Un pitch-tracker basé sur l'autocorrélation

version 1.00

Stéphane Rossignol

20 janvier 2016

Table des matières

Chapitre 1

Introduction

Le livre de Wolgang Hess, [Hes83], est la référence, en terme de pitch-tracking. Un pitch-tracker est présenté ici, basé sur l'autocorrélation. Il est écrit en octave/matlab. Une mesure de voisement/non-voisement est elle aussi récupérée.

Chapitre 2

La méthode

Le but est de proposer l'algorithme le plus simple possible. Pour les signaux de parole, il est nécessaire d'extraire aussi le voisement. Les deux étapes de post-traitement auraient pu être omises. Le code est court et on peut le mettre aisément en concordance, à sa lecture, avec les paragraphes qui suivent.

Le système est basé sur l'autocorrélation (notamment sur le fait que l'autocorrélation a la même période que le signal original et que l'amplitude de ses maximums décroissent linéraiement avec τ). Les pitchs possibles sont compris entre 100 Hz et 600 Hz. La taille des trames est de 40 ms. Le taux de décalage entre 2 trames successives est de 10 ms. Le programme a été optimisé pour des signaux échantillonnés à 8 kHz, alors les signaux sont systématiquement rééchantillonnés à cette fréquence (via la fonction *resample* d'octave/matlab, qui est efficace). Le signal est ensuite filtré passe-bande par un filtre de Butterworth du cinquième ordre, entre 70 Hz et 650 Hz (filtrage d'octave/matlab efficace). Ensuite, trame par trame :

- L'autocorrélation biaisée du signal est calculée.
- Un coefficient de voisement est calculé (0 ou 1). Il est basé sur deux critères, qui doivent être tous les deux respectés :
  - La distance relative minimale entre l'aucorrélation et la droite qui lie le maximum de l'autocorrélation à son extrémité, ce pour des τ correspondants à la plage de fréquences fondamentales possibles, est calculée. Un premier seuil est utilisé, pour décider : *threshb*.
  - L'énergie de la trame est elle aussi utilisée. Un deuxième seuil est utilisé, pour décider : *threshc*.
- Même si la trame est non-voisée, on détermine f_0, ainsi :
  - Les τ_{min} premiers et τ_{min} derniers coefficients d'autocorrélation sont mis à la moyenne de l'autocorrélation. Puis le maximum de l'autocorrélation est normalisé à 1 et son minimum à 0.
  - On détecte le premier maximum local de l'autocorrélation, à partir des petits τ, dans la zone des τ possibles, qui dépasse un troisième seuil, *thresha*. La position de ce maximum est utilisée pour déterminer f_0.
  - Le calcul d'un polynôme d'ordre 2, passant par le maximum local et ses deux voisins, est effectué ; la position de son maximum est calculé. Ceci permet d'améliorer

la précision de la détermination de f_0 (utile pour les fréquences proches de 600 Hz).

— Le post-traitement se décompose en deux étapes, successives :
  — Si une trame est non-voisée alors que ses deux voisines précédentes et ses deux suivantes sont voisées, on la met voisée et on récupère la valeur de f_0 obtenue à l'étape précédente pour elle. Si une trame est voisée alors que ses deux voisines précédentes et ses deux suivantes sont non-voisées, on la met non-voisée et on rejette la valeur de f_0 obtenue à l'étape précédente pour elle. Note : bien sûr, les valeurs de f_0 trouvées à l'étape précédente est rejettée pour toutes les trames dorénavant définitivement non-voisées.
  — Si la différence relative entre $f_0(j-1)$ et $f_0(j+1)$ est plus petite qu'un seuil, alors que la différence relative entre $f_0(j)$ et $f_0(j+1)$ est plus grande que ce seuil et que la différence relative entre $f_0(j)$ et $f_0(j-1)$ est plus grande que ce seuil aussi, alors $f_0(j)$ est mis à la moyenne de $f_0(j-1)$ et $f_0(j+1)$. On a le quatrième seuil : *threshd*.

Chapitre 3

Tests

3.1 Somme de cosinus – Amplitudes non-contraintes

On va utiliser un modèle de signal simple : une somme de cosinus harmoniques. La fréquence fondamentale f_0 est tirée aléatoirement entre 100 Hz et 600 Hz. L'amplitude du partiel i, de fréquence $f_i = if_0$, est tirée entre 0 et 1. La phase de chaque partiel est tirée aléatoirement entre 0 et 2π. Le nombre de partiels est égal à l'entier juste inférieur ou égal à $f_e/2/f_0$.

On ne mesure que les performances de *thresha* (les deux seuils utilisés pour le voisement, et celui utilisé pour le post-traitement, ne sont pas optimisés ici).

Si le f_0 trouvé est éloigné de moins de 10 % du f_0 qu'on devrait trouver, on l'accepte.

Le plus bas taux d'échecs obtenu est de : 2.20 % pour un seuil valant 1.0 (sur 11340 tests).

Chapitre 4

Conclusion

Utilisez ce pitch-tracker. Faites-moi profitez de vos remarques (notamment, de vos plaintes, en cas de dysfonctionnement).

Bibliographie

[Hes83] Wolfgang J. Hess. *Pitch Determination of Speech Signal – Algorithms and devices*. Springer-Verlag, 1983.