A Novel XML Music Information Retrieval Method Using Graph Invariants

ALBERTO PINTO and GOFFREDO HAUS
Università degli Studi di Milano

The increasing diffusion of XML languages for the encoding of domain-specific multimedia information raises the need for new information retrieval models that can fully exploit structural information. An XML language specifically designed for music like MX allows queries to be made directly on the thematic material. The main advantage of such a system is that it can handle symbolic, notational, and audio objects at the same time through a multilayered structure. On the model side, common music information retrieval methods do not take into account the inner structure of melodic themes and the metric relationships between notes.

In this article we deal with two main topics: a novel architecture based on a new XML language for music and a new model of melodic themes based on graph theory.

This model takes advantage of particular graph invariants that can be linked to melodic themes as metadata in order to characterize all their possible modifications through specific transformations and that can be exploited in filtering algorithms. We provide a similarity function and show through an evaluation stage how it improves existing methods, particularly in the case of same-structured themes.

Categories and Subject Descriptors: H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Retrieval models*; H.2.3 [**Database Management**]: Languages; H.5.5 [**Information Interfaces and Presentation**]: Sound and Music Computing—*Modeling*; G.2.2 [**Discrete Mathematics**]: Graph Theory; J.5 [**Computer Applications**]: Arts and Humanities—*Performing arts*

General Terms: Languages, Theory

Additional Key Words and Phrases: Graphs, music information retrieval, invariants, melodic similarity, metadata, music, structural properties, XML

ACM Reference Format:

Pinto, A. and Haus, G. 2007. A Novel XML music information retrieval method using graph invariants. ACM Trans. Inform. Syst. 25, 4, Article 19 (October 2007), 44 pages. DOI = 10.1145/1281485.1281490 http://doi.acm.org/10.1145/1281485.1281490

19

Authors' address: A. Pinto, (corresponding author), G. Haus, Dipartimento di Informatica e Comunicazione, Università degli Studi di Milano, Via Comelico 39, 20135 Milano, Italia; email: {pinto,haus}@dico.unimi.it.

1. INTRODUCTION

The importance of music information retrieval in large musical databases increases as the number and dimension of multimedia databases increases.

Music can take advantage of XML languages because of its intrinsically layered structure [Haus 1984], from audio to structural information. Unfortunately, there is currently no defined, independent standard for representing music information that can describe and process all the different layers which characterize it. For each layer of music information, there is one or more accepted standards, for example, MIDI (musical instrument digital interface) for performances, NIFF (notation interchange file format) for notation, and so on, and more proprietary formats. None of them can be suitably applied to other layers [Roads 1996].

XML provides an effective way to represent multimedia information at different levels of abstraction and XML metadata provide useful tools for the retrieval processes [Baeza-Yates and Ribeiro-Neto 1999]. The IEEE Definition of a Commonly Acceptable Musical Application Using the XML Language project has been developing an XML application defining a standard language for symbolic music representation. The language is a metarepresentation of music information for describing and processing the aforesaid music information within a multilayered environment, for achieving integration among structural, score, MIDI, and digital sound levels of representation.

Furthermore, the proposed standard should integrate music representation with already defined and accepted common standards.

The standard will be accepted by any kind of software dealing with music information, such as score editing, OMR (optical music recognition) systems, music performance, musical databases, and composition and musicological applications.

1.1 Overall System Architecture

The most innovative feature of our architecture is that it provides tools for efficient storage of structural musical data and for performing content-based queries on such data. The overall architecture of the musical data management module is illustrated in Figure 1. The module consists of two main environments: the musical storage environment and the musical query environment. The musical storage environment has the purpose of representing musical information in the database to make query-by-content efficient. The Musical query environment provides methods to perform query-by-content on music scores, starting from a score or audio fragment given as input.

The matching between input and the scores stored into the database is performed in several steps, graphically illustrated in Figure 1. The input can be either an audio file or a score fragment, played by the user on a keyboard, sung, or whistled into a microphone connected to the system [Haus and Pollastri 2000].

—From the audio files, note-like attributes are extracted by converting the input into a sequence of note numbers. Such a step is performed by the symbolic

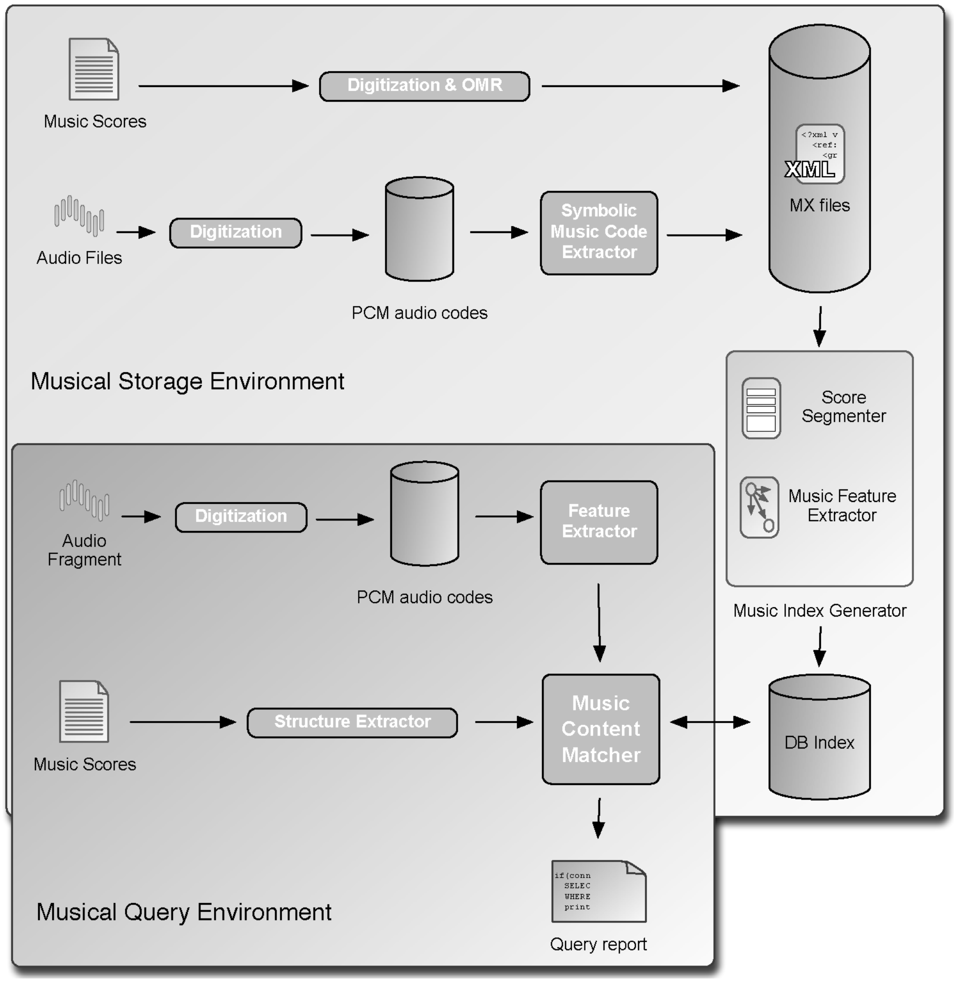

Fig. 1. General architecture.

music-code extractor module (see Figure 1). The conversion uses a pitch-tracking algorithm based on conventional frequency-domain analysis. If the input is entered from a keyboard or is a score fragment, conversion is not necessary and the sequence can be directly built.

—The feature extractor identifies the structure of the input by converting acoustic input first into an audio feature sequence and then into its related mathematical (graph) representation.

—The matching phase is worked out by the music-content matcher that applies filtering algorithms based on graph invariants and computes the similarity function (refer to Section 3.6) based on the music graph representation stored in the database index.

As we show in Section 3.6, the comparison of the input with a given score has been replaced with graph matching.

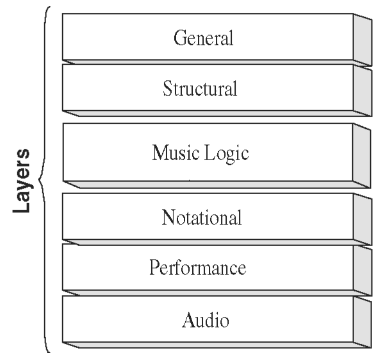

Fig. 2. Music information layers.

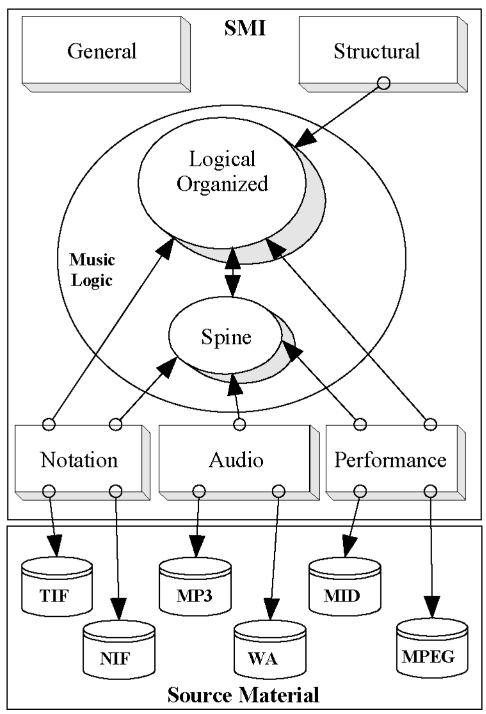

Fig. 3. Relations among layers.

The next section shows the hidden XML environment where the structural metadata can be stored and whose multilayered structure allows the integration of audio and symbolic features of music.

1.2 The MX Structure

In MX, music information is represented according to both a multilayered structure and to the concept of the space-time construct. In fact, music information can be structured by a layer subdivision model, as shown in Figures 2 and 3.

Each layer is specific to a different degree of abstraction in music information: general, structural, music logic, notational, performance, and audio. Haus and Longari [2005] give an exhaustive description of this format, and the issue of integration between MX and other formats is covered in Longari [2004].

The main advantage of MX is the richness of its descriptive features, which are based on other commonly accepted encodings aimed at more specific descriptions.

The multilayered music information structure is kept together by the concept of *spine*. Spine is a structure that relates time and spatial information (see Figure 4), where measurement units are expressed in relative format. Through such a mapping, it is possible to fix a point in one layer instance (e.g., notational) and investigate the corresponding point in another (e.g., performance or audio).

The *structural* layer, which is also the most interesting from our perspective, contains explicit descriptions of music objects, together with their causal relationships, from both compositional and musicological points of view, that is, how music objects can be described as the transformation of previously described music objects. A particular structural object is a theme which represents exactly the concept of a musical theme of the particular piece (or part of it) under consideration. Theme objects may be the output of an automatic segmentation process [Haus et al. 2004] or the result of musicological analysis.

Therefore, a content-based information retrieval method cannot leave aside the possibility of exploiting the presence of new specific metadata specifically oriented towards retrieval in order to improve processes and algorithms.

We will return to this subject in Section 4 after the introduction of the graph model of thematic fragments (themes) for music information retrieval processes.

2. RELATED WORK

2.1 Music Representation

XML is an effective way to describe music information. Nowadays, there are a number of good dialects to encode music by means of XML, such as MusicXML, MusiXML, MusiCat, MEI, MDL (see Longari [2004] for a thorough discussion). In particular, we have at least two good reasons to mention MusicXML [Good 2001]. MusicXML is a comprehensive way to represent symbolic information. As a consequence, it was integrated into a number of commercial programs. Among them, it is worthwhile to cite one of the leading applications for music notation: Coda Music Finale. One of the key advantages of MusicXML over other XML-based formats is represented by its popularity in the field of music software. However, all the encoding formats we listed before are not relevant for semantic descriptions of metadata. In the MPEG-7 (moving picture experts group) context, currently there are initiatives to integrate OWL (ontology web language) ontologies in a framework developed for the support of ontology-based semantic indexing and retrieval of audiovisual content. This initiative follows the semantic level of MPEG-7 MDS (multimedia description schemas), and

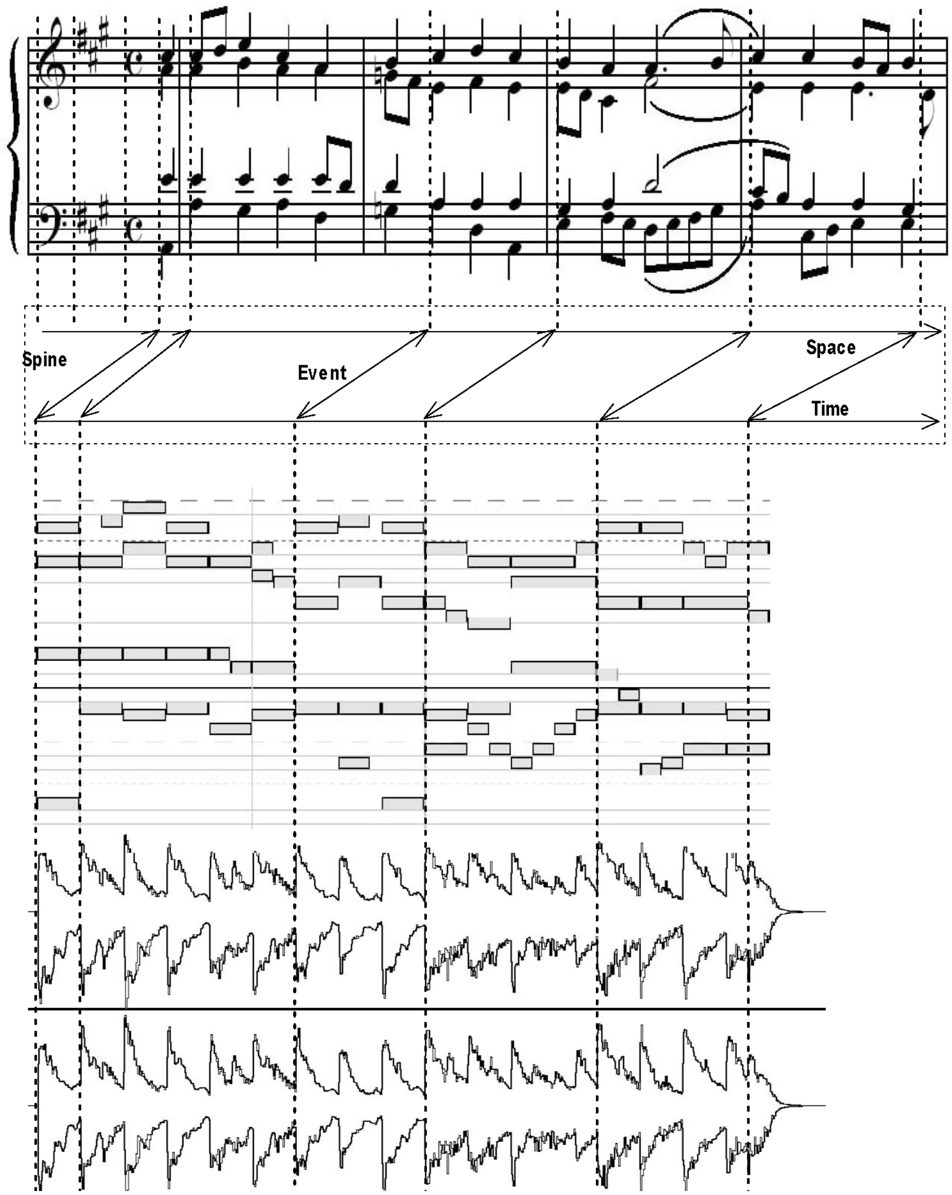

Fig. 4. Spine: relationships between notational, performance, and audio layers.

TV-Anytime standard specifications for metadata descriptions. Despite the MX semantic layer, the MPEG-7 semantic level describes music information from the real-world perspective, giving emphasis to the events, objects, concepts, places, and time in narrative worlds and in abstraction. Therefore, MPEG-7 ontology is only aimed at the description of music performance and not of score information, as in the case of MX. The interested reader can find a complete discussion of these topics in Tsinaraki et al. [2004].

Genre, an intrinsic property of music, is probably one of the most important descriptors used to classify music archives. Traditionally, genre classification has been performed manually, but many automatic approaches are provided by the state-of-the-art. In Aucouturier and Pachet [2003], three different categories of genre classification are proposed: (i) a manual approach based on human knowledge and culture; (ii) an automatic approach based on automatic extraction of audio features; and (iii) an automatic approach based on objective similarity measures. Taxonomy use is the main difference between the two automatic approaches: In the first mathod, a given taxonomy is necessary, while in the second this is not required. In MX, we have tried to classify genres by an OWL ontology in order to get a taxonomy as flexible as possible and capturing the complexity of real-world genre classifications.

Although important in music information retrieval, genre classification is not the main topic of this article. Here we are interested in the more specific retrieval of musical themes independently from their possible classification in one or another genre.

2.2 Formal Models

The treatment of musical themes cannot leave aside semantic considerations [Hewlett and Selfridge-Field 2005, 2000] about the particular context in which we operate. Musical themes are not pure sequences of bytes or symbols; on the contrary, they represent relationships among scale degrees. To the best of our knowledge, methods developed up to now to compare musical themes do not take into account these relationships and this is a big fault, particularly for analysis oriented towards the structural aspects of musical information. There are some approaches relying on string edit distance (see, e.g., Mongeau and Sankoff [1990]) that actually take into account such relationships, but only as editing weights and this does not reflect into a suitable formal model of themes.

Among formal models, most common approaches range between purely functional and purely reductionistic. We are going to introduce them in the next sections.

2.2.1 *Functional Metrics.* Most music information retrieval methods adopt a functional approach, that is to say, they treat musical themes like functions and therefore base similarity function on metrics in functional spaces [Diana 2004]. Unfortunately, in doing so we lose the underlying relationships between consecutive notes, which could be very important, from a musicological point of view, in finding affine similarities between themes [Verdi 1998]. Functional models basically rely on metrics like those of Tenney and Polansky's [Polansky 1996; Tenney and Polansky 1980]. They founded their approach on the concept of *morphology* [Polansky 1992], substantially a finite sequence of comparable elements.

In this context melodies are finite sequences, that is, functions defined on a finite subset of $\mathbf{N}$ and whose values are in mono ($\mathbf{Q}$) or n-dimensional ($\mathbf{Q}^n$) spaces, according to the characterizing parameters. Thus, distance concepts are those inherited by metrics defined on (discrete) functional spaces.

The measure of these distances involves the creation, recognition, and analysis of variations and transformation of morphological parameters like pitches, onsets, harmonic relations, sequences of timbre-related values, and, more generally, any kind of observable related to melody.

As pointed out by Tenney and Polansky [1980], we have to distinguish between statistical and morphological properties. Generally, the former statistical are *global* and *time independent*, like the mean value or standard deviation of a parameter, while the latter are described by the *profile* of parameters and depend on element ordering. It is also possible to use statistic measures of parametric profiles as parameters at a higher level (i.e., hierarchy of profiles); in this way we can analyze the melodic profile at different levels by the application of different metrics.

Definition 2.1. A morphology M is an ordered set of elements. The elements in M are identified by M_i, $i = 1, \dots, L$; $L = |M|$.

Some examples of morphologies are pitch sequences, rhythm sequences, harmonic sequences, etc. [Tenney and Polansky 1980; Polansky 1996].

Morphological metrics are metrics on morphologies (i.e., metrics on ordered sets).

Definition 2.2. Given a set S, a function

$$d : S \times S \longrightarrow \mathbf{R} \tag{1}$$

is a *distance function* or a *metric* if $\forall a, b, c \in S$ holds the following:

(1) $d(a, b) \geq 0$;
(2) $d(a, b) = 0$ iff $a = b$;
(3) $d(a, b) = d(b, a)$; and
(4) $d(a, b) \leq d(a, c) + d(c, b)$.

Moreover, (S, d) is called a *metric space*. Metrics on spaces of real-valued functions are useful models for morphological metrics. For example, given two continuous real-valued functions $f(t)$ and $g(t)$ defined on $[m, n]$, there are two intuitive functional metrics.

$$d(f, g) = \max\{|f(t) - g(t)|\} \quad \text{(max metric)} \tag{2}$$

These are the ordinary *sup* metrics, which become *max* because of the compactness of the domain, and

$$d(f, g) = \int_n^m |f(t) - g(t)| dt \quad \text{(amplitude metric).} \tag{3}$$

Working in discrete spaces, the integrals will be replaced by sums.

By replacing $f(t)$ and $g(t)$ with their derivatives of any order, Tenney and Polansky [1980] obtained metrics by measuring the mean amplitude difference of the corresponding rate of change of the two functions. For discrete functions the derivative is substituted by the *difference function* of first (second, third, $\cdots$)

order. Given two morphologies M, N, of length L, the amplitude metric is

$$\sum_{i=1}^{L} |M_i - N_i|, \tag{4}$$

or, normalized,

$$\sum_{i=1}^{L} \frac{|M_i - N_i|}{L}. \tag{5}$$

Generalizing the concept, it is possible to make an average of the higher-order derivatives, as in the Sobolev metric.

$$d(f,g) = \sum_{i=0}^{n} \left[\sqrt{\int (f(t)^i - g(t)^i)^2} \right], \tag{6}$$

where i is the order of derivative. An analogous metric is the L_1-version of the previous metric, normalized with respect to the number of derivatives.

$$d(f,g) = \sum_{i=0}^{n} \frac{\left[\int |f(t)^i - g(t)^i|\right]}{n} \tag{7}$$

Then, weighting difference functions, we obtain the (linear-ordered) metric.

$$d(M,N) = \frac{1}{\sum \alpha(i)} \sum_{i=0}^{n} \alpha(i) \cdot \frac{\sum_{j=1}^{L-i} \left|M_j^i - N_j^i\right|}{L-i}, \tag{8}$$

where i is the order of difference function on M and N. Obviously, the length decreases step-by-step.

Starting from $i = 1$ implicates the exclusion of the elements of M and N, so melody transposition will have zero distance.

Specifically, $\alpha(i)$ represents a weight function indexed by the order of difference function $n < L-1$, where L is the length of M and N. The weight function establishes the importance of each order of derivation.

Eq. (8) inspired our similarity function by a suitable replacement of derivatives with graph powers. As we will show, in this process we will lose symmetry, so the resulting similarity function will not be a metric. We will describe it later in detail after the introduction of some graph theory concepts.

2.2.2 *Reductionistic Approach.* For the sake of completeness, we want to introduce the reductionistic approach that has been used in order to lessen some of the typical "false positives" the functional approach implies.

This approach exploits additional musicological information related to the piece (e.g., time signature) and assigns different levels of relevance to single notes of the melody. For example, it may choose to assign higher importance to the stressed notes inside a bar. In other words, the goal of comparing two melodic sequences is achieved by reducing musical information into some "primitive types" and comparing the reduced fragments by means, for example, of functional metrics.

Fig. 5. J.S. Bach, B.A. 4,186: Score reductions.

A very interesting reductionistic approach refers to Fred Lerdahl and Ray Jackendoff's studies. Lerdahl and Jackendoff [1996] published their research oriented towards a *formal description of the musical intuitions of a listener who is experienced in a musical idiom*. Their work was not directly related to music information retrieval (MIR); their purpose was the development of a formal grammar which could be used to analyze any tonal composition. However, in the case of thematic fragments (themes), it would be possible to reduce the themes into primitive types showing formal similarities according to the defined grammar.

The aim is to describe, in a simplified manner, the *analytic system of the listener*, namely, those rules which allow the listener to segment and organize a hierarchy of musical events. On this basis, *score reductions* are applied, gradually deleting the less significant events. In this way we can obtain a simplification of the score and in the meantime preserve sufficient information to maintain recognizability.

The study of these mechanisms allows the construction of a grammar that is able to describe the fundamental rules followed by the human mind in recognition of the underlying structures of a musical piece.

In Figure 5 there is an example of score reductions provided by Lerdahl and Jackendoff [1996]. In Figure 6 we provide an example of analysis based on the incipit of Mozart's Symphony in G Minor KV 550. Grammar rules allow analyses such as those sketched in Figures 7(a) and 7(b). Both analyses are grammatically correct, but no musician would subscribe to them.

Fig. 6. W.A. Mozart, KV 550: correct analysis.

Fig. 7. W.A. Mozart, KV 550: improbable correct analysis.

The main difficulty with Lerdahl's and Jackendoff's [1996s] work is that their view of music is overly simplified. Even with melodic themes, numerous examples can be pointed out where different components of music form independent structures that overlap and cannot fit into only one hierarchy. A musical theme does not submit to just one interpretation, and a piece of music with artistic value may call forth several independent and even contradictory sets of intuitions in an experienced listener.

The problem is that the preference rules cannot be used at all without a judgment by the analyst, and even the well-formedness rules contain terms that are not formalized, such as "cadence" and "consonant". The end result of this is that the theory cannot be expressed as a computer program that would analyze pieces by itself without any additional input by a human analyst.

An exhaustive description of this approach goes beyond our purposes and would be very complex. Moreover, in our opinion, it cannot be satisfactorily exploited in an MIR system for the reasons cited earlier [Baroni et al. 1999].

3. THE GRAPH MODEL

3.1 Model Construction

The idea underlying the construction of our model is a typically geometric one in the Klein sense, that is to say, the study of a geometric object up to the specific transformations acting on it.

A very useful method in this kind of classification problem is usually to find quantities related to the object that are *invariant* under particular groups of transformations [Pinto 2003].

In our context, we consider a structured set of themes (a *database*) and a theme (the *query*) which has to be compared with every set-element. This is how we proceed:

(1) We build a representative graph for every theme; and
(2) We work with graphs instead of themes.

In this way, we can recognize a greater number of relevant musical similarities and can reduce the number of themes which should be compared. Moreover, those themes of the archive which have the same representative graph can be already identified.

For better comprehension of the model the reader can find a tutorial on basic graph-theoretic notions in the Appendix (Section A). Section B contains the original definitions and results which are used in this section for construction of the model; due to their typically mathematical content, they have been placed in the Appendix.

Let M be a theme of length $m = |M|$ and consider the sequence of pitches $\{h_s\}_{s\in\mathbf{I}}$, $\{I = 1, \ldots, m\}$. Then let (V, d) be a metric space on a finite set of elements (V). Specifically, V and d depend upon the musical system we are considering.

Now, let us consider the linear graph G_l obtained by associating a vertex labeled by h_s with every element $h_s \in V$ and an oriented arrow $a_s : h_s \to h_{s+1}$ to every pair (h_s, h_{s+1}), so that $\partial_0 a = h_s$ and $\partial_1 a = h_{s+1}$. Then we identify vertices with the same label and the result is a quotient set, obtained from the vertex set by the "same label" equivalence relation. Moreover, the vertex corresponding to the starting note is pointed to in order to track it.

Definition 3.1. Let M be a theme that we call the *musical graph representing M* (and we write as $G(M)$), the graph obtained by the process earlier described.

Now we are going to analyze the properties of the graph G(M) representing a theme M.

PROPOSITION 3.2. *Musical graphs are Eulerian, connected, oriented multigraphs.*

PROOF. The proof is trivial if one considers the construction described before. In fact, we have sent every interval of the original theme in an equally oriented arrow. The melody is a sequence of intervals, so it is clear that such a sequence represents an oriented trail in the graph which uses every edge once and once only. The closure of the trail is imposed by the definition because we suppose the last interval being the last note-first note one. □

The next proposition characterizes the themes which contain a *series*.

PROPOSITION 3.3. *A theme M contains a series iff its representative graph $G(M)$ is Hamiltonian.*

PROOF. In fact, given a series M, the resulting graph is necessarily Hamiltonian because it contains all the intervals of the series. Conversely, a Hamiltonian path in G obviously represents a series M. □

Example 3.4. Let us consider the dodecaphonic series in Figure 8 [Schoenberg 1911]. Figure 9 shows its representative graph, C_{12}, which is Hamiltonian.

Remark 3.5. Cyclic graphs C_n are a trivial example of a series.

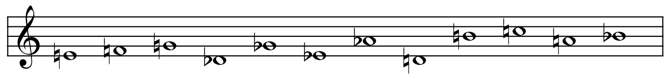

Fig. 8.

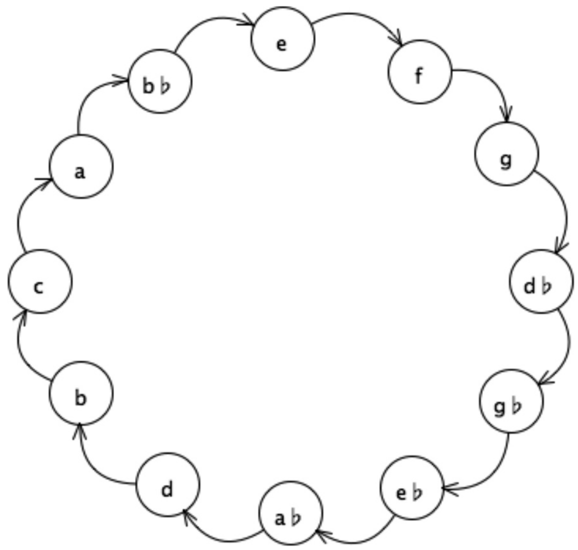

Fig. 9. The cyclic graph C_{12}.

3.2 Theme Equivalence

A formal model for melodic themes has to include all standard transformations of canonic-imitative music and might concern also more general ones, oriented, for example, to contemporary music. Regardless, one of the directions in which we need to enlarge the concept of similarity is the recognition of transformations, such as the permutation of melodic subfragments in order to allow the recognition of themes that are instances of the same theme by a reordering of its subfragments. We will treat this topic in more detail in the next sections; now let us analyze two concepts of the equivalence of themes that will be central to the model.

3.2.1 *Euler-Equivalence of Themes.* An equivalence concept arising from the representative graph construction concerns the different trails in the graph.

Definition 3.6. We say that two themes are *Euler-equivalent* iff they have the same representative graph.

Let us try to better understand what this could mean from a musical point of view by some propositions.

PROPOSITION 3.7. *A graph is Eulerian iff it is decomposable in edge-disjoint cycles.*

Fig. 10.

Fig. 11.

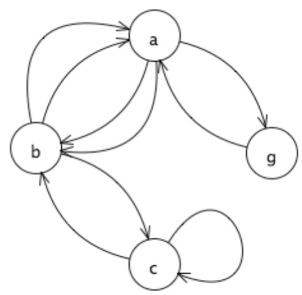

Fig. 12. The representative graph of example 3.8.

Musically, this proposition means that in an equivalence class there are themes which admit of a common cycle decomposition.

Example 3.8. Now, consider the theme in Figure 10 [Bach 1976], and let us permute the cycles with the evidenced start and stop points (B,A,B) and (B,A,G,A,B), as in Figure 11. The two themes have exactly the same representative graph, shown in Figure 12.

Therefore, when we consider a particular musical graph, we are actually considering all the themes corresponding to all the different Eulerian circuits of the graph (with fixed starting points).

Musically, this means that we *identify themes obtained by particular permutations of subfragments*. We want to point out that these are not arbitrary permutations of notes, as otherwise there would be no advantages with respect to Tenney and Polansky's nonordered interval metrics [Tenney and Polansky 1980; Polansky 1996]. Permutation of subfragments is a continuous operator in the sense that it maps consecutive notes to consecutive notes. This implies also that we cannot have a casual rearrangement of note order. The only admissable permutations of theme segments are those corresponding to repetition of notes (musically) or bifurcation in trails (graphically). This means that we can, for example, rearrange a theme in order to remove (or move) trills or other closed embellishments which start and end on the same note without changing the theme, as the representative graph does not change. This corresponds to the fact that if in a node there is more than one out-going arrow, there is more than one possible trail to explore. Moreover, it is possible to compute exactly the number of Euler-equivalent themes by the next result.

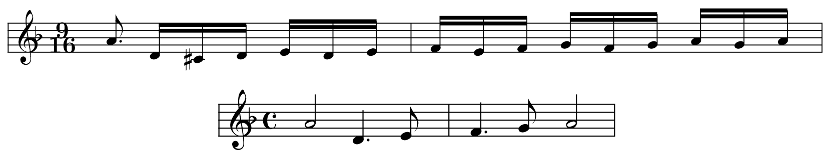

Fig. 13.

PROPOSITION 3.9. *Every class of Euler-equivalent themes has cardinality given by*

$$c \cdot \prod_{i=1}^{n} (d_i^+ - 1)!, \tag{9}$$

where c is the number of equally oriented spanning trees of the same representative graph.

PROOF. The proposition follows from the Cayley theorem (see Appendix A). □

3.2.2 *Equivalence of Themes.* Now we give a more general notion of equivalence which includes also the standard transformations of music theory.

Definition 3.10. Two themes are *equivalent* iff their representative graphs are isomorphic.

Remark 3.11. Equivalence implies Euler-equivalence.

Remark 3.12. Standard melodic transformations are included in the isomorphism definition. In fact, isomorphism implies an isometry between the metric spaces of vertices; therefore if we consider, for example, the standard equally tempered metric space (S^1) it is evident that transformations like *transposition* and *inversion* are isometries $i : V \to V$.

Retrogradation is a transformation of the theme that consists in the inversion of time orientation, so we have just to consider the opposite graph, which has the same arrows as the original but with opposite orientation.

3.3 Theme Inclusion

Besides Euler-equivalent themes, there is another interesting class of theme that can be obtained from a given one and this corresponds to *Eulerian subgraphs*. In fact, it is possible to choose vertex and arc subsets such that the resulting graph remains Eulerian.

Example 3.13. Consider the two themes in Figure 13 [Bach 1976]. Of course, the second theme is a Eulerian subgraph of the first, as shown in Figure 14.

This is a quite particular case, which points out how we can recognize thematic inclusions by a simple graph difference. The necessary condition in this case is that the embellishments must be *closed circuits* of the graph. Thus we have the following proposition.

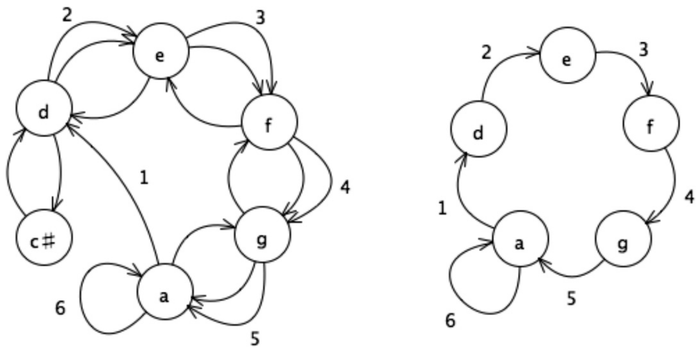

Fig. 14. The graph on the left contains a subgraph isomorphic to the graph on the right.

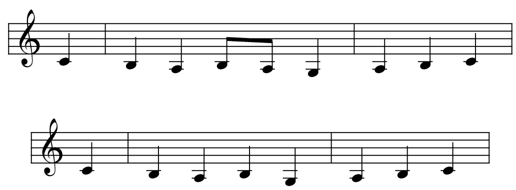

Fig. 15.

PROPOSITION 3.14. *Let M be a theme and M' another theme obtained from M by the addition of closed embellishment to the notes of M. Then $G(M)$ is an (Eulerian) subgraph of $G(M')$.*

Now we are going to introduce a notion of inclusion using the "weak inclusion" defined in Appendix A.

Consider the musical themes in Figure 15 [Bach 1976]. It is clear, musically speaking, that the first theme is a variation of the second. We depict their graphic representation in Figure 16. The next definition formalizes this concept.

Definition 3.15. We say that a theme A is included in a theme B if the representative graph of A is weakly included in the representative graph of B, $G(A) \subseteq G(B)$ (see Appendix B.8), such that $i : V_G(A) \to V_G(B)$ is an isometry.

The problem of deciding the inclusion of one theme into another, using the larger concept of inclusion defined in Appendix B, moves from the themes to their representative graphs.

Theoretically, it may also be possible to use *string matching* techniques, but there are at least three facts which led us to exclude those methods. First of all, we should compare all the themes without distinction, and this would be particularly expensive from a computational point of view, especially real-time. Another fault of the sequential approach is the fact that it is rigorously *note dependent*, namely it depends, in our graphic language, on vertex labeling. The transformation of themes should be not contemplated. In the end, the natural equivalence class of themes should be that concerning the isometries between the vertex set only, excluding permutations of subthemes.

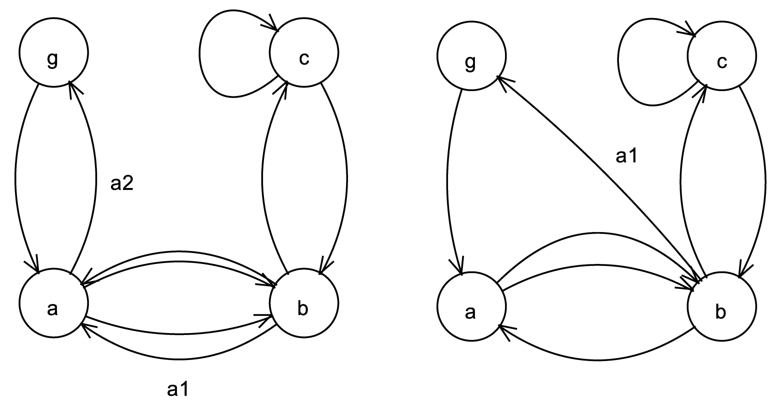

Fig. 16. Graph representation of the two themes.

A large part of our work consisted in searching for suitable graph invariants *monotonic* with respect to the partial-order relation of *inclusion* in the graph set defined in Appendix B.8 [Buckley and Harary 1990]. Using graph invariants we could consider a theme with all its transformations by isometries. This process really does not depend on vertex labeling and so we can consider a theme together with all its transformed ones.

To speed-up the inclusion test, however, it would be necessary to find a good set of *necessary conditions*, which can reduce the set of themes to be compared. Now we will explore the conditions so far identified.

3.3.1 *Order and Size.* The first invariants we are going to consider are graph order and graph size.

Definition 3.16. The *order* and the *size* of a theme M are the order $n(G)$ and size $m(G)$ of the graph G representing M.

Of course we can observe the following.

Remark 3.17. If $G \subseteq H$ then

$$n(G) \leq n(H) \qquad m(G) \leq m(H), \tag{10}$$

that is, the number of vertices and the number of arrows in G have to be less than or equal to their images in H.

From a musical point of view, the condition concerning arrows is evident: The fragment $M(H)$ must have more intervals than $M(G)$. The vertex condition is less trivial. In fact, it says that the total number of distinct notes must increase or remain stationary in an inclusion, so we could never have a situation like the one in Figure 17.

In the figure, the second fragment includes the note g, but the first does not. These fragments are not comparable. Moreover, our inclusion relation induces a *partial* order on the musical graph set. From our point of view this does not represent a problem. In fact, in the retrieval process queries are not supposed to contain unuseful or redundant information and this means

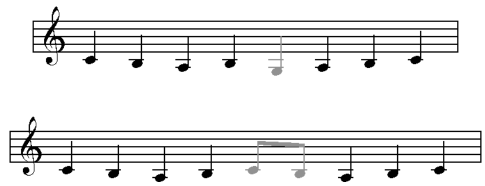

Fig. 17.

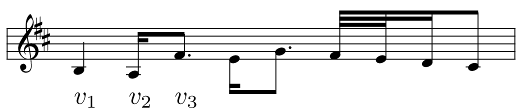

Fig. 18.

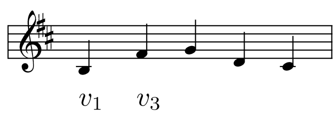

Fig. 19.

that a query cannot contain more notes than its variations stored in the database.

3.3.2 *The Complexity.* Another significative invariant is the number of spanning trees of the graph representing the theme. At the graph level, an embellishment is obtained by the substitution of an arrow with an equioriented trail of the same total weight, with the same beginning and end vertices.

Consider the simplest variation: the insertion of one note.

Example 3.18. In the fragment of Figure 18 the first note of the dotted rhythm (named *alla lombarda*), corresponding to vertex v_2, is the embellishment of the fragment in Figure 19. So, in the corresponding graph we have the following fragments:

—First fragment: the arrows $a_1 = (v_1, v_2)$ and $a_2 = (v_2, v_3)$; and
—Second fragment: the arrow $b_1 = (v_1, v_3)$,

where $|b_1| = |a_1| + |a_2|$.

In this example, vertex v_2 has minimum degree ($v^+ = v^- = 1$), that is, there are no other trails passing through it, because we wanted to point out the nondecreasing property of graph complexity in the substitution of M_2 with M_1.

In fact, the second fragment (i.e., M_2) has a $k(G(M_2))$ spanning tree and if we replace the arrow b_1 with the trail $a_1 a_2$, such trees continue to span the graph (M_1).

This fact illustrated by the example holds also with the added vertex belonging to the first graph. Let us better formalize this fact with the following definition.

Definition 3.19. The *complexity* $k(M)$ of a theme M is the complexity of its representative graph $g(M)$ (see Definition A.6).

We would remind the reader of the following.

Remark 3.20. The complexity of a graph is equal to its the number of spanning trees.

PROPOSITION 3.21. *Let M be a theme and M' a variation of M obtained by the insertion of notes. So we have $k(M) \leq k(M')$.*

PROOF. At the representative-graph level, the process which transforms M in M' consists of a substitution of oriented arrows by oriented trails. In this way the proposition follows from Propositions B.12 and B.13. □

3.4 The Metric on V_G

By definition, given a musical graph G, the set V_G is a metric space (see Appendix B). It is really important to consider the note set only as a metric space, as we are interested in the invariant objects of all musical transformations. The only really critical facts are the distance ratios between the notes. Certainly, the most general transformation in our context is an *arbitrary permutation* of nodes; however, even a learned listener should run into serious difficulties in recognizing such a transformation, even if it could be useful from a musicological perspective. The problem here is that this kind of transformation is not an isometry, and so the relationships between notes are no longer preserved. Moreover, if we change the rhythm, the probabilities of recognizing the theme should probably tend to zero.

We would remind the reader at this point of the musical transformations a theme can undergo: Standard (rigid) ones are transpositions, inversions, and retrogradations. We can sketch them in the simple case, when the space of musical notes is the $\mathbf{Z}_{12}$ ring, in order to provide a brief tutorial to musical nonexperts.

—*Transposition* (T): This is the addition of a constant n to a given pitch sequence, namely $T_n[x] = [x + n]$.

Example 3.22. Given the theme in Figure 20, one of its transpositions could be the one shown in Figure 21.

—*Inversion* (I): This consists in a sign change $I[x] = [-x]$. The transposed inversion will be $T_n I[x] = [-x + n]$.

Example 3.23. The (tonal) inversion of the theme in Example 3.22 is the one shown in Figure 22.

—*Retrogradation* (R): It is the inversion of the temporal flow. Given a pitch sequence $S = \{s_1, s_2, \ldots, s_n\}$, its retrogradation R is the operator that maps S into S^{-1}, so that $R : S = \{s_1, s_2, \ldots, s_n\} \longrightarrow S^{-1} = \{s_n, s_{n-1}, \ldots, s_1\}$, $s_i \in \mathbf{Z}_{12}$.

Example 3.24. The retrogradation of the theme in Exampl 3.22 is the one shown in Figure 23.

Fig. 20.

Fig. 21.

It is possible to give a necessary condition for the inclusion that will have great relevance in recognizing these transformations.

Proposition 3.25. *If a theme M is contained in another theme M', then the inclusion function $i_{V_G(M)}$*

$$\begin{tikzcd} A_G(M) \arrow[r, tail, "i_{A_G}"] \arrow[d, tail, "\partial_0, \partial_1"'] & \mathcal{P}(A_G(M')) \arrow[d, tail, "\partial'_0, \partial'_1"] \\ V_G(M) \arrow[r, tail, "i_{V_G}"'] & V_G(M') \end{tikzcd} \tag{11}$$

is an isometry.

Proof. The proposition is obvious, using the definitions of (weak) inclusion and musical transformation. □

3.5 The Power Graph

Let us consider the themes in Figures 24 and 25 [Bach 1976]. They differ by a passing note (i.e., closure of the third). At the graph level (see Figure 26) we can observe that the arrow $a_1 : b \to g$, which is present in the first graph, is replaced by two arrows $a_1 : b \to a$ and $a_2 : a \to g$ in the second one, so a trail of length 2 replaced an arrow.

Let us recall the transitive closure operation.

Definition 3.26. Let G be a graph. We call the *transitive closure* of G the graph $\overline{G}$ such that $V_{\overline{G}} = V_G$ and $A_{\overline{G}}$ contains all the arrows of A_G and the arrows a_i such that

$$a_i = b_i b_j \quad \wedge \quad \partial_1 b_i = \partial_0 b_j \ , \ b_i, b_j \in A_G. \tag{12}$$

Remark 3.27. The transitive closure is an internal unary operation on the set of Eulerian graphs; in fact we can observe the next proposition.

Proposition 3.28. *If G is a Eulerian graph of size m, then $\overline{G}$ is also Eulerian of size $2m$.*

Fig. 22.

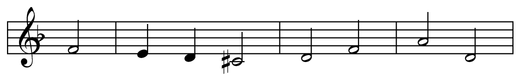

Fig. 23.

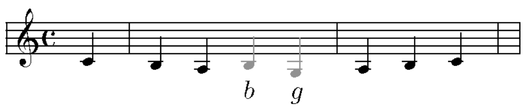

Fig. 24.

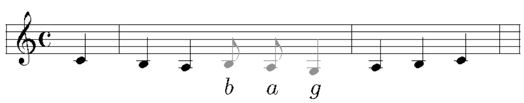

Fig. 25.

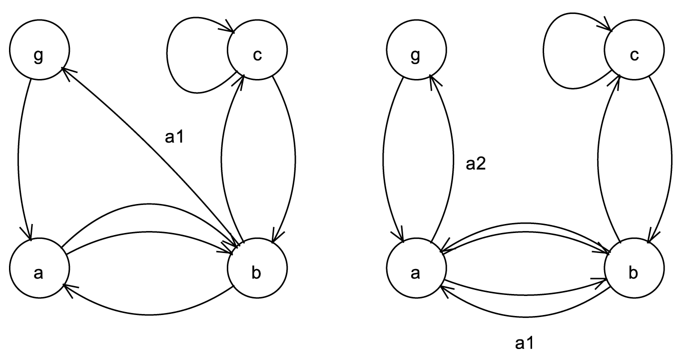

Fig. 26. Closure of thirds process.

Proof. Given a Eulerian cycle in G, the arrows of $\overline{G}$ are the arrows of G plus a number of arrows equal to the number of adjacent arrow pairs of G. Thus it is obvious that $|\overline{G}| = 2m$. By construction, the graph resulting from a closure operation on a Eulerian graph is obviously Eulerian. □

From a musical point of view, the operation consists of a union of two themes: $M(G)$ and the theme obtained from $M(G)$ taking all the notes of $M(G)$, but proceeding by jumps. Iterating the process we can obtain the kth *powers* of a graph.

Definition 3.29. We call the kth power M^k of a theme $M(G)$ that theme whose representative graph is obtained from the graph $G(M)$ iterating (k-1)-times the transitive closure operation.

The interesting result which results from these definitions is the next proposition.

PROPOSITION 3.30. *Let M and M' be two themes. M' contains a variation of M by adding single notes if and only if there exists an isometry $i : V_{G(M)} \hookrightarrow V_{H(M')}$ such that the equation*

$$G \setminus H \setminus (H \setminus G)^2 = \emptyset \tag{13}$$

is satisfied.

PROOF. If M' contains a variation of M which is obtained by adding no more than one note between two consecutive notes of M, we will therefore have a graph inclusion $i : G(M) \hookrightarrow H(M')$ such that the set A_H will be partitionable into classes formed by trails η such that

$$\eta = i_A(a) = h,\ h \in A_H \tag{14}$$

or

$$\eta = i_A(a) = h_1 h_2\ , \tag{15}$$

$$with\ \partial_1 h_1 = \partial_0 h_2\ \ h_i \in (A_H \setminus A_{i(G)}). \tag{16}$$

In the first case we have $\eta \in A_H \cap A_{i(G)}$, though in the second case $\eta \in A_{(H \setminus i(G))^2}$. Hence Eq. (13) is satisfied.

Conversely, if there exists an isometry $i : V_{G(M)} \hookrightarrow V_{H(M')}$ such that it satisfies Eq. (13), we will have $G \setminus H \subseteq (H \setminus G)^2$; hence $\forall a \in i(A_G)$ will be

$$a \in A_H \cap A_{i(G)} \quad \vee \quad a = h_1 h_2\ , \tag{17}$$

$$with\ \partial_1 h_1 = \partial_0 h_2\ \ h_i \in (A_H \setminus A_{i(G)}). \tag{18}$$

Therefore we can partition A_H following the definition of graph inclusion (see Appendix B.2). □

Finally, we have obtained an operative, sufficient, and necessary condition for the inclusion which can be usefully implemented into an algorithm for the recognition of inclusions.

3.6 Similarity Function

Now let us give the notion of a similarity function between graphs which will be useful to estimate the similarity between two themes. We note that this function applies only to themes that have passed the filtering of conditions order, size, and complexity.

Up to now we have introduced the concept of a representative graph of a melody, together with its implications as to the recognition of standard and nonstandard music transformations.

From a musical perspective, the transformations we are looking for are those preserving the structure of the theme (i.e., numerical relationships between notes). However, as we pointed out when introducing the notion of inclusion of themes, we want to derive a straightforward concept of similarity which permits to match both the structure of the themes and other possible differences between melodies due to note insertion. In fact, a melody M' may have nearly the same structure of another melody M in the sense that M' contains both the notes of M (exactly or transformed through the aforementioned transformations) and other notes (e.g., passing tones or extraneous notes) not contained in M.

We could compare this comparison process to string matching algorithms when the latter match a string A with a string B that contains A plus other symbols inserted between the symbols of A. In the matching phase, different weights (α_i) can be assigned to the presence in B of $1, 2, 3, \ldots$ extraneous symbols between pairs of symbols of A.

Thus, we define a similarity function whose domain coincides with the set of musical graphs and that provides a (normalized) degree of similarity between two themes M and M'. The function works as follows:

—Both representative graphs G and H of the two themes are considered;
—all possible isometries between the vertex sets V_G and V_H are computed;
—all the differences between graph G and the powers of H are computed (remember that the ith graph power contains the number of trails of length i);
—the last differences are weighted by the coefficients α_i, where coefficient i weights differently the presence of $1, 2, \ldots, i, \ldots, r$ passing tones that may have been inserted between note pairs of G. The weights are set in accordance to the musicological aim, which can assign, for example, low values to the insertion of single or odd numbers of notes (which can be considered trills on notes), but high values to even numbers of notes; and
—this function, that depends upon the selected isometry, is maximized over the isometry set in order to find the best matching value.

Definition 3.31. Let M and M', $M \leq M'$, be two themes with representative graphs $G = G(M)$ and $H = H(M)$. Then, given $r \in \mathbf{N}$, we call the *similarity function of order r between M and M'* the form

$$\sigma(M, M') = \sigma(G, H) = \max_{\phi} \sum_{i=1}^{r} \alpha_i \frac{|G| - \left|G \setminus H^i \setminus H^{i-1}\right|}{|G|}, \tag{19}$$

where $H^0 = \emptyset$, α_i are positive coefficients which depend upon the weight assigned to the different trail lengths and ϕ varies among all possible isometries from V_G to V_H.

Remark 3.32. The form $1/\sigma$ is not a metric on the set of themes because $\sigma(M, M') \neq \sigma(M', M)$.

Remark 3.33. The computation of the similarity function depends a lot upon the maximum order of graph powers required by the user. Here we do not address this topic because we are not dealing with calculation algorithms. We want to point out that a similarity function should be calculated for a relatively small number of themes because of the presence of the filtering phase by invariants, which is the main topic of our work.

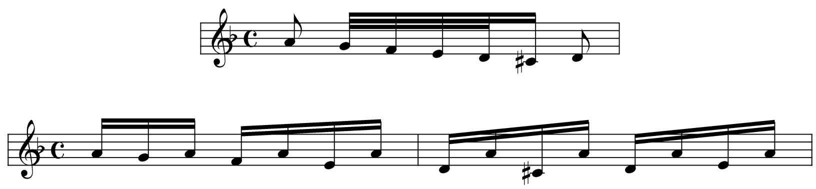

Fig. 27.

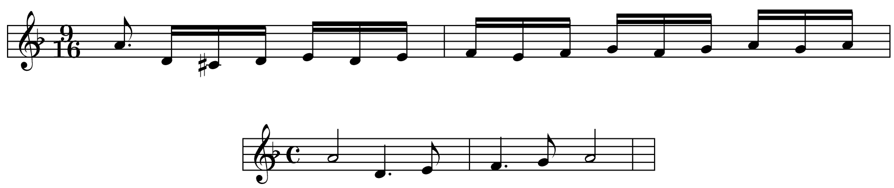

Fig. 28.

Example 3.34. Let us consider the themes in Figure 27 [Bach 1976]. The first theme (A) is clearly contained in the second (B) and the function which realizes the max is the identity function. Let us calculate the similarity function with $r = 2$ and $\alpha_1 = \alpha_2 = 1$. In this example it is evident that the max can be realized only by the identity function, thus we have

$$\begin{aligned}\sigma(A, B) &= \max_{\phi} \sum_{i=1}^{2} \alpha_i \frac{|G| - \left|G \setminus H^i \setminus H^{i-1}\right|}{|G|} = \\ &= 1\frac{|G| - \left|G \setminus H^2 \setminus H\right|}{|G|} + 1\frac{|G| - |G \setminus H \setminus \emptyset|}{|G|} = \\ &= 1\frac{7-6}{7} + 1\frac{7-1}{7} = \frac{1}{7} + \frac{6}{7} = 1. \end{aligned} \tag{20}$$

Example 3.35. Now, consider the two themes in Figure 28 [Bach 1976]. The second (B), as we have already pointed out in Section 3.3, is even a Eulerian subgraph of the first (A) because its intervals are present also in the first.

Let us calculate the similarity function with $r = 2$ and $\alpha_1 = \frac{1}{2}$, $\alpha_2 = 1$. Like in the previous example, it is evident that the max can be realized only by the identity function, thus we have

$$\begin{aligned}\sigma(A, B) &= \max_{\phi} \sum_{i=1}^{2} \alpha_i \frac{|G| - \left|G \setminus H^i \setminus H^{i-1}\right|}{|G|} = \\ &= \frac{1}{2}\frac{|G| - \left|G \setminus H^2 \setminus H\right|}{|G|} + 1\frac{|G| - |G \setminus H \setminus \emptyset|}{|G|} = \\ &= \frac{1}{2}\frac{6-6}{6} + 1\frac{6-0}{6} = 0 + 1 = 1. \end{aligned} \tag{21}$$

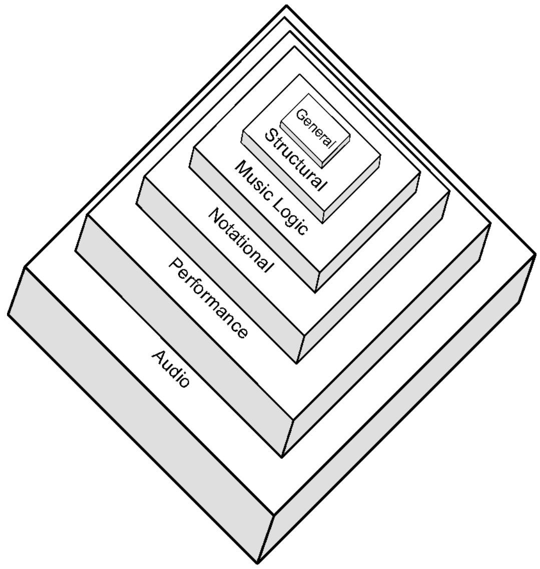

Fig. 29. Layer hierarchy.

4. MODEL IMPLEMENTATION IN MX

4.1 MX Layers

XML organizes information in a hierarchic structure, so MX represents each layer as a secondary branch of the source element. The conceptual hierarchy is represented in Figure 29. We are not interested here in describing all MX layers; for a complete treatment of the subject, see, for example, Haus and Longari [2005]. We will limit our description to the structural layer, after a short overview of the general layer.

```
<!ELEMENT mx (general, structural?,
             logic, notational?,
             performance?, audio?)
>
```

4.1.1 *General.* In this layer musical information is described as a whole. It provides a general description of a musical piece and contains information about possible connected instances. The following is its definition.

```
<!ELEMENT general (description, casting?,
         related_files?, analog_media?,
         notes?, rights?)
>
<!ELEMENT description (work_title?,
         work_number?, movement_title,
         movement_number?, genre?,
         author*)
```

```
>
<!ELEMENT genre (genre_spec+)>
<!ELEMENT genre_spec EMPTY>
<!ATTLIST genre_spec
          name CDATA #REQUIRED
          description CDATA #IMPLIED
          weight CDATA #IMPLIED
>
<!ATTLIST author
type CDATA #IMPLIED
>
<!ELEMENT author (#PCDATA)>
<!ELEMENT work_title (#PCDATA)>
<!ELEMENT work_number (#PCDATA)>
<!ELEMENT movement_title (#PCDATA)>
<!ELEMENT movement_number (#PCDATA)>
<!--
description of the music event (genre, date, place);
-->
<!ELEMENT casting EMPTY>
<!-- casting information;-->
<!ELEMENT related_files (related_file+)>
<!--
the table of related music data files, referring to all layers, with one or
more files for the summarization of each layer;
-->
<!--
the table of related multimedia data files, such as images, videos, and the
like;
-->
<!ATTLIST related_file
file_name CDATA #REQUIRED
file_format \%formats; #REQUIRED
encoding_format \%formats; #REQUIRED
file_size_byte CDATA #IMPLIED
>
<!ELEMENT related_file EMPTY>
<!ELEMENT analog_media EMPTY>
<!--
the table of related analog media;
-->
<!--
technical  information about related import/export/restoring/cataloguing/
other operations;
-->
<!ELEMENT notes EMPTY>
<!-- general notes.-->
```

4.1.2 *Structural.* The structural level of information was developed in order to contain the explicit description of musical objects and their causal relationships, both from musicological and compositional points of view. In other words, musical objects can be described as transformations of previously described musical objects. Those objects are usually the result of segmentation processes made by different musicologists, together with their own different musical points of view, or also by an automatic score segmenter like Scoresegmenter [Haus and Sametti 1991; Haus and Pollastri 2000].

This is a kind of description that shows causality connections instead of temporal connections. The information contained in this layer does not refer to temporal ordering and absolute time instances; on the contrary, it describes the causal relationships by transformations and displacements of musical objects in the score, as they derive from the analysis/synthesis framework.

At the moment there is not a definitive standard for this layer and our efforts are especially directed towards the definition of a common acceptable one. In this framework we believe that the introduction in this MX layer of MIR-oriented metadata as attributes of `Theme` could substantially improve MIR systems.

4.1.3 *Theme Recognition. Themes* are the result of a musicological segmentation process that can be automatic or human-based. A relevant number of automatic score-segmenters and theme extractors have been realized and many others are under development.

Our Scoresegmenter is based on algorithms for seeking out occurrences of music objects or their subfragments within a passage (i.e., objects that can be recognized by the computer itself or provided by the user).

The algorithm considers two elements: the attributes of the single note, and the music transformation the note has undergone, together with the previous and following notes. The attributes are timing, accent, note name, pitch in halftones, and intervals. As for transformations, here we have considered those applied to the attributes of position or degree, both in tonal and diatonic scale.

These transformations have been realized by means of three different types of algebraic operators and their combinations; precisely: transposition, mirror-inversion, and retrogradation operators. These operators realize the corresponding music tonal transformations when applied to the pitch in halftones, and to the real transformations when applied to the names of notes.

4.1.4 *Theme Invariants.* The graph model of melodic themes would be useful in MIR systems because it provides necessary conditions for the inclusion of thematic fragments by means of invariant quantities. Thus, invariants can be embedded as attributes of theme. It would certainly be possible to embed even more attributes corresponding to different formal approaches, like statistical ones, which we do not consider.

Now we list the part of DTD which deals with the structural layer and which also implements the concept of melodic theme.

```
<!-- Structural Layer -->

<!ELEMENT structural (analysis*, PN*)>

<!-- Melodic themes -->
<!ELEMENT analysis (theme*, segment*,
                    transformation*,
                    relationship*)
>
<!ATTLIST theme
          id ID #REQUIRED
          ordinal CDATA #IMPLIED
          desc CDATA #IMPLIED
>
<!--
Desc attribute provides a textual description of the theme.
Ordinal attribute describes the possible numeric characterization of the
theme. It should be encoded in roman numbers: I, II, III, IV, V, etc (e.g. I
and II themes in a Sonata.)
-->

<!ELEMENT theme (occurrence+)>
<!ELEMENT occurrence (thm_desc?,
          thm_spine_ref+,
          (transposition | inversion
          | retrogradation)*)
>
<!ATTLIST occurrence
          id ID #REQUIRED
>

<!ELEMENT thm_spine_ref EMPTY>
<!--
This element is needed for generalization of theme representation,
since there could be themes split in different sequences of notes belonging
to the same part or even to different parts.
-->
<!ATTLIST thm_spine_ref
          spine_start_ref IDREF #REQUIRED
          spine_end_ref IDREF #REQUIRED
          part_ref IDREF #REQUIRED
          voice_ref IDREF #REQUIRED
>
<!ELEMENT thm_desc (#PCDATA)>
<!ELEMENT transposition EMPTY>
<!--
Interval is an integer number, indicates the interval of transposition
```

```
and its interpretation is related to the type attribute. When type is real
interval indicates the distance in semitones, otherwise it indicates
 distance in tonal scale.
-->
<!ATTLIST transposition
          type (real | tonal) #REQUIRED
          interval CDATA #REQUIRED
>
<!ELEMENT inversion EMPTY>
<!--
Staffstep attribute must have the same interpretation as the staff_step
attribute of noteheads.
-->
<!ATTLIST inversion
          type (real | tonal) #REQUIRED
          staff_step CDATA #REQUIRED
>
<!ELEMENT retrogradation EMPTY>

<--
Invariants are quantities that refer to a theme and do not vary
even if the theme is transformed by canonical transformations
-->

<!ELEMENT theme (invariants?, graph?)>
<!ELEMENT invariants (order, size, complexity)>
<!ELEMENT order (#PCDATA)>
<!ELEMENT size (#PCDATA)>
<!ELEMENT complexity (#PCDATA)>
<!ELEMENT graph (#PCDATA)>

<!-- Petri Nets -->
<!ELEMENT PN EMPTY>
<!ATTLIST PN
          file_name CDATA #REQUIRED
>
<!-- Segments -->
<!ATTLIST relationship
          id ID #REQUIRED
          segmentAref IDREF #REQUIRED
          segmentBref IDREF #REQUIRED
          transformationref IDREF #REQUIRED
>
<!ELEMENT relationship EMPTY>
<!ATTLIST segment
id ID #REQUIRED
>
```

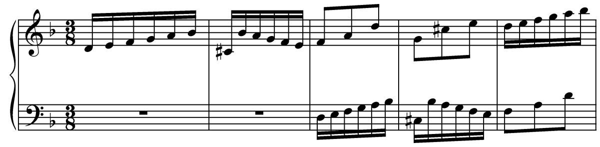

Fig. 30. Example: notational layer.

```
<!ELEMENT segment (segment_event+)>
<!ELEMENT transformation EMPTY>
<!ATTLIST transformation
          id ID #REQUIRED
          description CDATA #REQUIRED
          gis CDATA #IMPLIED
>
<!ATTLIST segment_event
          id_ref IDREF #REQUIRED
>
```

4.1.5 *Invariant Representation.* As we described in analyzing the model, there are various necessary conditions for the inclusion of themes. Inclusion-monotone invariants play an important role in the retrieval process and are necessary in order to delete any comparison which could surely fail. Those quantities, which preserve their values even if the fragment changes by the application of isometries, are extremely useful to deal with themes, together with all their transformations.

In the XML formalism, an invariant should have the form of a subelement of a `Theme`: this mainly for reason of visibility and better retrieval. Graph matrices can be obtained straightforwardly from the theme's encoding and in our first model this is worked out by the similarity engine. We have introduced the possibility of storing the adjacency matrix in the subelement "graph" of the element "theme".

4.2 Example

In this example of an MX file it is possible to identify the `general`, `structural`, and `logic` layers. In the structural layer it is possible to identify the subelement `invariants` and its subelements `order`, `size`, and `complexity`.

```
<?xml version="1.0" encoding="UTF-8"?>
<mx>
    <general>
         <description>
              <movement_title>Inventio #4
              </movement_title>
         </description>
    </general>
    <structural>
         <themes>
```

```
          <theme id="theme0">
               <thm_spineref endref="v1_12"
                    partref="part0" stafref="v1_0"
                    voiceref="voice0" />
               <thm_spineref endref="v2_12"
                    partref="part1" stafref="v2_0"
                    voiceref="voice0" />
               <invariants>
                    <order="7">
                    <size="12">
                    <complexity="54">
               </invariants>
          </theme>
     </themes>
</structural>
<logic>
     <spine>
     <event id="timesig_0" timing="0" hpos="0"/>
     <event id="keysig_0" timing="0" hpos="0"/>
     <event id="clef_0" timing="0" hpos="0"/>
     <event id="clef_1" timing="0" hpos="0"/>
     <event id="p1v1_0" timing="0" hpos="0"/>
     <event id="p1v1_1" timing="256" hpos="256"/>
     <event id="p1v1_2" timing="256" hpos="256"/>
     <event id="p1v1_3" timing="256" hpos="256"/>
     <event id="p1v1_4" timing="256" hpos="256"/>
     <event id="p1v1_5" timing="256" hpos="256"/>
     <event id="p1v1_6" timing="256" hpos="256"/>
     <event id="p1v1_7" timing="256" hpos="256"/>
     <event id="p1v1_8" timing="256" hpos="256"/>
     <event id="p1v1_9" timing="256" hpos="256"/>
     <event id="p1v1_10" timing="256" hpos="256"/>
     <event id="p1v1_11" timing="256" hpos="256"/>
     <event id="p1v1_12" timing="256" hpos="256"/>
     <event id="p2v1_0" timing="0" hpos="0"/>
     <event id="p2v1_1" timing="256" hpos="256"/>
     <event id="p1v1_13" timing="256" hpos="256"/>
     <event id="p2v1_2" timing="0" hpos="0"/>
     <event id="p2v1_3" timing="256" hpos="256"/>
     <event id="p1v1_14" timing="256" hpos="256"/>
     <event id="p2v1_4" timing="0" hpos="0"/>
     <event id="p2v1_5" timing="256" hpos="256"/>
     <event id="p1v1_15" timing="256" hpos="256"/>
     <event id="p2v1_6" timing="0" hpos="0"/>
     <event id="p2v1_7" timing="256" hpos="256"/>
     <event id="p1v1_16" timing="256" hpos="256"/>
     <event id="p2v1_8" timing="0" hpos="0"/>
```

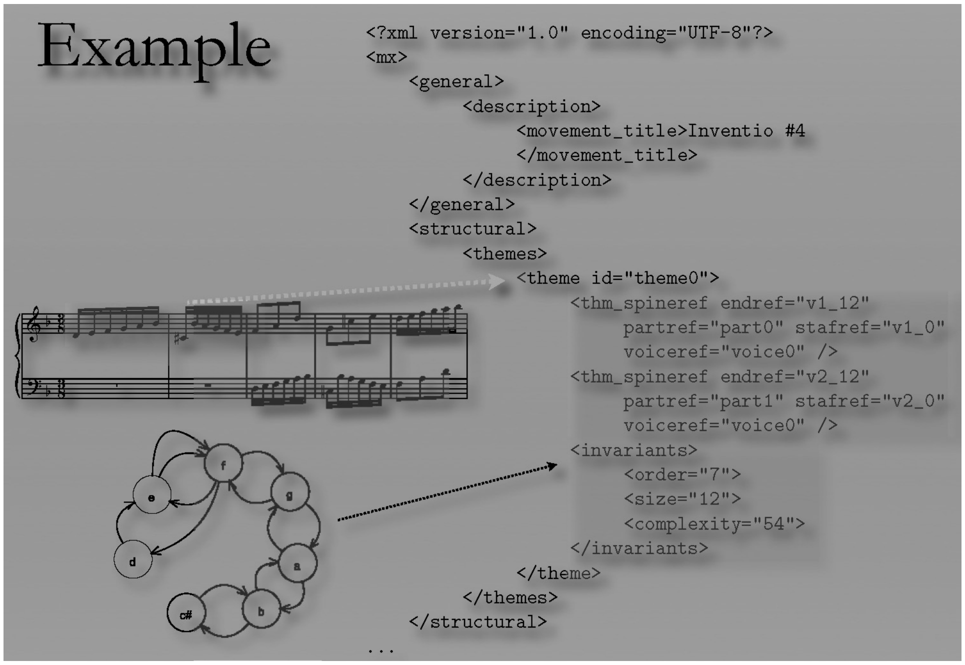

Fig. 31. Embedding of MIR metadata in an MX environment.

```
        <event id="p2v1_9" timing="256" hpos="256"/>
        <event id="p1v1_17" timing="256" hpos="256"/>
        <event id="p2v1_10" timing="0" hpos="0"/>
        <event id="p2v1_11" timing="256" hpos="256"/>
        <event id="p1v1_18" timing="256" hpos="256"/>
        <event id="p2v1_12" timing="0" hpos="0"/>
        <event id="p1v1_19" timing="256" hpos="256"/>
        <event id="p1v1_20" timing="256" hpos="256"/>
        <event id="p2v1_13" timing="0" hpos="0"/>
        <event id="p1v1_21" timing="256" hpos="256"/>
        <event id="p1v1_22" timing="256" hpos="256"/>
        <event id="p2v1_14" timing="0" hpos="0"/>
        <event id="p1v1_23" timing="256" hpos="256"/>
        </spine>
      ...
</mx>
```

5. EXPERIMENTATION

The behavior of a distance measure can be evaluated using a lot of very different criteria. One possible approach could be to gather information about human similarity ratings of musical material, and then see how close a certain distance measure is to the human ratings.

This is not our approach, essentially for two reasons. First of all, this approach has the practical disadvantage that it may be hard to obtain the necessary empirical data and, moreover, there is no proof of the existence of a universally accepted "ground truth" concerning the problem of musical similarity. The other point is that we are interested in structural similarity that could be easily dismissed by any music nonexpert. More pragmatically, we choose to compare the ratings of graph distance and other functional distance measures in recognizing structural similaiirities and to investigate possible differences in features like discriminating power.

The comparison of the different distance measures was performed using a database of musical themes from a catalog of 100 works by J. S. Bach and W.A. Mozart, with a mean duration of 5 bars and ordered by catalog number. The catalog also included all the incipits of The Art of Fugue counterpoints, which correspond to tracks 69–83, in order to test the effectiveness with standard musical transformations like transpositions, inversions, and structural similarities.

With a subset of 90 themes (excluding The Art of Fugue in order to avoid too many similar ones) we performed all the possible pair-wise comparisons using 3 different similarity measures: the graph similarity function of Section 3.6 and two functional metrics, namely absolute and Sobolev second-order. The first is quite rough and the second has the advantage of being transposition invariant.

5.1 Invariant Evaluation

Graph invariants are very important in filtering processes because of their monotonicity with note insertion, as pointed out in Section 3.3. As for the recognition of inclusion, order, size, and complexity are used as necessary conditions.

One issue in evaluating how such conditions can be useful in practical MIR is the statistical distribution of these 3 graph invariants on a significant set of themes. In fact, if they were closely clustered, for most themes the contribution for retrieval would be very limited. From Figures 32, 33, and 34 we can argue that they are not closely clustered. In Figure 32 graph-order clusters are plotted and here we can argue something about the genre of music incipits contained in the database. Specifically, graph order tells us how many different pitches are present in the melodies so that, for example, a database full of dodecaphonic music would be plotted as a high peak at value 12.

Another important factor in the evaluation process is to show the independence of order, size, and complexity. This can be achieved by analyzing the correlation matrix of Eq. (22), whose entries (x, y) give the correlation value between invariant x and invariant y. The low values of the off-diagonal elements show that they are low-correlated, thus graph invariants can be used independently as filtering parameters in order to retrieve variations of themes.

$$\rho = \begin{bmatrix} 1,0000 & 0,2106 & 0,2999 \\ 0,2106 & 1,0000 & 0,5033 \\ 0,2999 & 0,5033 & 1,0000 \end{bmatrix} \tag{22}$$

In Figure 35 we show a comparison of invariant values against the set of 90 themes from our database.

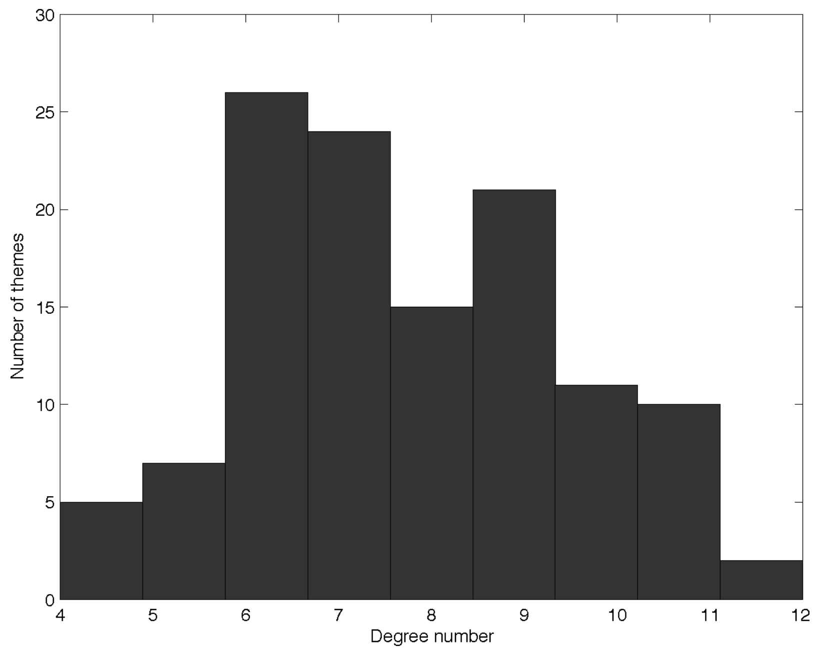

Fig. 32. Graph order evaluated on the database of musical incipits. The figure shows the number of themes as a function of clustered values of graph order.

5.2 Similarity Measure Evaluation

In order to verify the effectiveness of the graph similarity measure in discriminating similar themes, we decided to compare a famous theme (J. S. Bach's *The Art of Fugue*) with all other theme of the database.

In the database there were also the themes in Figure 39: They are variations of the theme of the The Art of Fugue, preserving the same structure. In particular, the first two themes come from the third by applying two pitch permutations: (d,g) for the first and (a,e) for the second.

The graph similarity measure returns zero for all the themes in Figure 39 because the corresponding graphs are all isomorphic. On the other hand results for the absolute and Sobolev second-order measures are not satisfying for Figure 39 themes, which are all identically structured, nor for a subpermuted one (track 70). In fact, tracks 66, 67, 69, and 70 correspond to those themes, but Figures 37 and 38 show that all functional metrics provide nonzero values and even false positives (e.g., tracks 81 and 86, which are completely differently-structured themes).

Figure 36, on the contrary, evidences better performance of the graph metric in this case, as it returns a zero value.

Moreover, it can be trivially proved that whereas the distance distributions of the Sobolev second-order and absolute metrics spread across the spectrum, the graph measure has much more clustered values for similar themes.

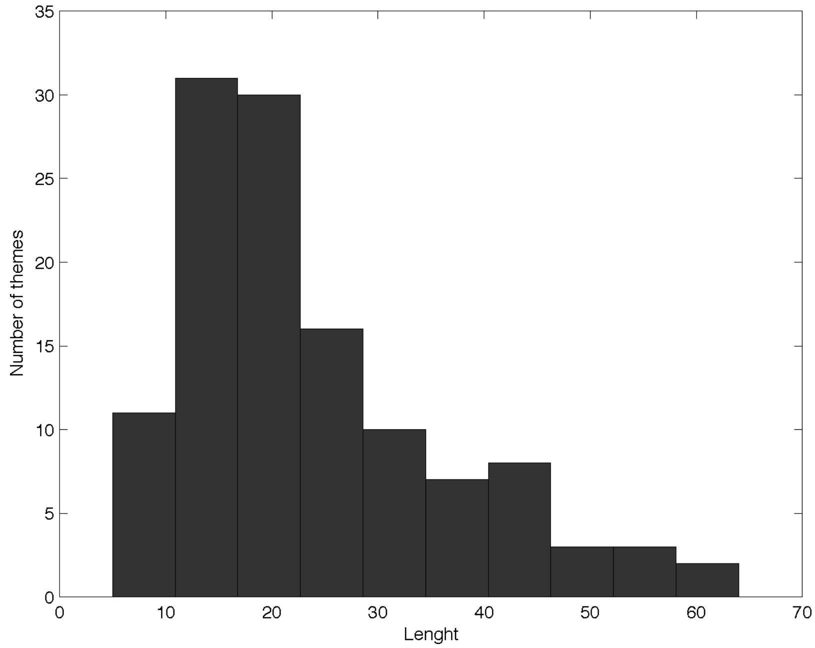

Fig. 33. Graph size evaluated on the database of musical incipits. The figure shows the number of themes as a function of clustered values of graph size.

6. SUMMARY AND CONCLUSION

Invariants reveal themselves as useful tools in MIR processes in order to reduce unproductive comparisons, especially in the exact match case. MX is a powerful framework in which invariants should be embedded because of the presence of a structural layer and a standard definition of musical theme to which invariant quantities are linked in a very natural way.

Moreover, the graph model presented here enlarges the similarity class of thematic fragments with respect to other models known in the literature. This proves useful in the case of mismatched starting points of compared themes. Moreover, we observe that the more a theme presents variety in melodic and interval construction, the smaller becomes its Euler-equivalence class. Conversely, themes with repetitions tends to be more similar, increasing the cardinality of their Eulerian class, so the graph model results cohere with the common musical intuition.

7. FUTURE WORK

The approach to MIR by invariants can be developed towards the enrichment of the invariant class derived from graphs or from other models. In particular, the graphical approach can be developed towards numerous directions: first of all, increasing the number and power of necessary conditions which are musically significant. A possible extension of the model should integrate rhythmic and accentual dimensions of themes.

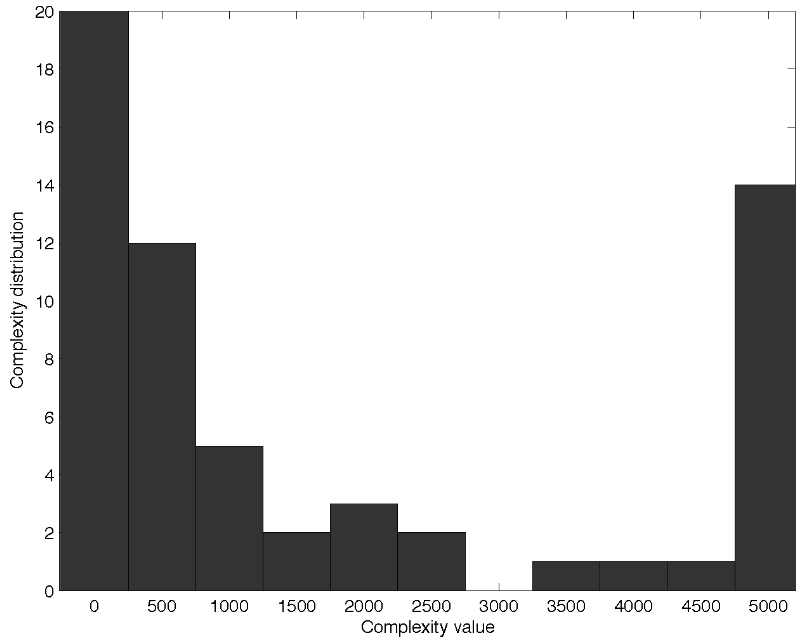

Fig. 34. Graph complexity evaluated on the database of musical incipits. The figure shows the number of themes as a function of clustered values of graph complexity.

APPENDICES

A. STANDARD NOTIONS AND RESULTS OF GRAPH THEORY

In this appendix we provide some notions of graph theory in order to make the article self-contained. Basic notions can also be found in books of graph theory like Bollobás [1998] and Godsil and Royle [2001]. Other definitions and propositions, such as graph operations and the monotonicity of complexity, are original and will be provided in Appendix B.

Definition A.1. A *finite oriented multigraph* G is a 4-tuple $G = \{V_G, A_G, \partial_0, \partial_1\}$, where V_G is a set of objects v_i, $i \in I = \{1, 2, \ldots, n\}$ (the *vertices*), A_G a set of arrows a_i, $i \in I = \{1, 2, \ldots, m\}$ (the *edges*), and ∂_0, ∂_1 is a function pair $\partial_0 : A_G \to V_G$ and $\partial_1 : A_G \to V_G$.

Definition A.2. A *graph morphism* $f : G \to G'$ is a function pair $f_A : A_G \to A_{G'}$ and $f_V : V_G \to V_{G'}$ such that the following diagram commutes.

$$\begin{tikzcd} A_G \arrow[r, "f_A"] \arrow[d, "\partial_0, \partial_1"'] & A_{G'} \arrow[d, "\partial'_0, \partial'_1"] \\ V_G \arrow[r, "f_V"'] & V_{G'} \end{tikzcd} \tag{23}$$

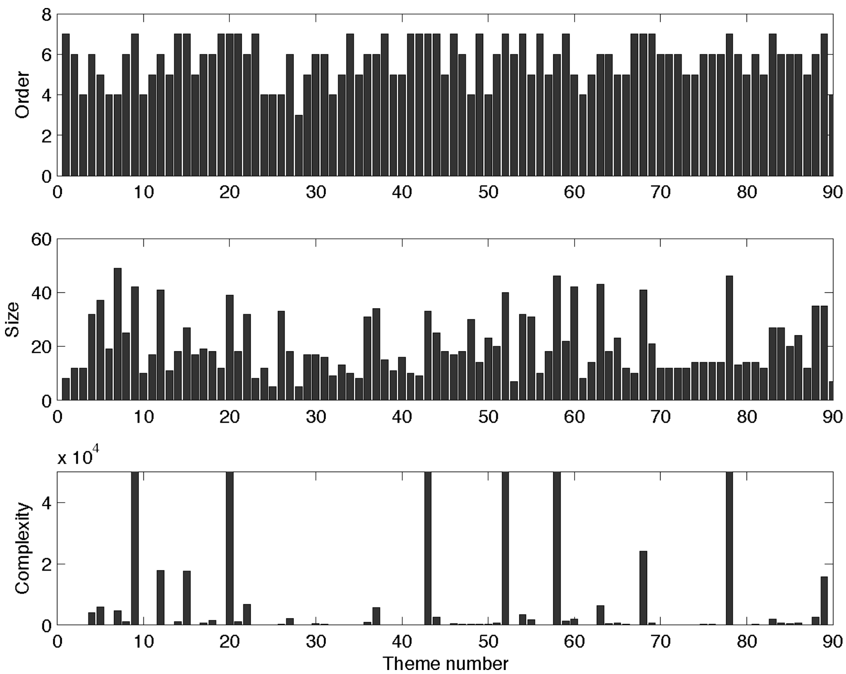

Fig. 35. Graph invariants evaluated on a subset of the database of musical incipits. The figure shows invariant values for each theme. Note the wide range of complexity values (10^4).

Definition A.3. We say that the graph G is *included* in graph G' if there is a morphism $f : G \to G'$ such that f_A and f_G are one-to-one correspondences. Further, $f(G)$ is said to be a *subgraph* of G'.

Definition A.4. If G and G' are two graphs, the *difference graph* $G \setminus G'$ is the graph obtained by G deleting all the arrows in common with G' (i.e., with the same source and target).

Definition A.5. A *spanning tree* of a graph is a subgraph that contains all the vertices and is a tree.

Definition A.6. The *complexity* $k(G)$ of a graph G is the number of equally oriented spanning trees of G.

Definition A.7. The *incidence matrix* of a graph is an L by N matrix, where N is the number of vertices in the graph (the *order* of the graph) and L the number of edges (the *size* of the graph). Rows are in one-to-one correspondence with arrows and each row contains, respectively, a -1 and a 1 corresponding the start-vertex and end-vertex of the arrow; the other elements are zeros.

Definition A.8. The *adjacency matrix* of a graph is an N by N matrix whose entries are as follows: if there are k edges from some vertex x to some vertex y, then the element $M_{(x,y)}$ is k, otherwise it is 0.

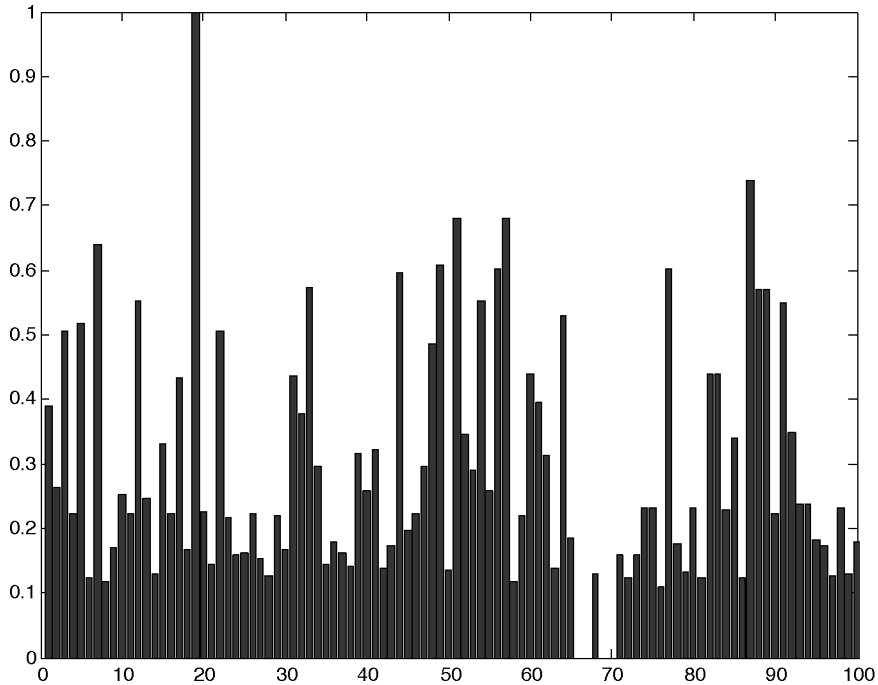

Fig. 36. Results of the comparison of track 69 with the graph similarity measure.

Definition A.9. The *Laplacian matrix* $Q(G)$ of a graph G is the product of the incidence matrix with its transposed matrix.

$$Q(G) = D(G)D_T(G) \tag{24}$$

Definition A.10. A *Eulerian path* or *Eulerian trail* is a path that uses each edge exactly once. If such a path exists, the graph is called *traversable*.

Definition A.11. A *Eulerian cycle* or *Eulerian circuit* is a cycle that uses each edge exactly once. If such a cycle exists, the graph is called *Eulerian* or unicursal.

Definition A.12. A *Hamiltonian path* is a path which visits each vertex exactly once. A *Hamiltonian cycle* is a cycle which visits each vertex exactly once and also returns to the starting vertex.

Definition A.13. A *Hamiltonian cycle*, *Hamiltonian circuit*, *vertex tour*, or *graph cycle* is a cycle that visits each vertex exactly once (excluding the start-/end-vertex). A graph that contains a Hamiltonian cycle is called a *Hamiltonian graph*.

Definition A.14. A graph is *connected* if every two vertices can be connected by a walk.

Definition A.15. Given an oriented graph G, the *opposite graph* G^{op} is the graph obtained from G reversing the orientation of all the arrows.

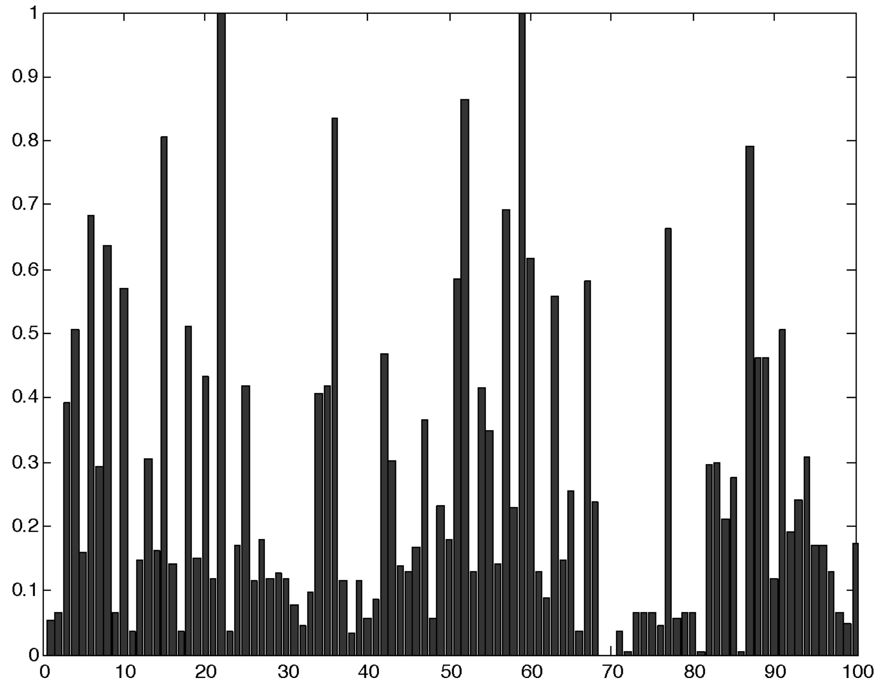

Fig. 37. Results of the comparison of track 69 with the absolute metric.

Definition A.16. A *graph invariant* of a graph G is a number associated with G which has the same value for any graph isomorphic to G.

Remark A.17. Order, size, and complexity are graph invariants.

The following Cayley theorem links the complexity to the Laplacian matrix.

THEOREM A.18. (CAYLEY). *Let D be the incidence matrix of a graph G and let Q be the Laplacian matrix. Every algebraic complement of Q equals the complexity of G.*

A.1 Power Graphs

In this section we will recall some results about the powers of a graph which will be useful in the closure operations.

THEOREM A.19. *Let G be a labeled graph with adjacency matrix A. Then $a_{i,j}^{(n)} \in A^n$ is the number of trails of length n from v_i to v_j.*

PROOF. The proof proceeds by induction on n. The result is obvious for $n = 1$ because $a_{i,j}^{(1)} = a_{i,j}$ is the number of arrows from v_i to v_j by definition. Now we suppose that the thesis is true for $(n - 1)$, namely, suppose $a_{i,j}^{(n-1)}$ is the number of distinct trails from v_i to v_j. Since $A^n = A^{n-1} \cdot A$, its elements $a_{i,j}^{(n)}$ are

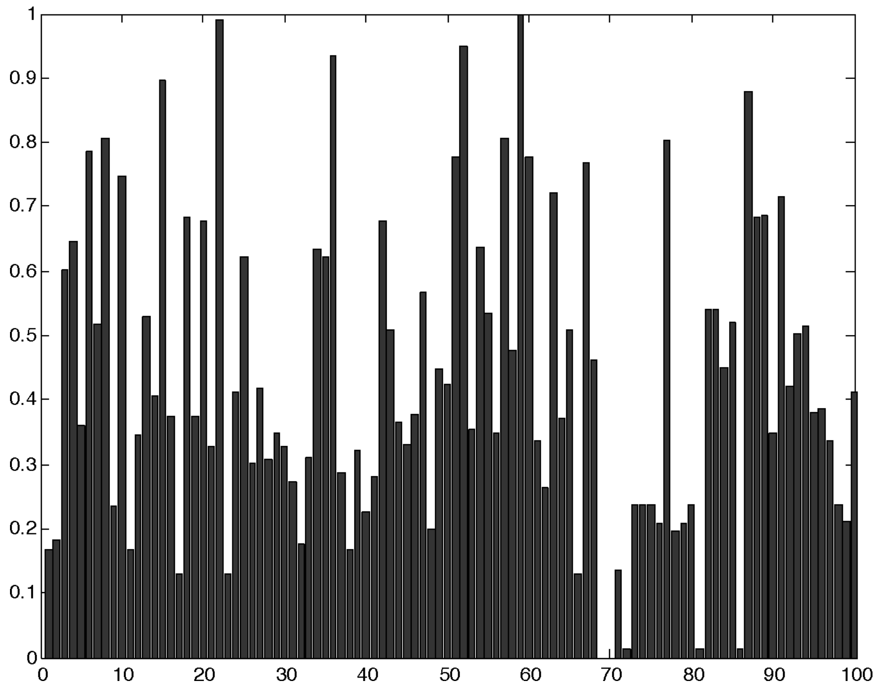

Fig. 38. Results of the comparison of track 69 with the Sobolev second-order metric.

Fig. 39. Example of identically structured themes.

given by

$$a_{i,j}^{(n)} = \sum_{k=1}^{p} a_{i,k}^{(n-1)} a_{k,j}, \tag{25}$$

where p is the number of the G vertex.

Since every n-trail from v_i to v_j is composed by an $(n-1)$-trail from v_i to some node v_k followed by an arrow from v_k to v_j, the inductive hypotesis and the Eq. (25) prove the theorem. □

Corollary A.20. *In an unoriented graph G, the elements $a_{i,i}^2$ of the diagonal of $A^2(G)$ equal the vertex v_i degree.*

Remark A.21. Being the adjacency matrix A of a graph G a square matrix, we can consider its determinant. We observe that $\det A$ is independent by vertex labeling, thus the results are invariant by arbitrary permutations of vertices. Particularly, if two vertexes have the same neighborhood, then $\det A(G) = 0$; this is due to the fact that in this case the two rows of A would be the same.

A.2 Eulerian Graphs

Here is an important theorem, due to Cayley, about the number of Eulerian circuits of a Eulerian graph G.

THEOREM A.22. *Let G be a Eulerian connected oriented graph. The number $s(G)$ of Eulerian circuits is*

$$s(G) = c \cdot \prod_{i=1}^{n} (v_i^+ - 1)!, \tag{26}$$

where c is the number of equally oriented spanning trees of G.

PROOF. Refer to, for example Bollobás [1998]. □

B. NONSTANDARD GRAPH-THEORETIC NOTIONS

Definition B.1. A *musical graph* G is a finite, connected, Eulerian multigraph such that V_G is endowed with a metric space structure (V, d).

B.1 Operation on Graphs

B.1.1 *Graph Inclusion.* We introduce the novel notion of graph inclusion that we use throughout the article.

Definition B.2. A graph G is *included* in a graph G' if $V_G \subseteq V_{G'}$ and exists a partition $\mathcal{P}(A_{G'})$ of the arrow set of G', that is, $A_{G'}$, in trails such that the diagrams

$$\begin{tikzcd} A_G \arrow[r, "i_A"] \arrow[d, "\partial_0, \partial_1"'] & \mathcal{P}(A_{G'}) \arrow[d, "\partial'_0, \partial'_1"] \\ V_G \arrow[r, "i_V"'] & V_{G'} \end{tikzcd} \tag{27}$$

commute, where $i = (i_A, i_V)$ is the usual graph inclusion.

Remark B.3. If the partition is *the finest* one, namely, all classes are singleton, the definition collapse, to standard inclusion.

Remark B.4. The partition may not be unique.

Example B.5. Let us consider the graph on three vertexes C_3 and the union graph $U = C_4 \sqcup C_3^{op}$ obtained from the cyclic graph C_4 and C_3^{op}.

$$\mathcal{D}(C_3) = \begin{bmatrix} -1 & 0 & 1 \\ 1 & -1 & 0 \\ 0 & 1 & -1 \end{bmatrix} \tag{28}$$

$$\mathcal{D}(U) = \begin{bmatrix} -1 & 0 & 0 & 1 & -1 & 0 & 1 \\ 1 & -1 & 0 & 0 & 0 & 1 & -1 \\ 0 & 1 & -1 & 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 \end{bmatrix} \tag{29}$$

The arc-partitions

$$\mathcal{P}_1(A_U) = \{\{1\}, \{2\}, \{3, 4\}, \{5, 6, 7\}\} \tag{30}$$

and

$$\mathcal{P}_2(A_U) = \{\{1\}, \{2\}, \{3, 4, 5, 6, 7\}\} \tag{31}$$

are both admissible in order to define an inclusion of C_3 in U. Thus, the partition is not unique in general.

PROPOSITION B.6. *The inclusion defined previously is a partial-order relation.*

PROOF. Reflexivity and transitivity are obvious. Using the injectivity of i and of its inverse, $V_G = V_{G'}$ and $A_G = A_{G'}$. So, by the commutativity of diagram (27), we have proved the antisimmetry. □

Remark B.7. The inclusion defined before works with labeled graphs. It may be useful to enlarge this concept to the situation where V_G is "nearly" included in $V_{G'}$. In fact, in musical graphs it is not relevant to preserve the labels because we study objects that are invariant by musical tranformations (i.e., permutation of labels preserving the metric structure defined on V_G).

Thus, the next definition formalizes this notion of "near inclusion".

Definition B.8. A graph G is *weakly included* in a graph H iff there exists a subgraph G' of H isomorphic to G.

$$G \cong G' \subseteq H \tag{32}$$

B.1.2 *Graph Union.* Let us consider a pair of graphs G and H.

Definition B.9. We define the *union* of G and H as the graph $G \sqcup H$ such that $V_{G \sqcup H} = V_G \sqcup V_H$ and $A_{G \sqcup H} = A_G \sqcup A_H$.

Remark B.10. If G and H are connected Eulerian graphs, it may be convenient to assume $V_G \sqcap V_H \neq \emptyset$. Doing so that, the result is equally connected and Eulerian.

PROPOSITION B.11. *The union operator defines the minimum superior bound $G \vee H$ for every pair of graphs G and H with respect to the relation of (standard) inclusion (see Appendix* A*).*

Proof. If a graph I containing G and H is contained in $G \vee H$, it has to be $V_I = V_{G\vee H}$ because it also holds that $V_G \subseteq V_I$, $V_H \subseteq V_I$ and $V_I \subseteq V_{G\vee H}$.

By the arc axiom, it should be that $|A_G|+|A_H| \leq |A_I| \leq |A_{G\vee H}| = |A_G|+|A_H|$. Then $I = G \vee H$. □

B.2 Graph Complexity

The following three propositions evidence the importance of graph complexity as monotone invariant with respect to the inclusion relation.

Proposition B.12. *Let H be the graph obtained from a graph G, substituting an arrow* $a : v_i \rightarrow v_j$ *with an arrow pair* $a_1 : v_i \rightarrow v_k$ *and* $a_2 : v_k \rightarrow v_j$*, with* $v_i, v_j, v_k \in V_G$*. Then*

$$k(H) \geq k(G). \tag{33}$$

Proof. Let T be a spanning tree which contains the arrow $a_1 : v_i \rightarrow v_k$. If we replace a with the trail a_1a_2, the result from T with the substitution of a by a_1 or by a_2 is yet a spanning tree, thus the complexity of G can only increase. □

Proposition B.13. *Let H be the graph obtained from a graph G of order* $n(G)$*, adding a vertex* v_{n+1} *to* V_G *and replacing an arrow* $a : v_i \rightarrow v_j$ *with the arrow pair* $a_1 : v_i \rightarrow v_{n+1}$ *and* $a_2 : v_{n+1} \rightarrow v_j$*. Then*

$$k(H) \geq k(G). \tag{34}$$

Proof. We can divide the spanning trees of G into two classes: X = {spanning trees containing a} and Y = {spanning trees which do not contain a}. Obviously, we have $k(G) = |X| + |Y|$. If $\alpha \in Y$, then $\alpha \vee a_2$ is a spanning tree of H. Thus the complexity of H is at least $|Y|$. Let us consider a tree $\beta \in X$. Replacing a by the trail a_1a_2 we obtain yet a spanning tree. Finally, we have $k(H) \geq |Y| + |X| = k(G)$. □

Proposition B.14. *Graph complexity is a monotonic function with respect to the order relation of graph inclusion previously defined.*

Proof. Let G be included in H following the standard notion of inclusion (see Appendix A). Then, if $V_G \subset V_H$, Corollary B.12 implies that $k(G) \leq k(H)$ and if $V_G \leq V_H$, Corollary B.13 implies $k(G) = k(H)$. Now let G be included weakly in H. The invariance of complexity under isomorphism and the previous case proves the theorem. □

REFERENCES

Aucouturier, J.-J. and Pachet, F. 2003. Respresenting musial genre: A state of the art. *J. New Music Res. 32*, 83–93.

Bach, J. S. 1976. *Orgelwerke*. Peters, Leipzig, Germany.

Baeza-Yates, R. and Ribeiro-Neto, B. 1999. *Modern Information Retrieval*. Addison-Wesley-Longman.

Baroni, M., Dalmonte, R., and Jacoboni, C. 1999. *Le Regole Della Musica*. EDT, Torino, Italy.

Bollobás, B. 1998. *Modern Graph Theory*. Springer, New York.

Buckley, F. and Harary, F. 1990. *Distance in Graphs*. Addison-Wesley.

DIANA, L. 2004. An XML-based querying model for MIR applications within a multilayered music information environment. Ph.D. thesis, Università degli Studi di Milano, Milano, I 20135. Also available as Università degli Studi di Milano, Department of Computer Science Rep.

GODSIL, C. AND ROYLE, G. 2001. *Algebraic Graph Theory*. Graduate Texts in Mathematics, vol. 207,Springer.

GOOD, M. 2001. MusicXML for notation and analysis. In *The VirtualScore: Representation, Retrieval, Restoration*. MIT Press, Cambridge, MA, 113–124.

HAUS, G. 1984. *Elementi di Informatica Musicale*. Jackson, Milano, Italy.

HAUS, G. AND LONGARI, M. 2005. A multi-layered, timebased music description approach based on XML. *Comput. Music J. 29*, 1 (Feb.), 70–85.

HAUS, G., LONGARI, M., AND POLLASTRI, E. 2004. A score-driven approach to music information retrieval. *J. Amer. Soc. Inf. Sci. Technol. 55*, 12, 1045–1052.

HAUS, G. AND POLLASTRI, E. 2000. A multimodal framework for music inputs (poster session). In *Proceedings of the 8th International Conference on ACM, Multimedia*. 382–384.

HAUS, G. AND SAMETTI, A. 1991. Scoresynth: A system for the synthesis of music scores based on petri nets and a music algebra. *IEEE Comput. 24*, 7, 56–60.

HEWLETT, W. B. AND SELFRIDGE-FIELD, E. 2000. Melodic similarity: Concepts, procedures, and applications. In *Computing in Musicology*, vol. 11. MIT Press, Cambridge, MA, 113–124.

HEWLETT, W. B. AND SELFRIDGE-FIELD, E. 2005. *Music Query*. MIT Press, Cambridge, MA.

LERDAHL, F. AND JACKENDOFF, R. 1996. *A Generative Theory of Tonal Music*. MIT Press, Cambridge, MA.

LONGARI, M. 2004. Formal and software tools for a commonly acceptable musical application using the XML language. Ph.D. thesis, Università degli Studi di Milano, Milano, IT 20135. Also available as Università degli Studi di Milano, Department of Computer Science Rep.

MONGEAU, M. AND SANKOFF, D. 1990. Comparison of musical sequences. *Comput. Humanities 24*, 3, 161–175.

PINTO, A. 2003. Modelli formali per misure di similarità musicale. M.S. thesis, Università degli Studi di Milano, Milano I 20135. Also available as Università degli Studi di Milano, Department of Mathematics Rep.

POLANSKY, L. 1992. More on morphological mutation functions: Recent techniques and developements. In *Proceedings of the International Computer Music Conference*.

POLANSKY, L. 1996. Morphological metrics. *J. New Music Res. 25*, 289–368.

ROADS, C. 1996. *The Computer Music Tutorial*. MIT Press, Cambridge, MA.

SCHOENBERG, A. 1911. *Harmonielehre*. Universal Edition, Leipzig, Germany.

TENNEY, J. AND POLANSKY, L. 1980. Temporal gestalt perception in music: A metric space model. *J. Music Theory 24*, 2, 205–41.

TSINARAKI, C., POLYDOROS, P., AND CHRISTODOULAKIS, S. 2004. Integration of OWL ontologies in MPEG-7 and TV-anytime compliant semantic indexing. In *(HDMS) the Hellenic Data Management Symposium*.

VERDI, L. 1998. Organizzazione delle altezze nello spazio temperato. In *Ensemble* '900 (Treviso, Italy).

Received October 2005; revised February 2007; accepted March 2007