

Rapport intermediaire pour Projet Long

Nicolas Peneloux, Ymri Scheiner

15 Mars 2024

1 Introduction

Notre projet consiste à réaliser un autotuneur, c'est-à-dire un programme qui reçoit en entrée un fichier audio d'une voix, et ressort en sortie une version de cet audio dont les notes sont modifiées selon les désirs de l'utilisateur, sans altérer le reste de l'audio. Grossièrement, nous pouvons diviser cela en deux :

- Pitch Tracking : récupérer les différentes notes de l'audio
- Pitch Correction : changer les notes vers les hauteurs souhaitées

Métriques

Pour mesurer le succès de notre projet, nous avons choisi une métrique de succès basée sur la distance entre les signaux de sortie produits par notre algorithme et ceux produits par une implémentation de référence. Plus précisément, nous définissons la mesure de succès comme suit :

$$\text{succes} = \max_{x \in \text{entrees_testees}} d(A(x), B(x)) \quad (1)$$

où A et B représentent respectivement notre algorithme et l'implémentation de référence, d est une fonction de distance entre les signaux audio, et *entrees_testees* est un tableau contenant les fichiers audio d'entrée à tester.

Nous définissons également un seuil maximal de distance autorisée entre les signaux de sortie de A et B , noté *seuil_max*. Si la mesure de succès est inférieure à ce seuil, nous considérons que notre projet est réussi.

Pour calculer la mesure de succès, nous parcourons tous les signaux audio d'entrée dans le *entrees_testees*. Pour chaque entrée, nous appliquons A et B pour produire deux signaux de sortie. Nous calculons ensuite la distance entre ces deux signaux de sortie à l'aide de d . Puis, nous mettons à jour la mesure de succès en prenant la valeur max entre la distance actuelle et la mesure de succès précédente.

Une fois que nous avons parcouru tous les signaux audio d'entrée, nous comparons la mesure de succès obtenue à notre seuil maximal *seuil_max*. Si la mesure de succès est inférieure à ce seuil, on considère que le test est passé correctement.

2 Implémentation

Nous réalisons ce projet en **Rust**, en nous appuyant sur l'**algorithme FFT (Fast Fourier Transform)** pour représenter l'audio sous une forme de données pragmatiques pour notre projet, une méthode très courante pour le traitement de signal. Nous utilisons également d'autres algorithmes basés sur cette technologie tels que l'algorithme PIN pour la correction de hauteur.

En termes de modules ou packages, la structure interne du projet est très simple, avec un fichier par tâche, à utiliser dans un main. Nous utilisons quelques packages de Rust pour certaines tâches auxiliaires au projet, telles que la création de l'interface graphique, ou la manipulation/création de fichiers MIDI.

3 Jalons

3.1 Tâches à Réaliser

Nous avons divisé le projet dans sa globalité en trois tâches principales, à savoir

- L'implémentation de l'algorithme FFT pour permettre de transformer des fichiers audio en données utilisables
- L'implémentation du pitch tracking
- L'implémentation de la pitch correction

Viennent ensuite s'ajouter des tâches auxiliaires telles que la mise en place de tests, ou encore la création de l'interface graphique.

3.2 Tâches Terminées

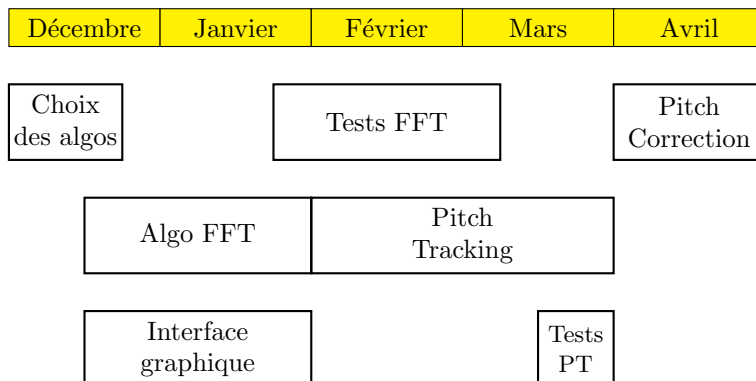
Les deux première tâches parmi celles énumérées au dessus ont été réalisés, ainsi que quelques tâches auxiliaires. Des tests ont été mis en place pour la partie FFT, et l'interface utilisateur a déjà été conçue.

3.3 Organisation Temporelle

Il manque ainsi la partie pitch correction et l'implémentation de tests pour la partie pitch tracking.

Idéalement, nous pouvons consacrer 2 semaines sur l'implémentations des tests et les semaines restantes pour coder la partie pitch correction.

Calendrier Visuel



Problèmes rencontrés

Nous avons d'abord mis du temps à trouver un sujet accepté, mais aussi après dans la réalisation du projet jusqu'à maintenant. Les difficultés majeures que nous avons rencontrées sont :

- L'utilisation de Rust, que nous avons tout les deux dû apprendre pour le cadre de ce projet. Cela à rajouter d'avantage de retard pour le projet, il était assez difficile de commencer le projet sans avoir un minimum de base solide pour bien débuter.
- La compréhension théorique des algorithmes et technologies que nous utilisons, c'était notre premier projet dans le domaine du traitement audio.
- Le choix des algorithmes à utiliser. En rapport avec le point précédent, les différences peuvent être nuancées et dure à saisir sans en connaître les tenants et aboutissants. Certains algorithmes sont plus ou moins détailler, certains peuvent être même plutôt récent, et sans s'y connaître dans le domaine, on peut vite se retrouver perdu.