UNIVERSITÉ DE LORRAINE

Reconnaissance de la parole

Modélisation de la perception – M1 Informatique

1 Introduction

Contrairement à l'analyse d'image qui fournit une majorité d'informations en traitant l'image elle-même dans le domaine spatial, l'information la plus utile en traitement de la parole réside dans le domaine fréquentiel.

Lorsque des experts analysent un *spectre vocal*, ils observent un diagramme à trois dimensions appelé spectrogramme (figure 1) : les axes sont le temps et la fréquence, tandis que le niveau de gris d'un pixel de cette image vocale est d'autant plus foncé que la puissance est élevée. On y voit des bandes sombres qui identifient les zones de résonance (ou *formants*) et en suivant leur évolution dans le temps, un expert peut avec beaucoup d'expérience reconnaître la phrase prononcée.

La reconnaissance automatique de mots isolés s'inspire de cette représentation : le signal sonore est découpé en morceaux au travers de fenêtres recouvrantes dans lesquelles le spectre du signal est calculé. Un mot prononcé est ainsi caractérisé par un ensemble de vecteurs dits *vecteurs acoustiques*, représentant les spectres obtenus sur chacune des fenêtres. Un mot à reconnaître peut ainsi être comparé à un ensemble de mots de référence par mise en correspondance des séquences de vecteurs acoustiques. La programmation dynamique est utilisée pour permettre des déformations temporelles sur les mots prononcés.

La section 2 explique comment découper le signal en fenêtres recouvrantes et comment calculer le spectre du signal sur chacune de ces fenêtres. Les sections 3 et 4 décrivent les caractéristiques du signal vocal et quels sont les vecteurs acoustiques qui en découlent. La section 5 rappelle brièvement comment utiliser la programmation dynamique pour comparer deux suites de vecteurs acoustiques.

2 Découpage du signal en fenêtres recouvrantes

La parole est une combinaison de différents types de signaux que nous allons brièvement présenter ci-après. Mais déjà nous savons que le transformée de Fourier ne permet pas de distinguer ces signaux entre eux. Afin de remédier à la perte de cette donnée temporelle, la parole est analysée au travers d'une fenêtre temporelle étroite, appelée *trame*. Cette fenêtre se déplace temporellement avec un recouvrement d'une position à l'autre afin d'assurer le maintient de la corrélation entre les fenêtres successives.

Il est important que la fenêtre soit suffisamment large pour obtenir une bonne résolution fréquentielle, mais elle ne doit pas non plus être trop large si l'on souhaite distinguer les signaux. Il y a donc un compromis à atteindre qui a conduit à des fenêtres d'environ 30 msec décalées de 10 msec. Par exemple, pour un signal échantillonné à 6 KHz, cela correspond à des fenêtres de 480 échantillons décalées de 160 échantillons.

Une fenêtre rectangulaire possède une transformée en sinus cardinal dont les lobes secondaires provoquent des modifications importantes du spectre analysé (cf. phénomène de Gibbs vu en cours). Elle est donc généralement remplacée par une fenêtre de Hamming (cf. Fig. 2) :

$$w(n) = 0.54 - 0.46\cos(2\pi n/(N-1)) \forall 0 \leq n \leq (N-1)$$

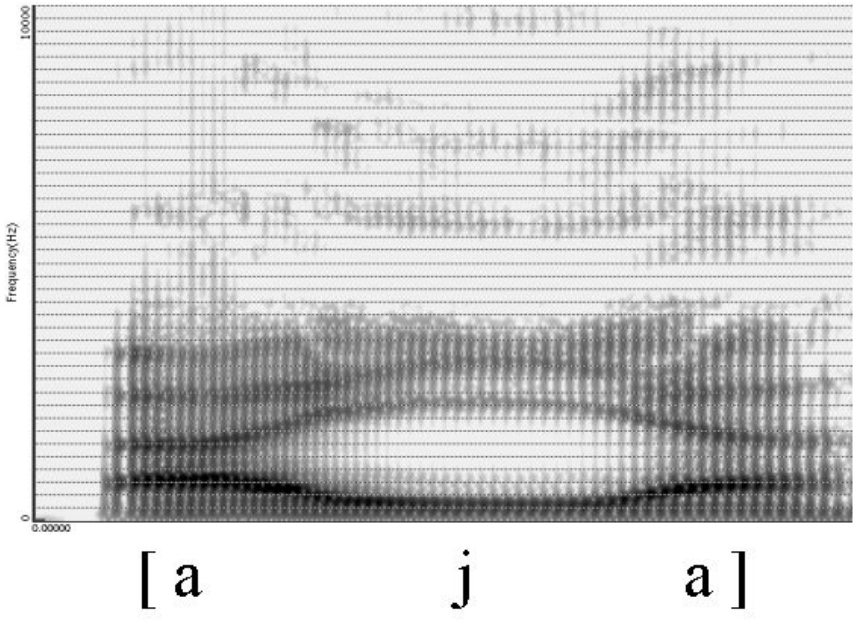


Figure 1: Exemple de spectrogramme

Les deux sections suivantes décrivent comment obtenir un vecteur acoustique dans chacune des fenêtres temporelles.

3 Caractéristiques de la parole

Il faut distinguer deux types de signaux vocaux :

les sons *voisés* : l'air propulsé par les poumons excite les cordes vocales comme l'archet fait vibrer les cordes d'un violon : la corde est accrochée par l'archet qui l'entraîne et se détache ensuite soudainement. L'excitation est alors considérée assimilable à un train impulsionnel périodique (peigne de Dirac) On parle de *fréquences glottiques* dont la fréquence fondamentale f_p oscille autour de $100Hz$ pour les hommes et $200Hz$ pour les femmes et les enfants. Cette fréquence est connue sous le nom de *pitch* dans la terminologie anglo-saxonne. Pendant l'élocution le pitch, qui détermine essentiellement la hauteur de la voix, varie légèrement et crée ainsi la mélodie de la parole comme par exemple l'élévation du ton en phrase interrogative ;

les sons *non voisés* : les cordes vocales ne participent pas à leur production. Il s'agit de phénomènes essentiellement buccaux ou labiaux. Ainsi, la prononciation d'un "P" se fait en comprimant l'air derrière les lèvres et en le relâchant brutalement. Pour le "S", on crée une turbulence d'air sur le bord des lèvres. Cette turbulence est assimilée à un bruit blanc (cf. Fig. 3)

Le signal d'excitation –train impulsionnel périodique pour un son voisé ou bruit blanc pour un son non voisé – est le signal d'entrée d'un tube acoustique : le conduit vocal dont la forme est modifiée au fil de notre élocution par les muscles faciaux, labiaux et la langue. Ce tube acoustique est modélisé par un filtre qui va déformer le signal d'excitation.

Dans le cas de nasales ("N", "M" ou "GN"), ce conduit se divise en deux car le conduit nasal participe également à la production vocale : l'air est expulsé simultanément par les narines et la bouche.

En conclusion, la parole est créée soit par un signal impulsionnel périodique soit par un bruit blanc (une turbulence) déformé par un conduit de forme variable dans le temps. La parole $s(t)$ est donc le

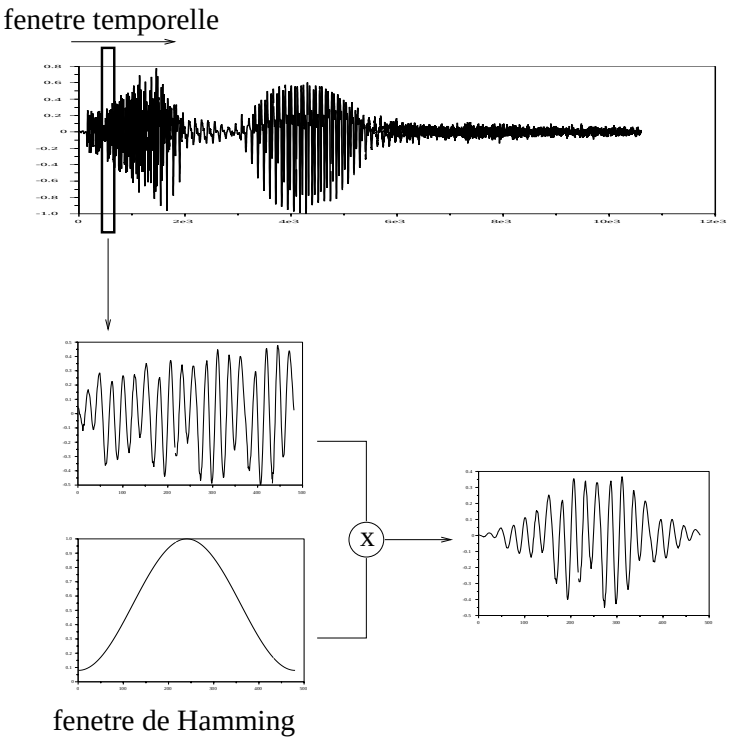
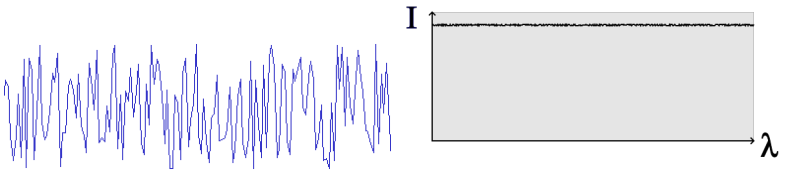


Figure 2: Découpage du signal en fenêtres recouvrantes.



(a) Échantillon de bruit blanc. (b) Spectre plat d'un bruit blanc.

Figure 3: Bruit blanc.

produit de convolution du signal d'excitation $e(t)$ par le filtre acoustique (du tube acoustique) $a(t)$:

$$s(t) = e(t) * a(t).$$

$e(t)$ est un peigne de Dirac de période $1/f_p$ (l'inverse du pitch) pour un signal voisé, ou un bruit blanc pour un signal non voisé.

D'après la transformée de Fourier d'un produit de convolution vue en cours, on peut écrire :

$$|S(f)| = |E(f)||A(f)|.$$

$E(f)$ est un peigne de Dirac de pas f_p (le pitch) dans le cas de sons voisés ou un bruit blanc (constant) dans le cas de sons non voisés.

Pour un son voisé, $|S(f)|$ est donc une structure fine, produit d'un spectre continu $|A(f)|$ par un train de raies fréquentielles $E(f)$ (figure 4). Pour un spectre de voix masculine, les raies sont très serrées (de l'ordre de 100Hz entre elles) et pour les voix féminines plus distantes (de l'ordre de 200Hz).

Ce qui caractérise une prononciation, c'est l'enveloppe de cette structure fine, qui dépend de la forme du conduit vocal, et plus particulièrement les maxima de cette enveloppe que l'on appelle les *formants* (figure 4). En reconnaissance de la parole, on s'efforce donc de déterminer l'enveloppe de $|S(f)|$.

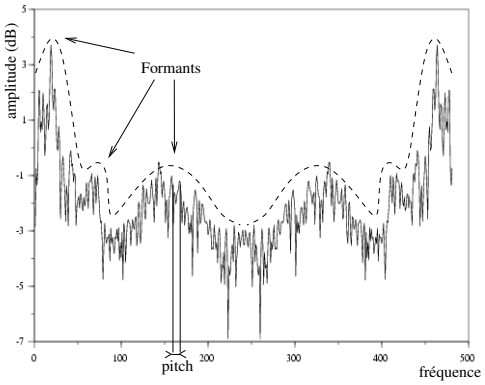


Figure 4: Spectre d'une zone voisée ($\log(|S(f)|)$.

4 Calcul des coefficients cepstraux

Pour déterminer l'enveloppe de $|S(f)|$, on commence par considérer les effets du signal d'excitation comme un terme additif du spectre du signal vocal : pour cela, on travaille sur le log du spectre (log-spectre) plutôt que sur le spectre lui-même :

$$\log|S(f)| = log|E(f)| + log|A(f)|.$$

L'élimination du terme additif $\log|E(f)|$ se fait par "lissage" du log-spectre : une manière simple de lisser une courbe $c(t)$ est de calculer sa transformée de Fourier $C(f)$, de ne conserver que les basses fréquences de $C(f)$ et de prendre comme courbe lisse la transformée de Fourier inverse de ces basses fréquences (cf. Fig. 5).

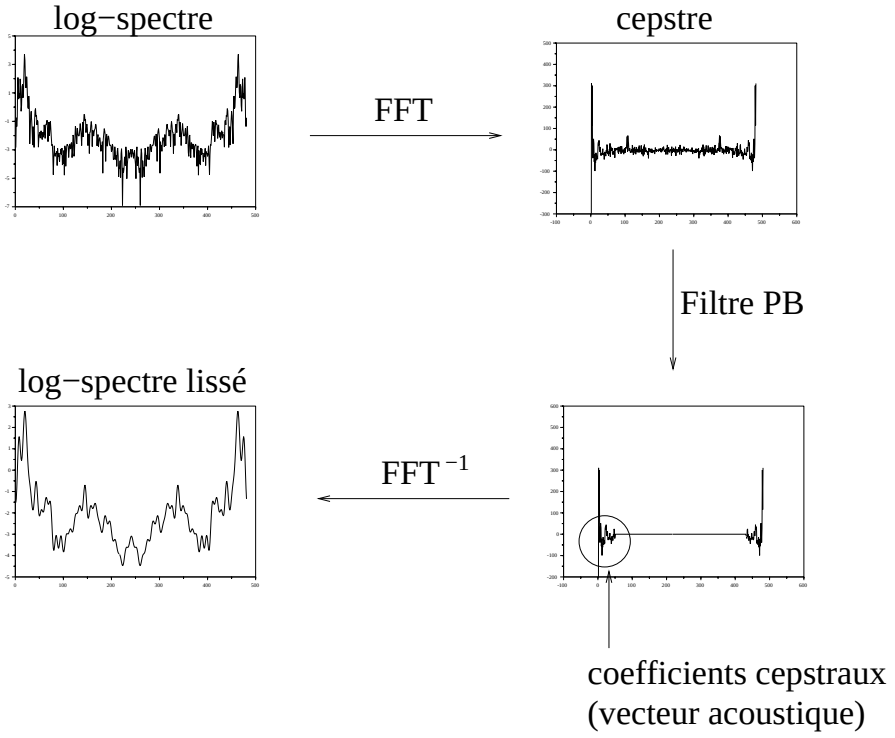


Figure 5: Calcul de l'enveloppe du log-spectre par passage au domaine *quéfrentiel*.

Ici, la fonction à lisser est un spectre exprimé dans le domaine fréquentiel. La transformée de Fourier de cette fonction est donc exprimée dans le domaine *inverse* du domaine fréquentiel, que l'on appelle par jeu de mots le domaine *quéfrentiel* (l'unité est la *quéfrence* !). De même, la transformée de Fourier du spectre est appelée *cepstre* ! En fait, plutôt que de représenter le signal par l'enveloppe du log-spectre, on le représente par les coefficients basses fréquences du cepstre. Ces deux représentations sont équivalentes, mais la deuxième est beaucoup plus intéressante en terme de compression des données représentatives. Ces coefficients sont appelés *coefficients cepstraux* et constituent le vecteur acoustique de la fenêtre considérée.

La figure 6 résume les étapes à suivre pour obtenir un vecteur acoustique sur une fenêtre du signal vocal.

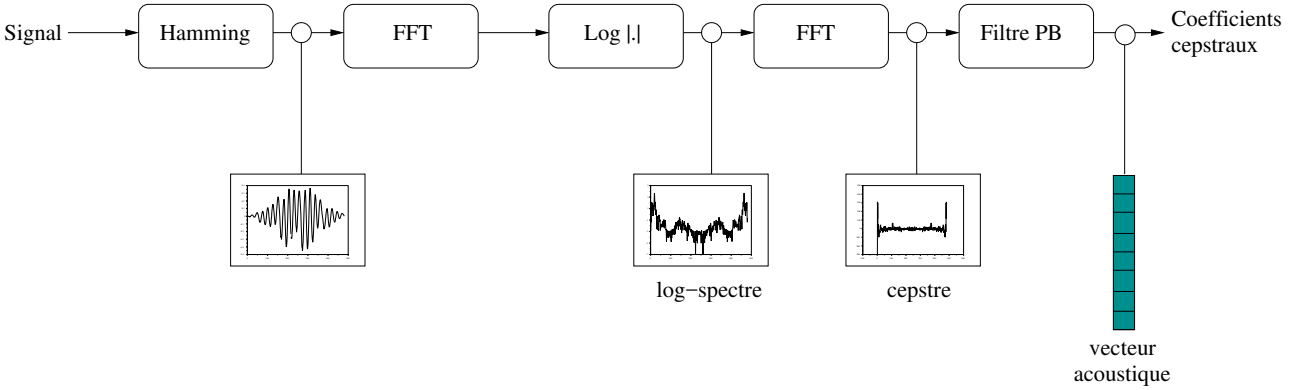


Figure 6: L'analyse cepstrale d'une fenêtre de signal.

5 Programmation dynamique

En appliquant l'analyse cepstrale présentée au paragraphe précédent, un mot prononcé est représenté par une succession de vecteurs acoustiques. L'objectif de la reconnaissance de la parole est de comparer un nouveau mot prononcé (mot test) avec une base de mots déjà identifiés (mots de référence). Pour cela, il faut être capable de comparer deux listes de vecteurs acoustiques deux à deux.

Une mesure de similitude entre deux vecteurs acoustiques doit d'abord être définie : pour des vecteurs acoustiques représentant les coefficients cepstraux du signal, la distance euclidienne entre ces deux vecteurs convient. Il s'agit ensuite de comparer deux à deux des *listes* de vecteurs. Le problème est alors un peu plus compliqué car il est rare que deux mots identiques prononcés à deux instants différents le soient à la même vitesse. Il faut donc utiliser un algorithme plus souple que celui qui consisterait simplement à calculer les distances entre les vecteurs de même position dans les deux listes à comparer : nous choisissons pour ce tp l'algorithme de programmation dynamique dû à Bellman. Les étapes à suivre pour obtenir une distance entre deux mots prononcés sont les suivantes :

1. On construit un tableau dont le nombre de colonnes est égal au nombre de vecteurs acoustiques du mot test et le nombre de lignes est égal au nombre de vecteurs acoustiques du mot de référence (figure 7). Dans chaque case du tableau, on inscrit la distance euclidienne $d(i, j)$ entre les vecteurs i et j correspondants ;

2. Dans le tableau de distances locales, il s'agit de chercher un chemin partant des premiers vecteurs de la référence et du test que l'on fait se correspondre et rejoignant leurs derniers vecteurs qui doivent également être associés (figure 7). Selon le principe de la programmation dynamique, on cherche à calculer $D(m, n)$ où m est le nombre de vecteurs acoustiques du mot test, n le nombre de vecteurs du mot de référence, et $D(i, j)$ la distance cumulée (j est l'indice courant des lignes de la colonne i) donnée par :

$$D(i+1, j) = d(i+1, j) + \min_{k \in \mathcal{P}(i+1,j)} D(i, k),$$

où $\mathcal{P}(i+1,j)$ est l'ensemble des prédécesseurs autorisés de la case $(i+1,j)$. On pourra prendre comme prédécesseurs autorisés ceux qui sont indiqués en figure 7 ;

3. On normalise la distance $D(m,n)$ en la divisant par le nombre de distances locales qui la composent.

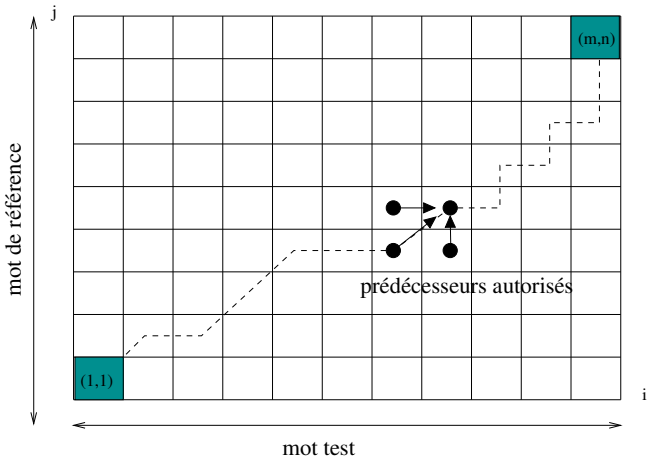


Figure 7: Programmation dynamique.

6 Enoncé du TP

L'objectif du TP est d'implémenter l'algorithme de reconnaissance de mots isolés qui vient d'être décrit, puis de l'appliquer sur une base de 6 mots de référence qui sont des commandes de mouvement 3D. L'implémentation se fera sous MatLab, cet outil étant particulièrement adapté à ce type de problème, comme nous l'avons illustré en cours et en tp (en particulier, la lecture d'un fichier wave et la fonction FFT sont déjà implémentés).

Les sons de référence sont disponibles sur Arche. Il s'agit des fichiers se terminant par 1.wav. Les noms de ces fichiers correspondent aux commandes de mouvement qu'ils contiennent :

- agauche1.wav : *"à gauche"*
- adroite1.wav : *"à droite"*
- avance1.wav : *"avance"*
- recule1.wav : *"recule"*
- enhaut1.wav : *"en-haut"*
- enbas1.wav : *"en-bas"*

Deux autres séries des mêmes mots sont aussi disponibles sur le site pour vous permettre de tester l'algorithme : les fichiers se terminant par 2.wav correspondent aux mots prononcés par le même locuteur, mais à des instants différents ; les fichiers se terminant par 3.wav correspondent aux mots prononcés par un locuteur différent.

Ces deux autres séries de mots vous permettront d'évaluer votre algorithme, mais celui-ci sera aussi testé sur d'autres séries, non communiquées... *Bien-sûr, l'objectif n'est pas d'atteindre un taux de* 100% *de réussite, surtout pour des séries prononcées par un locuteur différent.* Cependant, les valeurs de certains paramètres (comme le nombre de coefficients cepstraux à conserver) sont laissées à votre appréciation, et leur réglage permettra d'obtenir un algorithme plus ou moins pertinent.

Travail à rendre

Le tp est à réaliser en **binôme**, et à rendre pour le **dimanche 11 novembre 2016** au plus tard. À cette date, vous devrez avoir déposer l'ensemble de votre projet sur le dépôt du cours sur Arche. Le projet doit être un dossier compressé ayant pour identifiant vos noms et prénoms.

L'un des fichiers, d'identifiant `compare.m`, devra contenir la fonction

$$\texttt{function}\,[\texttt{score}] = \texttt{compare}(\texttt{motTeste}, \texttt{motBase})$$

qui sera utilisée pour tester votre algorithme. Cette fonction prend en argument deux vecteurs lignes représentant respectivement le mot à reconnaître et un mot de la base, et renvoie la distance entre les deux mots.

Attention : des tests automatiques seront effectués, la note sera donc arbitraire si ce profil n'est pas respecté. En temps utile, vous trouverez sur Arche le fichier *tests1.m* qui contient les appels à votre fonction `compareBase()` tels qu'ils seront utilisés pour la notation et qui calcule la partie de votre note (sur 20) correspondant aux jeux d'essais fournis.

Remarques :

- le nombre d'imbrications autorisé dans les processus récursifs est assez limité sous MatLab : il est plus prudent d'utiliser des algorithme non récursifs pour ce tp ;
- comme le log-spectre est une fonction paire réelle, le cepstre est théoriquement réel (cf. cours). Cependant, les approximations numériques inévitables sur des machines qui sont nécessairement à précision limitée, font que nous n'obtenons pas des parties imaginaires exactement nulles, mais très proches de zéro. Vous devrez donc prendre la partie réelle du cepstre comme coefficients cepstraux ;
- la fonction MatLab `audioread` permet de lire un fichier wave et de récupérer les valeurs lues dans un vecteur ligne ;
- **protégez votre répertoire de travail en lecture**. Toute récupération de code en provenance d'un autre binôme sera sanctionnée par une note nulle **pour les deux binômes**.