On symbolic representations of music

The theory of sound-types

Carmine-Emanuele Cella

Università di Bologna/IRCAM Paris

Bologna, 8th july 2011

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

ircam
Centre
Pompidou

On the nature of music (1)

- Music has a dual nature:
  - it is irrational and related to emotions and is difficult to describe it by means of formalized languages;
  - it follows strict systems of rules based on formal reasoning and many of its elements are defined *only* through mathematics.

On the nature of music (1)

- Music has a dual nature:
  - it is irrational and related to emotions and is difficult to describe it by means of formalized languages;
  - it follows strict systems of rules based on formal reasoning and many of its elements are defined *only* through mathematics.
- The dichotomy between a purely mathematical theory and a perceptually-related field is central to the development of musical theory.

On the nature of music (1)

- Music has a dual nature:
  - it is irrational and related to emotions and is difficult to describe it by means of formalized languages;
  - it follows strict systems of rules based on formal reasoning and many of its elements are defined *only* through mathematics.
- The dichotomy between a purely mathematical theory and a perceptually-related field is central to the development of musical theory.
- The interpretation of music as an *art* appeared in late XVI century.

On the nature of music (2)

- Any musical activity, is based on the identification of the relationship between adiacent tones called *interval*.

On the nature of music (2)

- Any musical activity, is based on the identification of the relationship between adiacent tones called *interval*.
- Between all the possible intervals, the *octave* represents two tones that are expressed by the ratio 2 : 1; it is a *class of equivalence* for tones.

On the nature of music (2)

- Any musical activity, is based on the identification of the relationship between adiacent tones called *interval*.
- Between all the possible intervals, the *octave* represents two tones that are expressed by the ratio $2:1$; it is a *class of equivalence* for tones.
- The link between mathematics and music has ancient roots and is based on the definition of intervals.

Maths and music: a short history (1)

Pythagoras

Pythagoras of Samos was probably the first, in VI century B.C., to define a clear connection between mathematics and music: he discovered that the intervals of octave, fifth and fourth can be expressed by simple ratios, namely $2:1$, $3:2$ and $4:3$. All intervals can be expressed, in the Pythagorean system, by means of the *tetractys*.

Maths and music: a short history (2)

Middle ages

The medieval system of musical notation was clearly correlated to the coeval system of measures. Both systems are based on a central value, the *brevis* and the *uncia* respectively, that can be multiplied to obtain bigger values or subdivided to have smaller ones. Moreover, both systems are based on values that can be subdivided by two or by three (binary and ternary subdivisions).

Maths and music: a short history (3)

Jean-Philippe Rameau

In his books *Traité de l'harmonie rèduite à ses principes naturels* (1722) Rameau defined the bases of contemporary harmonic theory. He described, using integer ratios, the structure of major (perfect) chords calling *fundamental* the most important note in a chord. He created *classes of equivalences* for chords, grouping together all chords described by the same ratios, showing interesting links between the rational theory of chords and musical praxis.

Maths and music: a short history (4)

The harmonic series

In XVIII century Joseph Saveur showed that each musical tone was made of several harmonics. Formally, a musical tone can be described by the sum of all integer fractions from 1 to ∞:

$$\sum_{n=1}^{\infty} \frac{1}{n} = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \dots$$

This discovery gave an important physical justification to Rameau's theory and led him, in 1737, to the formulation of the *corps sonore*, the final unification of the abstract theory of ratios and the coeval musical praxis.

Maths and music: a short history (4)

Hugo Riemann

Hugo Riemann, explored the logical relations present inside and between consonant chord, using mathematical logic as an instrument to evaluate the *syntax* of the language. With a dualistic construct made of major and minor chords sharing the fundamental (thus merging the physics of chords with a logical structure) Riemann illustrates the *modularity* and *finiteness* of the major/minor tonalities, organizing all tones in twelve pitch classes.

$e_\sharp/f$	c	g	d	a	e	b	$f_\sharp$	$c_\sharp$	$g_\sharp$	$d_\sharp$	$a_\sharp$
$c_\sharp$	$g_\sharp$	$d_\sharp$	$a_\sharp$	$e_\sharp$	$b_\sharp/c$	g	d	a	e	b	$f_\sharp$
a	e	b	$f_\sharp$	$c_\sharp$	$g_\sharp$	$d_\sharp$	$a_\sharp$	$e_\sharp/f$	c	g	d
f	c	g	d	a	e	b	$f_\sharp$	$c_\sharp$	$g_\sharp$	$d_\sharp$	$a_\sharp$
$d_\flat$	$a_\flat$	$e_\flat$	$b_\flat$	f	c	g	d	a	e	b	$f_\sharp$
a	$e/f_\flat$	$c_\flat$	$g_\flat$	$d_\flat$	$a_\flat$	$e_\flat$	b	f	c	g	d

Maths and music: a short history (5)

$\mathbb{Z}_{12}$ and the compositional perspective

In XX century composers started using *actively* mathematical concepts to create new music. This behaviours is directly linked to *equal temperament*, a new way of representing intervals in which every pair of adjacent elements has the same ratio: the octave (whose ratio is $2:1$) is divided into a series of equal steps (usually 12). Musical intervals and chords can finally be represented as *integers* in a mathematical structure called *group*, denoted by the symbol $\mathbb{Z}_{12}$.

Symbolic representations

- The key aspect of the link between maths and music is *symbolic representation*: musical entities are represented in the formal language of mathematics with symbols.

Symbolic representations

- The key aspect of the link between maths and music is *symbolic representation*: musical entities are represented in the formal language of mathematics with symbols.
- Written music *is not* actual music; only in the final stage of performance music becomes physically real, when it become a *musical signal*.

Symbolic representations

- The key aspect of the link between maths and music is *symbolic representation*: musical entities are represented in the formal language of mathematics with symbols.
- Written music *is not* actual music; only in the final stage of performance music becomes physically real, when it become a *musical signal*.
- The main purpose of this research is to find a **new representation method for music** that takes into account both dimensions of music: the score (symbol) and the sound (signal).

Very general (useless) questions

The research, moreover, focused also on the following questions:

1. Which is the relationship between *mathematical logic* and *musical logic*?

Very general (useless) questions

The research, moreover, focused also on the following questions:

1. Which is the relationship between *mathematical logic* and *musical logic*?
2. Has the formalism based on *musical reasoning* something in common with logic formalism?

Very general (useless) questions

The research, moreover, focused also on the following questions:

1. Which is the relationship between *mathematical logic* and *musical logic*?
2. Has the formalism based on *musical reasoning* something in common with logic formalism?
3. Can mathematical logic be useful to musicians, in order to clarify their reasoning?

A short review

1. **Axiomatic approach** (Langer - 1929);

A short review

1. **Axiomatic approach** (Langer - 1929);
2. **Set-theoretical algebra** (Åqvuist - 1973);

A short review

1. **Axiomatic approach** (Langer - 1929);
2. **Set-theoretical algebra** (Åqvuist - 1973);
3. **Modal logic** (Kunst - 1976);

A short review

1. **Axiomatic approach** (Langer - 1929);
2. **Set-theoretical algebra** (Åqvuist - 1973);
3. **Modal logic** (Kunst - 1976);
4. **Functional languages** (Orlarey, Fober, Letz, Bilton - 1994).

An axiomatic approach

In 1929, Susanne Langer supposed that there are relatively little elements involved in music and that there are only a few possibilities to combine them following definite principles. A set of axioms should represent the abstract form of music (the *logic of music*) and should be able to describe all possible musical situations. The abstract form of music is itself similar to a special algebra, neither numerical or Boolean, but of equally mathematical form and for which there exists at least one interpretation.

The 15 axioms

1. if $a, b \in K$ then $a \cdot b \in K$;
2. $\forall a \in K, a \cdot a = a$;
3. if $a, b \in K$ then $a \to b \in K$;
4. $\forall a, b \in K, a \to b = b \to a \implies a = b$;
5. $\forall a, b, c \in K, (a \cdot b) \cdot c = b \cdot (a \cdot c)$;
6. $\forall a, b, c \in K, \exists d \in K$ such that $(a \to b) \cdot (c \to d) = (a \cdot c) \to (b \cdot d)$;
7. $\exists r \in K$ such that $\forall a \in K, a \cdot r = a$;
8. there is a K-subclass T such that $\forall a, b \in K$ other than r and $\forall c \in K$, if $a = b \cdot c \implies b = c$ and $a = b \to c \implies b = r \vee c = r$ then $a \in T$;
9. $\forall a \in T, C(a \cdot a)$;
10. $\forall a, b, c \in K, \neg C(a \cdot b) \implies \neg C(a \cdot b \cdot c)$;
11. $\forall a \in K$ there exists K-subclass A such that $\forall b, c \in K, b \in A \iff C(a) \cdot b \equiv C(b) \cdot c$;
12. $\forall a, b \in T$ with $a \neq b$, $\neg(a < b) \equiv (b < a)$;
13. $\forall a, b, c \in T, a < b \wedge b < c \implies a < c$;
14. $\forall a, b \in T$ where $b \notin A$ and $\forall a\prime \in A$, $\exists b\prime \in B$ such that if $a < a\prime$ then $\neg(a < b < a\prime) \implies (a < b\prime < a\prime)$;
15. $\forall a \in T, \exists a^\circ \in A$ such that $\forall b \in A$ with $b \neq a, b \neq a^\circ$, $a < b \wedge a < a^\circ \implies a^\circ < b$ and $a < a^\circ \wedge b < a^\circ \implies b < a$.

The interpretation (1)

1. if a and b are musical elements, the *interval* $a \cdot b$ is a musical element;
2. if a is a musical element, it is equal to the *unison* $a \cdot a$;
3. if a and b are musical elements, the *progression* $a \rightarrow b$ is a musical element;
4. if a and b are musical elements and if $a \rightarrow b = b \rightarrow a$ than a and b are the same musical element;
5. if a, b and c are musical elements, then the interval $(a \cdot b) \cdot c = b \cdot (a \cdot c)$.
6. if a, b and c are musical elements, then exists the musical element d such that the interval of progressions $(a \rightarrow b) \cdot (c \rightarrow d)$ is equal to the progression of intervals $(a \cdot c) \rightarrow (b \cdot d)$;
7. there is at least one musical element r such that, if a is a musical element other than r, the interval $a \cdot r = a$;
8. there is a subclass T of musical elements (called *tones*) such that, if a and b are musical elements other than r, c is a musical element and if $(a = b \cdot c) \implies (b = c)$ and $(a = b \rightarrow c) \implies b = r \vee c = r$, then a is a tone (in other words, if a is an interval it is a unison, and if a is a progression every memebr but one is a rest);

The interpretation (2)

9. if a is a tone, the unison $a \cdot a$ is *consonant*;
10. if a, b and c are musical elements and $a \cdot b \cdot c$ is consonant then $a \cdot b$ is consonant;
11. for any musical element a other than r, there is a subclass A (called *recurrences*) such that for any b and c, b is the recurrence of a if and only if ($a \cdot c$ is consonant) is equivalent to ($b \cdot c$ is consonant);
12. if a and b are two different tones and a is not before b in order of pitch, then b is before a;
13. if a, b and c are tones, then if a is before b and b is before c then a is before c;
8. if a and b are distinct tones with b not being recurrence of a and a' is a recurrence of a, then there is at least one b', recurrence of b, such that if a is before a' and b is not between a and a', then b' is between a and a' in order of pitch;
8. For any tone a, there is a tone a°, recurrence of a, such that if b is any other recurrence of a and if a is before b and also a is before a° then a° is before b; and if a is before a° and b is before a°, then b is before a in order of pitch (in other words, there is at least a recurrence of a called *octave* such that no other recurrence can lie between it and a).

Set-theoretical algebra (1)

- The method proposed of by Susanne Langer is focused on the harmonic paramter and does not pay attention to *temporal* evolution.

Set-theoretical algebra (1)

- The method proposed of by Susanne Langer is focused on the harmonic paramter and does not pay attention to *temporal* evolution.
- A *time-limited frame* is defined an ordered quadruple $\langle T, t-, -t, \leq \rangle$ such that $\forall (t, t', t'') \in T$:

T1. $T \neq \emptyset$
T2. $t-, -t \in T$
T3. $t- \neq -t$
T4. $\leq \in T \times T$ (linear ordering)
T5. $t- \leq t$
T6. $t \leq -t$
T7. $t \leq t$ (reflexivity)
T8. if $t \leq t'$ and $t' \leq t''$ then $t \leq t''$ (transitivity)
T9. if $t \leq t'$ and $t' \leq t$ then $t = t'$ (antisymmetry)
T10. $t \leq t'$ or $t' \leq t$.

Set-theoretical algebra (2)

- A *frequency-limited frame* is an ordered quintuple $\langle P, p-, -p, \S, \leq \rangle$ such that $\forall (p, p', p'') \in P$ hold axioms **P1** - **P10** defined in the same way as axioms **T1** - **T10** and also holds:
  P11. $\S \notin P$ (namely, the *null*-frequency).

Set-theoretical algebra (2)

- A *frequency-limited frame* is an ordered quintuple $\langle P, p-, -p, \S, \leq \rangle$ such that $\forall (p, p', p'') \in P$ hold axioms **P1** - **P10** defined in the same way as axioms **T1** - **T10** and also holds:
  P11. $\S \notin P$ (namely, the *null*-frequency).
- A *musical frame* is the structure $\langle \langle T, t-, -t, \leq \rangle, \langle P, p-, -p, \S, \leq \rangle, V \rangle$ where:

  (i). $\langle T, t-, -t, \leq \rangle$ is a time-limited frame;
  (ii). $\langle P, p-, -p, \S, \leq \rangle$ is a frequency-limited frame;
  (iii). $V \neq \oslash$ (namely, the set of *voices*).

Set-theoretical algebra (3)

A *musical frame with voice-indexed temporal partitions* is the structure
$F = \langle\langle T, t-, -t, \leq\rangle, \langle P, p-, -p, \S, \leq\rangle, V, S_v\rangle$ such as:

(i). $\langle\langle T, t-, -t, \leq\rangle, \langle P, p-, -p, \S, \leq\rangle, V\rangle$ is a musical frame;

(ii). S_v is a function from V to the power-set of T (namely, the *point-selector*) such as $\forall v \in V, S_v$ is a finite subset of T and $t-, -t$ exist both in S_v.

With S_v it is possible to define the notions of *temporal segments* and *time interval* $\phi(S_v)$.

Set-theoretical algebra (4)

- On structure F, defined above, it is possible to define the *melodic-rhythmic specification* as the ordered couple of functions $\langle On, Att \rangle$ on V such as $\forall v \in V$, holds: On_v and $Att_v \subseteq T \times (P \cup \{\S\})$.

Set-theoretical algebra (4)

- On structure F, defined above, it is possible to define the *melodic-rhythmic specification* as the ordered couple of functions $\langle On, Att \rangle$ on V such as $\forall v \in V$, holds: On_v and $Att_v \subseteq T \times (P \cup \{\S\})$.
- An *abstract musical system* as the structure $M = \langle F, \langle On_v, Att_v \rangle \rangle$, such as:
  - (i). F is a musical frame with voice-indexed temporal partitions;
  - (ii). $\langle On_v, Att_v \rangle$ is a melodic-rhytmic specification on F.

Set-theoretical algebra (5)

With *musical model*, it's defined ordered triple $\mu = \langle \aleph, W, inst \rangle$ such as:

(i). $\aleph$ is a set of abstract musical systems;
(ii). W is a non-empty set (world) of *possible instantiations*;
(iii). $inst \subseteq \{PerfConc_w(M) : w \in W, M \in \aleph\}$ (i.e. is a selected subset of all perfect concretizations of M in w with $\forall w \in W$, while all $M \in \aleph$ are the selections of the perfect concretizations *really exhibited* in $w \in W$).

A modal approach (1)

- Formalism to model music in a dynamic way (J. Kunst, 1976).

A modal approach (1)

- Formalism to model music in a dynamic way (J. Kunst, 1976).
- Uses modality by means of operators $\Box$ (necessary) and $\diamond$ (possibile).

A modal approach (1)

- Formalism to model music in a dynamic way (J. Kunst, 1976).
- Uses modality by means of operators $\Box$ (necessary) and $\diamond$ (possibile).
- $\Box c$ means "c is always true, in the past and in the present".

A modal approach (1)

- Formalism to model music in a dynamic way (J. Kunst, 1976).
- Uses modality by means of operators $\Box$ (necessary) and $\diamond$ (possibile).
- $\Box c$ means "c is always true, in the past and in the present".
- $\diamond c$ means "c is true at some point in the past or in the present".

A modal approach (2)

- These propositions can refer to any musical object.

A modal approach (2)

- These propositions can refer to any musical object.
- During the musical flux, objects can change their modality (*BivFunc*).

A modal approach (2)

- These propositions can refer to any musical object.
- During the musical flux, objects can change their modality (*BivFunc*).
- Example: $\Box c$ can be used to say "the sound of the violin is necessarily nasal".

A modal approach (2)

- These propositions can refer to any musical object.
- During the musical flux, objects can change their modality (*BivFunc*).
- Example: $\Box c$ can be used to say "the sound of the violin is necessarily nasal".
- There are no evident relationships between "sound of the violin" and the concept of "nasality".

Musical images (1)

- A model proposed by M. Leman in 2002 to relate modal sentences and sounds

Musical images (1)

- A model proposed by M. Leman in 2002 to relate modal sentences and sounds
- *Musical image*: a spatio-temporal representation entity to link a sound with its conceptualization

Musical images (1)

- A model proposed by M. Leman in 2002 to relate modal sentences and sounds
- *Musical image*: a spatio-temporal representation entity to link a sound with its conceptualization
- Images are created by extracting *low-level descriptors* from the sound itself

Musical images (1)

- A model proposed by M. Leman in 2002 to relate modal sentences and sounds
- *Musical image*: a spatio-temporal representation entity to link a sound with its conceptualization
- Images are created by extracting *low-level descriptors* from the sound itself
- To create a perceptual model, musical images must be processed through the so-called *auditory* system

Musical images (2)

- Let $c \rightarrow n$ be "the sound of the clarinet *implies* the nasal timbre".

Musical images (2)

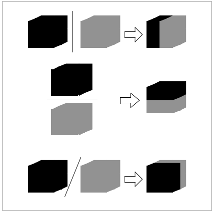
- Let $c \rightarrow n$ be "the sound of the clarinet *implies* the nasal timbre".
- Let $\overrightarrow{A}$ and $\overrightarrow{B}$ be two vectors of low-level features computed from real signals and filtered through the auditory system.

Musical images (2)

- Let $c \rightarrow n$ be "the sound of the clarinet *implies* the nasal timbre".
- Let $\overrightarrow{A}$ and $\overrightarrow{B}$ be two vectors of low-level features computed from real signals and filtered through the auditory system.
- If, after a computational stage, it is possible to transform vector $\overrightarrow{A}$ into vector $\overrightarrow{B}$ then the formal expression $c \rightarrow n$ is said to be *valid*.

A functional language (1)

$$
\begin{aligned}
cube &::= color \mid \; [cube_1 | cube_2] \\
&\quad \mid \; \left[\frac{cube_1}{cube_2}\right] \\
&\quad \mid \; [cube_1 \;/\; cube_2] \\
color &::= white \mid red \mid green \mid invisible \mid \ldots
\end{aligned}
$$



A functional language (2)

A typical sentence is the following:

$$\left[\frac{\mathit{white}|\mathit{invisible}}{\mathit{invisible}/\mathit{white}} / \frac{\mathit{invisible}|\mathit{green}}{\mathit{green}|\mathit{invisible}}\right]$$

A functional language (3)

To extend this language into a calculus we need to add:

- *abstraction* and *application* rules;
- reduction rules to deal with the application of colored cubes.

$$
\begin{aligned}
cube ::= &\ color \\
| &\ [cube_1 | cube_2] \\
| &\ \left[\frac{cube_1}{cube_2}\right] \\
| &\ [cube_1 / cube_2] \\
| &\ \lambda color.cube\ (\textbf{abstraction}) \\
| &\ (cube_1\, cube_2)\ (\textbf{application}) \\
color ::= &\ white \mid red \mid green \mid invisible \mid \ldots
\end{aligned}
$$

A functional language (4)

A β-reduction example

$$\lambda white.\lambda green.\left[\frac{white|invisible}{invisible/white}/\frac{invisible|green}{green|invisible}\right] \ \mathbf{blue} \quad \mathbf{red} \Rightarrow_\beta$$

$$\lambda green.\left[\frac{blue|invisible}{invisible/blue}/\frac{invisible|green}{green|invisible}\right] \ \mathbf{red} \Rightarrow_\beta$$

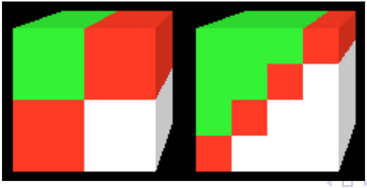
$$\left[\frac{blue|invisible}{invisible/blue}/\frac{invisible|red}{green|red}\right]$$

A functional language (4)

$$A = \left[\frac{green|red}{red|white} \right]$$

$$B = \left[\frac{green|\frac{green|red}{red|white}}{\frac{green|red}{red|white}|white} \right]$$

$$X = \left[\frac{green|X}{X|white} \right]$$



A functional language (5)

$$X = \left(\lambda red.\left[\frac{green|(red\ \ red)}{(red\ \ red)|white}\right]\ \lambda red.\left[\frac{green|(red\ \ red)}{(red\ \ red)|white}\right]\right)$$

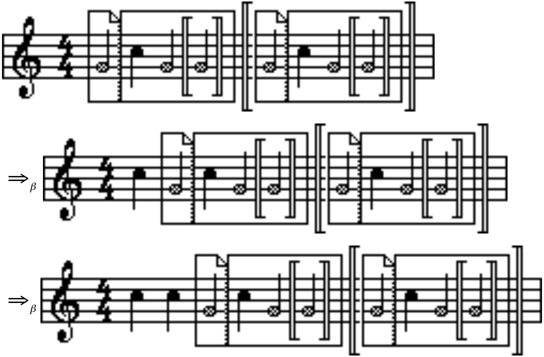
Elody (1)

$$
\begin{aligned}
score &::= \phi \mid event \mid [score_1; score_2] \\
&\mid \left[\frac{score_1}{score_2}\right] \\
&\mid \lambda event.score \mid (score_1\ score_2) \\
event &::= r \mid note \mid event\ tmodifier \\
note &::= pitch \mid pitch\ octave \mid note\ nmodifier \\
pitch &::= c \mid d \mid e \mid f \mid g \mid a \mid b \\
octave &::= 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9 \\
tmodifier &::= . \mid * \mid t \mid / \\
nmodifier &::= + \mid - \mid > \mid <
\end{aligned}
$$

Elody (2)

Elody is a music language extented with lambda-calculus:

$$X \equiv \left(\lambda f.\left[\mathrm{c4};(f \quad f)\right] \; \lambda f.\left[\mathrm{c4};(f \quad f)\right]\right)$$



A short review

1. **Classification of chords** (Babbitt et al. - 1961)

A short review

1. **Classification of chords** (Babbitt et al. - 1961)
2. **Pitch class set theory** (Forte et al. - 1977)

A short review

1. **Classification of chords** (Babbitt et al. - 1961)
2. **Pitch class set theory** (Forte et al. - 1977)
3. **Transformational theory** (Lewin - 1992)

Classification of chords (1)

Counting chords in $\mathbb{Z}_{12}$ is a combinatorial problem on groups, solved by Polya's theorem that states: the sum of weights of G-orbits on Y^X is given by

$$\sum_{u \in G \backslash\backslash Y^X} w(u) = \frac{1}{|G|} \sum_{g \in G} \prod_{k=1}^{|X|} \left(\sum_{y \in Y} h(y)^k \right)$$

where $jk(\bar{g})$ is the number of cycles of length k of the permutation $\bar{g}$ in its decomposition as a product of independent cycles. Depending on the group used, the number of chords (cycle index) is different.

Classification of chords (2)

The cycle index of the cyclic group $\mathcal{C}_n$ is given by the polynomial

$$P_{(\mathcal{C}_n,\mathbb{Z}_n)}(t_1,\ldots,t_n) = \frac{1}{n}\sum_{d|n}\varphi\left(\frac{n}{d}\right)t_{n/d}^{d}$$

where φ is the Euler's totient function .

Classification of chords (3)

The cycle index of the dihedral group $\mathcal{D}_n$ depends on the oddity of n and is given by:

$$P_{(\mathcal{D}_n,\mathbb{Z}_n)} = \begin{cases} \frac{1}{2}P_{(\mathcal{C}_n,\mathbb{Z}_n)} + \frac{1}{2}t_1 t_2^{(n-1)/2} & \text{if } n \text{ is odd} \\ \frac{1}{2}P_{(\mathcal{C}_n,\mathbb{Z}_n)} + \frac{1}{4}t_1^2 t_2^{(n-2)/2} + t_2^{n/2} & \text{if } n \text{ is even.} \end{cases}$$

Classification of chords (4)

The case of the affine group is more complicated since it splits into two distinct cases.
For $n = 2^a$: the cycle index of the affine group $\mathcal{A}_n$ is given by

$$P_{(\mathcal{A}_{2^a}, \mathbb{Z}_{2^a})} = \frac{1}{2^{(2a-1)}} \left(2^{2(a-1)-1} t_{2^a} + \sum_{i=1}^{a-1} \left(2^{2(i-1)} + \varphi(2^{i-1}) 2^{a-1} \right) t_{2^i}^{2^{a-i}} + \sum_{i=0}^{a-2} \varphi(2^i) \left(2^i t_1^{2^{a-i}} + 2^{a-1} t_1^2 t_2^{2^{a-i-1}-1} \right) \left(\prod_{k=1}^{i} t_{2^k} \right)^{2^{a-i-1}} \right)$$

for $a \geq 3$.

Classification of chords (5)

For $n = p^a$, with p prime and $a \geq 1$: the cycle index of the affine group $\mathcal{A}_n$ is given by

$$P_{(\mathcal{A}_{p^a}, \mathbb{Z}_{p^a})} = \frac{1}{p^{2a-1}(p-1)} + \left(\sum_{i=1}^{a} p^{2(i-1)}(p-1) t_{p^i}^{p^{a-i}} + \sum_{i=0}^{a-1} \sum_{d|p-1} p^{i+\delta(d)(a-i)} \varphi(p^i d) t_1 t_d^{(p^{a-i-1}-1)/d} \left(\prod_{k=1}^{i} t_{p^k d} \right)^{p^{a-i-1}(p-1)/d} \right)$$

where $\delta(x) = 1$ if $x > 1$ and $\delta(1) = 0$.

Classification of chords (5)

Table: The total number of assemblies for any k in $\mathbb{Z}_{12}$.

	k-chords	Group $\mathcal{A}_n$	Group $\mathcal{C}_n$	Group $\mathcal{D}_n$
1	Unison	1	1	1
2	Intervals	5	6	6
3	Trichords	9	19	12
4	Tetrachords	21	43	29
5	Pentachords	25	66	38
6	Hexachords	34	80	50
7	Eptachords	25	66	38
8	Octochords	21	43	29
9	Ennachords	9	19	12
10	Decachords	5	6	6
11	Endecachords	1	1	1
12	Dodecachords	1	1	1
	Total	157	351	223

Pitch class sets theory (1)

Pitch class sets

- It's called *pitch class (pc)* an integer number in $\mathbb{Z}_{12} = \{0, 1, \ldots, 11\}$ representing a note in a tempered-tuning system made of 12 steps, with the following association: $C = 0, C\sharp = 1, \ldots, B = 11$.

Pitch class sets theory (1)

Pitch class sets

- It's called *pitch class (pc)* an integer number in $\mathbb{Z}_{12} = \{0, 1, \dots, 11\}$ representing a note in a tempered-tuning system made of 12 steps, with the following association: $C = 0, C\sharp = 1, \dots, B = 11$.
- A pitch class set is in *normal order* when, arranged in ascending order, it's also put in the more compact form by a cyclic permutation. Formally: let $A = \{A_0, A_1, \dots, A_{k-1}\}$ be a pcset.

Pitch class sets theory (1)

Pitch class sets

- It's called *pitch class (pc)* an integer number in $\mathbb{Z}_{12} = \{0, 1, \ldots, 11\}$ representing a note in a tempered-tuning system made of 12 steps, with the following association: $C = 0, C\sharp = 1, \ldots, B = 11$.
- A pitch class set is in *normal order* when, arranged in ascending order, it's also put in the more compact form by a cyclic permutation. Formally: let $A = \{A_0, A_1, \ldots, A_{k-1}\}$ be a pcset.
- A pcset is in *prime form* (dihedral or Forte's prime forme) if its first integer is 0 and it's the most compact form among its inversion.

Pitch class sets theory (2)

Intervallic content

- It's called *interval class* (ic) of two pcsets the function d that maps $\mathbb{Z}_{12} \times \mathbb{Z}_{12} \longrightarrow \{0, 1, 2, 3, 4, 5, 6\}$ as follows:

$$d(x, y) = \begin{cases} |x - y| \bmod 12 & \text{if } |x - y| < 6 \\ -|x - y| \bmod 12 & \text{if } |x - y| \geq 6. \end{cases}$$

Pitch class sets theory (2)

Intervallic content

- It's called *interval class* (ic) of two pcsets the function d that maps $\mathbb{Z}_{12} \times \mathbb{Z}_{12} \longrightarrow \{0, 1, 2, 3, 4, 5, 6\}$ as follows:

$$d(x, y) = \begin{cases} |x - y| \bmod 12 & \text{if } |x - y| < 6 \\ -|x - y| \bmod 12 & \text{if } |x - y| \geq 6. \end{cases}$$

- The *interval vector* (iv) of a pcset A is a 6-tuple representing all individual interval classes present in A. The first entry counts the number of the smallest interval, the second entry counts the second smallest, etc.

Pitch class sets theory (2)

Intervallic content

- It's called *interval class* (ic) of two pcsets the function d that maps $\mathbb{Z}_{12} \times \mathbb{Z}_{12} \longrightarrow \{0, 1, 2, 3, 4, 5, 6\}$ as follows:

$$d(x, y) = \begin{cases} |x - y| \bmod 12 & \text{if } |x - y| < 6 \\ -|x - y| \bmod 12 & \text{if } |x - y| \geq 6. \end{cases}$$

- The *interval vector* (iv) of a pcset A is a 6-tuple representing all individual interval classes present in A. The first entry counts the number of the smallest interval, the second entry counts the second smallest, etc.
- Two pcsets A, B are said to be Z-related if they have the same interval vector ($iv(A) = iv(B)$) but they are not reducible to the same prime form.

Pitch class sets theory (3)

Similarity

- Two pcsets A and B of the same cardinality m are *p-similar* ($\sim_p$) if there exists at least a common subset of cardinality $m-1$ in the union of two representatives $\bar{A}$ and $\bar{B}$:

$$A \sim_p B \iff \exists C \quad |C| = m - 1, C \subset \bar{A} \cup \bar{B}.$$

Pitch class sets theory (3)

Similarity

- Two pcsets A and B of the same cardinality m are *p-similar* ($\sim_p$) if there exists at least a common subset of cardinality $m-1$ in the union of two representatives $\bar{A}$ and $\bar{B}$:

$$A \sim_p B \iff \exists C \quad |C| = m-1, C \subset \bar{A} \cup \bar{B}.$$

- Two pcsets of the same cardinality will be 0-similar if they have no equal values in the corresponding entries of the interval vector:

$$A \sim_0 B \iff \forall i, \; iv(A)_i \neq iv(B)_i$$

Transformational theory

Given a set X with finite elements and a multiplicative group G of intervals on X, it's possible to define a *generalized interval system* as a triple (X, G, int) where int is the function $X \times X \rightarrow G$ such as:

- $int(x, y) \circ int(y, z) = int(x, z) \;\; \forall x, y, z \in X$;
- $\forall x \in X, \forall g \in G$ there exists a single value $y \in X$ such that $int(x, y) = g$.

Transformational theory

- With 1_X is defined the *characteristic function* of the non-empty pcset X, such that:

$$1_X(u) = \begin{cases} 1 & \text{if } u \in X \\ 0 & \text{otherwise.} \end{cases}$$

Transformational theory

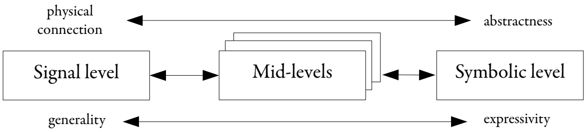
- With 1_X is defined the *characteristic function* of the non-empty pcset X, such that:

$$1_X(u) = \begin{cases} 1 & \text{if } u \in X \\ 0 & \text{otherwise.} \end{cases}$$

- The *interval function* $ifunc_{(X,Y)}$ for two non-empty pcsets X and Y is defined as the convolution of the characteristic functions $1_X^* \star 1_X$:

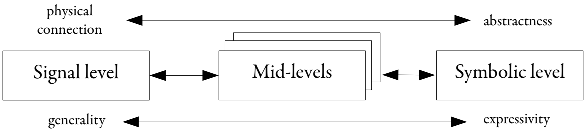
$$ifunc_{(X,Y)}(i) = \sum_j 1_X^*(j) \cdot 1_Y(i-j) = \sum_k 1_X(k) \cdot 1_Y(i+k).$$

Different representations (1)



- Music, in its final stage of *performance*, can be described in many ways (time-varying signal, symbolic system, etc.).

Different representations (1)



- Music, in its final stage of *performance*, can be described in many ways (time-varying signal, symbolic system, etc.).
- Each approach selects a particular degree of abstraction: the *signal level*, the *symbolic level*, a fixed mixture of both (*mid-levels*).

Different representations (2)

- The signal level is efficient and invertible but has a low degree of abstraction.

Different representations (2)

- The signal level is efficient and invertible but has a low degree of abstraction.
- The symbolic level can define complex relationships between objects but is not invertible and is not *physical*.

Different representations (2)

- The signal level is efficient and invertible but has a low degree of abstraction.
- The symbolic level can define complex relationships between objects but is not invertible and is not *physical*.
- Mid-levels are based on perceptual criteria related to hearing and are in between lower and higher levels.

Different representations (2)

- The signal level is efficient and invertible but has a low degree of abstraction.
- The symbolic level can define complex relationships between objects but is not invertible and is not *physical*.
- Mid-levels are based on perceptual criteria related to hearing and are in between lower and higher levels.
- They have a *fixed degree of abstraction*.

Different representations (2)

- The signal level is efficient and invertible but has a low degree of abstraction.
- The symbolic level can define complex relationships between objects but is not invertible and is not *physical*.
- Mid-levels are based on perceptual criteria related to hearing and are in between lower and higher levels.
- They have a *fixed degree of abstraction*.
- They impose *their own* concepts onto the signal.

Variable abstraction representation (1)

This research aims at creating a representation method for music (the *theory of sound-types*) that fulfills *by-design* the following requirements:

- *Signal-dependent semantics*: the involved concepts of the representation should be inferred from the signal.

Variable abstraction representation (1)

This research aims at creating a representation method for music (the *theory of sound-types*) that fulfills *by-design* the following requirements:

- *Signal-dependent semantics*: the involved concepts of the representation should be inferred from the signal.
- *Scalability*: it should be possible to change the *degree of abstraction* in the representation, ranging from the signal level to the symbolic level.

Variable abstraction representation (2)

- *Weak invertibility*: the representation method should be able to generate the represented signal; the generated signal must not be *waveform*-identical to the original one.

Variable abstraction representation (2)

- *Weak invertibility*: the representation method should be able to generate the represented signal; the generated signal must not be *waveform*-identical to the original one.
- *Generativity*: is the possibility to generate sounds *other* than the original one, according to some parameters in the domain of the representation.

Basic signal models (1)

The decomposition of a signal $x[n]$ into expansion functions is a linear combination of the form:

$$\overset{n}{\vec{x}} = \sum_{k=1}^{K} \alpha_k \, \overset{n}{\vec{g}}_k \; .$$

The coefficients α_k are derived from the analysis stage, while the functions $g_k[n]$ can be determined by the analysis stage or fixed beforehand.

Basic signal models (2)

An example of signal decomposition is given by the Discrete Fourier transform (DFT):

$$\vec{X}_k = \sum_{i=0}^{n-1} x_i \cdot e^{-j \cdot \frac{2 \cdot \pi}{n} \cdot k}.$$

Time-frequency decompositions of signals of N samples are also possibile, such as the Short-time Fourier transform (STFT), taking n samples at a time and hopping by t samples:

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

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

Features and classification (1)

- Low-level features: numerical values describing the contents of a signal according to different kinds of inspection: temporal, spectral, perceptual, etc.

Features and classification (1)

- Low-level features: numerical values describing the contents of a signal according to different kinds of inspection: temporal, spectral, perceptual, etc.
- They are computed on a small time scale which is usually between 40 ms and 80 ms; different kinds of temporal modeling (like mean and variance computation) can then be applied.

Features and classification (2)

- Examples: spectral shape (centroid, spread, skewness, kurtosis).

$$\mu = \int x \cdot p(x) dx.$$

$$\sigma^2 = \int (x - \mu)^2 \cdot p(x) dx.$$

Here x are the observed data (i.e. the frequencies of the spectrum) while $p(x)$ are the probabilites to observe x (i.e. the amplitudes of the spectrum).

Features and classification (3)

- Audio indexing: sound classification method based on the projection of low-level features over a set of sounds in a multi-dimensional space (*feature space*).

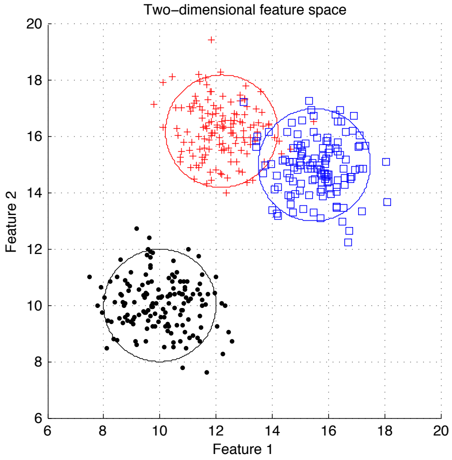
Features and classification (3)

- Audio indexing: sound classification method based on the projection of low-level features over a set of sounds in a multi-dimensional space (*feature space*).
- By analysing the space with some combined geometrical and statistical techniques (like K-means, Gaussian Mixture Models, Principal Component Analysis, etc.) it is possible to find the clusters of sounds present in the space.

Features and classification (3)

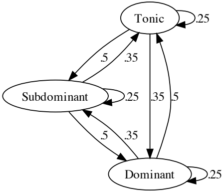
- Audio indexing: sound classification method based on the projection of low-level features over a set of sounds in a multi-dimensional space (*feature space*).
- By analysing the space with some combined geometrical and statistical techniques (like K-means, Gaussian Mixture Models, Principal Component Analysis, etc.) it is possible to find the clusters of sounds present in the space.
- With specific techniques, such as the BIC measure or the gap-statistic, it is possible to *evaluate* the computed clustering.

Features and classification (4)



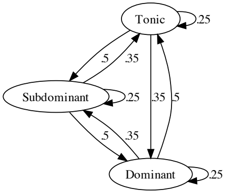
Markov models

- A *Markov model* is a stochastic model represented by a directed graph that can have infinite loops.



Markov models

- A *Markov model* is a stochastic model represented by a directed graph that can have infinite loops.
- The edges of the graph are labelled with *transition probabilities* such that the sum of outgoing probabilities from a single node is 1. Probabilities can be represented as a *transition matrix* T in which each element T_{ij} is the probability of moving from state i to state j.



Sound-types: basic ideas (1)

1. A theory to represent sounds by means of *types* and *rules* inferred by some low-level descriptions of signals and subsequent learning stages.

Sound-types: basic ideas (1)

1. A theory to represent sounds by means of *types* and *rules* inferred by some low-level descriptions of signals and subsequent learning stages.
2. It takes inspiration from the *simple type theory* and from λ-calculus.
3. The types represent classes of equivalences for sounds, while the rules represent transition probabilities that a type is followed by another type.

Sound-types: basic ideas (2)

Mathematically, this means to *translate* the original equation:

$$
\begin{aligned}
x[n] &= \sum_{k=1}^{K} \alpha_k g_k[n] \\
&= \alpha_1 g_1[n] + \ldots + \alpha_k g_k[n] \\
&= \beta_1 f_1[n] + \ldots + \beta_j f_j[n] \\
&\vdots \\
&= \omega_1 h_1[n] + \ldots + \omega_t h_t[n]
\end{aligned}
$$

where $\alpha, \beta, \ldots, \omega$ are any kind of weighting coefficients, $g_k, f_j, \ldots, h_t$ are variables belonging to different *types*. All the theory presented above can be realized by defining the **sound-types transform** (STT).

The sound-types transform (1)

- 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:

$$\overset{n}{\vec{a}} = \overset{n}{\vec{h}} \cdot \overset{n}{\vec{x}} \, .$$

The sound-types transform (1)

- 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:

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

- A **sound-cluster** as a set of atoms that *lie* in a defined area of the feature space (ie. that share a *similar* set of features):

$$\overset{k_r}{\vec{c}_r} = \{\overset{n}{\vec{a}}_{r,1}, \ldots, \overset{n}{\vec{a}}_{r,k_r}\}.$$

The content of $\overset{k_r}{\vec{c}_r}$ is given by a statistical analysis applied on the feature space.

The sound-types transform (2)

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

$$\mathcal{M}_{\overset{N}{\vec{x}}} = \{\overset{k_1}{\vec{c}_1}, \ldots, \overset{k_r}{\vec{c}_r}\}.$$

The sound-types transform (2)

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

$$\mathcal{M}_{\overset{N}{\vec{x}}} = \{\overset{k_1}{\vec{c}_1}, \dots, \overset{k_r}{\vec{c}_r}\}.$$

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

The sound-types transform (3)

- A sound-cluster has an associate **sound-type** $\overset{n}{\vec{\tau}_r}$, defined as the weighted sum of all the atoms in the sound-cluster where the weights $\overset{k_r}{\vec{\omega}_r}$ are the distances of each atom to the center of the cluster:

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

with $\omega_{r,j} \in \overset{k_r}{\vec{\omega}_r}$.

The sound-types transform (3)

- A sound-cluster has an associate **sound-type** $\overset{n}{\vec{\tau}}_r$, defined as the weighted sum of all the atoms in the sound-cluster where the weights $\overset{k_r}{\vec{\omega}}_r$ are the distances of each atom to the center of the cluster:

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

with $\omega_{r,j} \in \overset{k_r}{\vec{\omega}}_r$.

- The set of sound-types in the signal $\overset{N}{\vec{x}}$ is called **dictionary**:

$$\mathcal{D}_{\overset{N}{\vec{x}}} = \{\overset{n}{\vec{\tau}}_1, \ldots, \overset{n}{\vec{\tau}}_r\}.$$

The sound-types transform (4)

- Finally, it is possible to define the *sound-types transform* as a function of time and frequency obtained by multiplying the sound-types in a given dictionary with complex sinusoids:

$$\boxed{\overset{N}{\vec{\Phi}}_{\overset{n}{\vec{k}}} = \sum_{i=0}^{N/t} \overset{n}{\vec{\mathcal{T}}}_{r,p} \cdot e^{-j \cdot \frac{2\cdot\pi}{n} \cdot \overset{n}{\vec{k}}}}$$

where $\overset{n}{\vec{k}} = \{f_1, \ldots, f_n\}$ is a vector of frequencies.

The sound-types transform (5)

- The extreme case for $|\mathcal{M}| = N/t$ 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:

$$|\mathcal{M}| = N/t \implies \overset{1}{\vec{c}_r} = \{\overset{n}{\vec{a}_1}\} \implies \overset{n}{\vec{\tau}_r} = \overset{n}{\vec{a}_r} \cdot \omega_{r,1}.$$

The sound-types transform (5)

- The extreme case for $|\mathcal{M}| = N/t$ 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:

$$|\mathcal{M}| = N/t \implies \overset{1}{\vec{c}_r} = \{\overset{n}{\vec{a}_1}\} \implies \overset{n}{\vec{\tau}_r} = \overset{n}{\vec{a}_r} \cdot \omega_{r,1}.$$

- This leads to the important consequence that STT is a **generalization** of STFT:

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

with p defined as above.

Computational criteria

- Some criteria are needed to define types and rules

Computational criteria

- Some criteria are needed to define types and rules
- They can be obtained by means of a twofold process:
  - **types inference**: first, the types involved in the representations are discovered by looking for common entities in a sound
  - **rules inference**: a second stage is needed to discover the rules that link one type to another by means of a sequential analysis

Creation of sound-types (1)

A possible realization is based on low-level descriptors plus classificaiton for types inference and Markov models for rules inference:

- **(atomic decomposition)**: subdivide a sound into small grains of approximately 40 ms called *atoms* or *0-types* overlapping in time and frequency

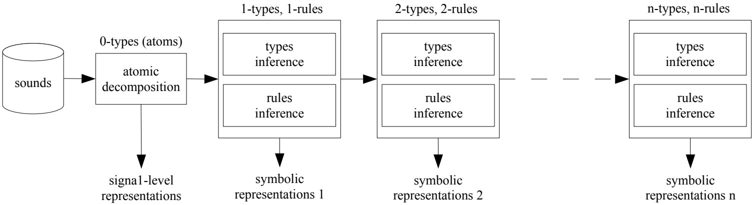
Creation of sound-types (1)

A possible realization is based on low-level descriptors plus classificaiton for types inference and Markov models for rules inference:

- **(atomic decomposition)**: subdivide a sound into small grains of approximately 40 ms called *atoms* or *0-types* overlapping in time and frequency
- **(1-types inference)**: compute a set of low-level descriptors on the atoms obtained in the previous step, project the descriptors in a multi-dimensional space and compute the *clusters*; each cluster will represent a *1-type*
- **(1-rules inference)**: implement a Markov model to describe the sequences of types present in the analysed sound (*1-rules*)

Creation of sound-types (2)

- **(n-types inference)**: compute a set of low-level descriptors on the whole sequences found in the previous step; project again the descriptors and compute the clusters: each cluster will represent a *n-type*
- **(n-rules inference)**: repeat the Markov model until there are no more sequences (*n-rules*).



Sound-types: properties (1)

1. The number of iterations represents the degrees of abstraction

Sound-types: properties (1)

1. The number of iterations represents the degrees of abstraction
2. Discovered types are defined in the time-frequency plane and have increasing time scale

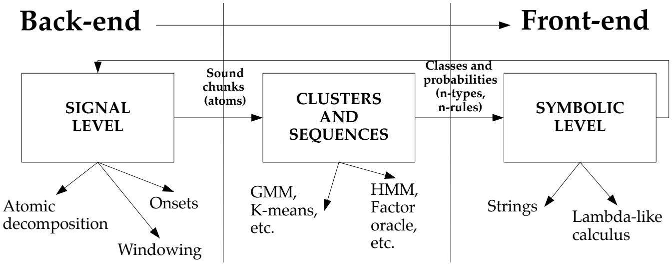
Sound-types: properties (1)

1. The number of iterations represents the degrees of abstraction
2. Discovered types are defined in the time-frequency plane and have increasing time scale
3. The higher the level of a type, the more will be expressive (the less generic)

Sound-types: properties (2)

- *Signal-dependent semantics*: the atoms and the + relation are derived from the signal
- *Scalability*: the possibility to scale over abstraction is implicit to theories of types; we showed how it is possible to translate a representation to another by changing the involved elements and operators
- *Weak invertibility* and *generativity*: there are many possibilities to create a signal back from sound types: pick up randomly an element of each cluster used, pick up the element closest to the center of the cluster or to generate a weighted sum of all the elements of a cluster, etc.

A generalized framework



Implementated techniques for each layer

LAYER	TECHINQUES
Back-end	Windowing, onsets separation
Concatenation layer	Low-level features + GMM
	Low-level features + K-means
	Markov models
Front-end	Descriptive language (strings)

Clusters (1)

A tool called **Clusters** has been implemented in C++ to analyze sounds and produce quasi-symbolic representations:

- Low-level features analysis of sounds (spectral and temporal) with dimensionality reduction (PCA).

Clusters (1)

A tool called **Clusters** has been implemented in C++ to analyze sounds and produce quasi-symbolic representations:

- Low-level features analysis of sounds (spectral and temporal) with dimensionality reduction (PCA).
- Clusters estimation with K-means and GMM (types).

Clusters (1)

A tool called **Clusters** has been implemented in C++ to analyze sounds and produce quasi-symbolic representations:

- Low-level features analysis of sounds (spectral and temporal) with dimensionality reduction (PCA).
- Clusters estimation with K-means and GMM (types).
- Auto-estimation of number of clusters (gap statistic, BIC).

Clusters (1)

A tool called **Clusters** has been implemented in C++ to analyze sounds and produce quasi-symbolic representations:

- Low-level features analysis of sounds (spectral and temporal) with dimensionality reduction (PCA).
- Clusters estimation with K-means and GMM (types).
- Auto-estimation of number of clusters (gap statistic, BIC).
- Whitin-cluster dispersion for outliers detection.

Clusters (1)

A tool called **Clusters** has been implemented in C++ to analyze sounds and produce quasi-symbolic representations:

- Low-level features analysis of sounds (spectral and temporal) with dimensionality reduction (PCA).
- Clusters estimation with K-means and GMM (types).
- Auto-estimation of number of clusters (gap statistic, BIC).
- Whitin-cluster dispersion for outliers detection.
- Original sound reconstruction (inverse transformation): random, average, weighted, closest with various distance measures (euclidean, cosine similarity, taxicab).

Clusters (1)

A tool called **Clusters** has been implemented in C++ to analyze sounds and produce quasi-symbolic representations:

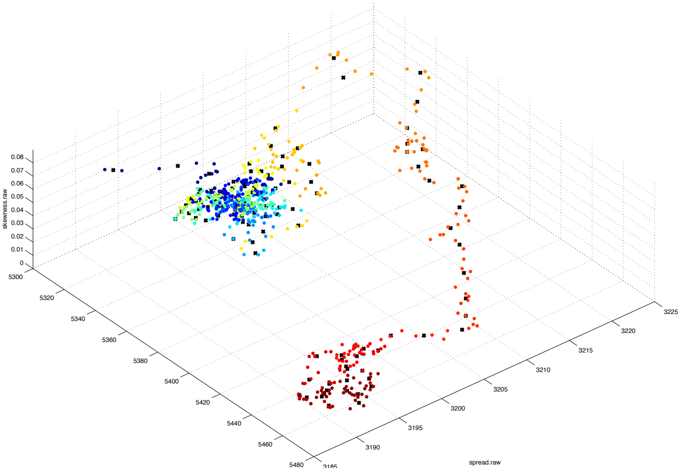
- Low-level features analysis of sounds (spectral and temporal) with dimensionality reduction (PCA).
- Clusters estimation with K-means and GMM (types).
- Auto-estimation of number of clusters (gap statistic, BIC).
- Whitin-cluster dispersion for outliers detection.
- Original sound reconstruction (inverse transformation): random, average, weighted, closest with various distance measures (euclidean, cosine similarity, taxicab).
- Transition probabilities with a Markov model.

Clusters (1)

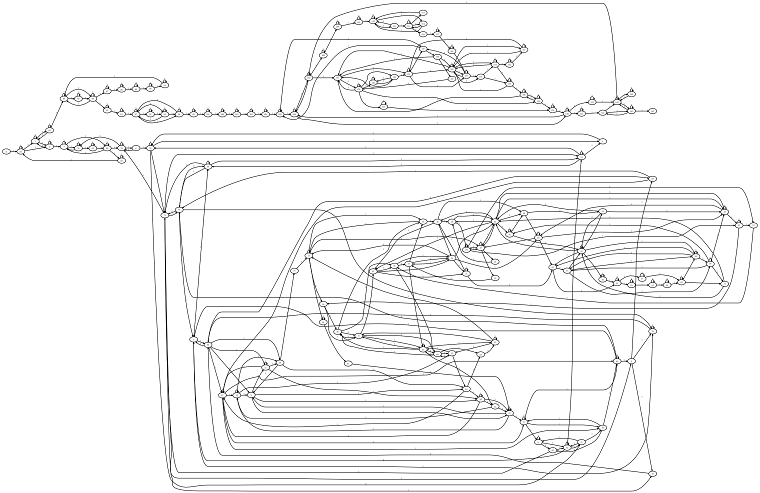
A tool called **Clusters** has been implemented in C++ to analyze sounds and produce quasi-symbolic representations:

- Low-level features analysis of sounds (spectral and temporal) with dimensionality reduction (PCA).
- Clusters estimation with K-means and GMM (types).
- Auto-estimation of number of clusters (gap statistic, BIC).
- Whitin-cluster dispersion for outliers detection.
- Original sound reconstruction (inverse transformation): random, average, weighted, closest with various distance measures (euclidean, cosine similarity, taxicab).
- Transition probabilities with a Markov model.
- Symbolic representation with a simple string.

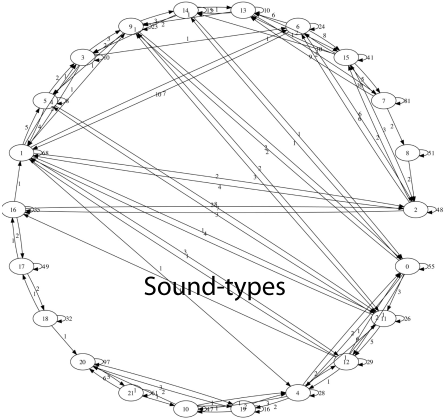
Clusters (2): types representation (3D case)

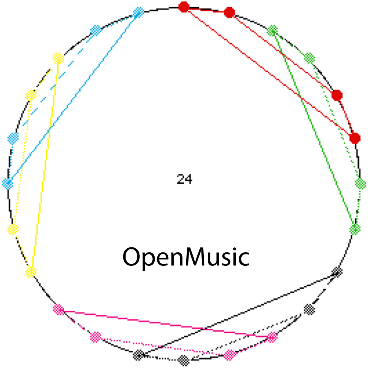


Clusters (3): rules representation



Clusters (4): OM-like representation





Clusters (5): inverse transformation examples 1

Reconstruction

- Original sample 1 (rock band).

Clusters (5): inverse transformation examples 1

Reconstruction

- Original sample 1 (rock band).
- 170 types (10%), 11 mixed features, weighted, taxicab.

Clusters (5): inverse transformation examples 1

Reconstruction

- Original sample 1 (rock band).
- 170 types (10%), 11 mixed features, weighted, taxicab.
- 860 types (50%), 4 spectral features, weighted, taxicab.

Clusters (5): inverse transformation examples 1

Reconstruction

- Original sample 1 (rock band).
- 170 types (10%), 11 mixed features, weighted, taxicab.
- 860 types (50%), 4 spectral features, weighted, taxicab.
- 1600 types (95%), 11 spectral features, weighted, cosim.

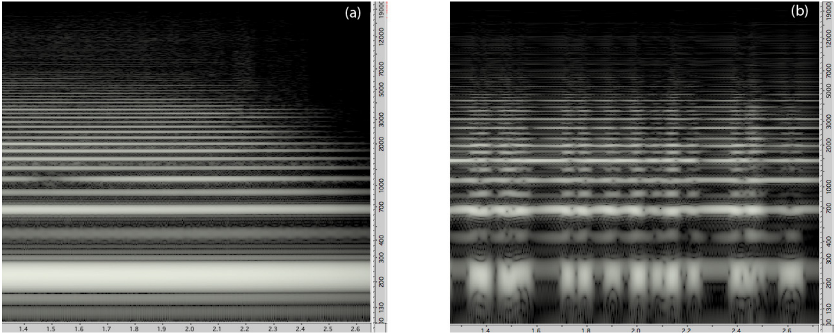
Clusters (5): inverse transformation examples 1

Reconstruction

- Original sample 1 (rock band).
- 170 types (10%), 11 mixed features, weighted, taxicab.
- 860 types (50%), 4 spectral features, weighted, taxicab.
- 1600 types (95%), 11 spectral features, weighted, cosim.
- Original sample 2 (orchestra) and 100% reconstruction.

Clusters (6): bad signal reconstruction

Under a given ratio between types and atoms, the signals are not reconstructed correctly. The following example shows such a situation:



Clusters (7): inverse transformation examples 2

Transformations and generations

- Original sample 3 (lute) and two times slower resynthesis.

Clusters (7): inverse transformation examples 2

Transformations and generations

- Original sample 3 (lute) and two times slower resynthesis.
- Original sample 4 (voice) and one fifth up resynthesis.

Clusters (7): inverse transformation examples 2

Transformations and generations

- Original sample 3 (lute) and two times slower resynthesis.
- Original sample 4 (voice) and one fifth up resynthesis.
- Original sample 5 (bass) and its probabilistic generation.

Clusters (7): inverse transformation examples 2

Transformations and generations

- Original sample 3 (lute) and two times slower resynthesis.
- Original sample 4 (voice) and one fifth up resynthesis.
- Original sample 5 (bass) and its probabilistic generation.
- Original sample 6 (piano) and its probabilistic generation.

Clusters (7): inverse transformation examples 2

Transformations and generations

- Original sample 3 (lute) and two times slower resynthesis.
- Original sample 4 (voice) and one fifth up resynthesis.
- Original sample 5 (bass) and its probabilistic generation.
- Original sample 6 (piano) and its probabilistic generation.
- A probabilistic generation of sample 2 (orchestra).

Clusters (8): current applications

- **Symbolic description**: analysing the created string of labels, it is possible to acquire information of *salient* properties of the sound.

Clusters (8): current applications

- **Symbolic description**: analysing the created string of labels, it is possible to acquire information of *salient* properties of the sound.
- **Audio compression**: while not being the main purpose of the approach, it is possible to compress a sound by a given ratio.

Clusters (8): current applications

- **Symbolic description**: analysing the created string of labels, it is possible to acquire information of *salient* properties of the sound.
- **Audio compression**: while not being the main purpose of the approach, it is possible to compress a sound by a given ratio.
- **Time and frequency transformations**: it is possible to perform various transformations such as time-stretch and pitch-shift.

Clusters (8): current applications

- **Symbolic description**: analysing the created string of labels, it is possible to acquire information of *salient* properties of the sound.
- **Audio compression**: while not being the main purpose of the approach, it is possible to compress a sound by a given ratio.
- **Time and frequency transformations**: it is possible to perform various transformations such as time-stretch and pitch-shift.
- **Probabilistic generation**: using the types and their probabilities it's possible to generate sounds *related* to the original, but different.

The reduction effect

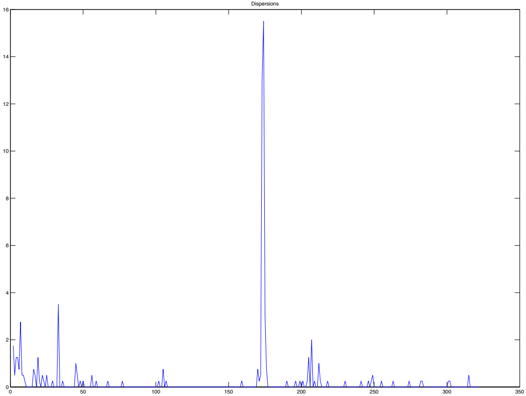
- The smaller the number of clusters (meaning that we reduce the number of clusters, grouping more entities in the same sound-type) the better will be the representation but the worst will be the sound resynthesis.

The reduction effect

- The smaller the number of clusters (meaning that we reduce the number of clusters, grouping more entities in the same sound-type) the better will be the representation but the worst will be the sound resynthesis.
- Sound quality is directly linked to the number of clusters, but if we augment this number we loose the possibility of having a compact representation analysis.

Measure for objective evaluation

Is the *within-cluster dispersion* is a measure for quality?



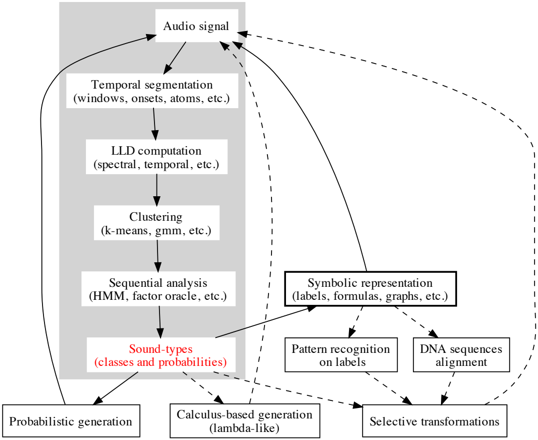
Future applications

- **Selective transformations**: it should be possible to perform various transformations only on some *selected* types.

Future applications

- **Selective transformations**: it should be possible to perform various transformations only on some *selected* types.
- **Pattern recognition on the representation**: the tool represents a sound using a string of labels; many algorithms could be applied on that string to discover patterns.

Possible expansions



Some considerations

- Logical approaches are mainly related to musical *representation* while algebraic methods are connected with both musical representation and musical *creation*.

Some considerations

- Logical approaches are mainly related to musical *representation* while algebraic methods are connected with both musical representation and musical *creation*.
- Most of logical representations of music *failed*.

Some considerations

- Logical approaches are mainly related to musical *representation* while algebraic methods are connected with both musical representation and musical *creation*.
- Most of logical representations of music *failed*.
- The inquiry into symbolic representations of music changed paradigm, moving from a **theoretical** approach to a **computational** one.

Some considerations

- Logical approaches are mainly related to musical *representation* while algebraic methods are connected with both musical representation and musical *creation*.
- Most of logical representations of music *failed*.
- The inquiry into symbolic representations of music changed paradigm, moving from a **theoretical** approach to a **computational** one.
- The theory of sound-types, while in an early stage, offers a promising environment for music representation adapted to the new paradigm.

Any questions?