SP3
Segmentation et Authentification de la parole

Rapport d'études sur la description de l'algorithme d'authentification vocale dépendante du texte

Anthony Larcher[1]

1 : Laboratoire d'Informatique d'Avignon - Université d'Avignon
Tél : +33 (0) 4 90 84 35 55 - Fax : + 33 (0) 4 90 84 35 01
anthony.larcher@univ-avignon.fr

4 juillet 2006

Introduction

Par nature, un système embarqué doit gérer au mieux les ressources disponibles afin d'obtenir un résultat optimal en fonction des moyens matériels à sa disposition. Pour un système de reconnaissance vocale embarqué, cela se traduit par deux contraintes : un coût de calcul minimum et une occupation mémoire minimum. Ces exigeances s'ajoutent au besoin de sécurité d'un tel système, c'est à dire à ses performances, qui implique une robustesse au milieu d'utilisation du système (environnement bruité). Ce document propose une réponse au problème de la reconnaissance vocale embarquée en y ajoutant des informations temporelles issues de la vidéo dans le but d'améliorer les performances.

Du point de vue de l'utilisateur, le système s'utilise de la façon suivante :
durant une phase d'apprentissage l'utilisateur s'enregistre environ trois fois. Pour chaque enregistrement il prononce un même mot de passe de son choix. Les informations sont éventuellement transférées sur le matériel embarqué dans le cas où l'entraînement s'effectue sur un autre système. Une fois cette phase effectuée, il n'aura par la suite qu'à prononcer son mot de passe face à la caméra de son appareil pour être reconnu par son PDA ou son téléphone portable.

La méthode proposée est basée sur des modèles GMMs (Gaussian Mixture Model) utilisés dans une architecture algorithmique à trois niveaux. Après l'introduction des outils et de l'architecture propre à ce système nous présenterons les techniques mises en œuvres pour utiliser ce modèle et tirer partie des informations issues de la vidéo.

GMM et hypothèses bayésiennes

Dans le domaine de la biométrie, le formalisme le plus employé repose sur la thérorie Bayésienne de la décision. Un test est représenté par un rapport d'hypothèses. Soit un individu connu, L, et un signal audio , S, enregistré par un individu dont l'identité doit être déterminée. Pour décider si le signal S a été enregistré par l'individu L, le test bayésien fait intervenir les hypothèses suivantes :

H : le signal S provient de L
$\overline{H}$: le signal S ne provient pas de L

Le test qui permet de décider entre ces deux hypothèses est un rapport de vraisemblances (Likelihood Ratio) :

$$P_0(S) = P(H|S) \; et \; P_1(S) = P(\overline{H}|S)$$

où $P_0(S)$ est la probabilité d'avoir une bonne identification sachant qu'on a enregistré S et $P_1(S)$ est la probabilité d'avoir une imposture sachant qu'on a enregistré S. Le rapport d'hypothèses bayésien est ici :

$$\frac{P_0(S)}{P_1(S)} \begin{array}{c} accept \\ > \\ < \\ reject \end{array} Seuil$$

Un seuil apparait dans cette expression, celui-ci est au centre des considérations bayésiennes puisque c'est ce seuil qui va fixer la limite d'acceptation d'une hypothèse ou de l'autre et influer sur les performances du système biométrique. Ce seuil est très lié à l'application visée.

En utilisant les lois de Bayes on obtient :

$$\frac{P_0(S)}{P_1(S)} = \frac{P(H|S)}{P(\overline{H}|S)} = \frac{\frac{P(H)}{P(S)} \times P(S|H)}{\frac{P(\overline{H})}{P(S)} \times P(S|\overline{H})} = \frac{P(H).P(S|H)}{P(\overline{H}).P(S|\overline{H})}$$

Les *a priori* sont intégrés au seuil :

$$\frac{P(H).P(S|H)}{P(\overline{H}).P(S|\overline{H})} \begin{array}{c} accept \\ > \\ < \\ reject \end{array} Seuil \Longrightarrow \frac{P(S|H)}{P(S|\overline{H})} \begin{array}{c} accept \\ > \\ < \\ reject \end{array} Nouveau\,Seuil$$

En pratique le test est réalisé dans le domaine logarithmique.

$$log(P(S|H) - log(P(S|\overline{H}) \begin{array}{c} accept \\ > \\ < \\ reject \end{array} Seuil$$

Depuis quelques années la façon la plus courante de représenter un locuteur en reconnaissance indépendante du texte est d'utiliser des mélanges de gaussiennes (GMM pour Gaussian Mixture Model) [Reynolds et Rose, 1995], [Douglas A. Reynolds et Dunn, 2000], [Bimbot *et al.*, 2004]. Un GMM est une fonction de densité de probabilité (de dimension variable) composée d'une somme de

Gaussiennes utilisée pour approximer une fonction de densité de probabilité complexe en soulignant ses différents modes propres.

Pour obtenir une modélisation pertinente des caractéristiques d'un locuteur ce GMM est entrainé à partir des vecteurs issus du signal de parole de ce locuteur (on utilise par exemple les MFCC). Si il existe plusieurs techniques permettant de calculer les paramètres du GMM, la plus courante consiste à maximiser la vraisemblance en utilisant l'algorithme EM (Expectation-Maximization) [Bimbot *et al.*, 2004] couplé à une entité de maximisation de la vraisemblance (ML pour Maximum Likelihood). La vraisemblance d'un GMM λ pour une séquence de T vecteurs d'entraînement $X = \{\overrightarrow{x_1}, ..., \overrightarrow{x_T}\}$ s'écrit :

$$p(X|\lambda) = \prod_{t=1}^{T} p(\overrightarrow{x}_t|\lambda)$$

L'idée de base de l'algorithme EM est donc de commencer avec un modèle initial λ et d'estimer un nouveau modèle λ' qui vérifie $p(X|\lambda') \geq p(X|\lambda')$. Le nouveau modèle devient alors le modèle initial pour l'iteration suivante jusqu'à convergence du procédé. Le résultat de cette méthode est une représentation du locuteur dans un espace de dimension dépendante du nombre de paramètres (le locuteur peut être représenté par les moyennes, les variances et les poids des gaussiennes dans le modèle). La reconnaissance consiste ensuite à décider si une séquence a été prononcée par un locuteur L ou par un locuteur différent de L en effectuant un rapport de vraisemblances avec les hypothèses suivantes :

H : la séquence S a été prononcée par L
$\overline{H}$: la séquence S n'a pas été prononcée par L

Si le modèle de L peut être assez bien estimé, le modèle de $\overline{L}$ doit potentiellement représenter tous les locuteurs excepté L. Une première façon de représenter $\overline{L}$ est d'utiliser une "cohorte de locuteurs" pour représenter tous les locuteurs. C'est à dire un groupe de locuteurs divers qui représente les locuteurs autres que L. Ceci demande de nombreux modèles de locuteurs et le calcul de la vraisemblance avec tous ces modèles demande beaucoup de ressources. Une autre technique majoritairement employée est de créer un *modèle du monde* (UBM pour Universal Background Model) [Bimbot *et al.*, 2004], [Carey *et al.*, 1991]. Ce modèle GMM est créé à partir de données audio de nombreux locuteurs représentatifs d'une population générique. La création de ce modèle nécessite une grande quantité de données et de temps de calcul mais présente l'avantage de pouvoir être utilisé pour tous les locuteurs testés. Lors de la phase de test, il s'agit alors de ne calculer qu'un rapport de vraisemblances par locuteur.

Une façon de modéliser l'information temporelle contenue dans la parole est d'employer des modèles de Markov cachés (HMM) [Rabiner, 1989]. Très utilisés en reconnaissance de la parole car ils permettent une bonne représentation de la structure temporelle du langage, les HMMs peuvent de la même façon être utilisés en reconnaissance du locuteur. Les HMM permettent de représenter un système à états finis par ses différents états et par les transitions probabilistes entre ces états. La grande majorité des travaux en reconnaissance du locuteur utilise des GMMs pour représenter les distributions des états des HMMs. Les HMMs sont généralement coûteux à utiliser et nécessitent une quantité de données assez importante. Ils présentent cependant l'avantage de s'adapter de façon très efficace aux variations de l'environnement ou du locuteur. Les variantes quant à l'utilisation des HMM en reconnaissance du locuteur portent essentiellement sur le choix des états et sur la façon de les adapter. Une première idée est de tenir compte des informations linguistiques, c'est à dire modéliser les phonèmes (généralement par des HMMs à trois états émetteurs) ou des triphones [Matsui et Furui, 1993]. Mais le nombre d'états peut aussi être fixé sans tenir compte d'aucune signification physique évidente.
Une fois le nombre d'états choisi, il faut entrainer les GMMs de ces états. Il est possible d'apprendre autant de GMMs pour un locuteur qu'il y a d'états dans le HMM, ou d'utiliser une représentation semi-continue, et de n'apprendre qu'un GMM par locuteur en apprenant les transformations qui permettent de passer de ce GMM global à chaque état du HMM en modifiant les poids, les variances ou les moyennes de chaque gaussienne [Levy *et al.*, 2006]. On peut également partager des

états dans le cas de phonèmes contextuels.

Chapitre 1

Présentation du système

Le système de reconnaissance est fortement inspiré du système GDW (Gaussian Dynamic Warping) décrit dans [Bonastre *et al.*, 2003].

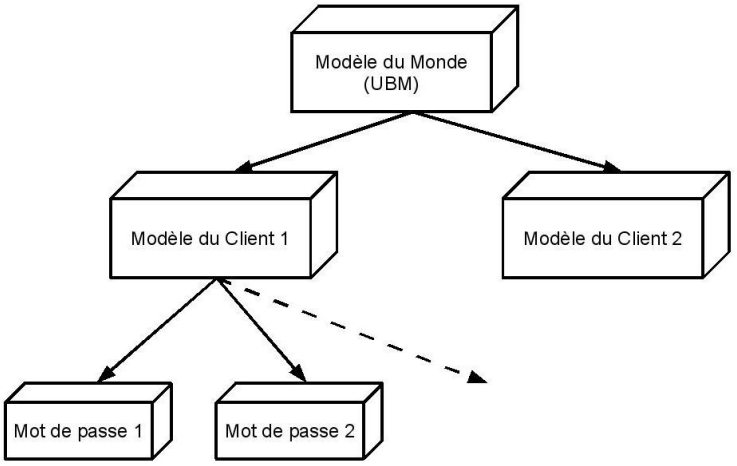

FIG. 1.1 – Schéma d'ensemble du système

Comme le montre la figure 1.1, la modélisation comporte trois niveaux.
Le premier niveau est un GMM, modèle du monde (UBM) obtenu à partir de données provenant de nombreux locuteurs enregistrés au préalable dans différents milieux. Ce GMM modélise l'ensemble des locuteurs ainsi que l'ensemble des environnements possibles. Ce modèle peut être assez important puisqu'il sera le seul GMM audio stocké complètement en mémoire et qu'il est calculé hors ligne.
Le deuxième niveau est dépendant du locuteur. On a sur la figure 1.1 deux modèles de locuteurs différents. Chaque modèle est obtenu par une adaptation de l'UBM en utilisant toutes les données disponibles pour un locuteur.
Le troisième niveau est dépendant du locuteur et du texte. Chaque locuteur peut avoir plusieurs mots de passe qui sont modélisés par des HMMs. C'est à ce niveau qu'intervient l'information temporelle issue de la vidéo.
La partie suivante détaille ces trois niveaux et les techniques utilisées pour les obtenir.

Chapitre 2

Adaptation des différents modèles

La façon dont le modèle UBM est obtenu a été décrite précédemment, cette partie étant hors ligne, elle ne nécessite pas de modification. Toutes les données audio utilisées pour ce système sont extraites sous forme de coefficients cepstraux. Les modèles de locuteur sont habituellement obtenus par une adaptation MAP (Maximum A Posteriori), ils peuvent également l'être par des adaptations du type MLLR (Maximum Likelihood Linear Regression). Cependant afin de ne pas stocker les modèles complets en mémoire on a préféré une adaptation LIAMAP (Unique Linear Transformation) décrite dans [Matrouf *et al.*, 2003] et [Levy *et al.*, 2006]. Cette adaptation peut être effectuée sur une machine et le résultat transféré par la suite sur le téléphone ou le PDA.

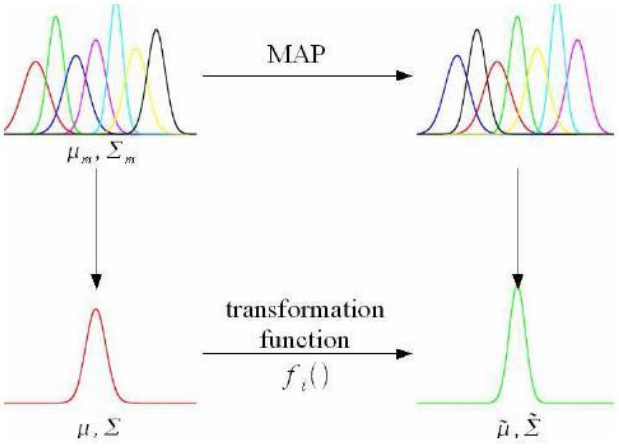

FIG. 2.1 – LIAMAP : Méthode de calcul d'une transformation unique pour toutes les gaussiennes d'un GMM [Levy *et al.*, 2006]

La figure 2.1 illustre la façon dont on obtient la transformation qui permet de passer du modèle UBM au modèle d'un locuteur. On effectue une adaptation MAP standard et on fusionne ensuite toutes les gaussiennes du GMM de départ pour n'en garder qu'une, et de la même façon on ne garde qu'une gaussienne pour le GMM final. On calcule alors la transformation qui permet de passer de la gaussienne de départ à la gaussienne finale. Cette transformation est une transformation linéaire de la forme :

$$\mu_{StateGMM} = \alpha \times \mu_{GlobalGMM} + \beta$$
$$\sigma_{StateGMM} = \alpha^2 \times \sigma_{GlobalGMM}$$

où μ est la moyenne de la gaussienne et σ sa variance.

Les modèles de mot de passe, éléments du troisième niveau sont des HMMs semi-continus pour lesquels chaque état est un GMM. Ces GMMs sont obtenus en effectuant une ULT puis en

ré-estimant les poids des gaussiennes et enfin en sélectionnant les N gaussiennes les plus représentatives. Cette sélection permet de réduire la taille mémoire nécessaire. L'estimation des poids est effectuée grâce à la méthode WRE (Weight Re-Estimation) développée dans [Levy *et al.*, 2006]. L'estimation des probabilités de transition du HMM du modèle de mot de passe va être expliquée dans la partie suivante. Ces estimations et calculs peuvent également être effectués hors ligne puis transférés sur le mobile.

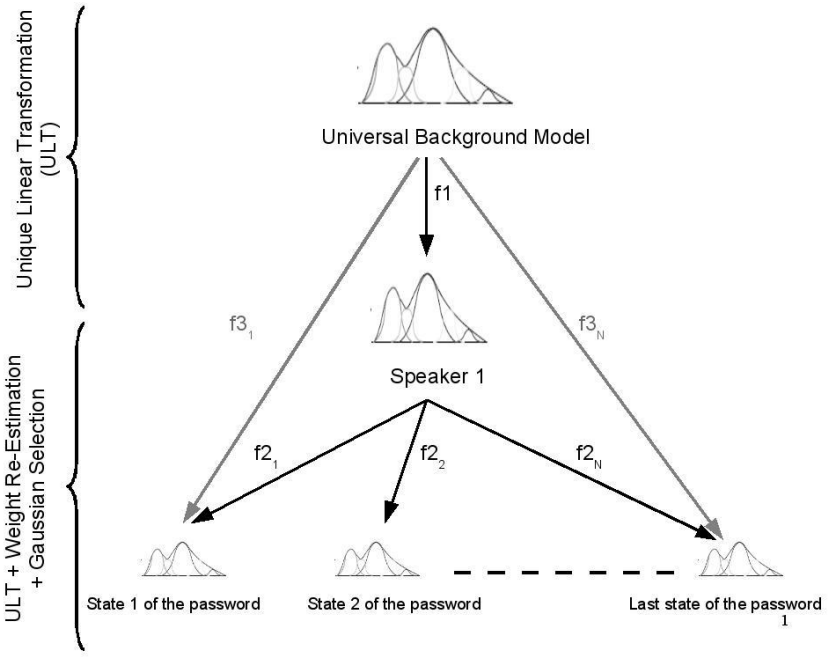

FIG. 2.2 – Les transformations successives f1 et f2 permettent d'obtenir les états du HMM

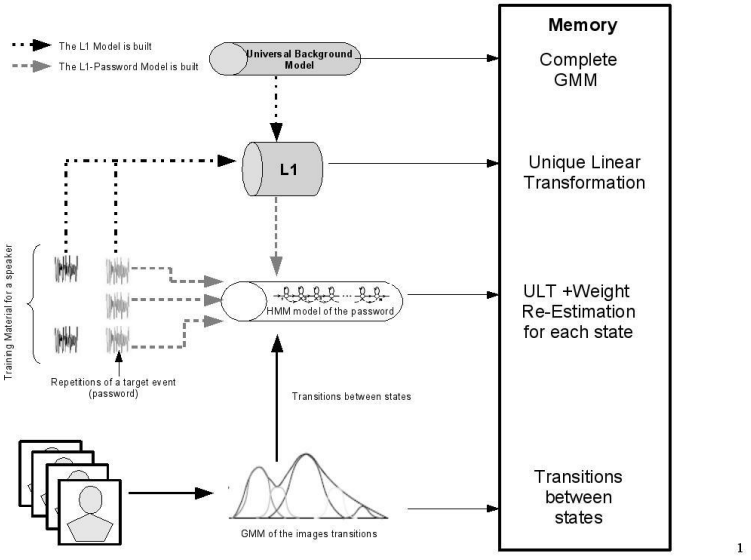

FIG. 2.3 – Pour chaque niveau de modélisation on stocke en mémoire les paramètres qui seront utilisés lors du test

Chapitre 3

Apport de la Vidéo

Les transitions du HMM audio sont calculées à partir de la vidéo. Une première étape consiste à extraire l'information de la vidéo. Pour ce faire les transitions entre 2 images consécutives sont modélisées par un GMM de la façon suivante. Chaque image est découpée en un certain nombre de sous-images régulières. Pour chaque couple d'images consécutives le mouvement de chaque sous-image est évalué (la sous-image la plus probable dans l'image suivante est retenue). Les vecteurs de déplacement de chaque sous-image (cf. figure 3.1) sont concaténés et forment un "vecteur de passage" entre deux images.

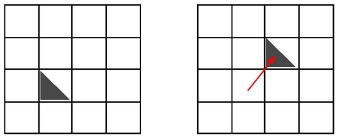

FIG. 3.1 – Vecteur de mouvement d'une sous-image

Ces vecteurs sont utilisés durant la phase d'entraînement pour estimer un modèle GMM des transitions entre images consécutives. Durant une phase d'apprentissage préalable, le modèle GMM des transitions entre images est adapté par le processus LIAMAP pour différentes transitions vidéo. Les transformations sont stockées en mémoire. Un jeu de probabilité de transition est calculé pour chaque GMM de transition vidéo. Le vecteur de transition vidéo actuel est comparé à ces GMM et l'état qui est le plus proche de ce vecteur est sélectionné (cette approche peut être assimilée au suivi de trajectoire acoustique décrit dans [Gagnon *et al.*, 2001]. Pendant la période de sélection de cet état on estime, à partir du flux audio, les probabilités de transitions entre les états du HMM audio. Cette phase nécessite beaucoup de données mais peut être réalisée hors ligne. Le nombre de gaussiennes du GMM de transition vidéo peut également être restreint.

En phase de test, réalisée sur le système embarqué, chaque mot de passe de locuteur connu est testé de la façon suivante.

- Le modèle de locuteur est obtenu en appliquant la transformation mémorisée au modèle UBM.
- On effectue un test de reconnaissance du locuteur classique.
- Pour chaque état du HMM les gaussiennes les plus pertinentes sont sélectionnées.
- Le modèle UBM est adapté grâce à la transformation linéaire apprise pour chaque état du mot de passe.
- Les poids de ces gaussiennes sont modifiés.

– Le décodage est effectué grâce à l'algorithme de Viterbi [Forney, 1973] de façon classique. Les probabilités de transition du HMM audio sont cependant calculées en tenant compte de la "trajectoire vidéo". Pour chaque changement d'image, le vecteur de transition est calculé et les transitions du HMM audio sont celles qui correspondent à l'état vidéo auquel il appartient. Ainsi le décodage audio dépend du flux vidéo.

Chapitre 4

Originalité du système et gains prévus

Le système que nous avons décrit présente plusieurs point particulièrement intéressants pour la reconnaissance de locuteur embarquée. En effet il nécessite peu de données pour apprendre les modèles en ligne. Ce système tire avantage des méthodes probabilistes puisqu'il utilise des modèles GMMs, mais en prenant en compte l'aspect temporel du mot de passe il permet de restreindre la quantité de données nécessaire. L'adaptation des modèles à l'environnement est effectuée au niveau du modèle UBM et n'influe donc pas sur les transformations à appliquer par la suite pour obtenir les modèles de locuteur et de mot de passe.

4.1 Mémoire de la partie Audio

Les transformations linéaires successives *f1* et $f2_i$ (cf. Figure 2.2 qui permettent de passer du modèle UBM à l'état i du HMM sont combinés pour obtenir les transformations *f3*. Seules les transforamtions *f1* et *f3* seront gardées en mémoire. Ainsi que les numéros des gaussiennes conservées lors de la sélection pour chaque état du HMM, et les poids ré-estimés de ces gaussiennes.

Pour un modèle où l'on conserve N gaussiennes par état du HMM on stocke alors :
- Pour le modèle UBM : 512 gaussiennes à 13 composantes soit : 19 968 floats (79 872 octets).
- Pour les transformations *f1* : les coefficients α et β de la transformation (cf. ci-dessus) soit : 2×13 floats : 104 octets.
- Pour chaque transformation $f3_i$: les coefficients α et β de la transformation et pour les N gaussiennes sélectionnées on conserve le numéro de la gaussienne et son poids ré-estimé : $N \times (1\, int + 1\, float)$ soit : $N \times 2 + 2 \times 13$ floats.

On peut considérer pour une première approximation que la mémoire occupée par la modélisation d'un mot de passe pour un locuteur (modèle UBM, modèle du locuteur et modèle HMM compris) pour un HMM à 20 états où chaque état comporte un GMM à 20 gaussiennes est de :

85 256 octets

4.2 Mémoire de la partie Vidéo

On considère que les images sont découpées en 32 sous-images et qu'on crée un GMM à 32 gaussiennes (plus de gaussiennes serait difficile en terme de quantité de données d'apprentissage mais trop restreindre ce nombre pourait nuire à la qualité du modèle). Le HMM est un HMM gauche droite classique avec sauts, à 20 états émetteurs (57 transitions) ; il faut stocker alors 57

probabilités de transition par gaussienne, soit : $32 \times 57 \times 4$ octets : 7296 octets. (cette estimation est moins fiable que pour les données audio, des tests devront permettre se faire une meilleure idée de cette valeur)

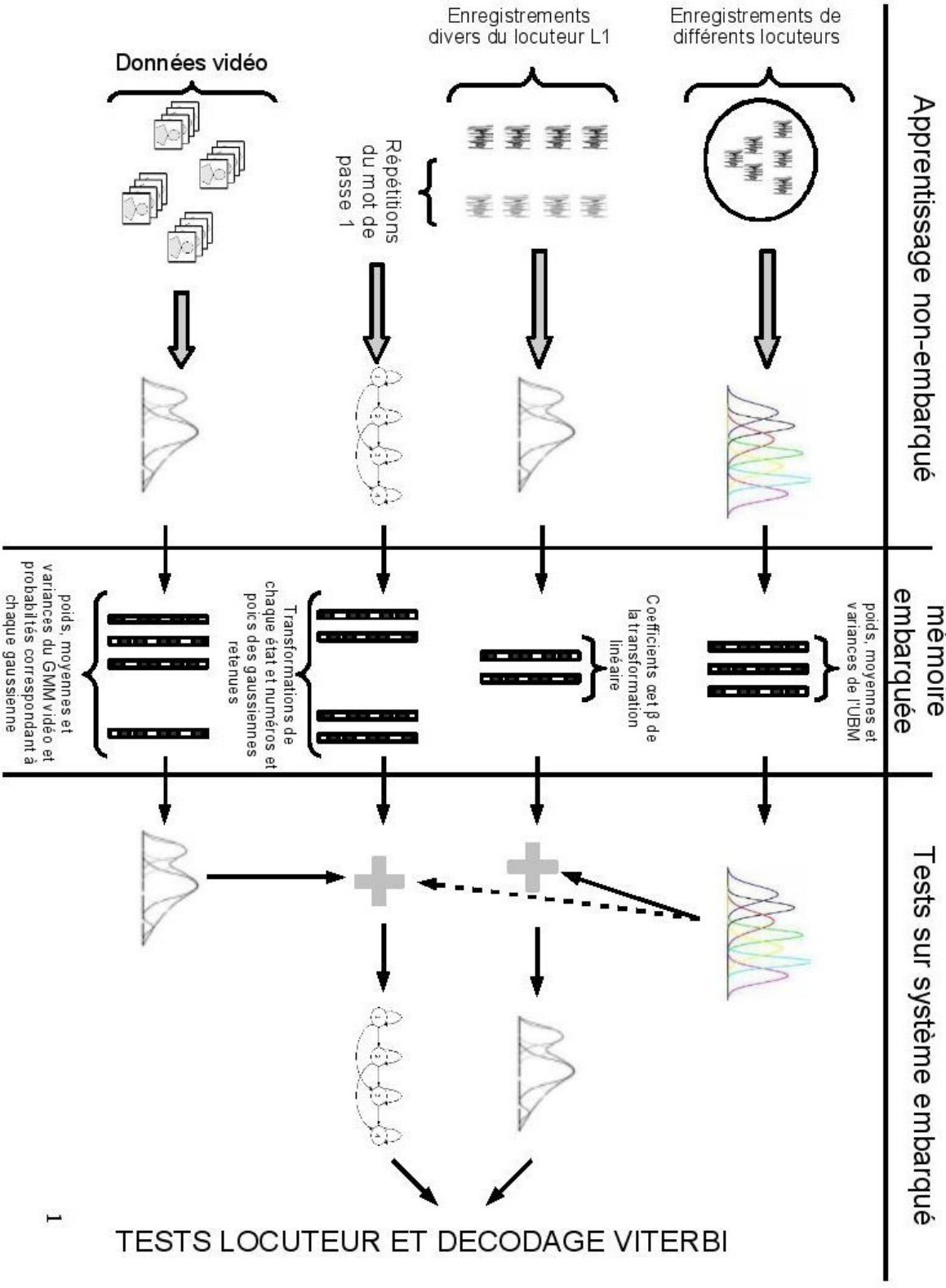

FIG. 4.1 – Schéma général de répartition embarqué/non-embarqué

Bibliographie

[Bimbot *et al.*, 2004] Frederic Bimbot, Jean-François Bonastre, Corinne Fredouille, Guillaume Gravier, Ivan Magrin-Chagnolleau, Sylvain Meigner, Teva Merlin, Javier Ortega-Garcia, Dijana Petrovska-Delacretaz et Douglas A. Reynolds (2004). « A tutorial on text-independent speaker verification ». *EURASIP Journal on Applied Signal Processing*, 4:430–451.

[Bonastre *et al.*, 2003] Jean-François Bonastre, Philippe Morin et Jean-Claude Junqua (2003). « Gaussian dynamic warping (gdw) method applied to text-dependent speaker detection and verification ». *In Eurospeech, Interspeech*, Geneva (Switzerland).

[Carey *et al.*, 1991] Michael J. Carey, Eluned S. Parris et John S. Bridle (1991). « A speaker verification system using alpha-nets ». *In IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP*, volume 1, pages 397–400, Toronto (Canada).

[Douglas A. Reynolds et Dunn, 2000] Thomas F. Quatieri Douglas A. Reynolds et Robert B. Dunn (2000). « Speaker verification using adapted gaussian mixture models ». *Digital Signal Processing*, 10:19–41.

[Forney, 1973] G. David Forney (1973). « The viterbi algorithm ». *Proceedings of the IEEE*, 61(3):268–278.

[Gagnon *et al.*, 2001] Luc Gagnon, Peter Stubley et Ghislain Mailhot (2001). « Password-dependent speaker verification using quantized acoustic trajectories ». *In IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP*, volume 1, pages 449–452, Salt Lake City (USA).

[Levy *et al.*, 2006] Christophe Levy, Georges Linares, Pascal Nocera et Jean-François Bonastre (2006). *Mobile Phone Embedded Digit-Recognition*, chapitre 7 in Digital Signal Processing for In-Vehicle and Mobile Systems 2. Springer Sciences.

[Matrouf *et al.*, 2003] Driss Matrouf, Olivier Bellot, Pascal Nocera, Georges Linares et Jean-Francois Bonastre (2003). « Structural linear model-space transformations for speaker adaptation ». *In Eurospeech, Interspeech*, Geneva (Switzerland).

[Matsui et Furui, 1993] Tomoko Matsui et Sadaoki Furui (1993). « Concatenated phoneme models for text-variable speaker recognition ». *In IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP*, volume 2, pages 391–394, Minneapolis (USA).

[Rabiner, 1989] Lawrence R. Rabiner (1989). « A tutorial on hidden markov models and selected applications in speech recognition ». *Proceedings of the IEEE*, 77(2):257–286.

[Reynolds et Rose, 1995] Douglas A. Reynolds et Richard C. Rose (1995). « Robust text-independent speaker identification using gaussian mixture speaker models ». *IEEE Transactions on Acoustics, Speech and Signal Processing*, 3(1):72–83.