 128 Gaussian distributions and 32 acoustic parameters. (It is also possible to tie some parameters between models for each system; this optimization is not taken into account here.)

	Number of parameters	Number of log-likelihood computations
Speaker Recognition + Phonemic Model	927, 360	442, 368
Speaker Recognition + Password HMMs	1, 276, 800	614, 400
EBD	**33, 600**	**24, 576**
Speaker Recognition	28, 800	24, 576[a]

[a]The cost of the weighted means is neglected.

Table 1: Evaluation of the memory (in terms of parameters) and computational resources (in terms of log-likelihood computation) of EBD and two baseline systems per input frame

3. EXPERIMENTS ON VALID-DB

3.1 Valid Database

The experiments are performed on the Valid Database [7]. This database contains audio-video records from 106 speakers: 77 men and 29 women. Based on the XM2VTS linguistic content, two occurrences are recorded in 5 sessions for each of these speakers. The first one - denoted in this paper *DIGIT* - is a sequence of digits : *"5 0 6 9 2 8 1 3 7 4"* and the second - denoted in this paper *SENTENCE* - is the same phonetically well-balanced sentence as in M2VTS *"Joe took father's green shoe bench out"*.
The five sessions are recorded over a period of one month. The first session is a clean one (this session is recorded under controlled acoustic and illumination conditions). The other four sessions are recorded in a real world scenario without any control on illumination or acoustic noise.
The Valid database was chosen as we aim to include the video stream in the system, despite its small size. However Valid presents several drawbacks for our work. The sentences are too long for a real password dedicated system and only two different sequences are recorded. Moreover, these sequences are too different (digits vs. sentence) to represent realistic passwords. Finally, the number of speakers is small for a speaker verification experiment. In order to withdraw the gender mismatch problem, only male sessions have been used in this work.

3.2 Protocol

The 77 male speakers are separated into three sets: the *UBM-set* with 25 speakers, the *CLIENT-set* with 25 other speakers, and the *IMPOSTOR-set* with the 27 remaining males.
As indicated by the set names, the UBM-set is used in order to train the UBM model, the Client-set is used to enrol the client speakers and for the true access and the last set, the impostor-set is only used for the impostor trials. Three background models are computed using the UBM-set of speakers. The *UBM-DIGIT* is learned on the 5 *DIGIT* sessions of the 25 ubm-speakers, *UBM-SENTENCE* is learned on the 5 *SENTENCE* sessions of the UBM-set of speakers. Finally, a third UBM model, *UBM-ALL* is learned with the 5 sessions of both occurrences.
The UBM models are learned using the noisy and the clean sessions but the clean sessions are no longer used in the rest of the protocol as only one clean session is available per speaker.

Starting with the three UBMs, eight conditions are defined:

- *1occ-DIGIT*: in this condition, each client model is derived from *UBM-Digit* using one noisy *DIGIT* occurrence of this speaker. Due to the small number of speakers available in the database, a jacknifing process is used by learning a client model on each available noisy *DIGIT* occurrence. Of course, the used training occurrence will not be used for the true speaker access. This process gives 100 speaker models (25 speakers, 4 noisy *DIGIT* occurrences).
- *1occ-SENTENCE*: respectively, 100 speaker models are derived from the *UBM-SENTENCE* using the noisy *SENTENCE* occurrences. The same jacknifing process is used.
- *2occ-DIGIT*: each client model is now trained using two *DIGIT* noisy occurrences of one given speaker and the *UBM-DIGIT*. The jacknifing process selects all the two from four combinations in order to give six models of the original speaker and a total of 150 speaker models (25 speakers, 6 couples of noisy *DIGIT* occurrences).
- *2occ-SENTENCE*: respectively, 150 speaker models are defined using *UBM-SENTENCE* and the noisy *SENTENCE* occurrences.
- *1occ-all-DIGIT*: this condition is the same as *1occ-DIGIT* but using the *UBM-ALL* instead of *UBM-DIGIT*.
- *1occ-all-SENTENCE*: this condition is the same as *1occ-SENTENCE* but using the *UBM-ALL* instead of *UBM-SENTENCE*.
- *2occ-all-DIGIT*: this condition is the same as *2occ-DIGIT* but using the *UBM-ALL* instead of *UBM-DIGIT*.
- *2occ-all-SENTENCE*: this condition is the same as *2occ-SENTENCE* but using the *UBM-ALL* instead of *UBM-SENTENCE*.

The number of target accesses per condition is constant for all the above eight conditions. 300 target accesses are defined. For *1occ-X* conditions, 100 target models are compared to the three available test sequences (four noisy occurrences are available for each password, one is used in order to train the speaker model, the three remaining ones for the client accesses). For *2occ-X* conditions, 150 target models are compared to the two available test sequences.

The impostor tests are computed using the *IMPOSTOR-set* of speakers. Two configurations are proposed:

- PASSWORD: in this configuration, the linguistic content of the impostor test sequences is the same as the occurrences used to train the client models. Each speaker model of the *Xocc-DIGIT* and *Xocc-all-DIGIT* sets is compared to the four noisy *DIGIT* occurrences of each of the 27 speakers of the IMPOSTOR-set. Respectively, each speaker model of the *Xocc-SENTENCE* and *Xocc-all-SENTENCE* sets is compared to the four noisy *SENTENCE* occurrences of each of the 27 speakers of the IMPOSTOR-set.
- WRONG: in this condition, the linguistic content of the impostor test occurrences is different to the one of client models training material. Each speaker model of the *Xocc-DIGIT* and *Xocc-all-DIGIT* sets is compared to the four noisy *SENTENCE* occurrences of each impostor speaker. Respectively, each client model of the *Xocc-SENTENCE* and *Xocc-all-SENTENCE* sets is compared to the four noisy *DIGIT* occurrences of each of the 27 impostor speakers.

For both configurations, the number of impostor tests is constant for a given client model. However, the global number of impostor tests performed depends on the selected client condition:

- *1occ-X* and *1occ-all-X* conditions are composed on 100 client models. These conditions give 10, 800 impostors accesses (27 impostor speakers, four noisy occurrences of the chosen linguistic content for each of the 100 client models).
- *2occ-X* and *2occ-all-X* conditions give 150 client models. It gives 16, 200 impostor accesses (27 impostor speakers, four noisy occurrences of the selected linguistic content and 150 client models).

The EBD and the baseline systems are developed using the ALIZE toolkit [9].

3.3 Parametrization

Mel-scaled frequency cepstral coefficients (MFCC) are used, computed every 10ms. An energy labelling is applied to separate the *speech* frames from the *non-speech* frames. Acoustic feature frames are 32-dimension vectors, 15 cepstral coefficients, the log-energy and the corresponding Δ coefficients.

3.4 Results

Firstly, we wish to evaluate the influence of the number of components of the GMM models on the EBD system overall performance. Figure 2 shows the performance of the

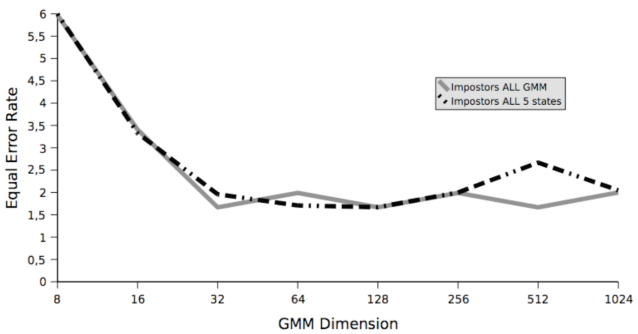

Figure 2: EER for GMM and EBD 5-state systems against GMM size (1-occ-Sentence, both WRONG/PASSWORD impostor tests)

EBD system with five states against the number of components in the GMM, from 8 to 1, 024. The performance is evaluated in terms of EER using *UBM-ALL*, SENTENCE and both PASSWORD and WRONG conditions (1occ-all-SENTENCE plus PASSWORD and WRONG impostor conditions). The performance of the baseline UBM/GMM system is given for comparison. The experiment shows that a sufficient number of components is 64 to achieve a good level of performance for both systems. Due to the small amount of available training data, the performance reaches a maximum when the number of components is about 256, which is similar to the observations made by [13]. In the configuration retained for the rest of the experiments, the other fixed parameters are:

- dimension of GMMs is fixed to 256 and all mean parameters of the text-independent speaker models are adapted.
- 32 weight parameters are adapted for each SCHMM state

The main expected advantage of the EBD system compared to a classical UBM/GMM is to incorporate the password-based information like the password itself and the relative TSI. This point is evaluated by the experiments presented in the Tables 2 and 3. These tables present the performance

Client Set	GMM baseline	Number of states of EBD		
		5	10	15
1occ-Digit	2.33	2.67	2.35	2.33
1occ-Sentence	1.67	1.33	1.27	1.00
1occ-all-Digit	1.30	1.67	1.32	1.00
1occ-all-Sentence	0.38	0.08	0.02	0.03

Table 2: EER of GMM and EBD systems (with different number of states) - WRONG test configuration

Client Set	GMM baseline	Number of states of EBD		
		5	10	15
1occ-Digit	1.96	2.02	2.64	2.67
1occ-Sentence	2.00	1.73	2.93	3.27
1occ-all-Digit	1.74	3.02	3.59	3.35
1occ-all-Sentence	2.00	2.33	3.67	4.41

Table 3: EER of GMM and EBD systems (with different number of states) - PASSWORD test configuration

Client Set	GMM baseline	Number of states of EBD		
		5	10	15
1occ-Digit	1.96	2.02	2.64	2.67
2occ-Digit	1.00	1.00	1.00	1.27
1occ-Sentence	2.00	1.73	2.93	3.27
2occ-Sentence	0.73	**0.42**	1.00	1.28
1occ-all-Digit	**1.74**	3.02	3.59	3.35
2occ-all-Digit	**1.05**	1.67	1.95	1.68
1occ-all-Sentence	**2.00**	2.33	3.67	4.41
2occ-all-Sentence	0.92	**0.73**	1.34	1.99

Table 4: Comparison of EER with one and two training occurrences - WRONG test condition

of the EBD system depending on the number of states and

the nature of the impostor tests. The performance of the baseline GMM in the same conditions is provided for comparison. It is important to note that the GMM system is equivalent to an EBD system with only one state. Table 2 (WRONG condition) shows a clear advantage for the EBD system when the impostors do not know (pronounce) the speaker passwords. Increasing the number of states from one to 15 allows a continuous gain in terms of EER. This result is not confirmed by Table 3 where the impostors know the target speaker passwords. A loss of performance is observed when the number of states increases. It seems that, in the speaker matching score, the TSI shadows the speaker specific information, i.e. the password is recognised and not the speaker. A solution to this problem is to increase the amount of speaker specific information in the EBD model, for example by adding several training occurrences during the speaker enrolment. In order to investigate this possibility we propose a similar experiment to the previous one but with two occurrences of the speaker password during the training phase. Table 4 presents the results using the PASSWORD condition, depending on the number of training occurrences (one or two) and on the number of EBD states. As expected, the performance improves when the amount of training data increases, for all conditions. More interesting is the comparative behaviour of the GMM system and EBD system. The EBD system gains more from the increase in training data than the GMM, even if the GMM performs generally better than the EBD. This result indicates clearly that the balanced problem between text-content information and speaker specific information could be solved by adding more training occurrences during the speaker enrolment. For the WRONG conditions, the same behaviour is noticed.

4. CONCLUSIONS AND FUTURE WORKS

We present a new acoustic architecture for a password-dependent speaker recognition dedicated to embedded applications. The proposed approach associates the advantages of a text-independent GMM/UBM system with an HMM/Viterbi-based text-dependent system. It follows both the ergonomic constraints like the small amount of training data and the computing resource constraints. The preliminary experiments demonstrate the password recognition abilities of EBD, even if a very small amount of training data is provided. Even if a large decrease in EER was observed for several experimental conditions compared to a GMM, the results have to be validated on a larger database as Valid is limited for a speaker recognition task.
Future work will focus on the TSI by exploring the ratio between passwords and the number of states of the EBD model. We aim also to incorporate TSI from a second modality like the video stream in order both to increase the performance and in an attempt to thwart replay attacks.

REFERENCES

[1] Frederic Bimbot, Jean-François Bonastre, Corinne Fredouille, Guillaume Gravier, Ivan Magrin-Chagnolleau, Sylvain Meigner, Teva Merlin, Javier Ortega-Garcia, Dijana Petrovska-Delacretaz, and Douglas A. Reynolds, "A tutorial on text-independent speaker verification," *EURASIP Journal on Applied Signal Processing*, vol. 4, pp. 430–451, April 2004.

[2] Mark A. Przybocki, Alvin F. Martin, and Audrey N. Le, "NIST speaker recognition evaluations utilizing the mixer corpora - 2004, 2005, 2006," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 15, no. 7, pp. 1951–1959, 2007.

[3] Mohamed Faouzi BenZeghiba and Hervé Bourlard, "User-customized password speaker verification using multiple reference and background models," *Speech Communication*, vol. 48, no. 9, pp. 1200–1213, 2006.

[4] Jiri Navratil, Upendra V. Chaudhari, and Stephane H. Maes, "A speech biometrics system with multigrained speaker modeling," in *Conference for Natural Speech Processing*, 2000.

[5] Jean-François Bonastre, Philippe Morin, and Jean-Claude Junqua, "Gaussian dynamic warping (gdw) method applied to text-dependent speaker detection and verification," in *European Conference on Speech Communication and Technology (Eurospeech)*, Geneva (Switzerland), 2003.

[6] Christophe Lévy, Georges Linares, Pascal Nocera, and Jean-François Bonastre, *Mobile Phone Embedded Digit-Recognition*, chapter 7 in Digital Signal Processing for In-Vehicle and Mobile Systems 2, Springer Sciences, 2006.

[7] Niall A. Fox, Brian A. O'Mullane, and Richard B. Reilly, "The realistic multi-modal valid database and visual speaker identification comparison experiments," in *International Conference of Audio and Video-Based Person Authentication, AVBPA*, New York (US), July 2005.

[8] Steve J. Young, "The general use of tying in phoneme-based hmm speech recognisers," in *IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP*, San Francisco (USA), 1992, vol. 1, pp. 569–572.

[9] Jean-François Bonastre, Nicolas Scheffer, Driss Matrouf, Corinne Fredouille, Anthony Larcher, Alexandre Preti Gilles Pouchoulin, Nicholas Evans, Benoît Fauve, and John S. Mason, "Alize/spkdet: a state-of-the-art open source software for speaker recognition," in *Odyssey Conference*, 2008, http://mistral.univ-avignon.fr/.

[10] Jean-Luc Gauvain and Chin-Hui Lee, "Maximum a posteriori estimation for multivariate gaussian mixture observations of markov chains," in *IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP*, Adelaide (Australia), April 1994, vol. 2, pp. 291–298.

[11] Arthur Pentland Dempster, Nam M. Laird, and Donald B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society*, vol. 39, no. 1, pp. 1–38, 1977.

[12] Roland Auckenthaler and John S. Mason, "Gaussian selection applied to text-independent speaker verification," in *Odyssey Conference*, Crete, 2001, pp. 83–86.

[13] John S. D. Mason, Nicholas W. D. Evans, Robert Stapert, and Roland Auckenthaler, "Data-model relationship in text-independent speaker recognition," *EURASIP Journal on Applied Signal Processing*, vol. 4, pp. 471–481, 2005.