Reconnaissance automatique du locuteur embarquée dans un téléphone portable

Anthony Larcher, Christophe Lévy, Driss Matrouf, Jean-François Bonastre

Université d'Avignon et des Pays de Vaucluse,
Laboratoire Informatique d'Avignon (UPRES 931),
F-84018 Avignon, France
{anthony.larcher, christophe.levy, driss.matrouf, jean-francois.bonastre}@univ-avignon.fr

ABSTRACT

Embedded speaker recognition in mobile devices involves a limited amount of computing resources. However, the performance of state-of-the-art systems are usually evaluated without any limitation of material resources. In this paper, we evaluate several downscaled configurations of the LIA UBM/GMM speaker verification system. The impact of scalability is evaluated in terms of memory resources and computational time. We propose two downscale configurations which allow a good compromise between resource consumption and performance degradation. Experiments performed on the Banca database show that the memory allocation and the computational time decrease of about 70% and 88% when the error rate raises from $3,48\%$ to $4,77\%$ or still comparable to the baseline.

Keywords: speaker recognition, scalability, embedded systems

1. Introduction

Au regard des dernières campagnes d'évaluation internationales organisées par le NIST [7], la Reconnaissance Automatique du Locuteur (RAL) a réalisé de réels progrès ces dernières années. Ces derniers sont dus à l'introduction de nouvelles méthodes telles que le Factor Analysis ([5], [6]) ou la Nuisance Attribut Projection ([8]) qui ont notamment permis de compenser les dégradations induites par l'usage de différents microphones et/ou de différentes liaisons téléphoniques, ou encore de limiter l'influence de la variabilité due à l'environnement acoustique dans lequel est utilisé le système. Ces avancées, ainsi que la maturité atteinte par les systèmes actuels, rendent envisageable le développement d'applications grand-public sécurisées par la biométrie vocale.
La multiplication des systèmes embarqués, observée ces dernières années (téléphones portable, GPS, PDA, ...), a ouvert de nouveaux marchés. Cependant, le développement du matériel et des réseaux s'est accompagné d'une croissance équivalente du nombre d'applications disponibles sur ces terminaux. Aussi, les contraintes matérielles induites par ces clients légers constituent une problématique majeure pour l'intégration des technologies de reconnaissance du locuteur.
Cet article propose une étude des performances d'un système état-de-l'art dans un contexte de ressources matérielles limitées. Nous considérons l'influence des principaux paramètres scalables qui peuvent être modifiés afin de satisfaire aux ressources restreintes d'un système embarqué. Différentes configurations du système UBM/GMM développé à partir de la plateforme Mistral (évalué lors des principales évaluations internationales de vérification du locuteur [2]) sont comparées en fonction de leurs performances, mais également en fonction des ressources qu'elles nécessitent tant en termes de temps CPU que d'occupation mémoire.
Dans un premier temps, nous donnons un exemple de la capacité des téléphones portables actuels. Ensuite, le système de reconnaissance développé à partir de la plateforme biométrique MISTRAL[1] [3] est présenté dans la section 3. La partie 4 contient les détails concernant le protocole expérimental utilisé. Les paramètres scalables du système ainsi que les performances obtenues sont commentés dans la partie 5. De cette analyse, deux configurations particulières du système de vérification du locuteur ont été sélectionnées : une solution nécessitant le moins de ressources possibles mais qui a des performances non optimales et une configuration représentant le meilleur compromis. Enfin, la dernière partie présente quelques conclusions issues de ce travail.

2. Exemple de ressources

Cette étude se situe dans le contexte du projet européen MOBIO[2] au cours duquel un système de vérification du locuteur doit être implémenté sur un téléphone portable. Le portable choisi pour cette intégration est le Nokia© N900.

vendor id	ARM
model name	Cortex-A8
CPU MHz	600
RAM	256 MB
CPU cores	2

Tab. 1: Caractéristiques du téléphone Nokia© N900 sur lequel doit être intégré le système de vérification du locuteur dans le projet européen MOBIO.

Les ressources présentées dans le tableau 1 correspondent à celles disponibles pour l'ensemble des applications actives sur le téléphone à un instant donné et non pas aux ressources disponibles pour la seule application de vérification du locuteur.

3. Description du système

Les différentes configurations du système UBM/GMM présentées par la suite sont basées

[1] http://mistral.univ-avignon.fr/
[2] http://www.mobioproject.org/

sur le système développé au sein du Laboratoire Informatique d'Avignon (LIA) à partir de la plateforme biométrique MISTRAL [3]. Les performances de ce système, évaluées lors des dernières campagnes internationales de vérification du locuteur NIST-SRE [2], le placent au niveau des systèmes état-de-l'art.

Ce système est basé sur le paradigme UBM/GMM associé au Latent Factor Analysis [6] ; ce qui permet de prendre en compte la variabilité session intra-locuteur. Les paramètres acoustiques utilisés sont issus d'une analyse cepstrale en banc de filtres (obtenus avec SPRO [4]). Les vecteurs de paramètres utilisés sont composés des 19 coefficients statiques (c), des 19 coefficients dérivés du premier ordre (Δc), de 11 coefficients dérivés du second ordre ainsi que du coefficient dérivé (du premier ordre) de l'énergie (ΔE). Une sélection des trames de plus haute énergie est appliquée de façon standard (pour séparer les trames *parole* des trames *non-parole*) avant d'appliquer une normalisation (soustraction de la moyenne et normalisation de la variance) au niveau de chaque fichier.

Le modËèle du monde (UBM) comprend 512 distributions Gaussiennes et la matrice de covariance est supposée diagonale. Chaque modèle de locuteur est dérivé de l'UBM par une adaptation basée sur le critère du Maximum A Posteriori (MAP). Il faut enfin noter qu'en raison du contexte (RAL embarquée), le choix a été fait de ne pas normaliser les scores. En effet, les normalisations classiques, type ZTNorm, sont consommatrices de temps CPU et d'espace mémoire.

4. Protocole expérimental

Les différentes configurations présentées dans ce papier ont été évaluées sur la base de données Banca [1]. Cette base comprend les enregistrements vidéo de 52 personnes réparties en deux sous-groupes G1 et G2 contenant chacun 13 hommes et 13 femmes.

Chaque locuteur présent dans la base BANCA a participé à 12 sessions d'enregistrement dans différentes conditions avec différentes caméras. Les quatre premières sessions sont enregistrées avec une luminosité et un environnement sonore contrôlés. Les sessions 5 à 8 et 9 à 12 correspondent respectivement à des conditions « dégradées » et « adverses ». Ces différentes conditions d'enregistrement permettent d'obtenir une plus grande variabilité au niveau de l'arrière plan sonore.

Cette étude, préalable à l'implémentation du système de reconnaissance du locuteur au sein du téléphone portable, a été réalisée sur un ordinateur et non sur le téléphone[3]. Son but n'est pas seulement d'évaluer les taux d'erreurs du système de RAL, mais bien de les rapprocher de l'occupation mémoire et du temps de calcul consommés. Les caractéristiques du PC utilisé sont présentées dans le tableau 2.

Le temps de calcul évalué avec la commande système `time` a permis de déduire le temps CPU nécessaire à l'exécution du processus (le chargement des différents modèles et des données est inclus dans le temps mesuré). La quantité de mémoire nécessaire a été estimée en utilisant le profileur Valgrind qui permet de mesurer le pic de mémoire allouée. Dans le cadre d'une implémentation sur système embarqué, la notion de maximum de mémoire allouée à un instant t est plus importante que la taille totale allouée. Enfin, l'estimation des ressources nécessaires a été réalisée à partir d'un sous-ensemble représentatif des tests de la base de données BANCA (les taux d'erreurs, eux, sont donnés pour l'ensemble du corpus).

[3]Le Nokia© N900 n'est pas encore sur le marché à l'heure actuelle, mais de premiers essais ont pu être réalisés par l'intégrateur partenaire du projet.

vendor id	GenuineIntel
model name	Intel(R) Core(TM)2 Quad CPU Q9300
CPU MHz	2000
cache size	3072 KB
CPU cores	4 **(1 seul est utilisé)**
bogomips	4987.44

Tab. 2: Caractéristiques du PC sur lequel ont été effectués les tests présentés dans cet article.

5. Paramètres scalables

Cette partie présente l'influence de trois paramètres scalables sur les performances du système :
- le nombre de distributions Gaussiennes des GMM ;
- la taille des vecteurs acoustiques ;
- la proportion d'échantillons traités en phase de test.

Ces trois paramètres, choisis car ils impactent fortement les besoins du système en termes de mémoire et de temps de calcul, influent sur deux facteurs directement liés aux ressources utilisées :
- le nombre de paramètres à stocker pour chaque modèle, exprimé par :

$$nb_{param} = nb_{gauss} \times (2 \times tailleVectAc + 1) \quad (1)$$

où nb_{param} correspond au nombre de paramètres à stocker pour chaque modèle GMM (UBM + locuteurs), nb_{gauss} est le nombre de composantes Gaussiennes du modèle et $tailleVectAc$ est la taille des vecteurs de paramètres utilisés ;
- le nombre de fonctions de log-vraisemblance (qui constituent l'essentiel du coût de calcul), donné par :

$$nb_{log-vrais} = 2 \times nb_{gauss} \times tailleVectAc \times nb_{trames} \quad (2)$$

où $nb_{log-vrais}$ est le nombre de fonctions de log-vraisemblance à calculer et nb_{trames} correspond au nombre de trames traitées de la séquence de test.

5.1. Nombre de composantes du modèle du monde

Au regard des équations 1 et 2, le nombre de composantes Gaussiennes du modèle du monde détermine de façon explicite la mémoire nécessaire au stockage des modèles de locuteurs. Le tableau 3 présente les résultats obtenus pour des modèles comprenant de 512 à 32 distributions Gaussiennes.

La réduction du nombre de distributions des modèles entraîne une augmentation du taux d'égales erreurs (EER) mais permet de réduire significativement le temps de calcul et la mémoire nécessaire. L'utilisation de modèles à 128 Gaussiennes permet par exemple, de conserver des performances comparables à celles

du système de référence, tout en divisant par 3 la mémoire allouée par le système et par 4 le temps de calcul. Le système avec 32 Gaussiennes permet, lui, de réduire le temps de calcul par un facteur 14 et les allocations mémoire par 5 tout en conservant un taux d'erreurs compris entre 4% et 5%.

# distributions	**512**	**256**	**128**	**64**	**32**
G1 (EER %)	3,48	2,19	3,86	4,23	5,15
G2 (EER %)	2,94	3,32	3,32	2,19	3,85
mem. (MB)	7,84	4,29	2,70	1,90	1,50
mem. rel. (%)	100	57	36	25	20
temps CPU (s)	2,06	1,07	0,53	0,27	0,15
temps rel. (%)	100	52	25	13	7

Tab. 3: Évolution des performances (EER) et des ressources (temps CPU et mémoire) utilisées en fonction du nombre de composantes du modèle. La consommation des ressources est donnée de manière relative par rapport au système de base.

5.2. Taille du vecteur acoustique

La dimension des vecteurs acoustiques utilisés est en lien direct avec l'espace mémoire et le temps de calcul nécessaires puisqu'elle apparaît dans les équations 1 et 2. Différentes dimensions de vecteurs acoustiques ont été testées. Ces configurations diffèrent également par la nature des coefficients utilisés (c, Δc, $\Delta\Delta c$ ou ΔE). Considérant le nombre très important des combinaisons possibles, nous avons choisi de ne retenir que quatre configurations (en plus de celle de référence), détaillées dans le tableau 4. Ce tableau présente les performances du système de vérification du locuteur utilisant ces différentes paramétrisations comparées au système de référence.

# paramètres	**50**	**41**	**30**	**25**	**20**
# c	**19**	**15**	**10**	**15**	**10**
# Δc	**19**	**15**	**10**	**10**	**10**
# $\Delta\Delta c$	**11**	**11**	**10**		
# ΔE	**1**				
G1 (EER %)	3,48	4,77	3,48	5,52	5,15
G2 (EER %)	2,94	3,85	4,23	3,85	4,23
mem. (MB)	7,84	6,60	5,48	4,99	4,46
mem. rel. (%)	100	88	73	67	60
temps CPU (s)	2,06	1,92	1,80	1,79	1,71
temps rel. (%)	100	95	87	86	83

Tab. 4: Évolution des performances (EER) et des ressources (temps CPU et mémoire) utilisées en fonction de la taille du vecteur acoustique (512 distributions par GMM). La consommation des ressources est donnée de manière relative par rapport au système de base.

La réduction de la dimension des vecteurs acoustiques réduit de manière significative l'espace mémoire utilisé ainsi que le temps de calcul nécessaire à la vérification du locuteur. À nombre de paramètres équivalent, il semble que la suppression des coefficients dynamiques $\Delta\Delta c$ entraîne une dégradation importante du taux d'égales erreurs ; en effet le taux d'erreurs moyen (sur G1 et G2) passe de 3,85% pour le système 30 coefficients ($10c + 10\Delta c + 10\Delta\Delta c$) à 4,65% pour le système 25 coefficients ($15c+10\Delta c$), ce qui représente une augmentation relative du taux d'erreurs de plus de 20%.

5.3. Sélection de trames

Les paramètres acoustiques fournis au système de reconnaissance du locuteur sont extraits de façon classique à la fréquence d'une trame toutes les 10ms. Dans cette sous-partie, la réduction est opérée sur la proportion de vecteurs acoustiques qui sont traités par le système pour le calcul de la vraisemblance. Il est important de noter que même si tous les vecteurs ne sont pas utilisés pour calculer le score du test, tous ces vecteurs doivent être extraits. En effet, un sous-échantillonnage réalisé durant la phase d'extraction des paramètres ne permettrait plus de calculer les paramètres dynamiques (Δc et $\Delta\Delta c$).

% de trames traitées	**100**	**50**	**25**
G1 (EER %)	3,48	3,86	4,23
G2 (EER %)	2,94	3,48	4,60
mem. (MB)	7,84	7,84	7,84
mem. rel. (%)	100	100	100
temps CPU (s)	2,06	1,20	0,68
temps rel. (%)	100	58	33

Tab. 5: Évolution des performances (EER) en fonction du nombre de trames utilisées pour la calcul de la vraisemblance du modèle de locuteur. La consommation des ressources (temps CPU et mémoire) est donnée de manière relative par rapport au système de base.

Les résultats présentés dans le tableau 5 sont obtenus en ne traitant qu'une trame sur deux ou une trame sur quatre. Le traitement d'une trame sur n permet de réduire le temps de calcul de façon considérable. Cependant, les performances du système sont fortement dégradées si le ratio entre les trames reçues et les trames traitées atteint 0,25. Dans ce cas, le taux d'erreurs moyen sur G1 et G2 passe de 3,21% (avec 100% des trames traitées) à 4,41% (avec un quart des trames traitées) soit une augmentation relative de près de 40%.

Toutefois, la sélection ici opérée consiste en un sous-échantillonnage régulier et pourrait sans doute être améliorée par l'utilisation de critères de sélection plus pertinents. Une sélection périodique présente, néanmoins, l'avantage de ne nécessiter aucune analyse des vecteurs de paramètres à traiter.

5.4. Conclusion

Dans les trois sous-parties précédentes, nous avons étudié l'influence qu'avait, indépendamment les uns des autres, trois paramètres : le nombre de composantes des modèles GMM, la taille du vecteur acoustique et le nombre de trames acoustiques traitées.
Deux configurations, correspondant à 2 situations distinctes, sont détaillées ci-après :

système minimal cette solution correspond à celle qui nécessiterait le moins de ressources ;

meilleur compromis cette configuration est celle qui représente le meilleur compromis entre les ressources matérielles et les performances.

Les performances relatives à chaque configuration sont présentées dans le tableau 6.

% Systèmes	**ref.**	**minimal**	**compromis**
G1 (EER %)	3,48	7,72	4,77
G2 (EER %)	2,94	7,34	2,94
mem. (MB)	7,84	1,37	2,19
mem. rel. (%)	100	17	28
temps CPU (s)	2,06	0,04	0,24
temps rel. (%)	100	1,7	11,7

Tab. 6: Comparaison des performances obtenues par le système LIA de référence, sa configuration minimale et le meilleur compromis. La consommation des ressources (temps CPU et mémoire) est donnée de manière relative par rapport au système de base.

Système minimal Le système minimal correspond à celui pour lequel les valeurs minimales de chaque paramètre ont été choisies, *i.e.* les modèles GMM comprennent 32 gaussiennes, les vecteurs acoustiques contiennent 20 coefficients ($20c$ et $20\Delta c$) et la vraisemblance est estimée en utilisant une trame sur quatre. Dans cette configuration, le système a des performances nettement en deçà du système de référence. Nous pouvons noter une augmentation relative du taux d'erreurs moyen de 130% ; cependant le taux d'erreurs reste proche de 7% ce qui en fonction de l'application finale choisie peut s'avérer suffisant. Dans le même temps, cette solution nécessite beaucoup moins de ressources. En effet, le temps CPU est divisé par 60 et le pic mémoire passe de 7,84Mo à 1,37Mo soit une baisse relative de 83%.

Meilleur compromis Pour le projet MoBio, l'intégration est prévue dans le téléphone Nokia© N900, pour lequel les ressources disponibles sont supérieures à celles requises par le système minimal. Un compromis entre les 3 paramètres a donc été choisi afin d'obtenir un système ayant de meilleures performances que le système minimal tout en nécessitant moins de ressources. Dans cette configuration, les modèles GMM contiennent 128 composantes Gaussiennes, les vecteurs acoustiques sont composés de 30 coefficients (10 statiques, 10 dynamiques de premier ordre et 10 dynamiques de second ordre) et la vraisemblance est estimée en utilisant une trame sur deux. Ce système obtient des performances satisfaisantes. En effet, le taux d'erreurs moyen passe de 3,21% à 3,84%, soit une hausse relative de 20% alors que dans le même temps les ressources requises sont nettement diminuées : le temps CPU est divisé par 10 et le pic mémoire est divisé par 4.

6. Conclusion et perspectives

Au cours de ces dernières années, les systèmes de reconnaissance automatique du locuteur ont fait des progrès significatifs, à tel point que des applications grand-public deviennent envisageables. Ces applications soulèvent, dans le contexte de l'embarqué, de nouvelles problématiques jusqu'alors peu considérées dans les approches classiques (où les ressources ne sont généralement pas limitées). Dans ce travail, nous avons proposé deux configurations du système UBM/GMM de vérification du locuteur développé au LIA. Ces configurations, faisant varier trois des paramètres majeurs des systèmes de reconnaissance du locuteur, sont adaptées au contexte embarqué et particulièrement aux ressources restreintes.
La première configuration proposée nécessite uniquement 28% de l'espace mémoire et 12% du temps CPU utilisés par le système état de l'art du LIA (reposant sur la plateforme Mistral) alors que le taux d'erreurs n'augmente que de 20% (restant sous le seuil des 4%). Une seconde solution, plus « extrême », a été proposée pour des applications moins sécurisées. Cette configuration obtient un taux d'erreurs inférieur à 8% mais ne nécessite que 17% de l'espace mémoire et 1,7% du temps CPU requis par le système de référence.
De futurs travaux viseront à développer un système scalable dynamique, s'adaptant aux ressources disponibles sur le téléphone portable à un instant donné tout en garantissant les meilleures performances possibles.

Remerciements

Cette étude a été en partie financée par le projet européen MoBio[4]. Ce projet a pour objectif le développement d'une solution pour la biométrie multi-modale embarquée sur un téléphone portable.

Références

[1] E. Bailly-Bailliere, S. Bengio, F. Bimbot, M. Hamouz, J. Kittler, J. Mariethoz, J. Matas, K. Messer, V. Popovici, F. Poree, et al. The BANCA database and evaluation protocol. *Lecture Notes in Computer Science (LNCS)*, 2688 :625–638, 2003.

[2] Jean-François Bonastre, Nicolas Scheffer, Driss Matrouf, Corinne Fredouille, Anthony Larcher, Alexandre Preti, Gilles Pouchoulin, Nicholas Evans, Benoît Fauve, and John S.D. Mason. ALIZE/SpkDet : a state-of-the-art open source software for speaker recognition. In *Speaker and Language Recognition Workshop (IEEE Odyssey)*, 2008. http ://mistral.univ-avignon.fr/.

[3] Eric Charton, Anthony Larcher, Christophe Lévy, and Jean-Francois Bonastre. Mistral : open source biometric platform. In *Symposium on Applied Computing (ACM)*, Sierre (Switzerland), march 2010.

[4] G. Gravier. SPro : speech signal processing toolkit. *Software available at http ://gforge. inria. fr/projects/spro.*

[5] Patrick Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel. Factor analysis simplified. In *IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP*, volume 1, 2005.

[6] Driss Matrouf, Nicolas Scheffer, Benoit Fauve, and Jean-Francois Bonastre. A straightforward and efficient implementation of the factor analysis model for speaker verification. In *International Conference on Speech Communication and Technology*, 2007.

[7] M. Przybocki and A.F. Martin. NIST speaker recognition evaluation chronicles. In *Speaker and Language Recognition Workshop (IEEE Odyssey)*. ISCA, 2004.

[8] A. Solomonoff, W.M. Campbell, and I. Boardman. Advances in channel compensation for svm speaker recognition. In *IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP*, volume 1, pages 629–632, 18-23, 2005.

[4] `http://www.mobioproject.org/`