SYNTHÈSE CONCATÉNATIVE EN RÉALITÉ VIRTUELLE

Jonathan Bell

SYNTHÈSE CONCATÉNATIVE EN RÉALITÉ VIRTUELLE

Jonathan Bell
Aix Marseille Univ, CNRS, PRISM, Marseille, France
belljonathan50@gmail.com

RÉSUMÉ

Cette étude examine comment la métaphore d'un espace tridimensionnel peut être utilisée de manière créative pour la composition instrumentale. De nombreux outils de synthèse sonore concaténative permettent de représenter dans un espace à n dimensions de grandes quantités de sons, affichant ainsi sur une carte (2d) ou dans un espace (3d) des sons alors décorrélés de leur agencement dans le temps. Si le potentiel de ces systèmes pour créer des instruments interactifs est une évidence, leur statut en tant que partitions musicales reste à évaluer plus en détail, car la dimension temporelle y est absente. En réalité virtuelle, une telle représentation de données sonores - la partition - devient une carte (ou un monde) tridimensionnel dans lequel l'utilisateur navigue librement. L'expérimentation de ce dispositif par le biais de la composition, de la conception d'instruments et de l'improvisation a montré un potentiel de simulation d'instruments acoustiques plausibles, en utilisant des techniques d'apprentissage machines pour modéliser des instruments virtuels à partir de quantités relativement faibles de données (par exemple, 20 minutes d'audio pour modéliser une clarinette). Cette méthode offre des possibilités prometteuses pour l'exploration de fragments instrumentaux regroupés par timbre, registre, dynamique et techniques instrumentales. Que ces cartes s'identifient ou non à des partitions musicales, elles contribuent à répondre à un problème esthétique formulé par Lev Manovitch : "comment fusionner base de données et narration dans une nouvelle forme".

1. INTRODUCTION

1.1. Are scores maps ?

La question soulevée dans cet article est en grande partie inspirée par une étude de Daniel Miller, *Are scores maps ? A cartographic response to Goodman* [1], dans laquelle une tension dialectique entre deux concepts (partition et carte) conduit à un questionnement sur le rôle et la fonction de la notation musicale. Miller propose que, malgré les conventions divergeantes en apparence, la structure sous-jacente des partitions est étroitement liée à celle des cartes : "*Les composantes notationnelles des partitions sont mieux comprises comme des caractéristiques de surface contingentes, renforcées par une structure représentationelle sous-jacente de type carte. Dans cette optique, les partitions sont considérées comme des cartes hautement codifiées, et les symboles notationels des partitions ne constituent qu'un des multiples modes de représentation exploités par ce cadre*". Les partitions peuvent donc être conçues comme un simple sous-ensemble de cartes. Le rapprochement entre ces deux notions est mieux illustré par un exemple historique : des œuvres célèbres de l'avant-garde d'après-guerre (de M. Feldman, J. Cage, E. Brown, P. Boulez, K. Stockhausen et A. Boucourechliev pour n'en citer que quelques-unes) ont pris la forme de partitions graphiques dans lesquelles l'interprète pouvait librement naviguer. Boucourechliev a nommé nombre de ses compositions "Archipels" qui évoque la même métaphore. Ces partitions présentaient typiquement des fragments de notation répartis sur toute la page, afin d'émanciper l'œuvre de la linéarité imposée par la notation traditionnelle, une idée largement discutée dans *Opera Aperta* d'U. Eco [2]. Dans la deuxième sonate pour piano de Boulez, par exemple, cette coexistence de bribes de notation conventionnelle et d'une disposition en forme de carte sur la page illustre comment les cartes permettent des cas hybrides convaincants, dans lesquels tous les symboles sur la page ne fonctionnent pas comme une carte. De même, dans les partitions plus conventionnelles/linéaires, Miller souligne que seules certaines caractéristiques sont isomorphes (ou semblables à des cartes) : "*Les partitions sont des cartes qui sont isomorphes avec les structures spatiales et temporelles des œuvres musicales qu'elles représentent, tandis que d'autres caractéristiques graphiques peuvent être purement contingentes ou accidentelles. Cela met en évidence une propriété intéressante des cartes : il suffit qu'elles soient isomorphes par rapport à un sous-ensemble de propriétés de l'espace qu'elles représentent*". Les cartes, d'une manière qui rappelle les partitions, visent à guider un utilisateur ou à inciter un exécutant à agir, et cet objectif ne doit pas nécessairement obéir à des relations de correspondance strictes, systématiques et univoques. Pour Miller en effet, ce mélange d'isomorphisme et de contingence répond à la célèbre attaque de Goodman contre John Cage, formulée comme suit : "*Sans stipulation d'unités significatives minimales d'angle et de distance, la p. 53 du Concert pour piano et orchestre de John Cage de 1960 n'est pas syntaxiquement différenciée*". L'observation du philosophe l'a conduit à une critique virulente de l'approche de John Cage en matière de notation graphique : "*Sous le système proposé, il n'y a pas de caractères disjoints et différenciés ou de classes de conformité, pas de*

notation, pas de langage, pas de partition"[3], un point de vue qui, en retour, s'est avéré largement impopulaire. Les réflexions sur les cartes et les partitions peuvent sembler ici trop théoriques. L'ère numérique, cependant, pousse les compositeurs à penser les partitions d'une manière nouvelle, dans laquelle l'hybridation entre systèmes interactifs et notation laisse plus de place à notre notion de carte comme partition. Dans "The digital score, musicianship, creativity and innovation" [3], Craig Vear propose que : "*L'objectif principal de toute partition numérique est de communiquer des idées entre musiciens à l'aide de la technologie numérique*", plaçant ainsi la technologie au centre de la notation contemporaine, et élargissant par la même occasion l'éventail des systèmes jusqu'alors qualifiés de "partitions". Vear propose également que *"certaines partitions numériques puissent ressembler à des jeux informatiques où l'interprète prend des décisions sur ce qui se passe ensuite.*" L'image du jeu vidéo, tout comme celle d'une carte que l'interprète peut parcourir, évoque d'abord la liberté, mais aussi un important changement d'orientation lorsqu'on la considère du point de vue d'un compositeur traditionnel. Son métier se rapproche alors de celui d'un concepteur d'instruments, que Vear considère toujours comme appartenant au domaine des partitions numériques. Dans un chapitre intitulé *"La nature des partitions numériques : étendre les signatures de base*", il déclare : "*la partition peut être intégrée dans la conception d'un instrument, l'instrument peut être une partition-système d'électronique contrôlée par un logiciel génératif*". Dans de telles circonstances, la frontière entre la partition et la conception d'un instrument numérique s'estompe, et diverses formes de systèmes interactifs interrogeant une base de données seront considérées comme une partition. Dans certains des premiers comptes rendus des utilisations artistiques de CataRT [4], Diemo Schwartz décrit ses représentations d'un corpus sonore comme des "partitions navigables" ou des "instruments de partition" : le sujet de la pièce était les "partitions navigables" ou les "instruments de partition", dans lesquels différents types d'utilisateurs joueraient le son ou la musique, en naviguant à travers la partition" [5]. La découverte de Schwartz présente alors ce paradoxe qu'elle propose une partition qui représente le temps.

1.2. Les cartes ne représentent pas le temps

Les partitions musicales sont utilisées pour transmettre des informations sur les matériaux musicaux en fonction du temps. Les cartes, en revanche, sont des représentations statiques d'une zone. Elles ne représentent pas le temps ou les changements dans le temps, et sont typiquement utilisées pour transmettre des informations sur les caractéristiques géographiques et les relations spatiales, et sont également un outil puissant de visualisation en science des données, comme nous le développerons au chapitre 5. "*"De nombreux objets des nouveaux médias ne racontent pas d'histoires ; ils n'ont ni début ni fin"* [6]. Cette citation de Manovitch nous aide ici à introduire une idée stimulante pour un compositeur traditionnel : ignorer la dimension temporelle d'une partition peut conduire à des approches innovantes de la composition. Nous allons maintenant exposer comment de telles parcelles de données, ou plus globalement des bases de données, peuvent également être considérées comme une forme d'art. Enfin, l'absence de temps inhérente à ces concepts nous incite à utiliser dans la section 1.2.2 des arguments esthétiques pour comprendre ce qui pourrait rester artistiquement intéressant dans le cadre d'un contrôle relativement souple du temps dans la composition musicale.

1.2.1. La base de donnée comme matériau artistique

Les bases de données et l'exploration de données en musique peuvent être considérées comme une forme d'art dans le sens où elles impliquent l'utilisation de la créativité et des compétences analytiques pour extraire des idées et des connaissances de grands ensembles de données sonores. Cela nécessite la capacité d'intégrer la créativité musicale en identifiant des modèles, des tendances et des relations dans les données qui pourraient ne pas être apparents autrement. Les romans et les films sont, comme la musique, la danse ou le théâtre, des formes d'art temporelles. La perspicacité de Manovitch sur la forme d'art de la base de données suggère que les nouveaux médias ne sont pas subordonnés au temps, ou narratifs, de la même manière ; cette considération peut aider un compositeur à modifier certaines idées préconçues sur la base de données. Cette considération peut aider un compositeur à modifier certains à prioris sur son approche de la forme, de l'articulation ou de la narration : "Après que le roman, et par la suite le cinéma, aient privilégié la narration comme forme clé de l'expression culturelle de l'ère moderne, l'ère informatique introduit son corrélat - la base de données. De nombreux objets des nouveaux médias ne racontent pas d'histoires ; ils n'ont ni début ni fin ; en fait, ils n'ont aucun développement, thématique ou autre, qui organiserait leurs éléments en une séquence"[6]. Cette absence de narration dans l'art des nouveaux médias, que Manovitch rattache à la post-modernité, trouve un écho intéressant dans la pensée de Morton Feldman, qui voyait dans l'avant-garde européenne un désir excessif de contrôle de la temporalité.

1.2.2. Morton Feldman and the European clock makers

Dans ses écrits [7], Morton Feldman se souvient d'une discussion qu'il a eue avec Karlheinz Stockhausen : *"Il était convaincu qu'il me démontrait la réalité, que le rythme, et le placement possible des sons par rapport à celui-ci, était la seule chose à laquelle le compositeur pouvait s'accrocher de façon réaliste. [. . .] Franchement, cette approche du temps m'ennuie. Je ne suis pas un horloger. Ce qui m'intéresse, c'est de saisir le temps dans son existence non structurée. C'est-à-dire que je m'intéresse à la façon dont cette bête sauvage vit dans la jungle - et non dans un zoo*". Feldman, souvent avec humour, insiste sur cette idée : *"Laisse les sons tranquilles, Karlheinz, ne les pousse pas - même pas un peu ?"*. Cette approche

presque passive de la composition fait écho à la pensée d'inspiration zen de John Cage, et s'inspire aussi et surtout des peintres : *"un peintre sera peut-être d'accord avec le fait qu'une couleur insiste pour avoir une certaine taille, indépendamment de ses souhaits [...] Il peut simplement lui permettre d'"être"[...] Depuis quelques années, je me rends compte que le son a une prédilection pour suggérer ses propres proportions [...] Tout désir de différenciation doit être abandonné*". La musique de Feldman a exploré cette idée de surface de nombreuses façons, notamment par l'utilisation de lignes musicales longues, lentes et statiques, la répétition de motifs simples, l'absence de contraste (qu'il appelle différenciation) et l'utilisation de dimensions temporelles non conventionnelles (certaines œuvres, comme le deuxième quatuor à cordes, durent plus de quatre heures). Cette approche non conventionnelle de la forme se rapporte plus largement à une opposition entre les avant-gardes européenne et américaine dans les années 50-60. Les outils décrits au chapitre 5 ont incité l'auteur à penser la forme musicale comme de longues toiles temporelles statiques, comme les appellerait M. Feldman : "*Mon obsession de la surface est le sujet de ma musique. En ce sens, mes compositions ne sont pas du tout des "compositions". On pourrait les appeler des toiles temporelles dans lesquelles j'amorce plus ou moins une teinte globale de la musique*". En écoutant presque toutes ses œuvres, on se rend compte que l'ensemble du morceau partage le plus souvent la même atmosphère du début à la fin. Une pièce avec une combinaison instrumentale reconnaissable, comme *Why patterns* pour flûte, piano et célesta ou Clarinette et Percussion, ont une forte empreinte acoustique et illustrent comment l'écoute par apprentissage automatique pourrait être utilisée pour travailler sur de tels matériaux.

2. CORPUS-BASED CONCATENATIVE SOUND SYNTHESIS (CBCS), LA SYNTHÈSE CONCATÉNATIVE AUJOURD'HUI

La synthèse sonore concaténative basée sur un corpus (CBCS) est une technique utilisée en informatique musicale qui consiste à construire un son ou un morceau de musique en concaténant (en joignant ensemble) de plus petites unités sonores, comme des phonèmes dans la synthèse vocale ou des phrases musicales dans la synthèse musicale. Il peut être utilisé pour modéliser un musicien instrumental improvisateur en créant une base de données de phrases ou de segments musicaux enregistrés qui peuvent être combinés et réarrangés en temps réel pour créer une performance musicale qui semble être improvisée.

Aujourd'hui âgé de près de 20 ans si l'on se réfère aux premières publications de CataRT [4], CBCS jouit aujourd'hui d'une popularité croissante. Diverses applications sont aujourd'hui basées sur des principes similaires (AudioStellar, Audioguide, LjudMAP ou XO). La démocratisation des outils d'analyse audio et d'apprentissage automatique tels que le package FluCoMa (pour Max, SuperCollider et Pure Data) encourage les praticiens de l'informatique musicale à s'engager dans ce domaine à la croisée de la création musicale et de la science des données/apprentissage automatique.

2.1. Espace Timbre

Malgré des avancées prometteuses dans le domaine de l'apprentissage profond appliqué à la synthèse sonore [8] [9], les outils CBCS doivent leur popularité à une métaphore qui remonte aux débuts de l'informatique musicale : la notion d'espace timbrique, développée par Wessel [10] et Grey [11], selon laquelle les qualités multidimensionnelles du *timbre* peuvent être mieux comprises à l'aide de métaphores spatiales (par exemple, le timbre du cor anglais est plus proche du basson que celui de la trompette).

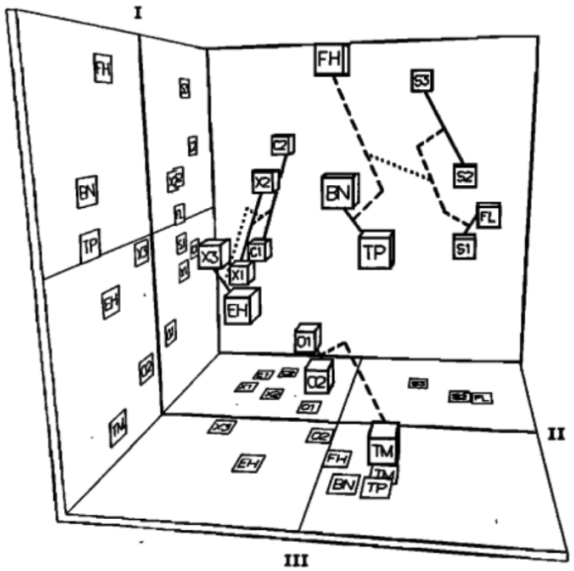

Figure 1.

L'héritage de la métaphore Timbre Space se retrouve également dans diverses itérations du projet Orchidea [12] et continue d'inspirer des générations de compositeurs et de réalisateurs en informatique musicale. [1]

Les pionniers des études sur la perception du timbre tels que Grey [11], J.C. Risset, D. Wessel, [15] ou Stephen McAdams [16] [17] définissent le plus souvent le timbre en soulignant ce qu'il n'est pas. Risset et Wessel, par exemple, le définissent comme suit : *Il s'agit de l'attribut perceptif qui nous permet de distinguer les instruments d'orchestre jouant la même hauteur et ayant le même volume sonore*.

La co-variance de ces paramètres (hauteur, intensité sonore et timbre) conduit cependant Schwarz à distinguer les

1 . Daniele Ghisi, co-auteur du package *bach* [13] pour Max, occupe un rôle ici puisqu'il a travaillé à la fois sur les projets *Orchidea* et *FluCoMa*. Certains objets de sa bibliothèque *dada* plus tardive [14] montrent également une influence de CataRT (l'objet dada.catart a été renommé plus tard dada.cartesian). La base dada.base, enfin, pourrait avoir été une source d'inspiration pour la manipulation de bases de données dans Flucoma/Max. Un extrait d'une de ses présentations est disponible ici : https://youtu.be/LD0ivjyuqMA?t=3032

notions d'espace timbrique et de CBCS : '*Notez que ce concept est similaire mais non équivalent à celui de l'espace timbrique mis en avant par Wessel et Grey [7, 24], puisque le timbre est défini comme les caractéristiques qui servent à distinguer un son d'un autre, qui restent après avoir retiré les différences de sonie et de hauteur. Notre espace sonore inclut explicitement ces différences qui sont très importantes pour l'expression musicale.* [18]

Le workflow décrit dans le chapitre 5 a donné dans la pratique une forte évidence de l'interdépendance entre le registre, le timbre et la dynamique, en particulier lorsque l'analyse est effectuée sur un seul fichier sonore d'instrument (par exemple 30 minutes de flûte solo), et découpé en échantillons courts. Le système sera alors précisément capable de trouver des similitudes entre des passages instrumentaux joués dans le même registre, la même dynamique et la même technique de jeu (par exemple, une flûte jouant des trilles rapides mezzo forte, dans un registre moyennement bas, avec de l'air).

2.2. Synthèse concaténative basée sur un corpus - État de l'art

Un large éventail de technologies peut aujourd'hui être qualifié de synthèse concaténative basée sur un corpus, dans le sens où elles permettent, par la segmentation et l'analyse, d'explorer de grandes quantités de sons. Certaines d'entre elles se présentent comme des solutions "prêtes à l'emploi", comme le récent *Audiostellar* [19], ou SCMIR [2] pour SuperColider. L'AudioGuide de Hackbarth [20] offre une approche légèrement différente puisqu'il utilise la morphologie/chronologie d'un fichier son pour produire une sortie concaténée. Dans le monde de Max enfin, deux environnements apparaissent comme hautement personnalisables : MuBu [21] de l'IRCAM et le projet plus récent FluCoMa [22] financé par l'UE. CataRT est maintenant entièrement intégré dans MuBu, dont l'objectif englobe l'analyse audio multimodale ainsi que la reconnaissance automatique des mouvements et des gestes [23]. Cela rend MuBu extrêmement polyvalent, mais aussi difficile à appréhender. Les outils de traitement de données de MuBu sont principalement exposés dans le cadre du plugin pipo [24], qui peut calculer par exemple l'analyse mfcc sur un tampon audio donné [3] en intégrant le plugin pipo.mfcc dans l'objet mubu.process.

FluComa se veut également polyvalent, mais semble particulièrement adapté à la réalisation de deux tâches spécifiques populaires. Avec une connaissance limitée du cadre de travail et de la théorie sous-jacente aux algorithmes qu'il utilise (comme ceux de la réduction de la dimentionnalité, de l'analyse mfcc ou de l'entraînement des réseaux neuronaux), le cadre de travail permet : 1/ de segmenter, analyser et représenter/restituer un corpus sonore 2/ d'entraîner un réseau de neurones pour contrôler un synthétiseur, d'une manière qui rappelle le Wekinator de Fiebrink [25].

Seuls les outils de segmentation, d'analyse, de représentation et de lecture (décrits en détail dans le chapitre 5) ont été utilisés ici, car ils répondent précisément aux besoins de la synthèse par corpus.

3. PRÉMICES

Certaines des premières versions du package FluCoMa fournissaient déjà des algorithmes efficaces de détection d'onset (onsetslice [4]), ce qui a encouragé l'auteur à creuser davantage dans leur environnement.

À l'époque, tout le matériel de composition généré avec le package utilisait l'objet fluid.bufstat [5] pour exécuter une analyse statique de la hauteur (et une analyse de confiance de la hauteur) sur chaque tranche d'un fichier sonore préexistant donné. Comme on peut l'entendre dans cet exemple d'accompagnement [6], les tranches ont été classées par hauteur (plus l'indice est bas, plus le registre est bas), bien qu'une certaine imprécision puisse se produire en raison de la simplicité de l'analyse (la section 5 décrira des stratégies plus élaborées).

La détection d'onset combinée à l'analyse statistique de la hauteur a été utilisée pour la première fois à grande échelle dans une pièce (Chef 2.0 [7]) dans lequel toutes les parties instrumentales avaient été générées avec des techniques rudimentaires comparables de recherche d'informations musicales.

Certaines expériences de diverses formes de contrôle de la RV sur de tels outils d'analyse semblaient déjà prometteuses à cette époque [8]. Le logiciel PatchXR sera abordé dans la section 5.9.

4. MOTIVATIONS

L'un des objectifs de la présente étude est de tirer parti des nombreuses capacités de visualisation, d'interaction et de données sur les gestes de mouvement disponibles dans un environnement VR. Le portage dans la VR d'une analyse réalisée dans Max/Flucoma présente des difficultés,

2 . Une démo est disponible sur : https://youtu.be/jxo4StjV0Cg

3 . MFCC signifie Mel-Frequency Cepstral Coefficients. Il s'agit d'une méthode d'extraction de caractéristiques couramment utilisée dans les systèmes de reconnaissance de la parole et du locuteur. Les CCFM sont utilisés pour représenter les caractéristiques spectrales d'un son sous une forme compacte, plus facile à analyser et à traiter que la forme d'onde brute. Elles sont calculées en appliquant une série de transformations au spectre de puissance d'un signal sonore, y compris une déformation à l'échelle de Mel de l'axe des fréquences, en prenant le logarithme du spectre de puissance et en appliquant une transformée en cosinus discrète (DCT) aux coefficients résultants. Les coefficients résultants, appelés MFCC, capturent les caractéristiques spectrales du son et sont généralement utilisés comme caractéristiques pour l'apprentissage de modèles d'apprentissage automatique pour des tâches telles que la reconnaissance de la parole et l'identification du locuteur.

4 . https://learn.flucoma.org/reference/onsetslice/

5 . https://learn.flucoma.org/reference/bufstats/

6 . https://youtu.be/UNj7_TI8SVs

7 . Simulation : https://www.youtube.com/watch?v=MkMKVm3G3W8 Résultat : https://youtu.be/Sc4Ye_rnSO8?t=9893. Bien que cela ne soit pas pertinent pour la discussion ici, le mouvement du bras du chef d'orchestre a été mis en correspondance avec la vitesse du curseur sur les écrans des interprètes à l'aide de INScore [26] et Gesture Follower [27] dans Mubu for Max.

8 . https://youtu.be/DC_BL_HGPLA

mais j'explore actuellement diverses manières possibles d'interagir avec une analyse basée sur un corpus [9] . Après quelques essais dans lesquels les coordonnées x y z d'un monde représentaient directement des descripteurs audio tels que la hauteur du son et la centroïde, j'ai utilisé plus systématiquement l'analyse mfcc et la réduction de dimensionnalité, comme décrit dans la section suivante.

L'utilisation de l'apprentissage automatique (réduction de la dimensionnalité) dans ce dernier cas rend un monde dans lequel les coordonnées absolues de chaque point n'ont plus de lien avec l'espace des descripteurs (les sons aigus ne peuvent pas être mis en correspondance avec l'axe des y par exemple), mais offre de manière frappante des informations de regroupement convaincantes, qui sont liées aux différents styles de jeu de l'instrument analysé : par exemple, dans cet extrait basé sur des sons de flûte, l'ouverture montre une opposition claire entre deux types de gestes : 1/ les sons staccato et 2/ les matériaux de type échelle legato. Ce contraste sonore est rendu explicite par un mouvement de l'avatar qui saute d'un groupe de boutons à un autre.

5. WORKFLOW

Après de nombreuses tentatives d'écoute et de jeu avec ce système [10] , j'étudie maintenant comment diversifier les métaphores VR pour exciter le moteur de synthèse basé sur le corpus, ainsi que les différentes manières dont la synthèse peut être rendue sur un orchestre de haut-parleurs équipés de RaspberryPi (voir chapitre 6).

5.1. Corpus Selection

Mes expériences se sont concentrées presque exclusivement sur les corpus d'instruments acoustiques. Les outils présentés ici peuvent générer efficacement de la musique instrumentale virtuose plausible (comme c'était également le cas dans le morceau Chef 2.0 présenté plus haut), mais des utilisations récentes ont permis d'obtenir des résultats plus satisfaisants dans des types de textures plus lentes, plus calmes, de type "Feldman". Diverses limitations du côté de la lecture (soit dans VR, soit sur un échantillonneur Pure Data pour RaspberryPi décrit dans le chapitre 6) ont imposé des restrictions dans les premières étapes sur la quantité de données qu'il pouvait gérer (moins de 5minutes en AIFF dans VR) ou le nombre de tranches dans lesquelles l'échantillon pouvait être découpé (256 à cause de la limitation des listes dans Max). Ces deux limitations ont été surmontées par la suite (utilisation du format ogg dans VR, augmentation de la taille du buffer interne dans fluid.buf2list), ce qui a permis d'obtenir des modèles bien plus convaincants.

5.2. Analyse dans FluCoMa

L'utilisation de la synthèse concaténative pour modéliser un musicien instrumental improvisateur implique généralement plusieurs étapes :

1. Segmentation d'un grand fichier sonore : Il s'agit de diviser un grand enregistrement audio de la performance du musicien en unités ou segments plus petits.
2. Analyse : Ces segments sont ensuite organisés dans une base de données en fonction de diverses données descriptives (mfcc dans notre cas).
3. Mise à l'échelle/pré-traitement : la mise à l'échelle est appliquée pour une meilleure visualisation.
4. Réduction de la dimension : Sur la base des descripteurs mfcc, la dimensionnalité des données est réduite afin de les rendre plus gérables et plus faciles à travailler. Cette réduction peut être réalisée à l'aide de techniques telles que l'analyse en composantes principales (ACP), la décomposition en valeurs singulières (SVD) ou l'approximation et la projection uniformes des formes (UMAP, méthode préférée dans notre cas). §équençage des proches voisins : Une fois les segments organisés et analysés, le logiciel les sélectionne et les combine en temps réel sur la base de certains paramètres ou règles d'entrée pour créer une performance musicale simulée qui sonne comme si elle était improvisée par le musicien. Nous utilisons ici un algorithme de voisinage proche, qui sélectionne les segments qui sont similaires d'une certaine manière (par exemple, en termes de hauteur, de volume ou de timbre - grâce aux similarités révélées par umap sur les mfccs dans notre cas) au segment en cours de lecture.

Nous allons maintenant décrire ces étapes plus en détail :

5.3. Slicing

Nous avons vu au chapitre 3 comment le découpage déclenche musicalement des possibilités. Dans MuBu, la détection de l'onset se fait avec pipo.onseg ou pipo.gate. FluCoMa expose cinq algorithmes de détection d'onset différents

1. fluid.ampslice : Trancheur de détritique basé sur l'amplitude
2. fluid.ampgate : Détection de porte sur un signal
3. fluid.onsetslice : Trancheur de buffer audio basé sur la différence spectrale
4. fluid.noveltyslice : Basé sur la matrice d'auto-similarité (SSM)
5. fluid.transcientslice : Implémente un algorithme de dé-clicage

Seul Onsetslice a été testé de manière approfondie. Les seuls paramètres modifiés étaient un "seuil" direct ainsi qu'un argument "longueur minimale de la tranche", déterminant la tranche la plus courte autorisée (ou la durée minimale d'une tranche) en *hopSize*. Ceci introduit une limitation commune aux outils de CBCS : le système incite

9 . https://youtu.be/cM-utIxv2Ww

10 . Extrait Cello; Extrait Flûte; Playlist pour divers instruments...

fortement l'utilisateur à choisir des échantillons courts pour de meilleurs résultats d'analyse, et plus d'interactivité, lorsqu'il contrôle la base de données avec un suiveur de geste. Aaron Einbond remarque dans l'utilisation de CataRT comment les échantillons courts convenaient le mieux à son intention : "Les échantillons courts contenant des attaques rapides et sèches, telles que des clics de touches enregistrés de près, étaient particulièrement adaptés à une impression convaincante de mouvement de la source WFS unique. L'effet obtenu est celui d'un instrument virtuel se déplaçant dans la salle de concert en même temps que les changements de son contenu timbral, réalisant ainsi la proposition initiale de Wessel."[28]

Une limitation connexe de la synthèse concaténative réside dans le fait que des échantillons courts démontreront l'efficacité de l'algorithme ; mais en même temps, ils s'éloignent de la "simulation plausible" recherchée dans la présente étude. Il faut donc trouver un équilibre entre la liberté imposée par de grands échantillons et le contrôle affiné que l'on peut obtenir avec des échantillons courts.

Une concaténation directe des tranches clique dans la plupart des cas sur le point d'édition, ce qui peut être évité par l'utilisation de rampes. Le deuxième défaut le plus notable de la concaténation concerne l'interruption des résonances du registre grave, que même une grande réverbération ne parvient pas à rendre plausible. Avoir un seuil bas et une grande "minslicelenght" résulte en des tranches équidistantes, toutes de durées identiques, comme le ferait l'objet pipo.onseg dans MuBu.

Comme nous écoutons le son dans le temps, ce paramètre responsable de la *durée des échantillons* est d'une importance primordiale.

5.4. MFCC sur chaque tranche - sur une tranche/un segment entier

Multidimensionnelle, l'analyse MFCC (Mel-Frequency Cepstral Coefficient) est une technique utilisée pour extraire des caractéristiques des signaux audio qui sont pertinentes pour la reconnaissance de la parole et de la musique. Elle consiste à calculer un ensemble de coefficients qui représentent l'enveloppe spectrale du signal audio, et peut être utilisée pour capturer les caractéristiques spectrales du style de jeu du musicien.

5.5. Analyse statistique sur chaque tranche

BufStats est utilisé pour calculer des mesures statistiques sur les données stockées dans un buffer. Un buffer est ici un type de structure de données qui contient des informations de séries temporelles, des données de descripteurs audio dans ce cas. BufStats calcule sept statistiques sur les données du buffer : moyenne, écart type, asymétrie, aplatissement, valeurs basses, moyennes et hautes. Ces statistiques fournissent des informations sur la tendance centrale des données et la façon dont elles sont distribuées autour de cette tendance. En plus de calculer des statistiques sur le buffer d'origine, BufStats peut également calculer des statistiques sur jusqu'à deux dérivés des données d'origine, appliquer des pondérations aux données à l'aide d'un buffer de pondération, et identifier et supprimer les trames aberrantes. Ces mesures statistiques peuvent être utiles pour comparer différentes données de séries temporelles, même si les données d'origine sont de longueurs différentes, et peuvent fournir une meilleure distinction entre les points de données lorsqu'elles sont utilisées dans la formation ou l'analyse. La sortie de BufStats est un buffer comportant le même nombre de canaux que les données d'origine, chaque canal contenant les statistiques des données correspondantes dans le buffer d'origine.

5.6. Normalisation

Le package FluCoMa propose plusieurs outils de mise à l'échelle/pré-traitement, parmi lesquels la normalisation et la standardisation ont été utilisées. La normalisation et la standardisation sont des techniques utilisées pour transformer les variables afin qu'elles puissent être comparées ou combinées dans des analyses statistiques. Les deux techniques sont utilisées pour rendre les données plus comparables, mais elles fonctionnent de manière légèrement différente.

La normalisation consiste à mettre à l'échelle une variable pour qu'elle ait une moyenne de 0 et un écart-type de 1. Pour ce faire, on soustrait la moyenne de la variable de chaque point de données, puis on divise par l'écart-type. La normalisation est souvent utilisée lorsque les variables à comparer se situent sur des échelles différentes ou ont des unités de mesure différentes. Elle permet de comparer des variables qu'il serait autrement difficile de comparer directement.

La normalisation consiste à mettre à l'échelle une variable afin qu'elle ait une valeur minimale de 0 et une valeur maximale de 1. Pour ce faire, on soustrait la valeur minimale de la variable de chaque point de données, puis on divise par l'étendue (c'est-à-dire la différence entre les valeurs maximale et minimale). La normalisation est souvent utilisée lorsque les variables comparées ont une distribution asymétrique, ou lorsque les variables ne sont pas normalement distribuées. Elle permet de comparer des variables qui seraient autrement difficiles à comparer directement en raison de l'asymétrie de leur distribution.

La normalisation met à l'échelle une variable pour qu'elle ait une moyenne de 0 et un écart-type de 1, tandis que la normalisation met à l'échelle une variable pour qu'elle ait une valeur minimale de 0 et une valeur maximale de 1. La normalisation ccaling a été jugée plus facile à utiliser à la fois en 2-D (dans FluCoMa, l'objet fluid.plotter), ainsi que dans le monde VR 3D dans lequel l'origine correspond à un coin du monde. L'objet fluid.nomalize possède un attribut "@max" (1 par défaut), qui correspond alors directement aux dimensions du monde VR.

5.7. Réduction de la dimensionnalité

La réduction de la dimensionnalité est une technique utilisée en apprentissage automatique pour réduire le nombre

de caractéristiques (dimensions) dans un ensemble de données. L'objectif de la réduction de la dimensionnalité est de simplifier les données sans perdre trop d'informations. Divers algorithmes de réduction de la dimensionnalité sont présentés dans une étude FluCoMa précoce[29], sans mention, curieusement, de l'UMAP, qui a ensuite été privilégié.

SOM est l'un des algorithmes de réduction de la dimensionnalité les plus populaires. Il est mis en œuvre dans la bibliothèque ml.starmlstar pour Max, une bibliothèque simple et pratique pour l'apprentissage automatique, l'une des nombreuses structures et algorithmes d'apprentissage automatique célèbres dans la communauté NIME. [30] [31] [32] [33].

SOM (Self-Organizing Map) et UMAP (Uniform Manifold Approximation and Projection) sont deux techniques de réduction de la dimensionnalité. SOM est un type de réseau neuronal qui est formé par apprentissage non supervisé. Il se compose d'une grille de neurones, dont chacun est associé à un ensemble de poids. Le SOM est formé en lui présentant des données d'entrée et en ajustant les poids des neurones de manière à ce que les modèles d'entrée similaires soient mis en correspondance avec les neurones voisins sur la grille. La carte résultante est une représentation à faible dimension des données d'entrée qui préserve la structure topologique des données d'origine. L'UMAP, quant à elle, est une technique non linéaire de réduction de la dimensionnalité qui repose sur les principes de l'analyse topologique des données. Elle utilise une combinaison de techniques telles que les k-voisins les plus proches, la construction de graphes pondérés et l'intégration à faible dimension pour produire une représentation à faible dimension des données d'entrée. Contrairement au SOM, qui est limité à une structure de grille fixe, l'UMAP peut produire une représentation continue et flexible des données. SOM et UMAP peuvent tous deux être utiles pour visualiser des données hautement dimensionnelles et pour découvrir des modèles et des relations dans les données. Toutefois, l'UMAP présente certains avantages par rapport au SOM, notamment la possibilité de traiter plus efficacement de grands ensembles de données et de produire des résultats plus interprétables.

L'UMAP (Uniform Manifold Approximation and Projection) peut être utilisé pour visualiser des données à haute dimension dans un espace à plus faible dimension. Appliqué à des données sonores analysées à l'aide de MFCC (Mel-Frequency Cepstral Coefficients), l'UMAP réduit la dimensionnalité des données et crée une représentation visuelle du son dans un espace à deux ou trois dimensions. Les CCFM, là encore, sont une technique d'extraction de caractéristiques couramment utilisée dans le traitement de la parole et de l'audio. Elles consistent à décomposer un signal sonore en un ensemble de bandes de fréquences et à représenter le spectre de puissance de chaque bande par un ensemble de coefficients. Les coefficients MFCC résultants capturent des caractéristiques spectrales importantes du signal sonore (bien que difficilement interprétables par l'utilisateur novice), telles que la fréquence et l'amplitude des pics spectraux. En appliquant l'UMAP aux coefficients MFCC d'un signal sonore, il est possible de créer une représentation visuelle du son qui préserve les relations entre les différents coefficients MFCC (voir Fig. 2). Cela peut être utile pour des tâches telles que l'exploration de la structure d'un ensemble de données sonores, l'identification de modèles ou de tendances dans les données et la comparaison de différents sons.

Figure 2. La réduction de la dimensionnalité des MFCC permet de révéler les similitudes spectrales. UMAP produit des coordonnées en 2d ou 3d.

L'UMAP est donc utilisé en premier lieu pour ses capacités de regroupement, ce qui facilite la classification. Elle permet d'identifier des modèles ou des tendances qui ne sont pas forcément évidents à partir des données brutes. Plus important encore, les dimensions non linéaires proposées par UMAP (que ce soit en 2d dans Max ou en 3 dimensions dans PatchXR, et lorsqu'elles sont comparées à des analyses linéaires dans lesquelles, par exemple, x, y et z correspondent à la hauteur, à l'intensité sonore et à la centroïde) ont donné lieu à des regroupements bien plus "intelligents" que les types de représentations plus conventionnels et conformes aux paramètres.

5.8. Requêtes de voisinage

La fonction de recherche de voisinage est légèrement différente à chaque fois, mais elle est basée dans FluCoMa sur les arbres K-d et l'algorythme knn. Dans MuBu , l'objet mubu.knn, ainsi que l'objet ml.kdtree ml.star, donnent des résultats très comparables à ceux obtenus avec fluid.kdtree.

Les arbres K-d (abréviation de "k-dimensional trees") et les k-voisins les plus proches (k-NN) sont deux algorithmes apparentés, mais dont les objectifs sont différents.

Un arbre k-d est une structure de données utilisée pour stocker et interroger efficacement un ensemble de points

dans un espace à k dimensions. Il fonctionne en partitionnant les points dans un arbre binaire, chaque nœud de l'arbre représentant un hyperplan qui divise l'espace en deux moitiés. Les points sont partitionnés de manière récursive dans les sous-arbres gauche et droit en fonction du côté de l'hyperplan sur lequel ils se trouvent. En organisant les points de cette manière, il est possible de trouver rapidement les plus proches voisins d'un point donné en ne recherchant qu'un sous-ensemble de l'arbre plutôt que l'ensemble des points.

D'autre part, l'algorithme k-NN est un algorithme d'apprentissage automatique utilisé pour la classification ou la régression. Étant donné un ensemble de points étiquetés et un nouveau point non étiqueté, l'algorithme k-NN détermine les k points de l'ensemble qui sont les plus proches du nouveau point, puis utilise les étiquettes de ces points pour prédire l'étiquette du nouveau point. La valeur de k est un hyperparamètre choisi par l'utilisateur, qui détermine le nombre de voisins pris en compte lors de la prédiction.

En résumé, un arbre k-d est une structure de données utilisée pour stocker et interroger efficacement un ensemble de points dans un espace à k dimensions, tandis que l'algorithme k-NN est un algorithme d'apprentissage automatique utilisé pour la classification ou la régression. Ces deux algorithmes sont souvent utilisés dans des applications telles que la reconnaissance des formes, la classification des images et l'exploration des données.

Alors que CataRT ou Audiostellar sont typiquement utilisés pour la génération de textures électroniques/design sonore, j'ai le plus souvent utilisé FluCoMa pour générer des instruments monophoniques (un interprète joue un instrument à la fois), dans lesquels l'avatar reproduit ce que knn ferait avec un instrument automatique : il privilégiera dans son choix l'échantillon qu'il peut atteindre à portée de main, plutôt que de sauter une grande distance entre 2 éléments (voir Fig. 3) .

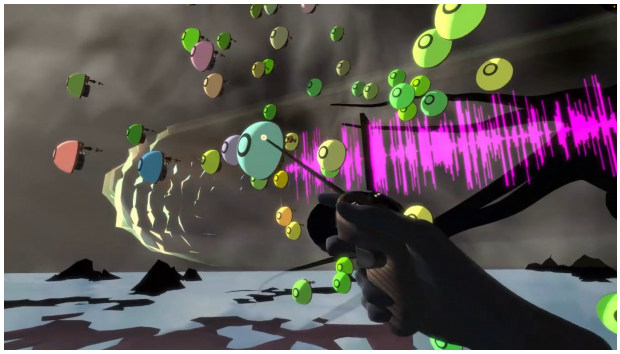

Figure 3. Une interface VR dans laquelle chaque bouton du monde correspond à une tranche du fichier sonore. L'apprentissage automatique permet de rapprocher les sons qui partagent des caractéristiques spectrales communes.

5.9. PatchXR

PatchXR [34] est une station de travail audio numérique ludique permettant de faire de la musique en VR. Sa métaphore de base correspond à ce que l'équipe FluCoMa appelle CCE (environnements de codage créatifs) dans la mesure où il fonctionne à bien des égards comme Max ou Pure Data.

L'une des raisons d'utiliser la VR pour explorer un ensemble de données 3D est qu'elle permet aux utilisateurs d'interagir avec les données d'une manière plus naturelle et immersive, en l'utilisant comme un outil de visualisation et d'analyse des données. Les utilisateurs peuvent se déplacer et explorer les données sous différents angles, ce qui peut les aider à mieux comprendre les relations entre les différents points de données et à identifier des modèles. Les utilisateurs obtiennent un sens plus intuitif des données et comprennent mieux comment elles sont structurées et comment les différents points de données sont liés les uns aux autres.

La structure d'un fichier .patch (un monde patchXR) suit la syntaxe d'un fichier .maxpat (pour Max) ou .pd (pour pure data) dans le sens où il déclare d'abord les objets utilisés, puis les connexions entre eux. Cette structure a rendu relativement trivial le fait de générer une routine javascript prenant en entrée un dictionnaire (fichier json) avec les coordonnées 3D de chaque segment, et en sortie un nouveau fichier .patch (un monde accessible en VR, voir le workflow général sur la Fig. 4). Une étude plus poussée du patching dans PatchXR a permis d'implémenter cette acrtographie de manière plus subtile (sans représenter chaque point par un bouton, comme dans d'autres exemples plus récents. [11]

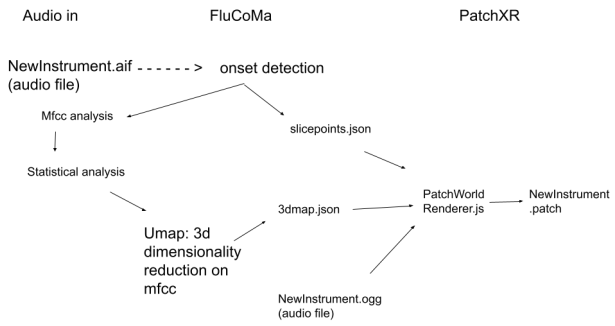

Figure 4. Flux de travail général : d'un fichier audio d'entrée à sa représentation .patch 3d dans PatchXR.

5.9.1. Interaction et communication OSC

PatchXR expose un large éventail de blocs (un bloc correspond à un objet dans Max ou Pure Data), ce qui permet d'accéder facilement aux données gestuelles telles que :

— La position/distance entre les mains/contrôleurs et une référence.

11 . https://youtu.be/jQsZT_Tlecs?t=8

— Les angles de rotation (x y z) des contrôleurs des deux mains
— Contrôleurs 2-d de type écran tactile, où l'utilisateur déplace la position xy d'un sélecteur sur un plan en le saisissant manuellement.
— 2-d lazer-like controllers, où l'utilisateur déplace la position xy d'un sélecteur à distance, comme s'il utilisait un pointeur lazer vers un écran distant.
— 2-d pads, qui permettent d'accéder à la vélocité à laquelle le pad est frappé
— contrôleurs de type theremine 3-d, où l'utilisateur déplace la position xyz d'un sélecteur sur un plan en le saisissant manuellement.
— 1-d sliders, knobs, boutons...

L'un des défis actuels consiste à diversifier les modes d'interrogation du corpus. Les mappings un à un des résultats UMAP tels que ceux décrits dans le chapitre 5.9 privilégient les simulations pour solistes, ou duo en mode multijoueur, dans lesquelles les boutons de l'interface se font face, afin d'inciter les joueurs à se faire face (voir https://youtu.be/LP1g79BdIpY).

Une simulation pour un plus grand nombre d'instruments, en particulier lorsqu'on joue seul, incite à utiliser des types de contrôle de plus haut niveau sur les automates, le plus important étant la simple capacité de concaténation automatique : jouer l'échantillon suivant dès que le précédent s'est arrêté.

6. FUTURS TRAVAUX : L'ORCHESTRE RASPBERRY PI VS NMP EN MULTIJOUEUR

Dans le cadre d'une résidence d'artiste à l'UCA (Université côte d'azur), un des questionnements majeurs interroge comment les outils présentés ci-dessus (ceux qui concernent le domaine de la recherche d'information musicale - MIR) peuvent servir à contrôler une plateforme immersive composée d'un orchestre de 64 modules *Pré*. [35]

Au moment de la rédaction de ce rapport, les résultats les plus satisfaisants ont été obtenus en envoyant des messages à chaque RaspberriPi indépendamment, selon son adresse IP (statique) spécifique, avec une syntaxe simple d'une liste de 2 nombres entiers correspondant à : 1/ quel buffer consulter 2/ quelle tranche de ce buffer jouer. Poursuivant l'élaboration d'illustrations de l'espace timbrique, les modules *Pré*, avec les différentes acoustiques que sa mobilité permet, favoriseront une densité contrastée d'événements en fonction de l'acoustique de l'espace dans lequel se déroule l'écoute.

L'utilisation de Raspberry Pi dans notre cas impose des contraintes réseau (local) qui peuvent potentiellement entrer en conflit avec les possibilité récentes de multi-joueur dans PatchXR qui se déroulent en distanciel. [12]

12 . https://youtu.be/LP1g79BdIpY

7. CONCLUSIONS

Après avoir examiné si les partitions sont des cartes ou non, nous avons proposé un workflow pour la synthèse concaténative basée sur un corpus (CBCS), en faisant valoir que les outils d'apprentissage automatique pour la visualisation des données offrent des informations révélatrices et exploitables sur la qualité timbrale du matériel qui est analysé. Du point de vue d'un compositeur, la disparition de l'axe temporel x incite à envisager la composition non pas dans un sens narratif (compris selon les réflexions sur les nouveaux médias et la notion d'art basé sur les données développée par [6]), mais plutôt, comme des "toiles temporelles", suivant l'esthétique de Morton Feldman.

Les outils de "machine listening" présentés ici (FluCoMa, MuBu) aident à construire des instruments intelligents avec des quantités de données relativement petites ; nous avons insisté sur le contrôle de la durée des échantillons qui semble crucial en CBCS. Un équilibre doit être trouvé entre 1/ l'analyse d'échantillons de courte durée qui sont plus faciles à traiter et à catégoriser et 2/ les échantillons longs qui sonnent plus naturels dans le contexte des simulations instrumentales.

8. REMERCIEMENTS

Je suis reconnaissant pour le soutien de l'UCA/CTEL, dont le programme de recherche de résidence d'artiste a permis de réaliser ces expériences, et pour le soutien de PRISM-CNRS.

9. REFERENCES

[1] D. Miller, "Are scores maps ? a cartographic response to goodman," in *Proceedings of the International Conference on Technologies for Music Notation and Representation – TENOR'17*, H. L. Palma, M. Solomon, E. Tucci, and C. Lage, Eds. A Coru na, Spain : Universidade da Coru na, 2017, pp. 57–67.

[2] U. Eco, P. Eco, A. Cancogni, and D. Robey, *The Open Work*. Harvard University Press, 1989. [Online]. Available : https://books.google.fr/books?id=7jroM0M8TuwC

[3] C. Vear, *The Digital Score : Musicianship, Creativity and Innovation*. Routledge, 2019. [Online]. Available : https://books.google.fr/books?id=oSblwQEACAAJ

[4] D. Schwarz, G. Beller, B. Verbrugghe, and S. Britton, "Real-Time Corpus-Based Concatenative Synthesis with CataRT," in *9th International Conference on Digital Audio Effects (DAFx)*, Montreal, Canada, Sep. 2006, pp. 279–282, cote interne IRCAM : Schwarz06c. [Online]. Available : https://hal.archives-ouvertes.fr/hal-01161358

[5] D. Schwarz, R. Cahen, and S. Britton, "Principles and Applications of Interactive Corpus-Based

Concatenative Synthesis," in *Journées d'Informatique Musicale (JIM)*, Albi, France, Mar. 2008, pp. 1–1, cote interne IRCAM : Schwarz08a. [Online]. Available : https://hal.archives-ouvertes.fr/hal-01161401

[6] L. Manovich, "Database as symbolic form," *Convergence : The International Journal of Research into New Media Technologies*, vol. 5, pp. 80 – 99, 1999.

[7] M. Feldman, B. Friedman, and F. O'Hara, *Give My Regards to Eighth Street : Collected Writings of Morton Feldman*, ser. Exact Change. Exact Change, 2000. [Online]. Available : https://books.google.fr/books?id=hfgHAQAAMAAJ

[8] J.-P. Briot, G. Hadjeres, and F.-D. Pachet, *Deep Learning Techniques for Music Generation – A Survey*, Aug. 2019. [Online]. Available : https://hal.sorbonne-universite.fr/hal-01660772

[9] P. Esling, A. Chemla-Romeu-Santos, and A. Bitton, "Generative timbre spaces with variational audio synthesis," *CoRR*, vol. abs/1805.08501, 2018. [Online]. Available : http://arxiv.org/abs/1805.08501

[10] D. L. Wessel, "Timbre space as a musical control structure," *Computer Music Journal*, vol. 3, no. 2, pp. 45–52, 1979. [Online]. Available : http://www.jstor.org/stable/3680283

[11] K. Fitz, M. Burk, and M. McKinney, "Multidimensional perceptual scaling of musical timbre by hearing-impaired listeners," *The Journal of the Acoustical Society of America*, vol. 125, p. 2633, 05 2009.

[12] C.-E. Cella, "Orchidea : a comprehensive framework for target-based computer-assisted dynamic orchestration," *Journal of New Music Research*, vol. 0, no. 0, pp. 1–29, 2022. [Online]. Available : https://doi.org/10.1080/09298215.2022.2150650

[13] A. Agostini and D. Ghisi, "A Max Library for Musical Notation and Computer-Aided Composition," *Computer Music Journal*, vol. 39, no. 2, pp. 11–27, 06 2015. [Online]. Available : https://doi.org/10.1162/COMJ_a_00296

[14] D. Ghisi and C. Agon, "dada : Non-standard user interfaces for computer-aided composition in max," in *Proceedings of the International Conference on Technologies for Music Notation and Representation – TENOR'18*, S. Bhagwati and J. Bresson, Eds. Montreal, Canada : Concordia University, 2018, pp. 147–156.

[15] J.-C. Risset and D. Wessel, "Exploration of timbre by analysis and synthesis," *Psychology of Music*, pp. 113–169, 1999.

[16] S. Mcadams, S. Winsberg, S. Donnadieu, G. De Soete, and J. Krimphoff, "Perceptual scaling of synthesized musical timbres : Common dimensions, specificities, and latent subject classes," *Psychological research*, vol. 58, pp. 177–92, 02 1995.

[17] A. Caclin, S. Mcadams, B. Smith, and S. Winsberg, "Acoustic correlates of timbre space dimensions : A confirmatory study using synthetic tones," *The Journal of the Acoustical Society of America*, vol. 118, pp. 471–82, 08 2005.

[18] D. Schwarz, "The Sound Space as Musical Instrument : Playing Corpus-Based Concatenative Synthesis," in *New Interfaces for Musical Expression (NIME)*, Ann Arbour, United States, May 2012, pp. 250–253, cote interne IRCAM : Schwarz12a. [Online]. Available : https://hal.archives-ouvertes.fr/hal-01161442

[19] L. Garber, T. Ciccola, and J. C. Amusategui, "Audiostellar, an open source corpus-based musical instrument for latent sound structure discovery and sonic experimentation," 12 2020.

[20] B. Hackbarth, N. Schnell, P. Esling, and D. Schwarz, "Composing Morphology : Concatenative Synthesis as an Intuitive Medium for Prescribing Sound in Time," *Contemporary Music Review*, vol. 32, no. 1, pp. 49–59, 2013. [Online]. Available : https://hal.archives-ouvertes.fr/hal-01577895

[21] N. Schnell, A. Roebel, D. Schwarz, G. Peeters, and R. Borghesi, "Mubu and friends -assembling tools for content based real-time interactive audio processing in max/msp," *Proceedings of the International Computer Music Conference (ICMC 2009)*, 01 2009.

[22] P. A. Tremblay, G. Roma, and O. Green, "Enabling Programmatic Data Mining as Musicking : The Fluid Corpus Manipulation Toolkit," *Computer Music Journal*, vol. 45, no. 2, pp. 9–23, 06 2021. [Online]. Available : https://doi.org/10.1162/comj_a_00600

[23] F. Bevilacqua and R. Müller, "A gesture follower for performing arts," 05 2005.

[24] N. Schnell, D. Schwarz, J. Larralde, and R. Borghesi, "Pipo, a plugin interface for afferent data stream processing operators," in *International Society for Music Information Retrieval Conference*, 2017.

[25] R. Fiebrink and P. Cook, "The wekinator : A system for real-time, interactive machine learning in music," *Proceedings of The Eleventh International Society for Music Information Retrieval Conference (ISMIR 2010)*, 01 2010.

[26] D. Fober, Y. Orlarey, and S. Letz, "INScore - An Environment for the Design of Live Music Scores," in *Linux Audio Conference*, Stanford, United States, 2012, pp. 47–54. [Online]. Available : https://hal.archives-ouvertes.fr/hal-02158817

[27] F. Bevilacqua, B. Zamborlin, A. Sypniewski, N. Schnell, F. Guédy, and N. Rasamimanana, "Continuous realtime gesture following and recognition," vol. 5394, 02 2009, pp. 73–84.

[28] A. Einbond and D. Schwarz, "Spatializing timbre with corpus-based concatenative synthesis," 06 2010.

[29] G. Roma, O. Green, and P. A. Tremblay, "Adaptive mapping of sound collections for data-driven musical interfaces," in *New Interfaces for Musical Expression*, 2019.

[30] M. M. Wanderley and M. Battier, "Trends in gestural control of music," 2000.

[31] F. Bevilacqua, R. Müller, and N. Schnell, "MnM : a Max/MSP mapping toolbox," in *New Interfaces for Musical Expression*, Vancouver, France, May 2005, pp. 85–88, cote interne IRCAM : Bevilacqua05a. [Online]. Available : https://hal.archives-ouvertes.fr/hal-01161330

[32] R. Fiebrink and B. Caramiaux, "The machine learning algorithm as creative musical tool," *ArXiv*, vol. abs/1611.00379, 2016.

[33] B. Caramiaux and A. Tanaka, "Machine learning of musical gestures," in *New Interfaces for Musical Expression*, 2013.

[34] V. Bauer and T. Bouchara, "First steps towards augmented reality interactive electronic music production," in *2021 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*, 2021, pp. 90–93.

[35] *« PrÉ » : connected polyphonic immersion*. Zenodo, Jul. 2022. [Online]. Available : https://doi.org/10.5281/zenodo.6806324