Agglomerative Clustering for Audio Classification using Low-level Descriptors

Frédéric Le Bel

Agglomerative Clustering for Audio Classification using Low-level Descriptors.

Frédéric LE BEL – frdric.lebel@gmail.com
Centre de recherche en informatique et création musicale, EDESTA – Université Paris 8,
Pédagogie et action culturelle, IRCAM – Centre Pompidou,
Paris – France, 2015-2016.

1. INTRODUCTION

Mainly inspired by the work of Geoffroy Peeters, Diemo Schwarz and Grégoire Carpentier in the field of music information retrieval[1] (MIR), this document presents the work that I have achieved in the frame of Ircam Cursus 2 (Specialized Training in Composition, Research and Music Technology) under the guidance of Mikhail Malt and Hèctor Parra in 2015-2016. The main lead of this project was to elaborate a computerized classifier for large corpuses of sounds. In other words, the idea was to be able to organise and re-organise easily a database under specific constraints or concepts that could be useful in preparation of scoring a musical piece. Although parts of this idea have been investigated to achieve different tasks such as corpus-based concatenative synthesis [Schwarz, 2006], musical genre recognition [Peeters, 2007] and computer-assisted orchestration [Carpentier, 2008], the challenge remained, and still remains, to find a way to adapt this approach for compositional purposes; not only to generate material but mostly to analyse, to explore and to understand the full potential, inside out, of a sound material. That may be seen as a kind of audio data mining[2] applied to computer-aided composition. As the title of this document reveals some pieces of answer to this interrogation, the following explanations aim at unfolding the algorithmic structure in order to examine the methodology, to discuss a few important co-lateral problematics and to analyse different clustering results. The objective is also to take a look at the influence of this research on a first artistic outcome: Alors que le monde est décomposé, for piano & electronics - performed by Wilhem Latchoumia, to expose the limitations of this approach in relation to specific artistic goals and finally, to discuss ideas for future development.

2. DEFINITIONS

Before going further, it is important to clarify two fundamental elements of this work.

a. Agglomerative clustering

Agglomerative clustering is one of the two strategies related to hierarchical clustering (also called hierarchical cluster analysis or HCA). In data mining and statistics, HCA is an unsupervised learning method[3] of cluster analysis that seeks to build a hierarchy of clusters, usually presented in a dendrogram (from Greek *dendro* "tree" and *gramma* "drawing"). As opposed to the divisive strategy, the agglomerative one is a "bottom up" approach where each element starts in its own cluster and pairs of clusters are merged as one moves up the hierarchy [Hastie, 2009].

b. Low-level descriptors and audio features

Low-level descriptors are mathematical operators that measure different types of information on given audio signals. They are usually used to transform a raw signal into a smaller space of variables representing a specific audio feature such as the loudness, the sharpness or the spectral variation, to name a few. Audio features are thus measurable properties of sounds, usually containing information relevant for pattern recognition [Malt, 2012].

3. STRUCTURAL OVERVIEW

In order to have a clear idea of the overall process of classification, here is an outline of the workflow.

---

[1] Music information retrieval (MIR) is the interdisciplinary science of retrieving information from music. MIR is usually used to categorize, manipulate and even create music. Those involved in MIR may have a background in musicology, psychology, academic music study, signal processing, machine learning or some combination of these.

[2] Data mining is an interdisciplinary subfield of computer science. It is the computational process of discovering patterns in large data sets involving methods at the intersection of artificial intelligence, machine learning, statistics and database systems. The overall goal of the data mining process is to extract information from a data set and transform it into an understandable structure for further use.

[3] Unsupervised learning is the task of inferring a function to describe hidden structures from unlabeled data. Since the examples given to the learner are unlabeled, there is no error or reward signal to evaluate a potential solution. This distinguishes unsupervised learning from supervised learning and reinforcement learning.

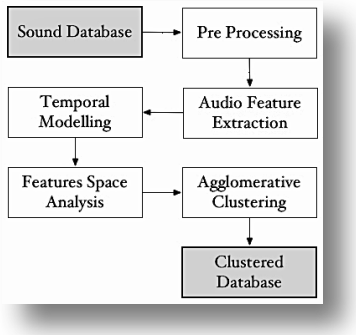

Figure 1. Workflow chart

4. PRE PROCESSING

Considering that the database is composed of pre-segmented sounds (user defined), the process of classification starts with a very simple but crucial procedure which is to prepare it for later audio feature extraction by applying different types of pre-processing to its components. In other words, that is to optimize the input format for extracting higher quality features from the signal. It may seem superficial, but this step may have a strong impact on the clustering results.

First of all, it is important to discuss the sampling rate and the bit depth because these parameters can be manipulated for aesthetic purposes. As one would like to use very low sampling rates and bit depth to transform any given sound, it is important to resample the sound corpus at a relatively high, but most importantly, a uniform sampling rate and bit depth for the sake of feature analysis. In the frame of this research, a sampling rate of 48kHz and a bit depth of 24bit appeared to be the optimal configuration for preserving the wide range and the complexity of sounds but also to avoid oversizing the database.

The second element of pre-processing is simply the number of channels. Unless the position in space is required for the analysis, it is preferable to mix down all the sound files to a single channel (monophonic). It may alter some elements of the perception but in most cases, it provides a better representation of the sound as a whole and surely reduces the computing time. Following this procedure, one should consider to normalize the amplitude of each sounds for a better feature extraction; unless the volume of sounds is a parameter of clustering.

The third aspect is a delicate matter because it is about cleaning the sounds. Proceeding to a complete and automatized cleaning (de-noising, hum removal, spectral repairing, etc.) of the sound can quickly damage some of the key elements of aesthetic and lead to strange classification results. That being said, this part of pre-processing should be approached with special care and be adapted to every case. The main idea is to use the appropriate techniques to remove unwanted elements from the sound files. Since the feature extraction should be as precise as our perception of sounds, an accurate listening of the sounds is inevitable in order to avoid introducing undesired peaks and gaps that could interfere with the quality of the clustering.

The last element in preparation of the database is about segmenting and trimming the sound files. The segmentation should be done in order to break down a stream of sound into single elements in order to simplify the analysis and obtain an intelligible clustering. It is also important to consider reducing the length of sounds with little variations over time by selecting a sample that best represent the whole. After the segmentation, it must be considered to proceed to a basic trim of the sound files in order to avoid introducing unwanted silence into the extracted features. The basic trim consists of removing silence from the ends of each sound files. Another trim approach that is used to optimize the feature extraction, and computing time, is to reduce the signal to its effective duration. The latter is approximated by the time the energy envelop is above a given threshold. A threshold of 40% appeared to be generally low enough to preserve the part of the signal that is perceptually meaningful.

As mentioned above, one should choose wisely what types of pre-processing are to be done in preparation of the database. Depending on the objectives, one could choose to work with stereo files for spatial analysis or decide to keep the original volume of each sound file intact for a clustering based on general amplitude. The principle remains the same for cleaning, segmenting and trimming the sound files. The idea is to clearly determine what is perceptually consistent for the clustering and prepare the database consequently.

5. AUDIO FEATURES EXTRACTION

The second step towards the clustering process is to build the data structure upon which the classification paradigms will take basis. At this stage, very specific properties of the sounds need to be extracted in order to compare them in different ways. The idea is to be able to analyse the database on multiple levels and to have a multidimensional understanding of its components.

a. Low-level descriptors

In the frame of this work, different engines were used to extract various audio features. For low-level features, the Ircamdescriptors-2.8.6[4] seemed to be the most adequate. For other tasks such as partial tracking and chord sequence analysis, the pm2[5] engine was used. In the same area, the superVP[6] engine was chosen to extract the peak analysis and the masking effects[7]. Without being exhaustive, the selection of features to extract covers a wide range of the complexity of the sounds. Under two different models (physical and perceptual), the following descriptors can be separated into five categories: global temporal, instantaneous temporal, instantaneous harmonic, instantaneous energy and instantaneous spectral.

Physical model	Perceptual model
1) Global temporal descriptors	**2) Instantaneous energy descriptors**
a. Log attack time b. Temporal increase c. Temporal decrease d. Amplitude modulation (amp. & freq.) e. Energy envelope f. Effective duration (time) g. Temporal centroid	a. Loudness b. Loudness spread c. Relative specific loudness
3) Instantaneous temporal descriptors	**4) Instantaneous spectral descriptors**
a. Auto-correlation b. Signal zero crossing rate	a. Mel frequency cepstral coefficient (MFCC) b. Sharpness (spectral centroid) c. Spectral spread d. Spectral skewness e. Spectral kurtosis f. Spectral decrease g. Spectral roll-off h. Spectral variation i. Spectral deviation j. Spectral flatness k. Spectral crest
5) Instantaneous harmonic descriptors	
a. Fundamental frequency (f0) b. Inharmonicity c. Noisiness d. Chroma e. Chord sequence analysis f. Partial tracking g. Peak analysis (basic peak analysis) h. Masking effects (advanced peak analysis) i. Harmonic spectral deviation j. Odd to even energy ratio k. Tristimulus	

Table 1. Selected low-level descriptors

The two models (physical and perceptual) imply a pre-processing stage in order to provide the adequate signal representations for computing the descriptors. In both cases, but depending on the feature to extract, the pre-processing may consist of:

- Energy envelope estimation (sample based or non-overlapping frames),
- Temporal segmentation (overlapping windowing),
- Short-time Fourier transform (STFT) calculation,
- Harmonic sinusoid model approximation.

[4] Ircamdescriptors-2.8.6 is a command line software for the extraction of audio features from a raw signal.
[5] Partial manager 2 (pm2) is a command line software for sound analysis based on the sinusoidal model.
[6] Super Phase Vocoder (superVP) is an executable for analysis, synthesis and transformations of sounds based on the FFT model.
[7] Masking effects occur when a sound is made fully or partially inaudible by another sound or by some of its internal components.

The perceptual model differs from the other one because it implies an additional set of pre-processing that is to simulate the human auditory system. This particular chain of processing consists of a mid-ear filtering emulation and a logarithmic band conversion, namely the Mel or the Bark bands [Zwicker, 2007]. Below is the complete flowchart of the feature extraction process.

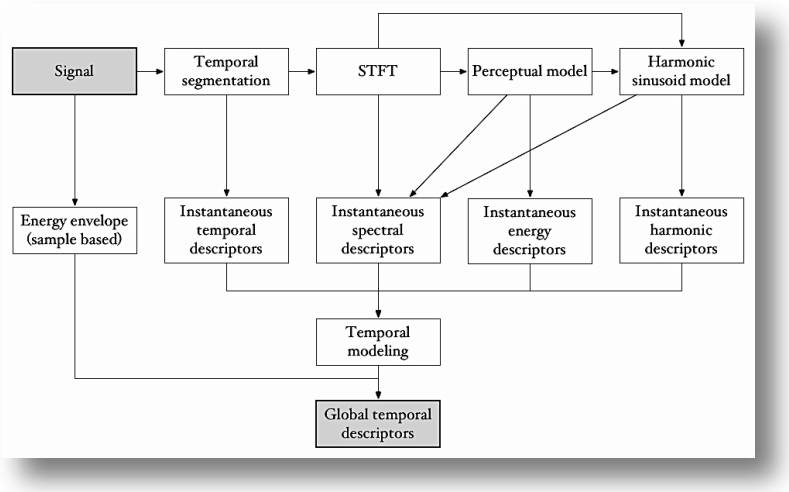

Figure 2. Features extraction flowchart

As shown in figure 2, the spectral descriptors can be computed directly over the STFT but also over the perceptual model and the harmonic sinusoid model. In this work, the spectral descriptors have been mostly computed over the perceptual model and others over the harmonic sinusoid model such as the centroid, the spread, the skewness, the kurtosis, the decrease, the roll-off and the variation. It is also shown that the harmonic descriptors may be computed over the STFT or the perceptual model as for the harmonic spectral deviation, the odd to even energy ratio and the tristimulus. In this case, both approaches have been used.

b. Physical model

The energy envelope is used for the calculation of the global temporal descriptors: log-attack time, temporal centroid, etc. It relies on the instantaneous root mean square[8] (rms) values computed on the signal. These descriptors are then computed on the energy envelope and not on the raw signal. In the frame of this work, a window size of 100ms (10Hz) appeared to be appropriate in order to apply a low-pass filter independent from the sampling-rate.

The short-time Fourier transform (STFT) is to divide a longer time signal into shorter segments of equal length for computing the Fourier transform (FFT) separately on each shorter segment. It is used to determine the sinusoidal frequency and phase content from specific bits of a signal as it changes over time. In other words, it is used to convert a signal from its original domain (time) to a representation in the frequency domain [Cooley, 1965]. In this case, a window size of 200 milliseconds (ms) and hop size of 50ms were used in order to analyse sounds with pitches down to 20Hz (4 periods of 20Hz = 200ms). Although these settings were rather efficient for this work, one should be careful with these parameters if working with very short sounds. The sound with the shortest duration within the database defines the minimum window size for computing the STFT. This floor duration then imposes itself in the frequency domain as the lowest pitch to be detected. Regarding the pre-processing, the length of the window (200ms in this case) should be added, as silence, at the beginning of each sound file for the analysis to be performed through the whole entity. Since the STFT requires a full window size to render a value, the waveforms need to be shifted forth to be included in the first frame of the analysis.

The harmonic sinusoid model is an estimation of the frequency and amplitude content of an audio signal computed on its STFT. For each window of a signal, the STFT peaks are compared to the local fundamental frequency (f0) and those being close to multiples of this frequency are chosen to define the current frequency and amplitude content [Depalle, 1993].

[8] The root mean square (rms) is defined as the square root of the arithmetic mean of the squares of a set of numbers.

c. Perceptual model

The mid-ear filtering emulator is to simulate the attenuation due to the human middle ear. Based on the equal-loudness contours, namely the Fletcher-Munson curves[9], this filter is applied to the FFT of each frame [Moore, 1997]. The figure 3 is a representation of the middle-ear EQ transfer functions on which the filter is based.

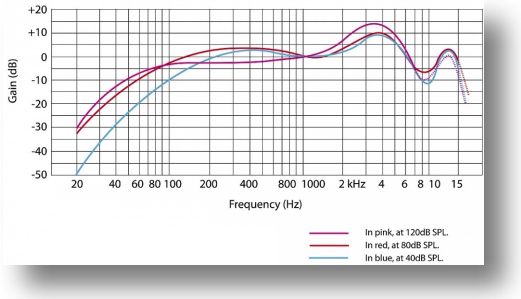

Figure 3. Mid-ear EQ Transfer Curves

The Mel scale is a perceptual scale of pitches judged by listeners to be equal in distance from one another. Thus, this aspect of the human auditory system can be modelled with a set of critical band filter. The Mel bands are one of these. The Mel scale is linear at low frequencies (< 1000Hz) and logarithmic at high frequencies (> 1000Hz). This band conversion is particularly popular in the automatic speech recognition (ASR) community where it is used to calculate the famous MFCC (Mel Frequency Cepstral Coefficients) [Rabiner, 1993].

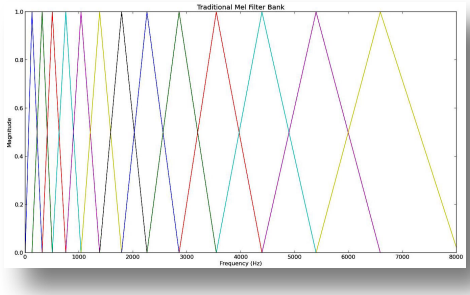

Figure 4. Traditional Mel Filter Bank

The Bark scale is related to the Mel scale but is somehow less popular. Although the Mel bands are used to calculate the MFCC (because of its popularity in the ASR community), the Bark bands can model a better approximation of the human auditory system [Zwicker, 1980]. For that reason, all the perceptual descriptors, except the MFCC, are computed over the Bark bands.

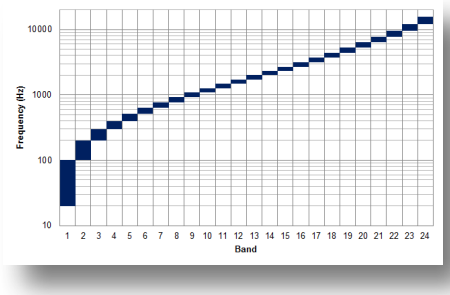

Figure 5. Bark bands

[9] The Fletcher-Munson curves are one of many sets of equal-loudness contours for the human ear. An equal-loudness contour is a measure of sound pressure (dB SPL) over the frequency spectrum, for which a listener perceives a constant loudness when presented pure steady tones.

Calculation of the critical band energy:
The linear frequency axe is first converted into Mel or Bark scale. The new scale axe is then divided into equally spaced bands; 20 bands for the Mel scale and 24 bands for the Bark scale. After weighting by the Mel or the Bark band window shape, the energy of each FFT bins corresponding to each Mel or Bark bands are then summed up to form the perceptual bands [Peeters, 2004].

Amplitude and frequency scales:
When features are extracted from the signal spectrum, from the harmonic peaks or from the filter banks, various amplitude scales may be considered: the linear amplitude (raw amplitude), the amplitude converted to an energy scale (amplitude^2) or an amplitude converted to a logarithmic scale (log-amplitude). Regarding the frequencies, they can be considered as linear frequencies (raw frequency) or they can be converted to a logarithmic scale (log-frequency). The choice of these parameters are of little importance for clustering purposes; the idea is to work with a single setting throughout the whole process. This work has been done using linear amplitudes and linear frequencies all along.

6. TEMPORAL MODELLING

The third step, before proceeding to the features space analysis, is the temporal modelling of the instantaneous descriptors. At this stage, all the corresponding features can be represented by a unidimensional or a multidimensional time series. In other words, the features are structured as series of data points listed (or graphed) in time order. Thus they are a sequence of discrete-time data. Below are two graphic representations of different features. The leftmost is a unidimensional time series and the rightmost is a multidimensional time series.

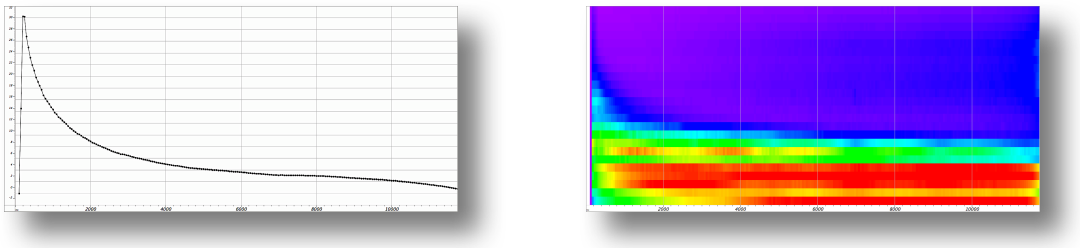

Figure 6. [left] Instantaneous loudness (LDN), [right] Instantaneous relative specific loudness (RSL)

a. Global temporal model

As seen in figure 6, the features (except the global descriptors) are extracted using a frame-by-frame analysis. The instantaneous descriptors can then be used in real-time context for recognition (to some extent) but they can also be used in order to create a Hidden Markov Model representing the behaviour of a feature [Cella, 2011]. Another approach is to model the features over time using simple statistics such as calculating the means, the variances and the derivatives [Peeters, 2004]. Beforehand, each feature may be weighted by the energy of the corresponding signal in order to strengthen the perceptual model, and may also be filtered using a median filter to remove speckle noise.

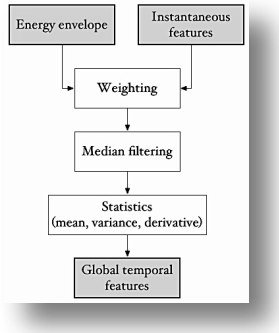

Figure 7. Global temporal modelling flowchart

The median filter is a nonlinear digital filtering technique able to perform some kind of noise reduction on a signal. Such noise reduction is a typical pre-processing step to improve the results of later processing [Huang, 1979]. The

way of the median filter is to run through the signal frame by frame, replacing each of them with the median of neighbouring frames. This process is also known as the sliding window median. A typical sliding window involves five consecutive frames (5*200ms/frame = 1 second of sound) for each iteration (5^{th} order filter = 2 before + 1 current + 2 after) in the case of unidimensional time series. Although different window patterns are possible for multidimensional signals (box pattern and cross patterns), the 5^{th} order filter mentioned before should be applied independently on each dimension of the multidimensional features. Thus, each dimension is considered as an individual time series and is treated consequently. Nonetheless, the resulting temporal model would remain multidimensional (vectors), as the unidimensional features would result in unidimensional temporal models (scalars). Since the multiple dimensions originate from a merging process (FFT bins merged into Bark bands), it seems to be logical to avoid box or cross filtering elements that are purposefully assembled.

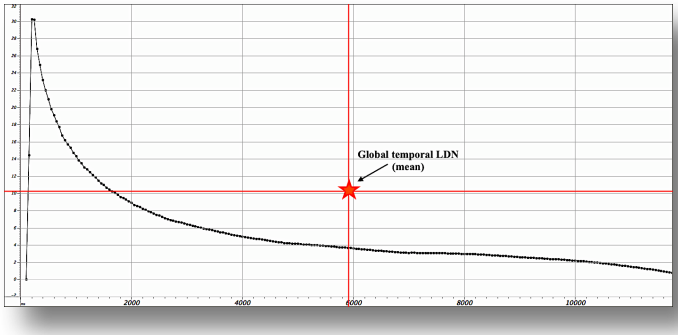

Figure 8. Global temporal LDN (mean) over Instantaneous LDN

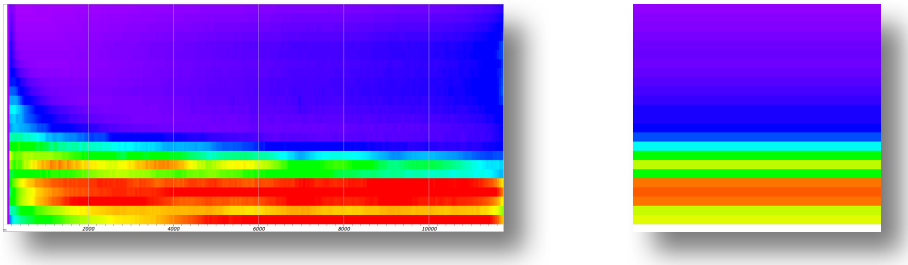

Figure 9. [left] Instantaneous RSL, [right] Global temporal RSL (means/Bark bands)

The idea of creating such global temporal models is to obtain a general, but still accurate, description of sounds. To simplify the data as such is also an assumption on how the sounds are perceived and described by humans in a clustering context. In order to achieve other tasks, different types of temporal modelling should be investigated.

Although the scheme of temporal modelling described above appears to be very useful to classify sounds with simple internal variations (a musical note played with an acoustic instrument), it should be adapted to every context. The database used in the frame of this work is mostly maladapted to this procedure due to the internal complexity of its components. For that reason, the instantaneous features were only weighted by the energy (amplitude^2) of their corresponding signal and filtered with the 5^{th} order median filter mentioned previously. Thus no statistics (mean, variance, derivative) were used to model them. Nonetheless, the instantaneous features do not bypass the question of temporal modelling. They actually raise another question related to the alignment of time series with different length. Now, the problem is to compare sounds with different durations.

b. Temporal alignment

From both a perceptual and a mathematical angle, the question of temporal alignment should be given some pieces of answer before going further into the clustering process. From a mathematical angle, this question is fundamental because most of the following formulations (see Features space analysis) require equal length variables in order to yield correct results. From a perceptual angle, this question should be approached with different assumptions.

- The first assumption is to consider that sounds can be compared only if they exist simultaneously. The underlying statement is that the extent of the comparison is equal to the extent of the shortest sound.

Considering a pair of sounds synchronized at instant $t = 0$ (could be elsewhere), two operations can be done to translate the previous assumption. The first one (subtractive) is to crop the longest sound to match the size of the shortest. The second one (additive) is to pad the shortest sound to match the longest. If the padding is done using numerical values of zero, the latter is known as the zero-padding technique.

- The second assumption is to consider that sounds can be compared outside their time frame. In this case, the statement is that the extent of the comparison goes beyond the duration of sounds; unless it is the object of comparison. This can be translated by levelling down the durations so that they are not considered into further calculations. Concretely, that is to resample the sounds (audio features) to an equivalent duration (number of frames). As in the first case, the longest can be down sampled to the length of the shortest or, the shortest can be over sampled to the length of the longest.
- The third assumption is to consider that sounds can be compared dynamically. Here, the statement is that the extent of the comparison is somehow adaptive. The latter assumption refers to dynamic time warping [Salvador, 2004]. In time series analysis, dynamic time warping (DTW) is an algorithm for measuring similarity between two temporal sequences which may vary in speed. In general, DTW is a method that calculates an optimal match between two given sequences (time series) with certain restrictions. In this case, the DTW could be used, not to measure the similarity between a pair of sounds, but simply to align them on the time axis.

Although the latter assumption is particularly attractive, experiments conducted in the frame of this work have shown that this approach is not advisable for this specific task (temporal alignment). That is because the DTW renders the optimal match between two given time series. In other words, it transforms the audio features much (given the radius constraint) that the similarity between them increase drastically. Besides, it could not even be used to measure the levels of similarity themselves because of the specific clustering method elaborated later (see Variation on HCA). Not being a nearest neighbour classifier[10], other constraints had to be taken into account (see Features space analysis). Since warping audio features cannot be considered here, it is clear that the first assumption above cannot be neither. Consequently, the second assumption appears to be the most plausible way for comparing a pair of sounds. In the frame of this work, the audio features were thus resampled in order to compare them outside the time dimension. The best way to do so, with minimal impact on the original features, appeared to be over sampling the shortest to the length of the longest. The other way around may lead to a serious loss of information as demonstrated in the following figures.

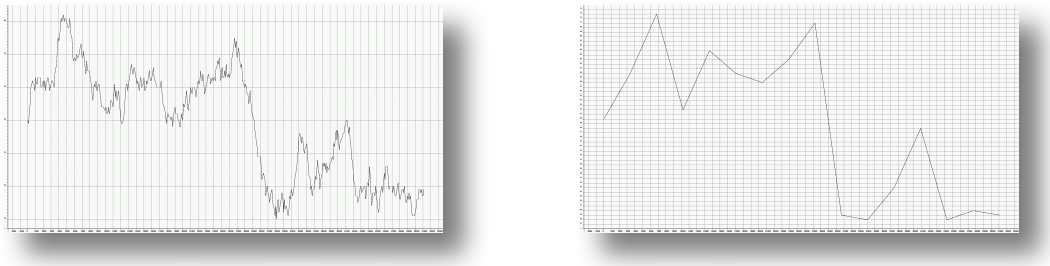

Figure 10. [left] Original feature with 512 frames, [right] Down sampled feature with 16 frames

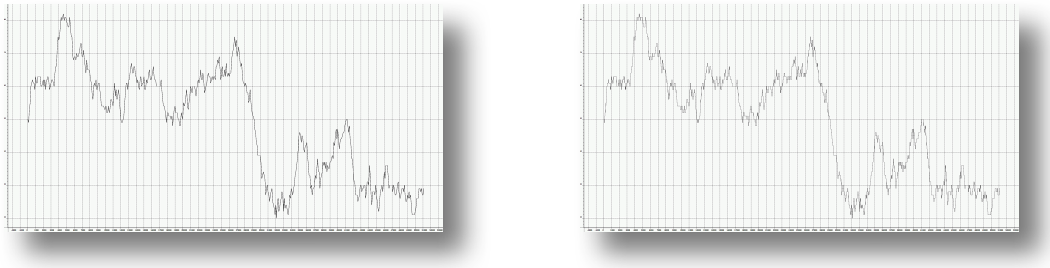

Figure 11. [left] Original feature with 512 frames, [right] Over sampled feature with 16384 frames

[10] In pattern recognition, the *k*-Nearest Neighbours algorithm (*k*-NN) is a non-parametric method used for classification and regression. In both cases, the input consists of the *k* closest training examples in the feature space. In *k*-NN classification, the output is a class membership. An object is classified by a majority vote of its neighbours, with the object being assigned to the class most common among its *k* nearest neighbours (*k* is a positive integer, typically small). If $k = 1$, then the object is simply assigned to the class of that single nearest neighbour.

7. FEATURES SPACE ANALYSIS

The fourth step towards agglomerative clustering is to determine the level of similarity (or dissimilarity) between each component of the database. As discussed later in this section (see Dimensionality reduction), this can be done using one or more of their features. Considering that only instantaneous features (no global temporal features) are taken into account at this point, three different approaches were adopted to process such data structures (time series/vectors). The first approach is based on distances (magnitudes), the second one is based on similarities (orientations) and the third one is based on correlations (dependencies). As demonstrated further, the previous approaches may also be merged in order to account for higher level descriptions of sounds into the clustering process (see Distance triangulation).

a. Distance (magnitude)

In this work, three types of distances have been used, namely the Minkowski distance, the Mahalanobis distance and the Jaccard distance. The Minkowski and the Jaccard have not been used to compare the features themselves but for other tasks such as triangulating the overall level of similarity between them and to measure the distance between clusters of sounds (see Distance triangulation and Inter-cluster analysis). Hence, the Mahalanobis distance is the only type of distance that was calculated between the features themselves. As discussed below, the Mahalanobis distance appeared to be the most adequate measure of magnitude for comparing features with different units and different scales (amplitudes, frequencies, modulation rates, etc.).

Minkowski distance [0. +inf.]:
The Minkowski distance is a metric in a normed vector space which can be considered as a generalization of both Euclidean distance and the Manhattan distance. The Minkowski distance of order p, in reference to L^P spaces[11], between two points is defined as follow [Verley, 1997].

Formulation (1):

$$d(x,y) = \left(\sum_{i=1}^{n} |x_i - y_i|^p\right)^{1/p}$$

For $p \geq 1$, the Minkowski distance is a metric as a result of the Minkowski inequality. When $p < 1$, the distance between $(0, 0)$ and $(1, 1)$ is $2^{1/p} > 2$, but the point $(0, 1)$ is at a distance of 1 from both of these points. Since this violates the triangle inequality[12], the Minkowski distance is not a metric when $p < 1$. Thus the Minkowski distance is typically used with $p \geq 1$, and more specifically with $p = 1$ or $p = 2$. The latter is the Euclidean distance while the former is known as the Manhattan distance. In mathematics, the Euclidean distance is the straight-line distance between two points in a Euclidean space. In other words, the Euclidean distance between two points, *d(x, y)*, is the length (magnitude) of the line segment connecting them. The Manhattan distance, also known as the city block, the taxicab, the rectilinear or even the snake distance, refers to a metric in which the distance between two points is the sum of the absolute differences of their Cartesian coordinates. The different names allude to the grid layout of the streets on the island of Manhattan which causes the shortest path a car could take between two intersections to have a length equal to the intersections distance.

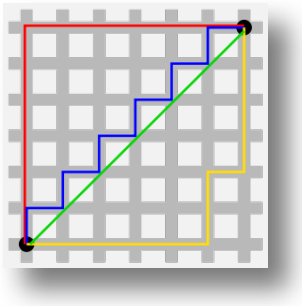

Figure 12. [straight line] Euclidean distance, [other lines] Manhattan distances

[11] In mathematics, the L^P spaces (also known as the Lebesgue spaces, named after Henri Lebesgue) are function spaces defined using a natural generalization of the *p*-norm for finite-dimensional vector spaces.
[12] In mathematics, the triangle inequality states that for any triangle, the sum of the lengths of any two sides must be greater than or equal to the length of the remaining side ($z \leq x + y$).

Looking at the previous formulation, it is understandable why this type of distance was not used to compare the features together. That is because the Minkowski distance (for $p \geq 1$) is expressed in the same units as the compared elements. Resulting in dimensional quantities (scale dependent), these values are very unpractical for multidimensional space analysis (our case). Nonetheless, the Minkowski distance becomes very useful to compute some kind of triangulation between multiple variables (see Distance triangulation). The definition of the p order is discussed later (see Minkowski distance order).

Mahalanobis distance [0. +inf.]:
The Mahalanobis distance is a measure of the distance between a point P and a distribution D. It is a multidimensional generalization of the idea of measuring how many standard deviations[13] (σ) away P is from the mean (μ) D. This distance is zero if P is at the μ of D, and grows as P moves away from the μ. This type of distance is thus unitless, scale-invariant and takes into account the correlations of the data set [Hazewinkel, 2002].

Formulation (2):

$$d(\vec{x}, \vec{y}) = \sqrt{(\vec{x} - \vec{y})^T S^{-1} (\vec{x} - \vec{y})}$$

Introducing the concept of correlation, this formulation does not give information on the dependencies between variables (see Correlation coefficients) but rather on the probabilities of a vector $\vec{x}$ belonging to another vector $\vec{y}$ distribution. In other words, the closer $\vec{x}$ is to the μ of $\vec{y}$ (in terms of σ), greater the chances are that they belong to the same cluster. In this sense, the Mahalanobis distance belongs to statistical interpretation, and not to geometrical interpretation as in the case of the Minkowski distance. For that, the Mahalanobis distance should be subjected to the rules of the bell-shaped normal distribution[14]. Following this assumption, it is possible to refer to the central limit theorem[15] and say that, for all distributions for which the σ is defined, the amount of data within a number of σ from the μ can be defined by the empirical rule (also known as the *68-95-99,7* rule).

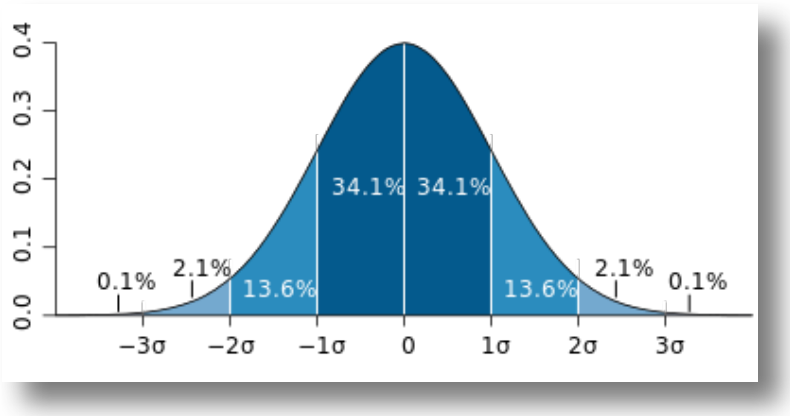

Figure 13. Graphic representation of the empirical rule

The previous graph (figure 13) shows that, if a data distribution is approximately normal then about 68% of the data values are within 1σ of the μ, about 95% are within 2σ, and about 99,7% lie within 3σ. Following this principle, the range of the Mahalanobis distance can be approximated between [-4σ 4σ], where lower and higher values are simply considered as outliers. That being said, it has to be recalled that a distance cannot be negative by definition. Thus the Mahalanobis distance range is usually expressed in absolute values. The negative part being only an indication of position (below or above μ), the range can be approximated between [0σ 4σ]. Then, the final assumption becomes: if $d(\vec{x}, \vec{y}) > 4\sigma$, the chances of $\vec{x}$ belonging to $\vec{y}$ cluster are very low (4σ = roughly 0.006%).

[13] In statistics, the standard deviation is a measure that is used to quantify the amount of variation or dispersion of a set of data values. A low standard deviation indicates that the data points tend to be close to the arithmetic mean of the set, while a high standard deviation indicates that the data points are spread out over a wider range of values.
[14] In probability theory, the normal distribution is a very common continuous probability distribution. Normal distributions are important in statistics and are often used in the natural and the social sciences to represent real-valued random variables with unknown distributions.
[15] In probability theory, the central limit theorem (CLT) states that, given certain conditions, the arithmetic mean of a sufficiently large number of iterates of independent random variables, each with a well-defined expected value and finite variance, will be approximately normally distributed, regardless of the underlying distribution.

Considering the classification task to achieve later, this assumption allows to normalize the Mahalanobis distance (magnitude) for it to be compared with other types of measurements (orientation and dependencies).

Jaccard index [0. 1.]:
The Jaccard index, also known as the Jaccard similarity coefficient (coined after the Swiss botanist Paul Jaccard), is a statistic used for comparing the similarity and diversity of sample sets. The Jaccard coefficient is measured between finite sample sets, and is defined as the size of the intersection[16] divided by the size of the union[17] of the sample sets [Jaccard, 1901].

Formulation (3):

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$$

If A and B are both empty, $J(A,B) = 1$. Thus, $0 \leq J(A,B) \leq 1$.

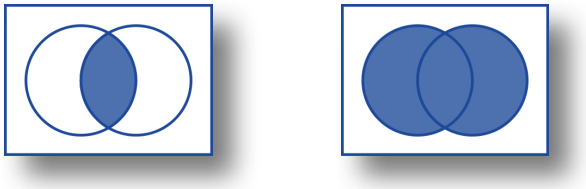

Figure 14. [left] Intersection of two sets, [right] Union of two sets

The Jaccard distance, which measures dissimilarity between sample sets, is complementary to the Jaccard index and is obtained by subtracting the latter from 1, or equivalently, by dividing the difference of the sizes of the union and the intersection of two sets by the size of the union.

Formulation (4):

$$J(A,B) = 1 - J(A,B) = \frac{|A \cup B| - |A \cap B|}{|A \cup B|}$$

As mentioned above, the Jaccard distance is not used to compare the audio features but rather, combined with the Minkowski distance, to measure the distance between clusters of sounds at the very end of the clustering process (see Inter-cluster analysis).

b. Similarity (orientation)

In statistics and related fields, a similarity measure quantifies the similarity between two objects. Although no single definition of a similarity measure exists, usually such measures are in some sense the inverse of distance metrics. They take on large values for similar objects and either zero or a negative value for very dissimilar objects. The cosine similarity is a common similarity measure used in information retrieval, among other fields, to score similarity of various elements in the vector space model [Singhal, 2001].

Cosine similarity [-1. 1.]:
The cosine similarity is a measure of similarity between two non-zero vectors of an inner product space that measures the cosine of the angle between them. The cosine of 0° is 1, and it is less than 1 for any other angle. It is thus a judgment of orientation and not of magnitude. For example, two vectors with the same orientation have a cosine similarity of 1, two vectors at 90° have a similarity of 0, and two vectors diametrically opposed (180°) have a similarity of -1, independent of their magnitude (translation-invariant). Cosine similarity is particularly used in positive space (our case), where the outcome is neatly bounded in [0. 1.].

[16] In mathematics, the intersection $A \cap B$ of two sets A and B is the set that contains all elements of A that also belong to B (or equivalently, all elements of B that also belong to A), but no other elements.
[17] In set theory, the union $A \cup B$ of a collection of sets is the set of all distinct elements in the collection. It is one of the fundamental operations through which sets can be combined and related to each other.

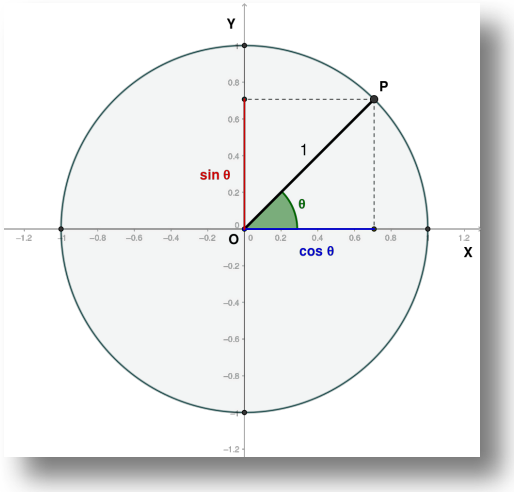

Figure 15. Cosine function of an angle constructed geometrically from a unit circle

Formulation (5):

$$cos(\theta) = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} y_i^2}}$$

The previous formulation 5 states that the resulting measure is unitless and scale invariant (similar to the Mahalanobis distance). For the same reasons mentioned before, the cosine similarity is thus another tool to privilege in a multidimensional context. That being said, it is important to denote some particularities of working in a positive space and its impact on the interpretation of this measure of similarity. Considering the type of data that are manipulated (instantaneous audio features), it can certainly be said that the cosine of 0° remains 1 (in the case of two identical sounds) and that the cosine of $\theta \geq 90^{\circ}$ is eliminated (in the case that time cannot be expressed as a unique moment, nor backward, and that all the features exist in a range of [0. +inf.]). Thus the notion of two vectors being diametrically opposed, in this particular case, refers to the orientation in time and not to the orientation in range.

For instance, if two vectors, expressed in the frequency (Hz) range, are compared using the cosine similarity (one vector being an upward straight line and the other being a symmetrical downward straight line), they cannot be defined as diametrically opposed, like it could be intuitively formulated, but rather be considered as roughly 58% similar. This can be easily worked around by transposing one of the two vectors in the vertical negative space but, in the next section, we will see that the correlation coefficients are more suited for this task (covariance analysis). Nonetheless, this particularity raises interesting questions about the definition of opposition in the field of sound perception. The cosine similarity thus implies that antitheses (diametrical opposition) do not exist in a positive space and that they might be more similar than one would expect.

c. Correlation (dependencies)

In statistics, dependence is any statistical relationship, whether causal or not, between two random variables or two sets of data. Correlation is any of a broad class of statistical relationships involving dependence, although in common usage it most often refers to a linear relationship. Familiar examples of dependent phenomena include the correlation between the physical statures of parents and their offspring, and the correlation between the demand for a product and its price. There are several correlation coefficients measuring the degree of correlation. The most common of these is the Pearson correlation coefficient, which is sensitive only to a linear relationship between two variables (which may exist even if one is a nonlinear function of the other). Other correlation coefficients have been developed to be more robust than the Pearson correlation, that is more sensitive to nonlinear relationships. A correlation coefficient is thus a number that quantifies some type of correlation and dependence (statistical relationships) between two or more random variables or observed values [Rakotomalala, 2015].

Common types of correlation coefficients include:

- The **Pearson product-moment correlation coefficient (*r*)** is a measure of the strength and direction of the linear relationship between two variables that is defined as the covariance of the variables divided by the product of their standard deviations.
- The **Spearman's rank correlation coefficient (*ρ*)** is a nonparametric measure of rank[18] correlation (statistical dependence between the ranking of two variables). It asses how well the relationship between two variables can be described using a monotonic function, whether linear or not.
- The **Kendall tau rank correlation coefficient (*τ*)** is a statistic used to measure the association between two measured quantities. It is also a nonparametric measure of rank correlation but unlike the Spearman correlation, this one is not affected by how far from each other ranks are but only by whether the ranks between observations are equal or not, and thus is appropriate for discrete variables but not for continuous variables.

The followings examples show different types of relationships (functions) that can be detected by the correlation coefficients mentioned above.

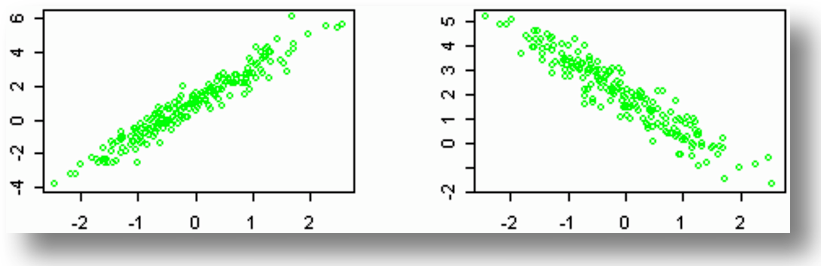

Figure 16. Monotonic linear relationships; [left] positive, [right] negative

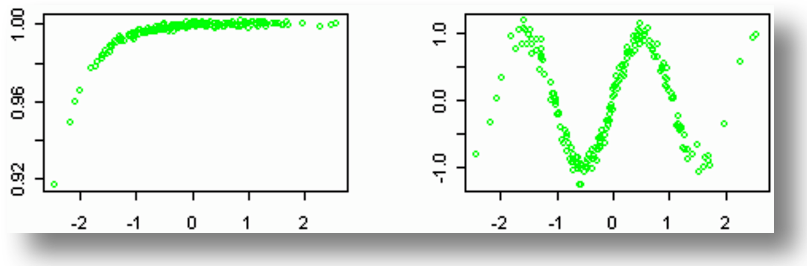

Figure 17. [left] Monotonic non-linear relationship, [right] Non-monotonic non-linear relationship

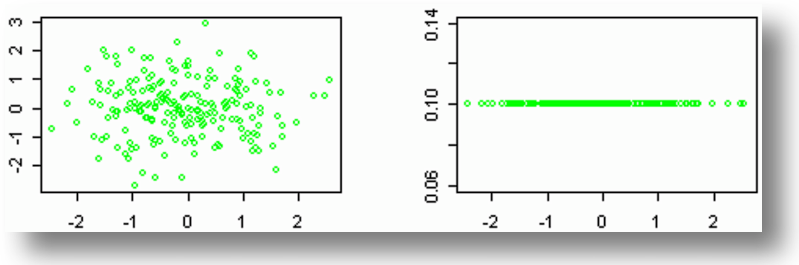

Figure 18. Absence of relationship

From the previous descriptions, it can be understood that Pearson's coefficient only detects the types of relationships shown in figure 16, that Kendall's coefficient is best suited to detect the types of relationships from figure 18, and that Spearman's coefficient can detect the types of relationships presented in figure 16 and figure

[18] The rank is the relative position label of the observations within the variable: 1st, 2nd, 3rd, etc.

17-left. Considering the components of the dataset (audio features), it is possible to conclude that the Spearman's rank correlation coefficient (ρ) is the most appropriate. The latter is also known to be extremely robust against outliers. That is because the correlation is calculated from the ranks and not from the original values. That implicitly smoothens the compared intervals.

Spearman's rank correlation coefficient [-1. 1.]:
The Spearman correlation between two variables is equal to the Pearson correlation between the rank values of those two variables. While Pearson's correlation assesses linear relationships, Spearman's correlation assesses monotonic relationships whether linear or not. A perfect Spearman correlation of +1 or −1 occurs when each of the variables is a perfect monotone function of the other. Intuitively, the Spearman correlation between two variables is high when observations have a similar (or identical for a correlation of 1) rank between the two variables, and low when observations have a dissimilar (or fully opposed for a correlation of -1) rank between the two variables. A value of zero denotes a total independence (zero relationship) between the rank of the two variables. Unlike the cosine similarity, this type of coefficient gives a clear indication about the direction of the function. As mentioned before, the coefficient tends to -1 if both variables are in opposite direction (see figure 16-right), and tends to 1 if the variables are in the same direction (see figure 16-left).

Since the Spearman correlation coefficient is defined as the Pearson correlation between the ranked variables, it can be formulated as follow [Rakotomalala, 2015].

Formulation (6):

$$\text{If Pearson's } r = \frac{cov(X,Y)}{\sigma_X \sigma_Y} \quad \text{then, Spearman's } \rho = \frac{\sum_{i=1}^{n}(R_i - \bar{R})(S_i - \bar{S})}{\sqrt{\sum_i (R_i - \bar{R})^2}\sqrt{\sum_i (S_i - \bar{S})^2}}$$

where

- $cov(X, Y)$ is the covariance of the rank variables,
- σ_X and σ_Y are the standard deviations of the rank variables,
- R and S are the rank variables,
- $\bar{R}$ and $\bar{S}$ are the arithmetic means of the rank variables.

If ties are present in the variables, the previous equation yields incorrect results. Identical values should be assigned fractional ranks equal to the average of their positions in the ascending order of the values.

Individuals	Variable	Original ranks	Averaged ranks
A	0	1	1.5
B	0	2	1.5
C	1	3	3
D	2	4	5
E	2	5	5
F	2	6	5
G	5	7	7
H	6	8	8
I	7	9	9
J	8	10	10.5
K	8	11	10.5
L	12	12	12

Table 2. Averaged ranks calculation

Then, the following correction factor can be calculated [Rakotomalala, 2015].

Formulation (7):

$$T_R = \sum_{g=1}^{G}\left(t_g^3 - t_g\right)$$

where

- T_R is the tie-correction factor,
- G represents the distinct values of the averaged ranks,
- t_g is the number of occurrence of a distinct rank.

	G	**8**
Averaged ranks values	t_g	$(t_g^3 - t_g)$
1.5	2	6
3	1	0
5	3	24
7	1	0
8	1	0
9	1	0
10.5	2	6
12	1	0
	Total T_R:	36

Table 3. Tie-correction factor calculation

Once calculated for both variables, the T_R and the T_S should be included in the initial formulation 6 of Spearman's correlation coefficient in order to yield correct results. The following is known as the tie-corrected Spearman's coefficient formula [Rakotomalala, 2015].

Formulation (8):

$$\text{tie corrected } \rho = \frac{(n^3 - n) - 6\sum_{i=1}^{n} d_i^2 - (T_R + T_S)/2}{\sqrt{(n^3 - n)^2 - (T_R + T_S)(n^3 - n) + T_R T_S}}$$

where

- n is the length of the population (total number of individuals for each variable)[19],
- d is the difference between R_i and S_i,
- T_R and T_S are the tie-correction factors.

Now it is clear that the Spearman correlation coefficient, like the Mahalanobis distance and the Cosine similarity, is unitless and scale-invariant. Thus it should also be privileged when working in a multidimensional space like ours. Contrary to the Cosine similarity, the correlation coefficients imply the existence of antitheses as they were described before. Observations that are contradictory tend to -1, the latter describing a perfect opposition (upward straight line compared to a symmetrical downward straight line). This question is further discussed in the next section (see Distance triangulation).

d. Distance triangulation

After calculating the three similarity levels presented before (distance, similarity and correlation), the following idea is to merge them into a single multidimensional score. Since each type describes a different aspect of similarity (magnitude, orientation and dependency), the goal is to obtain a single value that includes all three perspectives. Obviously, the classification process could also be done using one or two types of measurements only. Based on the concept of triangulation[20], the principle is to project the previous measurements in a three-dimensional space where each axis represents a different one of them. This allows to triangulate the resulting location for later calculating the Minkowski distance (the p order being yet to be determined) between this new coordinate (x, y, z) and the origin $(0, 0, 0)$. For that, the measurements must be adapted to the new space for the origin to be the right target (minimum distance).

[19] This clearly implies that both variables must have the same number of components (see Temporal alignment). That is also true for the Minkowski distance and the Cosine similarity but not for the Mahalanobis distance because the pooling is proportional to each variable length.
[20] In trigonometry, triangulation is the process of determining the location of a point by forming triangles to it from known points.

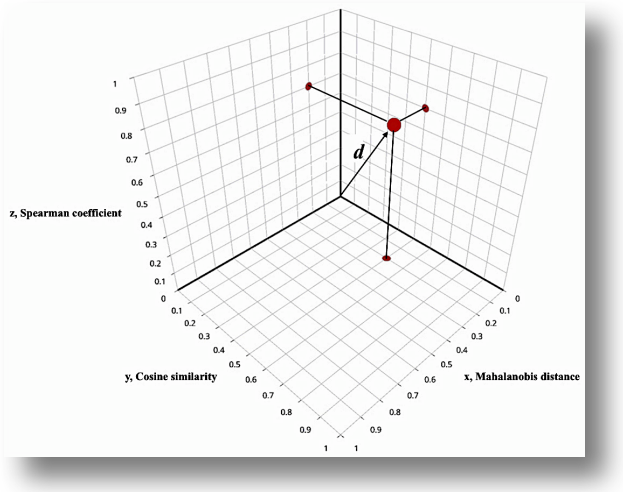

Figure 19. 3D distance triangulation where *d* is the Euclidean distance

Knowing that the Mahalanobis distance range can be approximated between [0. 4.], that the Cosine similarity between audio features is included between [0. 1.], and that the Spearman coefficient is bounded by [-1. 1.], it becomes obvious that they can be normalized between [0. 1.] and rescaled to [1. 0.] for fitting the previous space and for its origin to represent the minimum distance. The reason to rescale the previous data is to convert the degrees of similarity and correlation into distance interpretable values. The reason to normalize them is related to the classification method used in the frame of this work (see Variation on HCA). However, it should be mentioned that in the case of the Mahalanobis distance, the values do not need to be rescaled since it is already a distance measure. It should also be mentioned that in the case of the Spearman coefficient, the question of antitheses raises again when normalizing its values as suggested before. Considering that a coefficient of -1 denotes a perfect negative relationship and that a coefficient of 0 denotes the complete absence of relationship, four options remain.

- The first option is to simply consider a coefficient of -1 as the farther. Then the normalized values would become (in terms of distance): -1; 1, 0; 0.5 and 1; 0. The underlying assumption is that the antitheses are considered the farthest elements in the space while two others, denoting a complete absence of relationship ($\rho = 0$), are considered closer to each other.
- The second option is to clip the values below zero. In this case, the normalized values would become (also in terms of distance): -1; 1, 0; 1 and 1; 0. This one makes a similar assumption as in the first case but here, the antitheses and the pairs with $\rho = 0$ are considered the farthest elements in space.
- The third option is to rescale the negative part [-1. 0.] somewhere between [0. 1.]. If we take a range of [0.5 1.] as target, the values would become (also in terms of distance): -1; 0.5 and 0; 0. This last assumption recalls the previous discussion about the cosine similarity where the antitheses are considered similar at roughly 58%. In this case, it would depend on the target scale.
- The fourth option is to turn the coefficients into absolute values. In this case, the normalized values would become (always in terms of distance): -1; 0, 0; 1 and 1; 0. This one makes the assumption that the antitheses are considered as similar as two identical elements.

Since the objective is to take into account three types of measurements, or three different perspectives, the last option appears to be the most logical. From a correlation point of view, the antitheses are then considered as similar as two identical elements while from a cosine similarity point of view, they are considered as roughly 58% similar. In the case of the Mahalanobis distance, the results depend on their respective registers. If the registers are identical, the distance between antitheses is zero. If not, the resulting distance may vary a lot. Nonetheless, this particular question should be further investigated from a perceptual angle.

e. Minkowski distance order

As seen previously in figure 19, the most intuitive approach to calculate the distance from the origin and multiple variables is to use the Euclidean distance (Minkowski distance with $p = 2$). Nonetheless, this approach implies that the variables are projected in a Euclidean space. Consequently, the latter are subjected to its set of rules. In this

sense, it is important to consider other types of space (L^p spaces) where the variables could be better represented. In other words, various p order should be experimented with the Minkowski distance.

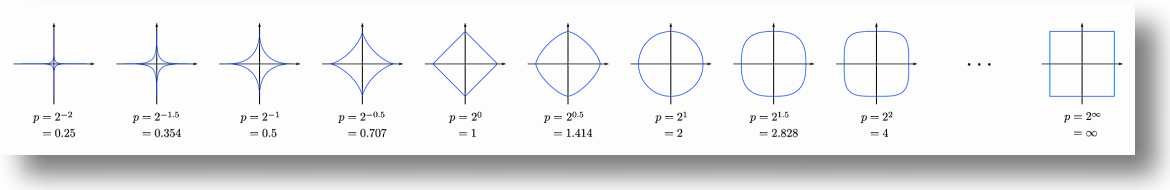

Figure 20. Unit circles with various p order

If the following task is to target the k nearest neighbour (k-NN), the type of space has no real impact on the results, as long as one remains consistent through repeating the operation. However, if the target has to fall within a specific part of the space or below a given threshold to be considered (our case), the definition of space and the number of dimensions have a significant impact on the results, thus on setting the threshold itself (see Distance threshold).

Two dimensions (2D)														
x, y	***p***	**Δ**		***x, y***	***p***	**Δ**		***x, y***	***p***	**Δ**		***x, y***	***p***	**Δ**
0.5	2^0	1.		0.5	$2^{0.5}$	0.816		0.5	2^1	0.707		0.5	2^2	0.595
0.5				0.5				0.5				0.5		

Three dimensions (3D)														
x, y, z	***p***	**Δ**		***x, y, z***	***p***	**Δ**		***x, y, z***	***p***	**Δ**		***x, y, z***	***p***	**Δ**
0.5	2^0	1.5		0.5	$2^{0.5}$	1.087		0.5	2^1	0.866		0.5	2^2	0.658
0.5				0.5				0.5				0.5		
0.5				0.5				0.5				0.5		

Table 4. Distance triangulation with various p order

In this sense, the shape of a cluster is linearly correlated to the shape of the space. In the following case (figure 21), it is clear that the object at ($x = 0.3$, $y = 0.3$) is more or less distant from the origin depending on the shape of the space. In the following square space, it is more than 0.5 (cluster radius) but in the circular space, it is less than 0.5. The former can be verified using the Manhattan distance and the latter using the Euclidean distance.

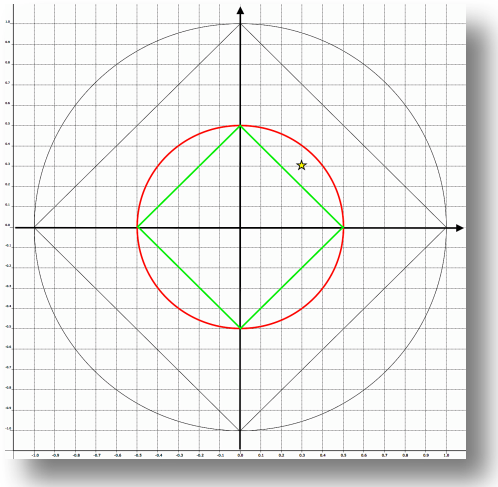

Figure 21. 2D representation of a square and a circle shaped cluster

Since no decisive elements have been found to describe the best space for projecting the variables at this point, the Euclidean space (circular space from figure 21) appears to be, based on intuition, the most adapted to the following clustering method. Nonetheless, this question is another one that should be further investigated from a perceptual angle.

f. Distance matrices

In mathematics, computer science and especially graph theory, a distance matrix is a table (two-dimensional array) containing the distances, taken pairwise, between the elements of a set. If there are *N* elements, this matrix will have a size of *N*x*N*. In graph-theory the elements are more often referred to as points, nodes or vertices (see Single-layer relational space). Depending on the application involved, the distance being used to define this matrix may or may not be a metric. When distance is defined as a metric (our case), the distance matrix satisfies properties directly related to the defining properties of a metric [Gentle, 2007].

That is,

- Non-negativity: $d(x, y) \geq 0$
- Identity of indecernibles: $d(x,y) = 0$ when $x = y$
- Symmetry: $d(x, y) = d(y, x)$
- Triangle inequality: $d(x, z) \leq d(x, y) + d(y, z)$

Then (for a distance matrix),

- The entries on the main diagonal are all zero,
- All the off-diagonal entries are positive,
- The matrix is a symmetric matrix,
- For any x and y, $d(\mathrm{x}, \mathrm{z}) \leq d(x, y) + d(y, z)$ = triangle inequality

At this point, it is important to understand that the level of similarity between each pair of elements in the database has been determined by the previous operations (see Features space analysis). Then, the resulting values are gathered inside distance matrices, each of those representing a different audio feature. Meaning that the level of similarity is computed between corresponding audio features only: $d(x_i, y_i)$ and not $d(x_i, y_j)$ nor $d(x_i, y_j)$. This way, each matrix includes the distance between each pair of sounds under a specific property of sounds. The resulting number of matrices is thus linked to the number of extracted features. Nonetheless, the final classification may account for multiple features at once (see Dimensionality reduction).

Audio feature				
Sounds	a	b	c	d
a	0	0.67	0.4	0.63
b	0.67	0	0.32	0.21
c	0.4	0.32	0	0.22
d	0.63	0.21	0.22	0

Table 5. Single distance matrix (4x4)

g. Weighted scores

After computing multiple distance matrices (one for each selected feature), the following step is about applying different weights (w_i) to them. In other words, that is to assign independent degrees of importance (w_i) to the different features that are considered for the classification. That is usually done by multiplying each scores within a single matrix by a weight factor ranging from 0 to 1 [Gentle, 2007].

Spectral feature					w_i	Weighted spectral feature				
Sounds	a	b	c	d		***Sounds***	a	b	c	d
a	0					a	0			
b	0.67	0			1.	b	0.67	0		
c	0.4	0.32	0			c	0.4	0.32	0	
d	0.63	0.21	0.22	0		d	0.63	0.21	0.22	0

Harmonic feature					w_i	Weighted harmonic feature				
Sounds	a	b	c	d		***Sounds***	a	b	c	d
a	0					a	0			
b	0.43	0			0.75	b	0.323	0		
c	0.99	0.9	0			c	0.743	0.675	0	
d	0.09	0.78	0.62	0		d	0.068	0.585	0.465	0

Energy feature					w_i	Weighted energy feature				
Sounds	a	b	c	d		***Sounds***	a	b	c	d
a	0					a	0			
b	0.4	0			0.5	b	0.2	0		
c	0.45	0.51	0			c	0.225	0.255	0	
d	0.5	0.07	0.5	0		d	0.25	0.035	0.25	0

Table 6. Weighted distance matrices

From the previous weighting approach, it is clear that more importance is given to the spectral feature (w_i =1), that a little less is given to the harmonic feature (w_i =0.75) and even less to the energy feature (w_i =0.5). The underlying assumption is that the spectral feature is perceptually more significant than the harmonic feature, even more than the energy feature and that the harmonic feature is also more significant than the energy feature for discriminating sounds. Since this aspect remains at an experimental level in the frame of this work, this question should also be investigated further from a perceptual angle.

h. Dimensionality reduction

After computing different weights on the distance matrices, the next step is to reduce the dimensionality of the resulting space (*N*-features = *N*-matrices = *N*-dimensions) in order to simplify the imminent clustering process. In machine learning and statistics, dimensionality reduction is the process of reducing the number of variables under consideration through obtaining a set of principal components [Roweis, 2000]. It may be used for feature selection and feature extraction [Pudil, 1998]. One of the most common method to do so is known as the principal component analysis (PCA). Although the latter technique could be used at this point, or earlier in the process to reduce the number of audio features, it was not considered to be necessary considering that the experimentations conducted in the frame of this work required to use a relatively low number of features at once (1 to 20). Hence, there was no serious reasons to fear the curse of dimensionality[21] [Bellman, 1957]. Besides, it also appeared to be more consistent to keep control over the processed information for interpreting the clustering results more easily.

In this sense, the following solution is not to reduce the number of dimensions themselves but rather to project them in a common space. The idea is the same as described before in the case of merging the three levels of similarity (magnitude, orientation and dependencies) into a single multidimensional score (see Distance triangulation). Although the resulting space may be expressed in more or less than three dimensions here, the approach remains the same: calculating the Minkowski distance between the new coordinate (x, y, z, …, *n*) and the origin (0, 0, 0, …, 0). In other words, the multiple dimensions are triangulated and summarized by their distance to the origin (minimum distance). Consequently, this approach allows to bring down the number of distance matrices to a single one without applying any transformation to the data themselves. From another perspective, it allows to account for higher level descriptions of sounds into the clustering process.

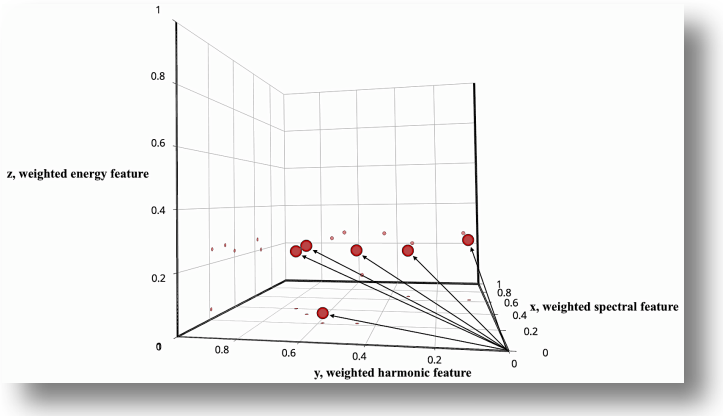

Figure 22. 3D Euclidean feature space representation of the data from table 6

[21] The curse of dimensionality refers to various phenomena that arise when analyzing and organizing data in high-dimensional (often with hundreds or thousands of dimensions) that do not occur in low-dimensional settings such as the three-dimensional physical space of everyday experience. The common theme of these problems is that when the dimensionality increases, the volume of the space increases so fast that the available data become sparse. This sparsity is problematic for any method that requires statistical significance.

From the previous space representation (figure 22), it becomes easy to deduce the underlying distance matrix. Using the Minkowski distance (formulation 1) where: x = features, y = origin and p = 2 (Euclidean space), the multiple dimensions can be summarized within a single distance matrix such as the following.

Feature space				
Sounds	a	b	c	d
a	0			
b	0.77	0		
c	0.873	0.789	0	
d	0.681	0.623	0.572	0

Table 7. Distance matrix representation of the feature-space from figure 22

Obviously, this raises again the question of defining the space into which the variables are projected. Although the situation is identical as described in the first case (see Distance triangulation), the sequencing adds an element of complexity to the problem. In other words, two types of space have to be defined (if so) at this point: one to project the similarity levels and another one to project the features. In the frame of this work, both cases were intuitively projected in the same type of space. Thus the Minkowski distance is calculated using the same p order for both instances in the sequence (distance triangulation and dimensionality reduction). Nonetheless, this question should remain open to further investigations.

8. SINGLE-LAYER RELATIONAL SPACE

At this point, although this is not an intrinsic part of the classification process, it is interesting to mention that the database can be viewed as some kind of relational space[22]. With the help of Gephi [Bastian, 2009], an open source software for exploring and manipulating networks, any feature-space distance matrix can be visualized as such.

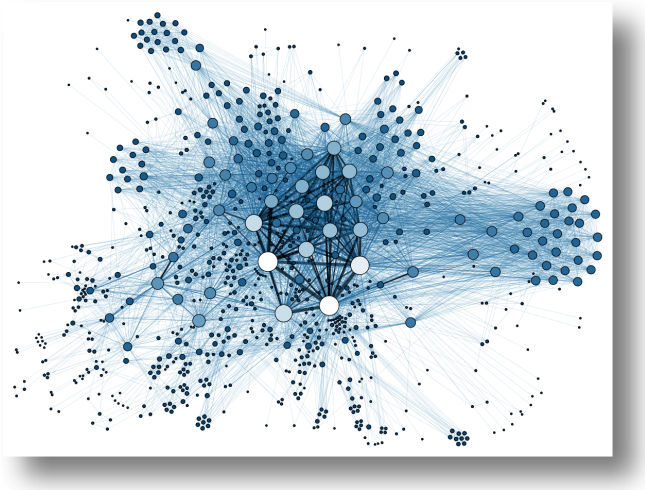

Figure 23. Single-layer features space network plane designed with Gephi

In the above figure 23, the nodes represent the sounds and the edges represent the distances between them. From this perspective, it is clear that the matter of this work is not about positioning the sounds in space but rather to define the strength of their respective networks. That is essential for further cluster analysis.

9. AGGLOMERATIVE CLUSTERING

The fifth step is where the classification happens. With respect to the objectives of this work, the following section discusses the use of a specific clustering method, namely the hierarchical cluster analysis (HCA), through its strengths and weaknesses. It also covers how the latter inspired the implementation of a variation based on a

[22] The relational theory of space is a metaphysical theory according to which space is composed of relations between objects, with the implication that it cannot exist in the absence of matter. Its opposite is the container theory. A relativistic physical theory implies a relational metaphysics, but not the other way around: even if space is composed of nothing but relations between observers and events, it would be conceptually possible for all observers to agree on their measurements, whereas relativity implies they will disagree.

threshold constraint, the specificities of the outcome and its impact on further processing towards a multi-layer relational space.

a. Hierarchical cluster analysis (HCA)

As mentioned before (see Definitions), the HCA is an unsupervised learning method that seeks to build a hierarchy of clusters using either an agglomerative (bottom-up) or a divisive (top-down) strategy. Only the former has been used in the frame of this work. In general, the merges (our case) or the splits are determined in a greedy[23] fashion. In order to decide which clusters should be combined or where a cluster should be split, a measure of similarity (or dissimilarity) between sets of observations is required. In most methods of hierarchical clustering, this is achieved by using an appropriate metric and a linkage criterion. The linkage criterion determines the distance between two clusters as a function of the pairwise distances between observations [Hastie, 2009]. As discussed earlier (see Minkowski distance order), the choice of a metric implies a particular shape of space thus a particular shape of cluster, as some elements may be close to one another according to one distance and farther away according to another.

Some commonly used linkage criteria between two sets of observations are:

- Single-linkage: is defined as the distance between the closest two elements from each cluster,
- Complete-linkage: is defined as the distance between the farthest two elements from each cluster,
- Average linkage: is defined as the mean (average) distance between all elements from each cluster.

Other linkage criteria include:

- The sum of all intra-cluster variance,
- The decrease in variance for the cluster being merged (Ward's criterion),
- The probability that candidate clusters spawn from the same distribution function (V-linkage),
- The increment of some cluster features after merging two clusters.

Usually based on the *k*-NN principle, hierarchical clustering has the distinct advantage that any valid measure of distance can be used. In fact, the observations themselves are not required. All that is needed is a distance matrix as described before (see Distance matrices).

For example, suppose a dataset is to be clustered using the Euclidean distance and the complete-linkage criterion. The latter is obviously the most adapted linkage technique so far, considering the objective is to form clusters with the closest possible elements (distance wise). If there are six elements, {a}{b}{c}{d}{e}{f}, the first step is to determine which elements to merge in a cluster. With respect to the linkage criterion, the two closest elements are usually chosen. If the closest are {b} and {c}, the remaining clusters are: {a}{b, c}{d}{e}{f}. After merging clusters, the distance matrix needs to be updated in order to proceed to the next step. That is done by merging the corresponding rows and columns. The following step is identical to the preceding one and the operation is repeated until the clusters are too far apart to be merged (distance criterion) or until there is a sufficiently small number of clusters (number criterion). In the following figure, the rows represent each step (iteration) of the process.

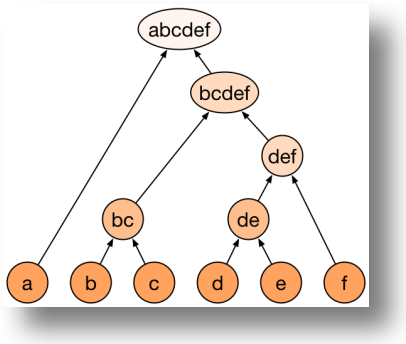

Figure 24. Agglomerative clustering (bottom-up)

[23] A greedy algorithm is an algorithmic paradigm that follows the problem solving heuristic of making the locally optimal choice at each stage with the hope of finding a global optimum. In many problems, a greedy strategy does not produce an optimal solution, but nonetheless a greedy heuristic may yield locally optimal solutions that approximate a global optimal solution in a reasonable time. This is achieved by trading optimality, completeness, accuracy, or precision for speed. In a way, it can be considered a shortcut.

Looking at the first iteration of the process illustrated in figure 24, it is clear that the distance between {b} and {c} is equal to the distance between {d} and {e} but, what if the distance between {e} and {f} is equal to the distance between {d} and {e} and not to the distance between {d} and {f}? The distance matrix from figure 25 clearly demonstrates this case.

	a	b	c	d	e	f
a	0.000	0.700	0.600	0.500	0.400	0.300
b	0.700	0.000	0.100	0.200	0.300	0.400
c	0.600	0.100	0.000	0.100	0.200	0.300
d	0.500	0.200	0.100	0.000	0.100	0.200
e	0.400	0.300	0.200	0.100	0.000	0.100
f	0.300	0.400	0.300	0.200	0.100	0.000

Figure 25. Distance matrix corresponding to figure 24

Intuitively, {e} and {f} should also form a cluster, as in the case of {b, c} and {d, e}, but the greedy aspect of this algorithm leads to local optimum (tie breaks) in chronological order. For example, if it is chronologically determined that {e} belongs to {d} and later determined that {f} belongs to {e} but not to {d}, then {f} cannot belong to {d, e}. However, {f} may be merged with {d, e} through the next iteration of the process as seen in figure 24. In the next figure 26, the situation is similar if the elements (rows or columns from figure 25) are simply permutated from: {a}{b}{c}{d}{e}{f} to {a}{b}{c}{e}{f}{d}. This clearly demonstrates the chronological tie break method inherent to a classic HCA [Defays, 1977]. Although the tie break approach may make use of weights or random processes instead of sequencing, it will not be discussed here (see Variation on HCA). The following dendrograms were generated with Orange, a data mining toolbox in Python [Orange, 2013].

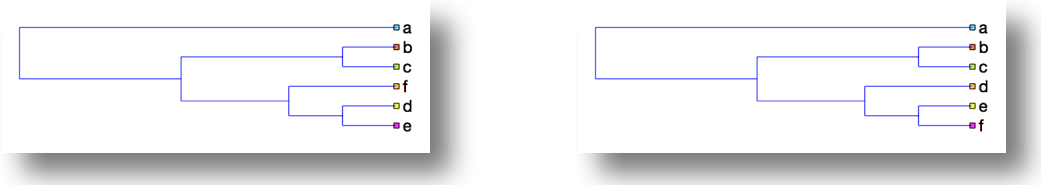

Figure 26. Dendrograms; [left] original sequence clustering, [right] permutated sequence clustering

The fact that {f} is later merged with {d, e} (figure 26-left) reveals another drawback of this method. That is, the nearest neighbours are always further away (unevenly) from each other, whatever the linkage method, going forth in the process of iteration. Meaning that more iterations equals weaker similarity within and between clusters. As mentioned before, this may be regulated (to some extent) using an appropriate distance criterion (stopping constraint), but the tie break problem still leads to an inevitable loss of information in the classification process. Thus, for perceptual reasons, a more complete and accurate classification method should be preferred at this stage (see Variation on HCA). Nonetheless, the classic HCA still appears to be a good solution in this particular case of audio classification in comparison to other unsupervised learning methods such as the k-means[24] or the mixture models[25]. In this sense, a classic HCA could be used for later processing in the frame of this work (see Graph theory).

Although the other techniques of linkage (variance, probability and feature oriented) could provide a very interesting and different insight on the dataset, they were not used nor investigated in the frame of this work. That is because the clustering should be exclusively distance oriented at this point, the objective being to form clusters with the closest possible elements. Besides, the notions of variance, probability and feature are already included in the calculation of the distance between the sounds (see Distance triangulation).

[24] A *k*-means clustering is a method of vector quantization, originally from signal processing, that is popular for cluster analysis in data mining. *k*-means clustering aims to partition *n* observations into *k* clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster.

[25] A mixture model is a probabilistic model for representing the presence of subpopulations within an overall population, without requiring that an observed data set should identify the sub-population to which an individual observation belongs. Formally a mixture model corresponds to the mixture distribution that represents the probability distribution of observations in the overall population.

b. Variation on HCA

As mentioned before, the HCA inspired a variation based on a threshold constraint. In this case, contrary to the greedy algorithms, the speed is traded for completeness and accuracy of the classification. For that, the agglomeration strategy has to change. Instead of targeting the nearest neighbours (open space), a user defined distance threshold (closed space) determines the targets. The threshold represents the maximum distance between two elements to be merged. It can be seen as some kind of perceptual threshold. In this sense, the targets should be as many as they fall below the maximum distance allowed. For that to be assessed, the number of iterations has to be as much as the square number of elements (*N*x*N* distance matrix). In other words, each pair of elements has to be tested before defining a cluster. That means the algorithm is looking for the global optimum, including the possibility of overlapping clusters.

Although the agglomeration strategy is quite different here, the previous linkage criteria remain effective. In this case, the complete-linkage is still the preferred method in order to avoid inconsistency within and between clusters. The following figure 27, where the circles represent the distance threshold (diametrically) as well as each of them an iteration of the process, partially demonstrates the agglomeration strategy described above.

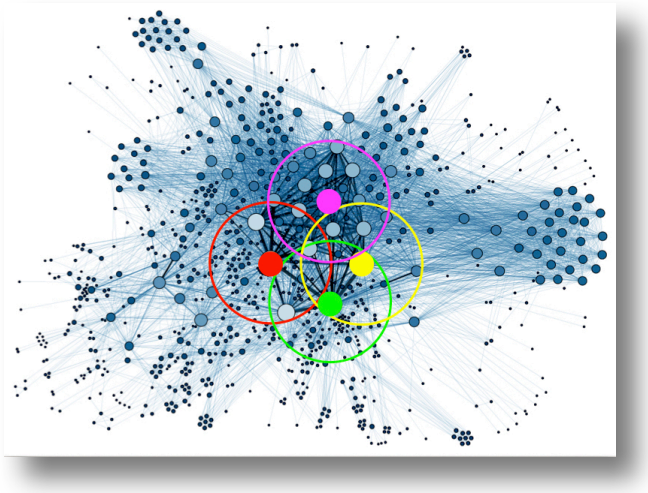

Figure 27. Euclidean distance threshold in the feature space

This approach solves both of HCA drawbacks mentioned before. First of all, the problem related to the ties (tie break) is bypassed by the fact that clusters can overlap. Meaning that, an element can belong to multiple clusters at once instead of being somehow parsed. Secondly, the problem of consistency within and between clusters is also bypassed, but this time, by the fact that clusters are standardized by the distance threshold constraint (user defined) and the complete-linkage method combined. This technique finally explains why the previous measures of similarity had to be normalized (see Distance triangulation). At this point, it is clear that if their scales were different, applying a consistent threshold would be impossible; the latter being inherent to the measurements.

c. Distance threshold

As mentioned before, if the target has to fall below a given threshold to be considered, the definition of space and the number of dimensions have a significant impact on the results, thus on setting the threshold itself (see Minkowski distance order). In other words, the threshold is a space dependent variable and thus should be adapted to every case. Although the ultimate objective is to identify clusters of sounds sharing strong similarities upon particular audio features, where the distance threshold would act as some kind of perceptual witness, the reality is that datasets are inflexible. In this sense, the algorithm cannot find what do not exist. Consequently, it appears to be useful to perform some descriptive statistics[26] on the feature-space distance matrix before clustering. The goal is to extract information that may help defining a distance threshold relevant to the current dataset.

[26] Descriptive statistics are statistic that quantitatively describe or summaries features of a collection of information. They are distinguished from inferential statistics (or inductive statistics), in that descriptive statistics aim to summarize a sample, rather than use the data to learn about the population that the sample of data is thought to represent. This generally means that descriptive statistics, unlike inferential statistics, are not developed on the basis of probability theory.

Some common statistics used to describe a dataset [Mann, 2006] are measures of:

- Central tendency: mean, median and mode.
- Variability or dispersion: extrema, standard deviation (or variance), kurtosis and skewness.
- Distribution: histogram and stem-and-leaf display.

In this case, information on the distribution of the dataset appears to be the most relevant. More specifically, the histogram seems to provide all the necessary information to guide the choice of a distance threshold. A histogram is a graphical representation of the distribution of numerical data. To construct a histogram, the first step is to divide the entire range of values into a series of intervals (bins). For that, the extrema are needed. Then count how many values fall into each bin. The bins are usually specified as consecutive, non-overlapping intervals of a variable. They must be adjacent and usually of equal size. Hence, histograms give a rough idea of the density of the underlying distribution of the data [Freedman, 2007]. The following figure 28, shows two histograms constructed upon the same dataset but using two different bin widths (*x* axis). The *y* axis represents the densities.

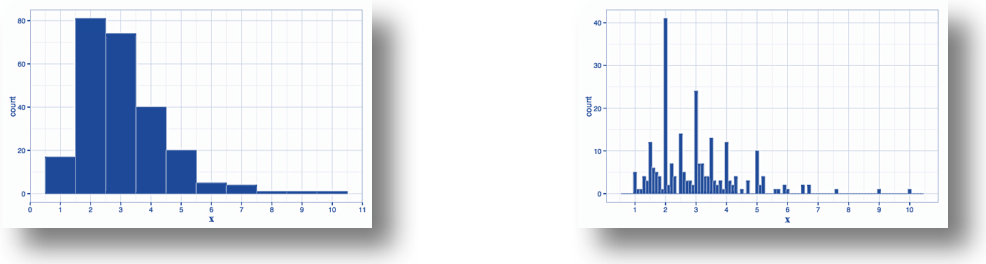

Figure 28. Histograms; [left] bin width = 1 [right] bin width = 0.1 [17]

Similar to resampling audio features (see Temporal alignment), this clearly demonstrates that different bin sizes can reveal different features of the data. Consequently, the bin width must be adapted to each case. However, there is no systematic guidelines to determine this parameter. Some theoreticians have attempted to determine an optimal number of bins (or bin width), but these methods generally make strong assumptions about the shape of the distribution [Venables, 2002]. Depending on the actual data distribution and the goals of the analysis, different bin width may be appropriate, so experimentation is usually needed to determine this parameter. Consequently, there are various approaches.

One of the most common method to determine the number of bins (k), used by Excel histograms [Cameron, 2009], is to take the square root of the number of elements (n) in the dataset; ($k = \sqrt{n}$). However, the number of bins can also be calculated from a suggested bin width (h) using the next formulation [Venables, 2002].

Formulation (9):

$$k = \left\lceil \frac{\max x - \min x}{h} \right\rceil$$

where, max x and min x are the extrema in the feature-space distance matrix.

In order to determine the bin width (h), different approaches may also be used. Two common ones of them are known as the Scott's normal reference rule and the Freedman-Diaconis rule [Freedman, 1981].
Formulation (10):

$$\text{Scott's rule, } h = \frac{3.5\sigma}{\sqrt[3]{n}}; \qquad \text{Freedman-Diaconis rule, } h = 2\frac{\text{IQR}(x)}{\sqrt[3]{n}}.$$

The noticeable difference between the previous rules is that the Freedman-Diaconis is based on the interquartile range[27] (IQR) instead of the standard deviation in the case of Scott's rule. The IQR, also known as the middle 50%,

[27] The IQR is a measure of variability, based on dividing a data set into quartiles. Quartiles divide a rank-ordered data set into four equal parts. The values that divide each part are called the first, second, and third quartiles; and they are denoted by Q_1, Q_2, and Q_3, respectively.

is a measure of statistical dispersion equal to the difference between the first and the third quartiles[28]; (IQR = $|Q_1 - Q_3|$). It is thus a trimmed estimator, defined as the 25% trimmed range, and is the most significant basic robust measure of scale[29] [Rousseeuw, 1993].

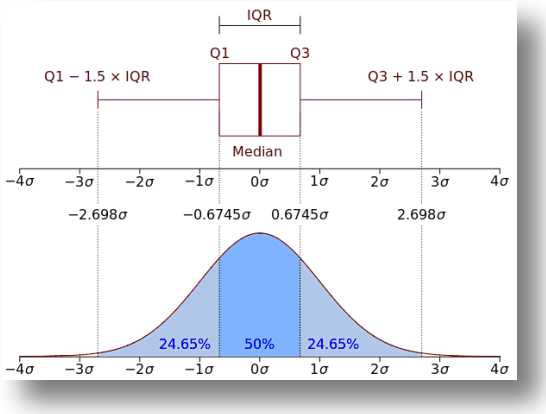

Figure 29. Interquartile range

Considering that the standard deviation is based on the arithmetic mean of a distribution, it is more likely to be inefficient for detecting the centre of mass in presence of outliers. On the other hand, being based on the median of a distribution, the IQR seems more adapted to detect where the real bulk of the values lie. Since the histogram should be as representative as possible of its underlying population, the Freedman-Diaconis rule was used to determine the bin width in the frame of this work. The number of bins can then be calculated from formulation 9 and 10 as follow.

Formulation (11):

$$k = \left\lceil \frac{\max x - \min x}{2\frac{\mathrm{IQR}(x)}{\sqrt[3]{n}}} \right\rceil$$

As mentioned before, the resulting histogram should be helpful to define a distance threshold. In this sense, the prominent peak(s) may serve as reference point(s). The following figures 30 and 31 illustrate various distributions to be taken into account [Freedman, 2007].

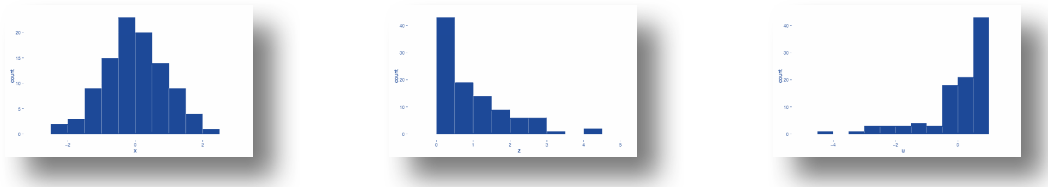

Figure 30. Unimodal; [left] symmetric [centre] skewed right [right] skewed left

In each case illustrated in figure 30, there is clearly a single prominent peak (unimodal) showing the largest amount of a specific distance between audio features. This information could be used straight forward to determine the distance threshold but, since the objective remains to cluster elements as similar as possible, it could lead to deceiving results. In the case of figure 30-right, where the histogram is completely skewed left, targeting the highest peak to set the distance threshold would lead to very large clusters. Meaning that the levels of similarity

[28] The quartiles of a ranked set of data values are the three points that divide the data set into four equal groups, each group comprising a quarter of the data. The first quartile (Q_1) is defined as the median between the smallest number and the median of the dataset. The second quartile (Q_2) is the median of the dataset. The third quartile (Q_3) is the median between the median and the highest value of the dataset.

[29] A robust measure of scale is a statistic that quantifies the statistical dispersion in a set of numerical data. The most common of such statistics are the interquartile range (IQR) and the median absolute deviation (MAD). These are contrasted with conventional measures of scale, such as sample variance or sample standard deviation, which are non-robust, meaning greatly influenced by outliers.

between the components could range from very high to very low. This recalls the problem of consistency within (not between) clusters regarding the classic HCA. This problem is also true in the cases presented in figure 31, where there are multiple prominent peaks (multimodal).

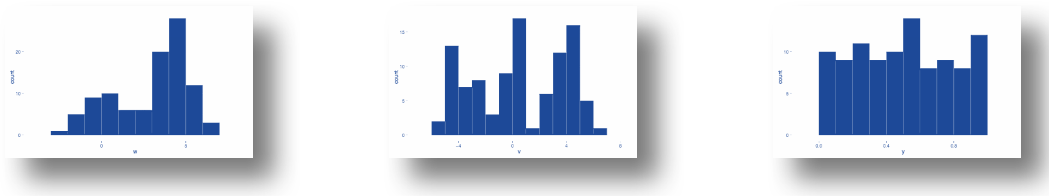

Figure 31. [left] bimodal [centre] multimodal [right] symmetric

Consequently, the histograms should be used to determine the region of the population where the distances are minimal for determining a proper threshold. However, in the case that none seem to satisfy the objectives, one may consider selecting other audio features to compare, and even consider modifying the original dataset itself. Although the histograms appear to be very convincing for taking such decisions, depending on the actual data distribution and the goals of the analysis, different distance threshold may be appropriate, so experimentation is usually needed in order to optimize this particular parameter.

d. The outcome

Using the clustering method described before leads to a particular outcome. Being more accurate and complete that in the case of a classic HCA, the resulting space is more complex. Related to the possibility of clusters to overlap, a new genre of hierarchy appears among the elements. That is because the overlaps allow a single element to belong to multiple clusters at once (see figure 27), thus making the clusters more difficult to distinguish. In a way, the hierarchy is blurred. Consequently, further processing is required to simplify the resulting data structure.

In the case that all the elements of one cluster are included within another cluster (sub-cluster), it appears to be relevant to merge them in order to avoid duplicates with different labels. The following figure 32 illustrates two cases of sub-clusters to be merged.

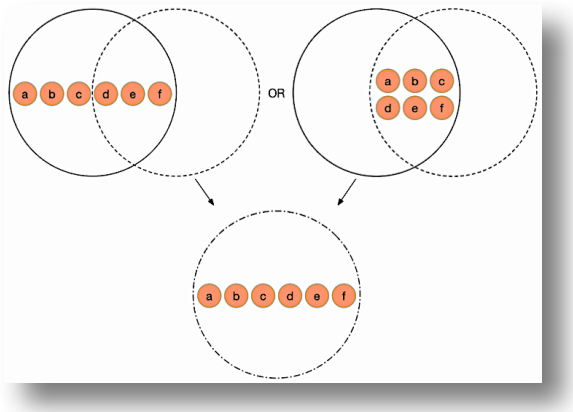

Figure 32. Sub-clusters merge

In the case where two clusters have distinguished and shared elements, many solutions may apply such as breaking the two clusters in three parts, but all lead to a conceptual clash. The essence of this work being based on the concept of agglomeration, it would be ludicrous to partition the resulting outcome. If so, another clustering method should be used earlier in the process (k-means, mixture models). For that reason, two overlapping clusters are simply considered as two different entities with common elements.

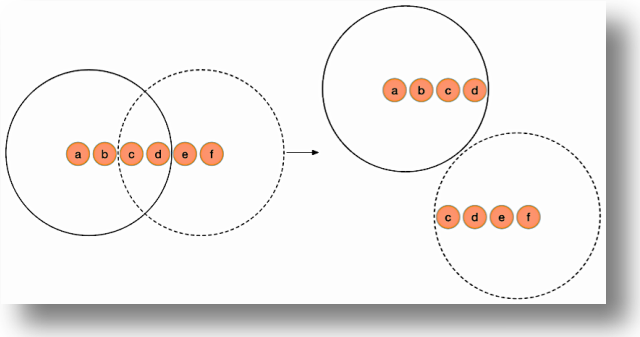

Figure 33. Overlapping clusters split

Being inherent to the agglomeration strategy described earlier (see Variation on HCA), this particularity becomes very interesting when translated into musical terms. In this sense, the overlaps may be seen as common notes between different chords. Such case often leads to what is known as voice-leading patterns. This technique is widely used in music to sequence various harmonic structures such as the typical (VI-II-V-I) chord progression in tonal language. This work being audio feature oriented, the previous analogy may translate to some kind of timbre voice-leading where the overlaps, or intersections between clusters, become pivots for commuting between two distinct groups of sounds [Huron, 2001]. In this sense, it appeared to be interesting to go one step further in order to have more insight on the clustered space. That is going towards a multi-layer relational space.

10. CLUSTERS SPACE ANALYSIS

The last step (post-clustering) is about gaining more insight on the resulting space. In the same way as the single-layer relational space presented earlier, the idea is to visualize the clustered space network. By analogy, it is to take a look at a down sampled version of the original dataset or looking at it using a larger bin width histogram in order to have a better understanding of its core structure. Consequently, this section covers the intra-cluster modelling and the inter-cluster analysis that are required to lead this way.

a. Intra-cluster modelling

Similar to the temporal modelling described earlier (see Temporal modelling), the intra-cluster modelling consists of generalizing the features of each cluster. The idea of creating such global models is to obtain a general, but still accurate, description of every single cluster. In a way, it is to find the theoretical centre of mass or the barycentre of each cluster. As for the previous single-layer relational space, that information is essential to outline the resulting network. Since the clusters are composed of sounds, the global models are obtained by merging the corresponding audio features and by accumulating the results as a list (vector), the latter being the global model itself. In this case, each data point of the resulting vector is calculated as the sum of the sums of each audio feature respectively.

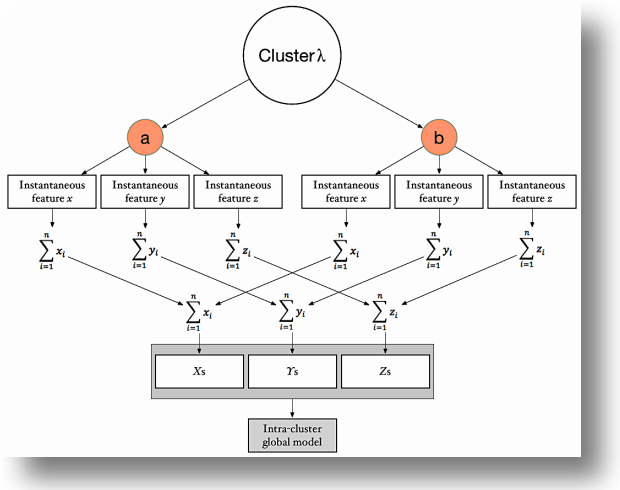

Figure 34. Intra-cluster modelling process

Although the resulting information is barely interpretable in terms of audio feature, it remains perfectly suited for distance based analysis (see Inter-cluster analysis). In addition to be a low cost processing, this approach does not alter the initial data in any way unlike averaging. In this sense, as mentioned earlier (see Global temporal model), it is important to remember that in the case of multidimensional features like the RSL (see figure 9), the summation and the accumulation must be done for each band (coefficient) separately in order to avoid any loss of information.

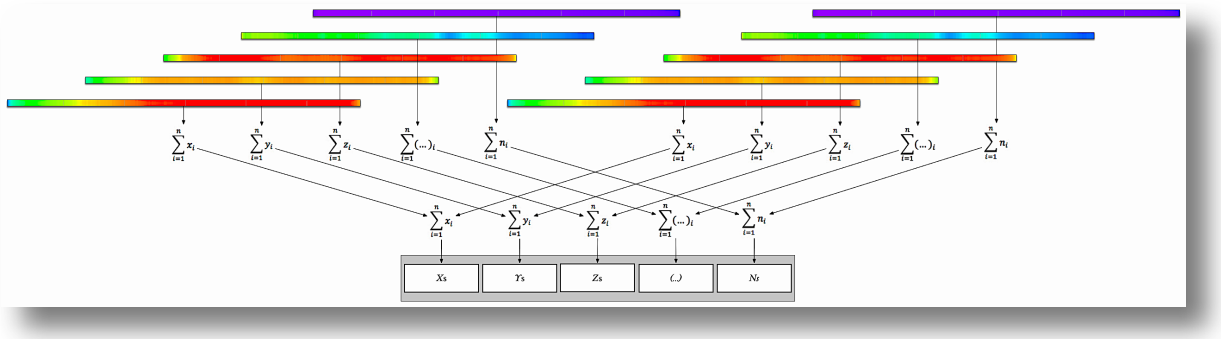

Figure 35. Multidimensional features modelling process

From the previous figures 34 and 35, it is clear that the resulting intra-cluster global model is expressed as a vector of information where each data point represents the sum of the sums of a specific audio feature. Below is a formulation of a global model ($\vec{\lambda}$) including unidimensional (x, y) and multidimensional (z) instantaneous features.

Formulation (12):

$$\vec{\lambda} = \left(\sum_{i=1}^{n}\left(\sum_{i=1}^{n}(x_i)\right), \sum_{i=1}^{n}\left(\sum_{i=1}^{n}(y_i)\right), \sum_{i=1}^{n}\left(\sum_{i=1}^{n}(z_{i_x})\right), \sum_{i=1}^{n}\left(\sum_{i=1}^{n}(z_{i_y})\right), \sum_{i=1}^{n}\left(\sum_{i=1}^{n}(z_{i_z})\right), \sum_{i=1}^{n}\left(\sum_{i=1}^{n}(z_{i_{(\dots)}})\right), \sum_{i=1}^{n}\left(\sum_{i=1}^{n}(z_{i_n})\right), \sum_{i=1}^{n}\left(\sum_{i=1}^{n}(\dots_i)\right), \sum_{i=1}^{n}\left(\sum_{i=1}^{n}(n_i)\right)\right)$$

b. Inter-cluster analysis

As mentioned above (see Intra-cluster modelling), the resulting vectors may be seen as the theoretical centre of mass or the barycentre of each clusters. Hence, the following inter-cluster analysis consists of measuring the distance between them. In this case, any type of distance may be used because the objective is no longer about clustering but rather about mapping (see Graph theory). In this sense, the scale and the shape of the space has no impact on the interpretation of the results (see Minkowski distance order) as long as they remain proportional into further manipulations. Since the Euclidean distance (Minkowski distance with order $p = 2$) was intuitively preferred throughout this work, the latter remains preferred here also.

Formulation (13):

$$d(\vec{\lambda}, \vec{\delta}) = \left(\sum_{i=1}^{n}|\lambda_i - \delta_i|^2\right)^{1/2} = \sqrt{\sum_{i=1}^{n}(\lambda_i - \delta_i)^2}$$

As for the single-layer relational space, a distance matrix is needed at this point to outline the network. That is done the same way as mentioned before (see Distance matrices) but using the previous formulation 13. Since the calculation is based upon clusters of sounds (audio features), the resulting matrix may be seen as a cluster-space distance matrix, by analogy to the previous feature-space distance matrix.

Although this could be sufficient to position the clusters between themselves and to outline the resulting network, it appeared to be wise to add a second element in the process. That is to use the Jaccard distance as described before in this document (see Jaccard index). While the Euclidean distance informs a metric distance between two theoretical barycentre, the latter informs a similarity level (distance interpretable) based on the ratio of two given cluster's intersection (shared components) and their union (combined components). Similar to the idea of distance triangulation, here also, it is about combining two different perspectives on a unique case in order to strengthen the results of the analysis. In this case, the Euclidean distance [0. +inf.] is simply weighted (multiplied) by the Jaccard distance [0. 1.]. While the latter is useful to arbitrate between two clusters having the same Euclidean

distance to a third one, the former is useful to keep track of two others not sharing any components (no intersection) with another one.

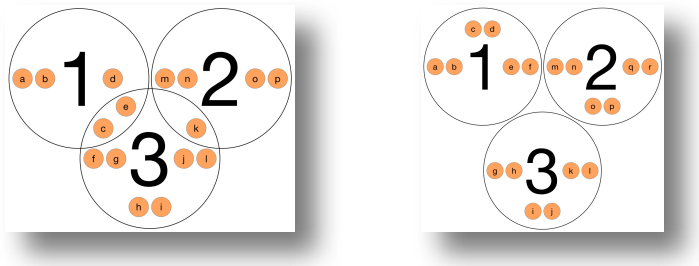

Figure 36. [left] Clusters with shared components, [right] Clusters without shared components

In the example above (figure 36-left), if the Euclidean distance between cluster no.1 and cluster no.3 equals 100, and that between cluster no.2 and cluster no.3 also equals 100, but that the Jaccard distance between cluster no.1 and cluster no.3 equals 0.82, and that between cluster no.2 and cluster no.3 equals 0.91, the conclusion is that cluster no.1 and cluster no.3 are closer (smallest distance). In the other case (figure 36-right), since there is no intersection between any of the clusters, the Jaccard distance equals 1 (maximum distance) for every pair. Thus, the Euclidean distance remains the only, and sufficient, measurement index to compare them.

c. Multi-layer relational space

Following the intra-cluster modelling and the inter-cluster analysis, the resulting cluster-space distance matrix can be used, also with the help of Gephi [Bastian, 2009], to visualize the underlying network. In this case, the nodes represent the clusters and the edges represent the distances between them. From this perspective, the resulting relational space is multi-layered. In other words, the network is itself composed of networks. Meaning that each node (cluster) embeds a smaller network of the same type. The latter may be seen as a local feature space.

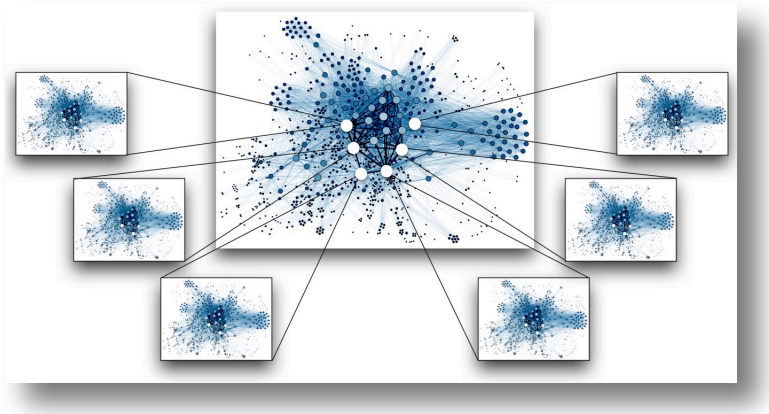

Figure 37. Multi-layer clusters space network plane designed with Gephi

Regarding the overlapping clusters, it is now clear that the more they overlap, the closer they are. Consequently, this approach already suggests ways for navigating through the network (distance based) towards the idea of timbre voice-leading (see The outcome). That concept being investigated later in this document (see Graph theory), clustering examples come next in order to demonstrate the application of such framework in different contexts.

11. EXEMPLIFICATION

The framework being completely unfolded at this point, this section aims at demonstrating the algorithm's clustering abilities. For that, the following examples were achieved using the same parameters over different sound corpuses. Hence, the idea is not to demonstrate the effects of the various combinations of parameters upon the clustering but rather to demonstrate the algorithm's global skills in different contexts. In this sense, the following examples target specific sources of sounds, natural and synthetic, that are compared and clustered from the same perspective without any posterior human manipulations.

a. Parameters and low-level descriptors

Below (see Table 8) is a complete list of the parameters and the low-level descriptors that were used to compute the following clustering. As mentioned above, they were applied exactly the same way for each sample of each dataset in order to observe different contexts from the same perspective. In this sense, the Mahalanobis distance was used solely for the sake of clarity. To some extent, that forces the clustering to be timbre oriented rather than morphology oriented or a mix of both, depending on the combination of distances (see Distance triangulation).

Parameters	Low-level descriptors
Analysis:	[Physical model]
Window type = Blackman	Temporal:
Max frequency = 22050Hz	Auto correlation
Window size = 0.4 seconds	Signal zero crossing rate
Hop size = 0.05 seconds	Harmonic:
FFT padding = 4	Fundamental frequency (F0)
Spectral descriptors parameters:	Inharmonicity
Auto correlation coefficients = 12	Noisiness
Reduced band(s) = [20 22050]Hz	Chroma
Perceptual descriptors parameters:	Chord sequence analysis (pm2)
Perceptual bands (bark) = 24	Masking effects (superVP)
MFCC coefficients = 13	[Perceptual model]
Harmonic descriptors parameters (pm2):	Energy:
Harmonics = 16	Loudness
F0 Max analysis frequency = 11025Hz	Relative specific loudness
F0 Min frequency = 20Hz	Loudness spread
F0 Max frequency = 5512.5Hz	Spectral:
Parameters that can apply to more than one descriptor:	MFCC
Roll off threshold = 0.9	Spectral centroid
Deviation stop band = 10	Spectral spread
Energy descriptors parameters:	Spectral skewness
Decrease threshold = 0.4	Spectral kurtosis
Noise threshold = 0.15	Spectral decrease
Chroma descriptors parameters:	Spectral rolloff
Chroma Min frequency = 20Hz	Spectral variation
Chroma Max frequency = 5512.5Hz	Spectral flatness
Chroma resolution (division of the octave) = 12	Spectral crest
	Harmonic:
	Spectral deviation
	Odd to even partials ratio
	Tristimulus

Table 8. Parameters and Low-level descriptors

b. Clustering examples

For the first example, the clustering algorithm was computed over a sound corpus made of prepared piano recordings. More specifically, the sounds are all produced from the same string (A4) but with different preparation using various object such as a clothes pin, a coin, an eraser, a mobile phone, two types of screws, foil paper, regular paper, glass, and different playing techniques such as muted, harmonics or normal (sustained) for a total of 13 different sounds. In this particular case, the idea is to test the algorithm upon what may seem as a highly correlated population. In other words, the task is to distinguish the different timbres despite the fact that the pitches, the dynamics and the energy envelopes are very similar and even identical in some cases.

The following figure 38 shows the clustering flowchart from which is extracted some information that clearly illustrate what happens along the process. As mentioned earlier (see Distance threshold), the histogram is used to visualize the sparsity of the features space with respect to the Freedman-Diaconis rule. The latter is built upon the distance matrix resulting from the features space analysis, meaning that each bin reflects the density of a specific range of distance. It is also used to determine a distance threshold prior to the clustering phase. Generally, a meaningful threshold is found below the first quartile (Q1) of the distances, and between the median and the arithmetic mean of the densities. In the following case, it is situated below the Q1 but a little above the mean at 0.56347.

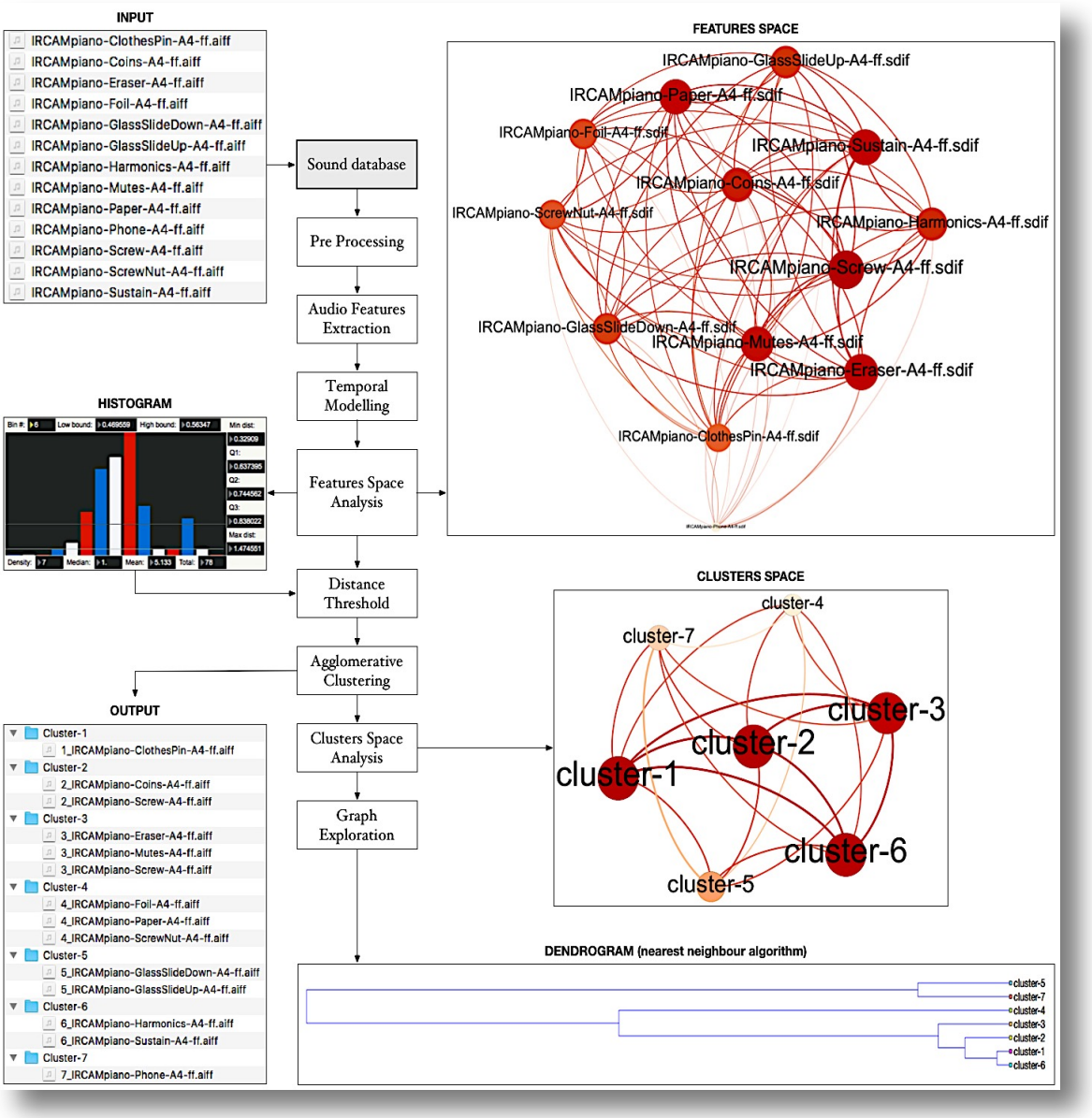

Figure 38. Clustering flowchart

The features space graph and the clusters space graph on the right side of the above figure 38 are not used for the clustering but simply to illustrate the shape of the data. Both are constructed using a multidimensional scaling[30] (MDS) algorithm [MDSJ, 2009] in order to optimize the components location in a 2D scatterplot. Consequently, the layout is not as accurate as the distance matrix used for the clustering but enough to render a fair representation of the global distribution. The colour and the size of each node indicate the level of their average weighted degree[31], meaning that the bigger and the darker they are, the higher is their average similarity to all the other components in the set, and vice versa. From a network perspective, the higher is their weighted degree, the more likely they are to be crossed. This creates a hierarchy based upon the strength of their network and fairly reflects their centrality or eccentricity within the group. The dendrogram is simply put as an introductory to the graph search algorithms further discussed in the next section (see Graph theory). Read from right to left, the horizontal lines

[30] Multidimensional scaling (MDS) is a mean of visualizing the level of similarity between components of a dataset. It refers to a set of related ordination techniques used in information visualization, in particular to display the data contained in a distance matrix. An MDS algorithm aims at placing each object in *N*-dimensional space such that the distances between objects are preserved as well as possible. Each object is then assigned coordinates in each of the *N* dimensions. Choosing *N*=2 optimizes the object locations for a two-dimensional scatterplot.

[31] In graph theory, the degree of a node in a graph is the number of edges incident (connected) to the node, with loops counted twice.

representing the distance between pairs of clusters, it may be seen as a way of sequencing them by proximity, from the closest to the farthest. In this case, one possible sequence of clusters could be: {6, 1, 2, 3, 4, 5, 7}.

Due to layout restrictions, the other examples are made available at http://repmus.ircam.fr/lebel. Those demonstrate the algorithm's clustering abilities in other contexts such as various saxophone multiphonics, different violoncello playing techniques, a pack of electronic sound design samples and a corpus of abstract electronic sounds.

12. GRAPH THEORY

The following section aims at shedding light on a potential bridge between the analytical approach and the compositional approach. Far from being exhaustive, due to time constraint, this part is simply an opening on the topic. Further development is planned for future works. Nonetheless, it remains interesting to tackle the subject in order to prefigure the use of the previous approach into compositional endeavours. Since the whole process leads to what may be represented as a graph (see Multi-layer relational space), the graph theory appears to be a legitimate start point for exploration. In this sense, what follows is an overview of the graph theory explained through different approaches and their potential applications in the continuity of this work.

a. Overview

In mathematics, graph theory is the study of graphs as mathematical structures (such as distance matrices) used to model pairwise relations between objects. A graph in this context is made up of nodes, vertices or points which are connected by edges, arcs or lines. A graph may be undirected, meaning that there is no distinction between two nodes, or may be directed, meaning that the distinction between two nodes is define by the direction of their common edge. A graph structure can be extended by assigning a weight to each edge of the graph. Graphs with weights (weighted graphs) are used to represent structures in which pairwise connections have some numerical values [Chartrand, 1985]. For example, if a graph represents a road network, the weights could represent the length of each road. The multi-layered cluster-space network presented earlier (see figure 37) is thus an undirected weighted graph where the weights represent the distances between all nodes. In computer science, graphs are used to represent networks of communication, data organization, computational devices, the flow of computation, etc. For instance, the link structure of a website can be represented by a directed graph, in which the nodes represent web pages and directed edges represent links from one page to another. A similar approach can be taken to problems in social media, travel, biology, computer chip design, and many other fields. The development of algorithms to handle graphs is therefore of major interest in computer science [Gibbons, 1985].

b. Approaches

In the frame of this work, the graph search algorithms seem to be particularly interesting. From the many topics related to the graph theory, these algorithms deal with problems linked to finding paths within graphs. By definition, a path in a graph is a finite or infinite sequence of edges connecting a sequence of nodes. In this sense, the definition of a path becomes specific to a given problem. Consequently, there are as many solutions as there are problems. However, three types of approaches seem to cover a fair amount of them. These are respectively related to the Eulerian graph, the Hamiltonian graph and the Spanning tree graph.

Eulerian graph:

The Eulerian graph originates from the Seven Bridges of Königsberg, which is a historically notable problem in mathematics. Its negative resolution by Leonhard Euler in 1736 laid the foundations of graph theory and prefigured the idea of topology[32]. The city of Königsberg in Prussia (now, Kaliningrad, Russia) was set on both sides of the Pregel River, and included two large islands which were connected to each other and the mainland by seven bridges. The problem was to devise a walk through the city that would cross each bridge once and only once, with the contingency that: the islands could only be reached by the bridges and every bridge once accessed must be crossed to its other end. Euler proved that the problem has no solution [Euler, 1736]. This lead to the definition of the Eulerian path and the Eulerian cycle. The former (path) is a trail in a graph which visits every edge exactly once. The starting and ending points do not need to be the same. The latter (cycle) is thus a trail which starts and ends on the same node. Consequently, a cycle is always a path but a path is not always a cycle. The necessary condition for the existence of such cycles is that all nodes in the graph have an even degree. For the existence of Eulerian paths, it is necessary that zero or two nodes have an odd degree; this means the Königsberg graph is not

[32] In mathematics, topology (from the Greek topo, *place*, and logy, *study*) is concerned with the properties of space that are preserved under continuous deformations, such as stretching and bending, but not tearing or gluing. This can be studied by considering a collection of subsets, called open sets, that satisfy certain properties, turning the given set into what is known as a topological space. Important topological properties include connectedness and compactness.

Eulerian. If there are no nodes of odd degree, all Eulerian paths are cycles. If there are exactly two nodes of odd degree, all Eulerian paths start at one of them and end at the other. A graph that has a Eulerian path but not a Eulerian cycle is called semi-Eulerian [Hazewinkel, 2002].

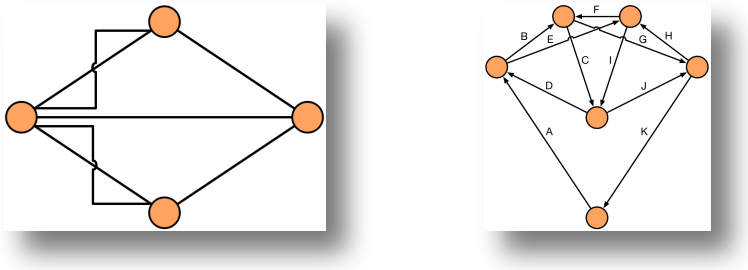

Figure 39. [left] Königsberg bridges graph, [right] Alphabetical ordered Eulerian cycle

Hierholzer[33] algorithm for constructing Eulerian cycles [Hierholzer, 1873]:

- Choose any starting node *v*, and follow a trail of edges from that node until returning to *v*. It is not possible to get stuck at any node other than *v*, because the even degree of all nodes ensures that, when the trail enters another node *w* there must be an unused edge leaving *w*. The cycle formed in this way is a closed cycle, but may not cover all the nodes and edges of the initial graph.
- As long as there exists a node *u* that belongs to the current cycle but that has adjacent edges not part of the cycle, start another trail from *u*, following unused edges until returning to *u*, and join the cycle formed in this way to the previous cycle.
- Besides, the algorithm needs to maintain the set of unused edges incident to each node, to maintain the list of nodes on the current trail that have unused edges, and to maintain the trail itself.

Another famous problem related to Eulerian cycles is the route inspection problem, also known as the Chinese postman problem[34]. Similarly, the task is to find the shortest closed path that visits every edge of an undirected graph. When the graph has a Eulerian cycle, the latter is the optimal solution. Otherwise, the optimization problem is to find the smallest number of graph edges to duplicate (or the subset of edges with the minimum possible total weight) so that the resulting multigraph does have a Eulerian cycle [Kwan, 1962].

Considering that in the multi-layer relational space each cluster is connected to every other clusters, thus all having the same number of connections (*n*-1), the only way for it to have an even number of edges is to have an odd number of nodes (clusters). In such case, no Eulerian paths can be find but only Eulerian cycles. That also brings the number of cycles to be *n* factorial and the problem of finding them to be more like a creative one than an optimization one. However, the following approaches suggest otherwise.

Hamiltonian graph:

Contrary to the Eulerian graph, the Hamiltonian one is verified when a path in a graph can visit each node exactly once (not each edge). However, the Hamiltonian cycle is also a trail which starts and ends on the same node [Hazewinkel, 2002]. Determining whether a Hamiltonian path (or cycle) exists in a given graph is known as the Hamiltonian path problem[35]. Cut short, the latter is a special case of the well-known traveling salesman problem (TSP). Similar to the route inspection problem mentioned before (see Eulerian graph), the TSP asks the following question: Given a list of cities and the distances between each pair of them, what is the shortest possible path that visits each city exactly once and returns (or not) to the origin city? In this sense, the TSP can be modelled as an undirected weighted graph where the cities are the graph's nodes, paths are the graph's edges and a path's distance is the edge's length [Applegate, 2007]. Often, the model is a complete graph[36] where each pair of nodes is connected by an edge (our case).

[33] Hierholzer proved that a graph has a Eulerian cycle if and only if it is connected and every vertex has an even degree (excluding the starting and terminal nodes). This result had been given, without proof, by Leonhard Euler in 1736. Hierholzer apparently explained his proof, just before his premature death in 1871, to a colleague who then arranged for its posthumous publication which appeared in 1873.
[34] The problem was originally studied by the Chinese mathematician Kwan Mei-Ko in 1960, whose Chinese paper was translated into English in 1962. The alternative name "Chinese postman problem" was coined in his honor.
[35] The Hamiltonian path problem is a special case of the travelling salesman problem obtained by setting the distance between two cities to one if they are adjacent, or two otherwise, and verifying that the total distance travelled is equal to *n*. In such case, the path is a Hamiltonian cycle. If there is no Hamiltonian cycle, then the shortest path will be longer.
[36] In the mathematical field of graph theory, a complete graph is a simple undirected graph in which every pair of distinct nodes is connected by a unique edge. In other words, all the nodes are connected to each other.

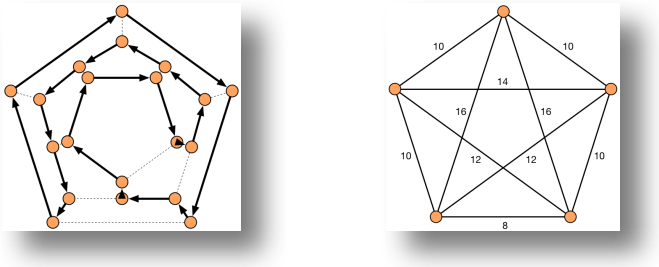

Figure 40. [left] Hamiltonian cycle in a dodecahedron, [right] Undirected weighted complete graph

Usually, two approaches may apply to solve a TSP. The first is to use exact algorithms[37], which works reasonably fast for small problem sizes. The second one is to use heuristic algorithms[38], which deliver either seemingly or probably good solutions but cannot be proved to be optimal. Considering that the running time for exact algorithm lies within the factorial of the number of cities, $O(n!)$, this approach becomes impractical even for only 20 cities [Bellman, 1962]. In this sense, the second approach (heuristic algorithms) is most likely to be preferred.

One of the first algorithm used to solve the TSP, is the nearest neighbour (NN) algorithm. Similar to the *k*-NN mentioned before (see Hierarchical cluster analysis), it is used to sequence the nodes rather than to cluster them. In a way, it lets the salesman choose the nearest unvisited city as his next move. This algorithm quickly yields an effectively short route. For *n* cities randomly distributed on a plane, the algorithm yields, on average, a path 25% longer than the shortest possible path [Johnson, 1997]. However, there exist many specially arranged city distributions which make the NN algorithm give the worst route [Gutin, 2002].

Another one that has been used to produce near-optimal solutions to the TSP is known as the Ant colony optimization algorithm (ACO). Artificial intelligence researcher Marco Dorigo described in 1992 a method of heuristically generating good solutions to the TSP using a simulation of an ant colony [Dorigo, 2010]. In the natural world, ants of some species initially wander randomly, and after finding food return to their colony while laying down pheromone trails. If other ants find such a path, they are most likely not to keep travelling randomly, but instead to follow the trail, returning and reinforcing it if they eventually find food. Over time, however, the pheromone trail starts to evaporate, thus reducing its attractive strength. The more time it takes for an ant to travel down the path and back again, the more time the pheromones have to evaporate. A short path, by comparison, gets marched over more frequently, and thus the pheromone density becomes higher on shorter paths than longer ones. If there were no evaporation at all, the paths chosen by the first ants would tend to be excessively attractive to the following ones. In that case, the exploration of the solution space would be constrained. The overall result is that when one ant finds a short path from the colony to a food source, other ants are more likely to follow that path, and positive feedback eventually leads to all the ants following a single path. The idea of the ant colony algorithm is to mimic this behaviour with simulated ants walking around the graph representing the problem to solve.

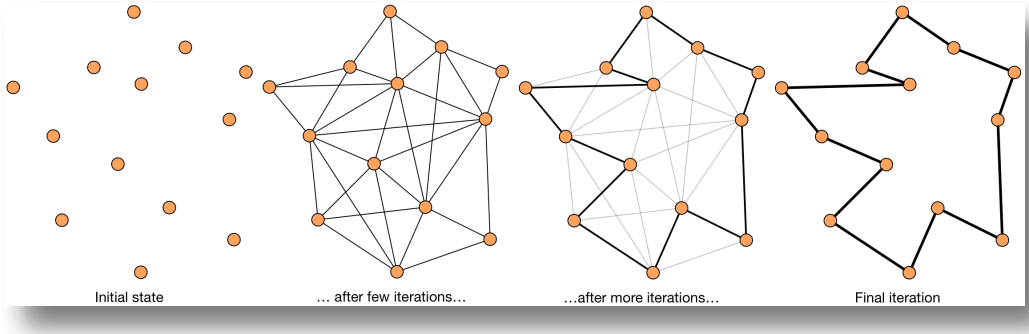

Figure 41. Steps of the ACO algorithm to solve the TSP

[37] In computer science, exact algorithms are those that always solve an optimization problem to optimality.
[38] In computer science, artificial intelligence and mathematical optimization, a heuristic is a technique designed for solving a problem more quickly when classic methods are too slow, or for finding an approximate solution when classic methods fail to find any exact solution.

Clearly, the TSP relates to the previous multi-layer relational space in which the clusters, also separated by fixed distances, could be seen as the cities or the food sources. Therefore, the idea of finding the shortest path that visits each cluster exactly once apply quite naturally. In such case, the path would represent an ordered sequence of clusters for which the total distance is minimal. Depending on the algorithm, the distance between each of them may also be minimal. In other words, the clusters would be organized by proximity. In this sense, this approach appears to be an interesting way of formalizing music towards the idea of timbre voice-leading (see The outcome).

Spanning tree graph:

A spanning tree *T* of an undirected graph *G* is a subgraph that is a tree which includes all of the nodes of *G*. In general, a graph may have several spanning trees, but a graph that is not connected[39] will not contain a spanning tree but rather a spanning forest. The main difference with the previous graphs (Eulerian and Hamiltonian) is that a tree has no cycle. Thus it is not an ordered sequence nor a path. In this sense, it can be defined as a maximal set of edges of *G* that contains no cycle [Bollobas, 1998], or as a minimal set of edges that connects all nodes [Cameron, 1994]. Several pathfinding algorithms, including Dijkstra's algorithm[40] [Dijkstra, 1959] and the A* search algorithm[41] [Hart, 1968], internally build a spanning tree as an intermediate step in solving the problem. The number of spanning tree of a complete graph t(*G*) is a well-studied invariant. For these with *n* nodes, Cayley's formula gives the number of spanning trees as n^{n-2} [Aigner, 2014]. Also, if there are *n* nodes in the graph, then each spanning tree has n-1 edges.

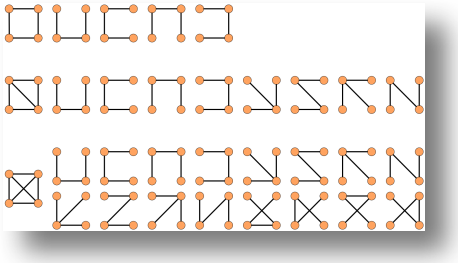

Figure 42. Spanning trees where the bottom line demonstrates the Cayley's formula

A single spanning tree of a graph can be found using either depth-first search[42] or breadth-first search[43] [Shimon, 2011]. Both of these algorithms explore the given graph starting from an arbitrary node *v*. Then, by looping through the neighbours of the nodes they discover, and by adding each unexplored neighbour to a data structure to be later explored. They differ in the way this data structure is ordered. In either case, they form a spanning tree by connecting each node, other than the root node *v*, to the node from which it was discovered. This tree is known as a depth-first search tree or a breadth-first search tree according to the algorithm used to construct it. The former is associated to the maze solving algorithms such as the wall follower[44].

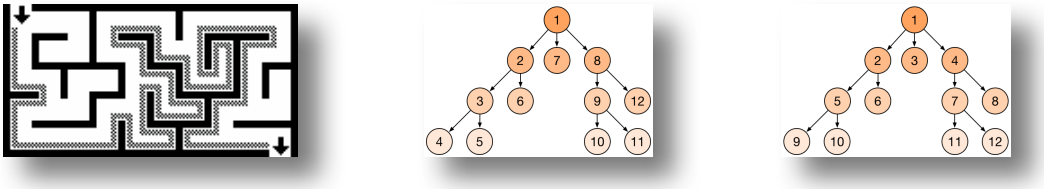

Figure 43. [left] Right-hand wall follower, [centre] Depth-first search, [right] Breadth-first search

[39] A graph is connected when there is a path between every pair of nodes. In a connected graph, there are no unreachable nodes. A graph that is not connected is disconnected. A graph with just one node is connected. An edgeless graph with two or more nodes is disconnected.
[40] Dijkstra's algorithm is an algorithm for finding the shortest paths between nodes in a graph, which may represent, for example, road networks. It was conceived by computer scientist Edsger W. Dijkstra in 1956 and published three years later.
[41] In computer science, A*, an algorithm that is widely used in pathfinding and graph traversal, is a process of plotting an efficiently traversable path between multiple nodes.
[42] Depth-first search (DFS) is an algorithm for traversing or searching tree or graph data structures. It starts at the root (selecting some arbitrary node as the root in the case of a graph) and explores as far as possible along each branch before backtracking.
[43] Breadth-first search (BFS) is an algorithm for traversing or searching tree or graph data structures. It starts at the root (selecting some arbitrary node as the root in the case of a graph) and explores the neighbor nodes first, before moving to the next level neighbors.
[44] The wall follower, the best-known rule for traversing mazes, is also known as either the left-hand rule or the right-hand rule. If the maze is simply connected, meaning that all its walls are connected together or to the maze's outer boundary, then by keeping one hand in contact with one wall of the maze the solver is guaranteed not to get lost and will reach a different exit if there is one.

In certain fields of graph theory, it is often useful to find a minimum spanning tree (MST) of a weighted graph. Other optimization problems on spanning trees have also been studied, including the maximum spanning tree, the minimum tree that spans at least *k* nodes, the spanning tree with the fewest edges per node, the spanning tree with the largest number of leaves (closely related to the Hamiltonian path problem), the minimum diameter spanning tree, and the minimum dilation spanning tree [Eppstein, 1996]. For similar reasons as for the previous TSP (see Hamiltonian graph), the MST seems to be particularly interesting to explore the multi-layer relational space.

Similar to the TSP, a minimum spanning tree (MST) is a subset of the edges of an undirected connected weighted graph that connects all the nodes together with the minimum possible total edge weight but without any cycles. In other words, it is a spanning tree (not a path) whose sum of edge weight is as small as possible [Graham, 1985].

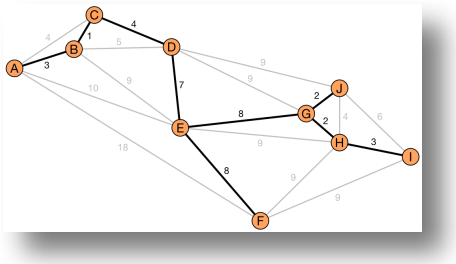

Figure 44. Minimum spanning tree (MST)

The first algorithm for finding a MST was developed by Czech scientist Otakar Boruvka in 1926. Its purpose was an efficient electrical coverage of a specific region of the country (Moravia). The algorithm being sequential, for each step it identifies a forest[45] *F*, consisting of the minimum-weight edge incident[46] to each node in the graph *G*, to form the graph $G_1 = G/F$ as the input to the next step. Here, G/F denotes the graph derived from *G* by contracting edges in *F*. The process is repeated until a tree covers all nodes in the graph [Nesetril, 2001].

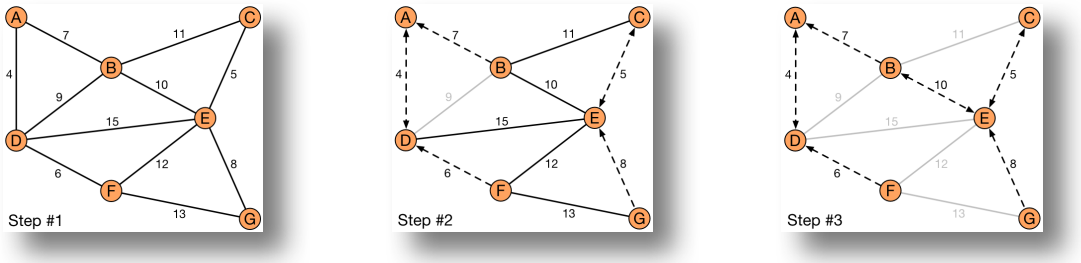

Figure 45. [From 1 to 3] Iteration process of Boruvka's algorithm

Another one is the Prim's algorithm, which was invented by Jarnik in 1930 and rediscovered by Prim in 1957 and Dijkstra in 1959. Similar to the previous algorithm, the latter grows the MST (*F*) one edge at a time. Initially, *G* contains an arbitrary node. For each step, *F* is augmented with the minimum-weight edge connecting to another node. The process is also repeated until a tree spanning all nodes is completed [Prim, 1957].

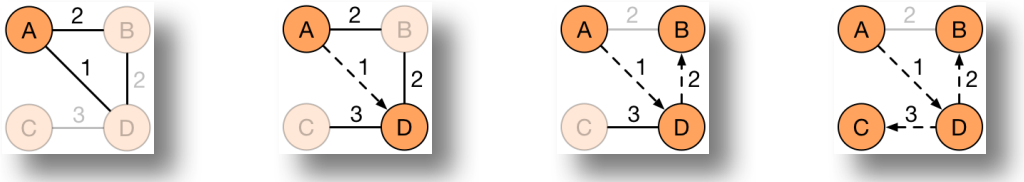

Figure 46. [left] Initial state, [centre-left] First iteration, [centre-right] Second iteration, [right] Final iteration

[45] A forest is an undirected graph where all the connected components are trees. In other words, the graph consists of a disjoint union of trees. Equivalently, a forest is an undirected acyclic graph. As special cases, an empty graph, a single tree, and the discrete graph on a set of nodes (that is, the graph with these nodes that has no edges), are examples of forests.
[46] Two edges are called incident, if they share a node.

Seemingly, the Prim's algorithm acts similar to the nearest neighbour algorithm mentioned before (see Hamiltonian graph). In this sense, the resulting MST may be quite different for the same graph.

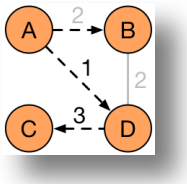

Figure 47. Same graph as figure 45 with a different MST

One more is the Kruskal's algorithm that first appeared in 1956, named after its creator Joseph Kruskal. Similar to the previous ones, it is a greedy algorithm that finds a MST for a connected weighted graph by adding increasing weight edges at each step. First, it creates a forest *F* where each node in the graph is a separate tree. Then, it creates a set *S* containing all the edges in the graph. While *S* is nonempty and *F* is not yet spanning, it removes an edge with minimum weight from *S* and add it to *F*, combining two trees into a single tree. The procedure is repeated until a tree covers all nodes in the graph [Kruskal, 1956].

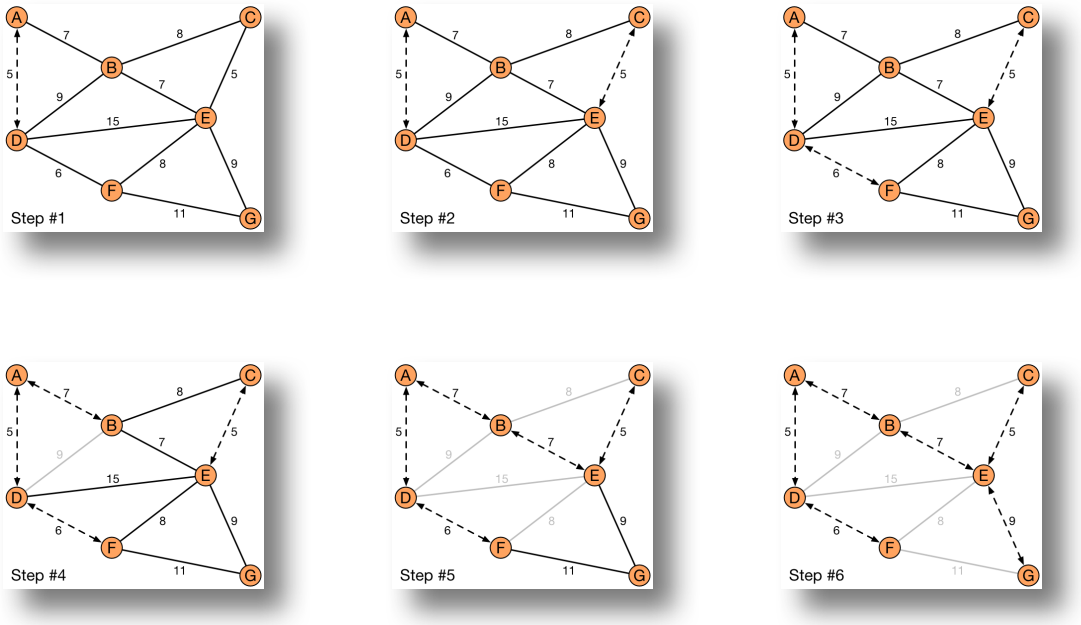

Figure 48. [From 1 to 6] Iteration process of Kruskal's algorithm

As described, the MST is similar to the TSP in the way that both approaches look for a subset of edges that connects all the nodes together with the minimum possible total weight (or distance). The main difference between them is that the TSP leads to a sequenced solution (path or cycle) while the MST leads to a solution that is not sequenced (no path nor cycle). Hence, they provide a different point of view on the same problem. While the algorithms used to solve the TSP suggests a closed pattern solution for exploring the space, the MST suggests an open one. In a way, the latter offers more flexibility to explore the space and eventually to formalize music. In this sense, the MST approach also appears to be an interesting way to explore the clustered space and to nourish the idea of timbre voice-leading.

c. Conclusion

Clearly, the graph theory suggests various ways to explore the multi-layer relational space resulting from the agglomerative clustering technique described through this document. Although the Eulerian approach does not seem to be particularly helpful to explore complete graphs (our case), the concept remains interesting and could lead to develop various creative process of exploration within the same clustered space. Based on further analysis, the exploration may be guided by various user defined sets of constraints. Regarding the Hamiltonian approach, it

clearly leads to optimization problems such as the travelling salesman problem (TSP) mentioned before. Although the idea of finding the shortest path that visits each cluster exactly once appears to apply quite naturally from a timbre voice-leading perspective (see The outcome), it may be too restrictive in the frame of other compositional endeavours. In this sense, the minimum spanning tree (MST) is another interesting way of exploring graphs. Contrary to the TSP, the MST leads to a solution that is not sequenced (no path nor cycle). Consequently, it offers more flexibility to explore the space and eventually to formalize music. Although this section covered only a few types of graph search algorithms, it is clear that there is a large potential into using those towards formalizing music posterior to audio classification. Therefore, further investigations will be conducted in this field to attempt bridging the clustering process and the compositional process. However, it should be kept in mind that since the definition of a path is problem specific, the definition of a problem remains composition specific.

13. FROM SCIENCE TO MUSIC

Also due to time constraint, the artistic outcome could not make exhaustive use of the clustering process nor the graph search algorithms. Nonetheless, many concepts related to this specific research program were translated into musical ideas and compositional techniques. Consequently, this section mostly focuses on the influence of such approach rather than on its technical integration into the compositional process. Without being a detailed analysis of the piece (Alors que le monde est décomposé, for piano & electronics), what follows intends to highlight some of the key concepts that were used to nourish musical ideas, compositional techniques and various scenarios of interaction between the piano and the electronics.

a. Compositional approach

The agglomerative clustering concept being the core of this work, musical ideas had to emerge from related concepts. In this sense, the piece massively exploits the notion of distance that is inherent to the clustering process. The idea was not to mimic the algorithm itself for grouping similar sounds together, but rather to expand the notion of distance and similarity to multiple layers, locally and globally, in a way to create different scenarios of interaction between the musical components within the piano part as well as between it and the electronic part.

Local scale:
As illustrated in the following figure 48, the opening of the piece is laid out as some kind of timbre drifting. Simply by gradually removing the finger from the string, the sound shifts from percussive (unpitched) to harmonic (A4), and from harmonic to normal (A3) before introducing the lower octave (A2). In this simple sequence, the spectrum changes quite drastically while the chroma (pitch content) remain relatively constant. In a way, the spectrum evolves by overlapping. As the harmonic sound introduces the pitch aspect, the percussive element remains, and as the normal sound enriches the harmonic spectrum, the previous harmonic naturally blends into it. Adding the lower octave just keep the harmonic spectrum growing without any serious spectral modulation. Extended by the electronic part, the sound also grows into a subtle polyphony of timbre as they slowly spread into the diffusion space. Here, the approach is to translate the idea of sounds smoothly commuting from one state to the other without any kind of transformations. In a way, the sounds are gradually moving away from each other.

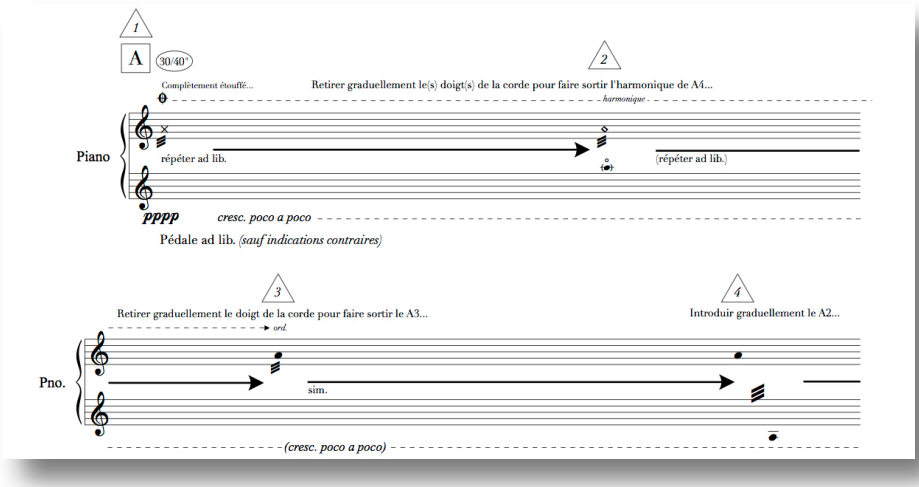

Figure 49. Timbre drifting (0:00 – 0:27)

Based on the audio features related to the low-level descriptors mentioned before (see Table 1), each type of scenarios intends to use a specific property of the sound material such as the durations, the dynamics, the pitch structures, the spectrum, the densities, the spreads, the articulations or the behaviours to name a few, in order to build a dialectic that can evolve through time. Once again, the goal remains to exploit the similarities between the components in order to contextualise them, not to morph one into another but simply to connect them.

In the following example (figure 50), the articulation of the sound material is exploited in order to transit from one section (letter C) to the other (letter D). Starting from a trill, material used in the previous section, the sequence evolves in a way to install a dialogue between the arpeggi and the tremoli that is itself later developed towards the next section (letter E). Looking at the beginning of the following figure, it is clear that the constant parameter is the fast iterative aspect (articulation) while the changing parameter is the pitch content (density), gradually growing from a group of two notes to a group of eight before shifting at letter D. From another perspective, the trill element introduced in the previous section (letter C), then acting as some kind of ornament, gradually becomes the transitional element. This can be seen as some sort of functional shifting.

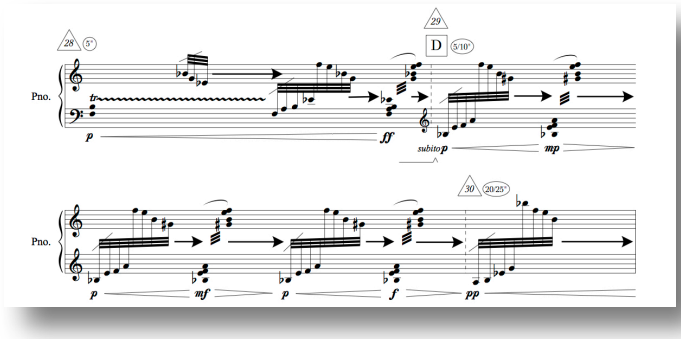

Figure 50. Articulation transformation (3:34 – 3:43)

In figure 51, the two elements (tremoli and martelés) are gradually blended using a simple rhythmic compression. Here, the idea is to go from a smooth chord tremolo to a percussive cluster martelé. As the tremolo opens this passage, the cluster is then introduced as distinct *sffz* impulsions in between which the duration shortens every time, creating an acceleration effect, until reaching the speed of a tremolo before ending into a violent sequence of glissandi. Then, as the instrument resonates ferociously, the tremoli reappear to transit from letter D to letter F. Similar to the previous example, the principle remains to use similar features between two components in order to transit from one to another, this time being the rhythmic and the behavioural aspect of the sound.

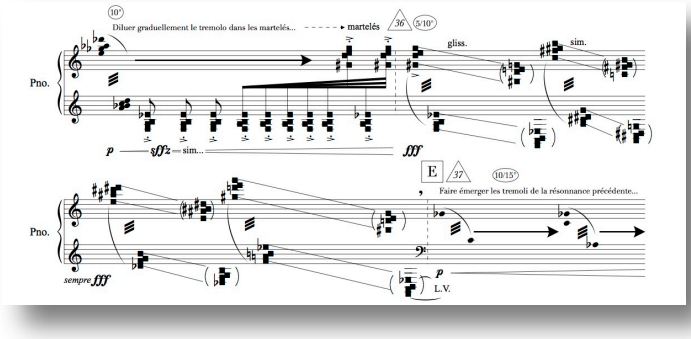

Figure 51. Rhythmic transformation (3:56 - 4:08)

In the continuity of the previous example, the tremoli reappear to transit from letter D to letter F (see figure 52). Here, the transitional elements are the density and the register of the chord. This passage may be seen as a single chord that is gradually unfolded in a certain way and then refolded in another way, the folding point being at *ff senza dim*. This can be verified as the number of notes evolves for each step of the sequence E: {2, 3, 4, 5, 6, 5, 4, 3, 2, 1}. Sliding from the upper to the lower register, it is clear that the chord density increases, then decreases (<>) before ending to the repeated C#. The pitch being stabilised and preparing the upcoming chord in letter F, the rhythmic dilation serves as the transitional element towards setting down the pulse for the next section (q=40).

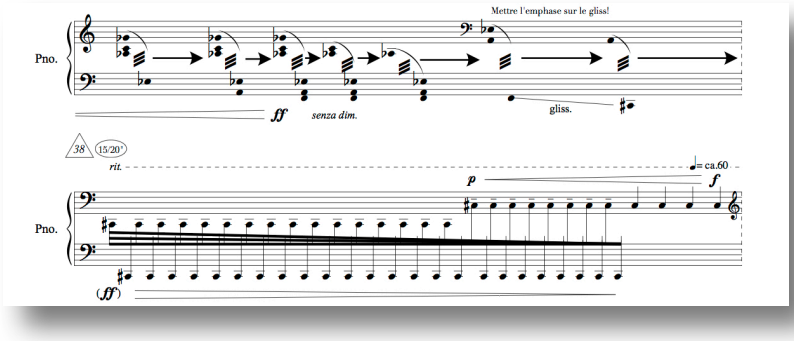

Figure 52. Density transformation (4:08 – 4:42)

Following the previous example, the C# extends over the next sections, F and G, as a floor note sustaining a slow process of harmonic transformation. Before entering the process at G, the pitch structure reveals itself gradually as each note in the sequence adds its own flavour to the final chord (minor 9th). In this sense, as the B pitch seems to fade in at the beginning of this passage, it is actually relayed from the electronic part. After revealing the full chord, a process of harmonic transformation starts at letter G. The transformation starts by adding foreign pitches, or some sort of distortion, in the resonance of the chord. Then, those are gradually integrated in the chord itself, altering the original structure, and the process goes on the same way until number 45. Basically, the passage from letter E to letter H included may be seen as a long sequenced process of transformation, or a segmented interpolation, inside of which each part leads to the other relatively smoothly.

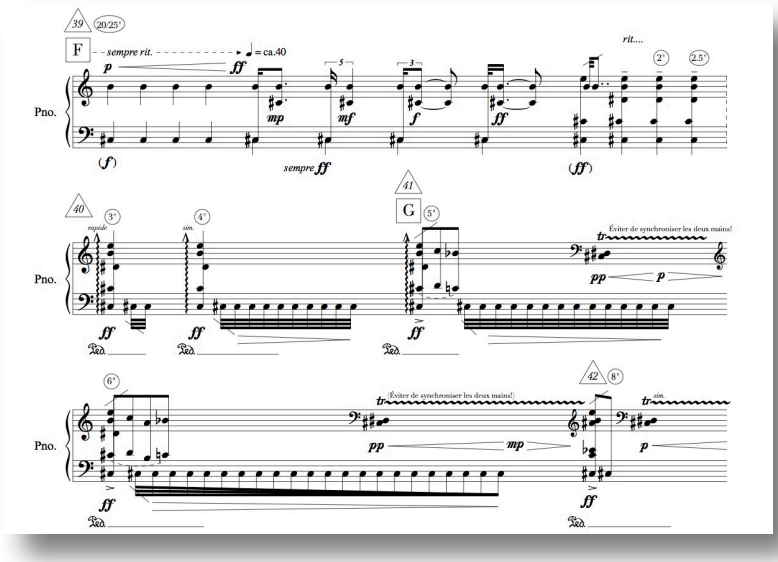

Figure 53. Pitch structure transformation (4:42 – 5:12)

Global scale:
Now that the notion of distance is clearly illustrated as a way of sequencing the material for creating smooth transition between various musical components (locally), it is interesting to mention that it is also used through relatively abrupt changes (globally). That is simply to reflect various levels of similarity, sometimes being strong (as in the previous examples) and some others being weaker (as in the following examples), through different contexts. In this sense, the next examples emphasize three types of transitions that use lower levels of similarity between their components.

The first example (figure 54) simply make use of a common note (A4) to transit from one section to the other. Although the timbre (normal piano sound) remains the same in this case, the transition is definitely more abrupt than seen in the previous examples since the similarity is based on fewer elements.

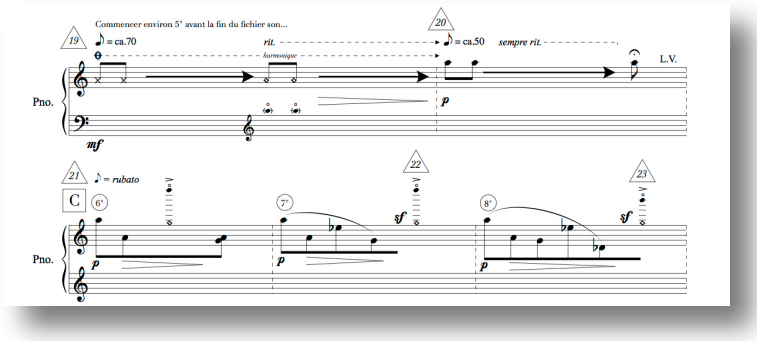

Figure 54. Common note transition (2:05 – 2:27)

The second example illustrated in figure 55 also makes use of a common note (C#3) to transit from one section to the other, but this time, with a different timbre (from normal sound to harmonic sound). A little more abrupt than in the previous case, this change seems to have a stronger impact on the perception although the pivot element (C#) is clearly stated.

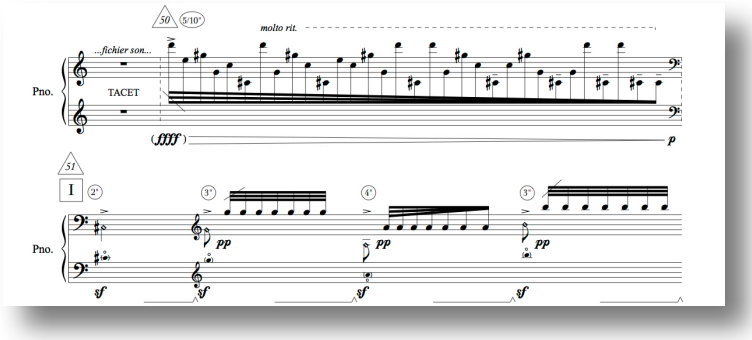

Figure 55. Common note and different timbre transition (6:36 – 6:56)

In the third example, both the pitch and the timbre change through the transition between one section (#52) to the other (#53). In this case, although the registers are very close to each other, the perception is somehow tricked by these two simple changes and thus makes this transition again a little more abrupt.

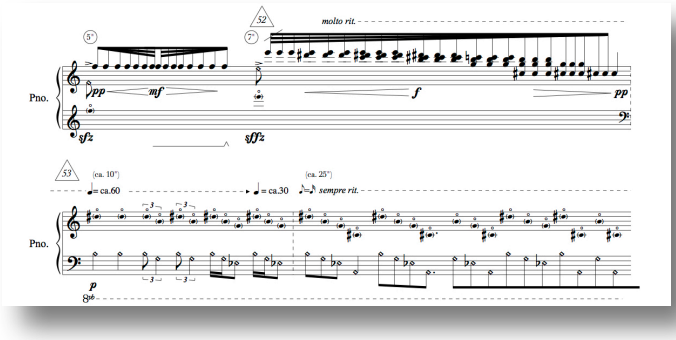

Figure 56. Different note and different timbre transition (7:18 – 8:20)

As seen in the previous examples, each scenario intuitively targets specific audio features (energy, pitch, spectrum, density, time, behaviour, rhythm, etc.) and projects them into various shifting processes in order to create some kind of a smooth and perpetual drifting motion through the music. In a way, that is to reflect the idea of exploring the shortest path(s) within a graph mentioned earlier in this document (see Graph theory).

b. Electronic approach

As mentioned before, the artistic outcome could not make exhaustive use of the clustering process nor the graph search algorithms at this point but, here also, the notions of distance and similarity apply into creating different scenarios of interaction between the piano part and the electronic part. Not being a real-time version of the clustering algorithm described in this document, the signal processing is designed in the same way as the piano part is composed, somehow to expand the notion of distance through mixing both mediums together. In this sense, what follows is a few highlights of the key concepts that influenced the design of the electronic part.

Generally, not too distant from the original sound of the piano, the electronic part is mostly based upon spectral transformations. Ranging from spectral delays to FFT based granular synthesis, passing by spectral resonators, physical model-based sound synthesis, FFT based time stretching and classic flanging, the idea is to keep a relatively high level of control on the depth of the transformations applied to the source sound. Somehow, it is to keep control over the distance, or the blend, between the two parts. It is also a way to drift smoothly from one to the other with a certain elegance and transparency.

Online:
An example of such drifting appears in the first section of the piece (letter A) as seen in the previous figure 49. Extended by the use of granular synthesis, the sound of the piano grows into a subtle polyphony of timbre as it slowly spreads into the diffusion space. For that, the sources slowly move along the azimuth, gradually increasing the intimacy, until reaching their final destinations and covering 360 degrees around the audience. As illustrated in the following figure 57, the process is split into four interpolated scenes.

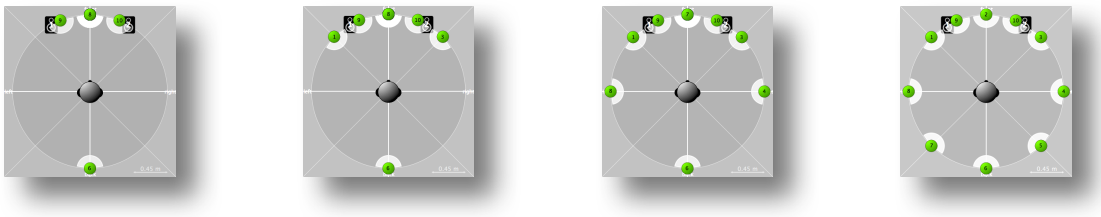

Figure. 57 [from left to right] Evolution of the diffusion space (0:00 – 0:55)

While the sound gradually fills the physical space, it also gets richer and thicker, mixing different timbres and opening the register, until the culminating point at #7 in the following figure 58. Synchronized with the previous spatial pattern, the impulsion (*sffz*) at the end of the process is amplified by the use of a spectral delay acting as some kind of gong. Creating a massive spectral wave, the idea is to scatter the source sound into the diffusion space. Similar uses appear at #27 (3:25 – 3:35) and #50 (6:37 – 6:56) in the score.

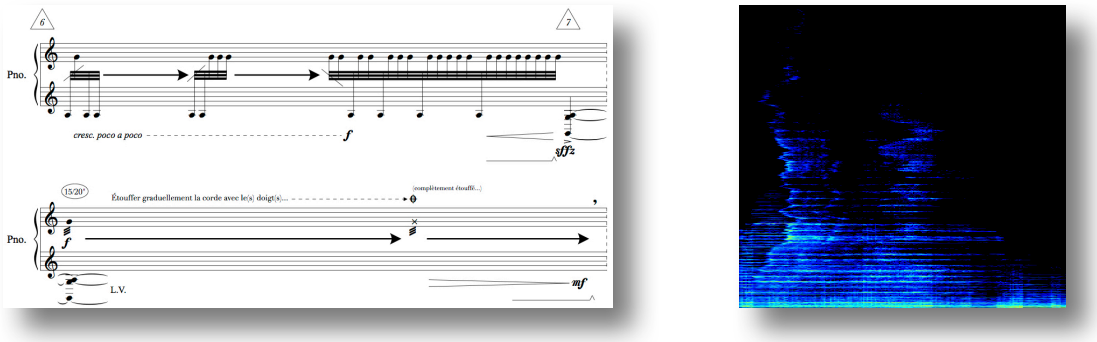

Figure. 58 Spectral delay (0:32 – 0:55)

In the continuity of the previous example, the following one (figure 59) extends the granular synthesis by the use of a transposition pattern based on the piano part where A2 is the reference pitch. For each note added in the piano part, the corresponding transposition is added to the electronic part. That way, the colour and the density of the sound keep evolving similarly in both parts. In this sense, each grains of sound from the electronic part is randomly spatialized over 360 degrees in a way to mimic the shivering aspect of the piano part. In this case, the idea is to project the same material on two different scales simultaneously for creating a strong cohesion between them.

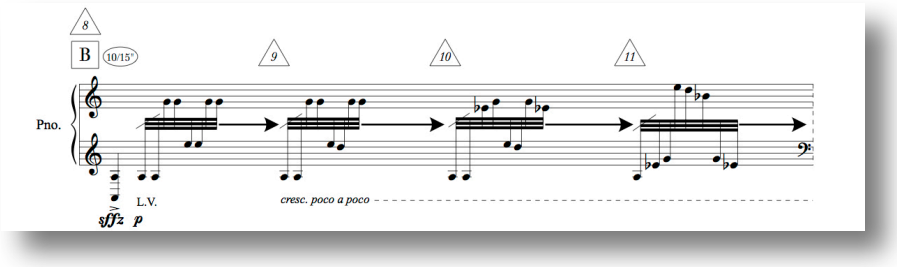

Figure. 59 Transposition pattern (0:55 – 1:09)

Similar to the previous example, the idea behind the following passage (figure 60) is to project the piano part onto a different scale. In this case that is done using adaptive spectral resonators. Those are constructed in real-time using the piano part as the harmonic model upon which the multiple resonators are aligned. More specifically, harmonic peaks are extracted from a real-time analysis computed on the piano part, and are used to target specific bands of the same spectrum in order to let them resonate simultaneously. Each successive configuration of resonators, corresponding to each frame of the analysis, is interpolated for smoothing the transitions between the different resulting sounds. Various frequency shifting scales are also used to vary the positions of the latter, thus creating some distance between the acoustic and the electronic resonances. In the passage below, the dynamic waves result in a repetitive crossfade pattern letting the two parts affirm their colours alternately.

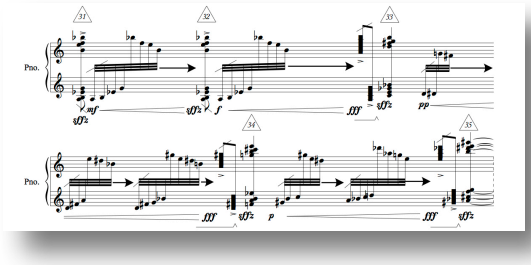

Figure. 60 Spectral resonators (3:43 – 3:56)

The next example demonstrates a different kind of scenario where the transformations do not involve the spectrum of sounds but rather their temporality. More specifically, two types of temporal processing are applied simultaneously. The first one is a common time stretching and the other one is some sort of time mangling. In the first case, the piano part is stretched by a hundred times its original length. Affecting drastically the morphology of sounds, this transformation results in a voice-like effect giving the impression of a choir accompanying the piano. In the second case, the mangling effect is literally produced by cutting chunks of sounds, between one and three seconds each, in order to sequence them in a different order. Affecting nothing but the chronology of events, this action creates some kind of mnemonic disturbance as it is played almost at the same time as the original material. Resulting in a very homogeneous texture, the three superimposed layers of sounds are then smoothly moved around the diffusion space, gradually accelerating, in order to create a kind of temporal storm. Contrary to the previous scenarios, this one focuses on creating spatial and temporal distance between the two parts.

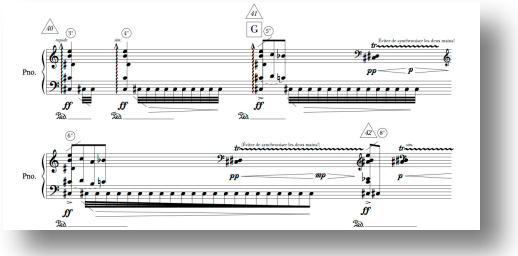

Figure. 61 Time stretching and time mangling (5:01 – 5:25 – […] – 6:05)

The next case, corresponding to figure 62, also demonstrate a different scenario of interaction. Here, the signal is not processed in any way but rather used to excite two types of physical model-based synthesised instruments. Basically, the raw input signal (piano) passes through the virtual instruments in order to put them into vibration and let them resonate. In a way, it is like trading the body of the piano for the body of another instrument. Acting as some kind of natural resonators, this may be seen as a type of sound hybridization. In the following passage, a C#1 (+25cents) synthetic string is used at #51 and four parallel tubes are used at #52. Built to produce different pitches (B1, C2, D#2, E2) that fit the piano part, the tubes setup acts like some sort of natural harmonizer. This subtle interaction creates a genuine intimacy, or closeness, between the piano and the electronics. A similar case appears at letter C in the score (2:21 – 3:22) where metal plates are used as the virtual instruments.

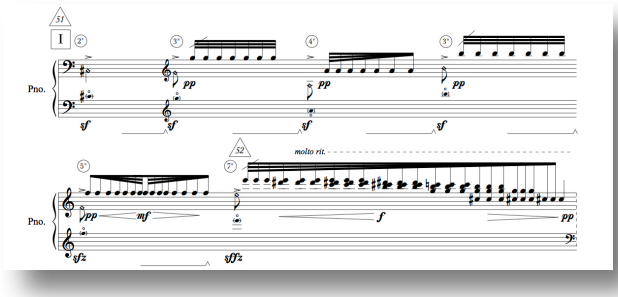

Figure. 62 Physical model-based synthesis (6:50 – 7:40)

Offline:
The next examples demonstrate scenarios of interaction based on pre-processed sound files that are used to transit between different sections of the piece. In these cases, the idea is less about transforming and mixing the mediums but more about complementing them. In other words, it is to use one as the extension of the other in a way to expand the auditory scene beyond the acoustical instrument itself. Somehow, the idea is to explore the space between the physical and the digital world.

In the case shown in the following figure 63, the sound file appears in the continuity of the local climax created by the piano part. Through a slow crossfade, the electronic sound emerges from the mass as a second level of resonance that keeps transforming until a sharp A5 pierces the high register. Through another crossfade, the latter is then relayed by the piano part in order to complete the transition towards the next section C as described previously in figure 54. A similar example, using a common note in both parts, appears through letter E and F as described in figure 52 and 53.

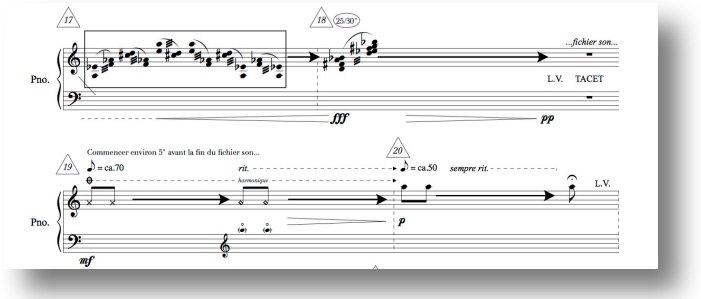

Figure. 63 Offline transition (1:35 – 2:22)

The next example (figure 64) illustrates another use of a sound file to transit from one section to another. In this case, it is used to articulate the massive resonance created by the piano in the low register and to transform it into a completely different sound in the high register. Impossible to achieve with live electronics, elements from the initial state (#49) are gradually transformed into elements from the final state (#50). More specifically, the register shifts from low to high, the density modulates from heavy to light, the colour turns from dark to bright and the texture transforms from smooth to granular. Culminating in a material fragmentation, the piano is reintroduced mimicking the electronic part before leading to the next section as described before in figure 55.

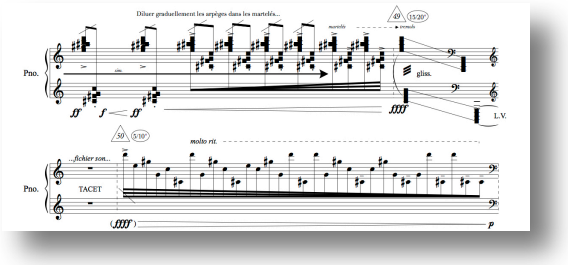

Figure. 64 Another offline transition (6:05 – 6:51)

The last example as such is shown in figure 65. This one does not actually involve a sound file but the effect is similar, giving the impression of an independent element. In this case, the piano part is synthesized in real time, using a mix of the previous spectral resonators and the physical model-based instruments (the four parallel tubes) passed through granular synthesis, and time stretched by a hundred times its original length, in order to create what could be perceived as a sound file. In conclusion of the piece, the idea is to let the electronic sound drift away from its source as some kind of a floating ghost-like spectral hallo. Contrary to the previous cases, this one is a one-way process towards the unknown.

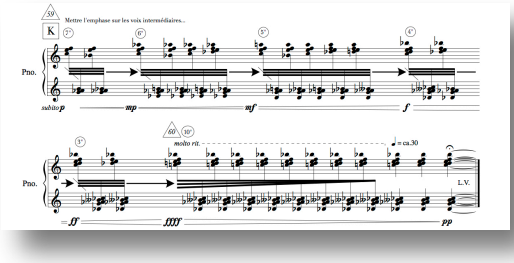

Figure 65. Online transition (9:31- al fine)

As seen in the previous examples, each type of DSP leads to a different level of interaction, sometimes intimate and some others distanced, based on various features of sounds (energy, pitch, spectrum, density, time, behaviour, space, etc.). In this sense, the electronic part is designed to reflect the notions of distance and similarity through the polyphony of sounds by adding texture and depth to the instrumental part, vertically as well as horizontally.

14. CONCLUSIONS AND PERSPECTIVES

Based on previous works achieved in the field of music information retrieval such as corpus-based concatenative synthesis [Schwarz, 2006], musical genre recognition [Peeters, 2007] and computer-aided orchestration [Carpentier, 2008], this document exposed a different framework for audio classification with applications to computer-aided composition, as well as a first artistic outcome: Alors que le monde est décomposé, for piano & electronics. Contrary to its predecessors, this framework is built towards formalizing music rather than generating sound material. In other words, it is engineered to act on a larger scale than in the other cases. Consequently, each part of the structure is designed in an attempt to render this level of perspective through analysis and clustering.

The first part, including the pre-processing, the audio features extraction and the temporal modelling, is designed in a way to optimize the transformation of a raw signal into a smaller space of variables [Malt, 2012] that can be perceived and interpreted by humans. In other words, the idea is to shape the data in the same way the ear does it for the brain, as well as the brain does it for the mind; aesthetical and emotional affects put aside. In this sense, the pre-processing stage consists of reducing the information to what is perceptually consistent in the audio files. Applying different kinds of filters, the idea is to emulate the human selective listening skill. The audio features extraction consists of decomposing the sounds into specific elements based on their energy, their spectrum and their harmonic content, either from a physical or a perceptual model [Zwicker, 2007] of a raw signal. As one may listen to the same sound from different perspectives, segregating the different components, the idea is to project this ability into a computerized sound analysis. The temporal modelling can be divided in two steps. The first one, similar to the pre-processing stage, consists of reducing the information to what is perceptually consistent in the

audio features by applying different processing [Peeters, 2004]. Here again, the idea is to mimic the selective listening skill. The second step is about shaping the data considering the effect of time on audio perception. Briefly, it is to determine, based on different assumptions, how the various components of sounds are to be compared together despite having different durations. Overall, this part of the framework focuses on moulding the digital data through the perceptual data in order to obtain clustering results that are consistent for listeners.

The second part, including the features space analysis and the agglomerative clustering, is designed in a way to find aggregates of sounds having high levels of similarity based on specific audio features. In this sense, the features space analysis consists of determining the level of similarity (or dissimilarity) between sounds prior to the clustering. Somehow, that is to unravel the latent network for later identifying the subpopulations within the whole. In order to achieve such task, the first step is to compute a distance matrix that quantifies the connections between each pair of sounds [Gentle, 2007]. For that, three approaches may be used. The first approach is based on distances [Hazewinckel, 2012], the second one is based on similarities [Singhal, 2001] and the third one is based on correlations [Rakotomalala, 2015]. Since each approach describes a different aspect of a relationship (magnitude, orientation and dependency), they may be merged later into a single multidimensional score using a simple triangulation technique. The idea is to obtain a single value that includes all three perspectives. Using a similar technique, the audio features may also be used collectively in order to account for higher level descriptions of sounds into the clustering process. The second and last step towards identifying the subpopulations is based on the hierarchical cluster analysis (HCA) method [Hastie, 2009]. More specifically, the clustering is done using an ad hoc variation that makes use of a distance threshold instead of a linkage criterion. In this case, the speed of HCA is traded for more accuracy. Informed by a histogram based on the Freedman-Diaconis rule [Freedman, 1981], the distance threshold is applied to the features space distance matrix in order to agglomerate similar sounds together, and to reveal the subpopulations and their network. Overall, this part of the framework strives to computerize the way one would intuitively aggregate sounds by proximity, or similarity, including the possibility of sounds to be part of multiple subpopulations at once. Despite the absence of evaluation procedures for measuring the quality of the clustering, the experiments conducted in various contexts had yet shown very positive results from a perceptual angle.

The third and last part, including the clusters space analysis and the graph exploration, is design in a way to provide more insight on the resulting clusters space and to suggest ways of navigating through it and towards formalizing music. In this sense, similar to the features space analysis, the clusters space analysis consists of determining the distance between the resulting clusters of sounds in order to outline the resulting network. For that, the first step is the intra-cluster modelling. In a way, the idea is to find the theoretical centre of mass, or the barycentre, of each cluster in order to be able to measure the distance between them. In other words, it is to generalize, or summarize, the data (audio features) contained in each cluster. The second step, inter-cluster analysis, is then to make the measurements. In this case, that is done weighting a metric distance (Euclidean) that is calculated between the barycentre [Verley, 1997], by a similarity index (Jaccard) that is calculated between the sample sets [Jaccard, 1901]. Leading to a well-defined clusters space network, the graph theory may then be used to explore the latter towards formalizing music. In this sense, different approaches may be adopted. Although this topic remains to be further investigated, two of them yet appear to apply quite naturally to this kind of network. Both based on the idea of finding the shortest path within a graph, the first one is related to Hamiltonian graphs [Hazewinckel, 2002] and the other one is related to minimum spanning trees [Graham, 1985]. Resulting in a closed sequence pattern, the former quickly suggests a way of organizing the different clusters of sounds on the time axis while the latter leads to a solution that is not sequenced (no path nor cycle). Hence, the MST offers more flexibility to explore the space and to develop different patterns towards formalizing music. That being said, since no experiments were conducted at this point, further development is planed into future works.

Due to time constraint, the artistic outcome could not make exhaustive use of this framework neither, but many concepts could be translated into musical ideas and compositional techniques. In this sense, the piece massively exploits the notion of distance that is inherent to the clustering process, locally and globally, in a way to create different scenarios of interaction between the musical components within the piano part as well as between it and the electronic part. More specifically, each scenario intuitively targets specific audio features (energy, pitch, spectrum, density, time, behaviour, space, etc.) and projects them into various shifting processes in order to create some kind of a smooth and perpetual drifting motion through the music. In a way, that reflects the idea of exploring the shortest path(s) within a graph. Consequently, the electronic part is designed to reflect the same concepts but through the polyphony of sounds, sometimes intimate and some others distanced, by adding texture and depth to the instrumental part, vertically as well as horizontally. Witnessing the great potential of such approach through the compositional process, a work for augmented string quartet is already underway in order to continue developing applications for computer-aided composition.

REFERENCES

[Peeters, 2007] G. Peeters, "A Generic System for Audio Indexing: Application to Speech/Music Segmentation and Music Genre Recognition", Proc. Of the 10th Int. Conference on Digital Audio Effects (DAFx-07), Bordeaux, France, 2007.

[Schwarz, 2006] D. Schwarz, G. Beller, B. Verbrugghe, S. Britton, "Real-time Corpus-based Concatenative Synthesis with Catart", Proc. Of the 9th Int. Conference on Digital Audio Effects (DAFx-06), Montreal, Canada, 2006.

[Carpentier, 2008] G. Carpentier, "Approche computationelle de l'orchestration musicale: Optimisation multicritère sous contraintes de combinaisons instrumentals dans de grandes banques de sons", Thèse de doctorat, Université Paris VI – Pierre et Marie Curie (EDITE), Paris, France, 2008

[Hastie, 2009] T. Hastie, R. Tibshirani and J. Friedman, "The Elements of Statistical Learning: Data Mining, Inference, and Prediction: Hierarchical Clustering", Springer Series in Statistics, Springer New York, 2009.

[Malt, 2012] M. Malt, "Une proposition pour l'analyse des musiques électroacoustiques de Xenakis à partir de l'utilisation de descripteurs audio", La musique électroacoustique de Xenakis – actes du colloque, Université Paris 8, 2012.

[Zwicker, 2007] E. Zwicker and H. Fastl, "Psychoacoustics: Fatcs and Models", Springer-Verlag Berlin Heidelberg, 2007.

[Zwicker, 1980] E. Zwicker and E. Terhardt, "Analytical expressions for critical-band rate and critical bandwidth as a function of frequency", The Journal of Acoustical Society of America. Vol. 68, 1523, 1980.

[Peeters, 2004] G. Peeters, "A Large Set of Audio Features for Sound Description (Similarity and Classification) in the CUIDADO Project", Project report, Ircam, Paris, France, 2004.

[Cooley, 1965] J. W. Cooley and J. W. Tuckey, "An Algorithm for the Machine Calculation of Complex Fourier Series", Mathematics of Computation, Vol. 19, No. 90, pp.297-301, 1965.

[Depalle, 1993] P. Depalle, G. Garcia, et al., "Tracking of Partials for Additive Sound Synthesis using Hidden Markov Models", Proceedings of the IEEE-ICASSP, Mineapolis, USA, 1993.

[Moore, 1997] B. C. J. Moore, B. R. Glasberg and T. Baer, "A Model for the Prediction of Threshold, Loudness, and Partial Loudness", Departement of Experimental Psychology, University of Cambridge, Cambridge, UK, pp.224-240, 1997.

[Rabiner, 1993] L. Rabiner and B-H. Juang, "Fundamentals of Speech Recognition", Prentice-Hall, New Jersey, USA, 1993.

[Cella, 2011] C. E. Cella, "On Symbolic Representations of Music: The Theory of Sound-Types", Dottorato di ricerca in mente, logica e linguaggio, Settore scientifico di afferenza: M-FIL/02, Universita di Bologna, Bologna, Italy, 2011.

[Huang, 1979] T. Huang, G. Yang and G. Tang, "A fast two-dimensional median filtering algorithm", IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. 27, no. 1, pp.13-18, 1979.

[Salvador, 2004] S. Slavador and P. Chan, "FastDTW: Toward Accurate Dynamic Time Warping in Linear Time and Space", Department of Computer Sciences, Florida Institute of Technology, Melbourne, Florida, USA, 2012.

[Verley, 1997] J-L. Verley, "Dictionnaire des Mathématiques. Algèbre, analyse, géométrie", Encyclopedia universalis, Albin Michel, Paris, France, 1997.

[Hazewinkel, 2002] M. Hazewinkel (Ed.), "Encyclopedia of Mathematics", Springer Netherlands, 2002.

[Jaccard, 1901] P. Jaccard, "Distribution de la flore alpine dans le bassin des Dranses et dans quelques regions voisines", Bulletin de la Société Vaudoise des Sciences Naturelles, Vol. 37, pp.241-272, 1901.

[Singhal, 2001] A. Singhal, “Modern Information Retrieval: A Brief Overview”, Bulletin of the IEEE Computer Society Technical Committee on Data Engineering, Vol. 24, pp.35-43, 2001.

[Rakotomalala, 2015] R. Rakotomalala, “Analyse de correlation: Étude des dépendences – Variables quantitatives”, Version 1.1, Université Lumière Lyon 2, Lyon, France, 2015.

[Gentle, 2007] J. E. Gentle, “Matrix Algebra: Theory, Computations, and Applications in Statistics”, Springer-Verlag New York, 2007.

[Roweis, 2000] S. T. Roweis and L. K. Saul, “Nonlinear Dimensionality Reduction by Locally Linear Embedding”, Science, Vol. 290, Issue 5500, pp.2323-2326, 2000.

[Pudil, 1998] P. Pudil and J. Novovocova, “Novel Methods for Feature Subset Selection with Respect to Problem Knowledge”, Feature Extraction, Construction and Selection: A Data Mining Perspective, The Springer International Series in Engineering and Computer Science, Vol. 453, pp.101-116, Springer US, 1998.

[Bellman, 1957] R. E. Bellman, “Dynamic programming”, Princeton University Press, Princeton, New Jersey, USA, 1957.

[Bastian, 2009] M. Bastian, S. Heymann, M. Jacomy, “Gephi: an open source software for exploring and manipulating networks”, International AAAI Conference on Weblogs and Social Media, 2009.

[Defays, 1977] D. Defays, “An efficient algorithm for a complete link method”, The Computer Journal, Btritish Computer Society, Vol. 20, pp. 364-366, 1977.

[Orange, 2013] J. Demsar, T. Curk et al., “Orange: Data Mining Toolbox in Python”, Journal of Machine Learning Research, August 14, 2013.

[Mann, 2006] P. S. Mann, “Introductory Statistics”, Sixth revised edition, John Wiley & Sons, New Jersey, USA, 2006.

[Freedman, 2007] D. Freedman, R. Pisani and R. Purves, “Statistics”, Fourth edition, W W Norton & Co Inc., New-York, USA, 1997.

[Cameron, 2009] A. C. Cameron, “EXCEL 2007: Histogram”, Department of Economics, University of California, Davis, USA, 2009.

[Venables, 2002] W. N. Venables and B. D. Ripley, “Modern Applied Statistics with S”, Statistics and Computing, Springer-Verlag New York, 2002.

[Freedman, 1981] D. Freedman and O. Diaconis, “On the histogram as a density estimator”, Zeitschrigt für Wahrscheinlichkeitstheorie und verwandte Gebiete, Vol. 57, Issue 453, 1981.

[Rousseeuw, 1993] P. J. Rousseeuw and C. Croux, “Alternatives to the Median Absolute Deviation”, Journal of the American Statistical Association, Vol. 88, pp. 1273-1283, 1993.

[Huron, 2001] D. Huron, “Tone and Voice: A Derivation of the Rules of Voice-leading from Perceptual Principles”, Music Perception, Vol. 19, No. 1, pp. 1-64, 2001.

[Chartrand, 1985] G. Chartrand, “Introductory Graph Theory”, Dover Books on Mathematics, Dover Publications, 1985.

[Gibbons, 1985] A. Gibbons, “Algorithmic Graph Theory”, Cambridge University Press, King’s College London, London, UK, 1985.

[Euler, 1736] L. Euler, "Solutio problematis ad geometriam situs pertinentis", Commentarii academiae scientiarum Petropolitanae, Vol. 8, Issue 1741, pp.128-140, 1736.

[Hierolzer, 1873] C. Hierholzer, "Ueber die Möglichkeit, einen Linienzug ohne Wiederholung und ohne Unterbrechung zu umfahren", Mathematische Annalen, Vol. 6, Springer, 1873.

[Kwan, 1962] M-K. Kwan, "Graphic programming using odd or even points", in Chinese Mathematics, Vol. 1, pp. 273-277, 1962.

[Applegate, 2007] D. L. Applegate, R. M. Bixby, V. Chvatal, W. J. Cook, "The Travelling Salesman Problem", Princeton Series in Applied Mathematics, Princeton University Press, New Jersey, USA, 2007

[Bellman, 1962] R. Bellman, "Dynamic Programming Treatment of the Travelling Salesman Problem", Journal of the Association for Computing Machinery, Vol. 9, Issue 1, pp.61-63, New-York, NY, USA, 1962.

[Johnson, 1997] D. S. Johnson, L. A. McGeoch, "The Traveling Salesman Problem: A Case Study in Local Optimization", Local Search in Combinatorial Optimisation, E. H. L. Aarts and J. K. Lenstra (Eds.), John Wiley and Sons Ltd, London, UK, 1997.

[Gutin, 2002] G. Gutin, A. Yeo and A. Zverovich, "Traveling salesman should not be greedy: domination analysis of greedy-type heuristics for the TSP", Discrete Applied Mathematics, Vol. 117, Issues 1-3, pp.81-86, 2002.

[Dorigo, 2010] M. Dorigo and T. Stutzle "Ant Colony Optimization: Overview and Recent Advances", M. Gendreau and Y. Potvin (Eds.), Handbook of Metaheuristics, Vol. 146, International Series in Operations Research & Management Science, pp.227-263, Springer-Verlag New York, 2010.

[Bollobas, 1998] B. Bollobas, "Modern Graph Theory", Graduate Texts in Mathematics, Vol. 184, Springer-Verlag New York, p.350, 1998.

[Cameron, 1994] P. J. Cameron, "Combinatorics: Topics, Techniques, Algorithms", Cambridge University Press, Cambridge, UK, p.163, 1994.

[Dijkstra, 1959] E. W. Dijkstra, "A note on two problems in connexion with graphs", Numerische Mathematik, Vol. 1, pp.269-271, 1959.

[Hart, 1968] P. E. Hart, N. J. Nilsson and B. Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths", IEEE Transactions on Systems Science and Cybernetics, Vol. 4, Issue 2, 1968.

[Aigner, 2014] M. Aigner and G. M. Ziegler, "Proofs from THE BOOK", Springer-Verlag Berlin Heidelberg, 2014.

[Shimon, 2011] E. Shimon, "Graph Algorithms", G. Even (Ed.), Cambridge University Press, Cambridge, UK, 2011.

[Eppstein, 1996] D. Eppstein, "Spanning trees and spanners", Department of Information and Computer Science, University of California, Irvine, CA, USA, 1996.

[Graham, 1985] R. L. Graham and P. Hell, "On the History of the Minimum Spanning Tree Problem", Annals of the History of Computing, Vol. 7, No. 1, 1985.

[Nesetril, 2001] J. Nesetril, E. Milkova and H. Nesetrilova, "Otakar Boruvka on minimum spanning tree problem: translation of both the 1926 papers, comments, history", Discrete Mathematics, Vol. 223, Issue 1-3, pp.3-36, 2001.

[Prim, 1957] R. C. Prim, "Shortest connection networks and some generalizations", Alcatel-Lucent, The Bell System Technical Journal, Vol. 36, Issue 6, 1957.

[Kruskal, 1956] J. B. Kruskal, “On the shortest spanning subtree of a graph and the traveling salesman problem”, Proceedings of the American Mathematical Society, Vol. 7, pp.48-50, 1956.

[MDSJ, 2009] Algorithmics Group. MDSJ: Java Library for Multidimensional Scaling (Version 0.2). Available at http://www.inf-uni-konstanz.de/algo/software/mdsj/. University of Konstanz, 2009.