ADVANCED SOUND HYBRIDIZATIONS BY MEANS OF THE THEORY OF SOUND-TYPES

Carmine Emanuele Cella
IRCAM, Paris, France
carmine-emanuele.cella@ircam.fr

Juan José Burred
Paris, France
jjburred@jjburred.com

ABSTRACT

In this article some advanced methods for sound hybridizations by means of the theory of sound-types will be shown. After a short presentation of the theory, a formal definition will be given. A framework implementing the theory will be presented in detail, with an emphasis on the modules aimed at the sound transformations. Hybridization is achieved by means of two orthogonal processes called type matching (aimed at timbral transformation) and probability merging (aimed at the imitation of the temporal morphology). Finally, some relevant artistic applications of the proposed methods will be discussed.

1. INTRODUCTION

The theory of sound-types is a framework for sound analysis and synthesis designed to represent and manipulate signals at a quasi-symbolic level. While at its origin representational aspects were more prominent (the whole theory has its roots in a logical system called *simple type theory*), recent developments biased towards more musical and creative outcomes.

This paper is divided into two main parts: part one (Sect. 2) will provide a short summary of the fundamental concepts of this theory and will give a mathematical definition of the principal tools involved; part two (Sect. 3) will discuss some possibilities regarding advanced methods for sound hybridization and other transformations that are possible with the sound-types.

There are some connections between this theory and other approaches such as Audioguide [4] and CataRT [7], or work related to sound texture synthesis [6], but a detailed review of such analogies is out of the scope of this paper. Important differences between sound-types and other approaches are that with sound-types a full analysis-synthesis process is performed and also a probabilistic generation of new sounds is possible. For a full presentation of the theory of sound-types and its fundamental properties please refer to [3] and [2].

2. THE THEORY OF SOUND-TYPES

The theory of sound-types has been designed generic enough to be used for different kinds of signals, and should have **signal-dependent semantics**, should be **scalable**, **weakly invertible** and **generative**. Shortly: signal-dependent semantics means that the domain of the representation is inferred from the input; scalability means that it is possible to change the degree of *abstraction*; weakly invertible means that perceptually relevant parts can be reconstructed while not being waveform-identical to the input and generative means that it is possible to randomly generate output sounds different than the input.

The basic idea is to describe sounds by means of *types* (classes of equivalence) and *rules* (probabilities). Conceptually, the analysis is implemented by these steps:

1. **atomize**: divide a sound in small overlapping chunks called *atoms*. This can be done by windowing or more complex techniques such as atomic decomposition, onset-based segmentation, etc.;

2. **make classes**: compute a set of low-level features for each atom and project it onto a feature space; apply a clustering algorithm to find clusters of atoms in the space;

3. **compute probabilities**: apply any kind of sequential analysis (such as a Markov model) to estimate the probabilities that a cluster is followed by another one in the original signal.

From a purely theoretical standpoint, all the ideas presented above are based on a particular equation called the *sound-types transform*, introduced next.

2.1. The sound-types transform

Given a signal $\overset{N}{\vec{x}}$ of length N samples and a window $\overset{n}{\vec{h}}$ of length n samples, it is possible to define an **atom** as a windowed chunk of the signal of length n samples:

$$\overset{n}{\vec{a}}=\overset{n}{\vec{h}} \cdot \overset{n}{\vec{x}} \tag{1}$$

where the operator $\cdot$ is a multiplication *element-by-element*. Using an adequate hop size t during the analysis stage (for example $t \leq n/4$), it is possible to reconstruct a *perfect*[1] version $\overset{N}{\vec{x}'}$ of the original signal with a sum of

[1] As with the STFT, the reconstruction can be perfect only under special conditions (not detailed here) deriving from the type of window used and from the overlapping factor.

atoms as a function of time[2]:

$$\vec{\vec{x}}'^{N} = \sum_{i=0}^{N/t} \vec{\vec{a}}^{n}_{i \cdot t} \tag{2}$$

where N/t is the total number of atoms present in the signal $\vec{\vec{x}}^{N}$. It is possible, after the computation of a set of low-level features on each atom of $\vec{\vec{a}}^{n}_{i}$, to define a **sound-cluster** as a set of atoms that *lie* in a defined area of the feature-space (i.e. that share a *similar* set of features):

$$\vec{\vec{c}}^{k_r}_{r} = \{\vec{\vec{a}}^{n}_{r,1}, \ldots, \vec{\vec{a}}^{n}_{r,k_r}\}. \tag{3}$$

The content of $\vec{\vec{c}}^{k_r}_{r}$ is given by a statistical analysis applied on the feature-space that decides the position of each sound-cluster and its belonging atoms.

A **model** $\mathcal{M}_{\vec{\vec{x}}^{N}}$ of the signal $\vec{\vec{x}}^{N}$ is the defined as the set of the clusters discovered on it:

$$\mathcal{M}_{\vec{\vec{x}}^{N}} = \{\vec{\vec{c}}^{k_1}_{1}, \ldots, \vec{\vec{c}}^{k_r}_{r}\}. \tag{4}$$

The cardinality $|\mathcal{M}_{\vec{\vec{x}}^{N}}|$ of the model is also called the **abstraction level** of the analysis; since the number of atoms is N/t it is evident that $1 \leq |\mathcal{M}_{\vec{\vec{x}}^{N}}| \leq N/t$ with the highest abstraction being 1 and the lowest abstraction being N/t.

Each sound-cluster in the feature-space has an associate **sound-type** $\vec{\vec{\tau}}^{n}_{r}$ in the signal-space, defined as the weighted sum of all the atoms in the sound-cluster where the weights $\vec{\vec{\omega}}^{k_r}_{r}$ are the distances (or any kind of Bregman's divergences) of each atom to the center of the cluster:

$$\vec{\vec{\tau}}^{n}_{r} = \sum_{j=1}^{k_r} \vec{\vec{a}}^{n}_{r,j} \cdot \omega_{r,j} \tag{5}$$

with $\omega_{r,j} \in \vec{\vec{\omega}}^{k_r}_{r}$. The whole set of sound-types in the signal $\vec{\vec{x}}^{N}$ is called **dictionary** and is the equivalent, in the signal-space, of the model in the feature-space:

$$\mathcal{D}_{\vec{\vec{x}}^{N}} = \{\vec{\vec{\tau}}^{n}_{1}, \ldots, \vec{\vec{\tau}}^{n}_{r}\}. \tag{6}$$

The creation of a sound-type from a sound-cluster is also called *collapsing* and can be indicated with the symbol $\langle \vec{\vec{c}}^{k_r}_{r} \rangle = \vec{\vec{\tau}}^{n}_{r}$: this operation represents an interesting connection between the feature-space and the signal-space that leads to the equivalence $\langle \mathcal{M}_{\vec{\vec{x}}^{N}} \rangle = \mathcal{D}_{\vec{\vec{x}}^{N}}$.

It is possible to define a function Ψ that maps an atom to its corresponding sound-type as:

$$\Psi_{\vec{\vec{a}}^{n}_{i}} : \vec{\vec{a}}^{n}_{i} \rightarrow \langle \vec{\vec{c}}^{k_r}_{r} \rangle. \tag{7}$$

[2]The positions in time of the blocks of n-samples are given by an index i that counts the number of hops (i.e. $i = 4 \implies 4 \cdot t$).

For a complete decomposition of the signal, it is also useful to define a function Θ that returns the original time position of each atom:

$$\Theta_{\vec{\vec{a}}^{n}_{i}} : \vec{\vec{a}}^{n}_{i} \rightarrow i. \tag{8}$$

It is now possible to define the **sound-types decomposition** $\vec{\vec{x}}''^{N}$ of a signal by *replacing* each atom of equation 2 with the corresponding sound-type defined through Ψ, in the right time position given by Θ:

$$\vec{\vec{x}}''^{N} = \sum_{i=0}^{N/t} \vec{\vec{\tau}}^{n}_{r,p} \tag{9}$$

where $p = \Theta_{\vec{\vec{a}}^{n}_{i}}$. Finally, it is possible to define a function of time and frequency by multiplying the sound-types in a given dictionary with complex sinusoids:

$$\vec{\vec{\Phi}}^{N}_{\vec{k}^{n}} = \sum_{i=0}^{N/t} \vec{\vec{\tau}}^{n}_{r,p} \cdot e^{-j \cdot \frac{2 \cdot \pi}{n} \cdot \vec{\vec{k}}^{n}} \tag{10}$$

where $\vec{\vec{k}}^{n} = \{f_1, \ldots, f_n\}$ is a vector of frequencies. Eq. 10 is called the forward **sound-types transform** (STT); the inverse transform can recreate the sound-types decomposition and is given by:

$$\vec{\vec{x}}''^{N} = \frac{1}{n} \sum_{i=0}^{N/t} \vec{\vec{\Phi}}^{n}_{i \cdot t, \vec{\vec{k}}^{n}} \cdot e^{j \cdot \frac{2 \cdot \pi}{n} \cdot \vec{\vec{k}}^{n}}. \tag{11}$$

It should be noted that the term "transform" is used in a wide sense here. The transform operation in the STT does not only consist on the multiplication with the complex exponential bases, as Eq. 10 could suggest, but should be interpreted instead as including also the sound-type mapping operator of Eq. 7 and the index mapping operator of Eq. 8. Alternatively, the STT could be interpreted as an STFT in which each windowed signal segment has been replaced by its corresponding sound-type.

2.2. STT and STFT

The usual way to mathematically define the discrete short-time Fourier transform (STFT) $\vec{\vec{X}}^{N}_{\vec{k}^{n}}$ of a signal $\vec{\vec{x}}^{N}$ of length N-samples taken n at a time while hopping by t-samples, is a function of both time and frequency:

$$\vec{\vec{X}}^{N}_{\vec{k}^{n}} = \sum_{i=0}^{N/t} \vec{\vec{x}}^{N} \cdot \vec{\vec{h}}^{n}_{i \cdot t} \cdot e^{-j \cdot \frac{2 \cdot \pi}{n} \cdot \vec{\vec{k}}^{n}} \tag{12}$$

where $\vec{\vec{h}}^{n}$ is a window of length n-samples.

Eqs. 10 and 12 have a strong resemblance. As observed in the previous section, the abstraction level of a model can be at most equal to the number of atoms (N/t) in the original signal. The extreme case for $|\mathcal{M}| = N/t$ is interesting: for that abstraction level, each sound-cluster

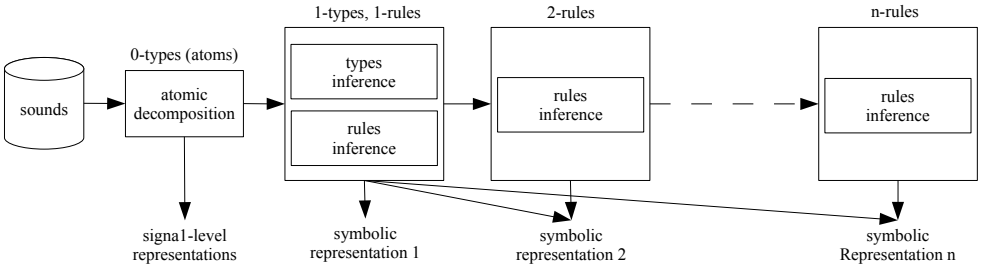

Figure 1. An outline of the proposed algorithm for types and rules inference

is a singleton made of a single atom and consequently each sound-type reduces to that single atom scaled in amplitude:

$$|\mathcal{M}| = N/t \implies \vec{c}_r = \{\vec{a}_1\} \implies \vec{\tau}_r = \vec{a}_r \,. \tag{13}$$

From Eq. 1, an atom is defined simply as a windowed chunk of the original signal; this also makes the sound-types decomposition $\vec{x}''$ equivalent to the simple decomposition $\vec{x}'$, leading to the important interpretation of the STT as a generalization of the STFT:

$$\vec{\tau}_r = \vec{a}_r = \vec{h} \cdot \vec{x} \implies \sum_{i=0}^{N/t} \vec{\tau}_{r,p} \cdot e^{-j \cdot \frac{2 \cdot \pi}{n} \cdot \vec{k}} = \sum_{i=0}^{N/t} \vec{h} \cdot \vec{x}_{i \cdot t} \cdot e^{-j \cdot \frac{2 \cdot \pi}{n} \cdot \vec{k}} \tag{14}$$

with p defined as above. This property also holds for the inverse transform. The abstraction level of a model is directly connected to the *goodness* of the representation: the higher the abstraction (closer to 1) the more compact the representation. On the contrary, the quality of the synthesis given by the inverse transform degrades with high abstractions and increases with low abstractions becoming a perfect reconstruction for $|\mathcal{M}| = N/t$.

3. SOUND HYBRIDIZATIONS

The next subsections will describe the individual components of the current implementation of the theory of sound-types. It should be pointed out that this is just a specific realization of the principles discussed above; the use of other signal processing and machine learning techniques is also possible.

Sections 3.1, 3.2 and 3.3 will describe, respectively, the analysis, sound-type synthesis and rebuild/generation modules, which form the core of the system and can be used for either the analysis and generation of individual sounds, or to separately analyze two sounds (one source and one target) for the hybridization. Sections 3.4 and 3.5 (sound-types matching and probability merging) will address the new modules, specifically aimed at the hybridization of two sounds. Finally, 3.6 will discuss some improvements introduced into the resynthesis module.

3.1. The analysis stage

A twofold process, aimed at discovering the sound-types and their associated rules (transition probabilities), is at the core of the creation of a verifiable model for the proposed theory. The following procedure shows a possible implementation of such process, using low-level features and statistical learning for types inference and Markov models for rules inference:

1. **atoms creation**: create chunks of approximately 40 ms of sound, called *atoms* or *0-types*, overlapping in time and frequency; these atoms can be produced either by overlapping windows, by onset-based segmentation or by other approaches such as adaptive atomic decompositions;

2. **1-types inference**: compute a set of low-level features on each atom obtained in the previous step, project the features onto a multi-dimensional space and compute the *clusters* by means of unsupervised learning; each cluster will represent a *1-type*;

3. **1-rules inference**: estimate a Markov model to describe the sequence of types present in the original sound;

4. **1-level representation**: represent the sound in a symbolic language using the discovered 1-types and 1-rules;

5. **n-rules inference**: estimate a Markov model of order n to describe the sequences of 1-types;

6. **n-level representation**: represent the sound in a symbolic language using the discovered 1-types and n-rules;

7. **repeat n-level rules and n-level representations**: until the desired level number has been reached.

Figure 1 illustrates the present approach. It is an evolution of the original formulation in [3], which contained a full abstraction hierarchy, not only of rules but also of types (which were re-estimated at each level), but which lacked real-time capabilities. The present version is thus more oriented to musical performance.

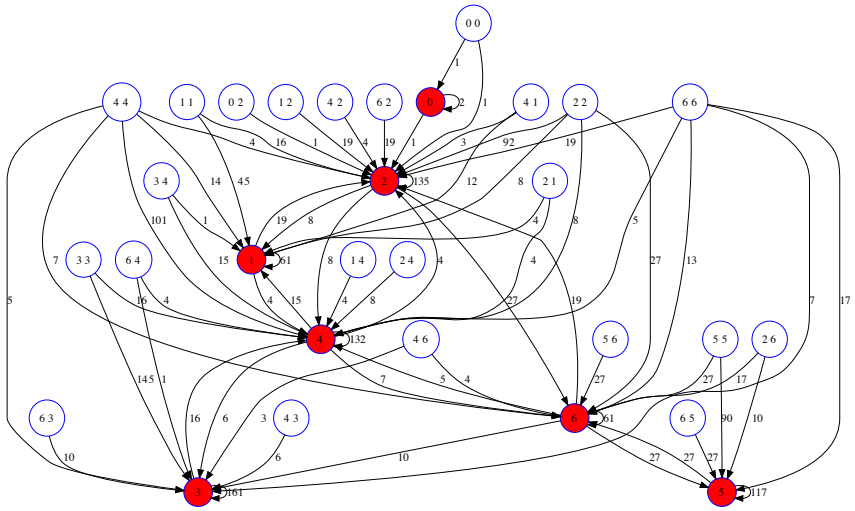

Figure 2. Illustration of estimated transitions, using 2 different levels and 7 sound-types (in red, 1-types).

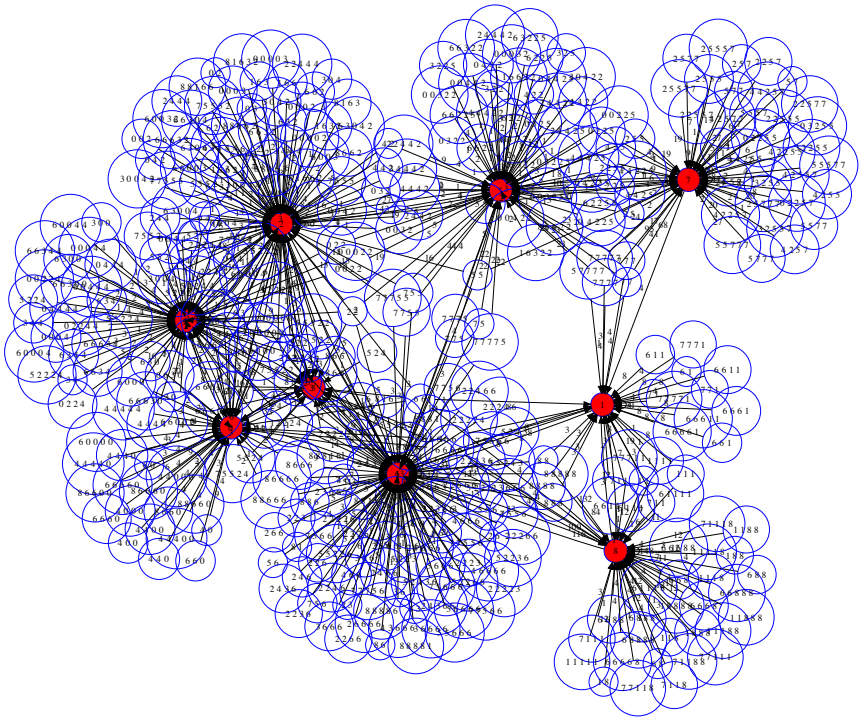

Figure 3. Illustration of estimated transitions, using 5 different levels and 9 sound-types (in red, 1-types).

As mentioned above, the 1-type inference is implemented by the extraction of low-level features with subsequent clustering. Several well-known features from the field of content analysis and music information retrieval have been implemented, including spectral centroid, spectral spread, zero crossings rate, Mel-Frequency Cepstral Coefficients and estimated fundamental frequency. The choice of features will obviously affect the timbre of the generated sound, but preliminary experiments were performed to pre-select a satisfactory subset. If the feature dimensionality is high (all features are unidimensional apart from the mel coefficients, which are 12-dimensional), an optional Principal Component Analysis stage helps reducing the subsequent computational requirements and ensuring that the dimensions in feature space are uncorrelated.

Concerning the clustering, two different popular methods were implemented: k-means and Gaussian Mixture Models. Both are closely related, but the first searches for the clusters by an iterative partition of the space, and the second assumes that each cluster is described by a multivariate Gaussian distribution and estimates the cluster assignment in a probabilistic way.

The n-rule inference at level n is performed by estimating a Markov transition matrix of order n on the original sequence of 1-types. Figures 2 and 3 show graphical representations of two examples of estimated transitions. Fig. 2 shows a 2nd order matrix. The sound-types correspond to the red dots, and the bigrams (subsequences of 2 states) correspond to the blue circles. The edges are labeled by the transition probabilities (showed as absolute frequencies). Fig. 3 corresponds to a level-5 estimation. Note that in this case, the representation corresponds to the combination of all transition matrices estimated up to level 5, as can be seen from the presence of bigrams, 4-grams and 5-grams.

3.2. Sound-type synthesis

By the definition of Eq. 5, a sound-type is generated as the weighted sum of the atoms belonging to its associated cluster, where the weighting is related to the distance of each atom to the cluster's centroid[3]. This is the basic synthesis method, but several other approaches have been investigated, namely:

- **mean:** the atom waveforms are simply averaged;
- **witness:** the sound-type equals the atom whose feature is closest to the centroid;
- **random:** the sound-type equals a randomly selected atom from the cluster according to a given probability distribution;

The choice of synthesis method has a crucial effect on the sound output and will depend on the task to be accomplished. For instance, for highly non-stationary signals (such as voice), the witness or random methods usually provide better results, while the summation-based methods are more suitable for slower-evolving sounds. While the overall quality of the reconstructed signal also depends strongly on the used abstraction level, it is nonetheless difficult to objectively provide a quality measure for it.

3.3. Rebuild and probabilistic generation

Once an input sound has been subjected to analysis, and a type and rule inference has been performed, the obtained sound-type dictionaries and transition matrices can be used to generate new output sounds in two ways:

- **Rebuild.** A state sequence is generated by observing the original input atoms and assigning each one to its closest sound-type. Then, each input atom is replaced by its corresponding sound-type. This is in effect the direct implementation of the STT of Eq. 10. The end effect is an output signal similar to the input, but whose atoms have been "timbrally quantized" into the dictionary of sound-types. Note that in this case, the transition matrix is not used.
- **Probabilistic generation.** The estimated transition matrix is used to generate a random sequence of states. Each generated symbol will then be replaced by its corresponding sound-type, and the result will be a signal with a complete new temporal structure, but with local temporal evolutions recognizable from the input signal. The granularity of the recognizable temporal events will be determined by the Markov order chosen for the analysis. Some generation constraints have been included to avoid repetitions and loops.

3.4. Sound-types matching

An important new extension to the sound-types framework is the possibility to hybridize two sounds in terms of timbral and temporal characteristics. It is possible to subject two different sounds to separate types and rules inferences, and then impose or merge one sound's types or rules with the others'. We consider here two hybridization methods: *sound-types matching*, which will be introduced in this section, and *probability merging*, which will be the subject of the next section.

In *sound-types matching*, the sound-types inferred from a signal (the *source*) are replaced by, or merged with, the sound-types inferred from a *target* signal. Each sound-type from the source is matched with a sound-type from the target, in terms of a similarity measure between the centroids of their corresponding feature clusters. Available similarity measures include the Euclidean, Mahalanobis and Manhattan distances, the cosine similarity, and the symmetrized Kullback-Leibler divergence.

Once each source sound-type has been matched to a target sound-type, an output sound can be generated by observing the original type sequence of the source signal and performing one of the following operations:

- **Replacement.** The source types are replaced by the target types. In the lowest-abstraction case in which clusters are one-atom singletons, this corresponds to corpus-based concatenative synthesis [7] (in that context, matching is called *unit selection* or *audio mosaicing*).
- **Multiplicative cross-synthesis.** Source and target types are mixed together in the frequency domain, as described by the following equations:

$$A_o = \sqrt{A_s * A_t}, \quad \phi_o = (1-\alpha)\phi_s + \alpha\phi_t \quad (15)$$

where A represents an amplitude spectrum, ϕ a phase spectrum and α is the amount of hybridization.

[3] The centroid of a cluster is its multidimensional average, i.e., a point in its geometrical center. It should not be confused with the *spectral* centroid, one of the low-level features used in the analysis stage.

- **Source-filter cross-synthesis.** The source types are replaced by the target types after imposing the spectral envelope of one sound on the flattened spectrum of another. This process may be summarized as follows:

  1. compute the STT of source and target;
  2. compute the spectral envelope of each sound-type;
  3. flatten the spectrum of the source signal dividing it by its own spectral envelope (Sect. 3.6);
  4. multiply the flattened spectral frame by the envelope of the corresponding target frame.

- **Morphing.** Source and target types are interpolated together in the frequency domain, as described by the following equations:

$$A_o = A_s + (A_t - A_s) * \alpha, \quad \phi_o = \phi_s(\phi_t/\phi_s)^\alpha \tag{16}$$

  where A, ϕ and α are defined as above.

Note that in sound-type matching, the rules (transition probabilities) of neither sound are taken into account, since the output type sequence is fully determined by the input type sequence. In other words, the instantaneous timbres change, but the temporality is imposed by the source.

3.5. Probability merging

As a second, more experimental hybridization method, *probability merging* aims at combining the transition probability matrices of source and target sounds. In contrast to sound-type matching, probability merging enables the automatic generation of random type sequences whose temporality is partially influenced by either source or target signal, or by both of them at the same time.

In order to merge two probability matrices, the types are again matched to each other in terms of feature similarity, so probability merging has always an implicit type matching. The matrices are rearranged (and possibly resized) so that their columns and rows are aligned in terms of matched clusters. Then, they are added with a linear weight factor α to obtain the merged probability matrix. I.e., if the rearranged source and target matrices are, respectively $\mathbf{T}_S$ and $\mathbf{T}_T$, then the merged matrix is

$$\mathbf{T}_M = \alpha\mathbf{T}_S + (1-\alpha)\mathbf{T}_T. \tag{17}$$

Fig. 4 illustrates many of the ideas involved in both sound-type matching and probability merging. The red dots denote the feature vectors (only two dimensions are retained for the plot) corresponding to the source signal, the black dots correspond to the atoms of the target signal. The inferred clusters are denoted by the centroids (marked by squares) surrounded by ellipses that correspond to Gaussian unit-variance contours. We remind that the centroids in feature space correspond to the inferred sound-types.

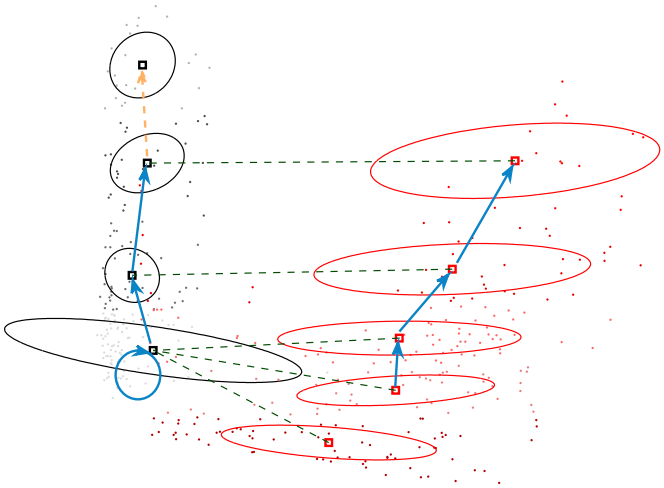

Figure 4. Illustration of sound-type matching and probability merging. See text for details.

Thus, the dashed lines linking the source and target centroids indicate the matching (closest) sound-types. The arrows denote transitions between sound-types. The blue arrows on the source signal illustrate a typical sub-sequence between source types, which will likely induce the sub-sequence indicated by the blue arrows on the corresponding matched types of the target signal. With probability merging, some transitions between target types might still be possible, even if the concerned types do not have a match with the source types. This later case is illustrated by the dashed, orange arrow in the upper left part.

3.6. Frequency domain processing

In order to achieve a good quality in the synthesis process, sound-types matching uses frequency-domain techniques for both phases and amplitudes. First, *phase locking* can be applied in order to improve the vertical coherence of the resynthesized signal. Second, *envelope preservation* can be applied in order to maintain the main morphology of a sound after the operations of pitch-shifting and cross-synthesis. Here is a summary of these operations:

- **Phase locking.** When a signal is analyzed by the Discrete Fourier Transform (DFT), each component of the signal falls into a specific channel k of the transformed domain (Eq. 12) and has a specific phase. Intuitively, if the component changes frequency between one frame and the other, it is necessary to handle its phase in order to preserve coherence in time. One of the best approaches to preserve phase coherence in time was proposed in [5] and is related to the estimation of the peaks in the magnitude spectrum. The basic idea is to create an entity that preserves phase coherence for each frequency analyzed called *phasor*. The algorithm to apply phase locking is outlined below; the steps are only intuitively described:

1. for each magnitude spectral frame $\vec{X}_i^k$ compute the positions of the peaks (peak-map);
2. for each peak k_l in the peak-map calculate its true analysis frequency ω_a, then map this to the true synthesis frequency and synthesis phase; calculate the phasor $z_{k_l} = e^{j\phi}$;
3. for each k calculate the synthesis frame $\vec{Y}_i^k = z_{k_l} \cdot \vec{X}_i^k$.

- **Spectral envelope preservation.** When applying pitch-shifting with the phase-vocoder the spectral envelope will necessarily be also transposed. This leads to unnatural sounds that, sometimes, are very different from the original ones. To avoid this, the spectral envelope has to be kept constant, while the partials *slide* along it to their new position in frequency. Two simple methods for envelope computation are: interpolation between the peaks and the use of the *cepstrum*. The cepstrum is calculated from the DFT by taking the inverse transform of the magnitude of its logarithm:

$$c_p = \frac{1}{K} \sum_{k=0} log(|\ \vec{X}_k\ |) \cdot e^{j \cdot \frac{2 \cdot \pi}{K} \cdot k \cdot p} \tag{18}$$

where $\vec{X}_k$ is the DFT of the signal and p is the number of coefficients used in the transformation. The spectral envelope is then computed by applying a lowpass window to the cepstrum (called *liftering*) and by taking again the DFT:

$$E = DFT(W_{LP}(c_p)) \tag{19}$$

where W_{LP} is the lowpass window.

4. POSSIBLE OUTCOMES

While the theory of sound-types has been conceived in the context of symbolic representations of signals, previous sections showed powerful capabilities for creative applications by means of sound hybridizations.

As mentioned before, the theory represents a given sound (or family of sounds) in terms of classes of equivalences and transition probabilities between them. In other words, it finds *salient elements* that are representative of a sound and recreates that sound (or generates new sounds) using only these essential elements. Each type is the fundamental acoustic element shared by many real instances of it: as in Plato's epistemological view, here a type is a sort of pure sonic idea able to generate an infinite number of concrete instances. The theory is also generative: it is in fact possible to create new sounds by merging discovered sound-types with discovered probabilities, thus creating something intimately *linked* to the original material.

In the context of an artistic project, the theory of sound-types could be an appropriate tool to render musical and poetic ideas and could be also an innovative approach to sound synthesis and transformation. Among possible applications there are: **time and frequency transformations** (such as time-stretch and pitch-shift with formants preservation), **probabilistic generation** (creation of *affine* sounds to imitate temporal morphology), **generalized sound hybridizations** (types matching, probabilities merging and various cross-synthesis methods).

These capabilities are currently under investigation by the authors, especially in the context of artistic creation: part ot the theory is implemented as offline processing tools, while other parts are working in real time. Some audio examples of the described methods are available for listening[4].

Nonetheless, it must be noted that the term "real time" is used in a wide sense here. The statistical learning and the hierarchical structure of sound-types only become meaningful if the analysis is performed on a significative signal length (in the range of seconds, not samples). For this reason it is more appropriate to consider sound-types as a *relaxed real time tool*. It is important to point out again, however, that the proposed algorithm is only a *possible* realization of a general idea (see [2] and [1]).

4.1. The piece *Reflets de l'ombre*

The original idea behind the theory of sound-types is related to the process of knowledge creation: a type is a sort of **idealized** image of a concrete sound. The poetic contrast between these two worlds, real sounds and pure sonic-ideas, has been investigated in *Reflets de l'ombre* for large orchestra and electronics by Carmine Emanuele Cella premiered in june 2013 by the Orchestre Philarmonique de Radio France and conducted by Jukka-Pekka Saraste; the electronics was based on sound-types analysis, synthesis and generation.

5. CONCLUSIONS AND PERSPECTIVES

The theory of sound-types is still in an early stage of development and needs expansions and improvements both at the symbolic-level and at the signal-processing level. It is not totally clear, moreover, the potential of the theory in terms of *artistic* applications. The following list focuses on possible relevant research directions, roughly depicted in figure 5.

1. **Selective transformations.** Since the representation provided by sound-types is based on a generalized version of the STFT it should be possible to transform a sound working only on *selected elements*. For example, it should be possible to perform pitch-shift or time-stretch only on types that satisfy certain conditions in the feature-space (e.g. only the types that have a spectral centroid close to a given value and spectral spread close to another given value and so on). This selection could be also

[4] www.soundtypes.com

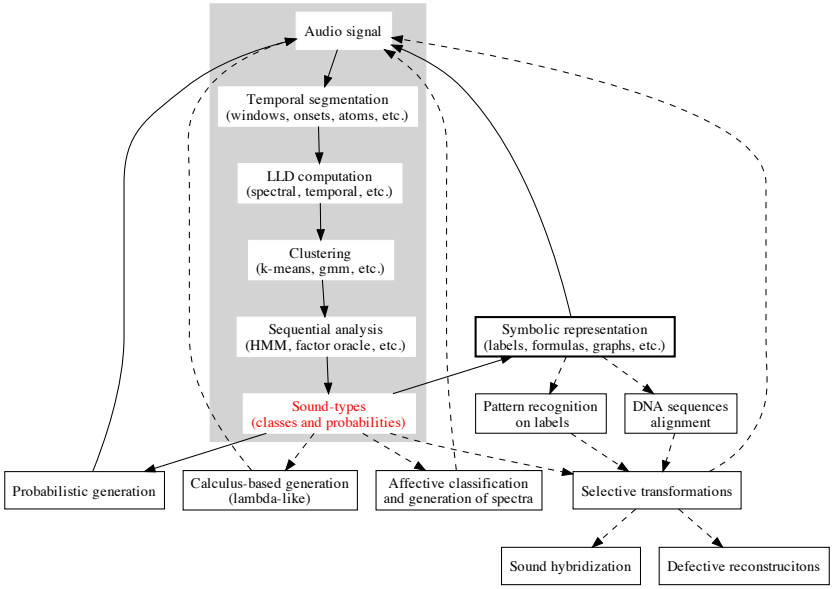

Figure 5. An overview of some possible expansions of the theory of sound-types; the dashed lines represent missing or incomplete parts.

applied to extract some given types from the original sound, in order to perform *semantic source synthesis*. It could also be interesting to create *defective* reconstructions of a signal using, for example, only some types over the whole set discovered. This kind of transformation could produce a sound *similar* to the original, but different in many aspects.

2. **Affective classification and generation of spectra.** While most modern systems for sound analysis and synthesis are centered on low-level techniques and have a low degree of abstraction, the theory of sound-types tries to provide high level instruments for sound manipulation. For this reason, it could be possible to use this representation method in order to classify and generate sounds on an *affective* basis. When a sound has been represented in terms of sound-types and probabilities, a supervised labeling could be applied on the discovered elements in order to classify them by means an affective model: some types could be called, for example, *rough* or *sad*. It could be possible, then, to ask the machine to generate similar sounds by means of the discovered probabilities. This approach could lead to an innovative interaction model in which artists could focus more on their mood than on the parameters of the machine.

Sound-types seem to be promising entities to represent music because they are physically related to sound, are invertible and are also capable to represent formal relationships and hierarchies.

6. REFERENCES

[1] C. E. Cella, "Towards a symbolic approach to sound analysis," in *Second international conference on Mathematics and computation for music*, Yale University, New Haven, CT, 2009.

[2] ——, *On symbolic representations of music*. PhD Thesis, University of Bologna, Italy, 2011.

[3] ——, "Sound-types: a new framework for symbolic sound analysis and synthesis," in *Proc. ICMC*, Huddersfield, UK, 2011.

[4] B. Hackbarth, N. Schnell, and D. Schwarz, "Audioguide: a framework for creative exploration of concatenative sound synthesis," in *Research report, IRCAM*, Paris, France, 2010.

[5] J. Laroche and M. Dolson, "New phase-vocoder techniques for real-time pitch shifting, chorusing, harmonizing, and other exotic audio modifications," in *Journal of the Audio Engineering Society*, vol. vol. 47, 1999.

[6] D. Schwarz, "State of the art in sound texture synthesis," in *Proc. DAFX*, Paris, France, 2011.

[7] D. Schwarz, R. Cahen, and S. Britton, "Principles and applications of interactive corpus-based contatenative synthesis," in *Proc. Journées d'Informatique Musicale*, Albi, France, 2008.