Description du système proposé 5

La spécification d'un système d'intelligence artificielle utilisant des HMM peut s'effectuer en trois phases distinctes, mais interagissantes entre elles (voir figure 5.1). La première phase, que nous nommerons prétraitement par la suite, consiste en l'ensemble des actions nécessaires à la transformation des données en séquences temporelles. La deuxième phase, dite d'apprentissage, consiste en la transformation de certaines des séquences construites en HMM, grâce à un algorithme d'apprentissage, tel que ceux décrits au chapitre précédent. La dernière phase, dite de post-traitement, consiste en l'utilisation des HMM produits en deuxième phase et de séquences produites par la première phase pour effectuer le traitement. Les traitements pouvant être réalisés par un tel système sont très variés : classification, segmentation, analyse, décision,…

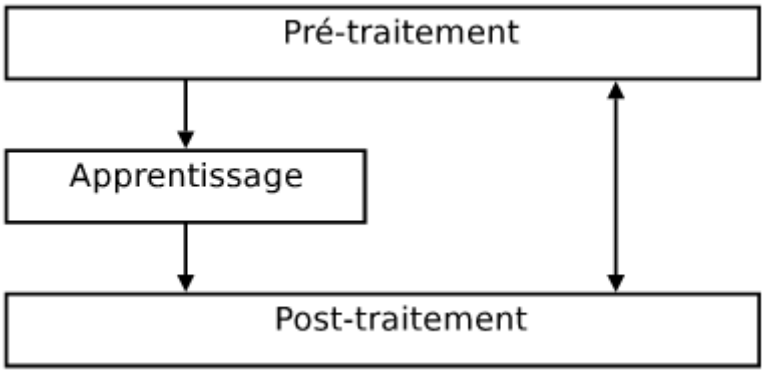

Figure 5.1 – Phases de spécification d'un système d'intelligence artificielle utilisant des HMM.

La phase d'apprentissage joue un rôle central au sein d'un tel système mais, en pratique, peu d'attention lui est accordée dans le cas des HMM. Dans de nombreuses applications des modèles sous optimaux sont utilisés avec succès. Cependant, ces applications s'appuient sur des principes théoriques qui ne sont valables que lorsque les modèles sont optimaux. Par conséquent, il est communément admis que des modèles optimaux permettraient, du moins en théorie, d'améliorer les performances du système d'intelligence artificielle.

La RAP s'applique à ce jour sur de nombreux signaux de qualité différente (fréquence d'échantillonnage, quantification, codage, conditions d'enregistrement). Nous rappelons que la parole est l'un des moyens les plus naturels par lequel des personnes communiquent. La RAP a pour objet la transformation du signal acoustique en une séquence de mots qui,

idéalement, correspond à la phrase prononcée par un locuteur. Les systèmes de reconnaissance qui utilisent comme entrée uniquement le signal acoustique atteignent leurs limites surtout dans des cas de situations environnementales bruités donc réelles. Dans ces cas, l'intégration de l'information visuelle dans le système de reconnaissance peut constituer une voie de solution (Rogozan 1999). A cet effet nous nous intéressons à la mise en œuvre d'un système de reconnaissance intégrant conjointement les deux informations acoustique et visuelle de la parole se sont focalisés sur une interaction sensorielle de type fusion ou intégration. A ce niveau, reste posée la question du ou et comment cette fusion des modalités acoustique et visuelle se passe-t-elle chez l'homme. Pour répondre à cette question, il existe plusieurs modèles cognitifs qui diffèrent de par leur lieu d'intégration des informations en vue de leur intégration. La RAP audiovisuelle est née de l'idée que si l'homme exploite les informations provenant du visage du locuteur pour améliorer l'intelligibilité, la machine peut en faire autant, si d'une part le principe d'intégration des deux modalités est suffisamment bien connu, et si d'autre part les informations visuelles sont exploitées d'une façon optimale (Adjoudani and Benoît 1995).

Dans ce chapitre nous définissons les différentes méthodes que nous utiliserons par la suite dans la partie expérimentale.

5.1 Architecture de système de reconnaissance par fusion audiovisuelle

Le système AVASR comprend trois modules qui sont: le module de reconnaissance acoustique, le module de reconnaissance visuelle et le module de fusion.

Le module de reconnaissance acoustique utilise l'approche stochastique basée sur les modèles de Markov cachées (HMM) qui sont un type particulier des réseaux bayésiens. On processus générique est basé sur trois phases qui sont : la para métrisation du signal acoustique utilisant dans notre cas l'analyse log RASTA-PLP (RelAtive SpecTral Analysis-Perceptual Linear Predictive), l'apprentissage des modèles repose sur une recherche génétique d'un bon modèle parmi une population hétérogène des HMM (contenant différentes architectures) et une optimisation par un algorithme de gradient (Baum-Welch) et leur décodage sur l'algorithme de viterbi. Le module de reconnaissance visuelle utilise la même approche stochastique, il diffère uniquement par la phase de para métrisation basée elle sur la DCT (Discrete Cosine Transform).

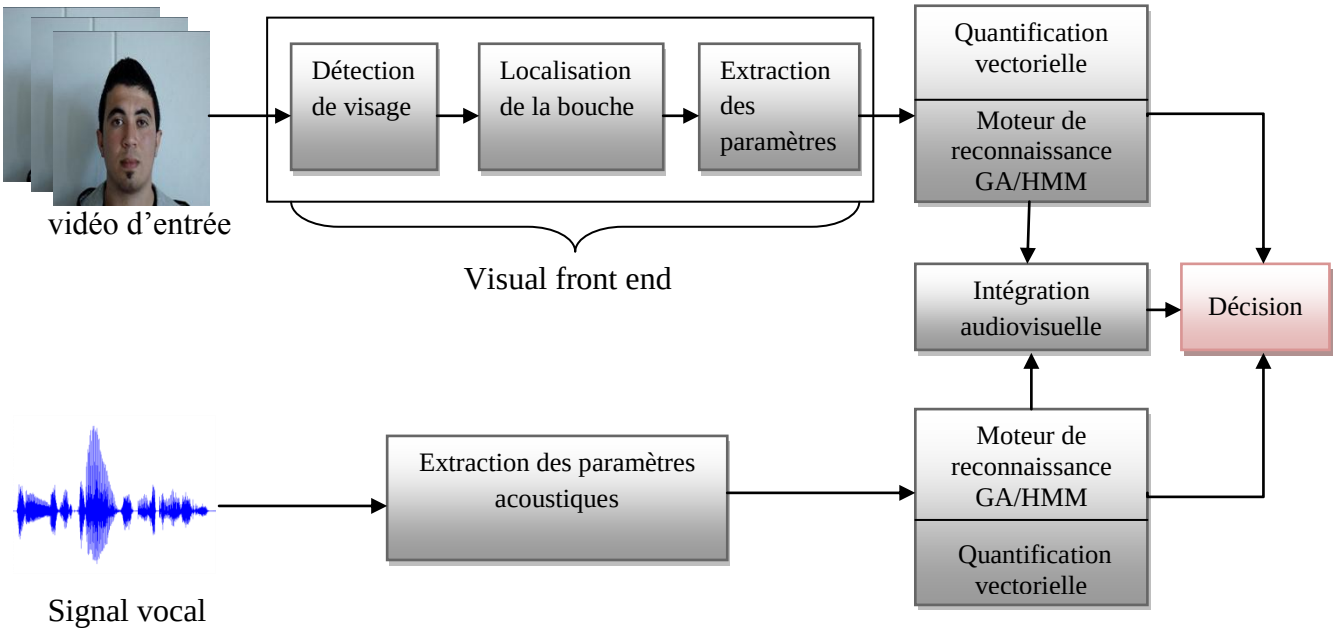

Figure 5.2 – Système d'un AVASR mis en œuvre.

La figure 5.2 présente les différentes étapes dans les processus d'apprentissage et de reconnaissance du système proposé. Chacun des éléments présents sur cette figure sera détaillée dans les prochaines sections.

5.1.1 Traitement visuel

Nous savons que les lèvres véhiculent la majeure partie de l'information visuelle utile pour la compréhension de la parole.

Les êtres humains emploient l'information visuelle de façon subconsciente afin de comprendre les paroles, particulièrement dans des environnements bruyants, mais également quand les conditions acoustiques sont bonnes. Le mouvement des lèvres du locuteur apporte une série d'information importante. L'effet McGurk (McGurk and MacDonald 1976) apporte la preuve en montrant que le cerveau, soumis à des stimuli auditifs et visuels inconsistants, perçoit un son différent de celui qui a été dit.

5.1.1.1 Détection de visage

La détection des visages pose le problème de la localisation des visages présents dans une image d'entrée. Idéalement, la détection fourni aussi leurs dimensions pour un éventuel traitement ultérieur.

Tous les AVASR nécessitent l'identification et le suivi de la ROI, qui peut être soit seulement la bouche, ou une région plus vaste, comme tout le visage. Cela commence généralement par localisation de visage du locuteur, en utilisant un algorithme de détection de visage.

Une avancée majeure dans le domaine a été réalisée par (Viola and Jones 2001). Ces derniers ont proposé une méthode basée sur l'apparence ("Appearance-based methods") rapide et robuste. La renommée de cette approche se base essentiellement sur trois contributions:

- **Algorithme de Viola & Jones**

Comme nous avons déjà mentionnés Viola et Jones ont proposé une méthode basée sur l'apparence ("Appearance-based methods") robuste et tournant à 15 fps pour des images de 384 x 288 pixels sur un pc Intel Pentium III 700Mhz. Ce fut la première méthode en temps réel présentée. La renommée de cette approche est faite sur trois concepts :

A. L'image intégrale

L'algorithme se base sur les caractéristiques de Haar (Haar features) pour localiser les visages présents sur une image d'entrée. Dans le but d'extraire rapidement ces caractéristiques, l'image est représentée sous forme intégrale. En effet, sous cette forme, l'extraction d'une caractéristique à n'importe quel endroit et à n'importe quelle échelle est effectuée en un temps constant tandis que le temps de conversion vers la représentation intégrale ne remet pas en cause ce gain de temps offert par l'utilisation de la représentation en image intégrale. La définition des caractéristiques de Haar et la manière dont la représentation intégrale accélère considérablement leur extraction sont présentés ci-après pour une image en niveaux de gris.

Dans toute image, une zone rectangulaire peut être délimitée et la somme des valeurs de ses pixels calculée. Une caractéristique de Haar est une simple combinaison linéaire de sommes ainsi obtenues.

Plusieurs caractéristiques de Haar peuvent être définies selon le nombre, les échelles, les positions et les dimensions des zones rectangulaires considérées. 4 exemples sont présentés à la figure 5.3.

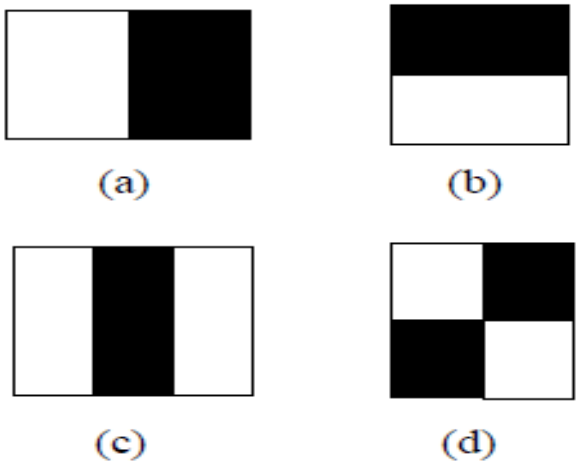

Figure 5.3 – Exemple de 4 caractéristiques de Haar. La somme des valeurs des pixels appartenant aux zones encadrées claires est soustraite à la somme des valeurs des pixels appartenant aux zones encadrées sombres pour obtenir la caractéristique de Haar. Chacune des quatre caractéristiques de Haar est représentée avec son cadre de détection respectif.

L'image intégrale est représentée mathématiquement par :

$$ii(x,y) = \sum_{x' \leq x, y' \leq y} i(x', y') \tag{5.1}$$

$$\forall\ 0 < x \leq width, 0 < y \leq height. \tag{5.2}$$

ou i(x, y) est l'image d'origine et i(x0, y0) l'image sous sa nouvelle représentation. Ainsi chaque pixel a pour valeur la somme des valeurs des pixels compris dans le rectangle défini par le coin supérieur gauche de l'image et lui-même.

Le calcul de la somme des valeurs des pixels appartenant à une zone rectangulaire s'effectue donc en accédant seulement à quatre pixel de l'image intégrale : Soit un rectangle ABCD dont les sommets sont nommés dans le sens des aiguilles d'une montre en commençant par le sommet supérieur gauche et soit x la valeur sous la représentation intégrale d'un sommet *X* du rectangle ($X \in$ {A, B, C, D}). La somme des valeurs des pixels appartement à ABCD est, quelle que soit sa taille, donnée par c - b - d + a. Une caractéristique de Haar étant une combinaison linéaire de tels rectangles ABCD, son calcul se fait alors en un temps indépendant sa taille.

B. Algorithme d'apprentissage basé sur Adaboost

Pour localiser les visages sur l'image d'entrée, cette dernière est scannée par une fenêtre de dimension déterminée. La fenêtre parcourt l'image et son contenu est analysé pour savoir

s'il s'agit d'un visage ou non. Comme dit plus haut, les caractéristiques de Haar sont extraites pour effectuer la classification et de ce fait la représentation intégrale de l'image accélère l'analyse. Mais, pour une fenêtre de 24x24 pixels il y a 45396 caractéristiques de Haar, les traiter toutes prendrait beaucoup trop de temps pour une application en temps réel. Pour surmonter ce problème, une variante de la méthode de boosting Adaboost est utilisée. Ci-dessous Adaboost est brièvement présenté suivi de sa variante qui constitue le deuxième apport du travail de Viola & Jones.

Adaboost est une méthode d'apprentissage permettant de "booster" les performances d'un classifieur quelconque nommé "classifieur faible". L'idée est de faire passer les candidats à classifier à travers plusieurs classifieurs faibles, chacun étant entrainé en portant plus d'attention sur les candidats mal classifiés par le classifieur précédent.

Pour arriver à ce résultat des poids sont associés aux échantillons du set d'entrainement ((xi, yi) i = 1, …., m), tout d'abord de manière équilibrée :

$$w_i^0 = \frac{1}{m} \tag{5.3}$$

pour i = 1, …, m. Le 0 en exposant indique qu'il s'agit des poids initiaux.
Adaboost sert donc à booster un classifieur déjà existant et à priori chaque classifieur faible possède le même espace d'entrée. Dans la variante d'Adaboost de Viola & Jones, les classifieurs faibles hj ∈H ont pour entrée une caractéristique de Haar différente. Adaboost s'apparente alors à une sélection de caractéristiques (feature selection).

Cette variante d'Adaboost est utilisée lors de l'apprentissage pour sélectionner les caractéristiques de Haar les plus à même de détecter un visage et permet ainsi de surmonter le problème du nombre élevé de caractéristiques de Haar existant pour une fenêtre de recherche.

C. Cascade

L'idée de base derrière le concept de Cascade est que parmi l'ensemble des candidats, c'est-à-dire l'ensemble des états de la fenêtre de recherche, une partie peut être éliminée sur base de l'évaluation de seulement quelques caractéristiques de Haar. Une fois cette élimination effectuée, les candidats restants sont analysés par des classifieurs forts plus complexes (utilisant plus de caractéristiques de Haar) demandant un plus grand temps de traitement. En utilisant plusieurs « étages » de ce type, le processeur évite d'effectuer des analyses lourdes en temps de calcul sur des échantillons pour lesquels il est rapidement

possible de se rendre compte qu'ils sont négatifs. Le processus de classification apparait alors comme une cascade de classifieurs forts de plus en plus complexes ou à chaque étage les échantillons classifiés négatifs sont sortis tandis que les échantillons classifiés positifs sont envoyés aux classifieurs suivants. Ceci est représenté à la figure 5.4.

Si le premier étage rejette un faux négatif, c'est un gros problème car il ne sera jamais récupéré par la cascade. Autrement dit c'est un visage qui ne sera pas détecté. Par contre, si le premier étage transmet un faux positif, il pourra toujours être éliminé aux étages suivants de la cascade. Ce petit raisonnement permet de mettre en évidence que les premiers nœuds constitutifs de la cascade peuvent se permettre d'avoir un taux de faux positifs élevés (de l'ordre de 40-50%) mais doivent absolument assurer un taux de détection maximum.

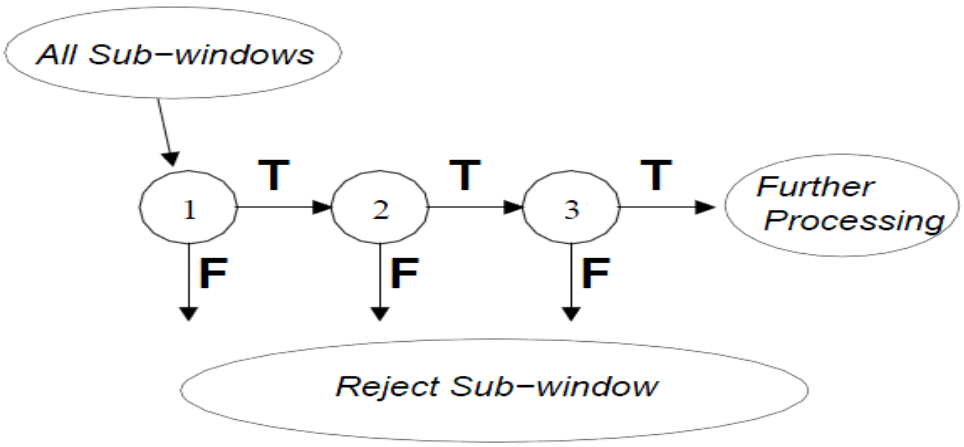

Figure 5.4 – Cascade de classifieurs forts. A chaque étage, uniquement les candidats classifiés positifs sont transmis à l'étage suivant.

Ce concept permet donc à l'algorithme de consacrer son temps à de longues analyses complexes uniquement lorsque cela en vaut la peine. Il s'agit à nouveau d'un mécanisme qui accélère la vitesse d'exécution de la méthode proposée par Viola & Jones.

5.1.1.2 Localisation de la bouche

Après la détection de visage avec l'utilisation de l'algorithme de Viola-Jones, il est possible d'extraire des zones à partir de la géométrie du visage trouvé, où les points devraient être. Ces zones sont les entrées relatives à l'extraction de la ROI.

Au moment où il est exécuté en utilisant la teinte distincte des lèvres. La lumière se reflète sur les lèvres et ce point est récupéré par une valeur de teinte définie. Contrairement

aux autres méthodes, cette méthode n'est pas indépendante de lumière, ainsi l'intensité et la direction de la lumière peut influencer les résultats (Pai et al. 2006).

Un visage humain typique suit un ensemble de normes anthropométriques, qui ont été utilisés pour affiner la recherche d'une caractéristique faciale particulière pour des régions plus petites de visage. Nous utilisons les étapes génériques suivantes pour la détection des caractéristiques faciales et l'extraction à partir de l'image du visage localisée (Khandait et al. 2009):

1) Pour une image couleur, la convertir en image en niveaux de gris. Réglez l'intensité des deux types d'images.
2) Appliquer projection horizontale pour trouver frontière gauche et droite de visage. Appliquer projection verticale pour trouver la frontière supérieure et inférieure de visage où trouver région d'intérêt d'une image.
3) Trouvez le gradient de la ROI de l'image détectée en utilisant Sobel / Prewitt opérateur de détection des frontières et ensuite prenez la partie inférieure du visage et prenez sa projection verticale pour obtenir la bouche.
4) Dessiner zone rectangulaire sur la composante caractéristique détectée.

5.1.1.3 Extraction des paramètres visuels

Dans cette étude l'extraction des caractéristiques vidéo est effectuée avec le DCT (Rodomagoulakis 2008). Il existe plusieurs types de caractéristiques qui peuvent être utilisées pour chiffrer les informations présentes dans une image. Nous avons appliqué une version modifiée de la DCT qui utilise les données contenues dans une image pour la compresser. Par exemple, la compression de l'image en format JPEG utilise cette méthode. La compression des données disponible dans l'image permet de rendre le travail de l'algorithme d'apprentissage plus facile. En plus la DCT est utilisée dans le domaine d'authentification et vérification du locuteur (Sanderson and Paliwal 2002). Cette étape se déroule en deux phases : La première est la phase de découpage de l'image, résultant de la phase de prétraitement, en sous-images. Ensuite, la seconde phase qui est l'extraction de vecteurs de caractéristiques consiste à appliquer la DCT. Ces étapes seront détaillées dans les paragraphes suivants.

5.1.1.3.1 Découpage de l'image

Le découpage de l'image consiste à subdiviser l'image en entrée en sous-images de dimension fixe qui se chevauchent dans les deux directions, l'axe des y et l'axe des x. Le découpage de l'image se passe de la manière suivante : la première sous-image de dimension N × N pixel se trouve aux coordonnées (0, 0), (N, N) de l'image d'entrée. La seconde sous-image chevauche la première sous- image d'une superposition de c pixels en direction de l'axe des x. Donc, la seconde correspond à la sous-image de coordonnées (N – c, 0), (2N – c, N). La troisième sous-image a une superposition dans la direction de l'axe des y avec la première sous-image. La troisième correspond à la sous-image de coordonnées (0, N - c), (N, 2N - c). Cette procédure se répète récursivement jusqu'à ce que toute l'image en entrée soit traitée. Le résultat d'un tel découpage est montré par la figure 5.4. Dans le cadre de ce projet, la dimension des sous-images a été fixée à 16 × 16 pixels avec une superposition de 8 pixels.

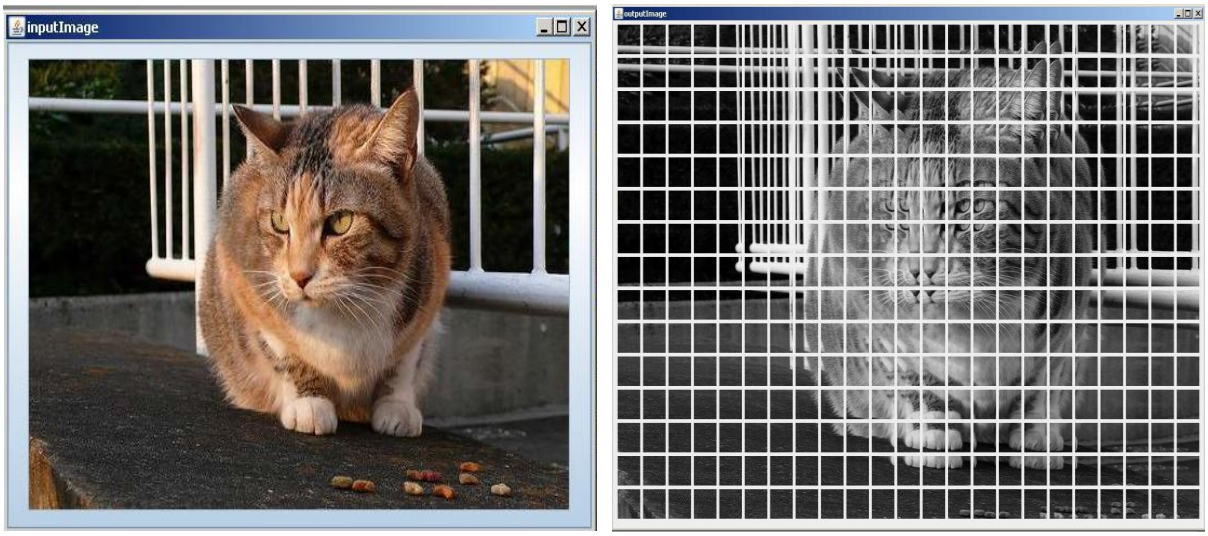

Figure 5.4 – Découpage de l'image de l'histogramme.

5.1.1.3.2 Extraction de caractéristiques

La phase d'extraction de caractéristiques présente un passage de l'image du domaine spatial au domaine fréquentiel. Comme mentionné au début de ce chapitre, nous avons choisi d'utiliser la DCT. Cette méthode consiste à présenter chaque image comme une matrice de vecteurs ou chaque vecteur correspond à une sous-image résultant d'un découpage régulier de l'image. Ces vecteurs sont des coefficients qui correspondent à des combinaisons linéaires de fonctions cosinusoïdales, ces fonctions sont la base du domaine fréquentiel.

Plus formellement, étant donnée une image qui est présentée par une matrice de sous-images de dimension N × N, ces sous-images sont le résultat du découpage précédemment expliqué. Pour chaque image I un vecteur de DCT est extrait. DCT transforme chaque composante de couleur en coefficients DCT en utilisant l'équation suivante (Gupta and Garg 2012):

$$F(u,v) = \frac{1}{\sqrt{MN}}\alpha(u)\alpha(v)\sum_{x=1}^{M}\sum_{y=1}^{N} f(x,y)\cos\left[\frac{(2x+1)u\pi}{2M}\right]\cos\left[\frac{(2y+1)v\pi}{2N}\right] \quad (5.4)$$

avec,

- u est la fréquence spatiale horizontale,

- v est la fréquence spatiale verticale,

- *f*(x, y) est la valeur de pixel aux coordonnées (x, y),

- *F*(*u*, *v*) est le coefficient de DCT au point de coordonnées (u, v), elle est dimensionnée de *M*×*N*, et α(•) est définis comme suit:

$$\alpha(w) = \begin{cases} \frac{1}{\sqrt{2}}, & w=1 \\ 1, & otherwise; \end{cases} \quad (5.5)$$

Cette matrice DCT(I) est une matrice des coefficients qui est définie à l'aide de fonctions cosinusoïdales. Ces fonctions constituent la base du domaine fréquentiel. La figure 5.5 présente ces fonctions de base à deux variables v, u = 0, 1, 2, …, 7.

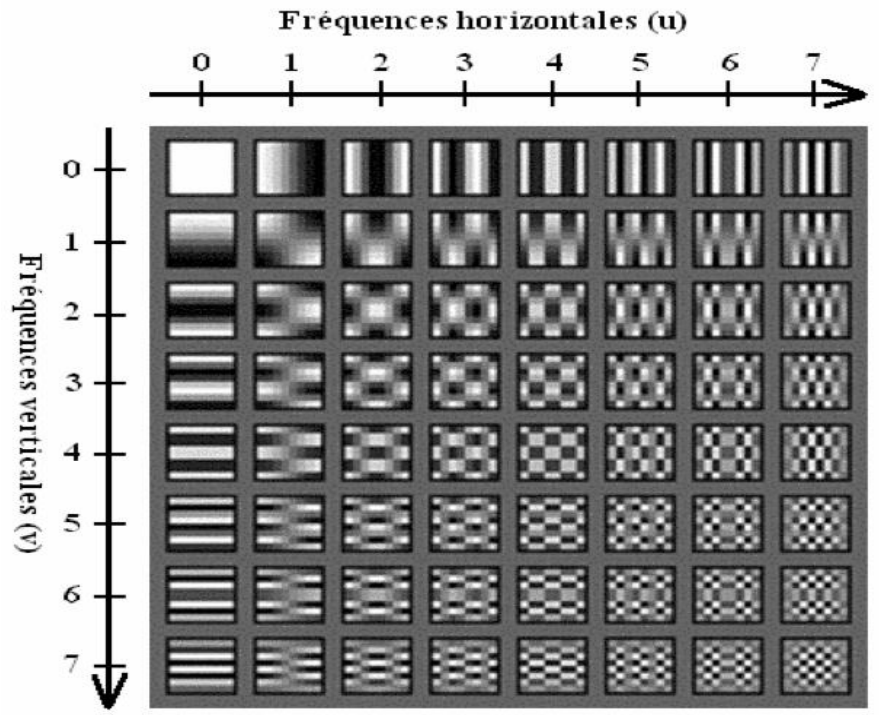

Figure 5.5 – Exemple de fonctions de base de DCT qui forme le domaine fréquentiel.

Afin d'obtenir un vecteur DCT qui est la transformée d'une sous-image I donnée, le parcours en zigzag est appliqué à la matrice DCT(I). La figure 5.6 montre l'ordre dans laquelle la matrice DCT(I) est parcourue selon le parcours en zigzag.

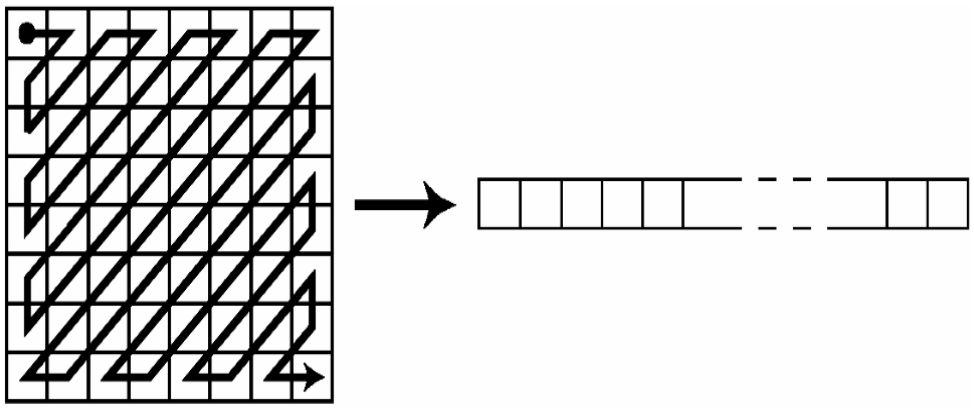

Figure 5.6 – Parcours en zigzag d'une matrice de dimension 8 × 8.

Les informations les plus importantes pour représenter l'image se trouvent dans les premiers coefficients. En utilisant uniquement les premiers coefficients et la transformation DCT inverse, il est possible de régénérer une image ayant presque le même rendu visuel que

la sous-image I d'origine. Généralement, la différence entre les deux images est totalement imperceptible. Afin de compresser les informations à l'aide de la DCT, une sous-image est présentée à l'aide des M premiers coefficients de vecteur DCT.

5.1.2 Traitement acoustique

Afin de pouvoir reconnaître le contenu d'un signal de parole correctement, il est nécessaire d'en extraire des paramètres caractéristiques et pertinents pour la reconnaissance. Le signal de parole n'est pas directement utilisable à cause de sa grande complexité (grande diversité d'information) et de son caractère redondant. Le but de la paramétrisation est d'extraire l'information pertinente pour la tâche proposée.

La première étape de la paramétrisation acoustique consiste à découper le signal de parole en fenêtres de taille fixe (variable de 20 ms à environ 40 ms) reparties de façon uniforme le long du signal (toutes les 10 ms). La taille des fenêtres est choisie en considérant que les propriétés du conduit vocal peuvent être considérées comme invariables sur une petite durée égale à la taille de la fenêtre. Le signal audio est donc considéré comme stationnaire sur la durée de la fenêtre. Pour ce faire, plusieurs techniques d'analyse du signal et d'extraction de paramètres peuvent être utilisées, mais dans le cadre de cette étude, seuls les paramètres acoustiques issus de systèmes de RAP de type énergie en sous-bande et de type RASTA-PLP seront utilisés. L'espace de représentation du signal de parole ainsi obtenu est muni d'une mesure de distance euclidienne adaptée à ces paramètres acoustiques. Cette mesure de distance est utilisée comme critère de similarité au sein de l'algorithme de comparaison du système de RAP considéré. L'extraction de paramètres acoustiques différents est un élément essentiel de cette thèse.

5.1.2.1 Analyse RASTA-PLP

Afin d'augmenter la robustesse des paramètres PLP, on peut envisager l'analyse spectrale relative RASTA (RelAtive SpecTrAl), présentée par (Hermansky and Morgan 1994) comme une façon de simuler l'insensibilité de l'appareil auditif humain aux stimuli à variation temporelle lente. Cette technique traite les composantes de parole non linguistiques, qui varient lentement dans le temps, dues au bruit convolutif (log-RASTA) et au bruit additif (J-RASTA). En pratique, RASTA effectue un filtrage passe-bande sur le spectre logarithmique ou sur le spectre compressé par une fonction non linéaire. L'idée principale est de supprimer les facteurs constants dans chaque composante du spectre à court-terme avant l'estimation du

modèle tout-pôle. L'analyse RASTA est souvent utilisée en combinaison avec les paramètres PLP (Hermansky and Morgan 1994). Les étapes d'une analyse RASTA-PLP sont décrites dans la figure 5.7. Les étapes grisées sont celles qui font la spécificité du traitement RASTA. La différence entre RASTA et J-RASTA se situe au niveau du logarithme (4ème étape) : ln(x) pour RASTA et ln*(1 + Jx)* pour J-RASTA.

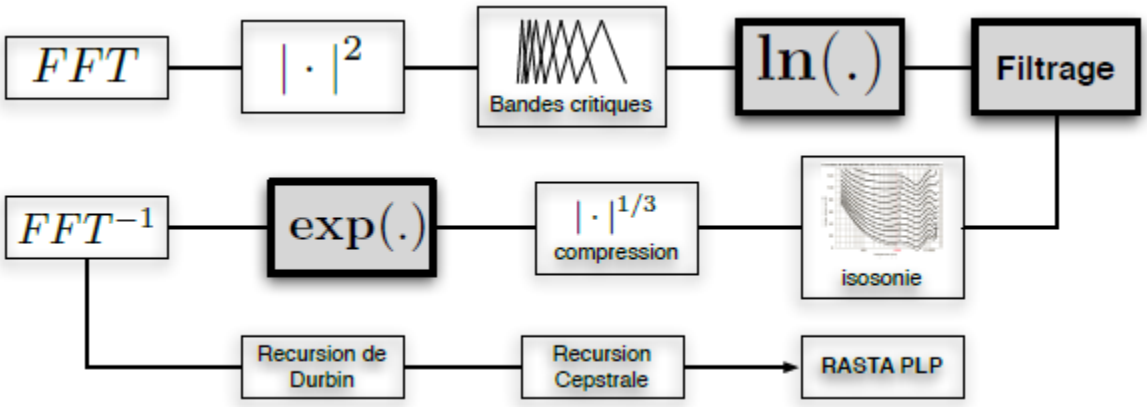

Figure 5.7 – Analyse RASTA PLP.

- **Utilisation des dérivées premières et secondes**

Dans les systèmes de reconnaissance actuels, il est très courant de compléter un jeu de paramètres par les dérivées premières (Δ) et secondes (ΔΔ) de ces paramètres. Les dérivées permettent d'inclure des caractéristiques dynamiques des paramètres acoustiques (vitesse et accélération). Le calcul des dérivées se fait sur des fenêtres centrées sur la trame analysée, ce qui assure la cohérence des informations présentes dans le vecteur. L'utilisation de ces Δ et ΔΔ est précisément un cas de concaténation de paramètres acoustiques. Une méthode de combinaison complète de modèles utilisant un jeu de paramètres (PLP), les Δ et les ΔΔ de ces paramètres est présentée dans (Misra et al. 2003). Chaque type de paramètres (statiques, Δ et ΔΔ) sont combinés de toutes les manières possibles pour former 7 jeux de paramètres acoustiques utilisés pour apprendre 7 modèles acoustiques différents, dont les probabilités sont ensuite combinées.

5.1.2.2 La quantification vectorielle

La quantification scalaire consiste à représenter une valeur d'un échantillon de signal pas forcement audio avec une précision réduite, par exemple la représenter avec une valeur

appartenant à un ensemble plus petit que l'ensemble original. C'est le cas typique de la conversion analogique/digitale.

Lorsque ce principe est appliqué par bloc d'échantillons (*vecteurs*), on peut parler de quantification vectorielle. La quantification vectorielle est alors une généralisation de la quantification scalaire. Mais, pendant que la quantification scalaire est dans sa forme la plus simple juste une conversion analogique/digitale, la quantification vectorielle est une méthode de codage/compression puissante. Elle est souvent utilisée dans les télécommunications pour le codage de la source, ou dans la compression des données notamment dans la compression des images. Elle est aussi un puissant outil de classification. La quantification vectorielle est définie par un doublet : un ensemble de vecteurs représentatifs appelés mots $C = c_1 c_2 \dots c_M$ qui forme un dictionnaire (*codebook* en anglais) et un critère de distorsion d (.,.) (Voir la figure 5.8).

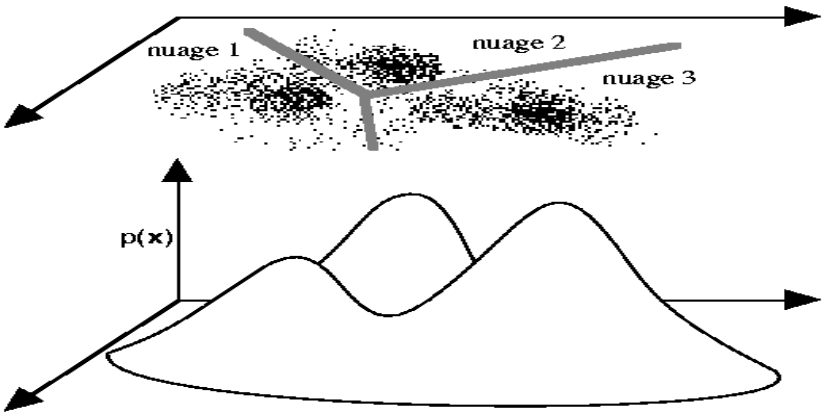

Figure 5.8 – Distribution de probabilités, un échantillon de points associés, et un découpage en nuages (clusters).

L'algorithme k-means est l'algorithme de clustering le plus connu et le plus utilisé, du fait de sa simplicité de mise en œuvre. Il partitionne les données d'un signal en K clusters. Contrairement à d'autres méthodes dites hiérarchiques, qui créent une structure en « arbre de clusters » pour décrire les groupements, k-means ne crée qu'un seul niveau de clusters. L'algorithme renvoie une partition des données, dans laquelle les objets à l'intérieur de chaque cluster sont aussi proches que possible les uns des autres et aussi loin que possible des objets des autres clusters. Chaque cluster de la partition est défini par ses objets et son centroïde. Le k-means est un algorithme itératif qui minimise la somme des distances entre chaque objet et le centroïde de son cluster. La position initiale des centroïdes conditionne le résultat final, de sorte que les centroïdes doivent être initialement placés le plus loin possible les uns des autres

de façon à optimiser l'algorithme. Cette méthode change les objets de cluster jusqu'à ce que la somme ne puisse plus diminuer. Le résultat est un ensemble de clusters compacts et clairement séparés, sous réserve qu'on ait choisi la bonne valeur K du nombre de clusters. Les principales étapes de l'algorithme k-means sont :

1. Choix aléatoire de la position initiale des K clusters.
2. (Ré-) Affecter les objets à un cluster suivant un critère de minimisation des distances (généralement selon une mesure de distance euclidienne).
3. Une fois tous les objets placés, recalculer les K centroïdes.
4. Réitérer les étapes 2 et 3 jusqu'à ce que plus aucune réaffectation ne soit faite.

5.1.3 Moteur de reconnaissance GA/HMM

Apprendre un HMM c'est ajuster les paramètres du modèle de manière à maximiser un certain critère. Différents critères sont disponibles dans la littérature. Nous n'allons pas tous les recenser, mais nous allons présenter une des plus importants et les plus couramment utilisés (Makhlouf et al. 2016).

Les GA constituent une large famille d'algorithmes statistiques, développés par (Holland 1983) et approfondis par (Goldberg 1999). Nous étudions ici comment une recherche de gradient « l'algorithme de Baum-Welch (BW) » peut être combinée avec des GA afin d'apprendre les HMM. Trois coopérations possibles entre les deux algorithmes sont étudiées, le GA peut être utilisé pour trouver un meilleur point de départ pour la recherche de gradient. Finalement, dans la méthode GA/HMM, le GA recherche automatiquement les trois probabilités, pour plus de détailles voir chapitre 4.

Pour mémoire, nous rappelons que l'opérateur de mutation consiste à modifier chaque coefficient de chaque modèle avec la probabilité P_m et que l'opérateur d'optimisation consiste à appliquer N_{BW} itérations de l'algorithme de Baum-Welch à chaque individu.

5.1.4 La fusion audiovisuelle

L'objectif d'un système de reconnaissance audio visuelle est de combiner au mieux les performances de deux systèmes audio et vidéo afin d'améliorer les performances de reconnaissance de la parole, en particulier en présence de bruit. Classiquement, on distingue deux types de fusion: la fusion des paramètres et la fusion des scores.

5.1.4.1 Fusion des paramètres

Cette fusion est réalisée au moment de la paramétrisation des signaux audio et vidéo. Une fois les paramètres de chaque modalité sont extraits, les vecteurs audio o^A et vidéo o^V de dimension d^A et d^V respectivement, sont concaténés à chaque instant t pour ne former qu'un seul vecteur de paramètres audio visuels de dimension $d^A + d^V$. Dans les étapes suivantes de la chaîne de reconnaissance de la parole (estimation des paramètres, décodage, évaluation), aucune modification n'est nécessaire.

5.1.4.2 Fusion des scores

La fusion de scores ou de décision est possible lorsque l'on dispose de systèmes séparés (ici, audio et vidéo) et que leur fusion est réalisée au moment de la décision, par combinaison de leurs scores respectifs. Des poids différents peuvent être affectes à chaque système (ou parties de ces derniers) afin de privilégier l'une ou l'autre des deux modalités. Dans le cas de système de reconnaissance ou les unités sub-lexicales (de type phone, par exemple) sont modélisées par des HMM et GA/HMM, cette fusion peut avoir lieu à différents niveaux qui sont l'état ou le phone ou le mot ou encore la phrase. Lorsque la fusion est effectuée à chaque état, elle est dite synchrone, sinon elle est asynchrone.

Plusieurs stratégies de fusion de décision ont été testés (produits, des sommes, minimum, maximum, vote ...) et tout montrer une amélioration significative des résultats par rapport à la considération d'une seule modalité, qui nous mener à se concentrer dans ce travail sur l'utilisation du le modèle de la fusion séparée, c.à.d. la fusion des scores provenant de chaque reconnaisseur GA/HMM. Leurs jeux de log-vraisemblance peuvent être combinés en utilisant les pondérations qui reflètent la fiabilité de chaque flux particulier, les scores combinés prennent alors la forme suivante (l'islam et Rahman 2010):

$$\log P(o^{AV}|\lambda) = w_A \log P(o^A|\lambda_A) + w_V \log P(o^V|\lambda_V) \tag{5.6}$$

Où λ_A et λ_V sont les GA/HMMs acoustique et visuels respectivement et $\log P(o^A|\lambda_A)$ et $\log P(o^V|\lambda_V)$ sont leurs log-vraisemblance. La fiabilité de chaque modalité peut être calculée par le plus approprié et le meilleur dans la performance (l'islam et Rahman 2010), la différence moyenne entre le log-vraisemblance maximum et les autres, peut être trouvé par :

$$R_s = \frac{1}{C-1}\sum_{i=1}^{C}\left(max \log P(o|\lambda^j) + \log P(o|\lambda^i)\right) \tag{5.7}$$

Où C est le nombre de classes étant considéré pour mesurer la fiabilité de chaque modalité et $s\epsilon\{A,V\}$. Après cela, nous pouvons calculer le poids d'intégration de la fiabilité audio A mesuré par:

$$w_A = \frac{R_A}{R_A + R_V} \tag{5.8}$$

Où R_A et R_V sont la mesure de fiabilité des sorties des GA/HMM acoustique et visuelle respectivement, et le facteur de pondération de la modalité visuelle peut être trouvée par la relation:

$$w_A + w_V = 1 \quad \text{for} \quad 0 < w_A, w_V < 1 \tag{5.9}$$

Le poids *W* permet de donner plus d'importance à une modalité ou à l'autre. Pour chaque système, *W* peut être choisi constant ou variable. Généralement, il dépend du rapport signal à bruit. Des travaux dans (Makhlouf et al. 2013a) montrent que les performances du système de reconnaissance audio visuelle sont meilleures pour un paramètre *W* dynamique.

5.2 Conclusion

Nous avons décrit, dans ce chapitre notre système proposé de reconnaissance de la parole audiovisuelle. Ainsi, nous avons abordé la fusion d'informations acoustiques et visuelles pour la RAP.

Nous nous intéressons dans le chapitre suivant à la description du système de reconnaissance audiovisuelle réalisé à base des HMM, et le modèle hybride GA/HMM. Egalement, la mise en œuvre de système qui a été appliqué sur deux corpus audiovisuels différents.