UNIVERSITÉ LAVAL

Département de génie électrique et de génie informatique
Faculté des sciences et de génie
Université Laval
1065, avenue de la Médecine
Québec (Québec) G1V 0A6
Téléphone : +1 (418) 656-2984
Télécopieur : +1 (418) 656-3159
Courrier électronique : gel@gel.ulaval.ca

Vocal Activity Algorithm
Module commande de la parole

Development and integration of a voice command for users with speech disabilities (dystrophy, paralysis, etc.) on the JACO robotic arm

Sous la tutelle de Benoit Gosselin

UNIVERSITÉ LAVAL

Département de génie électrique et de génie informatiques
Faculté des sciences et de génie
Université Laval

Table des matières

UNIVERSITÉ LAVAL

Département de génie électrique et de génie informatiques
Faculté des sciences et de génie
Université Laval

Résumé

L'objectif de ce rapport est de confronter les performances de mon algorithme de Détection d'Activité de la Voix dans une trame audio (DAV ou Vocal Activity Detection en anglais). Le DAV est un algorithme qui vise à extraite d'une trame audio tous les tronçons où il y a présence de voix et cherche à accélérer le processus de reconnaissance de la parole. En effet, en ne se concentrant que sur les tronçons de voix, on ne cherche pas à reconnaitre des silences entre des mots, ... mais que des mots, phonèmes, ... (cela dépend de la manière dont est réalisé l'algorithme de reconnaissance de la parole).

Actuellement, plusieurs méthodes sont utilisées pour réaliser cette opération de détection de flux de parole dans différents domaines (temporels et fréquentiels). Le domaine utilisé va bien sûr impacter sur la rapidité de l'algorithme. Le domaine temporel ne nécessitant pas de calcul complexe, il rendra notre algorithme plus rapide néanmoins il sera plus sensible au bruit. Le domaine fréquentiel est quant à lui moins sensible au bruit mais est plus gourmand en calcul. Les méthodes les plus connues dans le domaine temporel sont l'énergie à court terme combinée aux taux de passage à zéro (Short time Energy and Zero Crossing Rate) et dans le domaine fréquentiel : l'Entropie (Spectral Entropy), le Spectral Centroïd (qui est une sorte d'indicateur du "centre de masse" du spectre) mais aussi les Mel Feature Cepstrum Coefficents (MFCC).

Mon partis pris a été de choisir de travailler dans le domaine fréquentiel essentiellement afin d'avoir un algorithme très robuste au bruit et de combiné l'Entropy au MFCC. De plus, même si mes deux méthodes sont très peu sensible au bruit, elles perdent un peu de leur immunité à partir d'un certain SNR (Signal Noise Ratio ou Rapport de Signal à Bruit en français). De fait, afin de pouvoir travailler dans des environnements assujettis à des SNR très faible, j'ai donc généré une fonction d'initialisation qui récupère l'empreinte du bruit environnant et s'en sers de vecteurs modèles. Les vecteurs extraits des deux méthodes sont donc comparés à ces modèles de bruit par distance euclidienne ou par corrélation pour chaque trame. Les vecteurs modèles sont mis à jour à chaque trame pour suivre l'évolution des éléments en entrées.

L'étude comparative de mon modèle se fera par confrontation à quatre modèles et avec des contraintes de bruits.

1 Introduction

Ce rapport s'organise de la façon suivante. Tout d'abord quelques notions théoriques nécessaires à la bonne compréhension du sujet seront introduites. Un état de l'art non exhaustif de la détection de l'activité vocale sera présenté. Ensuite, les différents travaux réalisés seront décrits. Ces travaux reposent essentiellement sur les Mel Feature Cepstrum Coefficient (MFCC) et l'Entropie, dans l'objectif de rejeter toutes les trames audio non utiles. Les performances du VAD des méthodes proposées seront exposées puis comparées à des méthodes de références. Ce sera alors l'occasion de soulever des difficultés liées aux différentes méthodes mais aussi des améliorations qui pourraient y être éventuellement apportés.

2 Notions théoriques

2.1 Traitement à court-terme du signal vocal

On peut assez facilement constater que la forme d'onde d'un signal vocal met en évidence son caractère non stationnaire. Étant donné son caractère non stationnaire, un étude à long terme ou globale est généralement inefficaces. L'hypothèse la plus utilisée dans le traitement de la parole est le fait que les propriétés du signal vocal change lentement dans le temps [4]. Cette hypothèse conduit à un traitement à court terme. De plus, l'analyse à court terme permet de respecter le caractère temps réel que nous souhaitons donner à notre algorithme par la suite.

2.2 Méthodes dans le domaine temporel

Les méthodes ci-dessous sont une liste non exhaustive des méthodes temporelles utilisées. Elles sont néanmoins celle le plus couramment utilisées. Les définitions de chaque méthode est extraite de l'article suivant [6]

2.2.1 Énergie à court terme

L'énergie court-terme est un paramètre qui reflète les variations d'amplitude dans le signal vocal. Elle fut un de premiers paramètres utilisés dans la détection d'activité vocale. Elle permet de classifier les trames voisées et non voisées. Elle est définit de la manière suivante :

$$E_n = \sum_{k=n-N+1}^{n} [x(k).w(n-k)]^2 \tag{1}$$

où $w(n-k)$ est la fenêtre d'analyse, n est l'échantillon sur lequel est centré la fenêtre d'analyse et N la taille de la fenêtre d'analyse.

2.2.2 Taux de passage par zéro (Zero Crossing Rate, ZRC)

Le ZRC compte le nombre de passage par zéro du signal. Les trames de voisées ont un taux de passage à zéro faible alors l'inverse des trames non voisées qui ont un ZRC élevé. On peut définir le ZRC de la manière suivante :

$$Z_n = \sum_{m=-\infty}^{infty} |sgn[x(m)] - sgn[x(m-1)]lw(n-m) \tag{2}$$

où

$$sgn[x(n)] = \begin{cases} 1 & x(n) \geq 0 \\ -1 & x(n) < 0 \end{cases} \tag{3}$$

et

$$w(n) = \begin{cases} 1/(2N) & 0 \leq n \leq N-1 \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

2.3 Domaine fréquentiel

2.3.1 Entropie Spectrale

Nous ne pouvons pas parler d'Entropie spectrale sans parler auparavant de l'entropie introduite par Claude Shannon dans sa théorie de l'information. Cette dernière permet de mesurer la quantité d'information contenue dans un signal aléatoire. Elle est définit de la manière suivante :

$$H(x) = \sum_k p(x_k).log_2[P(x_k)] \tag{5}$$

où $x = x_{k0 \leq k \leq N-1}$ est une série temporelle, fréquentielle ou autre et où $P(k)$ est la probabilité d'un certain état x_k.

Travaux de Shen J, Hung J et Lee J [7] Shen J., Hung J. et Lee J. ont été les premiers a utilisé l'entropie dans le cadre de la détection de la parole. Ils ont démontré que l'entropie d'un flux de parole diffère de celui d'un flux sans parole. En effet, leur étude a montré que la structure de la voix reflète d'une organisation que l'on ne retrouve pas dans la structure spectrale du silence.

L'entropie spectrale L'entropie de Shannon mesurant la quantité d'information contenue dans un signal aléatoire, il est plus d'usage d'utiliser l'entropie spectrale : la structure harmonique d'un segment de voix n'apparaissant que dans son spectrogramme. L'entropie spectrale est définit tout d'abord par la transformée de Fourier à court terme (Short time Fourier Transform) :

$$S(k,l) = \sum_{n=1}^{N} h(n)s(n-l) \exp \frac{-j2\pi kn}{N} \text{ avec } 0 \leq k \leq K-1 \text{et } K = N \tag{6}$$

$S(k,l)$ représente l'amplitude de la k^{ime} composante fréquentielle, pour la l^{ime} trame d'analyse. $s(n)$ est l'amplitude du signal au temps n. N, le nombre de points considérés pour la transformée de Fourier. $h(n)$ est la fenêtre d'analyse (généralement, une fenêtre de hamming).

Energie spectrale On définit l'énergie spectrale de la manière suivante :

$$S_{energy}(k,l) = |S(k,l)|^2 \text{ avec } 1 \leq k \leq \frac{N}{2} \tag{7}$$

P(k,l) La probabilité associée à chaque composante spectrale est obtenue en normalisant :

$$P(k,l) = \frac{S_{energy}(k,l)}{\sum_{i=1}^{\frac{N}{2}} S_{energy}(i,l)} \text{ avec } 1 \leq i \leq \frac{N}{2} \text{ et} \sum_i P(i,l) = 1 \text{ pour tout } l. \tag{8}$$

Entropie Spectrale L'entropie spectrale s'appuie sur l'entropie de Shannon :

$$H(l) = \sum_{i=1}^{\frac{N}{2}} P(i,l). \log_2 |P(i,|)| \tag{9}$$

Il est plus souvent d'usage de considérer l'opposé de l'entropie pour obtenir des profils analogues à ceux de l'intensité.

Limitation Une des limitations de l'entropie spectrale exposé par Ouzounov [3] est que l'entropie d'une trame sans parole bruitée par un bruit blanc coloré peut-être équivalente à celle d'une trame avec parole mais bruitée.

2.3.2 Spectral Centroïd

Le "centroïde spectral" C_i de la i^{ime} trame est défini comme le centre de gravité de son spectre.

$$C_i = \frac{\sum_{k=1}^{N}(k+1)X_i(k)}{\sum_{k=1}^{N}X_i(k)}. \tag{10}$$

$X_i(k)$ avec $k = 1, ..., N$ sont les coefficients de la i^{ime} courte trame de la transformée de Fourier discrète (DFT). N est la longueur de la trame. Le centroïde spectral est une mesure de la position spectrale où des valeurs élevées correspondent à des sons plus brillant.

Ces explications sont issues de l'article de Theodoros Giannakopoulos [5]. Son algorithme et son étude seront utilisés par la suite pour comparer les résultats de notre algorithme.

2.3.3 Mel Feature Cepstral Coefficients (MFCC)

Les explications qui suivent sont issues et pour la plupart recopier du site *practicalcryptography.com* [1].

Les Mel Features Cepstral Coefficients (MFCCs) sont des caractéristiques majoritairement utilisées en reconnaissance de parole automatique. Les MFCCS ont été introduites par Davis and Mermelstein dans les années 80, et ont été "l'état de l'art" depuis. Avant la mise en place des MFCC, Les coefficients de prédiction linéaire (LPC) et les coefficients cepstaux de prédilection linéaire (LPCC) constituaient la caractéristique principale de reconnaissance automatique de la parole (ASR), en particulier avec les classificateurs HMM .

Etape pour déterminer les MFCCs Pour déterminer les MFCCs, ll faut commencer, comme tout au long de notre étude, par sectionner le signal en courte trame se recouvrant les unes aux autres (overlapping). Pour chaque trame, on calcule l'estimation du périodogramme du spectre de puissance (en d'autres termes, on regarde la distribution de notre périodogramme). Ensuite, on applique un banc de filtres Mel aux spectres de puissances, somme de l'énergie dans chaque filtre. On prend le logarithme de toutes les énergies du banc de filtre. Puis la transformée en cosinus inverse (DCT). On ne garde que les 12 premiers coefficients en omettant le premier.

3 Travaux réalisés

Cette partie va aborder du travail réalisé sur l'algorithme de détection vocale (VAD). Une grande partie est inspirée de méthodes et modèles existants. En ce qui concerne le modèle de bruit de fond avec mise à jour à chaque trame, il est extrait des travaux Hongzhi Wang et Yuchao Xu, Meijing Li [8]. J'ai donc transposé leur travaux à l'entropie afin de la rendre plus robuste au bruit. Nous avions parler d'une des limitations de l'entropie un peu plus que nous avons cherché à combler par cette méthode.

3.1 Fonctionnement général de l'algorithme

Avant de rentrer des explications plus approfondis sur les méthodes utilisées ainsi que leurs outils, nous allons expliquer globalement le fonctionnement de notre algorithme.

Tout d'abord, une première fonction *recorder.m* [2] permet l'acquisition de donnée par la carte son de mon laptop et de sauvegarder des données dans des veccteurs. Deux acquisition sont réalisées :

- Acquisition du bruit environnant
- Acquisition d'une trame de voix durant 3s (le temps est ici arbitraire. D'autant plus, que par la suite nous chercherons à être en temps réel).

Une fois ces deux acquisitions opérées, nous utilisons celle de bruit de fond afin de générer un vecteur $MFCCs_{noise}$. Pour cela, nous "découpons" le signal acquis en plusieurs trames de 15ms avec un recouvrement de 10ms afin de respecter l'hypothèse du caractère stationnaire exprimé un peu plus haut. Ces trames (ou fenêtres) sont ensuite envoyées à un algorithme : $short_time_Fourier_transform.m$ pour réaliser une transformée de Fourier (l'appellation de la fonction est un peu faussée car nous sommes déjà dans un cas de court terme). Enfin, on extrait les Mel Feature Cepstral Coefficients du signal de bruit de fond (toutes les trames) que nous moyennons. Ce vecteur est donc la moyenne des coefficients cepstraux du bruit (en considérant que seule du bruit de fond est présent dans cette acquisition). On opère similairement pour l'entropie, en calculant la valeur moyenne de l'entropie pour le signal de bruit de fond (on réalise la moyenne sur l'ensemble des trames).

Pour l'acquisition d'une trame de voix, on réalise aussi le "découpage" en trame avec chevauchement. Pour chaque trame, on calcule l'entropie et les coefficients cepstaux. On compare le vecteur de coefficients cepstraux et la valeur de l'entropie de chaque trame à respectivement, le vecteur de bruit de fond de la moyenne des coefficients cepstraux et à la valeur moyenne du bruit de fond de l'entropie. On utilise pour cela, respectivement, la corrélation et la distance euclidienne.

On applique ensuite des seuils pour chacune des deux variables évoluant au cours du temps afin d'être le plus pertinent possible. Chaque trame qui est au dessus du seuil est labellisée d'un "un" sinon elle l'est d'un "zéro". Le vecteur contenant les labels est ensuite envoyé à une fonction permettant de corriger les valeurs aberrantes (par exemple un "un" perdu au milieu de "zéro" et vis-versa. L'algorithme corrige jusqu'à plusieurs paquets de valeurs aberrantes en fonction des paramètres qu'on lui rentre en argument). Pour cela, on s'appuie du poids des labels au nombres paires entourant chaque label considéré.

Une fois l'opération de correction de label réalisée, nous n'avons plus qu'à extraire les segments (en ms et non en trame comme nous avions précédemment) en considérant nos labels de "un" comme de la

voix et les autres comme du silence, du bruit et des parasites sur le signal. Enfin, un dernier algorithme tri les segments récupérés. En effet, dans la littérature, chaque phonème dure un certain temps (selon la langue, le plus petit dure un $20ms$ et le plus long, souvent voisé, dure $100ms$. Dans notre cas, on suppose que notre algorithme reconnait un ensemble de phonème, soit une syllabe. Dans ce cas, on considère donc qu'en dessous de $80ms$, on rejette le segment considéré. LITTERATURE -REF ?

Finalité Nous obtenons donc de l'acquisition d'un signal de trois secondes (choix arbitraires), un ensemble de segments contenant de la parole dans la majeur partie des cas (en effet, des bruits peuvent-être présents). Nous conclurons sur les manières d'améliorer notre algorithme. Néanmoins, cela ralentira notre algorithme sachant qu'il est déjà plus lent que ceux utilisés dans la littérature (on travaille avec des transformées de Fourier sur $257pts$ pour chaque trame de $15ms$. La complexité des calculs est donc élevée). Nous allons développer par la suite les méthodes utilisées pour les différentes phases de l'algorithme de Détection d'activité vocale (on rappelle qu'il est composé d'un ensemble de sous-algorithmes).

3.2 L'idée d'utiliser la distance entre des vecteurs de bruit de fond et celui de chaque trame ...

L'article de Hongzhi Wang et Yuchao Xu, Meijing Li propose d'extraire les Mel Features Cepstral Coefficients (MFCCs) d'un signal vocal pour chaque trame en considérant les dix (10) premières trames comme des trames de bruit d'arrière plan. En réalisant la moyenne de ces vecteurs, on obtient un vecteur de $MFCC_{bruit}$. Ce vecteur va permettre de réaliser la Corrélation des MFCC de chaque nouvelle trame avec lui même (dans l'article, ils utilisent aussi la distance euclidienne, qui se révèle être moins performante).Le vecteur de $MFCC_{bruit}$ est mis à jour à chaque trame :

$$\underline{cno} = \underline{c}.p + (1-p).\underline{c} \tag{11}$$

où $\underline{cno}$ est le vecteur de $MFCC_{bruit}$, c sont les coefficients de chaque nouvelle trame et p est proche de 1. Le fait que p doit-être proche de 1 permet de garantir que le vecteur de MFCC bruité reste constitué de bruit de fond principalement malgré l'introduction de coefficients de parole à chaque nouvelle trame.

Nous nous sommes donc appuyé de ces travaux pour construire notre algorithme de détection vocale. Nous avons aussi appliqué le principe de corrélation à l'entropie spectrale, toujours dans le soucis de rendre notre algorithme le plus robuste au bruit.

3.3 Détermination des seuils

La détermination des seuils a été la partie la plus difficile à mettre en oeuvre, non pas que les algorithmes sont complexes, mais qu'une mauvaise méthode entraine la suppression des trames voulues. Ma première idée fut d'utiliser la distribution des valeurs obtenues. Cette distribution étant obtenue après avoir supprimé toutes les valeurs au-dessus de la moyenne du signal. Le but étant de récupérer la plus petites variations possibles du signal après analyse (Entropie et/ou MFCCs). Néanmoins, cette méthode bien qu'efficace était constante au court du temps ou du moins sur l'analyse d'un buffer d'échantillon. Il y avait donc des pertes et l'introduction de bruit après seuillage. La deuxième idée fut de trouver une méthode adaptée au chacun des paramètres utilisés.

3.3.1 Seuil pour les MFCCs

Concernant les MFCCs, je me suis appuyé d'une fonction sigmoïde mise à jour à chaque fenêtre. La sigmoïde a été améliorée afin d'avoir pour $x = 0$, une valeur proche de la valeur maximale du bruit (contrairement à la sigmoïde habituelle qui à pour valeur en $x = 0$, 0,5). Cette valeur maximale du bruit est généralement proche de zéro de part la méthode utilisée pour déterminer la distance entre des trames de bruit de fond et la trame en cours (la distance d'un vecteur de bruit de fond à un autre vecteur de bruit donne généralement un résultat faible).

3.3.2 Seuil pour l'entropie spectrale

La fonction de seuillage concernant l'entropie spectrale est quant à elle très standard du fait que les trames de parole sont très différentes de celle où il y a des silences ou du bruit. Pour cela, nous utilisons les 20 premières trames comme référence pour réaliser la valeur du seuil à l'instant 0. On considère la moyenne de ces 20 trames et on y ajoutant trois (3) fois l'écart type de ces 20 trames :

$$th_{noise} = \frac{1}{N}\sum_{k=1}^{N} x_k + 3.\sqrt{\frac{1}{N-1}\sum_{i=1}^{N}|x_i - \mu|} \tag{12}$$

où μ représente la moyenne du signal considéré. En effet, on considère que sur les 20 premières trames ne représentent que bruit. En y ajoutant trois fois l'écart-type, on s'assure d'être au-dessus du bruit de fond dans sa majeur partie.
Pour avoir un seuil pertinent et précis, on réalise une mise à jour ce seuil. La mise à jour est effective dès que l'on a 10 trames à traiter. En effet, nous n'avons pas de pertinence à utiliser la moyenne et l'écart type sur une trame. Dix trames nous semblaient convenable pour pouvoir exploiter ces deux indices pour une mise à jour correcte de notre seuil. Les étapes de la mise à jour sont les suivantes :

- On collecte dix nouvelles trames
- On réalise l'opération suivante :

$$th_{new} = \frac{1}{N}\sum_{k=1}^{N} x_k + 3.\sqrt{\frac{1}{N-1}\sum_{i=1}^{N}|x_i - \mu|} \tag{13}$$

- On met à jour la valeur du seuil pour les dix prochaines trames de la manière suivante (la mise à jour est similaire à celle du vecteur bruit pour l'entropie et les MFCCs) :

$$th = p.th_{noise} + (1-p)th_{new} \tag{14}$$

  Comme durant la mise à jour du vecteur bruit, on cherche à avoir une valeur de p proche de 1 (un) afin que la mise à jour ne soit pas polluée par une trame de voix. Cela rendrait nos seuil inutile car il ne considérerait plus les trames de voix ou alors celle avec des intensités plus élevées que celle considérée.

Amélioration possible du seuil On pourrait, tout comme le seuil des MFCCs, considérer une fonction sigmoïde particulière pour calculer le seuil optimal à chaque trame. Pour le moment, celle-là n'a pas été développée car les résultats étaient satisfaisants.

Références

[1] Mel frequency cepstral coefficient (mfcc) tutorial. `http://practicalcryptography.com/miscellaneous/machine-learning/guide-mel-frequency-cepstral-coefficients-mfccs/`. [Online ; Accessed : 2017-03-15].

[2] Record and play audio. `https://www.mathworks.com/help/matlab/import_export/record-and-play-audio.html`. [Online ; Accessed : 2017-02-04].

[3] Ouzounov .A. Robust features for speech detection a comparative study. *In Int. Conference on Computer System and Technology*, 2005.

[4] Patrick Durand and René Durand. Les tomates tueuses. *Le beau journal*, page 24, jan 2007.

[5] Theodoros Giannakopoulos. A method for silence removal and segmentation of speech signals, implemented in matlab. 2009.

[6] Faran Awais Butt Madiha Jalil and Ahmed Malik. Short-time energy, magnitude, zero crossing rate and autocorrelation measurement for discriminating voiced and unvoiced segments of speech signals. *978-1-4673-5613-8©2013 IEEE*, Mai 2013.

[7] Hung J. Shen J and Lee J. Robust entropy-based endpoint detection for speech recognition in noisy environments. *In Fifth International conference on spoken Language Processing*, 1998.

[8] Hongzhi Wang and Meijing Li Yuchao Xu. Study on the mfcc similarity-based voice activity detection algorithm. *Artificial Intelligence, Management Science and Electronic Commerce (AIMSEC), 2011 2nd International Conference on*, 2011.