Est-il possible de restaurer automatiquement des signaux audio corrompus par du bruit impulsionnel ?

Laurent OUDRE

ENS Cachan, CMLA
61 Avenue du Président Wilson, 94230 Cachan, France
laurent.oudre@cmla.ens-cachan.fr

Résumé – Cet article décrit une méthode automatique pour la détection et la suppression du bruit impulsionnel (clics, scratchs, ...) dans les signaux audio. En se basant uniquement sur l'hypothèse que le signal peut se modéliser localement comme une réalisation d'un processus autorégressif, l'algorithme développé ici est capable à la fois d'identifier et de restaurer les échantillons corrompus. Cette communication soutient qu'une analyse jointe et soignée de ces deux problèmes d'estimation et de restauration rend le défi de l'automatisation réalisable dans une grande mesure. L'algorithme proposé ici réalise de fait sans changement de paramètres une restauration satisfaisante dans une grande variété de signaux musicaux, et il peut être testé en ligne sur d'autres signaux du choix de l'utilisateur.

Abstract – This article presents a method for restoring audio signals corrupted by impulsive noise such as clicks, bursts or scratches. The algorithm takes as input a degraded audio signal and automatically detects the locations of the degraded samples and replaces them with more appropriate values. Both steps (detection and interpolation) are based on the assumption that the signal can locally be modeled as a realization of an autoregressive process. The results obtained on several types of signals (classical, jazz, vocal...) show that a fully automatic method, with a carefully fixed set of parameters, can achieve good performance on a wide range of degraded audio signals.

1 Introduction

Les dégradations présentes dans les signaux audio peuvent se regrouper en deux catégories principales : les dégradations *globales* qui affectent tous les échantillons (par exemple du bruit de fond), et les dégradations *locales* qui n'affectent que certains groupes d'échantillons, causant ainsi une discontinuité dans la forme d'onde. Nous nous focalisons ici sur ce second type de dégradations, qui désigne en fait un large spectre de défauts (clics, grésillements, claquements, ...) que l'on peut trouver par exemple sur d'anciens vinyles. L'ordre de grandeur pour la durée de ces dégradations varie entre moins de 20 microsecondes et 4 millisecondes (de 1 à 200 échantillons avec une fréquence d'échantillonnage de 44.1 kHz). Dans des cas extrêmes, l'on peut trouver plus de 2000 dégradations en une seconde [1].

Afin de débruiter des signaux audio corrompus par ce type de bruit, il existe des méthodes basiques qui préconisent l'application de filtrage (par exemple médian) directement sur la forme d'onde, afin de la lisser et d'enlever les échantillons aberrants. Néanmoins, dans le cas où les dégradations durent plus de quelques échantillons, il est nécessaire de recourir à des méthodes plus complexes. Parmi elles, de nombreuses sont basées sur l'hypothèse que localement, le signal est modélisable comme une réalisation d'un processus AR (autorégressif) ou ARMA (autorégressif et moyenne mobile) [3]. Cette hypothèse est en pratique souvent pertinente, si l'on considère que le mode de production du son (instrument ou voix) peut être vu comme la réponse d'un filtre linéaire (par exemple le conduit vocal) à une excitation (par exemple le souffle). Dans ce cas, et en supposant que l'excitation est un bruit blanc gaussien, ce modèle source-filtre a pour expression celle d'un modèle AR.

Dans cet article, nous utilisons aussi cette hypothèse afin dans un premier temps de localiser les échantillons corrompus (qui schématiquement sont ceux ne correspondant pas au modèle) puis de les remplacer par des valeurs adéquates (donc plus proches de celles du modèle). La plupart des méthodes existantes (en particulier commerciales) pour la suppression du bruit impulsionnel reposent sur l'ajustement fin de nombreux paramètres contrôlant et modélisant le type de dégradations à détecter et à restaurer. Au contraire, la méthode proposée offre la particularité de n'utiliser qu'un nombre restreint de paramètres, ce qui permet à la fois de mieux contrôler leur influence, mais aussi de tendre vers un algorithme automatique de restauration.

2 Méthode

Considérons une trame de signal audio $\mathbf{s} \in \mathbb{R}^{N_w}$, corrompu par un bruit impulsionnel additif $\mathbf{n}$. Le signal observé (et donc dégradé) $\mathbf{x}$ s'écrit

$$x_t = s_t + n_t. \tag{1}$$

Supposons que le signal original $\mathbf{s}$ est une réalisation d'un processus AR. Alors il existe un ordre $p \in \mathbb{N}^*$ et des coeffi-

cients $\mathbf{a} = [a_1, \ldots, a_p]^t \in \mathbb{R}^p$, $a_p \neq 0$ tels que

$$s_t = -\sum_{k=1}^{p} a_k \, s_{t-k} + e_t \tag{2}$$

où $\mathbf{e}$ est un bruit blanc de moyenne nulle et de variance σ_e^2.

2.1 Détection

Supposons que l'on dispose d'une estimation $\hat{\mathbf{a}}$ des paramètres AR (par exemple par un algorithme de Levinson-Durvin). Grâce à (1) et (2), on peut écrire :

$$\hat{d}_t := x_t + \sum_{k=1}^{p} \hat{a_k} \, x_{t-k} \tag{3}$$

$$= e_t + n_t + \sum_{k=1}^{p} a_k \, n_{t-k} + \sum_{k=1}^{p} (\hat{a_k} - a_k) \; x_{t-k}. \tag{4}$$

On voit ici que le critère $|\hat{d}_t|$ peut être utilisé comme critère de détection pour le bruit impulsionnel. En effet, si du bruit impulsionnel est effectivement présent ($n_t \neq 0$) et si

- l'erreur d'estimation $(\hat{a_k} - a_k)$ des paramètres AR est faible ;
- $|e_t|$ est petit devant $|n_t|$;
- les p échantillons précédents ne sont pas corrompus ou $\sum_{k=1}^{p} a_k \, n_{t-k}$ est petit

alors $|\hat{d}_t| \approx |n_t|$. Plus généralement, le terme $|\hat{d}_t|$ doit avoir des valeurs élevées lorsque du bruit impulsionnel est présent, et des valeurs peu élevées lorsque les échantillons ne sont pas corrompus.

On peut ainsi seuiller ce critère grâce à un seuil de la forme

$$\lambda_K = K \hat{\sigma_e} \tag{5}$$

qui permet de tester (si $K > 1$) que le bruit impulsionnel a bien une amplitude significative par rapport à l'excitation. Afin d'éviter que l'on détecte des échantillons dispersés, un deuxième paramètre b dit *de fusion* assure que si l'espace séparant deux échantillons détectés comme corrompus est inférieur à b, alors tous les échantillons situés entre eux sont automatiquement considérés également corrompus.

2.2 Interpolation

Une fois estimé l'ensemble T des échantillons corrompus sur la trame, notre but est d'estimer $\mathbf{s}(T)$, connaissant seulement $\mathbf{s}(\widetilde{T}) = \mathbf{x}(\widetilde{T})$ ($\widetilde{T}$ étant l'ensemble des échantillons non corrompus), et les estimations des paramètres AR. Pour l'interpolation, nous tentons de minimiser le critère suivant

$$Q = \sum_{t=p+1}^{N} \left| s_t + \sum_{k=1}^{p} \hat{a_k} \, s_{t-k} \right|^2 = \sum_{t=p+1}^{N} |e_t|^2 , \tag{6}$$

qui peut s'interpréter comme l'erreur quadratique de reconstruction. Dans notre cas, les valeurs de $\mathbf{s}$ ne sont connues que sur $\widetilde{T}$ et le critère peut se réécrire sous la forme

$$Q = \mathbf{s}(T)^t \, \mathbf{C} \, \mathbf{s}(T) + 2 \, \mathbf{s}(T)^t \, \mathbf{d} + \Gamma \left(\mathbf{x}(\widetilde{T}) \right), \tag{7}$$

où $\Gamma \left(\mathbf{x}(\widetilde{T}) \right)$ est un terme dépendant uniquement de $\mathbf{x}(\widetilde{T})$.

Les expressions de $\mathbf{C}$ et $\mathbf{d}$ peuvent être obtenues comme des fonctions explicites de $\mathbf{x}(\widetilde{T})$ et $\hat{\mathbf{a}}$ par simple identification. Finalement, on reconstruit les échantillons corrompus en résolvant

$$\mathbf{C} \, \hat{\mathbf{s}}(T) = -\mathbf{d}. \tag{8}$$

2.3 Résumé de l'algorithme

L'algorithme final prend en entrée un signal dégradé et le divise en trames de longueur N_w avec un recouvrement de 75%. Sur chaque trame, les paramétres AR d'ordre p sont estimés et utilisés pour la détection (paramétres K et b) puis pour l'interpolation. Le signal est ensuite reconstruit grâce à une procédure d'*overlap-add*. Le processus est ensuite itéré une seconde fois et donne en sortie le signal reconstruit.

3 Résultats

La méthode finale repose sur les 4 paramètres suivants

- Le seuil de détection $\lambda_K = K \hat{\sigma_e}$
- Le paramètre de fusion b
- L'ordre du modèle AR p
- La taille de fenêtre N_w

Notons que les paramètres N_w et p sont communs à la phase de détection et d'interpolation.

3.1 Influence de N_w et p

Pour déterminer ces paramètres, nous allons nous concentrer sur la seule phase d'interpolation. Intuitivement, ces deux paramètres dépendent fortement de la taille des clics que l'on veut détecter et restaurer.

- L'ordre p du modèle dépend de la complexité du signal sur la trame. Afin d'obtenir une interpolation acceptable *Janssen et al* [2] proposent d'utiliser $p = 3N_{max} + 2$ où N_{max} est la valeur théorique du plus grand clic à restaurer. Il est en effet logique de considérer que p doit être au moins supérieur à N_{max}.
- La taille de fenêtre N_w dépend de l'ordre p, mais aussi de la taille maximale de clic N_{max}. Un ordre de grandeur peut être estimé en sachant que l'estimation des paramètres du modèle AR nécessite de nombreux échantillons pour pouvoir être pertinente et que le traitement par trame avec 75% de recouvrement nécessite que l'on ait $N_w > \frac{8}{3}p$ pour que tous les échantillons soient traités au moins une fois.

L'influence de ces paramètres a été testée sur un signal de musique d'une seconde (The Beatles) échantilloné à 44.1 kHz et sur lequel N_{max} échantillons sont supprimés puis interpolés (on suppose donc T connu). L'idée étant de constuire un algorithme automatique, l'on a testé des valeurs de p et N_w paramétrées par N_{max}, et calculé le SNR entre le signal original

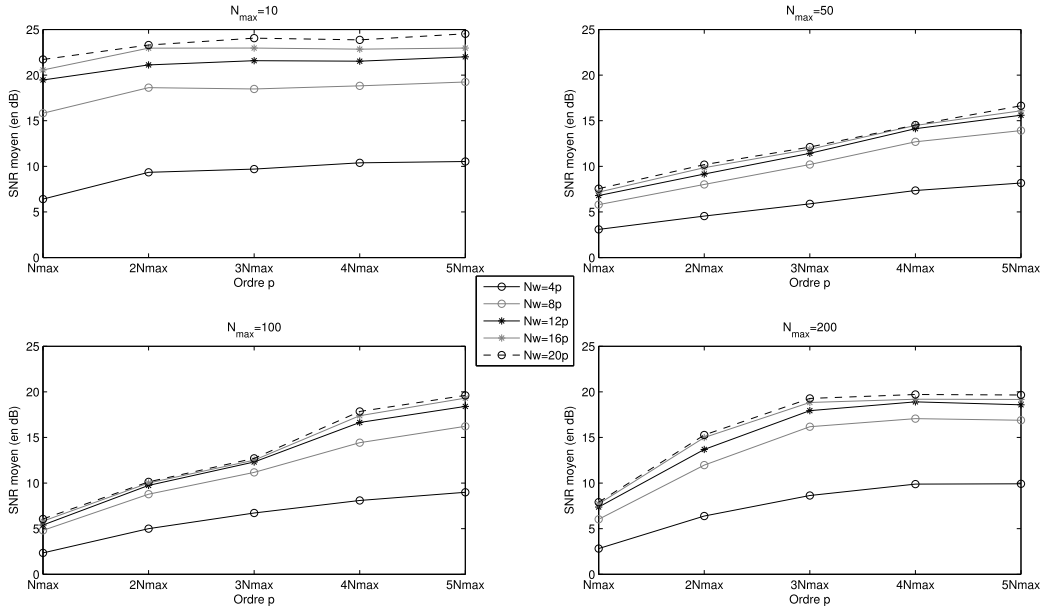

FIGURE 1 – SNR_T moyen (100 expériences) obtenus en reconstruisant une dégradation aléatoire de taille N_{max} pour différentes valeurs de p et de N_w.

s et le signal reconstuit $\hat{s}$ sur les échantillons manquants :

$$SNR_T = 10\log_{10}\frac{\sum_{k\in T}|s_k|^2}{\sum_{k\in T}|s_k-\hat{s_k}|^2}. \qquad (9)$$

Cette expérience a été ré-itérée 100 fois pour quatre tailles de dégradations (N_{max} = 10, 50, 100, 200) et les résultats sont présentés sur la Figure 1. Ils permettent de conclure que :

- La taille de fenêtre n'a pas une influence significative sur les résultats au delà de $N_w = 8p$. Pour toutes les simulations, si une différence siginificative existe entre $N_w = 4p$ et $N_w = 8p$, elle est négligeable pour des valeurs supérieures. Sachant que cette taille contrôle en partie le temps d'exécution de l'algorithme, il est intéressant de choisir la plus petite taille possible, qui est donc dans notre cas $N_w = 8p$.
- En ce qui concerne l'ordre p, les cas sont différents selon les simulations : pour $N_{max} = 10$, il n'a que peu d'influence, pour $N_{max} = 50$ et $N_{max} = 100$ il doit être le plus élevé possible, et finalement pour $N_{max} = 200$ une valeur de $p = 3N_{max}$ est suffisante. Considérant que Janssen et at [2] proposent d'utiliser $p = 3N_{max} + 2$ et toujours dans le souci d'optimiser le temps d'exécution du code, il semble en effet judicieux de supposer pour la suite que $p = 3N_{max} + 2$ est un bon compromis.

3.2 Influence de λ_K et b

Le processus de détection est quant à lui paramétrisé par λ_K et b. On suppose dans la suite que $N_{max} = 100$ ce qui implique $p = 302$ et $N_w = 2416$. Intuitivement :

- Si λ_K est élevé, le système est conçu pour détecter seulement les dégradations importantes (clics à large amplitude). Si λ_K est faible, tous les clics seront détectés, mais il se peut que le bruit de fond le soit aussi.
- Le paramètre b contrôle de façon indirecte la longueur des dégradations détectées. Il influence aussi largement le processus d'interpolation : en effet, les résultats obtenus ne sont pas les mêmes lorsque l'on reconstruit deux clics de petite taille ou un grand. On doit donc choisir b suffisamment grand pour créer des clics de taille raisonnable.

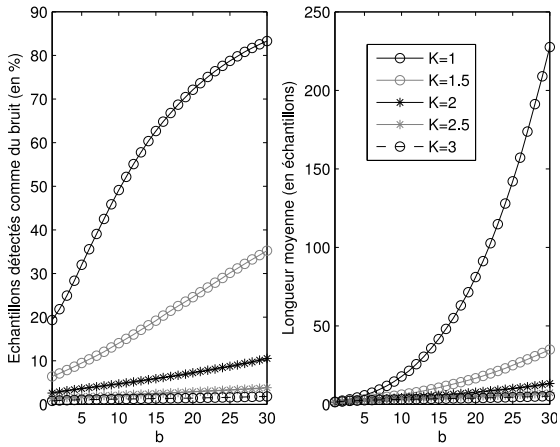

FIGURE 2 – Nombre d'échantillons détectés comme du bruit (en % du nombre total d'échantillons) et longueur moyenne d'un clic détecté pour différentes valeurs de λ_K et b.

Nous avons testé ces hypothèses sur un extrait de 30 secondes de musique classique, corrompu par des scratchs et des clics (Mussorgsky). Nous avons testé plusieurs valeurs pour λ_K (K = 1, 1.5, 2, 2.5 et 3) et b (tous les entiers entre 1 et 30).

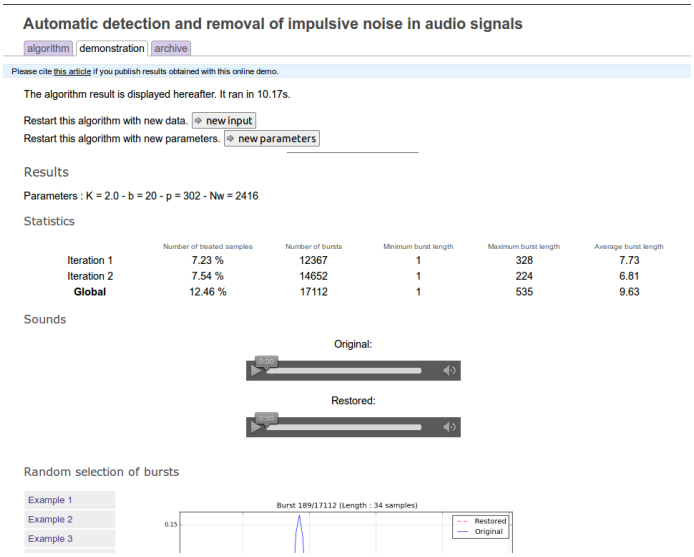

FIGURE 3 – Restauration d'un morceau de Mussorgsky. Les paramètres utilisés sont $K = 2$, $b = 20$, $p = 302$, $N_w = 2416$.

La figure 2 présente les statistiques liées aux différentes configurations : le pourcentage d'échantillons qui ont été détectés comme du bruit, ainsi que la longueur moyenne des clics qui ont été détectés. Cette expérience nous permet de réaliser qu'il est effectivement possible de contrôler le comportement de l'algorithme avec ces deux paramètres. En effet, selon les valeurs de λ_K et b on peut détecter de 1% à 82% des échantillons comme étant du bruit, et la longueur d'un clic peut varier entre 1 et 230 échantillons (0.02 ms et 5.2 ms). Sachant qu'on considère en général que les dégradations impulsionnelles concernent au maximum 10% des échantillons [1], il est évident que certaines configurations ne sont pas réalistes.

En écoutant les reconstructions, on peut faire les conclusions suivantes

- pour $K = 1$ et $b = 1$, tous les clics sont encore présents et le signal est légérement distordu ;
- pour $K = 3$ et $b = 1$, aucun traitement ne semble avoir été fait ;
- pour $K = 1$ et $b = 30$, tous les clics ont été enlevés mais le signal est très distordu et sourd ;
- pour $K = 3$ et $b = 30$, seulement quelques clics ont été enlevés mais le signal est moins distordu que dans les autres configurations.

Un bon compromis semble pouvoir être trouvé avec $K = 2$ et $b = 20$: on peut d'ailleurs voir sur la figure 2 que cela correspond à environ 7% d'échantillons détectés comme du bruit et à une taille moyenne de clic d'environ 8 échantillons, ce qui est parfaitement réaliste.

3.3 Démonstrateur en ligne

L'algorithme a été testé sur huit extraits musicaux de 30 secondes présentant de vraies dégradations (clics, scratchs, ...). Ces fichiers audio ont été fournis et utilisés par Godsill pour l'évaluation de diverses méthodes de débruitage [1] et consistent en un acapella, quatre morceaux de musique classique et trois morceaux de jazz. Après application de l'algorithme avec les paramètres standard ($K = 2$, $b = 20$, $p = 302$, $N_w = 2416$), les clics semblent avoir tous été supprimés et la qualité du signal n'est pas altérée. Une démonstration en ligne, ainsi que les signaux utilisés pour les tests sont disponibles sur

`http://dev.ipol.im/~oudre/ipol_demo/lo_ar2/`
(login = demo, password =demo).

Sur ce site, l'utilisateur peut aussi télécharger ses propres signaux pour tester l'algorithme, et varier les différents paramètres pour comparer les résultats.

La figure 3 présente un exemple de page de résultats. On voit que 7.23% des échantillons ont été détectés comme du bruit lors de la première itération de l'algorithme, et 7.54% à la deuxième itération. En tout, 12.46% des échantillons ont été traités. On trouve aussi des informations sur le nombre de clics détectés ainsi que des statistiques sur leur taille.

4 Conclusion

La méthode proposée ici permet de restaurer de façon automatisée des signaux audio corrompus par du bruit impulsif. Le nombre limité de paramètres, ainsi que l'étude de leur influence permet d'obtenir un algorithme simple, efficace et aisément adaptable. De plus, l'algorithme a été conçu dans un soucis de reproductibilité et il peut être testé avec plusieurs jeux de paramètres sur n'importe quel signal proposé par l'utilisateur.

Remerciements

Travaux financés en partie par Office of Naval Research Grant N00014-97-1-0839 et European Research Council, advanced grant "Twelve labours"

Références

[1] S.J. Godsill and P.J.W. Rayner. *Digital Audio Restoration - A Statistical Model-Based Approach*, chapter 5 - Removal of clicks, pages 99–134. Springer-Verlag London, 1998.

[2] A. Janssen, R. Veldhuis, and L. Vries. Adaptive interpolation of discrete-time signals that can be modeled as autoregressive processes. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 34(2) :317–330, 1986.

[3] SV Vaseghi and PJW Rayner. Detection and suppression of impulsive noise in speech communication systems. In *IEEE Proceedings on Communications, Speech and Vision*, volume 137, pages 38–46, 1990.

1. `http://www-sigproc.eng.cam.ac.uk/~sjg/springer/index.html`