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.

1.1. The sound-types transform

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

\[
\vec{a} = \vec{h} \cdot \vec{x}
\]

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 \( \hat{\vec{x}} \) of the original signal with a sum of

\[
\sum_{a} \left( \vec{a} \right) = \sum_{a} \left( \vec{h} \cdot \vec{x} \right) = \vec{x}
\]

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.

---

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.

1.1. The sound-types transform

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

\[
\vec{a} = \vec{h} \cdot \vec{x}
\]

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 \( \hat{\vec{x}} \) of the original signal with a sum of

\[
\sum_{a} \left( \vec{a} \right) = \sum_{a} \left( \vec{h} \cdot \vec{x} \right) = \vec{x}
\]

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.

---

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.

1.1. The sound-types transform

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

\[
\vec{a} = \vec{h} \cdot \vec{x}
\]

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 \( \hat{\vec{x}} \) of the original signal with a sum of

\[
\sum_{a} \left( \vec{a} \right) = \sum_{a} \left( \vec{h} \cdot \vec{x} \right) = \vec{x}
\]

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.

---

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.

1.1. The sound-types transform

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

\[
\vec{a} = \vec{h} \cdot \vec{x}
\]

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 \( \hat{\vec{x}} \) of the original signal with a sum of

\[
\sum_{a} \left( \vec{a} \right) = \sum_{a} \left( \vec{h} \cdot \vec{x} \right) = \vec{x}
\]

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:

\[ x_n(t) = \sum_{t=0}^{N} a_n e^{j w_n t} \]  

(2)

where \( N \) is the total number of atoms present in the signal \( x \). It is possible, after the computation of a set of low-level features on each atom of \( a_n \), 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):

\[ \tilde{c}_r = \{a_{i_1}, \ldots, a_{i_k}\} \]  

(3)

The content of \( \tilde{c}_r \) 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 \( \mathcal{M}_N \) of the signal \( x \) is defined as the set of the clusters discovered on it:

\[ \mathcal{M}_N = \{\tilde{c}_r\} \]  

(4)

The cardinality \( |\mathcal{M}_N| \) of the model is also called the abstraction level of the analysis; since the number of atoms is \( N/t \) it is evident that \( 1 \leq |\mathcal{M}_N| \leq N/t \) with the highest abstraction being 1 and the lowest abstraction being \( N/t \).

Each sound-cluster in the feature-space has an associated sound-type \( \tilde{r} \) in the signal-space, defined as the weighted sum of all the atoms in the sound-cluster where the weights \( \omega_{n,j} \) are the distances (or any kind of Bregman’s divergences) of each atom to the center of the cluster:

\[ \tilde{r} = \sum_{n,j} \omega_{n,j} a_{n,j} \]  

\( \omega_{n,j} \in [0,1] \). The whole set of sound-types in the signal \( x \) is called dictionary and is the equivalent, in the signal-space, of the model in the feature-space:

\[ \mathcal{D}_x = \{\tilde{r}_{i_1}, \ldots, \tilde{r}_{i_k}\} \]  

(6)

The creation of a sound-type from a sound-cluster is also called collapsing and can be indicated with the symbol \( \tilde{r} \rightarrow \tilde{r} \). This operation represents an interesting connection between the feature-space and the signal-space that leads to the equivalence \( (\mathcal{M}_N, \mathcal{D}_x) = (\mathcal{M}_N, \mathcal{D}_x) \).

It is possible to define a function \( \Theta \) that maps an atom to its corresponding sound-type as:

\[ \Theta_n \rightarrow \tilde{r}_n \]  

(7)

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

\[ \Phi_n \rightarrow x_n(t) \]  

(8)

It is now possible to define the sound-types decomposition \( \tilde{x}_r \) of a signal by replacing each atom of equation 2 with the corresponding sound-type defined through \( \Phi \), in the right time position given by \( \Theta \):

\[ \tilde{x}_r = \sum_{n=0}^{N} a_n e^{j w_n t} \]  

(9)

where \( r = \Theta_n \). Finally, it is possible to define a function of time and frequency by multiplying the sound-types in a given dictionary with complex sinusoids:

\[ \tilde{\Phi}_r (\tilde{\Omega}_r) = \sum_{n=0}^{N} a_n e^{j w_n t} \]  

(10)

where \( \tilde{\Omega}_r \) 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:

\[ \tilde{x}_r = \sum_{n=0}^{N} a_n e^{j w_n t} \]  

(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 STT in which each windowed 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) \( X_r \) of a signal \( x \) of length \( N \) samples taken at \( n \) at time while hopping by 1-samples, is a function of both time and frequency:

\[ X_r(t) = \sum_{n=0}^{N-t} x_n e^{-j 2\pi \frac{n}{N}} \]  

(12)

where \( \delta \) 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 \( |\mathcal{M}_N| = N/t \) is interesting: for that abstraction level, each sound-cluster is a singleton made of a single atom and consequently each sound-type reduces to that single atom scaled in amplitude:

\[ |\mathcal{M}_N| = N/t \Rightarrow \tilde{r} = \{a_i\} \Rightarrow \tilde{r} = a_i \]  

(13)

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

\[ \tilde{x}_r = \sum_{n} \tilde{r}_n e^{-j 2\pi \frac{n}{N}} \]  

(14)

with \( \tilde{r} \) as defined 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 the 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 \( |\mathcal{M}_N| = N/t \).

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.
atoms as a function of time:\(^\frac{N}{t}\) 
\[
\tilde{\psi}_i = \sum_{j=0}^{N/t} \tilde{a}_{j,i}
\]  
(2) 
where \(N/t\) is the total number of atoms present in the signal \(\tilde{\psi}\). It is possible, after the computation of a set of low-level features on each atom of \(\tilde{\psi}\), 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): 
\[
\tilde{\psi}_i = \{\tilde{a}_1, \ldots, \tilde{a}_{N/t}\}.
\]  
(3) 
The content of \(\tilde{\psi}_i\) 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 \(M_{\Omega}\) of the signal \(\tilde{\psi}\) is defined as the set of the clusters discovered on it: 
\[
M_{\Omega} = \{\tilde{\psi}_1, \ldots, \tilde{\psi}_i\}.
\]  
(4) 
The cardinality |\(M_{\Omega}\)| of the model is also called the abstraction level of the analysis; since the number of atoms is \(N/t\) it is evident that \(1 \leq |M_{\Omega}| \leq N/t\) with the highest abstraction being \(1\) and the lowest abstraction being \(N/t\). 

Each sound-cluster in the feature-space has an associated sound-type \(\tilde{\tau}_i\) in the signal-space, defined as the weighted sum of all the atoms in the sound-cluster where the weights \(\tilde{\omega}_i\) are the distances (or any kind of Bregman’s divergences) of each atom to the center of the cluster: 
\[
\tilde{\omega}_i = \sum_{j=0}^{N/t} \tilde{a}_{j,i} \omega_{i,j}
\]  
(5) 
with \(\omega_{i,j} \leq 1\). The whole set of sound-types in the signal \(\tilde{\psi}\) is called dictionary and is the equivalent, in the signal-space, of the model in the feature-space: 
\[
D_{\Omega} = \{\tilde{\tau}_1, \ldots, \tilde{\tau}_i\}.
\]  
(6) 
The creation of a sound-type from a sound-cluster is also called collapsing and can be indicated with the symbol \(\hat{\tilde{\psi}}_i\). This operation represents an interesting connection between the feature-space and the signal-space that leads to the equivalence \((M_{\Omega}, D_{\Omega})\). 

It is possible to define a function \(\Theta\) that maps an atom to its corresponding sound-type as: 
\[
\Theta: \tilde{a}_i \mapsto \tilde{\tau}_i.
\]  
(7) 
For a complete decomposition of the signal, it is also useful to define a function \(\Theta\) that returns the original time position of each atom: 
\[
\Theta: \tilde{a}_i \mapsto i.
\]  
(8) 
It is now possible to define the sound-types decomposition \(\tilde{\psi}_\Theta\) of a signal by replacing each atom of equation 2 with the corresponding sound-type defined through \(\Phi\), in the right time position given by \(\Theta\):

\[
\tilde{\psi}_\Theta = \tilde{\psi} = \sum_{i=0}^{N/t} \tilde{a}_{i} \cdot \tilde{\psi}_i
\]  
(9) 
where \(p = \Theta_i\). Finally, it is possible to define a function of time and frequency by multiplying the sound-types in a given dictionary with complex sinusoids:

\[
\Phi_k = \sum_{i=0}^{N/t} \tilde{a}_{i} \omega_{i,k} e^{-i \tilde{x} \cdot \tilde{k}}
\]  
(10) 
where \(\tilde{x} = \{f_1, \ldots, f_s\}\) 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:

\[
\tilde{a}_i = \frac{1}{N} \Phi_k e^{i \tilde{x} \cdot \tilde{k}}
\]  
(11) 

It should be noted that the term “transform” is used in a wide sense here. The transformation operator 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 STT in which each windowed segment has been replaced by its corresponding sound-type. 

2.2. STT and STF

The usual way to mathematically define the discrete short-time Fourier transform (STFT) \(X_n\) of a signal \(\tilde{\psi}\) of length \(N\)-samples taken \(n\) at a time while hopping by \(t\)-samples, is a function of both time and frequency:

\[
X_n = \tilde{\psi} = \sum_{i=0}^{N/t} \tilde{a}_{i} e^{-i \tilde{x} \cdot \tilde{k}}
\]  
(12) 
where \(\tilde{x}\) 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 \(|M_{\Omega}| = N/t\) is interesting: for that abstraction level, each sound-cluster is a singleton made of a single atom and consequently each sound-type reduces to that single atom scaled in amplitude:

\[
|M_{\Omega}| = N/t \Rightarrow \tilde{\omega}_i = \{\tilde{a}_1\} \Rightarrow \tilde{\omega}_i = \tilde{\omega}_i.
\]  
(13) 

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

\[
\tilde{x} = \tilde{a}_i = \tilde{\psi} = \sum_{i=0}^{N/t} \tilde{a}_{i} e^{-i \tilde{x} \cdot \tilde{k}}
\]  
(14) 
with \(\tilde{x}\) 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 \(|M_{\Omega}| = N/t\). 

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.
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 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 corresponding to the red dots, and the bigrams (subsequences of 2 states) correspond to the blue circles. The edges are labeled by the transition probabilities (shown 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. 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 sound types are assigned 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 the two different sounds to separate types and rules inferred, 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:
  \[ A_s = \sqrt{A_t \cdot A_i} \quad \phi_s = (1 - \alpha)\phi_s + \alpha\phi_t \]  
  where \( A \) represents an amplitude spectrum, \( \phi \) a phase spectrum and \( \alpha \) is the amount of hybridization.

As mentioned above, the 1-type inference is implemented by the low-level features with subsequent clustering. Several well-known features from the fields of content analysis and music information retrieval have been implemented, including spectral centroid, spectral spread, this section and 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.

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).
The choice of features will obviously affect the timbral and spectral characteristics of the generated sounds. Several well-known features from the field of content analysis and music information retrieval have been implemented, including spectral centroid, spectral spread, Mel-frequency Cepstral Coefficients, and estimated fundamental frequency. The choice of features will obviously affect the timbral and spectral representation of the sound.

As mentioned above, the 1-type inference is implemented by the lowest-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, Mel-frequency Cepstral Coefficients, and estimated fundamental frequency. The choice of features will obviously affect the timbral and spectral characteristics of the generated sounds. Several well-known features from the field of content analysis and music information retrieval have been implemented, including spectral centroid, spectral spread, Mel-frequency Cepstral Coefficients, and estimated fundamental frequency. The choice of features will obviously affect the timbral and spectral representation of the sound.

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 multi-variate Gaussian distribution and estimates the cluster assignment in a probabilistic way.

The n-rule inference at level i 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 (shown 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. 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

Where $A_t$ represents an amplitude spectrum, $\phi$ a phase spectrum, and $\alpha$ the amount of hybridization.
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 feature domain, as described by the following equations:

\[ A_s = A_t + (A_t - A_s) \cdot \alpha, \quad \phi_s = \phi_t \cdot (\phi_t / \phi_s)^\alpha \]

where \( A, \phi \) and \( \alpha \) 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 timbre 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 correspondence. The matrices are again matched to each other in terms of feature similarity and type. In the case 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 morphological properties 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 frequency band and the other, it is necessary to handle these 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 phase. The algorithm to apply phase locking is outlined below; the steps are only intuitively described:

1. for each magnitude spectral frame \( X_i \) compute the positions of the peaks (peak-map);
2. for each peak \( k \) in the peak-map calculate its analysis frequency \( \omega_k \), then map this to the true synthesis frequency and synthesis phase; calculate the phase \( z_k = e^{j \omega_k} \)
3. for each \( k \) calculate the synthesis frame \( Y_{i,k} = X_{i,k} \cdot z_k \)

• 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:

\[ c_p = \frac{1}{\pi} \sum_{k=1}^{K} \log| \tilde{X}_k | \cdot e^{j \frac{\pi}{K} k p} \]

where \( \tilde{X}_k \) 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 filtering) and by taking again the DFT:

\[ E = DFT(W_P(c_p)) \]

where \( W_P \) 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 music or generate new sounds. Some 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 of 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.

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 Philharmonique 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

www.soundtypes.com
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:

\[ A_s = A_s + (A_t - A_s) \alpha \phi_s = \phi_s(\phi_s/\phi_t)^\alpha \]

where \( A_s \), \( A_t \), and \( \alpha \) 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 rescaled) so that their columns and rows are aligned in terms of matched clusters. Then, they are added with a linear weight factor \( \alpha \) to obtain the merged probability matrix.

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.

3.6. Frequency domain processing
In order to achieve a good quality in the synthesis process, sound-type 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 morphing 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 \( X_s^k \) compute the positions of the peaks (peak-map);
2. For each peak \( l \) in the peak-map calculate its analysis frequency \( \omega_{kl} \), then map this to the true frequency synthesis and synthesis phase; calculate the phasor \( z_{kl} = e^{j \omega_{kl}} \);
3. For each \( k \) calculate the synthesis frame \( X_t^k \) = \( X_s^k \).

- **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:

\[ c_p = \frac{1}{K} \sum_{k=0}^{K-1} \log || X_s^k || e^{j \frac{2 \pi}{K} k} \]

where \( X_s^k \) 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 filterbank) and by taking again the DFT:

\[ E = DFT(W_L(c_p)) \]

where \( W_L \) 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 is possible to perform pitch-shifting and cross-synthesis with formants preservation, probabilities merging and various cross-synthesis methods.

These capabilities are currently under investigation by the authors, especially in the context of artistic creation: part of 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.

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 Celli 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

www.soundtypes.com
Sound-types seem to be promising entities to represent the machine. It could be possible, then, to ask the machine to classify and generate sounds on an affective basis. This paper explains the context for the project and its goals, and discusses some of the generic software that is being developed as part of the project, intended not only for use in the project itself but also to be freely available for others to use in the study of any electroacoustic work as appropriate.

1. INTRODUCTION

The evolution of electroacoustic music, from the pioneering post-Second World War era to the present day has resulted in a rich and varied legacy of works. As time advances the need to secure these compositions for posterity comes ever more pressing, no more so than in the case of the ones originally mastered on analog tape. Even in the digital domain, older formats such as DAT have become obsolete, adding further urgency to the task of copying, and if necessary re-mastering. But this mission, as important as it is, only achieves the first task of copying, and if necessary re-mastering. But this mission, as important as it is, only achieves the first stage in the quest to facilitate knowledge and understanding of the repertoire that has emerged.

Whereas instrumental composers provide access to the inner detail of their works through the medium of the score, using a notation that is universally understood, electroacoustic composers rarely provide clues to the underlying compositional process other than those that may be deduced from the aural experience. Although works that combine electronic materials with instrumental music sometimes provide descriptive clues via special scores provided for the players, these rarely provide any more than elemental cueing information to assist synchronisation in performance. This detachment from both the compositional processes and the technologies used in their production presents major challenges for those wishing to gain a greater insight into the realisation of these works. These challenges materialise with the passage of time as the memories of the composers become more distant and the technologies used become progressively harder to access. When the composer dies these processes of further investigation become even harder to pursue.

The importance of addressing these issues has been recognised in some quarters, but it is only relatively recently that the wider electroacoustic community has begun to grasp the true scale of what needs to be done here as a matter of increasing urgency. In terms of historical perspective a number of texts provide useful insights into the evolution of the medium, for example Chadabe [6], Holmes [14] and more recently Manning [19]. In so doing the scope and nature of the associated technologies are usefully contextualised, but for the most part such accounts do not embark on detailed analyses of individual works. Other texts have concentrated more specifically on the technical methods that have been employed over the years, for example Appleton [1], Naumann and Wagnerer [20], Roads [22], and Puckette [21], and there are further texts that focus on analytical considerations, for example Bennett and Bariatre [4], Camilleri and Smalley [5], Lican [15], Roy [23], and Simoni [24]. Further context-specific contributions to the associated literature include Clarke [8], [10], [11], [12], Clarke and Manning [9], Dufeu [13], and Manning [16], [17], [18].

In Dufeu [13], the author has discussed electroacoustic organology by emphasizing the status of the digital instrument as an object and possibly a tool for the analysis of music including real-time processes. Following theoretical ideas of Battier [2] and practical approaches of Baudouin [3] to software reconstructions as an analytical method, the musical instrument itself is described as a possible workspace for the musicologist. Indeed, the code constituting the instrument may hold important information on the processes, structures, and modes of interaction characterizing the work. Not only can this code be observed to trace the relevant data, it can as well and under certain conditions be rewritten and enhanced to enable measurements of, for instance, the processes ongoing from a gestural input to the sound output. The software may also be a basis for the implementation of ways of representation which may not be needed for the performance itself but prove useful for analytical purposes.

The above writings have made significant contributions to our understanding of the provenance of the associated repertoire, and indeed the analytical texts that provide perhaps the greatest insight into the issues identified above. However, so far, they have all faced a common problem, how to present analyses that are based primarily on sonic information in the form of...