MAPS AS SCORES: "TIMBRE SPACE" REPRESENTATIONS IN CORPUS-BASED CONCATENATIVE SYNTHESIS

Jonathan Bell
Aix-Marseille Université, PRISM-CNRS, XR2C2-UCA
Marseille, France
belljonathan50@gmail.com

ABSTRACT

The present study investigates ways in which the "timbre space" metaphor may be used in creative ways for instrumental composition. Numerous tools for concatenative sound synthesis share today the ability to represent in an n-dimensional space large quantities of sound, thus displaying on a map data which originally unfolded in time. If the potential of such systems for creating interactive instruments is an evidence, their affordance as musical scores needs further assessment, for control over time becomes unknown territory. When porting to VR such representation of sonic data, the *score* becomes a 3-dimensional map (or world) in which the user typically navigates freely. Experimentation through composition, instrument design and improvisation have shown a potential of simulation of plausibly automatised acoustical instrument, using machine learning techniques to model virtual instruments out of relatively small quantities of data (e.g. 20 minutes of audio to model a clarinet). The method offers promising avenues for the exploration of instrumental fragments clustered by timbre, register, dynamic, instrumental techniques. Whether or not such maps identify as musical scores, they contribute to addressing a problem formulated by Lev Manovitch: "how to merge database and narrative into a new form".

1. INTRODUCTION

1.1 Are scores maps?

The question raised in this article is in no small part inspired by a study by Daniel Miller, *Are scores maps? A cartographic response to Goodman*[1], in which a dialectical tension between two concepts (score and map) leads to interesting questions on the role and function of musical notation. Miller proposes that, in spite of surface-level conventions, the underlying structure of scores closely relates to maps: "*notational components of scores are better understood as contingent surface-level features leveraged by an underlying map-like representational structure. On this account, scores are seen to be highly conventionalized maps, and the notational symbols of scores constitute just one of multiple modes of representation and depiction harnessed by this framework*". Scores may therefore be conceived as a mere subset of maps. The rapprochement between both notions is best illustrated with a historical example: famous works of the post-war avant-garde (by M. Feldman, J. Cage, E. Brown, P. Boulez, K. Stockhausen and A. Boucourechliev to cite a few) took the form of graphical scores which the performer could freely navigate. Boucourechliev named many of his compositions "Archipels" (archipelago) which evokes the same metaphor. Those scores typically presented common notation fragments spread out all over the page, so as to emancipate the work from the linearity imposed by traditional notation, an idea thoroughly discussed in U. Eco's *Open Work* [2]. In Boulez's *second sonata* for piano for instance, this co-existence of snippets of conventional notation with a map-like layout on the page illustrates how maps allow for compelling hybrid cases, in which not every symbol on the page functions as a map. Similarly, in more conventional/linear scores, Miller underlines that only some features are isomorphic (or maps-like): "*Scores are maps that are isomorphic with the spatial and temporal structures of the musical works they represent, while other graphical features may be purely contingent or incidental. This highlights an interesting property of maps: they need only be isomorphic with regard to a subset of the properties of the space they represent*". Maps, in a manner reminiscent of scores, aim to guide a user or prompt a performer for action, and this goal needs not obey strict, systematic, one to one mapping relationships. For Miller indeed, this mixture of isomorphism and contingency responds to Goodman's famous attack against John Cage, formulated as follows: "*Without stipulation of minimal significant units of angle and distance, p. 53 from John Cage's Concert for Piano and Orchestra from 1960 is not syntactically differentiated*". The philosopher's observation led him to vitriolic criticism of John Cage's approach to graphic notation: "*Under the proposed system there are no disjoint and differentiated characters or compliance classes, no notation, no language, no score*"[3], a point of view which, in return, has proven widely unpopular.

Reflexions on maps and paper scores may seem here overly theoretical. The digital age, however, urges composers to think about scores in new ways, in which hybridization between interactive systems and notation leaves more room to our notion of map as score. In "The digital score, musicianship, creativity and innovation"[4], Craig Vear pro-

poses that: "*The core purpose of any digital score is to communicate ideas between musicians using digital technology*.", thus placing technology at the center contemporary notation, and expanding at the same time the range of systems hitherto coined as "scores". Vear also proposes that "*Some digital scores might feel like computer games where the performer makes decisions about what happens next.*" The image of the computer game, just as the one of a map which the performer can browse, evokes freedom in the first place, but also an important shift of focus when considered through the perspective of a traditional composer. His craft then becomes closer to the one of an instrument designer, which Vear still considers belonging to the realm of digital scores. In a chapter entitled "The nature of digital scores: expanding the core signatures", he states: "*the score might be embedded within the design of an instrument, instrument might be a system-score of electronics controlled by generative software*". Under such circumstances, the frontier between score and digital instrument design blurs, and diverse forms of interactive system querying a database will be considered a score.

In some early accounts of artistic uses of CataRT [5], Diemo Schwartz described his 2d representations of a sound corpus as "navigable score" or "score instruments": "*The [piece's] subject was "navigable scores" or score-instruments, in which different kinds of users would play the sound or music, cruising through the score.*"[6] Schwartz's discovery then presents this paradox that it proposes a score that does not represent time.

1.2 Maps do not represent time

Musical scores are used to convey information about musical material function of time.

Maps, on the other hand, are static representations of an area. They do not represent time or changes over time, and are typically used to convey information about geographical features and spatial relationships, and are also a powerful tool for visualisation in data science, as will be developed in Chapter 5.

"*Many new media objects do not tell stories; they don't have a beginning or end*" [7]. This quote by Manovitch helps us here introducing a challenging idea for a traditional composer: ignoring the temporal dimension of a score may lead to innovative approaches to composition. We will now expose how such data plot, or more globaly databases can also be considered as an art form. Finally, the absence of time inherent to these concept encourages to use in Section 1.2.2 aesthetic arguments to understand what might stay creatively interesting within a loose control over time in musical composition.

1.2.1 Databases as an art form

Databases and data mining in music can be thought of as an art form in the sense that they involve using creativity and analytical skills to extract insights and knowledge from large sound datasets. This requires the ability to inform musical creativity by identifying patterns, trends, and relationships in data that may not be otherwise apparent.

Novels and film, are, as music, dance, or theatre, temporal art forms. Manovitch's insight on database art form suggests that new media isn't subordinate to time, or narrative, in the same manner; this consideration may help a composer alter some preconception on his approach to form, articulation, or narrative: "*After the novel, and subsequently cinema, privileged narrative as the key form of cultural expression of the modern age, the computer age introduces its correlate - database. Many new media objects do not tell stories; they don't have a beginning or end; in fact, they don't have any development, thematically, formally or otherwise which would organise their elements into a sequence.*"[7] This absence of narrative in new media art, which Manovitch relates to post-modernity, finds an interesting echo in the thought of Morton Feldman, who saw in the European Avant-Garde an excessive desire for control over time.

1.2.2 Morton Feldman and the European clock makers

In his collected writings [8], Morton Feldman recalls a discussion he once had with Karlheinz Stockhausen: "*He was convinced that he was demonstrating reality to me. That the beat, and the possible placement of sounds in relation to it, was the only thing the composer could realistically hold on to. [. . .] Frankly, this approach to time bores me. I am not a clockmaker. I am interested in getting Time in its unstructured existence. That is, I am interested in how this wild beast lives in the jungle - not in the zoo*". Feldman, often with humour, insisted on that idea : "*Let the sounds alone, Karlheinz, don't push them — not even a little bit?*". This almost passive approach to composition echoes the zen-inspireed thought of John Cage, and also importantly took his inspiration from painters: "*A painter will perhaps agree that a color insists on bieng a certain size, regardless of his wishes [...] He can simply allow it to 'be'. In recent years we realize that sound has a predilection for suggesting its own proportions [...] Any desire for differentiation must be abandoned*"

Feldman's music explored this idea of surface in a number of ways, such as through the use of long, slow and static musical lines, the repetition of simple motifs, absence of contrast (which he called differenciation), and the use of unconventional temporal dimensions (some works such as the second string quartet last over four hours). This unconventional approach to form relates more broadly to an opposition between European and American avant-garde in the 50-60s.

The tools described in Chapter 5 inclined the author to think of musical form as long static *time canvases*, as would M. Feldman call them: " "*My obsession with surface is the subject of my music. In that sense, my compositions are really not "compositions" at all. One might call them time canvases in which I more or less prime with an overall hue of the music*". Listening to almost any of his works, one realises that the entire piece most often shares the same atmosphere from its beginning until its end. A piece with a recognisable instrumental combination such as *why patterns* for flute, piano and celesta or *Clarinet and Percussion* have a strong acoustical footprint and illustrate how

machine machine listening could be used on such materials.

2. CORPUS-BASED CONCATENATIVE SOUND SYNTHESIS (CBCS) TODAY

Corpus-Based Concatenative Sound Synthesis (CBCS) is a technique used in computer music that involves constructing a sound or music piece by concatenating (joining together) smaller units of sound, such as phonemes in speech synthesis or musical phrases in music synthesis. It can be used to model an improvising instrumental musician by creating a database of recorded musical phrases or segments that can be combined and rearranged in real-time to create a musical performance that sounds like it is being improvised.

Today nearly 20 years old if one refers to the first CataRT publications [5], CBCS today enjoys an increasing popularity. Various apps today are based on similar principals (AudioStellar, Audioguide, LjudMAP or XO). The democratisation of audio analysis and machine learning tools such as the FluCoMa package (for Max, SuperCollider and Pure Data) encourages computer music practitioners to engage in this field at the crux between music creation and data science/machine learning.

2.1 Timbre Space

In spite of promising advances in the domain of deep learning applied to sound synthesis [9] [10], CBCS tools may earn their popularity from a metaphor which leads back to the early days of computer music: the notion of timbre space, developed by Wessel [11] and Grey [12], according to which the multi-dimentional qualities of *timbre* may be better understood using spatial metaphors (e.g. the timbre of the English horn being closer to basson than is it of trumpet).

The heritage of the Timbre Space metaphor can also be found in various iterations of the Orchidea [13] project, and continues to inspire generations of composers and music technologists. [1]

Pioneers in the perception of timbre studies such as Grey [12], J.C. Risset, D. Wessel, [16] or Stephen McAdams [17] [18] most often define timbre by underline what it is not. Risset and Wessel, for instance, define it as follow: *It is the perceptual attribute that enables us to distinguish among orchestral instruments that are playing the same pitch and are equally loud.*

The co-variance such parameters (pitch, loudness and timbre), however, leads Schwarz to distinguish timbre space and CBCS notions: '*Note that this concept is similar but not equivalent to that of the timbre space put forward by Wessel and Grey [7, 24], since timbre is defined as those characteristics that serve to distinguish one sound from another, that remain after removing differences in loudness and pitch. Our sound space explicitly includes those differences that are very important to musical expression.*" [19]

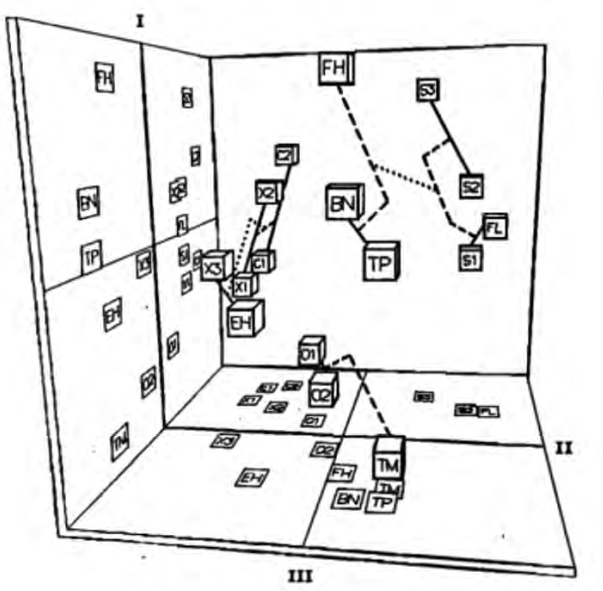

Figure 1. Multidimensional perceptual scaling of musical timbres (John M. Grey[12]). Sounds available at: https://muwiserver.synology.me/timbrespaces/grey.htm.

The workflow described in Chapter 5 gave in practice strong evidence of inter-dependance between register, timbre and dynamics, particularly when the analysis run over a single instrument sound file (e.g. 30 minutes of solo flute), and chopped in short samples. The system will then precisely be able to find similarity between instrumental passages played in the same register, same dynamic, and same playing technique (e.g. a flute playing fast trills mezzo forte, in mid-low register, with air).

2.2 Corpus-Based Concatenative Synthesis - State of the art

A wide array of technologies today can be called corpus-based concatenative synthesis, in the sense that they allow, through segmentation and analysis, to explore large quantities of sound. Some of them are presented as "ready-made" solutions, such as the recent *Audiostellar* [20], or SCMIR [2] for SuperCollider. Hackbarth's AudioGuide [21] offers a slightly different focus because it uses the morphology/timeline of a soundfile to produce a concatenated output. Within the Max world finally, two environments appear as highly customizable: IRCAM's MuBu [22] and the more recent EU funded FluCoMa [23] project. CataRT is now fully integrated in MuBu, whose purpose encompasses multimodal audio analysis as well as machine for movement and gesture recognition [24]. This makes MuBu extremely general purpose, but also difficult to grasp. The data processing tools in MuBu are mostly exposed in the pipo plugin framework [25], which can compute for in-

[1] Daniele Ghisi, co-author of the *bach* [14] package for Max, occupies a role role here as he's worked both on the *Orchidea* and *FluCoMa* projects. Some objects of his later *dada* library [15] also show an influence of CataRT (the dada.catart object was later renamed dada.cartesian). The dada.base based, finally, might have been a source of inspiration for the manipulation of databases in Max. An extract of one of his presentation is available here: https://youtu.be/LD0ivjyuqMA?t=3032

[2] A demo is available at: https://youtu.be/jxo4StjV0Cg

stance mfcc analysis on a given audio buffer [3] by embedding the pipo.mfcc plugin inside the mubu.process object.

FluComa also aims to be general purpose, but seems particularly suited to perform two popular specific tasks. With only limited knowledge of the framework nor of theory laying behind the algorithms it uses (such as those dimentionality reduction, mfcc analysis, or neural network training), the framework allows: 1/ to segment, analyse and represent/playback a sound corpus 2/ to train a neural network to control a synthesizer, in a manner reminiscent of Fiebrink's Wekinator [26].

Only the tools for segmentation, analysis representation and playback (described in detail in Chapter 5) were used here, for they precisely fit the needs of corpus-based synthesis.

3. FIRST ATTEMPTS

Some of the early versions of the FluCoMa package already provided efficient onset detection algorithms (onsetslice [4]), which encouraged the author to further dig into their environment.

At that time, all the compositional material generated with the package used the fluid.bufstat [5] object to run statical pitch analysis (and pitch confidence analysis) on each slice of a given pre-existing sound file. As can be heard in this accompanying example [6], slices were classified by pitch (the lower the index, the lower the register), although some inaccuracy could occur due to the simplicity of the analysis (section 5 will describe more elaborate strategies).

Onset detection combined with statistical pitch analysis were first used on a large scale in a piece (Chef 2.0 [7]) in which all the instrumental parts had been generated with comparable rudimentory Music Information Retrieval techniques.

Some experiments of various forms of VR control over such analysis tools already seemed promissing at that time [8]. The PatchXR software will be discussed in section 5.3.

4. MOTIVATIONS

One of the goals of the present study is to take advantage of the numerous capacities of visualisation, interaction and motion gesture data available within a VR environment. Porting to VR analysis made in Max/FluCoMa has its challenges but I am currently exploring various possible ways of interacting with a corpus-based analysis in VR. After a few trials in which the x y z coordinates of a world directly represented audio descriptors such as loudness pitch and centroid [9], I more systematically used mfcc analysis and dimensionality reduction, as described in the following section.

The use of machine learning (dimensionality reduction) in the latter case renders a world in which the absolute coordinates of each point has no more link to the descriptor space (the high sounds cannot be mapped to the y axis for instance), but offers compelling results for clustering information relating to the different playing styles of the instrument that is being analysed: as an example, in this extract [10] based on flute sounds, the opening shows a clear opposition between two types of gestures: 1/ staccato notes and 2/ legato scale-like type of material. This contrast in sound is made explicit by a movement of the avatar which jumps from a cluster of buttons to another.

5. WORKFLOW

After numerous attempts at listening to and playing with this system [11], I am now investigating how to diversify in VR metaphors for exciting the corpus based synthesis engine, as well as the different ways in which the synthesis may be rendered on an orchestra of RaspberryPi-equipped loudspeakers (see Chapter 6).

5.1 Corpus Selection

My experiments have focussed on musical instrument corpora almost exclusively. The tools presented here can efficiently generate plausible virtuosic instrumental music (as was sometimes the case in the piece Chef 2.0 discussed earlier), but recent uses found more satisfying results in slower, quieter, "Feldman-like" types of textures. Various limitations on the playback side (either in VR, or on a Pure Data sampler for RaspberryPi described in Chapter 6) have imposed restrictions in the first stages on the amount of data it could handle (less than 5minutes in AIFF in VR) or the number of slice the sample could be chunked into (256 because of limitation of lists in Max). Both limitations were later overcome (use of the ogg format in VR, increase of internal buffer size in fluid.buf2list), thus allowing for far more convincing models.

5.2 Analysis in FluCoMa

Using concatenative synthesis to model an improvising instrumental musician typically involves several steps:

1. Segmentation of a large soundfile: This involves dividing a large audio recording of the musician's performance into smaller units or segments.

[3] MFCC stands for Mel-Frequency Cepstral Coefficients. It is a type of feature extraction method that is commonly used in speech and speaker recognition systems. MFCCs are used to represent the spectral characteristics of a sound in a compact form that is easier to analyze and process than the raw waveform. They are calculated by applying a series of transformations to the power spectrum of a sound signal, including a Mel-scale warping of the frequency axis, taking the logarithm of the power spectrum, and applying a discrete cosine transform (DCT) to the resulting coefficients. The resulting coefficients, which are called MFCCs, capture the spectral characteristics of the sound and are commonly used as features for training machine learning models for tasks such as speech recognition and speaker identification.

[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 Result: https://youtu.be/Sc4Ye_rnSO8?t=9893. Although not relevant to the discussion here, the arm movement of the conductor was mapped to the speed of the cursor on the performers' screens with the help of INScore [27] and Gesture Follower [28] in MuBu for Max.

[8] https://youtu.be/DC_BL_HGPLA

[9] https://youtu.be/1LHcbYh2KCI?t=19

[10] https://youtu.be/777fqIIJCY4

[11] For cello: https://youtu.be/L-MiKmsIzjM For various instruments: https://www.youtube.com/playlist?list=PLc_WX6wY4JtnNqu4Lwe2YzEUq9S1IMvUk For flute: https://www.youtube.com/playlist?list=PLc WX6wY4JtlbjLuLHDZhlx78sTDm

2. Analysis: These segments are then organised in a database according to various descriptor data (mfcc in our case).
3. Scaling/pre-processing: scaling is applied for better visualisation.
4. Dimension reduction: Based on mfcc descriptors, the dimensionality of the data is reduced in order to make it more manageable and easier to work with. This can be done using techniques such as principal component analysis (PCA) singular value decomposition (SVD), or Uniform Manifold Approximation and Projection (UMAP, preferred in our case).
5. Near neighbours sequencing: Once the segments have been organised and analysed, the software selects and combines them in real-time based on certain input parameters or rules to create a simulated musical performance that sounds like it is being improvised by the musician. We use here a near neighbours algorithm, which selects segments that are similar in some way (e.g., in terms of pitch, loudness, or timbre - thanks to similarities revealed by umap on mfccs in our case) to the current segment being played.

We will now describe these steps in further detail:

5.2.1 Slicing

We saw in Chapter 3 how slicing musically trigger possibilities. In MuBu onset detection is done with pipo.onseg or pipo.gate. FluCoMa expose five different onset detection algorithms:

1. fluid.ampslice: Amplitude-based detrending slicer
2. fluid.ampgate: Gate detection on a signal
3. fluid.onsetslice: Spectral difference-based audio buffer slicer
4. fluid.noveltyslice: Based on self-similarity matrix (SSM)
5. fluid.transcientslice: Implements a de-clicking algorithm

Onsetslice only was extensively tested. The only tweaked parameters were a straigh-forward "threshold" as well as a "minslicelength" argument, determining the shortest slice allowed (or minimum duration of a slice) in hopSize. This introduce a common limitation in CBCS: the system strongly biaises the user to choose short samples for better analysis results, and more interactivity, when controlling the database with a gesture follower. Aaron Einbond remarks in the use of CataRT how short samples most suited his intention: "*Short samples containing rapid, dry attacks, such as close-miked key- clicks, were especially suitable for a convincing impression of motion of the single WFS source. The effect is that of a virtual instrument moving through the concert hall in tandem with changes in its timbral content, realizing Wessel's initial proposal.*"[29]

A related limitation of concatenative synthesis lies in the fact that short samples will demonstrate the efficiency of the algorithm [12], but at the same time moves away from the "plausible simulation" sought in the present study. A balance therefore must be found between the freedom imposed by large samples, and the refined control one can obtain with short samples.

A direct concatenation of slices clicks in most cases on the edit point, which can be avoided through the use of ramps. The second most noticeable glitch on concatenation concerns the interruption of low register resonances, which even a large reverb fails making sound plausible. Having a low threshold and large "minslicelenght" results in equidistant slices, all of identical durations, as would do the pipo.onseg object in MuBu.

Because we listen to sound in time, this parameter responsible for the *duration of samples* is of prior importance.

5.2.2 mfcc on each slice - across one whole slice/segment

Multidimensional MFCC analysis: MFCC (Mel-Frequency Cepstral Coefficient) analysis is a technique used to extract features from audio signals that are relevant for speech and music recognition. It involves calculating a set of coefficients that represent the spectral envelope of the audio signal, and can be used to capture the spectral characteristics of the musician's playing style.

5.2.3 statical analysis over each slice

BufStats is used to calculate statistical measures on data stored in a buffer channel. A buffer here is a type of data structure that holds time-series information, audio descriptor data in this case. BufStats calculates seven statistics on the data in the buffer channel: mean, standard deviation, skewness, kurtosis, low, middle, and high values. These statistics provide information about the central tendency of the data and how it is distributed around that tendency. In addition to calculating statistics on the original buffer channel, BufStats can also calculate statistics on up to two derivatives of the original data, apply weights to the data using a weights buffer, and identify and remove outlier frames. These statistical measures can be useful for comparing different time-series data, even if the original data is different lengths, and may provide better distinction between data points when used in training or analysis. The output of BufStats is a buffer with the same number of channels as the original data, with each channel containing the statistics for its corresponding data in the original buffer.

5.2.4 Normalization

The FluCoMa package proposes several scaling/preprocessing tools, amongst which normalization and standardization were used. Standardization and normalization are techniques used to transform variables so that they can be compared or combined in statistical analyses. Both techniques are used to make data more comparable, but they work in slightly different ways.

Standardization involves scaling a variable so that it has a mean of 0 and a standard deviation of 1. This is done by

[12] e.g. https://youtu.be/LD0ivjyuqMA?t=3032

subtracting the mean of the variable from each data point and then dividing by the standard deviation. Standardization is often used when the variables being compared are on different scales or have different units of measurement. It allows for comparison of variables that would otherwise be difficult to compare directly.

Normalization involves scaling a variable so that it has a minimum value of 0 and a maximum value of 1. This is done by subtracting the minimum value of the variable from each data point and then dividing by the range (i.e., the difference between the maximum and minimum values). Normalization is often used when the variables being compared have a skewed distribution, or when the variables are not normally distributed. It allows for comparison of variables that would otherwise be difficult to compare directly due to the skewness of their distribution.

Standardization scales a variable to have a mean of 0 and a standard deviation of 1, while normalization scales a variable to have a minimum value of 0 and a maximum value of 1. Normalization ccaling was found easier to use both in 2-D (in FluCoMa, the fluid.plotter object), as well as in the VR 3D world in which the origin corresponds to a corner of the world. The fluid.nomalize object features an "@max" attribute (1 by default), which then maps directly to the dimensions of the VR world.

5.2.5 Dimensionality Reduction

Dimensionality reduction is a technique used in machine learning to reduce the number of features (dimensions) in a dataset. The goal of dimensionality reduction is to simplify the data without losing too much information. Various dimensionality reduction algorithms are presented in an early FluCoMa study[30], with interestingly no mention of UMAP, later favoured.

SOM in one of the most popular algorithms for dimensionality reduction. It is implemented in the ml.star[31] library for Max, a simple hands-on library for machine learning, just one amongst the vast ammount of frameworks and machine learning algorithm famous across the NIME community [32] [33] [34] [35].

SOM (Self-Organizing Map) and UMAP (Uniform Manifold Approximation and Projection) are both techniques for dimensionality reduction. SOM is a type of neural network that is trained using unsupervised learning. It consists of a grid of neurons, each of which is associated with a set of weights. The SOM is trained by presenting it with input data and adjusting the weights of the neurons so that similar input patterns are mapped to nearby neurons on the grid. The resulting map is a low-dimensional representation of the input data that preserves the topological structure of the original data. UMAP, on the other hand, is a non-linear dimensionality reduction technique that is based on the principles of topological data analysis. It uses a combination of techniques such as k-nearest neighbours, weighted graph construction, and low-dimensional embedding to produce a low-dimensional representation of the input data. Unlike SOM, which is limited to a fixed grid structure, UMAP can produce a continuous, flexible representation of the data. Both SOM and UMAP can be useful for visualising high-dimensional data and for discovering patterns and relationships in the data. However, UMAP has some advantages over SOM, including the ability to handle large datasets more efficiently and the ability to produce more interpretable results.

UMAP (Uniform Manifold Approximation and Projection) can be used to visualize high-dimensional data in a lower-dimensional space. When applied to sound data analysed with MFCC (Mel-Frequency Cepstral Coefficients), UMAP reduces the dimensionality of the data and creates a visual representation of the sound in a 2- or 3-dimensional space. MFCCs, again, are a feature extraction technique commonly used in speech and audio processing. They involves decomposing a sound signal into a set of frequency bands and representing the power spectrum of each band with a set of coefficients. The resulting MFCC coefficients capture important spectral characteristics of the sound signal (albeit hardly interpretable by the novice user), such as the frequency and magnitude of the spectral peaks. By applying UMAP to the MFCC coefficients of a sound signal, it is possible to create a visual representation of the sound that preserves the relationships between the different MFCC coefficients (see Fig. 2). This can be useful for tasks such as exploring the structure of a sound dataset, identifying patterns or trends in the data, and comparing different sounds.

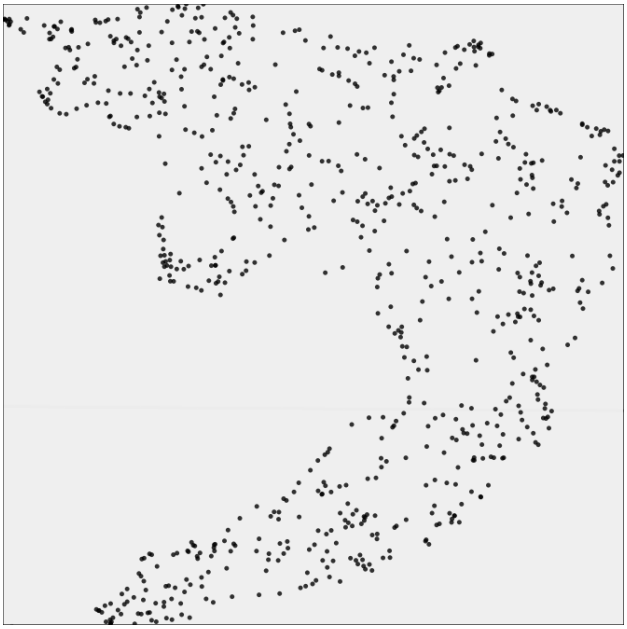

Figure 2. Dimensionality reduction of MFCCs help revealing spectral similarities. UMAP outputs coordinates in 2d or 3d.

UMAP is therefore used for its clustering abilities in the first place, helping for classification purposes. It helps identifying patterns or trends that may not be evident from the raw data. Most importantly, the non-linear dimensions proposed by UMAP (whether in 2d in Max or in 3 dimensions in PatchXR, and when compared to linear analyses in which, for instance, x, y and z correspond to pitch, loudness and centroid) gave far more "intelligent" clustering

than more conventional parameter-consistent types of representations.

5.2.6 Neighbourhood queries

The neighbourhood retrieval function in a slightly different way each time, but is based in FluCoMa on K-d trees and and the knn algorythm. In MuBu , the mubu.knn object, as well as the ml.kdtree object ml.star, give very comparable result than those achievable with fluid.kdtree.

K-d trees (short for "k-dimensional trees") and k-nearest neighbours (k-NN) are two algorithms that are related to each other, but serve different purposes.

A k-d tree is a data structure that is used to efficiently store and query a set of points in a k-dimensional space. It works by partitioning the points into a binary tree, with each node in the tree representing a hyperplane that splits the space into two halves. The points are recursively partitioned into the left and right subtrees based on which side of the hyperplane they fall on. By organising the points in this way, it is possible to quickly find the nearest neighbours of a given point by searching only a subset of the tree rather than the entire set of points.

On the other hand, the k-NN algorithm is a machine learning algorithm that is used for classification or regression. Given a set of labeled points and a new, unlabelled point, the k-NN algorithm determines the k points in the set that are nearest to the new point, and then uses the labels of those points to predict the label of the new point. The value of k is a hyper-parameter that is chosen by the user, and it determines the number of neighbours that are considered when making the prediction.

In summary, a k-d tree is a data structure that is used to store and efficiently query a set of points in a k-dimensional space, while the k-NN algorithm is a machine learning algorithm that is used for classification or regression. Both algorithms are often used in applications such as pattern recognition, image classification, and data mining.

While CataRT or Audiostellar are typically used for generation of electronic textures/sound design, I have most often used FluCoMa to generate monophonic instruments (one performer plays one instrument at a time), in which the avatar reproduces what knn would do with an automated instrument: he will privilege in his choice the sample he can reach at hand, rather than jump large distance between 2 items (see Fig. 3) .

5.3 PatchXR

PatchXR [36] is a playful digital audio workstation for making music in VR. It's core metaphor corresponds to what the FluCoMa team calls CCE (creative coding environments) insofar as it functions in many ways like Max or Pure Data.

One reason for using VR to explore a 3D dataset is that it allows users to interact with the data in a more natural and immersive way, using it as a tool for data visualisation and analysis. Users can move around and explore the data from different angles, which can help them to better understand the relationships between different data points and identify patterns. Users get a more intuitive sense of the data and

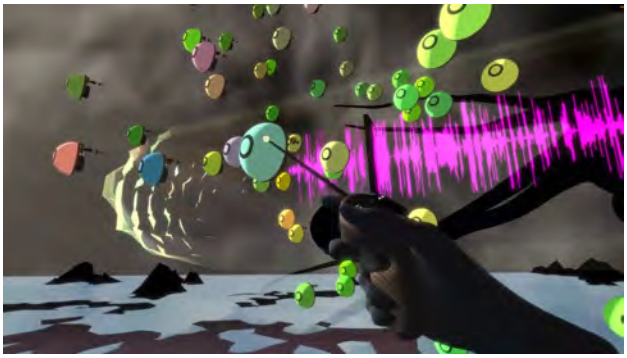

Figure 3. A VR interface in which each button in the world corresponds to a slice of the sound file. Machine learning helps bringing closer sounds that share common spectral characteristics.

better understand how it is structured and how the different data points relate to one another.

The structure of a .patch file (a patchXR world) follows the syntax of a .maxpat (for Max) or .pd file (for pure data) in the sense that it first declares the objects used, and then the connexions between them. This structure made it relatively trivial to generate a javascript routine taking as input a dictionary (json file) with each segment's 3d coordinates, and as output a new .patch file (a world accessible in VR, see general workflow on Fig. 4).

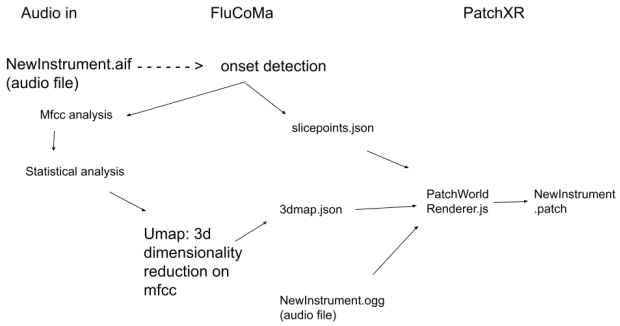

Figure 4. General Workflow: from an input audio file to its .patch 3d representation in PatchXR.

5.3.1 Interaction and OSC communication

PatchXR exposes a wide range of blocks (a block corresponds to an object in Max or Pure Data) making it simple to access gesture data such as:

- The position/distance between hands/controllers and a reference.
- The rotation angles (x y z) of both hands' controllers
- 2-d touchscreen like controllers, where the user moves the xy position of a selector across a plane by manually grabbing it.
- 2-d lazer-like controllers, where the user moves the xy position of a selector remotely, as if using a lazer pointer towards a remote screen

- 2-d pads, which allow to access the velocity at which the pad is hit
- 3-d theremine like controllers, where the user moves the xyz position of a selector across a plane by manually grabbing it.
- 1-d sliders, knobs, buttons...

One of the current challenges consists in diversifying the ways in which the corpus is queried. One to one mapping of UMAP results such as those described in Chapter 5.3 favour simulations for soloists, or duo in a multiplayer mode, in which the button interface buttons are facing each other, in order to prompt the players to face each other (see https://youtu.be/LP1g79BdIpY).

A simulation for more instruments, particularly when playing alone, encourages to use higher level type of control over automata, most importantly the simple ability to automatically concatenate: play the next sample as soon as the previous one has stopped.

6. FUTURE WORKS: THE RASPBERRY PI ORCHESTRA

In the frame of an artist residency at UCA (Université côte d'azur), the investigation questions how the tools presented above (those concern with the domain of Music Information Retrieval - MIR) may serve the control an immersive platform made of an orchestra of 64 *Pré* modules. [37]

At the time of writing this report, most satisfying results were achieved by sending messages to each RaspberriPi independently, according to its specific (static) IP address, with a simple syntax of a 2-integer list corresponding to: 1/which buffer to lookup 2/ which slice in this buffer to play. Pursuing on elaborations of timbre-space illustrations, the *Pré* modules, with the different acoustics its mobility allows, will encourage contrasted density of events according to the acoustics of the space in which the listening experience is happening.

7. CONCLUSIONS

After a discussion on the valuation of scores as maps, we've proposed a workflow for corpus-based concatenative synthesis CBCS, arguing that machine learning tools for data visualisation offer revealing and exploitable information about the timbral quality of the material that is being analysed. From a composer's perspective, the disappearance of the x time axis prompts to envisage composition not in a narrative sense (understood according reflections on new media and the notion of data-based art developed by [7]), but rather, as "time canvasses", as understood by Morton Feldman.

The discussed tools for "machine listening" (FluCoMa, MuBu) help building intelligent instruments with relatively small amounts of data, the duration of samples appear crucial in CBCS. A balance must be found between 1/ short duration sample analysis which are easier to process and categorise and 2/ long samples which sound more natural in the context instrument-based simulations.

Acknowledgments

I am grateful for the support of UCA/CTEL, whose artist residency research program has allowed to hold these experiments, and for the support of PRISM-CNRS.

8. 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] N. Goodman, *Languages of Art: An Approach to a Theory of Symbols*. Hackett, 1976. [Online]. Available: https://books.google.fr/books?id=e4a5-ItuU1oC

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

[5] 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

[6] 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

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

[8] 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

[9] 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

[10] 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

[11] 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

[12] 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.

[13] 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

[14] 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

[15] 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.

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

[17] 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.

[18] 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.

[19] 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

[20] 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.

[21] 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

[22] 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.

[23] 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

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

[25] 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.

[26] 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.

[27] 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

[28] 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.

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

[30] 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.

[31] B. D. Smith and G. E. Garnett, "Unsupervised play: Machine learning toolkit for max," in *New Interfaces for Musical Expression*, 2012.

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

[33] 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

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

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

[36] 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.

[37] *≪ PrÉ ≫ : connected polyphonic immersion*. Zenodo, Jul. 2022. [Online]. Available: https://doi.org/10.5281/zenodo.6806324