Rapports de TPs, traitement audio-visuel

Ewen Le Bihan

5 juin 2024

Table des matières

1 TP 5 : Débruitage et inpainting **2**
1.1 Débruitage . . . 2
1.2 Inpainting par variation totale . . . 2
1.3 Inpainting par rapiéçage . . . 3

2 TP 8 : Décomposition structure/texture **4**
2.1 Décomposition par modification du spectre . . . 4
2.2 Approche variationnelle . . . 4
2.3 Une application à la détection de contours . . . 6

3 TP 11 : Reconnaissance musicale **7**
3.1 Calcul des pics spectraux . . . 7
3.2 Création de paires . . . 7
3.3 Curiosité : Une anomalie sur l'analyse de morceaux "ambient" . . . 7

4 TP 12 : Séparation de sources musicales **10**
4.1 Séparation contenu percussif/harmonique . . . 10
4.2 Curiosité : Interchanger percussivité et harmonicité . . . 11

5 TP 10 : Manipulation de signaux audionumériques **12**
5.1 Transformée de Fourier à court terme . . . 12
5.2 Effets sur sonogramme : passe-bas . . . 12
5.3 Passe-bas modulé par enveloppe . . . 12
5.3.1 Passe-bas progressif . . . 13
5.4 Contour d'enveloppe . . . 14

1 TP 5 : Débruitage et inpainting

1.1 Débruitage

On utilise un modèle de débruitage par variation totale. On cherche à minimiser les gradients entre chacun des pixels, en minimisant l'énergie

$$\frac{1}{2}\iint_{\Omega}(u(x,y)-u_0(x,y))^2+\lambda|\nabla u(x,y)|^2\mathrm{d}x\mathrm{d}y$$

L'implémentation Matlab utilise une méthode de résolution itérative sur une version discrétisée du problème d'optimisation de l'équation d'Euler-Lagrange liée à la minimisation de l'énergie précédente.

```
function u_kp1 = debruitage(b,u_k,lambda,Dx,Dy,epsilon)
    gradient_values = mean(1 ./ sqrt(gradient(u_k).^2 + epsilon), 2);
    pixel_count = length(gradient_values);
    Wk = spdiags([gradient_values gradient_values], [0 pixel_count], pixel_count, pixel_count);
    u_kp1 = (speye(pixel_count) - lambda * ( -Dx' * Wk * Dx - Dy' * Wk * Dy)) \ b;
```

Cette fonction traduit le calcul d'une itération basé sur l'équation

$$\underbrace{(I_N-\lambda(-D_x^\top W^{(k)}D_x-D_y^\top W^{(k)}D_y)}_{A^{(k)}}u^{(k+1)}=\underbrace{u_0}_{b}$$

1.2 Inpainting par variation totale

L'inpainting, qui consiste à supprimer une zone indésirable de l'image et à tenter de re-construire ce que se trouverait à la place de l'objet délimité par cette zone, peut être implémenté par variation totale, en reprenant l'algorithme précédent.

La différence est que l'on ne calcule plus le terme d'attache aux données de l'énergie sur la totalité de l'image (Ω) mais sur l'image, privée de cette zone indésirable ($\Omega \setminus D$) : en effet, on ne cherche pas à coller aux pixels originaux pour la zone indésirable. Ceci conduit à l'expression suivante de l'énergie à minimiser

$$\frac{1}{2}\iint_{\mathbf{\Omega\setminus D}}(u(x,y)-u_0(x,y))^2\mathrm{d}x\mathrm{d}y+\iint_{\Omega}\lambda|\nabla u(x,y)|^2\mathrm{d}x\mathrm{d}y$$

Cependant, le résultat n'est satisfaisant que si la zone indésirable n'a pas de composantes connexes trop grandes en aire :

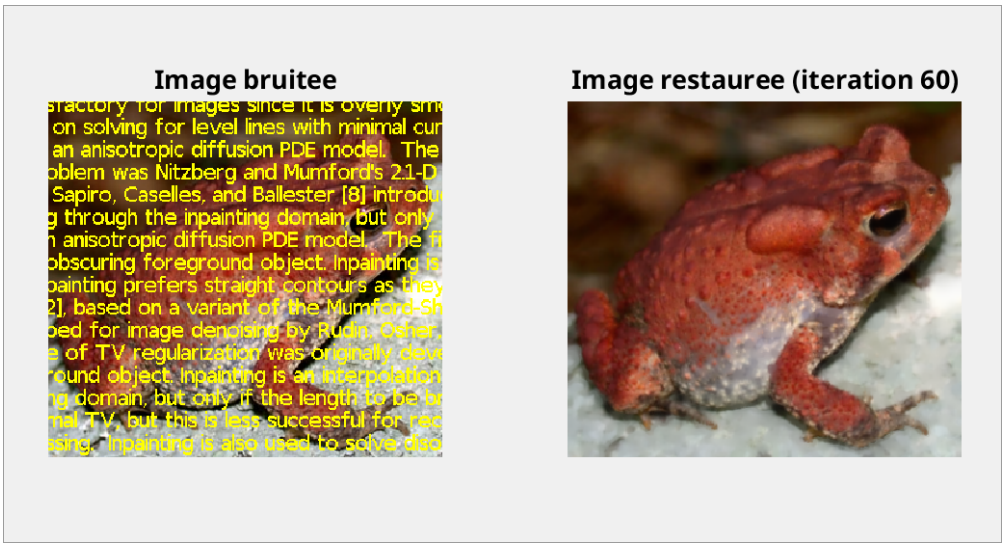

FIGURE 1 – Inpainting par variation totale, résultat satisfaisant

FIGURE 2 – Inpainting par variation totale, résultat insatisfaisant

Les composantes connexes suffisamment grandes nous permette de voir que l'algorithme remplie les zones indésirables par une sorte de moyenne des pixels en bordure de ladite composante connexe, ce qui ne donne pas un résultat satisfaisant.

1.3 Inpainting par rapiéçage

L'inpainting par rapiéçage consiste à chercher des "patches" dans des zones non-indésirables de l'image qui ressemblent le plus à la zone indésirable, et à les utiliser pour remplacer celle-ci. Cet algorithme suppose que la zone indésirable est connexe.

Pour chaque pixel p de la bordure, on regarde dans une certaine fenêtre $F(p)$ centrée en p tout les patches (zones carrées de pixels), et on garde le pixel central au patch le plus ressemblant au patch centré en p, en calculant sa *dissemblance*[1].

1. La dissemblance est définie comme la moyenne des écarts carrés entre le pixel p et les pixels *hors de la zone indésirable* du patch

2 TP 8 : Décomposition structure/texture

L'objectif ici est de séparer une image en deux complémentaires : une contenant le contenu fréquentiel haut de l'image (la *texture*) et l'autre les fréquences basses (la *structure*).

Cette formulation en termes fréquentiels amène naturellement à l'utilisation de transformées de Fourier pour décomposer l'image.

2.1 Décomposition par modification du spectre

Cette méthode consiste simplement à faire une transformée de Fourrier discrète sur l'image, ne garder que certains coefficients et appliquer une transofrmée inverse.

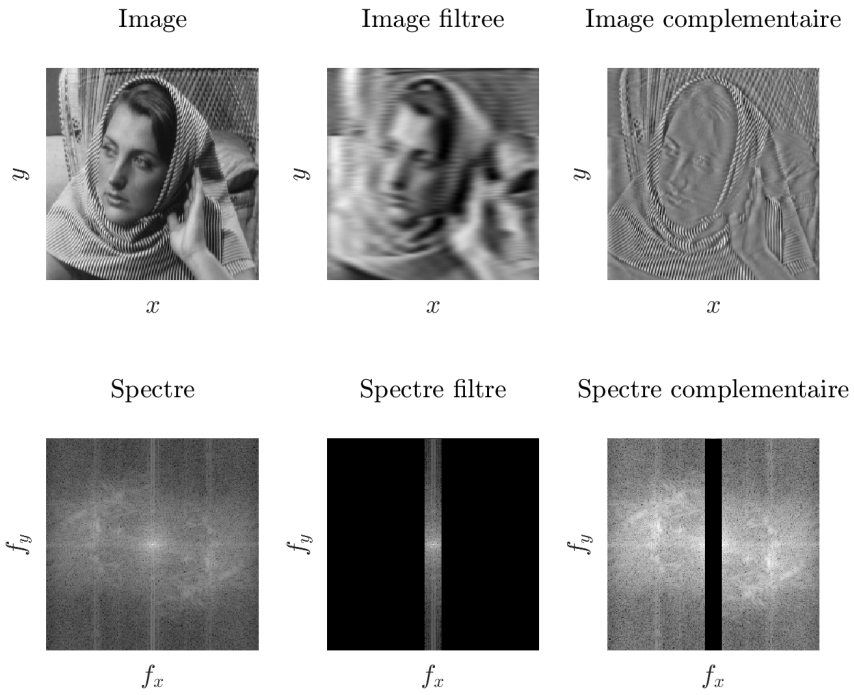

FIGURE 3 – Exemple

2.2 Approche variationnelle

On minimise une énergie comportant deux termes :

L'attache aux données On souhaite que l'image *structure* soit proche de l'image source

L'écart texture/structure On souhaite que l'image structure comporte des gradients moins forts que l'image texture : l'intensité des gradients est une conséquence du fait de comporter des hautes fréquences

On résout la minimisation de l'énergie via un algorithme itératif sur une discrétisation de l'équation d'Euler-Lagrange associée.

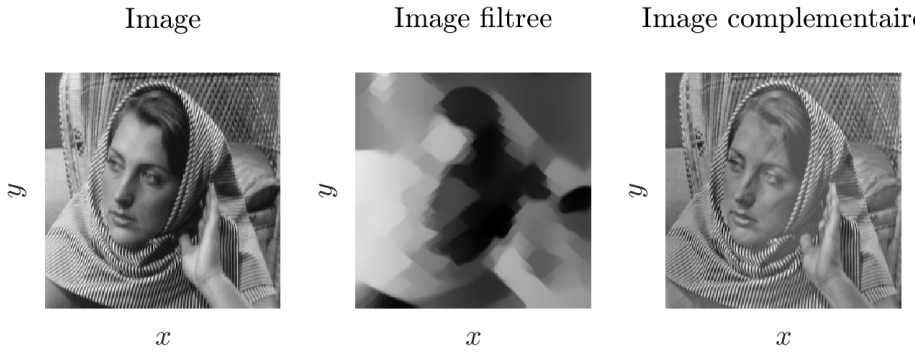

FIGURE 4 – Exemple

On a un facteur λ qui permet de gérer l'importance que l'on accorde au critère d'écart texture/structure par rapport au critère d'attache aux données. L'exemple précédent utilise une valeur modérée de $\lambda = 100$. On peut observer que ce λ traduit la proportion de fréquences hautes que l'on décide de "ranger" dans la texture. Le paramètre agit comme une sorte de fréquence de coupure, pour faire une analogie avec un filtre passe-bas en traitement audio.

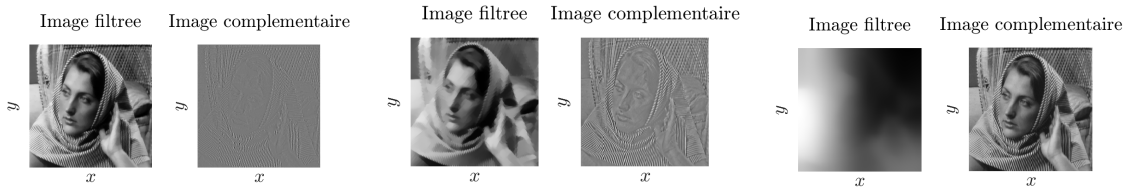

FIGURE 5 – Exemple avec $\lambda \in \{1, 10, 1000\}$ (croissant de gauche à droite)

2.3 Une application à la détection de contours

Cette séparation texture/structure nous donne au final deux images, plutôt qu'une, pour potentiellement guider un choix automatisé des seuils pour une détection de contours à hystérésis (seuil fort et seuil faible). C'est ce à quoi j'ai été confronté dans mon TIPE, qui portait sur la détection et classification de fractures osseuses à partir d'images radiographiques [2].

J'avais finit par faire du *Deep Reinforcement Learning* pour choisir des seuils en fonction d'informations comme la luminosité et le contraste, mais avoir deux images qui caractérise les hautes et basses fréquences d'une image peut être utile : la luminosité de l'image structure donne une bonne idée de la luminosité globale de l'image, et le contraste de l'image de texture donne une assez bonne idée de la *finesse* des potentielles fractures : plus un trait de fracture est flagrant, plus la texture portera une marque contrastée.

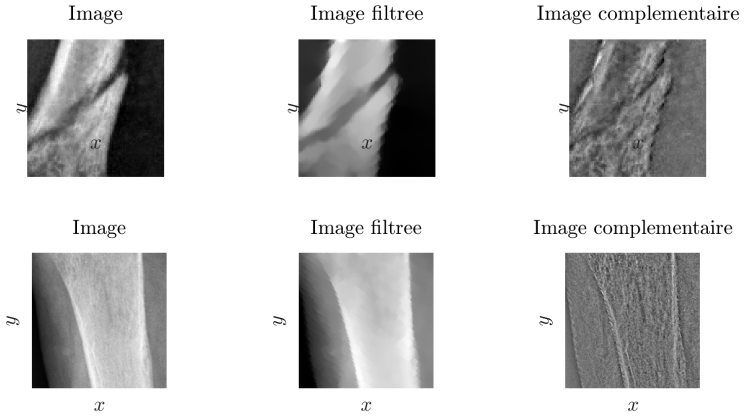

FIGURE 6 – Application à de la détection de contours à finalité médicale

2. Voir `https://github.com/ewen-lbh/bone-fracture-detection`

3 TP 11 : Reconnaissance musicale

Le principe consiste à stocker des paires de *pics spectraux* d'un morceau dans une base de données globale, en les associants au titre, artiste (et d'autres méta-données musicales telles que l'UNPC). Un · e utilisateur · ice de l'application peut ensuite enregistrer un extrait audio, en extraire des paires de pics spectraux et faire une recherche dans une base de données afin de trouver le morceau correspondant.

3.1 Calcul des pics spectraux

Les pics spectraux sont calculés sur le sonagramme : on prend le point de plus haute intensité parmi ses voisins, et ceux pour chaque point du sonagramme. Le sonagramme est calculé par transformée de Fourrier.

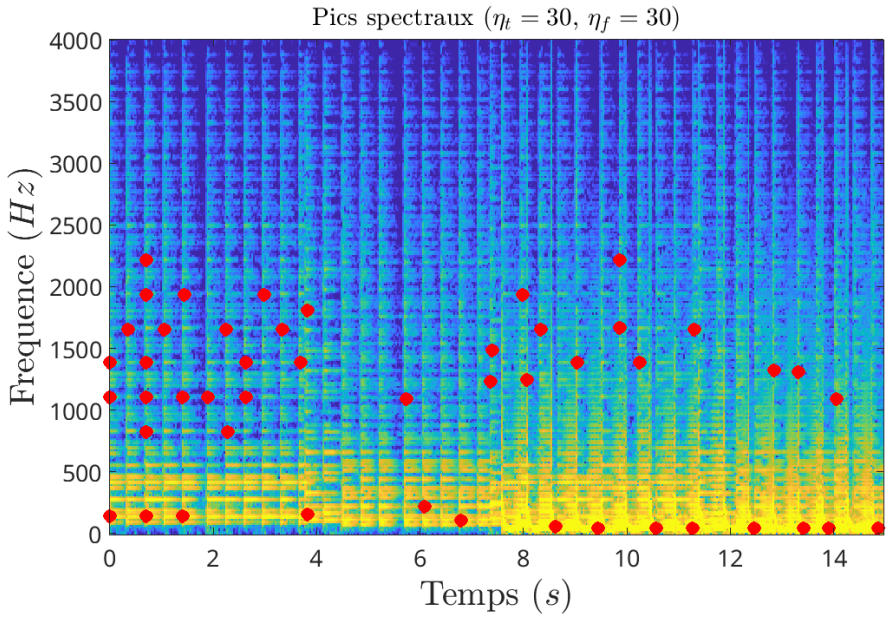

FIGURE 7 – Pics spectraux de *Ewen-lbh — Ame to Yuki*

3.2 Création de paires

Il faut ensuite créer des paires depuis ces pics spectraux. L'idée est que l'on apparie chaque point avec un autre s'ils sont suffisament proches en hauteur et temporellement, avec une condition supplémentaire pour la proximité temporelle : on n'apparie que "dans un sens", ce qui se traduit par le fait de ne considérer que les points ultérieurs au point en cours de traitement.

3.3 Curiosité : Une anomalie sur l'analyse de morceaux "ambient"

En tant qu'artiste distribuant sur les plateformes, j'ai accès aux statistques Shazam de mes morceaux. J'ai remarqué quelque chose de particulier sur un de mes morceaux :

FIGURE 8 – Statistiques Apple Music de *Ewen-lbh — ...until the novelty wears off*

On peut remarquer qu'il y a *très largement* plus de "shazams" [3] que d'écoutes, et c'est d'ailleurs peu réaliste qu'il y ait des shazams tout court étant donné la popularité de l'artiste et le caractère très "ambiant" du morceau :

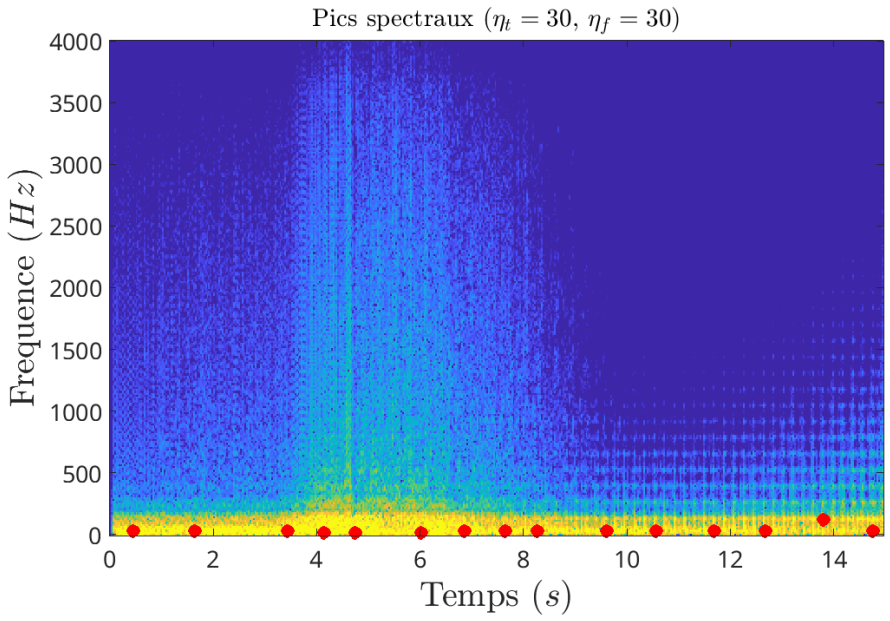

FIGURE 9 – Sonogramme et pics de *Ewen-lbh — ...until the novelty wears off*

Ma théorie est que les pics sont tellement peu nombreux est à fréquence basse que Shazam à tendance à matcher mon morceau quand on tente de *Shazamer* une ambiance calme sans musique apparente.

J'ai analysé [4] un enregistrement d'un lieu très calme [1]. Le résultat montre très peu de pics, dans les mêmes places de fréquence que le morceau. On peut imaginer qu'une ambiance un peu moins calme génèrera autant de pics que le morceau, à des emplacements fréquentio-temporels assez similaires pour induire Shazam en erreur.

3. Shazam est un service de reconnaissance musicale, qui a mis au point la technique de paires de pics spectraux
4. Les deux fichiers audio ont été convertis en OGG Vorbis avec échantillonage à 8 kHz via *ffmpeg*, pour coller au reste des morceaux originellement présents dans `data.min`

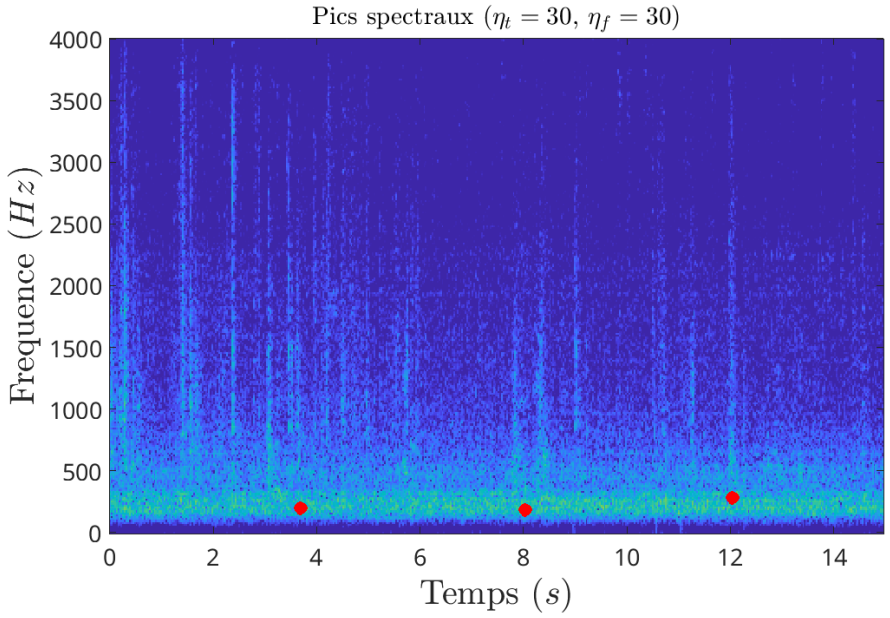

FIGURE 10 – Analyse de `classroom_ambiance.wav`

On peut en conclure que beaucoup de gens essaie de shazamer du vide, par curiosité :)

4 TP 12 : Séparation de sources musicales

4.1 Séparation contenu percussif/harmonique

Cette méthode est basé sur le constant que les sons percussifs sont, sur un sonogramme, fréquentiellement larges et temporellement étroits, et inversement pour les sons harmoniques.

En effet, un son harmonique comporte comparativement peut de fréquences différentes simultanées (sinon le cerveau ne peut "se décider" sur la note fondamentale à entendre), tandisque une percussion est considérablement plus proche du bruit blanc (c'est d'ailleurs en partant d'un bruit blanc que l'on peut créer la plupart des sons de caisse claire / hi hat /etc des musiques bas-débit "chiptune"), et le bruit blanc comporte toutes les fréquences. Et temporellement, un son percussif est bref tandisqu'un son harmonique est (souvent) plus long.

En filtrant pour ces caractéristiques, on peut obtenir deux sonogrames complémentaires, l'un contenant les sons percussifs et l'autre le reste.

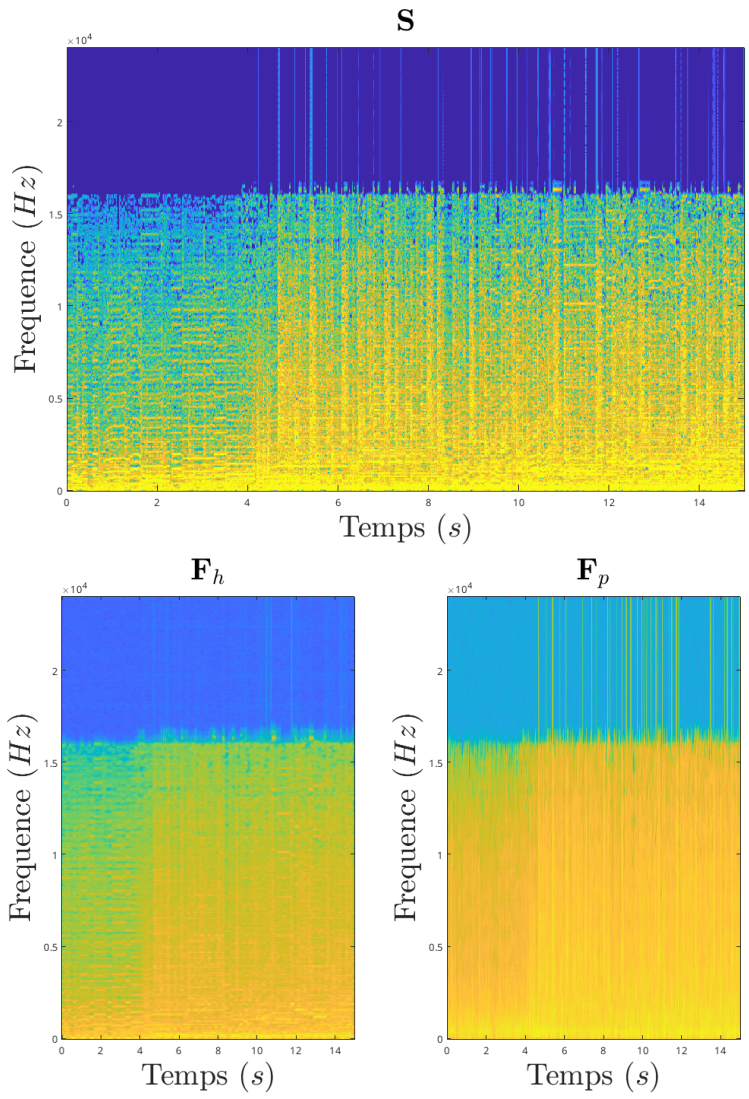

FIGURE 11 – Exemple avec un morceau du genre "Chiptune" (*meganeko — Milkshake*, de `00:01'30"` à `00:01'45"`)

Sur cet exemple[5], la séparation se fait plutôt bien, excepté le *kick*[6], qui apparaît plutôt dans la partie harmonique que percussive. On pourrait s'attendre à l'inverse, mais ce n'est en fait pas très surprenant : contrairement à la plupart des autres percussions, un kick n'occupe au final qu'une faible partie de la plage fréquentielle, et est comparable à une basse.

C'est d'ailleurs tellement comparable à une basse que, en production musicale, on a souvent recours à une technique appelée *sidechain* qui consiste à réduire le volume de la basse quand le kick est présent, parce que les deux instruments occupent la même plage fréquentielle mais sont tout les deux fondamentaux à un morceau, en particulier en musique électronique. Certains genres, comme la *psytrance*, vont même plus loin en utilisant des basses rapides qui n'ont jamais de note en même temps que le kick.

4.2 Curiosité : Interchanger percussivité et harmonicité

L'observation précédente amène une question intéréssante : que donne un morceau de musique si l'on effectue une rotation de 90 degrés de son spectre, transformant ainsi les sons harmoniques en percussions et inversement ?

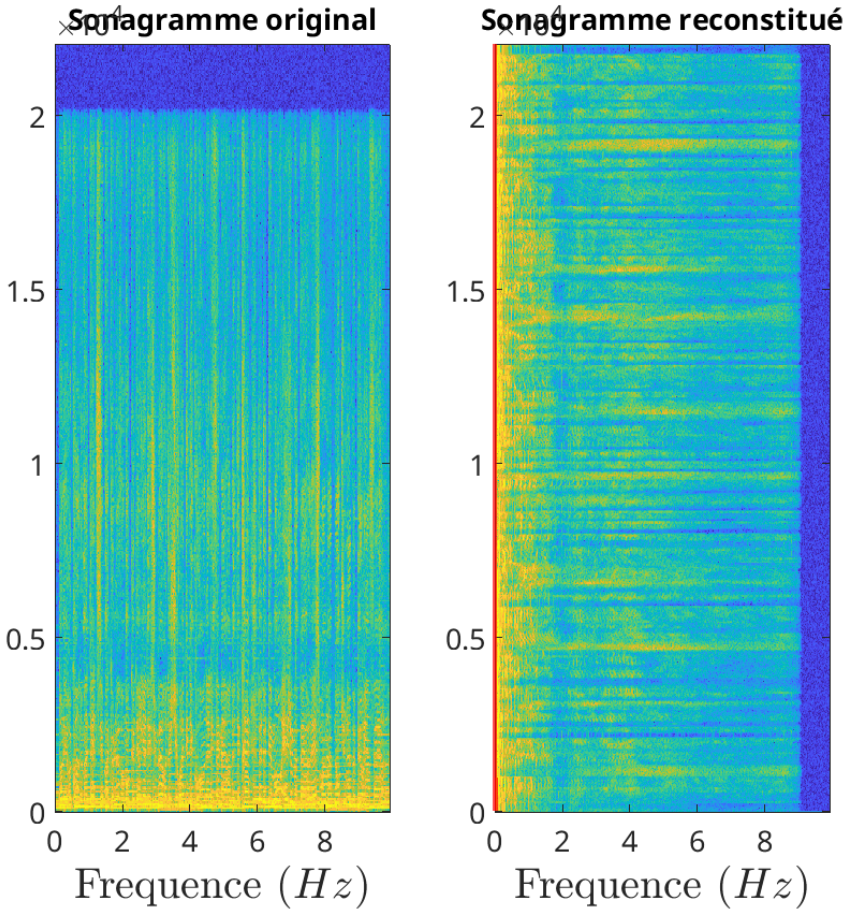

FIGURE 12 – Rotation à 90 degrés du sonogramme de *Audio/mpl.wav* (TP 10)

Résultat : `https://media.ewen.works/rapport-tav/rot90.mp3` (attention, ça peut faire mal aux oreilles, mais ça peut aussi faire office d'effet "glitch" pour un film de science-fiction avec des hackers)

5. Écoutable ici : `https://media.ewen.works/rapport-tav`
6. grosse caisse

5 TP 10 : Manipulation de signaux audionumériques

5.1 Transformée de Fourier à court terme

La *TFTC* est une manière d'avoir une représentation qui présente *à la fois* l'aspect temporel et fréquentiel d'un son.

De la même manière qu'on compteur de vitesse de voiture donne une vitesse "instantanée" expérimentale en "trichant" par l'emploi d'une fenêtre glissante d'un durée non-nulle pour calculer en fait une vitesse moyenne sur une durée courte, la TFTC calcule le spectre fréquentiel sur des durées courtes pour donner un *sonogramme* .

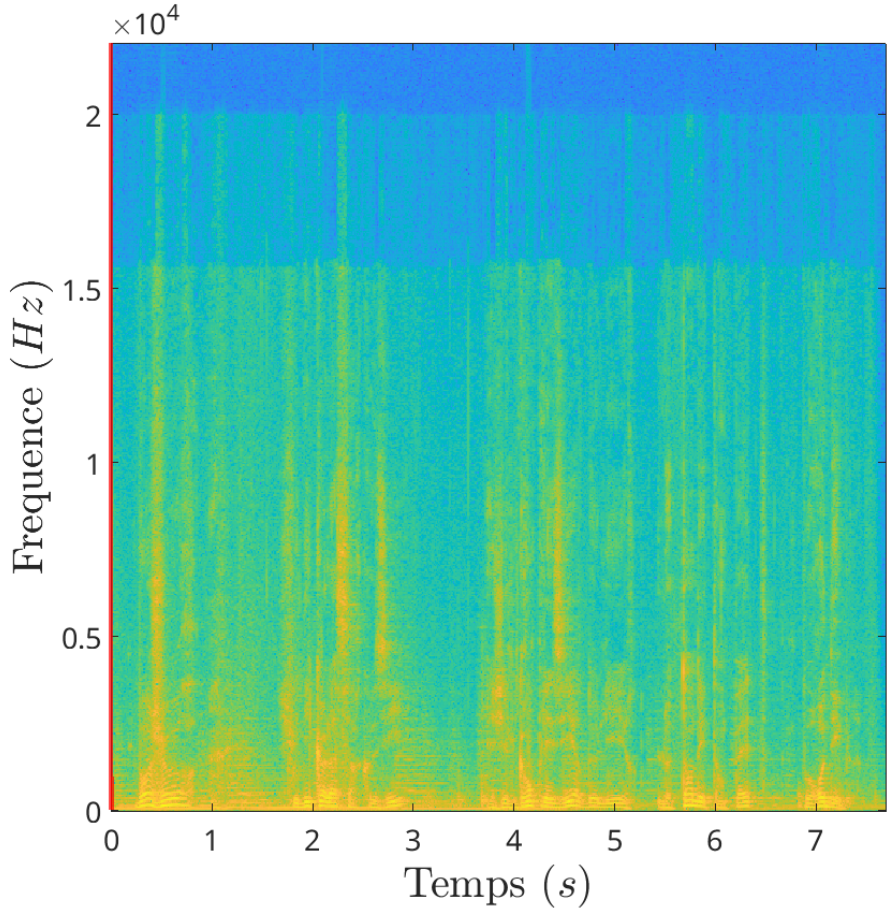

FIGURE 13 – Sonagramme de la publicité du *Temps des tempêtes*

5.2 Effets sur sonogramme : passe-bas

Un moyen d'implémenter un passe-bas assez naturel est de simplement mettre à 0 tout les points d'un sonogramme dont la fréquence est supérieure à un seuil, appelé fréquence de coupure (ou *cutoff*)

5.3 Passe-bas modulé par enveloppe

En production musicale électronique, beaucoup de sons de synthétiseurs sont créés via des instruments virtuels (*VST*[7]) permettant de la synthèse dite *additive* : on part d'oscillateurs, qui produisent des ondes de formes élémentaires (sinusoïde, *saw*[8], carré, triangle,...) dont la fréquence est controllé par la hauteur de la note jouée au clavier. Puis, on donne à ces oscillateurs une enveloppe ADSR (attack, decay, sustain, release)

7. Virtual Studio Technology
8. Abbréviation courante de *sawtooth*, signifiant "dent de scie"

modulant leur amplitude. Enfin, il est assez commun d'appliquer un filtre passe-bas au signal, le signal brut pouvant être assez perçant (en particulier avec des saw). Enfin, manipuler la fréquence de coupure permet de donner du mouvement au son du synthétiseur : on dit que l'on *ouvre le filtre* quand on augment sa fréquence de coupure.

Beaucoup de sons de synthés rythmés en musique électronique lie l'ouverture du filtre à l'enveloppe : quand on joue une note, le filtre s'ouvre puis se referme vers la fin de la note.

FIGURE 14 – Un synthé construit avec *Serum*, qui module le cutoff du filtre (section *FILTER*) avec l'enveloppe (section *ENV1*), dont la plage de modulation est indiqué par la section bleue sur le potentiomètre *CUTOFF*

L'idée est de tenter de reproduire cet effet *sans accès au données MIDI (donc au timings des notes telles qu'elles sont jouées*, en se basant simplement sur le signal audio d'un synthé dont le filtre n'est pas modulé par son enveloppe

Pour se faire, on va détecter les pics d'amplitude dans le signal puis appliquer un filtre passe bas dont la fréquence de coupure dépend du temps, ce qui se traduit parl'application d'un masque *diagonal* sur le sonogramme.

5.3.1 Passe-bas progressif

On commence par tester un filtre passe-bas dit "progressif", qui possède deux fréquences de coupure : celle du début du traitement, et celle de la fin du traitement. On fourni aussi la durée sur laquelle cette progression du cutoff doit s'effectuer. On interpolera linéairement par souci de simplicité, mais les VST offrent d'utiliser d'autres courbes, dont celle de l'amplitude.

```
function out = passe_bas_progressif(in, frequencies, starting_cutoff, ending_cutoff, duration)
    out = in;
    cutoff_step_size = (ending_cutoff - starting_cutoff) / duration;
    for i = 1:size(in, 2)
        cutoff = starting_cutoff + cutoff_step_size * i;
        out(frequencies > cutoff, i) = 0;
    end
```

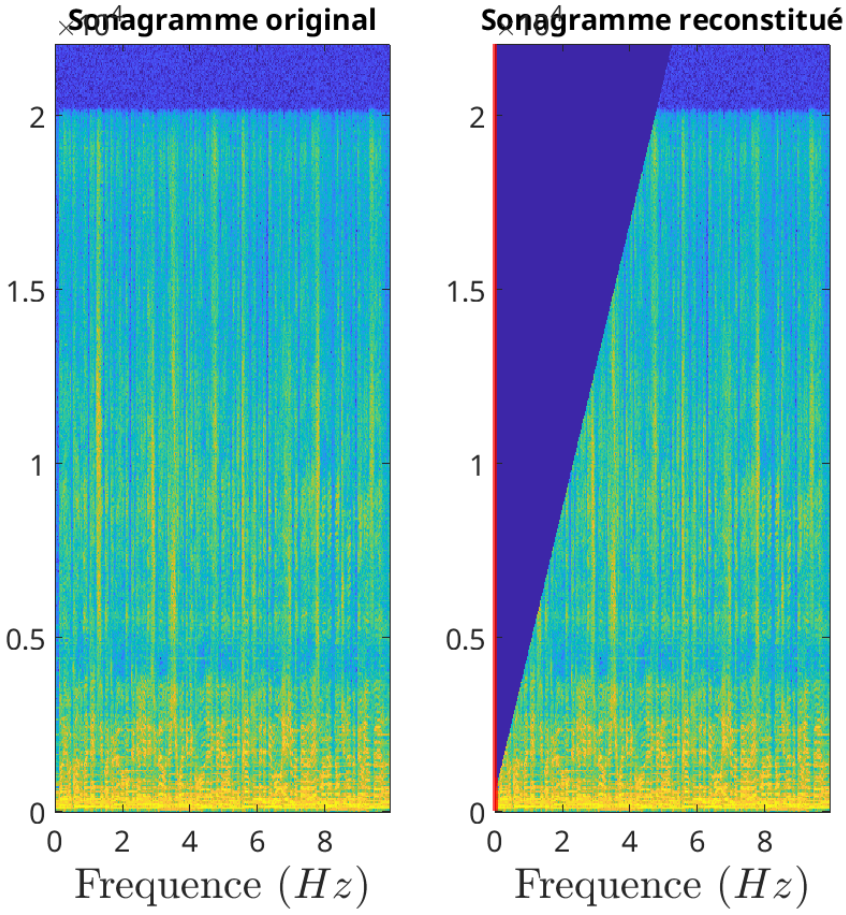

FIGURE 15 – Passe-bas “progressif” de 500 Hz à 10 kHz

Résultat : `https://media.ewen.works/rapport-tav/passe_bas_progressif.mp3`

On note aussi que ouvrir progressivement un passe bas depuis 0 Hz jusqu'à une fréquence assez élevée pour que le filtre n'agisse plus (10 kHz pour être sûr · e de soi) est également un moyen courant d'introduire un instrument (par exemple, la mélodie principale au début de `https://media.ewen.works/rapport-tav/ewen-lbh-petrichor.mp3` [9])

5.4 Contour d'enveloppe

Enfin, pour obtenir l'effet désiré, on applique cette rampe à chaque fois que l'on détecte que la moyenne des intensités à un instant est supérieur à la moyenne maximale, plus ou moins un certain ε.

9. exclu de mon prochain album “émoti*ns” d'ailleurs

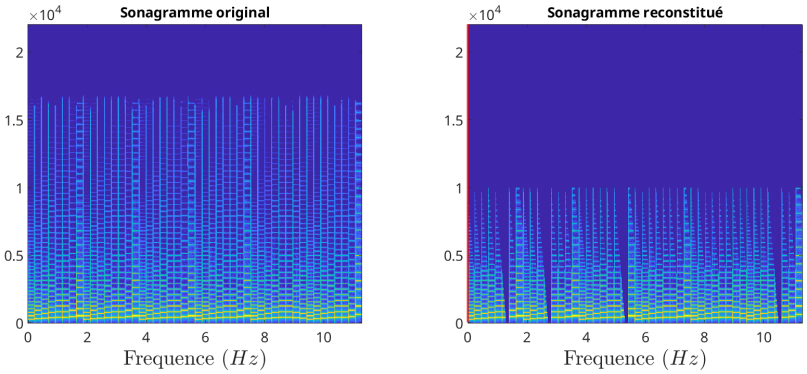

FIGURE 16 – Exemple sur un son de synthé

Son original `https://media.ewen.works/rapport-tav/synth_no_cutoff.mp3`

Effet reconstitué `https://media.ewen.works/rapport-tav/synth_cutoff_rebuilt.mp3`

Effet original `https://media.ewen.works/rapport-tav/synth_with_cutoff.mp3`

Synthé dans son contexte `https://media.ewen.works/rapport-tav/synth_full_context.mp3`

Références

[1] Joe DeShon. classroom_ambiance.wav. `https://freesound.org/people/joedeshon/sounds/258094/`, 2014.