Un *pitchtracker* monophonique

Damien Cirotteau Dominique Fober Stephane Letz Yann Orlarey

Grame - Département Recherche
9,rue du Garet BP 1185
69002 LYON CEDEX 01
Tel +33 (0)4 720 737 00 Fax +33 (0)4 720 737 01
(cirotteau,fober,letz,orlarey)@grame.fr

Résumé

Nous présentons ici un détecteur de hauteur de note basé sur une amélioration du vocodeur de phase. Cette amélioration, permettant une meilleure précision en temps et en fréquence, est parfaitement adaptée à une utilisation en temps réel. Une attention particulière a été portée sur la possibilité d'intégration de ce détecteur dans différents systèmes.

1 Introduction

La musique interactive nécessite des outils de déclenchement en temps réel. La majorité des détecteurs de hauteur de notes est basée sur une analyse des harmoniques du signal impliquant généralement une transformée de Fourier. Il en résulte le traditionnel compromis temps contre fréquence. En effet, la précision en fréquence de l'analyse est proportionnelle au nombre de points du signal discrétisé que l'on utilise pour cette analyse. Le vocodeur de phase est un moyen classique de minimiser ce compromis.

Une amélioration originale du vocodeur de phase utilisant la dérivée du signal a été réalisée dans [1] afin d'affiner la précision de la mesure. Ce vocodeur de phase nous permet d'obtenir la fréquence des différentes composantes d'un signal avec une excellente précision compte tenu de la taille du buffer d'analyse.

Une fois que nous avons les composantes du signal, il faut extraire la fondamentale. Cela revient à trouver la périodicité des composantes du signal. Une fonction de *maximum likelihood* permet de déterminer le meilleur candidat parmi les composantes du signal et éventuellement extrapole la fondamentale même si elle ne fait pas partie des fréquences détectées précédemment.

Enfin, il faut transformer la fréquence en une information directement compréhensible par un système extérieur. La norme Midi a donc tout naturellement été choisie.

Lorsque une note stable est détectée, le *pitchtracker* fourni une information de début de note, puis suit l'évolution de cette note au cours du temps : modulation de hauteur ou d'amplitude. Lorsque la note passe sous un certain seuil d'amplitude ou dépasse un seuil de hauteur paramétrable, une information de fin de note est renvoyée.

Le détecteur que nous avons mis en place n'est pas un interface Midi proprement dite : il n'envoie aucune commande Midi lui-même mais fournit les informations au système extérieur qui se charge des messages Midi. Le *pitchtracker* est fournit sous forme de librairie C. Notre

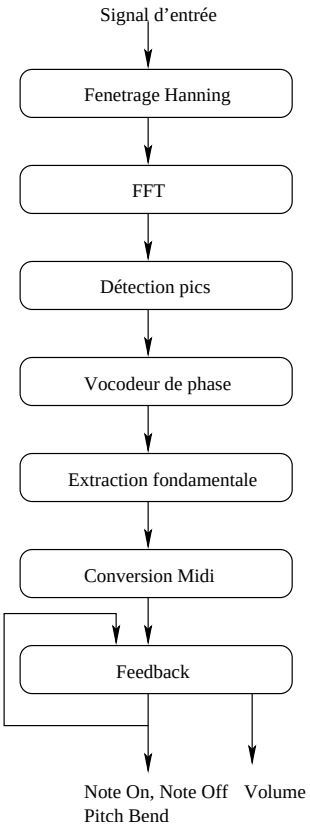

FIG. 1 – Principe de la détection

détecteur ne fonctionne pas de manière autonome et doit être intégré dans un système global. Par analogie notre système correspondrait à la partie mécanique d'un clavier numérique alors que le système hôte prendrait en charge la partie électronique.

2 Obtention des harmoniques

2.1 Récupération des pics significatifs

Il faut tout d'abord extraire du signal les informations fréquentielles pertinentes. Nous réalisons une FFT *N* points du signal pour obtenir son spectre. Le signal aura préalablement été fenêtré par une fenêtre de Hanning.

La recherche des maxima locaux nous donne les composantes sinusoïdales du signal. En plus d'être un maximum local, un pic doit être supérieur à un certain seuil pour être significatif. Typiquement le seuil est compris entre 1 et 5% du maximum du spectre.

Cependant, la précision fréquentielle de chaque composante ainsi obtenue est proportionnelle à la fréquence d'échantillonnage *R* et au nombre de points de la FFT. Ainsi,

$$f_p = m_p \frac{R}{N} \tag{1}$$

avec m_p le numéro du point correspondant au pic dans le spectre et f_p la fréquence de la composante sinusoïdale. Pour une fréquence d'échantillonnage de 44,1 kHz et une FFT de 512 points par exemple, l'écart entre deux points successif dans le spectre est de 86,13 Hz.

Nous avons donc une incertitude très grande sur la fréquence d'un pic. Nous allons voir qu'il est possible d'obtenir la fréquence exacte de ce pic à partir de cette FFT et de la FFT du signal dérivé.

Cependant, il est très important de noter que l'écart entre deux pics successifs donc entre deux harmoniques d'une note doit être supérieur à $\frac{N}{R}$ et donc il est nécessaire d'adapter la taille de la FFT à la hauteur de la note à détecter. Les valeurs précédentes, par exemple, permettent en théorie de détecter des notes supérieures au Fa 1.

2.2 Vocodeur de phase amélioré utilisant la dérivée du signal

Myriam Dessainte-Catherine et Sylvain Marchand ont montré dans [1] et [2] qu'il était possible d'obtenir une bien meilleure résolution en fréquence qu'avec un classique vocodeur de phase. Le principe de base est que la dérivée d'un sinus est un sinus de même fréquence mais de phase différente. Comme un signal audio est la somme d'oscillateurs sinusoïdaux, la dérivée d'un signal audio est un signal de même fréquence mais de phase différente.

L'expression d'un signal audio est donnée par la formule suivante :

$$a(t) = \sum_{p=1}^{P} a_p(t)\cos(\varphi_p(t)) \tag{2}$$

avec

$$\frac{d\varphi_p}{dt} = 2\pi f_p(t) \tag{3}$$

i.e.

$$\varphi_p(t) = \varphi_p(0) + 2\pi \int_0^t f_p(u)du \tag{4}$$

La phase initiale étant quelconque et n'intervenant pas dans la mesure, nous pouvons arbitrairement la choisir nulle. De plus, nous considérons que l'amplitude et la fréquence de chaque composante varie très lentement par rapport à la taille de la fenêtre d'analyse. Leurs dérivées sont donc proches de zéro.

Il vient alors de (2) et (4) :

$$\frac{da(t)}{dt} = \sum_{p=1}^{P} 2\pi f_p(t) a_p(t)\cos(\varphi_p(t) - \frac{\pi}{2}) \tag{5}$$

Le signal que nous traitons étant discret, notons DFT^k le spectre de la DFT de la *k*-ième dérivée du signal.

$$DFT^k[m] = \frac{1}{N}\left|\sum_{n=0}^{N-1} w[n]\frac{d^k a}{dt^k}[n]e^{-j\frac{2\pi}{N}nm}\right| \tag{6}$$

avec w une fenêtre d'analyse de taille N.

Comme nous avons obtenu les pics lors de la précédente étape, nous avons une fréquence approximée (1) de chaque composante. Pour chaque pic *p* nous avons un maximum dans DFT^0et DFT^1. Cela nous permet de déterminer avec précision la fréquence des harmoniques.

$$f_p = \frac{1}{2\pi}\frac{DFT^1[m_p]}{DFT^0[m_p]} \tag{7}$$

Il est également possible, par la même méthode, d'obtenir l'amplitude de chaque partiel de manière précise mais cela est hors du propos de cet article.

3 Extraction de la fondamentale

Maintenant que nous avons obtenu les fréquences des composantes de façon précise, il s'agit de déterminer la fondamentale. Nous avons en entrée de cet algorithme un jeu de fréquences correspondant aux différentes composantes sinusoïdales du signal.

Une fonction du type *maximum likelihood* suivant les idées de [4] est utilisée. Pour chaque composante, calculons la valeur de la fonction suivante :

$$\Gamma(f) = \sum_{p=1}^{P} O_p(f) Y_p(f) \tag{8}$$

avec f la fréquence de la composante du signal sur laquelle nous effectuons la mesure, p les autres composantes du signal de fréquence $h(p)$ et P le nombre de pics trouvés dans le spectre.

Le facteur Y_p est une fonction non nulle à proximité d'une certaine fréquence. C'est une fonction triangulaire centrée sur nf.

$$Y_p(f) = \frac{nf - h(p)}{h(p) - f_{min}} + 1 \;\; f_{min} \leq nf \leq h(p) \tag{9}$$

$$Y_p(f) = \frac{f_{max} - nf}{f_{max} - h(p)} \;\; h(p) \leq nf \leq f_{max}$$

$$Y_p(f) = 0 \;\; sinon$$

avec $n \in [2, P]$.

Ce facteur représente la tolérance acceptée pour qu'une composante A soit considérée comme l'harmonique d'une composante B. Plus A sera proche d'un multiple B, plus la valeur de Y sera importante et si A est multiple de B alors Y est égal à 1. Si A n'est pas proche d'un multiple de B, alors Y sera nulle et A ne sera pas considérée comme "participant" à la note d'harmonique B. Comme nous réalisons un *pitchtracker* monophonique, nous avons choisi une tolérance d'un quart de ton.

Le facteur $O_p(f)$ est fonction de l'index de l'harmonique. Nous considérons que la contribution des harmoniques bas est plus importante que celle des harmoniques élevés. Il faut donc un coefficient reflétant la hauteur de l'harmonique. Soit :

$$O_p(f) = \frac{0.9}{i - 0.1} \tag{10}$$

avec i la valeur arrondie de $\frac{h(p)}{f}$.

La valeur de f qui maximise Γ est la fréquence de la fondamentale. Cependant, il existe des notes dont la fondamentale est absente du signal (et parfois même plusieurs harmoniques bas sont absents en plus de la fondamentale).

Il est possible, grâce à notre fonction de *maximum likelihood* de déterminer une fondamentale même s'il elle n'est pas présente dans le jeu initial des composantes sinusoïdales. Si f_0 est la fréquence maximisant Γ et que f_1, sous-multiple de f_0, renvoie une valeur de Γ plus grande alors $f_1$devient la fondamentale du signal.

4 Conversion midi et feedback

Nous avons obtenu la fréquence de la fondamentale. Cette information n'est pas directement exploitable par un système interactif. Nous allons la convertir en information Midi.

A chaque note Midi correspond une fréquence donnée que nous appellerons valeur fréquentielle Midi. Cette fréquence est donnée par l'accord de l'instrument et les réglages du système. Typiquement, la note 60 correspond à 262 Hz soit Do3.

Pour chaque nouvelle mesure i, nous cherchons la note Midi la plus proche de la fondamentale trouvée au moyen d'une dichotomie. Nous déterminons aussi la valeur du pitch bend fonction de l'écart entre la fréquence vraie et la valeur fréquentielle Midi.

Comme le signal est continu et que nos buffers de mesures sont relativement petits, il est intéressant de prendre en compte les résultats de la mesure précédente et ceci pour différentes raisons. Soit i la mesure courante et $i-1$ la mesure précédente. Soit f la fréquence, M la valeur de la note Midi et P la valeur du pitch bend.

Si f_i est un multiple de f_{i-1}, il est possible que la détermination de la fondamentale lors de l'étape précédente est échouée. Nous vérifions alors, pour l'étape i, s'il existe ou non des pics de fréquence multiple à f_{i-1}. Si c'est le cas, f_i prend la valeur de f_{i-1} et nous mettons M_i et P_i à jour.

Pour la majorité des analyses, quand il n'y a pas de changement de note entre i et $i-1$, f_i et f_{i-1} sont de valeur proche. Cependant, selon le mode de jeu du musicien et les caractéristiques de l'instrument, la fréquence d'une même note peut varier de façon significative (quand il y a un vibrato par exemple).

Il est très important de noter qu'une note, en tant qu'unité mélodique, peut en réalité couvrir plusieurs notes du point de vue fréquentiel. Si un musicien module une même note, notre système ne doit pas envoyer d'information correspondant à une nouvelle note. Notre système doit avoir une mémoire du passé.

Soit ε un seuil paramétrable, si $|f_i - f_{i-1}| < \varepsilon$ nous pouvons considérer qu'il n'y a pas de nouvelle note mais qu'il y a eut modulation de la même note. Nous n'envoyons pas d'information de nouvelle note, i.e. nous avons toujours la même note Midi qu'à l'étape précédente. Seule la valeur du pitch bend est mise à jour. P_i correspondra alors à l'écart entre la valeur fréquentielle de la note Midi $Mi = M_{i-1}$ et la fréquence f_i.

Tant que P_i est inférieur à un certain seuil α, nous considérons que nous avons toujours la même note. Aucune nouvelle note Midi M n'est envoyée. Si P_i dépasse ce seuil, le *pitchtracker* considère que la note Midi a changé ; elle prend alors la valeur Midi correspondant à la fondamentale en cours. L'étendue du pitch bend est paramétrable.

5 Paramétrages

La taille de la FFT Ce paramètre est fonction de l'instrument à détecter. Plus la tessiture est aiguë, plus la taille de la FFT peut être petite. En effet, pour de grandes fréquences nous pouvons choisir de petite fenêtres d'analyse. De plus, nous avons vu en 2.1 que l'écart entre deux composantes du signal devait être supérieur à $\frac{R}{N}$. Plus la fréquence est élevée, plus l'écart entre deux harmoniques successifs est grand et plus la valeur de N peut être petite. Pour toutes les mesures, il faut choisir N le plus petit possible afin d'avoir la meilleure réponse en temps. La taille de la FFT est toujours une puissance de deux en raison des algorithmes de FFT optimisés pour des fenêtres de taille 2^n.

Un instrument tel que la flûte, ayant pour note la plus grave Do3 de fréquence 262 Hz, donne de bons résultats avec des fenêtres de 512 points soit 11 mls.

Étendu du pitch bend : α En fonction du type de détection désiré, nous pouvons choisir l'étendue du pitch bend. Une valeur faible limitera la possibilité de modulation mais renverra plus de notes qu'une valeur importante qui permettra une modulation de hauteur étendue.

Stabilité en fréquence entre deux mesure : ε ε est le seuil pour lequel deux fréquences successives sont considérées comme identiques. Cela permet de décider de la continuité d'une note modulée.

En d'autres termes, deux fréquences f_1 et f_2 sont identiques si et seulement si $|f_1 - f_2| < \varepsilon$.

Stabilité en temps Une note est considérée comme stable donc comme existante si elle se reproduit un certain nombre de fois. Ce paramètre permet de déterminer le nombre minimum de fréquences identiques et successives nécessaires pour déclencher une nouvelle note.

Accord

L'accord de l'instrument détermine la conversion fréquence/Midi et la valeur du pitch-bend. Nous devons donc paramétrer notre système en fonction de l'instrument.

Seuils du *noise gate* Le signal d'entrée passe par un *noise gate.*

Il y deux seuils : haut (entrée) et bas (sortie). Une note est déclenchée si elle passe le seuil haut et est éteinte si elle passe sous le seuil bas.

Dynamique Détermine la dynamique de la réponse en volume en fonction de l'amplitude de la note détectée.

6 Interface avec le système utilisateur

Notre système renvoie plusieurs éléments.

On : une note Midi quand une nouvelle note stable est détectée, 0 sinon. L'utilisateur doit traiter cette information quand elle est différente de 0 comme un NoteOn.

Off : une note Midi quand une note auparavant détectée ne l'est plus, 0 sinon. L'utilisateur doit traiter cette information quand elle est différente de 0 comme un NoteOff.

Vol : une valeur comprise entre 0 et 127, fonction du niveau de la note détectée. Cette information peut être soit la vélocité d'une note lors d'un NoteOn soit traitée comme un contrôleur de volume lorsqu'une note est en cours.

Bend : cette information doit être traitée par un contrôleur du type *pitchwheel.*

Intégration dans un système externe Notre détecteur est intégrable dans tout type de système utilisant le Midi. Plusieurs implémentations utilisant MidiShare ont été réalisées notamment sous Linux, Macinstosh et Windows. Les FFT sont calculées avec la *Fastest Fourier Transform in the West (FFTW)* du MIT. Ces implémentations fournissent une interface utilisateurs permettant d'ajuster les différents paramètres et gèrent les messages Midi.

Une bibliothèque C++ du *pitchtracke*r a été réalisée et permet une réutilisation aisée. Le système extérieur se charge de l'acquisition des données et les fournit au *pitchtracker*. Le *pitchtracker* renvoie les informations sur la note que le système utilisateur doit transformer en Midi.

7 Conclusion

Nous avons réalisé un détecteur de hauteur de note ou *pitchtracker* tirant avantage d'une amélioration originale du vocodeur de phase. Une utilisation dans de nombreux systèmes est possible et un paramétrage souple permet de l'adapter à de nombreux instruments. De nombreuses améliorations restent encore possibles notamment sur le contrôle du volume. Un paramétrage automatique en fonction de l'instrument est à étudier.

Relativement peu de tests ont été réalisés (surtout des tests avec flûte). Des outils de tests systématiques seraient à mettre au point. Ils permettraient de mesurer la fiabilité du système en fonction des différents paramètres et faciliteraient les réglages automatiques.

Références

[1] M. Dessainte-Catherine et S. Marchand, "*High Precision Fourier Analyse of Sounds using Signal Derivatives*", SCRIME 1998, Bordeaux

[2] S. Marchand, "*Improving Spectral Analyse Precision with an Enhanced Phase using Signal Derivatives*", SCRIME 1998, Bordeaux

[3] M. S. Puckette, T. Apel, D.D. Zicarelli, "*Real-time audio analysis tools for Pd and MSP"*, Proceedings of the ICMC 1998, p109-112, 1998

[4] Ö. Izmirli, S. Bilgen, "*Multiple Fundamental Tracking for Polyphonic Note Recognition"*, Recherches et applications en informatique musicale, pp. 305-314, Hermes 1998, Paris

[5] A. M. Noll, "*Pitch determination of human speech by the harmonic product spectrum, the harmonic sum spectrum , and a maximum likelihood estimate"*, Proceedings of the Symposium on Computer Processing in Communications, Vol. XIX, pp. 779-797, Polytechnic Press 1970, New York.

[6] E. Leipp, Acoustique et Musique, Masson 1984, Paris

[7] M.S. Puckette, "*Phase-locked Vocoder"*,Proceedings of the 1995 IEEE ASSP Conference on Applications of Signal Processing to Audio and Acoustics,1995, New-York

[8] M. Dolson, "*The phase vocoder : a tutorial*." Computer Music Journal, 10/4 : pp. 14-26,1986