UNIVERSITÉ
LAVAL

Transcription musicale automatique en python
Rapport Final

Antoine Veillette

Présenté à:
Simon Thibault
Simon Rainville

Dans le cadre du cours:
GPH-3001

Du département de génie physique, physique et d'optique
Faculté des sciences et de génie
Université Laval
2024-08-26

Table des matières

1. Sommaire

Ce rapport explore le développement d'une solution logicielle minimaliste pour la transcription musicale automatique, spécifiquement conçue pour simplifier le processus complexe et chronophage de transcription manuelle. Le projet vise à créer un programme capable de capturer et de transcrire des informations musicales de manière précise, en identifiant les notes, les rythmes, et les dynamiques à partir de divers types de sources audios. Après une revue des méthodes existantes et une étude approfondie des défis techniques, le rapport présente l'implémentation de plusieurs algorithmes, dont la détection polyphonique par spectre pseudo-2D et la reconnaissance d'accord par indice de similitude. Celui-ci est adapté à la transcription musicale polyphonique et monophonique ainsi que la reconnaissance d'accord. Les résultats expérimentaux démontrent l'efficacité du programme, tout en soulignant les domaines où des améliorations futures pourraient être apportées, notamment dans le posttraitement par réseaux de neurones. Le présent rapport conclut avec une discussion sur les performances comparatives du logiciel ainsi que les différentes voies d'amélioration de celui-ci.

2. Problématique

2.1. Problème et choix de la solution

La transcription musicale manuelle pose plusieurs défis majeurs. Elle exige une oreille musicale très développée, car il faut identifier et noter avec précision chaque note, accord, nuance rythmique et dynamique. Ce processus est chronophage, particulièrement pour les compositions complexes ou polyphoniques, où plusieurs notes sont jouées simultanément. De plus, il est susceptible aux erreurs humaines, ce qui peut entrainer des transcriptions inexactes. Les musiciens et transcripteurs doivent également posséder une solide compréhension de la théorie musicale. Pour ces raisons, la transcription manuelle est souvent considérée comme une tâche laborieuse et inaccessible à plusieurs. La transcription d'informations musicales est importante dans plusieurs contextes, notamment la production musicale par remixage, la gestion des droits d'auteur et des licences (celle-ci nécessite une version documentée), la capture d'idées pour les musiciens, etc.

Afin de simplifier la tâche, déjà colossale, des musiciens, nous avons décidé d'offrir une solution sous forme de programme informatique minimaliste. Ce programme offrira plusieurs options pour l'acquisition de donnée audio telles que : l'enregistrement spontané, téléchargement à partir d'une URL où alors à partir d'un fichier audio préexistant. Il pourra extraire les instruments pertinents à l'analyse par *Music Source Separation* (MSS). Il devra être capable d'identifier les notes et rythmes ainsi que les dynamiques (piano, pianissimo, forte, etc.). De plus, celui-ci devra pouvoir générer une partition au format PDF.

Le programme produit sera compatible avec macOS et devra être exempt de bogues. Il sera rapide et modulaire. À la fin du projet, le programme devrait être facilement modifiable et améliorable. Cette solution est en accord avec les contraintes en lien avec le développement des qualités de l'ingénieur:

- **Économique**: Elle est gratuite.
- **Sécurité**: Elle ne comporte aucun risque pour l'usager.
- **Environnement**: Elle implémente une solution optimisée, utilisant le moins de ressources informatiques possible

- **Éthique**: Le logiciel est disponible librement conformément à la licence GPL-3.
- **La conformité aux aspects règlementaires**: Le logiciel est testé avec une soixantaine de tests unitaires et d'intégrations.

2.2. Introduction et revue des connaissances

L'*Automatic Music Transcription* (AMT) est une tâche fondamentale dans le domaine de la recherche en musique numérique. Elle consiste à convertir un signal audio musical en une représentation symbolique, telle qu'une partition. Cette tâche est particulièrement complexe en raison de la nature polyphonique de la musique, où plusieurs notes peuvent être jouées simultanément. Les défis posés par l'AMT sont nombreux et incluent la détection des hauteurs multiples, l'identification des attaques et des durées des notes, ainsi que la reconstruction précise de la structure harmonique.

Les approches traditionnelles reposent souvent sur des techniques de modélisation acoustique, comme la *Non-Negative Matrix Factorization* (NMF) et les HMM. Cependant, ces méthodes montrent des limites, notamment en matière de précision et de capacité à capturer les dépendances temporelles et harmoniques complexes entre les notes. Des avancées récentes ont été réalisées avec l'introduction de modèles probabilistes, tels que les réseaux bayésiens dynamiques, qui permettent de modéliser à la fois les dépendances verticales (harmoniques) et horizontales (mélodiques) entre les notes [1].

Parallèlement, des techniques basées sur les réseaux de neurones, telles que les *Convolutional Recurrent Neural Network* (CRNN) et les architectures séquence à séquence (Seq2Seq), ont été proposées pour améliorer la transcription de la musique polyphonique en exploitant des représentations temporelles et spectrales plus robustes [2]. Ces approches modernes tendent à surmonter certaines des limitations des méthodes traditionnelles, offrant des résultats plus précis et une meilleure gestion de la complexité polyphonique dans les transcriptions musicales.

Dans une autre gamme de possibilités, l'analyse par spectre pseudo-2D[3] offre une alternative intéressante au bispectre couramment utilisé en traitement de signal, notamment dans le la détection d'ondes gravitationnelles et sysmiques[4]

En somme, bien que des progrès significatifs aient été réalisés et que certains problèmes de l'AMT soient considérés comme résolus tels que la détection monophonique, la transcription musicale automatique reste un domaine de recherche actif, avec de nombreux défis à surmonter pour atteindre une performance optimale, en particulier dans des contextes polyphoniques complexes.

3. Théorie associée

3.1. Exposé de la théorie

La musique est composée d'ondes acoustiques composant un ensemble fortement corrélé sur le plan rythmique et fréquentiel. Les notes générées par la plupart des instruments occidentaux tels que le piano et la guitare sont dites harmonique, c'est-à-dire qu'elles sont composées d'une fréquence fondamentale[1] et de partielles[2].

La hauteur nominale ou *chroma* ou alors *pitch class* en anglais, est une hauteur définie par son nom, indépendamment de l'octave dans laquelle elle se situe. La hauteur d'une note *pitch* correspond à un assemblage entre un « chroma » p. ex. « C » et d'une octave p. ex. « 4 ». Un *pitch* permet de différencier un « C4 » d'un « C5 » par exemple.

Comme vu dans la Fig. 1, une note jouée seule contient dans ses harmoniques, les fréquences associées à d'autres notes.

Harmonique	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
Fréquence (Hz)	32,7	65,4	98,1	130,8	163.5	196,2	228,9	261,6	294,3	327	359,7	392,4	425,4	457,5
Note$^{\text{octave}}$ de la gamme la plus proche	*do*$^{-1}$	*do*^{1}	*sol*^{1}	do^{2}	*mi*^{2}	*sol*^{2}	*si*^{2} ♭	*do*^{3}	*ré*^{3}	*mi*^{3}	*fa*^{3} ♯	*sol*^{3}	*sol*^{3} ♯	*la*^{3} ♯
Écart à la note la plus proche dans la gamme de tempérament égal (cents)	0	0	2	0	-14	2	-31	0	4	-14	-49	2	41	-31
Intervalle avec la fondamentale (cents)	0	1200	1902	2400	2786	3102	3369	3600	3804	3986	4151	4302	4440	4569

Fig. 1. – Fréquence des harmoniques d'un do^{-1} [6].

« Les instruments inharmoniques tels que le marimba, le vibraphone, les cloches et les timbales (timpani), contiennent des partielles non harmoniques, tout en donnant à l'oreille un bon sens de la hauteur. Les instruments qui ne génère pas de note, ou pas de note "définit", tels que les cymbales, les gongs ou les tamtams, rendent les sons riches en partiels inharmoniques. »[5]

Les notes de la musique occidentale sont accordées à « tempérament égal ».
« En musique, la **gamme tempérée**, aussi appelée tempérament égal, est un système d'accord qui divise l'octave en intervalles chromatiques égaux (c'est-à-dire que le rapport des fréquences de deux notes adjacentes est toujours le même) »[7]

[1] La plus basse fréquence.
[2] Les harmoniques partiels, où simplement, partielles, sont les fréquences qui sont des multiples entiers de la fréquence fondamentale, mais n'incluent pas la fondamentale.[5]

Ce ratio est égal à $2^{\frac{1}{12}}$, générant ainsi un espacement constant d'une fondamentale à l'autre sur l'échelle logarithmique .

La ***Constant Q Transform*** **(CQT)** est une transformée intégrale où le noyau de convolution est une fonction exponentielle complexe multiplié à une enveloppe (fonction de Hann) à laquelle on applique une série de filtres espacés également sur l'échelle logarithmique et ayant une largeur spectrale $\delta f_k = 2^{(1/n)^k} \delta f_{\min}$ où δf_k est la largeur spectrale du k-ième filtre, $f_{\min}$ est la fréquence centrale du filtre le plus bas et n est le nombre de filtres par octave. On l'appelle ainsi puisque le facteur de qualité, $Q = \frac{f_k}{\delta f_k}$ est constant.[8]

$$X_{CQT}(k) = \frac{1}{N_k} \sum_{n=0}^{N_k - 1} w(k,n) x(n) e^{-j2\pi Q n / N_k} \tag{1}$$

Où $w(k,n)$ est l'enveloppe du k-ième filtre et N_k, la longueur de la fenêtre du filtre.

Un **HMM** est un modèle statistique utilisé pour représenter des systèmes où l'on observe une séquence de résultats, mais où les états sous-jacents qui produisent ces résultats sont invisibles ou « cachés ». Ce modèle est constitué d'états cachés reliés par des probabilités de transition, et chaque état caché peut générer une observation selon une certaine probabilité. Les HMM sont très utilisés en reconnaissance vocale et en traitement du langage naturel.

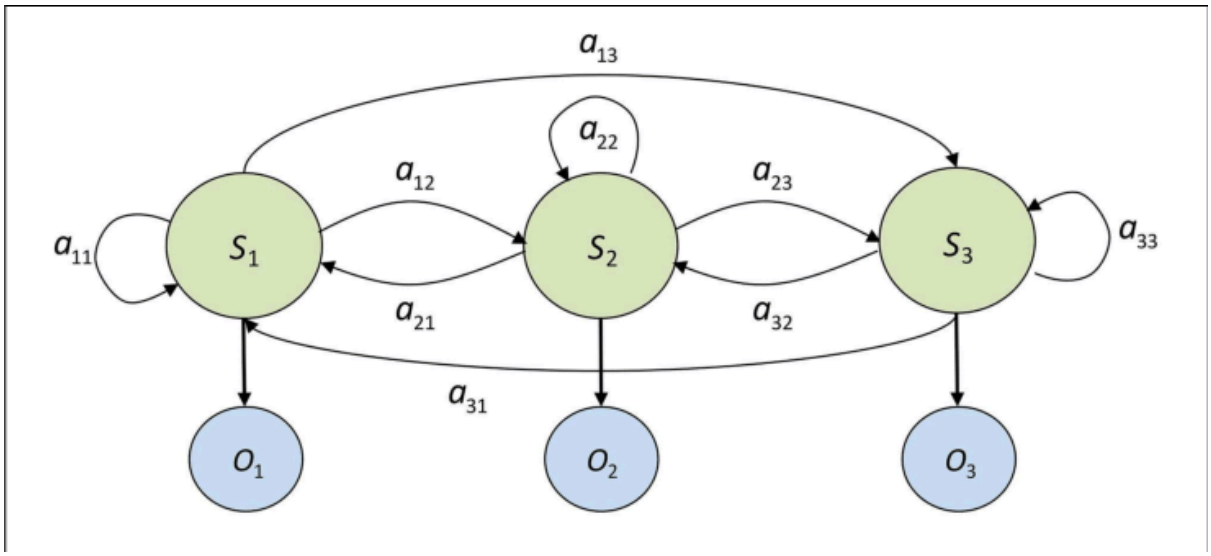


Fig. 2. – Représentation schématisée d'un HMM simple à 3 états où $\mathcal{S}$ est l'ensemble des états et $\mathcal{O}$, l'ensemble des observations[9].

Le processus d'inférence consiste à retrouver la séquence d'état caché qui a produit une séquence d'observation. L'algorithme utilisé pour l'inférence se nomme Viterbie.

PYIN est un algorithme de reconnaissance de hauteur de note (*pitch*) approprié pour la reconnaissance d'information musicale dans un signal monophonique. L'algorithme est disponible par l'intermédiaire de la librairie python **librosa**[10].

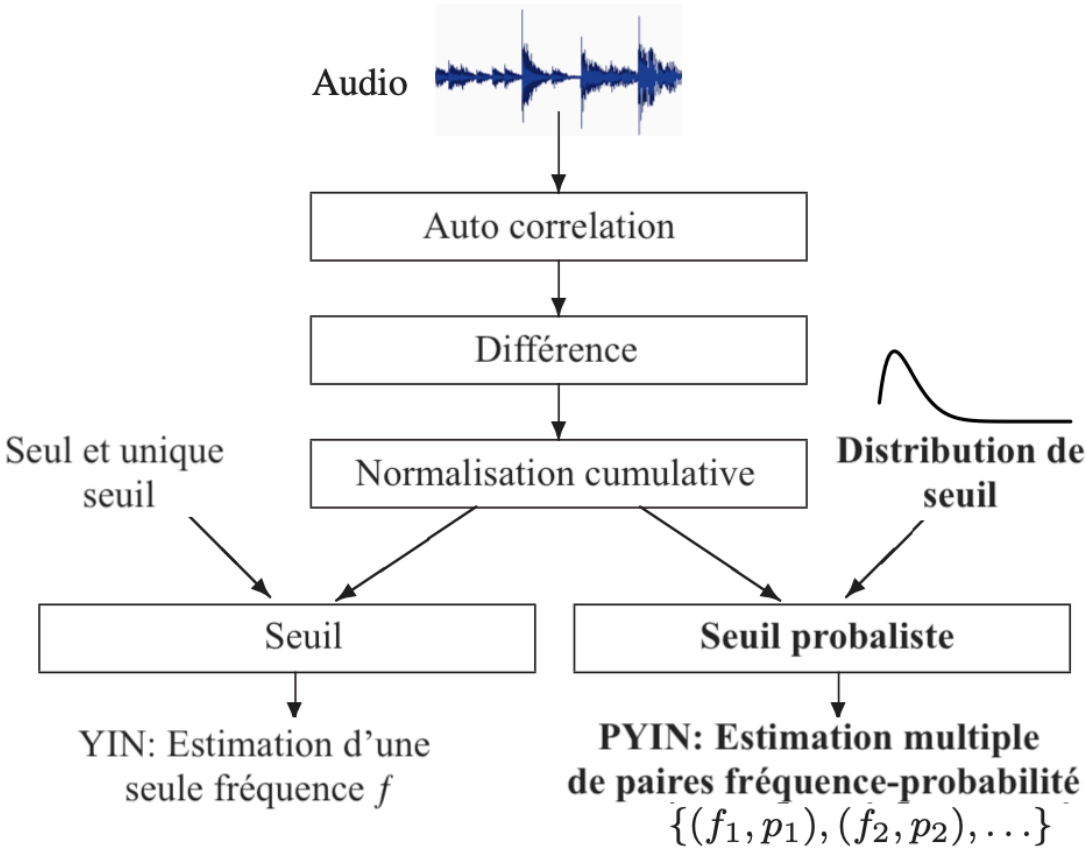


Fig. 3. – Les étapes de l'algorithme PYIN, voir [11] pour plus d'information. (schéma tiré de [11])

Le spectre **pseudo-2D** (voir [3]) se définit comme :

$$P_x(f_1, f_2) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x(t)x^*(\tau)e^{-j2\pi(f_1 t - f_2 \tau)} \, \mathrm{d}t \, \mathrm{d}\tau$$
$$P_x(f_1, f_2) = X(f_1)X^*(f_2) \tag{2}$$

Soit le produit cartésien du spectre avec son conjugué complexe.

3.2. Application au problème

Initialement, nous ne connaissions rien au domaine de la transcription musicale automatique. C'est après la lecture de [5] et de [12] que nous avons eu une idée du chemin sur lequel s'enligner. Dans [12], il est mention d'un modèle HMM à 3 états par note (silence, attaque, soutien) pour la récupération d'information musicale monophonique. Cette idée fut réutilisée par [13] puis par nous même (avec inspiration du GitHub de [13]) pour l'implémentation de la solution monophonique. Une variante de ce modèle avec un état supplémentaire par dynamique (piano, pianissimo, forte …) avait l'intention d'être effectuée, mais a dû être mise de côté par manque de temps.

La section sur la reconnaissance polyphonique est basée sur le document [3] avec quelques variantes, notamment, notre solution n'utilise pas de filtre *Nearest Neighbors* (NN) en posttraitement puisque cette approche générait des résultats encore plus mauvais. Il y a assurément

moyen d'implémenter une solution par posttraitement NN qui améliore les résultats, mais nous avons dû nous en tenir à ce que nous avions par manque de temps.

La solution par reconnaissance d'accord utilise la solution exposée à la fin du chapitre 5 du livre *Fundamental of Music processing*[5]. La lecture de ce livre fut une étape importante du processus d'ingénierie. Les détails du procédé sont davantage expliqués dans le Chapitre 3.2.2.

3.2.1. PYIN pour l'estimation de hauteur de note monophonique

La section du logiciel qui s'occupe de l'analyse monophonique utilise l'algorithme PYIN[11]. Cet algorithme est considéré comme étant le *State Of The Art* (SOTA) pour la reconnaissance monophonique. Les probabilités obtenues par PYIN ont été combinées aux attaques détectées par la fonction `onset_detect()` de librosa pour initialisé une matrice d'observation à priori. Cette matrice d'observation est par la suite utilisée conjointement avec une matrice de transition uniforme dans l'algorithme de Viterbie pour inférer[3] la hauteur des notes jouées.

3.2.2. Reconnaissance d'accord par indice de similarité et posttraitement par HMM

Essentiellement, on obtient un indice de similarité qui sera ensuite utilisé dans un HMM pour obtenir le résultat final. Dans notre cas, le HMM est utilisé comme une sorte de filtre informé. Par simplicité, on utilise le produit scalaire de vecteur normalisé comme indice de similarité. On définit x, un accord exprimé comme un vecteur dans la base des chroma $\mathcal{N}$, $x \in \{C, C\#, D, D\#, E, F, F\#, G, G\#, A, A\#, B\}$ et y, un vecteur observé dans cette même base.

$$s(x, y) = \frac{\langle x|y\rangle}{\|x\| \cdot \|y\|} \tag{3}$$

Puis on utilise la matrice d'indice de similarité dans le HMM comme une matrice de probabilité d'émission. Dans notre cas, on souhaite retrouver les états à partir des observations. La matrice d'observation identifie la probabilité d'être dans un état k sachant l'observation i, ce que nous appellerons, la probabilité d'émission :

$$\mathbb{P}[S_k|O_i] = s(S_k, O_i) \tag{4}$$

et la matrice de transition exprimant la probabilité de passé d'un état $S_i \in \{C, C\#, D, ..., B7\}$ à un état $S_j \in \{C, C\#, D, ..., B7\}$ où chaque état représente un accord:

$$T = \begin{pmatrix} \alpha_{C,C} & \alpha_{C,C\#} & \cdots & \alpha_{C,B7} \\ \alpha_{C\#,C} & \alpha_{C\#,C\#} & \cdots & \alpha_{C\#,B7} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{B7,C} & \alpha_{B7,C\#} & \cdots & \alpha_{B7,B7} \end{pmatrix} \tag{5}$$

La meilleure suite d'état sera celle dont la probabilité totale sera maximale:

$$\text{Max}\left\{ \prod_{i=1}^{N} \underbrace{\mathbb{P}[T_i|T_{i-1}]}_{\text{Transition}} \prod_{i=1}^{N} \underbrace{\mathbb{P}[S_i|O_i]}_{\text{Émission}} \right\} \tag{6}$$

[3]Le processus d'inférence consiste à retrouver la séquence d'état caché qui a produit une séquence d'observation.

où N est le nombre d'évènements considérés dans l'analyse (proportionnel à la durée d'analyse). L'algorithme de Viterbi permet de trouver efficacement[4] la suite d'état la plus probable en utilisant des concepts de programmation dynamique.

3.2.3. Spectre pseudo-2D pour l'estimation polyphonique

La technique du spectre pseudo-2D est une approche novatrice conçue pour résoudre le problème complexe de l'estimation de la hauteur dans la musique polyphonique, où plusieurs notes sont jouées simultanément. Les méthodes traditionnelles rencontrent souvent des difficultés en raison du chevauchement des harmoniques de ces notes, ce qui complique la détection des hauteurs. La méthode du spectre pseudo-2D améliore ces techniques en transformant un signal temporel en un espace fréquentiel bidimensionnel, permettant ainsi une séparation plus efficace des harmoniques qui se chevauche.[3]

Le processus commence par la CQT, qui convertit le signal temporel en une représentation temps-fréquence avec une échelle logarithmique des fréquences. Cette transformation permet une résolution fréquentielle variable, particulièrement utile dans l'analyse musicale où les intervalles de hauteur sont logarithmiques. Dans le cadre de notre logiciel, nous avons choisi 3 conteneurs par note, telle qu'effectuée dans [3]. Les données résultantes sont ensuite utilisées pour construire le spectre pseudo-2D.

Une fois le spectre pseudo-2D construit, l'étape suivante consiste à retrouver la corrélation croisée entre le spectre pseudo-2D et un modèle harmonique 2D (Fig. 4) prédéfini, représentant la structure harmonique des notes dans le plan fréquentiel. Les pics spectraux qui correspondent au modèle et dépassent un certain seuil sont identifiés comme des hauteurs potentielles. Cette méthode est particulièrement efficace, car elle réduit l'impact du chevauchement des harmoniques, un problème courant dans la musique polyphonique.

[4] $O(Np^2)$ au lieu de $O(p^N)$ où N est la longueur de l'échantillon et p, le nombre d'états considéré

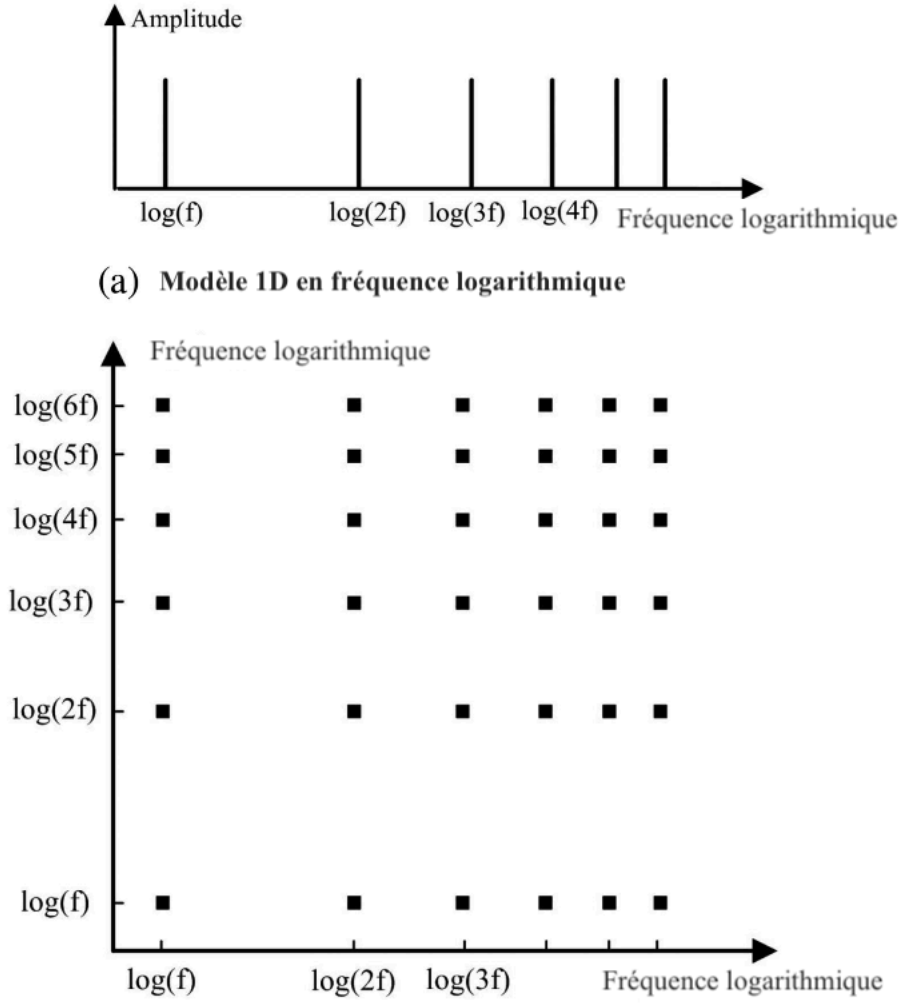


Fig. 4. – Modèle harmonique utilisé pour la corrélation croisé avec le spectre pseudo-2D (schéma tiré de [3])

Après l'identification des hauteurs potentielles, des étapes de posttraitement sont employées pour affiner les résultats. Cela comprend la formation de contours de notes en regroupant les estimations de hauteurs adjacentes qui sont continues dans le temps. Les hauteurs isolées qui ne répondent pas à certains critères (tels que la durée minimale des notes) sont écartées. La durée minimale d'une note est d'environ 80ms. Le record de la note la plus courte est d'environ 73ms[14].

4. Description du logiciel

Le logiciel se sépare en trois modes: monophonique, polyphonique et reconnaissance d'accord. L'interface utilisateur ainsi que les options du logiciel sont affichées dans le Appendix B, Appendix C et le Appendix D respectivement. Une fois un mode sélectionné, on fournit un échantillon audio avec l'une des méthodes disponibles (fichier, enregistrement ou URL) et on obtient une partition en PDF ou alors un rouleau de piano[5] en sortie. Il est aussi possible d'extraire un instrument de l'échantillon avant l'analyse par l'intermédiaire du logiciel Demucs[15]. Cette option est implémentée dans le logiciel par l'intermédiaire de l'option `--extract`. Si on ne spécifie pas l'option `--piano-roll`, le logiciel produira, par défaut, une partition PDF sous format de fichier temporaire. Ce PDF doit être enregistré par l'utilisateur s'il souhaite le conserver, autrement, il sera détruit automatiquement une fois fermée. L'option `--debug` sert à obtenir un graphique identique à Fig. 7. Cette option permet essentiellement de regarder ce qui se passe à un temps donné en termes de spectre pseudo-2D.

Le logiciel est testé sur une soixantaine de tests unitaires et d'intégrations assurant le bon fonctionnement de celui-ci. L'implémentation d'un *workflow* GitHub fait en sorte que le bon fonctionnement du logiciel est testé à chaque fois que du code supplémentaire est ajouté à la branche principale. Si les tests ne passent pas, le code n'est pas ajouté et un courriel est envoyé à la personne responsable de la maintenance du code.

Un exemple de partition produite peut être visualisé à la Fig. 9.

[5]Un rouleau de piano est un graphique exprimant les hauteurs de notes jouées pour chaque temps.

5. Résultats expérimentaux

Quelques définitions:

$$F_{\text{mesure}} = \frac{\#\text{TP}}{2\#\text{TP} + \#\text{FP} + \#\text{FN}} \tag{7}$$

où: #TP est le nombre de vrai positif, #FP, le nombre de faux positifs, #FN, le nombre de faux négatif et #TN, le nombre de vrai négatif.

$$\text{Rappel} = \frac{\#\text{TP}}{\#\text{TP} + \#\text{FN}} \tag{8}$$

$$\text{Précision} = \frac{\#\text{TP}}{\#\text{TP} + \#\text{FP}} \tag{9}$$

La performance du logiciel en mode polyphonique est testée sur la base de données *Maestro*[16] avec l'aide de la librairie *Mir eval*[17]. Les échantillons audios de la librairie proviennent de véritable performance jouée ***Live*** avec leur version en format midi correspondant. Les échantillons midi sont garantis ajustés à 3ms près de leur version audio[16].

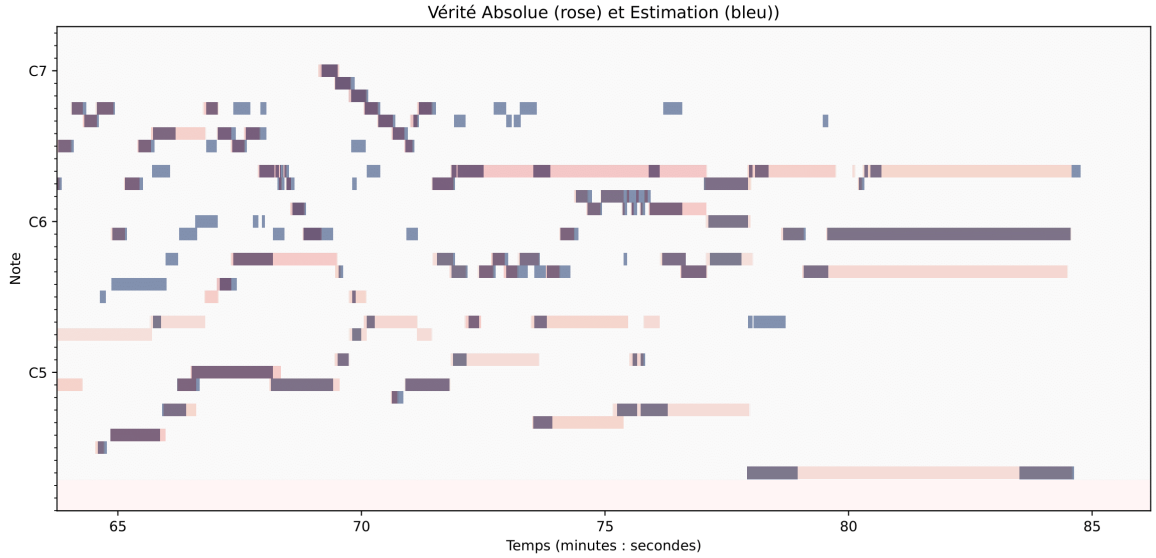


Fig. 5. – Les vrais positifs correspondent au mauve, les faux négatifs sont en rose et les faux positifs en bleu. La figure complète se trouve à Fig. 6

Afin d'améliorer les performances de la reconnaissance polyphonique, différents modèles de note ont été testés. Un modèle initialisé à partir d'un échantillon audio, un modèle qui décroit de 4.7dB par harmonique et un modèle où la fréquence fondamentale et les harmoniques ont tous une amplitude égale. Les résultats se trouvent dans le Tableau 1.

Métriques	Modèle A3 (MAESTRO)	Modèle −4.7dB (MAESTRO)	Modèle uniforme (MAESTRO)	Zhang (MAPS) [3]
Précision	0.5793	**0.6109**	0.5763	-
Rappel	0.5448	**0.5620**	0.5270	-
Exactitude	0.3904	**0.4139**	0.3798	-
Erreur de Substitution	0.1937	**0.1644**	0.1787	0.17
Erreur d'Omission	0.2614	0.2736	0.2942	**0.17**
Erreur de Fausses Alarmes	0.2019	**0.1936**	0.2087	0.23
Erreur Totale	0.6571	0.6315	0.6817	**0.57**
Précision Chroma	0.6572	**0.6611**	0.6322	-
Rappel Chroma	**0.6180**	0.6082	0.5781	-
Exactitude Chroma	**0.4674**	0.4637	0.4326	-
Erreur de Substitution Chroma	0.1205	**0.1182**	0.1276	-
Erreur d'Omission Chroma	**0.2614**	0.2736	0.2942	-
Erreur de Fausses Alarmes Chroma	0.2019	**0.1936**	0.2087	-
Erreur Totale Chroma	**0.5839**	0.5854	0.6306	-
F_{mesure}	0.5615	0.5855	0.5505	**0.59**

Tableau 1. – Comparaison des 3 modèles de note testés avec les résultats obtenus par [3] selon différentes métriques.[6]

Les métriques n'ont pas été évaluées pour les modes monophoniques et la reconnaissance d'accord, mais celle-ci sont comparable aux métriques exposées dans [11] et ainsi que dans le chapitre 5 de *fundamentals of music processing* [5].

[6]Les paramètres utilisés pour l'analyse sont détaillés dans le Tableau 3.

6. Commentaires sur les résultats et respect du cahier des charges

Les résultats démontrent, sans équivoque, que le modèle uniforme (entre autres utilisé par Zang dans [3]) est le pire des modèles et qu'il est plus efficace d'utiliser le modèle qui décroit de 4.7dB par harmonique. Surprenament, le modèle qui correspond au véritable timbre pour un A3 sur un piano a moins bien performé que celui qui décroit de 4.7 dB par harmonique. Néanmoins, nous n'avons pas réussi à surpasser les résultats obtenus dans [3] tels que vus dans le Tableau 1. Nous attribuons cette défaite au fait que le filtre NN n'a pas été implémenté correctement[7].

Avec du recul, les critères dans le cahier des charges ont été définis de façon peu rigoureuse et redondante. Au début du projet, nous n'avions pas les connaissances que nous avons maintenant sur le sujet en matière de métriques. Par exemple, la bonne métrique à utiliser aurait été la F_{mesure} pour délimiter l'efficacité du logiciel. Par exemple, le critère de séparation d'instrument aurait pu être quantitatif au lieu d'être qualitatif (nous aurions pu utiliser la SDR). La SDR est la métrique utilisée par le *Sound Demixing Challenge*[18]. Mais bon, de toute façon nous n'avons pas implémenté notre propre MSS, par manque de temps, nous avons utilisé *Demucs*[15] dans le cadre de ce projet.

Pour l'analyse monophonique, on considère que tous les critères du cahier des charges sont respectés puisque PYIN est le SOTA de l'analyse monophonique avec un F_{mesure} moyen de ~98%[11]. La détection des dynamiques ainsi que les détections d'articulation ne sont pas implémentées, et ce, pour tous les modes d'analyse. Le cahier des charges pour l'analyse polyphonique est rempli et disponible dans Tableau 2

[7]Les résultats étaient pires avec le filtre, nous avons manqué de temps et avons décidé de ne pas l'implémenter à l'algorithme pour cette raison.

Cahier des charges

Critère à évaluer	**Condition générale à respecter**	**Oui/ Non**
Reconnaissance des rythmes	Le bon rythme au moins 80% du temps	Non
Reconnaissance des notes	La bonne note au moins 80% du temps	Non
Identifier le bon tempo	Identifie le bon tempo à 2% de précision	Oui
Transcription des dynamiques	Les variations de volume et d'intensité bien capturées	Non
Reconnaissance des articulations	Capturer les accents, staccatos, et autres articulations au moins 70% du temps	Non
Précision de la transcription polyphonique et reconnaissance d'accord	Identifier correctement les notes jouées simultanément dans un ensemble polyphonique au moins 70% du temps	Non
Séparation des instruments	Séparer les instruments présents dans l'échantillon audio	Oui
Exportation au format PDF	Exporter la partition transcrite en un format PDF lisible et correct	Oui
Interface utilisateur	Facilité d'utilisation et clarté de l'interface utilisateur pour charger des fichiers audios et visualiser les résultats	Oui

Tableau 2. – Cahier des charges du projet.

7. Conclusion

Pour résumé, ce projet a été une superbe introduction au monde de l'AMT. Le logiciel fonctionne, mais n'implémente pas toutes les fonctionnalités qu'il devait implémenter initialement (tel que la reconnaissance de dynamique et d'articulations). Malgré que l'analyse polyphonique par spectre pseudo-2D donne des résultats comparables à ceux obtenus dans le rapport original (voir Tableau 1). Il serait intéressant d'utiliser les résultats de la corrélation croisée du spectre pseudo-2d en entrée dans un *Dynamic Bayesian Network* (DBN) ou alors un CRNN comme celui présenté dans [2] afin d'améliorer les performances du logiciel. Par manque de temps, nous avons dû laisser tomber cette piste, mais il y a fort à parier qu'un tel posttraitement pourrait grandement améliorer les performances du mode polyphonique. Sur une autre lancée, nous suspectons que l'évaluation des images non voisée pourrait, elle aussi, être améliorée sans trop de difficulté à l'aide d'un HMM à deux états cachés (activé/désactivé) qui pourraient, entre autres, utiliser en entrée l'écart type et l'énergie du signal. Cette manière de faire permettrait au mode polyphonique d'être plus robuste aux variations présentes dans les échantillons audios fournis en entrée tout en ouvrant la voie à la détection des dynamiques en utilisant ces mêmes propriétés.

Glossaire

AMT	*Automatic Music Transcription*
CQT	*Constant Q Transform*
CRNN	*Convolutional Recurrent Neural Network*
DBN	*Dynamic Bayesian Network*
HMM	*Hiden Markov Model*
MSS	*Music Source Separation*
NMF	*Non-Negative Matrix Factorization*
NN	*Nearest Neighbors*
PYIN	*Probalistic* Yin
SOTA	*State Of The Art*

A Interface utilisateur

```
usage: mir [-h] {monophonic,polyphonic,chord-only} ...

Automatic music transcription & identification for musicians.

options:
-h, --help            Show this help message and exit

Analysis modes:   {monophonic,polyphonic,chord-only}
monophonic          Monophonic mode
polyphonic          Polyphonic mode
chord-only          Chord-only mode

Music is the arithmetic of sounds as optics is the geometry of light. -Claude Debussy
```

B Monophonique

```
usage: mir monophonic [-h] [-pr] [-e <guitar|bass|piano|vocals|other>] (-u URL | -r | -f FILE)
[-o OUTPUT]

options:
-h, --help            Show this help message and exit
-pr, --piano-roll     Show piano roll visualization
-e <guitar|bass|piano|vocals|other>, --extract <guitar|bass|piano|vocals|other>
Extract the audio of an instrument using Music Source Separation
-u url, --url url     URL to the music file
-r, --recording       Record audio from microphone
-f file, --file file  Path to the music file with .wav extension
-o output, --output output
Output file name
```

C Polyphonique

```
usage: mir polyphonic [-h] [-pr] [-e <guitar|bass|piano|vocals|other>] [-b <path/to/midi/
file.midi>] [-t [0-1]] [-g [1-inf]]
[-std <float>] [-d <time in seconds>] [-u URL] [-r] [-f FILE] [-o OUTPUT]

options:
-h, --help            Show this help message and exit
-pr, --piano-roll     Show piano roll visualization
-e <guitar|bass|piano|vocals|other>, --extract <guitar|bass|piano|vocals|other>
Extract the audio of an instrument using Music Source Separation
-b <path/to/midi/file.midi>, --benchmark <path/to/midi/file.midi>
Compare against ground truth from midi file
-t [0-1], --threshold [0-1]
Threshold for the detection of note in with respect to the highest correlation value in a frame
-g [1-inf], --gamma [1-inf]
Gamma factor used in logarithmic compression, higher values increase the sensitivity
-std <float>, --standard-deviation <float>
Standard deviation threshold used to determine if a frame is voiced or not, 1e-6 work best for
polyphonic piano while 1e-3 work best for noisy guitar recording
-d <time in seconds>, --debug <time in seconds>
Debug a certain time frame, will show the cross-correlation with the template matrix and pseudo2d
spectrum
-u url, --url url     URL to the music file
-r, --recording       Record audio from microphone
-f file, --file file  Path to the music file with .wav extension
```

```
-o output, --output output
Output file name
```

D Reconnaissance d'accord

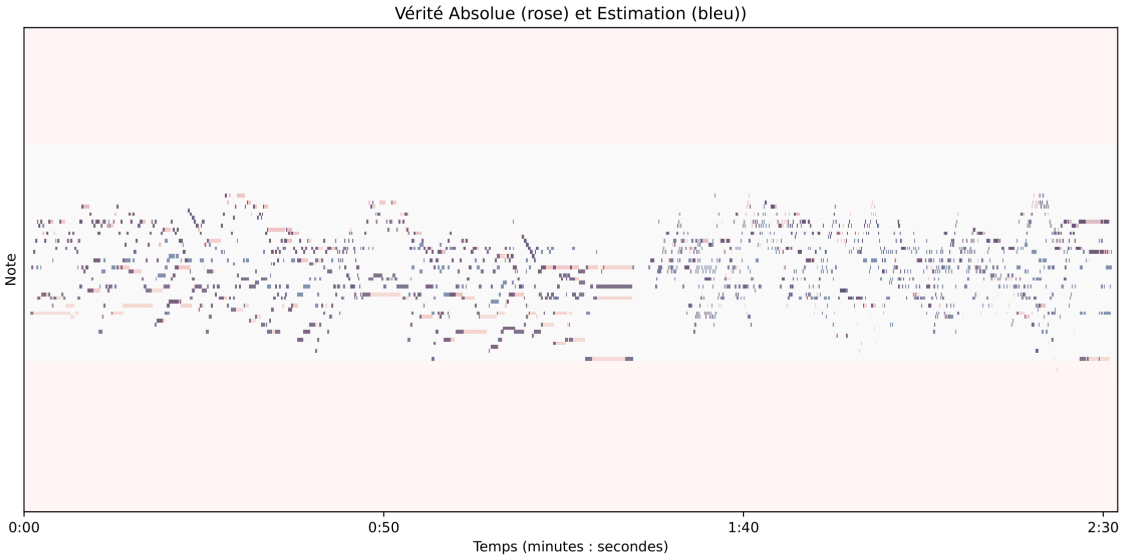
```
usage: mir chord-only [-h] [-pr] [-e <guitar|bass|piano|vocals|other>] (-u URL | -r | -f FILE)
[-o OUTPUT]

options:
-h, --help            Show this help message and exit
-pr, --piano-roll     Show piano roll visualization
-e <guitar|bass|piano|vocals|other>, --extract <guitar|bass|piano|vocals|other>
Extract the audio of an instrument using Music Source Separation
-u url, --url url     URL to the music file
-r, --recording       Record audio from microphone
-f file, --file file  Path to the music file with .wav extension
-o output, --output output
Output file name
```

E Figures et tableaux

Gamma (compression logarithmique)	500
Seuil	0.64
Seuil écart type	10^{-6}
Longueur minimale	$\sim$ 60ms soit 3 images[8]
Nombre d'harmonique	3

Tableau 3. – Paramètres utilisées pour l'analyse.



[8] *frame*

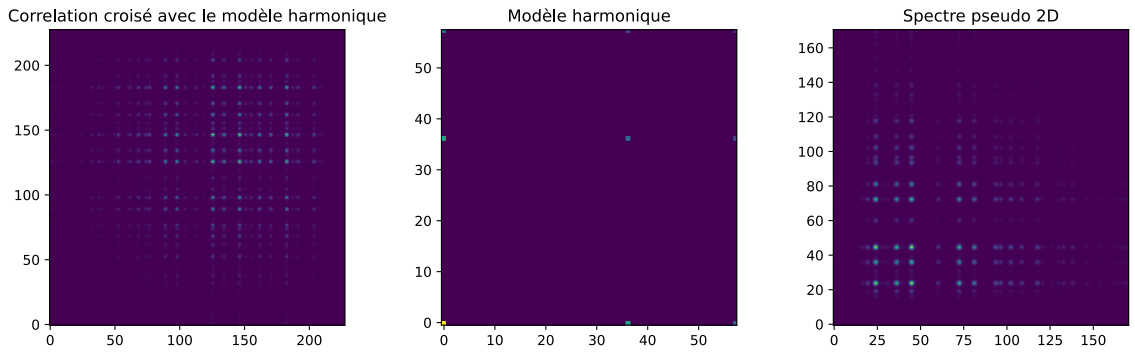


Fig. 7. – Correlation croisée, spectre pseudo-2D et modèle harmonique à trois harmoniques[9] pour une triade en do majeur joué sur une guitare classique.

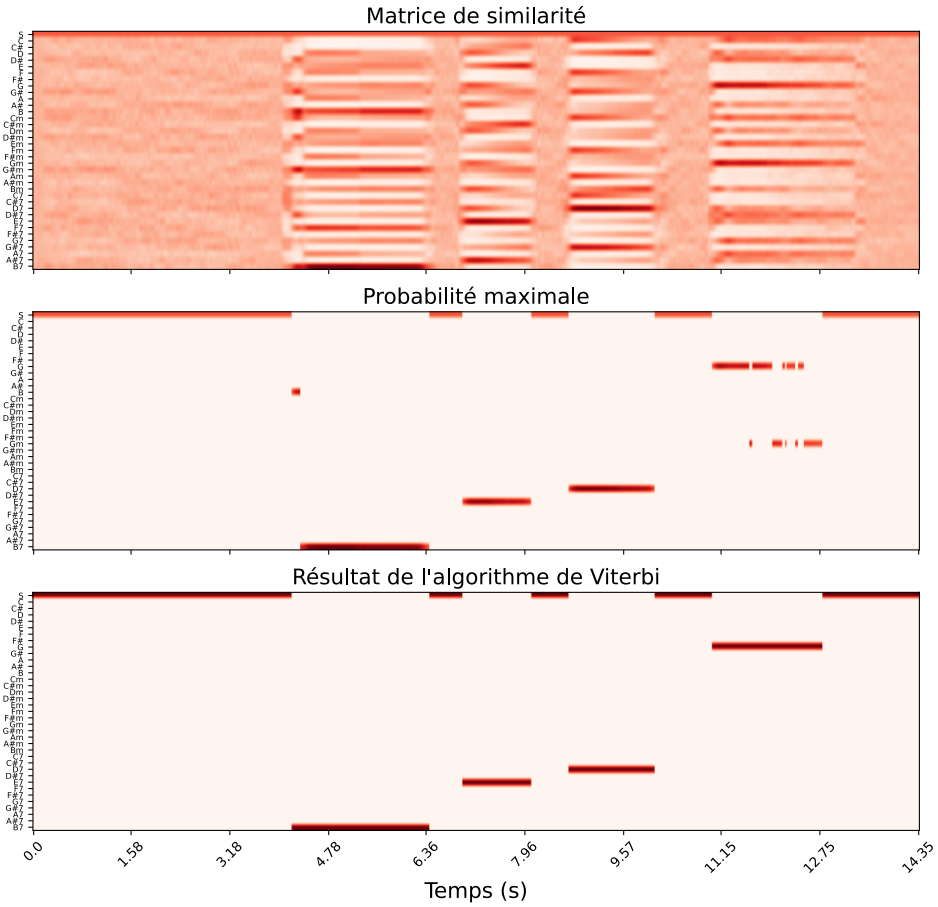


Fig. 8. – Identification correcte de la suite d'accord (B7,E7,D7,G) joué sur une guitare classique enregistré avec un téléphone cellulaire.

[9] –4.7 dB par harmonique

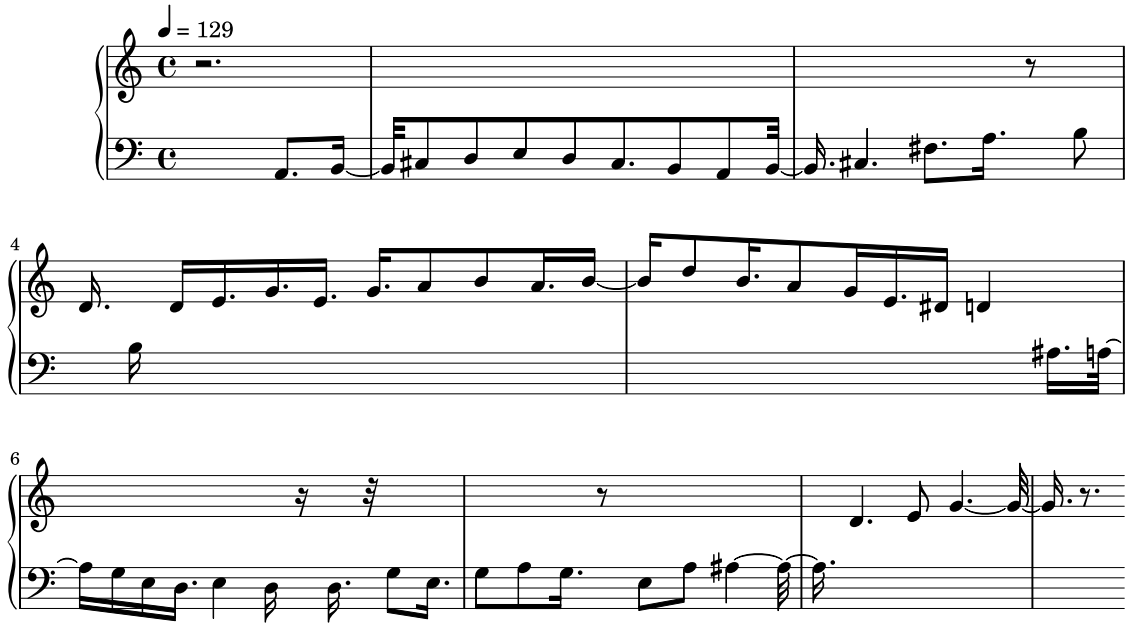


Fig. 9. – Exemple d'une partition produite automatiquement avec le mode Monophonique du logiciel.

F GitHub du projet

Le GitHub du projet se trouve ici [19].

Bibliographie

[1] Raczyński, S. A., Vincent, E., et Sagayama, S., « Dynamic Bayesian Networks for Symbolic Polyphonic Pitch Modeling », *IEEE Transactions on Audio, Speech, and Language Processing*, Vol. 21, N° 9, 2013, p. 1830-1840.. https://doi.org/10.1109/TASL.2013.2258012

[2] Liu, L., Morfi, V., et Benetos, E., « Joint Piano-roll and Score Transcription for Polyphonic Piano Music », 2020.

[3] Zhang, W., Chen, Z., et Yin, F., « Multi-Pitch Estimation of Polyphonic Music Based on Pseudo Two-Dimensional Spectrum », *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, Vol. 28, 2020, p. 2095-2107.. https://doi.org/10.1109/TASLP.2020.3007794

[4] contributors, W., « Bispectrum », 2024.. https://en.wikipedia.org/wiki/Bispectrum

[5] Müller, M., « Fundamentals of Music Processing: Audio, Analysis, Algorithms, Applications », Springer International Publishing, 2016.

[6] . https://fr.wikipedia.org/wiki/Harmonique_(musique)

[7] iMusic-School, « Tempérament égal », 2024.. https://www.imusic-school.com/blog/fr/encyclopedie/theorie/temperament-egal/

[8] contributors, W., « Constant-Q transform », 2024.. https://en.wikipedia.org/wiki/Constant-Q_transform

[9] . https://www.quantconnect.com/docs/v2/research-environment/applying-research/hidden-markov-models

[10] McFee, B., Raffel, C., Liang, D., Ellis, D. P., McVicar, M., Battenberg, E., et Nieto, O., « librosa: Audio and music signal analysis in python », Vol. 8, 2015.

[11] Mauch, M., et Dixon, S., « PYIN: A fundamental frequency estimator using probabilistic threshold distributions », Vol. 0, 2014, pp. 659-663.. https://doi.org/10.1109/ICASSP.2014.6853678

[12] Benetos, E., Dixon, S., Giannoulis, D., Kirchhoff, H., et Klapuri, A., « Automatic music transcription: challenges and future directions », *Journal of Intelligent Information Systems*, Vol. 41, 2013, p. 407-434.. https://api.semanticscholar.org/CorpusID:207169189

[13] Tavares, T. F.. https://github.com/tiagoft/audio_to_midi/blob/master/sound_to_midi/monophonic.py

[14] Bubble, P., « How Fast Can Pianists Play? », 2024.. https://pianobubble.com/how-fast-can-pianists-play/

[15] Défossez, A., « Hybrid Spectrogram and Waveform Source Separation », 2021.

[16] Hawthorne, C., Stasyuk, A., Roberts, A., Simon, I., Huang, C.-Z. A., Dieleman, S., Elsen, E., Engel, J., et Eck, D., « Enabling Factorized Piano Music Modeling and Generation with the MAESTRO Dataset », 2019.

[17] Raffel, C., McFee, B., Humphrey, E. J., Salamon, J., Nieto, O., Liang, D., et Ellis, D. P. W., « mir_eval: A Transparent Implementation of Common MIR Metrics », 2014.

[18] AIcrowd, « Sound Demixing Challenge 2023 », 2023.. https://www.aicrowd.com/challenges/sound-demixing-challenge-2023

[19] craqu, « GPH-3001 », 2024.. https://github.com/craqu/GPH-3001