

## Rapport final du mini projet de signal

Paul Audin - Paul Kern

8 Juin 2020

### **1 Introduction**

La problématique du projet est de cacher une information dans le spectre de fréquence d'un fichier sonore. Dans notre cas, nous chercherons à incruster un film dans le spectre d'une chanson afin de la faire passer incognito. Une fois incrustées, les images pourraient être récupérées par le destinataire dans le spectre de la musique reconstruite.

En pratique, le script prend en entrée une musique ainsi qu'un film, et enregistrera une nouvelle musique dont le spectre contiendra plusieurs images. Il permettra également de visualiser le spectre contenant les images pré et post-enregistrement.

Ce projet s'appuie sur l'utilisation de transformées de Fourier glissantes afin de construire un spectre sans discontinuités. Il faudra ensuite réussir à reconstruire le fichier audio à partir de ce spectre.

## 2 Principe de fonctionnement

Tout d'abord, nous devons effectuer une transformée de Fourier glissante. Il s'agit d'effectuer une TF sur un nombre  $N_y$  donné de points, puis une suivante sur le même nombre de points, mais décalée de  $pas$  points, puis une autre décalée de la même valeur, etc... Le schéma suivant explique le principe de la TF glissante :

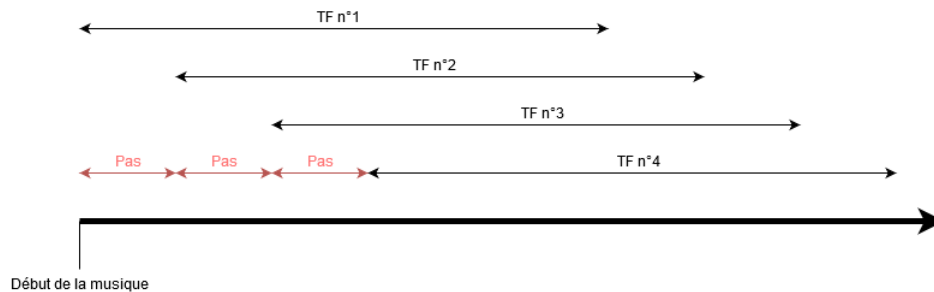


Figure 1: Principe de la transformée de Fourier glissante

Après l'avoir réalisée sur tout le fichier audio, on obtient une matrice dont les colonnes représentent des tranches temporelles, et les lignes des tranches fréquentielles, ce qui permet de l'afficher sous forme de spectre, comme le suivant :

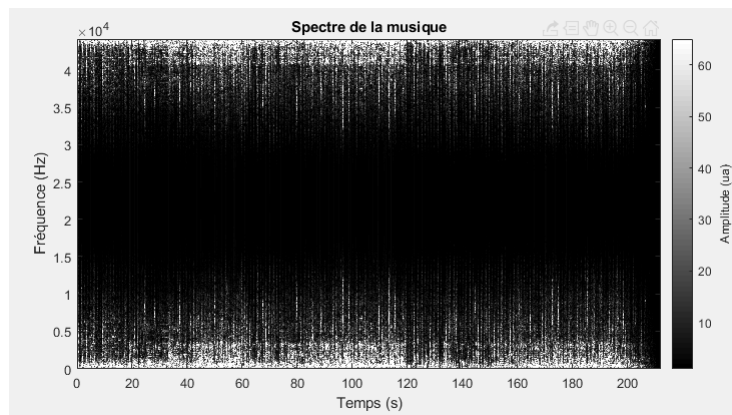


Figure 2: Spectre de la musique *Never Gonna Give You Up* de Rick Astley

Ensuite, nous devons ajouter une image, qu'il faut au préalable arranger : hauteur, largeur, couleur, intensité. Nous devons également l'insérer deux fois dans

le spectre, de façon symétrique par rapport à la fréquence médiane pour conserver une égalité stricte entre les deux moitiés du spectre :

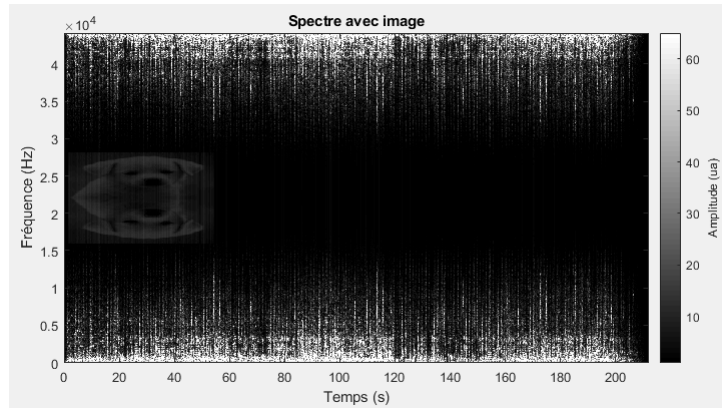


Figure 3: Spectre de la musique avec l'image insérée en double

Ensuite, il faut reconstruire la musique avec le spectre contenant l'image. Or chaque valeur du son de départ se trouve dans plusieurs colonnes à cause de la TF glissante. Pour ne pas perdre d'informations, nous effectuons d'abord la TF inverse sur le spectre, avant de faire la moyenne de toutes les apparitions de chaque valeur. Le schéma explicatif de ce principe est le suivant :

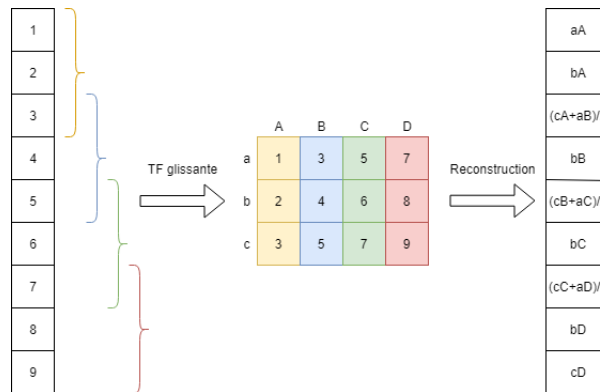


Figure 4: Algorithme de reconstruction du son. Chaque valeur pouvant apparaître plusieurs fois, nous faisons la moyenne de toutes leurs apparitions. Ici  $N_y = 3$  et  $pas = 2$ .

Nous pouvons ensuite recréer le fichier son à l'aide des commandes Matlab, et le tour est joué !

### 3 Résultats

L'exécution du code final produit trois figures qui présentent la chaîne de traitement de l'image.

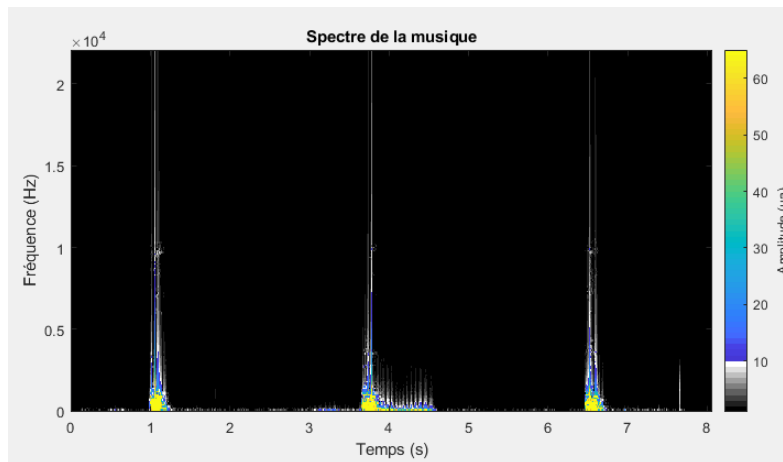


Figure 5: Spectre du fichier audio utilisé

Cette première figure représente le spectre du fichier audio utilisé (ici, prononciation du mot 'eau' à trois reprises). On y observe trois pics distincts correspondant aux trois mots. L'échelle de couleur utilisée est basée sur la colormap 'hsv'. Nous l'avons modifiée afin d'y incorporer une échelle de noir et blanc qui servira à représenter les images grayscale du film. Idéalement, cette échelle correspondrait à des valeurs très faibles (entre 0 et 1 par exemple), mais dans notre cas elle correspond à des valeurs de 0 à 10. Cette décision permet d'obtenir assez de variations de teintes dans l'intervalle et permettre de représenter correctement les images (si on prenait une échelle comprise entre 0 et 1, on aurait seulement 3 teintes : noir, blanc, et gris).

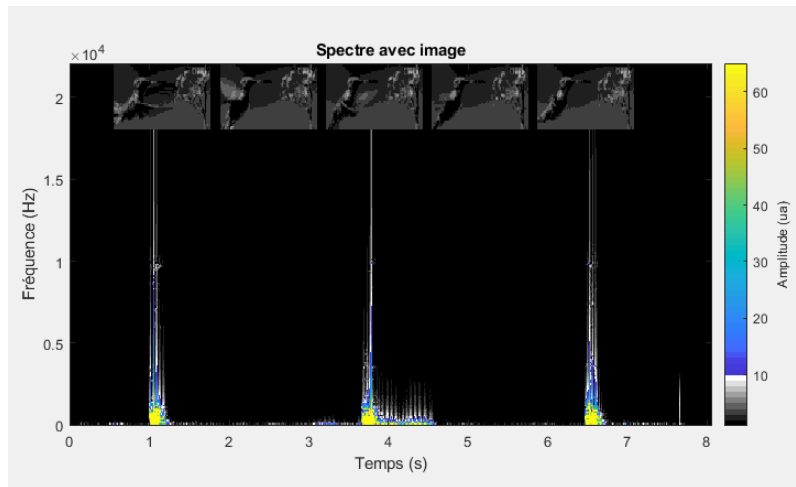


Figure 6: Spectre du fichier audio avec les images du film insérées

La seconde figure présente le spectre modifié, avec les images du films ajoutées. Différents paramètres permettent de choisir la largeur, la hauteur de l'image ainsi que le nombre d'images à ignorer entre chaque insertion (afin d'observer des changements notables). On voit sur la figure une nette progression du colibri, qui bat des ailes devant une fleur. L'ajout d'un décalage entre chaque image permet de les différencier et de les séparer.

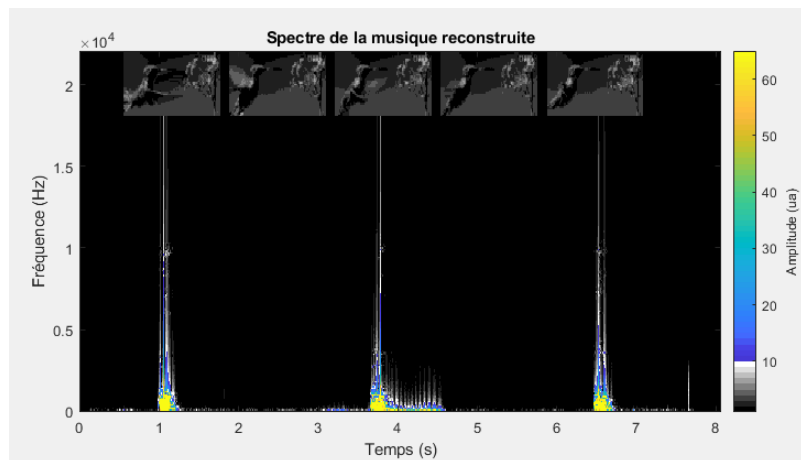


Figure 7: Spectre du fichier audio reconstruit

Cette dernière figure sert simplement à vérifier que la reconstruction s'est déroulée correctement et que l'on obtient bien le même spectre que celui utilisée pour reconstruire le fichier audio.

## 4 Problèmes rencontrés

Le problème principal rencontré dans ce projet est le compromis entre qualité sonore et qualité visuelle. En effet, afin d'obtenir assez de niveaux de gris pour représenter correctement une image, l'échelle de niveau de gris doit correspondre à une gamme d'amplitudes raisonnablement importante.

La conséquence néfaste de ce compromis est l'apparition d'un grésillement parasite dans le fichier audio reconstruit. Le son reste néanmoins audible et reconnaissable, ce qui est convenable.

Réduire la gamme d'amplitude améliore la qualité sonore, mais les images sont de moins en moins reconnaissables. Nous avons donc choisi d'utiliser des paramètres permettant d'offrir une représentation visuelle suffisante pour voir apparaître une évolution dans le film.

## 5 Conclusion

En conclusion, le projet est fonctionnel. En effet, nous avons réussi à incruster une image en niveau de gris d'abord, puis plusieurs dans le spectre d'un fichier audio, puis à reconstruire celui-ci correctement de telle sorte à ce qu'il soit intelligible, malgré des grésillements parasites. Nous nous sommes également intéressés à l'incrustation d'une image en couleur et avons eu des résultats convenables, mais ce travail n'étant pas demandé nous avons choisi de ne pas l'inclure.

De plus, ce projet nous a permis de découvrir ce qu'était un spectre audio, ainsi que l'intérêt d'une transformée de Fourier glissante dans le traitement d'un signal audio. Cela nous permet de mettre en pratique les aspects théoriques du cours dans un projet concret. Cela nous a également permis de prendre en main le logiciel Matlab et d'avoir un avant goût de ses fonctionnalités, notamment dans le domaine du traitement du signal.