Contraintes globales pour la sélection des unités en synthèse vocale

Adrian Popescu[1], Cédric Boidin[2], Didier Cadic[2]

[1] Département LUSSI, ENST Bretagne, Brest, France
[2] France Télécom, Division R&D, TECH/SSTP/VMI, Lannion, France
adrian.popescu@enst-bretagne.fr, {cedric.boidin, didier.cadic}@francetelecom.com

ABSTRACT

This work proposes an alternative unit selection method for corpus-based voice synthesis. It introduces the need of long term constraints in the cost function, which cannot be handled by the traditional Viterbi algorithm. Therefore another optimization algorithm, the simulated annealing, has been chosen for our experiments. It has been evaluated on a cost function encouraging long term F0 continuity. Although the results of our experiments do not show real improvement of the overall quality, they are a starting point for further research on this relevant issue.

1. INTRODUCTION

Les systèmes de synthèse par corpus ont amélioré significativement la qualité de la synthèse vocale. Leur succès est basé sur l'utilisation de grandes bases de données, associée à des algorithmes efficaces de sélection des unités. L'étape de sélection consiste à choisir la meilleure suite d'unités parmi toutes celles présentes dans le corpus.

Pour cela, on minimise une fonction de coût mesurant la fluidité du signal de parole synthétisé ainsi que son adéquation aux cibles issues des traitements linguistiques. Elle est généralement définie comme une somme pondérée de coûts-cibles et de coûts de concaténation [1], puis minimisée de manière optimale grâce à un algorithme de programmation dynamique. Ces choix ont prouvé leur efficacité et donnent de bons résultats pour la synthèse de phrases neutres. D'autres algorithmes d'optimisation ont été expérimentés, comme par exemple les algorithmes génétiques [2].

Cependant la forme de la fonction de coût limite le type de contraintes qu'il est possible de prendre en considération : elles ne peuvent porter que sur des unités prises isolément (grâce au coût-cible), ou bien sur des couples d'unités consécutives (coût de concaténation).

Cet article présente plus en détail les algorithmes de sélection actuels et leurs limites, propose alors une nouvelle fonction de coût qui inclut des contraintes à plus long terme, ainsi qu'un algorithme permettant d'intégrer cette fonction de coût : le recuit simulé. Il décrit ensuite les tests effectués et leurs résultats, engage une discussion sur ces résultats puis conclut.

2. SÉLECTION DES UNITÉS

La fonction de coût est généralement définie comme la somme pondérée de coûts-cibles et de coûts de concaténation [1], comme dans l'équation (1).

$$C_L(s) = w_C \sum_{k=2}^{n} C_C(u_{k-1}, u_k) + w_T \sum_{k=1}^{n} C_T(u_k) \qquad (1)$$

où s désigne la séquence des unités $(u_1, u_2, \ldots, u_n)$, $C_L(s)$ le coût total associé à cette séquence, $C_C(u_{k-1}, u_k)$ le coût de concaténation entre les unités u_{k-1} et u_k, $C_T(u_k)$ le coût-cible associé à l'unité u_k et enfin w_C et w_T les pondérations associées.

L'utilisation d'une telle fonction de coût est motivée par les contraintes suivantes : les unités doivent être choisies dans un contexte prosodique et linguistique adéquat (coût-cible) et les transitions entre unités consécutives doivent être fluides (coût de concaténation). Une telle fonction de coût permet également une réduction drastique de la complexité algorithmique.

En effet, pour une séquence de N unités, chacune représentée par M occurrences, le nombre total de combinaisons est M^N ; mais la minimisation d'une fonction de cette forme peut être effectuée avec une complexité réduite à $N \times M^2$ par un algorithme de programmation dynamique, généralement de type Viterbi [3]. Le gain de complexité est important mais, en contrepartie, les contraintes ne peuvent porter que sur une ou deux unités consécutives. Ceci semble cependant suffisant pour synthétiser des phrases neutres de bonne qualité.

3. CONTRAINTES GLOBALES

L'introduction de contraintes globales dans les algorithmes de sélection des unités semble nécessaire pour augmenter le contrôle de la sélection.

Par exemple, comme l'explique Hirai [4], la fonction de coût standard ne permet pas de suivre au mieux un contour de F0 (fréquence fondamentale) dont tous les paramètres ne seraient pas figés, comme par exemple sa composante continue (ou baseline) que l'on peut choisir de ne pas imposer. Il a proposé de résoudre le problème en effectuant plusieurs sélections pour

différentes valeurs des paramètres, puis en choisissant la séquence d'unités au meilleur coût global.

Nous proposons d'aborder différemment ce problème, en introduisant une nouvelle fonction de coût C_N :

$$C_N(s) = w_L C_L(s) + w_G C_G(s) \quad (2)$$

Le terme $C_L(s)$ correspond aux contraintes locales définies dans l'équation (1). Le terme $C_G(s)$ représente la nouvelle contrainte globale, intégrant des relations entre unités non adjacentes. w_L et w_G sont des pondérations.

Avec l'introduction du terme global, il est par exemple possible de suivre au mieux un contour de F0 indépendamment de sa composante continue : le terme C_G est alors égal à l'écart quadratique moyen entre le contour réel de la séquence courante et le contour-cible, ces deux contours étant auparavant centrés sur leurs moyennes.

Un autre exemple de contrainte globale est la volonté d'assurer une certaine continuité de F0 entre unités non adjacentes, plus particulièrement autour des régions non-voisées. Cette volonté ne peut pas être prise en compte par le coût de concaténation habituel d'ordre 1.

4. LE RECUIT SIMULÉ

L'introduction de contraintes globales augmente la complexité de la sélection, et nous ramène à un problème de dimension M^N.

Plusieurs algorithmes peuvent être utilisés pour résoudre ce problème, mais la solution optimale est, dans le cas général, hors de portée.

Dans cette étude nous utilisons l'algorithme du recuit simulé [5, 6] décrit en figure 1.

```
Seq_Courante      = Seq_Initiale;
Temp_Courante  = Temp_Initiale;
DO
      Seq_Perturbée     = PERTURB (Seq_Courante);
      Seq_Courante      = ACCEPT (Seq_Perturbée,
                  Seq_Courante, Temp_Courante);
      Temp_Courante  = UPDATE (Temp_Courante);
WHILE NOT (Critère_Arrêt);
RETOUR (Seq_Courante);
```

Figure 1 : Pseudo-code du recuit simulé.

Les variables utilisées dans la figure 1 sont les séquences d'unités et la température. La séquence initiale est choisie aléatoirement. Les trois fonctions utilisées dans l'algorithme sont PERTURB, ACCEPT, UPDATE :

PERTURB : cette fonction effectue une modification de la séquence d'unités courante pour obtenir une séquence perturbée. Dans notre cas, elle consiste à remplacer un certain nombre d'unités adjacentes par d'autres unités candidates aléatoirement choisies. Cette fonction de perturbation est très simple ; elle pourra par la suite être remplacée par des heuristiques plus complexes de choix des nouvelles unités guidant efficacement l'exploration.

ACCEPT : cette fonction accepte ou non la perturbation. La perturbation est acceptée avec une probabilité *P* donnée par l'équation (3).

$$P = \begin{cases} \exp\left(-\dfrac{C(s_p) - C(s_c)}{T}\right) & \text{if} \quad C(s_p) \geq C(s_c) \\ 1 & \text{if} \quad C(s_p) < C(s_c) \end{cases} \quad (3)$$

T est la température courante, $C(s_c)$ est le coût associé à la séquence courante, $C(s_p)$ celui associé à la séquence perturbée.

Ainsi, toutes les perturbations associées à une baisse de coût sont acceptées et celles associées à une augmentation du coût sont acceptées avec une probabilité dépendant de la hausse du coût et de la température.

UPDATE : cette fonction met à jour la température. Deux lois sont généralement proposées pour la descente de température : la loi logarithmique et la loi géométrique. La première assure la convergence vers l'optimum global mais, pour des raisons de temps de calcul, nous avons choisi la loi géométrique, plus rapide.

Une fois les trois fonctions PERTURB, ACCEPT et UPDATE définies, les paramètres suivants doivent être fixés : température initiale, température finale, ainsi que la raison de la loi géométrique.

Le choix des paramètres est important pour assurer un bon comportement de l'algorithme. A titre d'exemple, si la température reste trop élevée, les perturbations sont trop souvent acceptées et aucune convergence intéressante ne peut être constatée ; si la température reste trop faible, les augmentations de coût sont rarement acceptées et on s'enferme rapidement dans un minimum local. Il faut donc diminuer la température lentement et dans un intervalle approprié afin d'explorer suffisamment de possibilités tout en assurant une convergence convenable. Dans notre cas ces paramètres sont fixés empiriquement.

5. COMPARAISON OBJECTIVE

5.1. Conditions de test

Cette partie vise à comparer les méthodes de sélection de façon objective. Le corpus de référence (7 heures de phrases du "Monde" enregistrées par une voix d'homme à 16 kHz), les fonctions de coût-cible et de coût de concaténation sont celles mises en œuvre dans

le système de synthèse de parole de France Télécom. Ces coûts sont basés sur des paramètres acoustiques ainsi que sur des étiquettes linguistiques et prosodiques issues des niveaux linguistiques du moteur de synthèse. Les unités de base sont les diphones. Aucun algorithme de traitement du signal n'est appliqué sur les unités, excepté un overlap-add lors de leur concaténation. Pour chaque diphone-cible, les 100 meilleures unités candidates sont présélectionnées en fonction de leur coût-cible, quelle que soit la méthode de sélection utilisée par la suite.

Un premier ensemble de 75 phrases journalistiques aléatoirement choisies dans "le Monde" représente le corpus de test A.

5.2. Comparaison algorithme de Viterbi - Recuit simulé

Une première expérience vise à mesurer les performances du recuit simulé pour la minimisation de la fonction de coût C_L.

Plusieurs variantes sont implémentées pour la fonction PERTURB, qui consiste à modifier un sous-ensemble d'unités adjacentes de la séquence courante. Dans notre cas le sous-ensemble et les unités de remplacement sont choisis aléatoirement. Plusieurs tailles de sous-ensemble sont testées : 1 unité, 2 unités adjacentes, 3 unités adjacentes, ou encore une taille aléatoire comprise entre 1 et 5. Il est à noter que les unités remplacées sont adjacentes dans la séquence courante, mais pas nécessairement dans le corpus de référence.

La table 1 montre le surcoût moyen de la solution fournie par le recuit simulé par rapport à la solution optimale (*i.e.* celle du Viterbi), calculé sur le corpus A.

Table 1 : Surcoût moyen de la solution du recuit simulé par rapport à la solution optimale (Viterbi), pour plusieurs tailles de fenêtres remplacées.

Nombre d'unités remplacées	Hausse de coût
1	15%
2	10%
3	14%
Aléatoire entre 1et 5	13%

On choisit donc pour la suite des expériences une fenêtre de modification de taille 2 unités, induisant un surcoût moyen de 10% par rapport au coût minimal. A titre de comparaison, la distribution des coûts de toutes les séquences possibles a une moyenne et un écart-type respectivement égaux à 270% et 35% fois le coût optimal. Le test d'écoute présenté au paragraphe 6.2 mesure l'impact perceptif de ce surcoût.

L'utilisation du recuit simulé augmente également le temps de calcul. Typiquement, la sélection des unités pour une phrase prend 10 secondes avec le recuit simulé contre 1 seconde avec l'algorithme de Viterbi.

6. EXPÉRIENCES ET RÉSULTATS

6.1. Exemple de contrainte globale

Nous avons ensuite appliqué l'algorithme du recuit simulé à la fonction de coût C_N de l'équation (2), en y intégrant la contrainte globale suivante :

$$C_G(s) = \sum_{i=2}^{Nv} B\left(\left| f_0^i - f_0^{i-1} \right| \right) \quad (4)$$

avec :

$$\mathrm{B}(x) = \begin{cases} x & if \quad x \geq Seuil \\ 0 & if \quad x < Seuil \end{cases} \quad (5)$$

f_0^i désigne la fréquence fondamentale du *i*-ième noyau vocalique et *Nv* le nombre de noyaux vocaliques de la phrase.

La contrainte globale décrite ci-dessus pénalise les écarts de F0 entre noyaux vocaliques adjacents, en particulier autour des régions non-voisées, ce qui ne peut être intégré dans le coût de concaténation.

Le *Seuil* est fixé à 5 demi-tons. Ce seuil est égal à celui employé par Mertens [7] pour caractériser l'intervalle primaire. Cet intervalle a été défini en lien avec le seuil du glissando caractérisant la variation minimale de fréquence perceptible.

6.2. Test d'écoute

Un deuxième ensemble de 30 phrases journalistiques aléatoirement choisies dans "le Monde" a été construit pour le test d'écoute ; c'est le corpus de test B. 15 auditeurs naïfs ont participé au test.

Chaque auditeur est invité à noter son impression globale pour chacune des 30 phrases sur une échelle à 5 niveaux, suivant la recommandation P.800 de l'UIT, où 1 correspond à "mauvais" et 5 à "excellent".

Les résultats du test sont présentés dans la table 2.

Table 2 : Résultats du test d'écoute sur les phrases du corpus B.

Corpus B	MOS
Algorithme de Viterbi, C_L	3.7
Recuit simulé, C_L	3.46
Recuit simulé, C_N	3.35

L'algorithme de Viterbi donne de meilleurs résultats MOS que le recuit simulé pour la fonction de coût C_L traditionnelle, ce qui indique que le surcoût de 10% est audible.

La dernière ligne de la table donne les résultats MOS pour la nouvelle fonction de coût C_N intégrant la

contrainte globale. Les résultats sont un peu inférieurs à ceux obtenus pour la fonction de coût standard C_L, toujours avec le recuit simulé, ce qui semble discréditer la contrainte globale choisie pour cette expérience.

7. DISCUSSION

La présente étude aborde les limites de la synthèse par corpus et propose une méthode de sélection des unités alternative qui utilise une fonction de coût différente et un algorithme d'optimisation associé. Les contraintes globales utilisées et l'algorithme d'optimisation sont cependant loin d'être au point.

En effet, la contrainte globale choisie ici est trop restrictive, décourageant toutes les discontinuités de F0 supérieures à 5 demi-tons. Les discontinuités fortes de F0 ont une signification linguistique en parole spontanée. Il faudrait donc, sur la base d'informations linguistiques, encourager certaines de ces discontinuités et en décourager d'autres.

L'utilisation du recuit simulé pour résoudre ce problème complexe est pratique puisque flexible et relativement indépendante des contraintes globales que nous souhaitons imposer. On pourrait cependant choisir d'autres algorithmes, ou définir une fonction de perturbation plus élaborée, qui substituerait intelligemment les unités de manière à faire converger l'algorithme plus rapidement.

8. CONCLUSIONS

Cette étude propose une méthode alternative de sélection des unités. Nous insistons sur la nécessité d'introduire de nouvelles contraintes dans la fonction de coût, à portée plus large que celles actuellement traitées par les algorithmes de sélection. La forme et la portée de ces contraintes imposent le choix d'un nouvel algorithme d'optimisation. Pour nos expériences, nous avons adopté le recuit simulé, qui donne des séquences d'unités proches en qualité de celles du Viterbi, bien que sous-optimales. Nous avons évalué ce nouvel algorithme de sélection sur une contrainte globale encourageant la continuité de F0 à long terme. Les résultats montrent que cette méthode détériore légèrement la qualité moyenne des phrases synthétisées, la contrainte choisie restreignant indistinctement toutes les variations de pitch, naturelles ou non. Les efforts doivent être poursuivis pour mettre au point un catalogue de contraintes globales plus efficaces ainsi que des heuristiques capables de guider intelligemment les perturbations en fonction des contraintes imposées et ainsi accélérer la convergence.

BIBLIOGRAPHIE

[1] A.W. Black and N. Campbell, "Optimising selection of units from speech databases for concatenative synthesis", *Proceedings of Eurospeech*, Rhodes, Greece, September 1995.

[2] R. Kumar, “A Genetic Algorithm for Unit Selection based Speech Synthesis”, *Proceedings of ICSLP*, 2004.

[3] A.J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm", *IEEE Trans. Inf. Th.*, vol. IT13, pp. 260-269, 1967.

[4] T. Hirai, S. Tenpaku, and K. Sikano, "Speech Unit Selection based on Target Values driver By Speech Data In Concatenative Speech Synthesis", *IEEE Workshop on Speech synthesis*, 2002.

[5] P.Y. Le Meur, "Synthèse de la parole par unités de taille variable", PhD. Thesis, 1996.

[6] S. Kirkpatrick, C. Gelatt, and M. Vecchi, "Optimization by simulated annealing", *Science*, Number 4598, 13 May 1983.

[7] P. Mertens, "Automatic recognition of Intonation in French and Dutch", *Proceedings of Eurospeech*, Vol. 1, pp. 46-50, 1989.