Parcours ATIAM
Rapport de stage
08/02/16 - 15/07/16

Synthèse de textures sonores à partir de statistiques temps-fréquence

Hugo Caracalla

Encadré par Axel Roebel, IRCAM

Ce rapport est le compte-rendu du stage que j'ai pu effectuer auprès d'Axel Roebel au laboratoire des Sciences et Techniques de la Musique et du Son (STMS) à l'IRCAM (Institut de Recherche et Coordination Acoustique/Musique), du 8 février 2016 au 15 juillet 2016.

Comme il le sera rappelé au cours de ce rapport, ce stage s'inscrit dans la continuité de la thèse de Wei-Hsiang Liao "Modelling and transformation of sound textures and environmental sounds" [Liao, 2015], achevée en mai 2015. Il est aussi à noter que je poursuivrai le travail entamé avec ce stage en thèse à partir d'octobre prochain.

Mots-clefs : synthèse, textures sonores, imposition, statistiques.

Notations & abréviations

x	Signal fréquentiel (discret)
w	Fonction de fenêtrage
$\Re(x), \Im(x)$	Partie réelle/imaginaire de x
$\|x\|$	Module de x
$\arg(x)$	Argument de x
$\|\|x\|\|$	Norme 2 de x
$\mathcal{F}(x), \widehat{x}$	Transformée de Fourier (discrète) de x
$\mathcal{AC}_x$	Auto-corrélation de x
$\mathcal{IC}_{x,y}$	Inter-corrélation de x et y
$\mu(x)$	Moyenne (moment d'ordre 1) de x
$\sigma(x)$	Écart-type (moment d'ordre 2) de x
$\eta(x)$	Asymétrie (moment d'ordre 3) de x
$\kappa(x)$	Kurtosis (moment d'ordre 4) de x

TF	Transformée de Fourier (discrète)
STFT	Transformée de Fourier à court-terme
AC	Auto-corrélation
IC	Inter-corrélation

Table des matières

Chapitre 1

Introduction

Ce stage visant à poursuivre le développement d'un nouvel algorithme de synthèse de texture sonore, je commencerai par situer le projet vis-à-vis de son utilité et des méthodes existantes. Mais avant d'aller plus loin, commençons par introduire le terme au centre de ce rapport.

1.1 Définition d'une texture sonore

Les textures sonores représentent une catégorie de signaux sonores distincte de la parole et de la musique. Là où ces dernières servent à transmettre un message, une intention, les textures sonores sont généralement le fruit d'un environnement et sont principalement intéressantes en ce qu'elles contiennent des informations sur cet environnement.

Trouver des exemples de textures sonores n'est pas bien compliqué : le bruit que fait la pluie en tombant, un tonnerre d'applaudissements, le crépitement d'un feu, le ronronnement d'un moteur... la liste est longue. Mais si en citer ne pose aucune difficulté, la partie se corse lorsque l'on cherche à définir précisément ce qu'est une texture sonore : on semble y trouver une impression de "papier-peint sonore" que l'on identifie facilement après quelques secondes d'écoute, sans que l'on parvienne pour autant à y reconnaître une organisation ou une structure harmonique distincte.

Comme le fait remarquer Saint-Arnaud dans [Saint-Arnaud, 1995] certaines idées centrales apparaissent cependant assez rapidement lors de ces tentatives de définitions : la similitude entre échantillons d'une texture pourvu qu'ils soit assez longs, ou encore la présence d'un aspect aléatoire.

La conclusion à laquelle arrive Saint-Arnaud au fil de ses publications sur le sujet et que nous adopterons est synthétisée par Diemo Schwartz dans [Schwarz, 2011] comme suit : une texture est formée de particules sonores, ou atomes, se superposant de façon aléatoire tout en suivant une certaines organisation à haut niveau : ces caractéristiques de haut-niveau apparaissent après quelques secondes d'écoute, et doivent se maintenir tout au long de la texture.

Un point important est que ces atomes sonores ne sont pas discernables seuls : de la même façon que l'on ne perçoit pas chaque point de l'image 1.1 individuel-

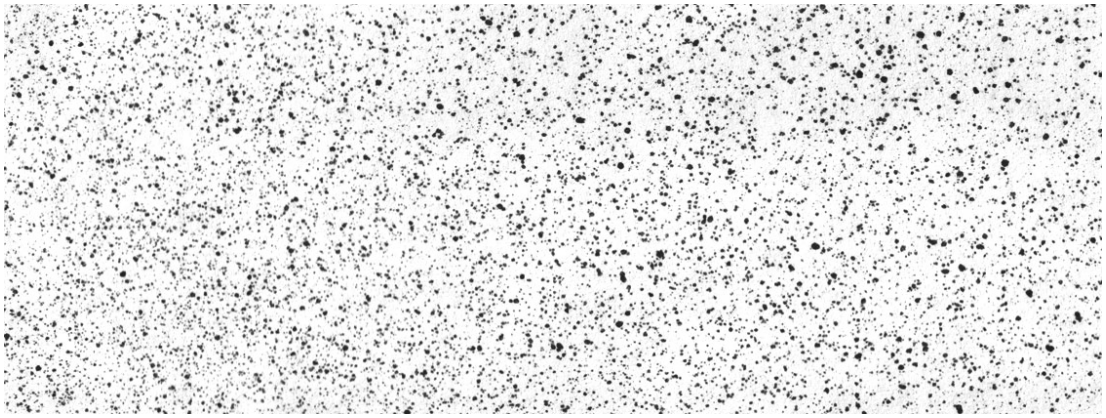

FIGURE 1.1 – *Analogie avec une texture visuelle : une texture.*

lement mais plutôt que l'on observe plutôt leur densité, ce sont les propriétés haut-niveau des textures qui sont perçues. Cela a pour conséquence directe qu'une portion de ce que nous appellerions couramment "texture sonore" n'en est pas, selon cette définition. Si l'on prends l'exemple d'une texture sonore de feu les craquements du feu sont perçus individuellement, ce qui sort donc du cadre de notre définition alors que nous parlerions assez naturellement toujours d'une texture : cette limitation de notre définition sera abordée à nouveau plus loin dans le rapport.

1.2 Usages & applications

La synthèse de ces textures sonores reste un domaine de recherche modérément exploré, surtout en comparaison avec des champs tels que la synthèse de voix humaine, mais est néanmoins très demandée par l'industrie : le lien que nous évoquions entre environnement et texture fait que la présence de cette dernière renforce l'immersion et la crédibilité d'un environnement. Cette effet est fréquemment recherché : qu'il s'agisse d'un film, d'un jeu vidéo (notamment avec l'essor récent de la réalité virtuelle) ou de n'importe quel type d'œuvre interactive, les textures tiennent une place de choix dans le design sonore. La synthèse de texture peut bien entendu aussi être utilisée librement à des fins musicales.

Chapitre 2

État de l'art

Avant d'aborder la méthode de synthèse particulière qu'est la synthèse par imposition de statistiques, nous commencerons par présenter brièvement deux des méthodes de synthèse de textures sonores les plus populaires. Nous tenterons ensuite de tirer de cet état de l'art les objectifs que se doit de viser un synthétiseur de texture afin de surpasser les méthodes existantes.

2.1 Synthèse par modèle physique

Ce type de synthèse repose sur la mise en place d'un modèle physique du phénomène à l'origine de la texture sonore, puis sur sa résolution numérique.

La simulation physique en question dépend entièrement du phénomène étudié et peut donc aller de la simulation hydrodynamique à celle d'un choc de solide, comme on peut en trouver dans l'article [O'Brien et al., 2002]. Elle peut aussi être hybridée avec d'autres méthodes de synthèse, plus empiriques, afin de leur conférer un certain réalisme tout en évitant une partie de la lourdeur des calculs de modélisation, souvent très coûteux.

En permettant la restitution du comportement de la source et en synthétisant la texture résultante, il devient alors possible d'avoir accès à des paramètres physiques très intéressants : dans le cas d'une texture de pluie, on pourra alors contrôler la fréquence de chute de gouttes d'eau, les propriétés du matériau sur lequel elles tombent, etc. Cependant, l'implication directe de cette méthode est que chaque synthétiseur sera lié à un unique modèle physique, et donc à un unique type de texture sonore.

La synthèse par modèle physique est donc une méthode réaliste et contrôlable à haut-niveau, mais à la fois relativement lourde et peu flexible.

2.2 Synthèse granulaire

Une autre méthode envisageable lors de la synthèse de texture sonore est la synthèse granulaire.

Il s'agit ici en réalité d'une méthode d'analyse-synthèse, contrairement à la synthèse physique. En partant d'une texture "stable", c'est-à-dire dont les caractéristiques perceptives n'évoluent pas au cours du temps (par exemple, une pluie ne variant pas en intensité), ou bien d'un corpus de textures (voir [Schwarz, 2007]), on découpe puis extrait dans un premier temps un ensemble de fragments, appelés grains, de taille de l'ordre de la dizaine de millisecondes.

La synthèse s'effectue alors choisissant l'un de ces grains comme point de départ, puis en se déplaçant dans cet ensemble de grains au cours du temps et en les concaténant (avec ou sans recouvrement) les uns après les autres. Un point crucial et central de cette méthode est la sélection du grain suivant : cela peut notamment être effectué en établissant des probabilités de transitions puis en suivant ces lois à chaque changement de grain.

Cette méthode de synthèse a pour grand avantage d'être légère et donc implémentable facilement en temps-réel, mais au contraire de la méthode précédente ne possède pas de moyen de contrôle des paramètres haut-niveau de la texture. Les seuls auxquels nous ayant accès sont en rapport directs avec les grains : leur taille, la sélection du grain suivant, etc. Autant de paramètres n'ayant pas de signification directe pour un utilisateur du synthétiseur.

2.3 Objectifs d'une méthode de synthèse de texture

Dans le but de clarifier les objectifs de ce stage, essayons de maintenant d'établir les points phares qu'une méthode de synthèse se doit de viser.

S'il parait au premier abord évident que le *réalisme* doit être son objectif principal, il apparaît aussi de l'état de l'art précédent qu'il ne s'agit pas de l'unique point sur lequel une nouvelle méthode de synthèse devrait se concentrer.

Les deux objectifs secondaires sont en lien directs avec l'utilisation que pourrait avoir un synthétiseur de texture. Afin de pouvoir être utilisé créativement celui-ci se doit de pouvoir être utilisé dans un éventail de situations le plus large possible, donc d'être *flexible*, et de permettre une manipulation de la texture au moyens de paramètres compréhensibles par l'utilisateur, c'est-à-dire *contrôlable à haut-niveau.*

En plus de ces objectifs pourrait s'ajouter celui d'une légèreté de calculs, ou *efficacité* : il ne s'agit cependant pas d'une priorité du fait de l'évolution constante (tendant cependant à s'affaiblir) des puissances de calcul disponible, mais aussi parce que trop se concentrer sur ce point pourrait nous couper prématurément des idées et possibilités intéressantes sur le plus long terme.

Chapitre 3

Synthèse par imposition : algorithme de départ

Maintenant que le paysage de la synthèse de texture a été décrit et que les objectifs qu'une nouvelle méthode se devrait de viser ont été mis au clair, commençons à nous pencher sur celle qui est à la base de ce stage : la synthèse par imposition de statistiques temps-fréquence. Ce chapitre a donc pour but de présenter l'état de l'algorithme avant que je ne commence à travailler dessus.

3.1 Principe général de fonctionnement

L'idée à la base de l'algorithme a été avancée et étudiée par Josh McDermott dans [McDermott and Simoncelli, 2011] : plus axée sur l'aspect psycho-accoustique de la perception des textures sonores par l'oreille humaine, sa recherche aboutit à la conclusion que la reconnaissance d'une texture se fait uniquement à partir des statistiques des bandes critiques de l'appareil auditif (qui sont les enveloppes des signaux extraits par le banc de filtre de l'oreille interne) et de leur sous-bandes en fréquences, aussi appelés bandes de modulation. Ces statistiques sont plus précisément la moyenne, variance, asymétrie, kurtosis (soit les moments d'ordre 1 à 4, voir Annexe A pour un rappel des définitions de ces statistiques), et les coefficients de corrélation entre bandes.

L'algorithme qu'a développé Wei-Hsiang Liao au cours de sa thèse ([Liao, 2015]) sous la direction d'Axel Roebel s'appuie sur les travaux de McDermott en cherchant à imposer à un bruit blanc les statistiques de la texture cible afin que le son généré soit identifié comme appartenant à la même catégorie de textures que celle-ci. Liao a cependant fait le choix, principalement pour des raisons de complexité de calculs, de ne pas pousser sa description sonore jusqu'aux bandes de modulations : seules sont prises en compte les enveloppes des bandes critiques. De plus, il substitue aux coefficients de corrélations les fonctions d'auto et d'inter-corrélations.

Cette méthode introduit donc un nouveau paradigme de synthèse sonore : le but est de décrire un son à partir de ses données statistiques puis de chercher à créer un autre son possédant ces mêmes paramètres, mais cette méthode de synthèse ne

se limite pas forcément à une utilisation pour les textures (bien qu'elle semble ici tout indiquée), et pourrait être développée afin d'être utilisée pour un panel de sons plus large. Enfin, il est important de noter qu'il ne s'agit par conséquent pas d'une méthode de synthèse pure mais d'une méthode d'analyse-synthèse : le but est ici de recréer une texture non identique à celle de départ, mais identifiée comme étant du même type. L'algorithme dans son ensemble est résumé par la figure 3.1 : dans les

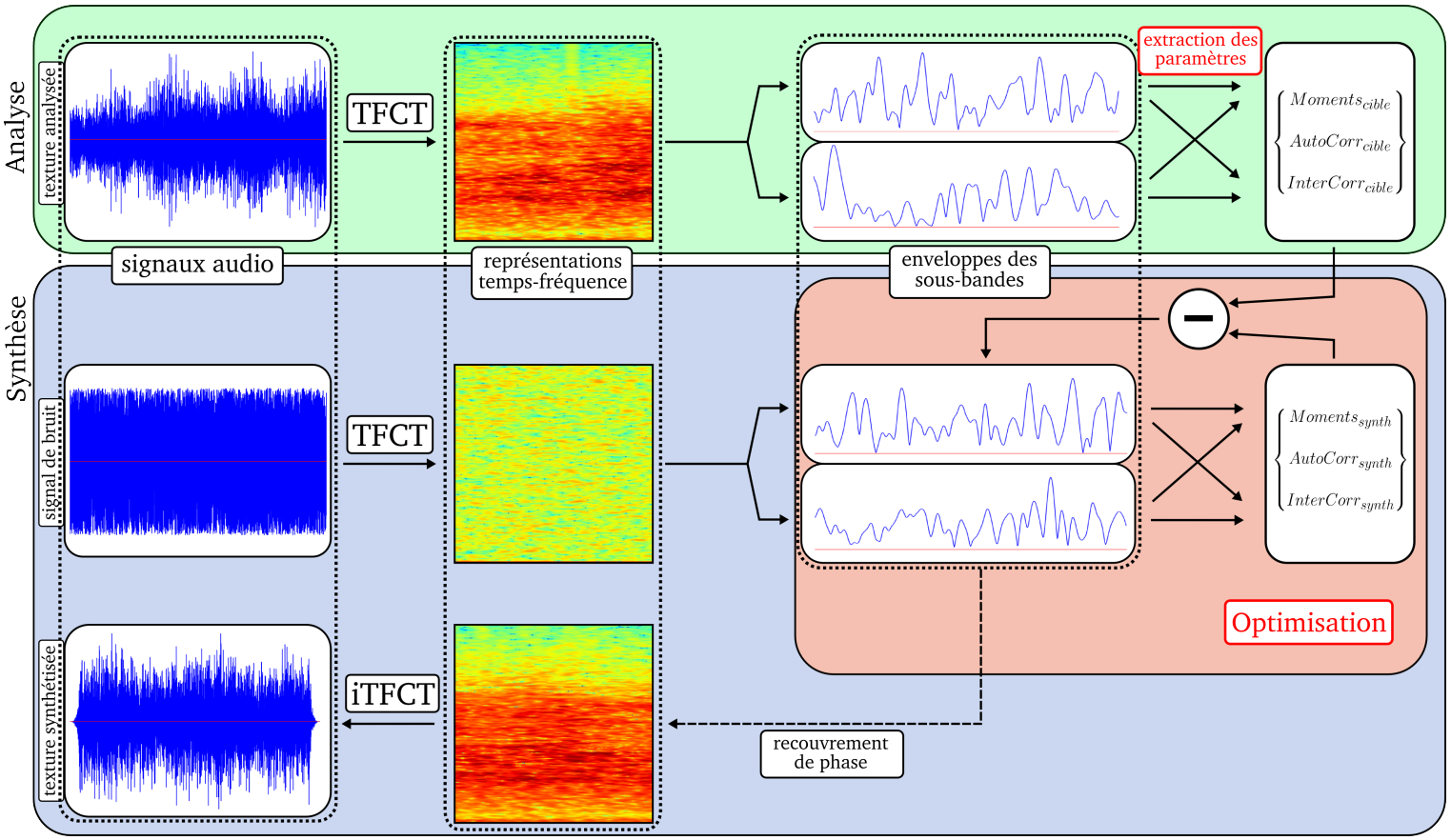

FIGURE 3.1 – *Structure de l'algorithme de synthèse par imposition : la partie analyse est en vert, la partie synthèse en bleu.*

parties suivantes nous entrerons plus en détail dans le fonctionnement de chacun de ses éléments.

3.2 Détail des étapes de l'algorithme

3.2.1 Extraction des statistiques cibles

Commençons par la partie analyse de l'algorithme : il s'agit ici d'extraire de la texture de départ les statistiques mentionnées plus haut, à savoir les moments, auto-corrélations (AC) et inter-corrélation (IC) des bandes critiques (correspondant

au bloc vert de la figure 3.1).

Le but initial est d'imiter l'extraction d'enveloppe des sous-bandes effectuée par l'oreille humaine interne. Le passage d'un signal audio à une représentation temps-fréquence devrait donc dans l'idéal se faire au moyen d'un découpage logarithmique de l'axe des fréquences tel qu'une transformée à Q-constant le permettrait : cependant certaines partie de l'algorithme seraient alors rendues très complexes du fait d'un manque de documentation sur le sujet, notamment la reconstitution de phase (voir 3.2.3). Il est plus aisé d'utiliser une transformée de Fourier à court-terme (TFCT), plus documentée concernant cette reconstitution, et c'est donc pour cette transformation que Liao a opté. L'extraction des enveloppes est pour le moment remplacée par l'utilisation de l'amplitude des sous-bandes.

Les moments et corrélations sont ensuite calculés à partir des signaux obtenus. Il s'agit maintenant d'imposer ces statistiques cibles à un signal de départ : s'il s'agit habituellement d'un bruit blanc du fait de la richesse de son contenu sonore, nous discuterons plus tard de la possibilité de partir d'une autre base.

3.2.2 Impositions des statistiques

Au sein de sa thèse, Liao présente deux méthodes d'imposition des statistiques cibles sur le bruit.

- **Imposition totale**

La première, que nous appellerons *imposition totale*, repose sur l'utilisation du théorème de Wiener-Khinchin, énoncé ainsi :

$$\begin{aligned} \mathcal{F}\{\mathcal{AC}_x\} &= |\widehat{x}|^2 \\ \mathcal{F}\{\mathcal{IC}_{xy}\} &= |\widehat{x}||\widehat{y}| \exp\left(i(\arg(\widehat{x}) - \arg(\widehat{y}))\right) \end{aligned} \tag{3.1}$$

Avec $\mathcal{F}$ la transformée de Fourier (TF), x et y deux signaux, $\widehat{x}$ et $\widehat{y}$ leurs TF, et $\mathcal{AC}$ et $\mathcal{IC}$ les auto et inter-corrélations : cela se traduit par le fait que l'auto-corrélation d'un signal fixe le module de la TF de celui-ci, et qu'une fois le module fixé c'est en modifiant sa phase qu'on peut lui imposer les inter-corrélations désirées.

Il est donc plus aisé de manipuler non pas directement les amplitudes des sous-bandes en fréquence mais leur transformée de Fourier : prenons deux amplitudes de sous-bandes, E_i et E_j. Dans un premier temps, nous cherchons à leur imposer les auto-corrélations cibles $\tilde{\mathcal{A}}_i$ et $\tilde{\mathcal{A}}_j$. D'après le théorème de Wiener-Khinchin, cela est immédiatement possible en remplaçant le module de $\widehat{E}_i$ et $\widehat{E}_j$ par :

$$\begin{aligned} |\widehat{E}_i| &\leftarrow \sqrt{\mathcal{F}\{\tilde{\mathcal{A}}_i\}} \\ |\widehat{E}_j| &\leftarrow \sqrt{\mathcal{F}\{\tilde{\mathcal{A}}_j\}} \end{aligned} \tag{3.2}$$

Une fois les auto-corrélations (et donc les modules) fixés, il est ensuite possible d'imposer l'inter-corrélation cible $\tilde{\mathcal{I}}_{ij}$ en manipulant la phase de $\widehat{E}_i$ ainsi :

$$\theta(\widehat{E_i}) \leftarrow \theta(\widehat{E_j}) + \theta\left(\frac{\mathcal{F}\{\tilde{\mathcal{I}}_{ij}\}}{\sqrt{\mathcal{F}\{\tilde{\mathcal{A}}_i\}}\sqrt{\mathcal{F}\{\tilde{\mathcal{A}}_j\}}}\right) \tag{3.3}$$

Cette méthode a pour grand avantage de ne pas être itérative, et donc très rapide à exécuter. Ceci étant dit, le seul degrés de liberté qu'il nous reste avec lequel travailler pour ne pas ruiner le travail effectué sur les corrélations se trouve dans la phase de $\widehat{E}_j$, ce qui re-transposé à notre système dans son ensemble revient à dire que nous ne pouvons que modifier la phase de la transformée d'une seule bande (arbitrairement choisie comme étant la première) sans gêner les impositions précédentes.

Puisque la moyenne de E_i est égale à $\widehat{E}_i[0]$ et que sa variance se retrouve dans la valeur en 0 de son auto-corrélation, les deux premiers moments sont imposés en même temps que l'auto-corrélation cible.

Pour imposer les deux moments suivants il est alors nécessaire d'avoir recours à une optimisation, ici une descente de gradient conjuguée sur une fonction d'erreur définie ainsi :

$$\mathcal{E}_{skew+kurt}(E) = |\eta(E) - \tilde{\eta}|^2 + |\kappa(E) - \tilde{\kappa}|^2 \tag{3.4}$$

Avec E la matrice de TFCT du signal en train d'être modelé, $\mu(E)$ et $\kappa(E)$ les vecteurs contenant les asymétries et kurtosis des amplitudes de ses sous-bandes, et $\tilde{\eta}$ et $\tilde{\kappa}$ les statistiques cibles correspondants.

Puisque notre seul degrés de liberté est le vecteur de phase de la première bande, que l'on nommera θ_0, il est nécessaire de calculer $\frac{\partial f(E)}{\partial \theta_0}$ et donc $\frac{\partial \eta(E)}{\partial \theta_0}$ et $\frac{\partial \kappa(E)}{\partial \theta_0}$. On obtient (cf [Liao, 2015]) :

$$\begin{aligned} \frac{\partial \eta(E)}{\partial \theta_0} &\propto \Im\left(\overline{\mathcal{IC}}_{\mathcal{A}_{\widehat{E}}\widehat{E}} \circ \widehat{E}\right) \\ \frac{\partial \kappa(E)}{\partial \theta_0} &\propto \Im\left(\overline{\mathcal{AC}}_{\widehat{E}} \circ \widehat{E}\right) \end{aligned} \tag{3.5}$$

Avec $\overline{X}$ dénotant le conjugué de X, $\widehat{E}$ la TF par bande de E (comprendre que chaque bande de $\widehat{E}$ est la TF de la bande correspondante de E), et $\circ$ le produit terme à terme. θ_0 est ainsi itérativement modifié jusqu'à ce que notre fonction d'erreur $\mathcal{E}_{skew+kurt}$ soit aussi faible que nous le désirons, et que nous soyons suffisamment proche de nos cibles.

Cette méthode impose donc dans un premier temps de façon parfaite les corrélations en manipulant les modules des TF des amplitudes des sous-bandes puis leur phase à un vecteur près : ce vecteur est ensuite fixé par descente de gradient afin de minimiser l'écart avec les moments d'ordre 3 et 4 cibles.

Cette imposition, que l'on retrouve dans [Liao et al., 2013], n'a cependant pas abouti à des résultats parfaitement convaincant puisqu'après reconstruction (partie 3.2.3) du signal audio il s'est avéré que la texture résultante n'était principalement qu'une version décalée dans le temps de la texture d'origine . Cela pourrait s'expliquer par le fait que nous ne laissons que très peu de degrés de liberté à la texture synthétisée : lorsque nous passons à l'étape d'optimisation, nous faisons donc partir notre descente de gradient d'un point très proche de la texture d'origine. Puisque nous faisons en sorte que θ_0 ne soit pas un vecteur nul (ce qui correspondrait à autoriser la texture de sortie a être une copie de la texture cible), les minimum locaux atteignables proche du point de départ sont donc possiblement ces versions décalées dans le temps du signal d'origine.

- **Imposition partielle**

C'est pour cette raison que Liao passe à une autre méthode d'imposition, dite *partielle* : l'idée étant de donner plus de liberté au signal synthétisé, donc de ne pas imposer parfaitement les statistiques cibles, il faut décider de la portion que l'on rejettera. Son choix est de conserver l'imposition totale des auto-corrélations, mais de n'imposer que les inter-corrélations à court-terme (moins de 2 secondes) en argumentant que les informations qu'elles contiennent sur le long-terme sont sûrement liées à une organisation spécifique à la texture analysée.

La méthode précédente ne peut donc plus être utilisée pour les IC : en effet, si le théorème de Wiener-Khinchin permettant une imposition des inter-corrélations, il n'était pas possible de nuancer celles-ci. Nous en sommes donc réduit à utiliser ici aussi une optimisation par itération comme nous en utilisions pour l'imposition des derniers moments.

L'idée va donc être de fenêtrer non pas les corrélation cibles, ce qui fixerait arbitrairement une partie de celles-ci à 0 mais la fonction d'erreur. À titre d'exemple, la fonction d'erreur des inter-corrélations liées à la k-ième bande ressemblera à :

$$\mathcal{E}_{\mathcal{IC}_k}(E) = \sum_{i \in bandes \setminus k} \left|\left| w \circ \left(\mathcal{I}_{ki} - \widetilde{\mathcal{I}}_{ki} \right) \right|\right|^2 \tag{3.6}$$

Avec w le vecteur de fenêtrage (dans notre cas il s'agit d'une fenêtre de Tukey), $\tilde{\mathcal{I}}_{ki}$ le vecteur d'inter-corrélation cible pour le couple des bandes k et i, $\mathcal{I}_{ki}$ une contraction de $\mathcal{IC}_{E_k,E_i}$, l'inter-corrélation des bandes k et i du signal synthétisé, et $||x||$ la norme 2 de x, soit $\sqrt{\sum_i x[i]^2}$.

Un aperçu post-optimisation est visible en figure 3.2, où l'on observe bien que la fonction d'inter-corrélation d'un couple de bandes de notre signal synthétisé correspond bien à celle du signal ciblé, mais uniquement sur la partie délimitée par la fenêtre en pointillés.

En plus de cela, Liao fait aussi le choix de ne pas prendre tous les couples d'inter-corrélations possibles en compte : pour chaque bande, il ne compte que

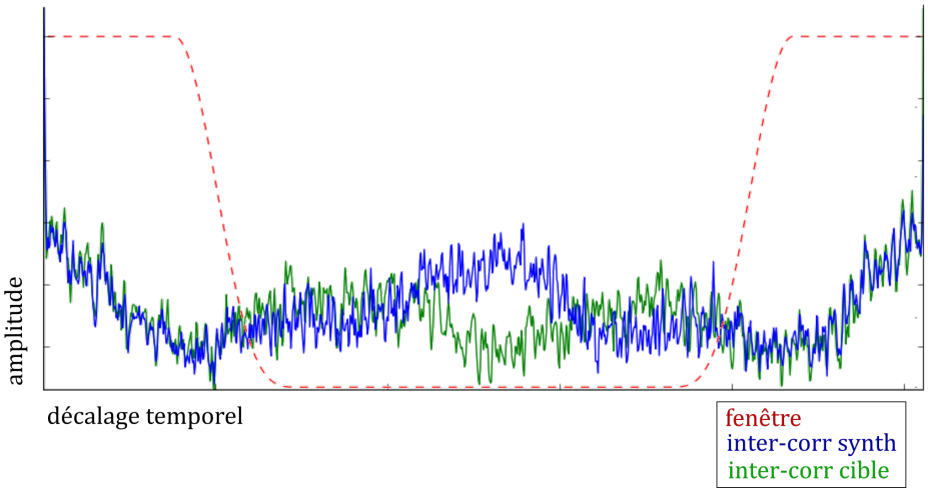

FIGURE 3.2 – *Imposition partielle d'une fonction d'inter-corrélation : la fonction cible est en vert, celle du signal synthétisé après descente de gradient en bleu, tandis que la fenêtre est représentée sans soucis d'échelle en pointillés.*

les IC avec les bandes distantes d'un multiple de 3 ainsi qu'avec les deux bandes voisines dans la fonction d'erreur $\mathcal{E}_{\mathcal{IC}_k}(E)$. Cette écrémage arbitraire est aussi fait pour une raison de praticité : si la méthode précédente avait pour avantage de ne nécessiter qu'une mise à jour du bruit initial pour les auto-corrélations et une pour les inter-corrélations, ce n'est plus le cas dorénavant. De plus, les IC sont imposée successivement en parcourant chaque bande de de la TFCT plusieurs fois : en effet en modifiant une bande on modifie toutes les inter-corrélations lui étant liée, obligeant à parcourir la TFCT plusieurs fois. Cette méthode est donc beaucoup plus lourde que la précédente.

Une fois les inter-corrélations partiellement imposées, les moments d'ordre 3 et 4 sont imposés comme avec la méthode d'imposition totale (les premiers moments étant toujours imposé avec l'AC) : puisque les IC ne dépendent que des différences entre phases, l'imposition des derniers moments se fait en déphasant les phases de chaque bande de la même manière.

Cette étape correspond sur la figure 3.1 au bloc rouge. Une fois réalisée, nous nous retrouvons donc avec le module d'une TFCT d'un signal synthétisé, qu'il y a donc besoin de ramener à l'état d'un signal audio.

3.2.3 Reconstruction du signal synthétisé

Pour se faire, il est donc nécessaire de restituer le phase d'une TFCT à partir de son amplitude, que nous nommerons toujours E : pour se faire, et à partir du moment ou le recouvrement utilisé est d'au moins 50%, il existe des méthodes de restitutions adaptée. La méthode pour laquelle Wei-Hsiang Liao opte est celle de Leroux ([Le Roux et al., 2010]). Celle-ci part de la méthode de Griffin-Lim, qui consiste à donner une phase initiale θ à E, puis à la mettre à jour ainsi :

$$\theta \leftarrow \arg\left(STFT\left(STFT^{-1}\left(E \circ \exp(j\theta)\right)\right)\right) \tag{3.7}$$

Pour ensuite expliciter les TFCT et TFCT inverse et réduire les sommations qu'elles impliquer à des sommes sur les termes proches. La méthode de Leroux est donc plus rapide en terme de temps de calculs que celle plus classique de Griffin-Lim, sans pour autant mettre beaucoup plus d'itérations à arriver au même point.

Une fois la phase recouvrée, il ne reste plus qu'à inverser la TFCT de $E \circ \exp(j\theta)$ pour obtenir le signal synthétisé.

3.3 Bilan sur l'algorithme

Le résultat de cet algorithme est auditivement relativement convaincant et n'est effectivement pas restreint à un seul type de texture, mais présente principalement les problèmes suivants :

- Les événements brefs et couvrant une large gamme de fréquences sont mal reproduits (par exemple les crépitements d'un feu).
- Aucun contrôle n'est possible sur la texture synthétisée.
- La durée de la fenêtre est choisie arbitrairement, et nécessite plusieurs tests pour chaque textures : cela nuit à la flexibilité de l'algorithme.
- La sélection des couples d'inter-corrélations est elle-aussi arbitraire, et pose les mêmes problèmes.

Ces deux derniers point témoignent d'une difficulté de cet algorithme : l'équilibrage. Si l'imposition est totale, en sélectionnant la totalité des fonctions d'auto-corrélations et l'ensemble des couples d'inter-corrélation (comme dans l'imposition totale de la partie 3.2.2) on se retrouve face à une simple copie du signal d'origine. Au contraire, si l'on choisit trop peu de couples et une fenêtre trop petite le signal résultant est extrêmement bruité.

C'est cet équilibrage, et donc la volonté d'améliorer la flexibilité de l'algorithme, qui a été au centre de mon stage.

Chapitre 4

Travail effectué

Ce stage a donc été effectué dans l'idée de poursuivre le travail effectué par Liao au cours de sa thèse en cherchant à aboutir à une version de l'algorithme ne requérant pas de choix arbitraire de la part de l'utilisateur, notamment concernant celui des parties de corrélations à imposer. Commençons d'abord par clarifier les axes principaux de recherche.

4.1 Objectifs du stage

Dans cette volonté d'améliorer la flexibilité de l'algorithme, mon stage s'est centré sur les points suivants :

— L'implémentation en python de l'algorithme et le départ d'une collaboration avec Florian Hecker, compositeur.

— La sélection des couples d'inter-corrélations.

— L'automatisation de l'imposition des corrélations.

4.2 Implémentation de l'algorithme et collaboration avec Florian Hecker

Commençons avec l'implémentation de l'algorithme de départ.
Une fois familiarisé avec la thèse de [Liao, 2015] et le travail bibliographique effectué, j'ai décidé de ne pas me servir du code que Wei-Hsiang Liao avait écrit. Celui-ci était codé sous MATLAB et n'était visiblement pas destiné à un usager extérieur, car difficilement lisible : j'ai donc implémenté directement l'algorithme décrit dans la thèse sous Python.

Ce choix a aussi été motivé par le fait qu'une partie de l'équipe Analyse-Synthèse, dont Axel Roebel, se sert de ce langagen mais aussi par le fait que Python permet la création de logiciels distribuables ne nécessitant pas l'installation de librairies externes, comme il serait le cas avec MATLAB.

L'ensemble de la thèse a été ainsi implémenté, y compris la version totale et la version partielle de l'imposition des caractéristiques. Par la suite, un certain temps a été consacré à l'optimisation du code en lui-même et à sa mise en forme, de sorte à le clarifier et le rendre facilement utilisable.

Bien que nous ayons mentionné plus haut le fait que l'optimisation ne devait pas être une priorité, elle s'est révélée nécessaire pour de basses raisons pratiques : il est compliqué d'avoir des retours sur l'algorithme si celui-ci mets des journées à calculer une seconde de son. C'est donc pour cela que j'ai travaillé à faire en sorte qu'une dizaine de secondes textures soient synthétisables en l'espace d'une heure.

Pour ce faire, il a été indispensable d'utiliser les librairies numpy et scipy qui permettent une vectorisation du code et des calculs matriciels (de façon similaire à ce que MATLAB permets). Étant donné que la majeur partie du temps de calcul était due aux descentes de gradient, processus non vectorisable et nécessitant l'utilisation de boucles logiques, relativement lentes avec Python, il m'a été nécessaire d'utiliser cython, une librairie permettant l'interface entre Python et du code écrit sous C : en utilisant le code C de descente de gradient conjuguée il a donc été possible d'accélérer drastiquement les temps de calculs.

Ce stage a aussi été l'occasion de m'occuper de la résidence à l'IRCAM de Florian Hecker, compositeur contemporain très intéressé par notre algorithme de synthèse sonore. Plus qu'une simple utilisation de cet algorithme pour la synthèse de texture, c'est l'ensemble des possibilités créatives que permet la synthèse par imposition de statistiques qui intéressait principalement le compositeur. Parmi ces possibilités, nous avons notamment éclaircit plusieurs utilisations intéressantes de l'algorithme, notamment :

— La possibilité d'hybrider deux textures, ou de façon plus générale deux sons, en se servant de la première comme cible et de la deuxième comme base, remplaçant le bruit blanc : l'objectif serait ensuite de parvenir à s'arrêter au cours de l'imposition des statistiques pour obtenir un son à "mi-chemin" entre les deux.

— La possibilité d'utiliser comme texture cible un son qui n'est pas à proprement parler une texture : un échantillon de voix, une pièce musicale, etc... Ceci est facilement faisable dans l'état actuel des choses.

— La possibilité de mélanger les statistiques cibles de plusieurs textures en entrée de l'algorithme d'imposition.

Ces souhaits et les discussions sont de plus l'occasion d'avoir certain recul par rapport au travail que nous poursuivons et nous donnent une certaine vision des fonctionnalités attendues par un utilisateur musicien de la part de notre synthétiseur. Abordons maintenant la partie à proprement parler scientifique de ce stage.

4.3 Sélection des couples de sous-bandes

La première étape de mon travail a été centrée sur une meilleure sélection des couples d'inter-corrélations imposés : il s'agit maintenant, au cours de la phase d'analyse, de créer pour chaque bande un score témoignant de la force de sa corrélation avec chacune des autres.

L'objectif étant de créer un score ne prenant en considération que la forme de l'amplitude des bandes et en aucun cas les valeurs prises, j'ai choisi de travailler sur des signaux standardisés (de moyenne nulle et de variance unitaire). Le score entre deux bandes E_m et E_n a donc été défini ainsi :

$$\alpha(m,n) = \max_i \left(|C_{E_m^* E_n^*}[i]| \right) \qquad\qquad E_k^* = \frac{E_k - \mu(E_k)}{\sigma(E_k)}$$

Avec μ la moyenne et σ l'écart-type de la bande. Ce score a de plus la propriété d'être systématiquement compris entre 0 et 1 (le plus grand score de similarité qu'une bande standardisée puisse obtenir étant avec elle-même, auquel cas celui-ci est égal à sa variance, ici 1), permettant d'établir des seuils valables pour toutes les bandes.

Une fois ce score établi, ses usages sont multiples. Il est tout d'abord possible de discriminer les bandes bruitées ou inutiles du signal cible en décidant de rejeter les bandes n'ayant que des scores faibles avec les autres, voire en se contentant de supprimer celles avec de faibles scores vis-à-vis de leurs deux bandes voisines. Pour le moment une valeur en score inférieure à 0.5 semble indiquer une "non-corrélation" entre deux bandes et nous permet de de déterminer les bandes bruitées : ce seuil a été établi en observant que les scores de corrélation entre les bandes d'un signal bruit ne dépassait jamais cette valeur, mais il faudra cependant attendre la mise en place de véritables tests perceptifs avant de trancher définitivement pour un seuil (ce problème revient d'ailleurs assez fréquemment, voir partie 5.1).

Une fois ces bandes bruitées repérées, il devient possible de les écarter du reste de l'imposition des corrélations : cela permet une certaine économie en temps de calcul et en ressources, précieux pour l'instant. Afin que ces bandes restent pour le signal synthétisé dans le même ordre de grandeur que celles du signal cibles, leurs moyennes est variance sont toutefois sommairement imposées ainsi :

$$E_{bruit} \leftarrow \frac{E_{bruit} - \mu(E_{bruit})}{\sigma(E_{bruit})} \times \widetilde{\sigma} + \widetilde{\mu} \tag{4.1}$$

Le score de corrélation peut ensuite être utilisé afin de discriminer les relations entre bandes. Pour le moment nous choisissons pour chacune d'entre elles un nombre fixé de bandes qui lui sont le plus corrélé, ce nombre étant choisi principalement pour des raisons de temps de calcul, mais il est souhaitable qu'à la suite de tests auditifs nous établissions un seuil au dessus duquel nous prendrons en compte le couple de bandes.

Enfin, il n'est pas encore possible de tester l'efficacité de cette sélection tant qu'un test auditif fiable (cf 5.1) n'aura pas été mis en place.

4.4 Amélioration de l'imposition

Il s'agit maintenant d'améliorer la façon dont les corrélations sont imposées : pour rappel, elles étaient jusque là imposées partiellement au moyen d'un fenêtrage de la fonction d'erreur. Notre première idée a donc été de poursuivre sur cette lancée.

4.4.1 Imposition partielle de l'auto-corrélation

Afin d'avoir la possibilité d'augmenter les degrés de liberté de notre système lors de l'imposition, j'ai dans un premier temps poursuivi le travail de Wei-Hsiang Liao en passant d'une imposition totale de l'auto-corrélation à une imposition fenêtrée, comme il était fait pour les inter-corrélations.

Sachant que l'auto-corrélation d'un signal ne dépends que du module de sa TF nous avons décidé de garder une imposition des IC et des moments 3 et 4 par manipulation de la phase, tout en rajoutant une imposition des auto-corrélations et moments 1 et 2 par manipulation du module (en effet, la moyenne et la variance ne sont plus systématiquement imposés avec l'AC si l'imposition n'est pas totale). La fonction d'erreur liée à l'auto-corrélation se présente donc ainsi :

$$\mathcal{E}_{\mathcal{A}}\left(\widehat{E}\right) = \sum_k \left|\left| w\left(\mathcal{A}_k - \widetilde{\mathcal{A}}_k\right)\right|\right|^2 = \sum_k \left|\left| w\left(\mathcal{F}^{-1}\left(\left|\widehat{E}_k\right|^2\right) - \widetilde{\mathcal{A}}_k\right)\right|\right|^2 \tag{4.2}$$

Avec w le vecteur de fenêtrage, $\mathcal{A}_k$ une contraction de $\mathcal{AC}_{E_k}$ et $\widetilde{\mathcal{A}}_k$ l'auto-corrélation cible. Cette fonction a pour dérivées partielles :

$$\frac{\partial \mathcal{E}_{\mathcal{A}}}{\partial \left|\widehat{E}_k\right|[i]} = \frac{8}{N}\left|\widehat{E}_k\right|[i] \cdot \Re\left[\mathcal{F}\left(w^2\left[\mathcal{F}^{-1}\left(\left|\widehat{E}_k\right|^2\right) - \widetilde{\mathcal{A}}_k\right]\right)[i]\right] \tag{4.3}$$

4.4.2 Impositions jointes

En plus de cela, il est aussi apparu que du fait de son imposition séquentielle des inter-corrélations par modification de phase puis des moments 3 et 4 par manipulation de l'offset des phases (cf 3.2.2), l'imposition des moments était trop contrainte et ne parvenait pas à réduire l'erreur sur les moments de façon conséquente. Nous avons donc opté pour une imposition jointe des IC et moments 3 et 4, et par la même occasion des AC et de moments 1 et 2. L'imposition se présente donc ainsi :

— Descente de gradient sur le module de la TF des modules des sous-bandes en fréquence de notre signal afin d'imposer les auto-corrélations fenêtrées ainsi que les moyennes et variances. La fonction d'erreur se présente donc ainsi :

$$\mathcal{E}_{\mathcal{AC}+\{1+2\}} = \mathcal{E}_{\mathcal{A}} + \beta \cdot (\mathcal{E}_{moy} + \mathcal{E}_{var}) \tag{4.4}$$

Avec $\mathcal{E}_{\mathcal{A}}$ l'erreur sur l'AC, $\mathcal{E}_{moy}$ celle sur les moyennes, $\mathcal{E}_{var}$ celle sur les variances, et β le coefficient de pondération entre les erreurs.

— Descente de gradient sur la phase de la TF des modules des sous-bandes en fréquence de notre signal afin d'imposer les inter-corrélations fenêtrées ainsi que les asymétries et kurtosis. La fonction d'erreur de chaque bande se présente donc sous la forme :

$$\mathcal{E}_{\mathcal{IC}+\{3+4\}} = \mathcal{E}_{\mathcal{I}} + \gamma \cdot (\mathcal{E}_{asym} + \mathcal{E}_{kurt}) \tag{4.5}$$

Avec $\mathcal{E}_{\mathcal{I}}$ l'erreur sur l'IC, $\mathcal{E}_{asym}$ celle sur les asymétries, $\mathcal{E}_{kurt}$ celle sur les kurtosis et γ le coefficient de pondération entre les erreurs.

Un problème persiste cependant : avec cette méthode, il nous est maintenant nécessaire de régler nous-même β et γ, ce qui va à l'encontre de notre volonté d'automatiser l'ensemble du processus. Nous développerons cette problématique en partie 4.4.4.

Comparaison chiffrée entre les deux méthodes d'imposition

Afin de comparer les impositions séquentielle et jointes, le tableau 4.1 compare les valeurs des fonction d'erreur des inter-corrélations, asymétries et kurtosis obtenues avec les deux méthodes sur 10 impositions indépendantes.

Les deux méthodes le même échantillon de bruit de grillons de 3 seconde à 22 kHz, et les mêmes réglages de STFT (fenêtre de 128 échantillons, recouvrement de 75%). Dans le cas séquentiel, nous avons réalisé 100 itérations d'optimisation sur l'inter-corrélation, puis 100 itérations sur les moments. Dans le cas joint, nous avons réalisé 100 itérations d'optimisation de la fonction 4.5 avec un paramètre $\gamma = 0.95$. Ce processus a été répété de façon indépendante 10 fois afin de s'affranchir d'éventuels hasards lors des initialisations.

On observe de façon assez visible que si les inter-corrélations semblent converger à même distance de l'objectif, les moments 3 et 4 bénéficient beaucoup des degrés de liberté supplémentaires apportés par l'imposition jointe.

TABLE 4.1 – Comparaison des erreurs sur l'inter-corrélation, l'asymétrie et le kurtosis entre imposition séquentielle et jointe sur une texture "bruit d'insectes"

	Séquentielle	Jointe
InterCorr	$2.62 \cdot 10^{14} \pm 1.06 \cdot 10^{9}$	$2.62 \cdot 10^{14} \pm 2.26 \cdot 10^{9}$
Asymétrie	713 ± 366	28.5 ± 1.28
Kurtosis	$1.17 \cdot 10^{4} \pm 331$	$1.48 \cdot 10^{3} \pm 223$

4.4.3 Raffinement du fenêtrage

Si le choix d'une fenêtre de 2 secondes, comme Liao la choisissait, peut sembler trop arbitraire, c'est en partie parce que cela implique de simplement garder

les corrélations proches et d'ignorer totalement les corrélations lointaines sans justifications approfondie : la première piste que nous avons emprunté a donc été de rendre ce fenêtrage automatique et non simplement basé sur une taille de fenêtre choisie par l'utilisateur.

Nous avons ainsi cherché à extraire durant la phase d'analyse un découpage de chaque fonction de corrélation cible indiquant les portions à imposer. Ce découpage pourrait s'obtenir en fixant un seuil à chaque fonction de corrélation des bandes standardisées : les plages de signal où les corrélations se trouvant au dessus du seuil seraient ensuite celles que le fenêtrage conserverait lors de l'optimisation de la fonction d'erreur.

Mais cette méthode suppose que nous sommes certains de l'inutilité des portions où les corrélations sont faibles : hors ce n'est pas le cas, rien n'infirme que celles-ci sont cruciales à l'authenticité de la texture générée. Nous avons donc abandonné cette méthode.

4.4.4 Utilisation des écarts-types des corrélations

La piste suivante, plus prometteuse, a été d'essayer d'imiter les variations naturelles des corrélations au sein d'une texture du même type.

Principe du critère d'arrêt

Dans l'idéal, il s'agirait d'extraire un très grand nombre de statistiques cibles de textures reconnues comme identiques (par exemple en segmentant une longue texture constante en de plus petits échantillons) puis de calculer la moyenne et la variances de ces statistiques. Prenons l'exemple de l'auto-corrélation d'une certaine bande k (le principe de cet exemple reste valables pour les IC et moments) : nommons sa moyenne sur l'ensemble des segments $\widetilde{\mathcal{A}}_k^{\mu}$ et son écart-type $\widetilde{\mathcal{A}}_k^{\sigma}$. $\widetilde{\mathcal{A}}_k^{\mu}$ pourrait alors être fixée comme valeur cible pour l'auto-corrélation de la bande k, tandis que $\widetilde{\mathcal{A}}_k^{\sigma}$ pourrait nous servir de critère d'arrêt à l'optimisation.

En effet, dès que l'AC de notre bande serait proche de son objectif d'une distance inférieure à l'écart-type, cela signifierait que nous sommes suffisamment proche de l'objectif pour que notre écart soit considéré comme "naturel". Dans l'idéal, nous nous retrouverions avec une situation telle que celle représentée sur la figure 4.1 : dès que la descente de gradient entraînerait la fonction d'auto-corrélation de la bande k à l'intérieur de la de l'espace compris entre $\widetilde{\mathcal{A}}_k^{\mu} - \widetilde{\mathcal{A}}_k^{\sigma}$ et $\widetilde{\mathcal{A}}_k^{\mu} + \widetilde{\mathcal{A}}_k^{\sigma}$, nous pourrions stopper la descente. Cette méthode pourrait de plus être appliquée aux moments pour les mêmes motivations : imiter les variations naturelles des statistiques.

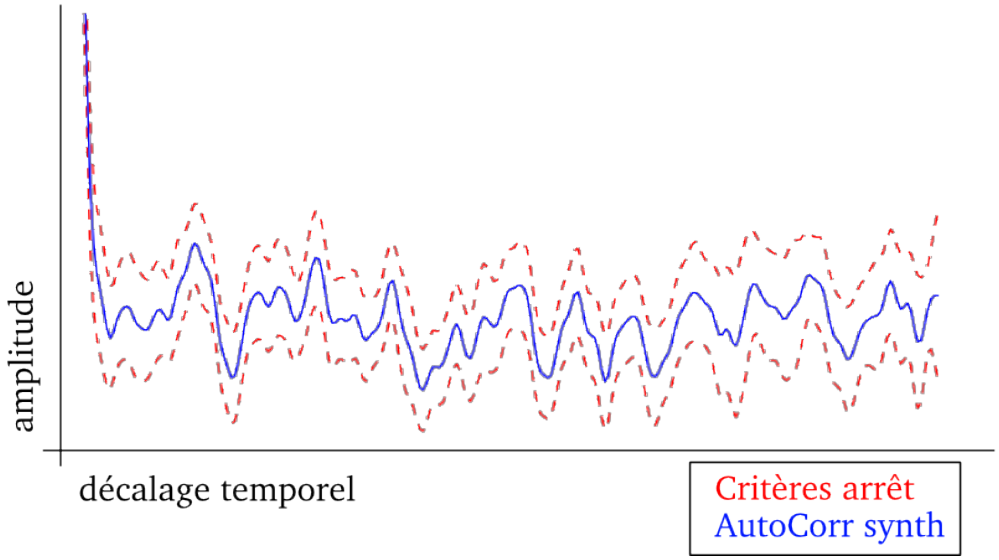

FIGURE 4.1 – *Imposition avec critère d'arrêt d'une fonction d'auto-corrélation : la fonction synthétisée est en bleue, les deux courbes pointillées correspondant à* $\widetilde{\mathcal{A}}_k^\mu - \widetilde{\mathcal{A}}_k^\sigma$ *et* $\widetilde{\mathcal{A}}_k^\mu + \widetilde{\mathcal{A}}_k^\sigma$.

Équilibrage de la fonction d'erreur

L'obtention des écarts-types de nos statistiques pourrait aussi nous permettre de mieux équilibrer les fonctions d'erreur des descentes de gradient, problème que nous mentionnions plus haut en partie 4.4.2 et qui n'était jusqu'ici réglé qu'en choisissant manuellement des coefficient de pondération à l'intérieur des fonctions d'erreur.

Ce problème pourrait en effet lui aussi être résolu par l'obtention d'un écart-type pour chacune des statistiques, qui permettraient de normaliser de façon intelligente nos erreurs : les zones où l'écart-type est faible (correspondant à des portions de nos statistiques variant peu à entre différentes textures du même type) se verraient alors appuyées, tandis que celles où l'écart-type est grand (correspondant à des portions de nos statistiques variant beaucoup entre différentes textures du même type) ne seraient pas priorisées.

Utilisation de la distance d'Itakura-Saito

Considérant notre volonté de cesser les mises-à-jour sur la bande une fois passé sous les écarts-types de nos statistiques cibles, qui nous servent alors d'erreurs cibles, et de notre volonté de pondérer nos fonctions d'erreur en fonction de ces erreurs cibles, nous avons opté pour l'utilisation de la distance d'Itakura-Saito.

Celle-ci peut-être définie ainsi :

$$IS(\epsilon, \epsilon_0) = \frac{\epsilon}{\epsilon_0} - \log(\frac{\epsilon}{\epsilon_0}) - 1 \tag{4.6}$$

Et dans le cas où ϵ dépend d'une variable x, cette distance se dérive aisément :

$$\frac{\partial IS(\epsilon(x), \epsilon_0)}{\partial x} = \frac{\epsilon(x) - \epsilon_0}{\epsilon(x) \cdot \epsilon_0} \cdot \frac{\partial \epsilon(x)}{\partial x} \tag{4.7}$$

Pour illustrer son utilisation, soit $\mathcal{E}_{stat}$ l'erreur entre l'une de nos statistiques et sa valeur cible. Jusqu'ici, $\mathcal{E}_{stat}$ était réduite à l'aide d'une optimisation sur une fonction d'erreur global telle qu'en 4.5, pondérée manuellement.

Cependant, si l'on se défait de cette pondération et que l'on remplace au sein de 4.5 $\mathcal{E}_{stat}$ par IS($\mathcal{E}_{stat}$,ϵ_0) avec ϵ_0 l'écart-type de notre statistique, considéré comme erreur cible, on se retrouve alors face à une fonction qui :

- lorsque $\mathcal{E}_{kurt} \gg \epsilon_0$ tends vers $\frac{\mathcal{E}_{stat}}{\epsilon_0}$, normalisant donc bien l'erreur initiale par notre erreur cible, ϵ_0.
- lorsque $\mathcal{E}_{stat} \simeq \epsilon_0$ tends vers 0.
- lorsque $\mathcal{E}_{stat} < \epsilon_0$ deviens négatif, empêchant $\mathcal{E}_{stat}$ de tendre vers 0.

Cela réponds donc idéalement à nos besoins, et nous pouvons donc ainsi remplacer toutes les fonctions d'erreur par leurs distances d'Itakura-Saito avec leurs erreurs cibles : cette distance agira bien comme une normalisation à grande distance de l'objectif, et cessera d'appuyer les statistiques lorsque les erreurs de celles-ci seront de l'ordre des écarts-types extraits de nos ensembles de textures.

Bootstrap

Malheureusement, toute cette méthode repose sur l'obtention d'un grand nombre de textures du même type sur lesquelles nous pourront calculer les "statistiques de nos statistiques" nécessaires à notre imposition : hors, le meilleur moyen pour cela est de travailler avec une texture très longue et constante que l'on pourra découper, chose difficile à trouver (notamment pour les textures d'origine naturelles). Il nous a donc apparu nécessaire de trouver un moyen d'utiliser cette méthode même en ne partant que d'une unique texture de base, trop courte pour être segmentée.

Cela implique d'inférer les moyennes et écarts-types des AC, IC et moments de notre texture cible : la méthode d'inférence avec laquelle nous avons travaillés est le *bootstrap* (pour plus de précision, se référer à l'Annexe B).

Celle-ci permet de simuler la création de signaux de synthèse suivant la même loi de probabilité cachée que le signal de départ en tirant aléatoirement et avec remise des éléments de ce dernier. En répétant le processus, cela rend le calcul de statistiques sur l'ensemble des signaux possible.

Le problème que nous rencontrons d'office est cependant le fait que le signal d'origine (qui dans notre cas serait chacune des amplitudes des sous-bandes en fréquence de notre signal cible) ne peut pas prétendre à l'hypothèse d'indépendance exigée par le bootstrap : cependant et après avoir effectué des tests sur des signaux ne la vérifiant pas, il semblerait que dans le cas où la statistique en question est indépendante de l'ordre des valeurs qui lui sont fournies, les estimations de la

statistiques par bootstrap restent valable (cela pourrait à un certain point être recherché afin de prouver la véracité de cette observation).

Dans le cas des moments d'ordre 1 à 4 qui nous intéressent cette propriété d'indépendance par rapport à l'ordre des valeurs dans le signal à partir desquelles ils sont calculés est évidente, mais le sujet est légèrement plus complexe pour les fonctions de corrélations. Pour rappel, pour chaque valeur de décalage i l'auto-corrélation de S_k est calculée ainsi :

$$\mathcal{A}_{S_k}[i] = \sum_n S_k[n] S_k[n+i] \tag{4.8}$$

Il n'y a donc dans ce cas pas d'indépendance par rapport à l'ordre des valeurs de S_k, mais une indépendance par rapport à l'ordre des valeurs du signal couplé $(S_k[n], S_k[n+i])$: cette suite étant spécifique à la valeur i, l'application du boostrap pour inférer les valeurs que prendrait l'auto-corrélation de la k-ième bande va donc nécessiter des tirages différents pour chaque valeur de décalage i de l'auto-corrélation.

Il est à noter que les premiers résultats donnés par cette méthode semblent cohérents dans leurs ordres de grandeur, mais les corrélation (que ce soit auto ou inter) ont tendance à posséder un écart-type presque constant pour tout décalage i (voir exemple figure 4.2), ce qui heurte un peu l'idée initiale de se servir de cet écart-type pour identifier les zones "importantes" des corrélations. Cependant les résultats sont encore trop récents pour pouvoir en donner une analyse pertinente.

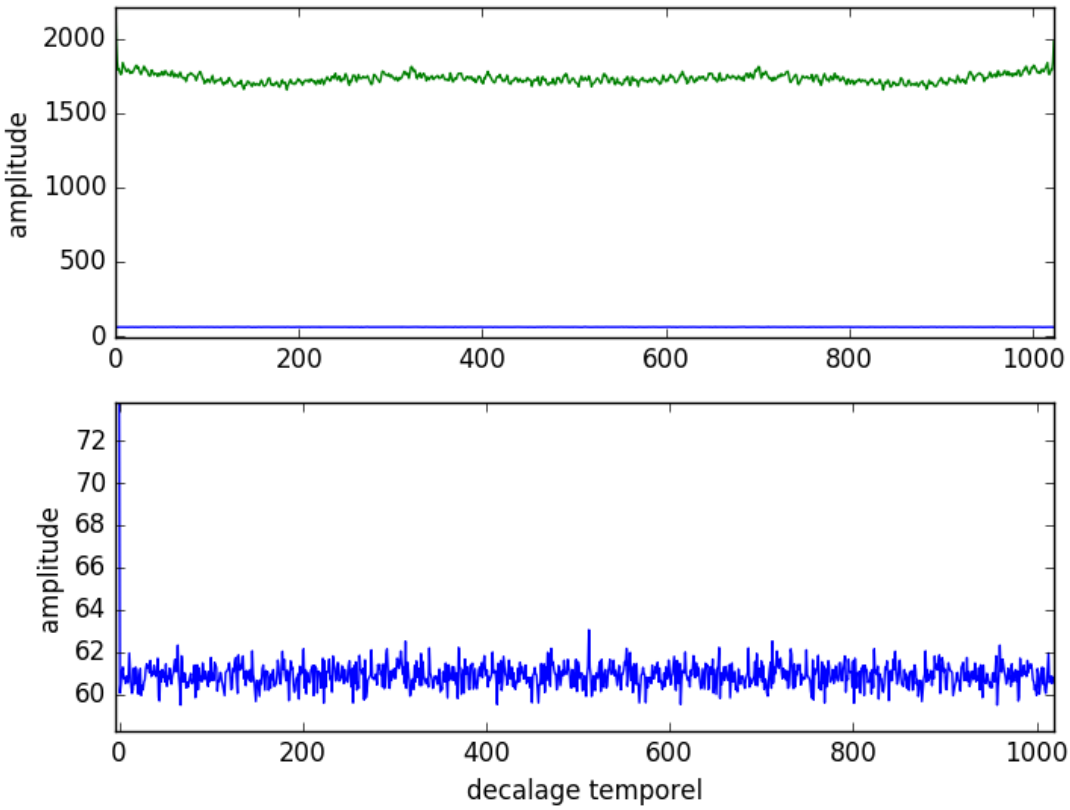

FIGURE 4.2 – *Exemple d'un résultat d'un bootstrap sur 5000 tirages : comparaison entre la fonction d'auto-corrélation cible, en vert, et l'écart-type qui en a été extrait par bootstrap, en bleu.*

Chapitre 5

Perspectives

Avant de faire le bilan de ce stage, faisons rapidement le tour des étapes qu'il nous reste à traverser avant que l'algorithme de synthèse par imposition ne soit entièrement fonctionnel.

5.1 Mise en place d'un test auditif

La première de ces étapes, comme nous l'évoquions plus haut, est la mise en place d'un test auditif.

Nous manquons pour le moment d'un moyen objectif nous permettant de trancher sur certaines décisions : la choix d'un niveau de score de corrélation seuil pour la sélection de couples d'inter-corrélations, la résolution fréquentielle et le recouvrement nécessaire lors des calculs de TFCT, la validité de notre démarche concernant le critère d'arrêt pour l'imposition des statistiques, etc. D'autres interrogations nécessitent elles-aussi la mise en place d'un test, notamment l'utilité du moment de 4^e ordre, le kurtosis, ou le retour aux coefficients de corrélation en lieu et place des auto et inter-corrélations.

Ces points arrivent maintenant à un stade ou une écoute de notre part ne peut suffire à prendre des décisions qui engageront le développement de l'algorithme, faisant de ce test une de nos priorités.

5.2 Contrôle sur la synthèse

Des axes cruciaux dont nous parlions en introduction (réalisme, flexibilité et contrôle à haut-niveau), le contrôle est celui que nous avons le plus délaissé pour le moment.

Présentement, le seul moyen que nous ayons pour contrôler la texture synthétisée est de modifier directement les statistiques cibles. Ces données sont peu claires et les conséquences de leurs modifications tout aussi obscures : il nous sera donc nécessaire d'établir un lien entre l'espace sémantique de description des textures (une pluie

"drue", "éparse", etc.) et l'espace des statistiques. Cela pourrait être fait en dressant une carte de ces espaces à partir d'une base de donnée de textures annotées, puis en se déplaçant dans ces espaces par interpolation.

5.3 Élargissement du terme "texture"

Enfin, le dernier point d'importance que nous avions déjà mentionné lorsque que nous avons dressé le bilan de la méthode de Liao en partie 3.3 est le suivant : notre algorithme n'est toujours pas performant lorsqu'il s'agit de synthétiser des événements distincts, brefs et reconnaissables d'une texture, analogues au tâches noires de la figure 5.1. Nous avions en effet abordé ce sujet lors de notre choix de définition de texture sonore : notre algorithme ne nous permets de synthétiser que des textures sonores comme elles sont définies par Saint-Arnaud dans [Saint-Arnaud, 1995], c'est-à-dire lorsque les atomes sonores constituants la texture sont décrits par des statistiques. La présence d'éléments reconnaissables eux-mêmes présents perceptuellement à "haut-niveau" et trop isolés pour être décrit par des statistiques sort ainsi du cadre de notre définition.

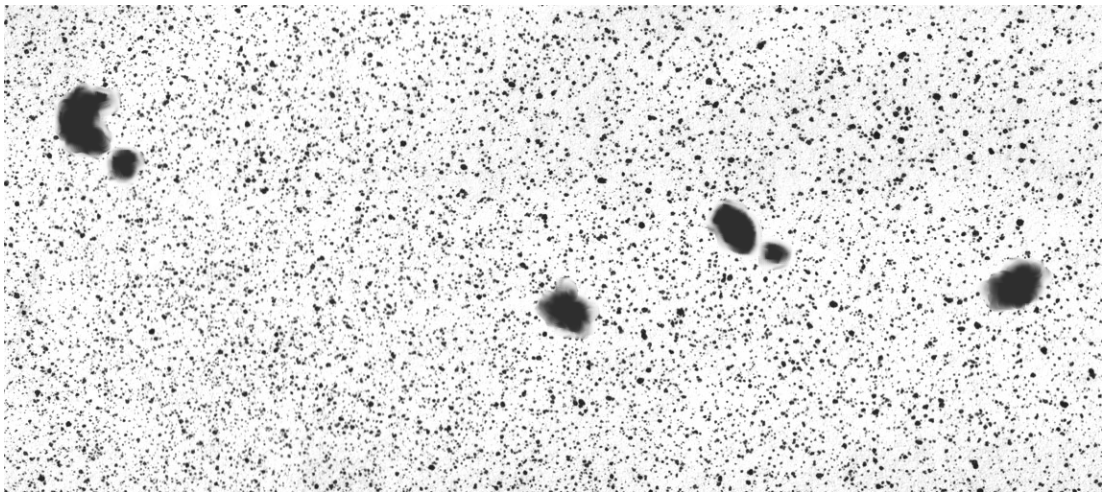

FIGURE 5.1 – *Analogie avec une texture visuelle : présence d'éléments identifiables.*

Notre algorithme est donc incapable de traiter par lui-même ces événements reconnaissables. Si nous voulons permettre la création de textures contenants de tels événements, il va donc s'agir de trouver une méthode hybride permettant de les synthétiser en parallèle. La tâche sera rendue d'autant plus ardue qu'il existe très certainement une corrélation entre ces points reconnaissables et la texture de fond : il ne sera donc sans doute pas possible de les traiter de façon complètement séparée.

Chapitre 6

Conclusion

Ce stage a été principalement l'occasion de travailler à l'amélioration de la flexibilité, c'est-à-dire la capacité à pouvoir synthétiser un grand panel de textures différentes sans interventions au niveau de l'algorithme, d'une nouvelle méthode d'analyse-synthèse de textures sonores : la synthèse par imposition de statistiques temps-fréquence.

Parce qu'une imposition totale de toute les statistiques ne menait qu'à la reproduction quasi-identique de la texture analysée, il a été nécessaire de mettre au point une méthode d'imposition partielle la moins arbitraire possible. Si les deux algorithmes de choix de couples de corrélation et de calcul de marge d'erreur souhaitée mis en place semblent prometteurs, il nous reste cependant à les tester de façon objective via la mise en place de tests auditifs. L'utilisation d'impositions jointes au lieu des impositions séquentielles utilisées jusqu'ici montre des résultats positifs, permettant une meilleur optimisation des moments statistiques.

Si l'uniformité de l'écart-type inféré des corrélations persiste, il pourrait être intéressant de se pencher sur la validité mathématique de la méthode de *bootstrap* utilisée : nous pourrions aussi comparer ces résultats avec d'autres obtenus de façon "naturelle", en segmentant une longue texture en un grand nombre de segments. Pour que ces segments soient perçus comme étant de la même texture, il sera probablement nécessaire de travailler sur une texture d'origine artificielle et facilement maintenable à l'identique sur une longue durée.

Enfin, la poursuite de l'ensemble des objectifs que nous avons mentionné dans la partie finale en thèse devrait permettre à notre algorithme de synthèse de devenir une alternative viable et intéressante par rapport aux méthodes existantes, tout en introduisant une méthode de synthèse sonore nouvelle pouvant être utilisée dans un cadre plus large que les seules textures : la synthèse par imposition de statistiques.

Résumé

Ce rapport est axé autour d'une méthode de synthèse de texture sonore nouvelle : parce qu'il a été montré que seul un nombre restreint de statistiques (corrélations et moments) des enveloppes des sous-bandes en fréquence d'un signal audio permettait de l'identifier comme texture, il se base sur l'imposition de ces statistiques sur un bruit blanc pour synthétiser un signal identifiable comme appartenant à la même catégorie que la texture analysée. Comme une imposition trop forte résultait en un signal synthétisé identique au signal analysé, le travail effectué au cours de ce stage a été de développer une méthode permettant de sélectionner les corrélations nécessaires à imposer et l'erreur que la phase d'imposition par descente de gradient devait cibler : cela s'est fait notamment par l'établissement d'un score de corrélation, la mise en place de l'écart-type des statistiques comme erreur cible et l'utilisation de la distance d'Itakura-Saito.

Abstract

This report is centered around a new method for sound texture synthesis : since it has been shown that only a handful of statistics (namely correlations and moments) from the envelopes of the frequency sub-bands of an audio signal were necessary to identify it as belonging to a given type of texture, the algorithm imposes those statistics upon a white noise to synthesize an output signal that is recognized as being of the same type as the analyzed sound texture. Since too strong of an imposition resulted in a synthesized signal being identical to the analyzed one, this internship mostly focused on developing a method which automatically selected the correlations needing to be imposed and the error that the gradient descent should aim for : this was done through the definition of a correlation score, the use of the standard deviation of the statistics as target error and the use of the Itakuro-Saito distance.

Bibliographie

[Efron, 1979] Efron, B. (1979). Bootstrap methods : Another look at the jackknife. *The Annals of Statistics*, 7(1) :1–26. 30

[Le Roux et al., 2010] Le Roux, J., Kameoka, H., Ono, N., and Sagayama, S. (2010). Fast signal reconstruction from magnitude STFT spectrogram based on spectrogram consistency. In *Proc. 13th International Conference on Digital Audio Effects (DAFx-10)*, pages 397–403. 11

[Liao, 2015] Liao, W.-H. (2015). *Modelling and transformation of sound textures and environmental sounds*. PhD thesis, Université Pierre et Marie Curie. 2, 5, 8, 12

[Liao et al., 2013] Liao, W.-H., Roebel, A., and Su, A. (2013). On the modeling of sound textures based on the stft representation. In *Proc. of the 16th Int. Conference on Digital Audio Effects (DAFx-13)*, page 33. 9

[McDermott and Simoncelli, 2011] McDermott, J. H. and Simoncelli, E. P. (2011). Sound texture perception via statistics of the auditory periphery : Evidence from sound synthesis. *Neuron*, 71(5) :926–940. 5

[O'Brien et al., 2002] O'Brien, J. F., Shen, C., and Gatchalian, C. M. (2002). Synthesizing sounds from rigid-body simulations. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 175–181. ACM. 3

[Saint-Arnaud, 1995] Saint-Arnaud, N. (1995). *Classification of sound textures*. PhD thesis, Massachusetts Institute of Technology. 1, 23

[Schwarz, 2007] Schwarz, D. (2007). Corpus-based concatenative synthesis. *IEEE signal processing magazine*, 24(2) :92–104. 4

[Schwarz, 2011] Schwarz, D. (2011). State of the art in sound texture synthesis. In *Digital Audio Effects (DAFx)*, pages 1–1. 1

Annexe A

Définitions des statistiques utilisées

Soit x et y deux signaux discrets de taille N.

A.1 Moyenne

La moyenne $\mu(x)$ de x est définie ainsi :

$$\mu(x) = \frac{1}{N} \sum_{0 \leq i \leq N-1} x[i] \tag{A.1}$$

A.2 Variance ou écart-type

La variance $v(x)$ de x est définie ainsi :

$$v(x) = \frac{1}{N^2} \sum_{0 \leq i \leq N-1} (x[i] - \mu(x))^2 \tag{A.2}$$

Et son écart-type $\sigma(x)$ se définit simplement :

$$\sigma(x) = \sqrt{v(x)} \tag{A.3}$$

A.3 Asymétrie

L'asymétrie $\eta(x)$ de x est définie ainsi :

$$\eta(x) = \frac{1}{N^3} \sum_{0 \leq n \leq N-1} \left(\frac{(x[n] - \mu(x))}{\sigma(x)} \right)^3 \tag{A.4}$$

A.4 Kurtosis

Le kurtosis $\kappa(x)$ de x est définie ainsi :

$$\kappa(x) = \frac{1}{N^4} \sum_{0 \leq i \leq N-1} \left(\frac{(x[i] - \mu(x))}{\sigma(x)} \right)^4 \tag{A.5}$$

A.5 Auto-corrélation

L'auto-corrélation $\mathcal{AC}_x$ de x est définie pour chaque valeur de décalage i ainsi :

$$\mathcal{AC}_x[i] = \sum_{0 \leq n \leq N-1} (x[n]x[n+i]) \tag{A.6}$$

Dans le cas où $k \geq N$, la valeur que l'on donne à $x[k]$ dépend de la corrélation adoptée. Donc le cas de la corrélation linéaire, on posera $x[k] = 0$ tandis dans le cas de la corrélation circulaire, qui est est celle utilisée pour le moment, on posera $x[k] = x[k\%N]$, avec $\%$ indiquant "modulo".

A.6 Inter-corrélation

L'auto-corrélation $\mathcal{IC}_{x,y}$ de x et y est définie pour chaque valeur de décalage i ainsi :

$$\mathcal{IC}_{x,y}[i] = \sum_{0 \leq n \leq N-1} (x[n]y[n+i]) \tag{A.7}$$

Annexe B

Le Bootstrap

Le bootstrap est une méthode introduite dans [Efron, 1979] permettant d'inférer les propriétés d'un estimateur statistique (un moment, par exemple) d'une suite X d'observations indépendantes et identiquement distribuées (iid) suivant une loi $\mathcal{F}$ inconnue en simulant la création de nouvelles suites d'observations.

Ces simulations sont basées sur l'idée que si X est constitué de tirages indépendants et suivant la même loi $\mathcal{F}$, alors pour une taille N de X suffisamment grande l'histogramme des valeurs de la suite approxime fidèlement la distribution donnée par la loi $\mathcal{F}$. Il devient donc possible de simuler une nouvelle série $\widetilde{X}$ d'observations en tirant N fois avec remise une observation parmi celles de X. En réitérant cette démarche un grand nombre de fois, il devient alors possible d'établir certaines propriétés, comme la moyenne ou l'écart-type, d'un estimateur.

À titre d'exemple, supposons que l'on s'intéresse à la moyenne $\mu(X)$ de X. Pour cela, on créé un premier signal synthétique en créant une liste P_0 de N entiers choisis aléatoirement et uniformément entre 0 et $N-1$. Notre premier signal simulé $\widetilde{X}_0$ pourra ensuite être défini pour chacun de ses termes par :

$$\widetilde{X}_0[i] = X\left[P_0[i]\right] \qquad 0 \leq i \leq N-1 \tag{B.1}$$

On pose alors $\widetilde{\mu}_0 = \mu(\widetilde{X}_0)$. En réitérant le procédé ad libitum, on obtient donc autant d'estimations de la moyenne $\mu(\widetilde{X}_1)$, $\mu(\widetilde{X}_2)$, $\mu(\widetilde{X}_3)$, etc., suivant une approximation de la loi $\mathcal{F}$: il devient ensuite possible d'effectuer une approximation des caractéristiques de l'estimateur qui nous intéresse, par exemple son écart-type.