Visualization of music collections based on structural similarity

Aurea Soriano, Fernando Paulovich, Luis Gustavo Nonato and Maria Cristina F. Oliveira
Institute of Mathematics and Computer Science
University of Sao Paulo, Sao Carlos, Brazil

asoriano@icmc.usp.br, paulovic@icmc.usp.br, gnonato@icmc.usp.br, cristina@icmc.usp.br

March 31, 2014

Abstract

Users interact a lot with their personal music collections, typically using standard text-based interfaces that offer constrained functionalities based on assigned metadata or tags. Alternative visual interfaces have been developed, both to display graphical views of music collections that attempt to reflect some chosen property or organization, or to display abstract visual representations of specific songs. Yet, there are many dimensions involved in the perception and handling of music and mapping musical information into computer tractable models is a challenging problem. There is a wide variety of possible approaches and the search for novel strategies to visually represent songs and/or collections persists, targeted either at the general public or at musically trained individuals. In this paper we describe a visual interface to browse music collections that relies on a graphical metaphor designed to convey the basic musical structure of a song. An iconic representation of individual songs is coupled with a spatial placement of groups of songs that reflects their structural similarity. The iconic representation is derived from features extracted from MIDI files, rather than from audio signals. The very nature of MIDI descriptions enables the identification of simple, yet meaningful, musical structures, allowing us to extract features that support both a music comparison function and the generation of the icon. A similarity-based spatial placement is obtained by projecting the feature vectors with the *Least Square Projection* multidimensional projection, with feature similarity evaluated with the *Dynamic Time Warping* distance function. We describe the process of generating such visual representations and illustrate potentially interesting usage scenarios. keywords:Visualization of Music CollectionsMultidimensional Projection High-Dimensional Data VisualizationSimilarity-based Visualizations

1 Introduction

Highly affordable personal computing devices with good processing and storage capacity, coupled with the proliferation of peer-to-peer networks and online music stores have facilitated the widespread accumulation of personal audio data. Browsing, sharing and talking about music collections is now commonplace to many users. With collections growing in size and diversity at a fast pace, they often struggle to recover relevant pieces, organize or just explore their collections. Most systems rely solely on text-based interfaces with constrained functionalities based on metadata, either assigned or retrieved. Users often feel frustrated as they want to compare, inspect or just talk about a song, not necessarilily being actually aware of title, performing artist or album.

This scenario has motivated several alternative visualizations of music, focusing on different user needs or expectations. Some solutions are targeted at helping users to organize their personal collections, others focus at conveying information about the nature of a particular musical piece to professionals or to the general public. These visualization systems typically require metadata as input, albeit some solutions rely on features extracted directly from audio signals or from other music representation formats. Surely the choice of features to describe the songs determines what can be conveyed by a visualization.

In order to be effective, feature-based music visualization tools must handle two main issues, namely, identifying similarities in musical compositions and conceiving an intuitive visual representation that facilitates the analysis of a single music or a collection. With a few exceptions, existing solutions handle these as two separate problems and devise independent solutions to each . Moreover, methods vary widely as to the metric employed to assess the similarity between pieces or between segments from a particular music. In fact, the concept of similarity is highly dependent on the application goals, and one can hardly expect that a single metric will be effective in multiple contexts [3]. Therefore, most visualizations focus on specific tasks such as highlighting songs from a common genre or grouping those with similar harmonic structure, tailoring the similarity metrics to their goal. The design of metrics that serve to more general purposes while being useful for visualization purposes is an issue that deserves further investigation.

This work introduces a novel visualization of a collection of songs that attempts to overcome some of the above issues. More specifically, we present a new mechanism to extract and structure features from musical data which enables measuring the similarity between music pieces as well as building meaningful visual representations. The proposed visual metaphor abstracts some aspects of the musical structure and favors an integrated manipulation of both single musical pieces and collections. It combines an iconic representation with a spatial placement that reflects the structural similarity of songs, an information that may be useful in different exploratory contexts. The spatial placement based on similarity is obtained by projecting the music features with the *Least Square Projection* (LSP) [23] multidimensional projection technique, with feature similarity evaluated with the DTW – *Dynamic Time Warping* – distance function. The design rationale behind our solution is: (i) obtain a simple abstraction of the musical structure of songs that reflects their global structural similarity; so that (ii) songs can be arranged in a visualization so as broadly to reflect their similar musical structures, (iii) without requiring user intervention to organize the visualization or describe similarity.

Unlike most current approaches for music visualization, we perform feature extraction on MIDI, a symbolic music format. MIDI files encode a description of the notes in the musical score and its very nature enables to identify meaningful musical structures. In contrast to audio file formats such as MP3, which store signals, the MIDI-encoded information allows for extracting and/or computing higher level abstract features. Our choice is further motivated by the fact that the MIDI format is adopted by recording companies and music professionals, and as such provides an interesting source of data for experimenting with new music visualization solutions that may serve distinct communities.

In summary, the main contributions of this work are:

- A new methodology to abstract structural features from MIDI files that allows for identifying similar songs while still serving for visualization purposes of a single music or a collection.
- An iconic representation of the basic structure of musical compositions. Such iconic representation enables selecting and analysing groups of songs that share a common structural similarity in terms of chord sequence repetitions.
- An interactive mechanism that combines the iconic representation with a multidimensional projection, providing an alternative interface for navigating music collections.

As shown in the results, the proposed approach turns out quite flexible and affords tasks such as visually identifying cover songs and temporal changes in the musical style of a particular artist or band.

2 Related Work

Several music visualization solutions are described in the literature, including visual interfaces proposed for handling collections and solutions targeted at visualizing single musical pieces.

Collections Visualizations of collections are typically devised to highlight some property of similarity, which may be computed from features automatically extracted from the audio signals, or based on user-assigned metadata, or yet from a combination of both. Approaches vary in their choices of representative features, similarity criterion and visual me- taphors. *Islands of Music* [22], an early example of graphical interface for digital music libraries, displays songs as islands with mountains and hills, placed so as to reflect their similarity. 1,200 features that reflect characteristics of the human auditory system are extracted from MP3 files, and later reduced to 80 dimensions with *Principal Component Analysis.* These are input into a *Self-Organizing-Map* (SOM) neural network to generate the geographical map metaphor.

Neumayer et al. [21] also use SOMs to create visual interfaces for PDAs and Tablets. Their focus, however, is on extracting audio features capable of characterizing perceived sounds to enable automatic classification of songs according to their rhythmic patterns. Results are reported from experiments on three distinct collections including pieces from several musical rhythms and genres. *MusicBox* [17] maps a music collection into a two-dimensional space representation, but spatialization is achieved applying *Principal Components Analysis* to a flexible combination of contextual and content-based features extracted from MP3. As in several other works, including our own, the goal is to display a similarity map that visually groups similar songs and separates dissimilar ones. The graph-based visual interface by Muelder et al. [20] also highlights song similarities. The Discrete Fourier Transform is applied to convert audio files into spectograms, which are then statistically reduced to signatures and compared with a similarity function. Songs are shown as graph vertices, and weighted edges indicate pairwise similarities. All previous approaches rely on properties extracted directly from the audio, and as such can only support the identification of similarity relations that can be captured from the audio signal. Tasks such as identification of songs with similar structure in terms of chord sequences but with distinct harmonic arrangement can hardly be accomplished by those methods.

Visualizations may also use metadata, either as a complement to other features, or as standalone features. Gulik et al. [12] describe a graph-based visualization for browsing music collections on small screen devices that highlights the similarity between artists, which is computed from features extracted directly from the audio, but with added metadata to provide contextual information, such as *mood*, *genre* and *year*. Some of these may be obtained from music services, whereas others, e.g. *tempo*, are obtained from the audio. The SOM-based visualizations of artists and songs by Risi et al. [26] may be built using either semantic descriptions learned from low level audio features or user-assigned tags in the *Last.fm* social music platform. *MusicalNodes* [7] uses graph visualizations of albums organized according to their tagged genre, while the interactive overview visualizations by Torrens et al. [29] organize songs considering multiple dimensions such as *genre*, *artist*, *year*, and others. Relying solely on user assigned metadata also has limitations, as errors, inconsistencies and missing information often affect the quality and precision of the resulting visualizations.

Single Music A few contributions address the problem of visualizing individual songs. In this case, information may be extracted from the audio signals [18], from MIDI descriptions [35, 4, 34], or yet from descriptive documents [38].

Akin to our contribution, Muller and Jiang [18] exploit the idea of visually conveying repetitive structures in music. They perform music structure analysis on audio files to segment the music and compute a fitness value that is assigned to each audio segment. Based on the induced segmentation, a distance measure is defined that allows comparing two arbitrary segments, and then map similar segments to similar colors and vice-versa. With such information and the fitness values authors derive scape plot visualizations that indicate the repetitive properties of segments and cross-segment relations. These may be combined into a single hierarchical representation called structure scape plot. The focus is on exploring the internal structure of a piece.

Wolkowicz et al. [35] generate a single image that represents a music, and like us use the MIDI description. A music is displayed as a rectangular color image made of squares, where each square represents the similarity between two corresponding notes played at times t_i and t_j. Similarity matrices, computed as described in earlier work by Foote [9], are used to create the visualization, which can reveal aspects of the musical structure and detail leading themes. Its generation requires defining a music representation and a comparison function.

Mapping note lengths and pitches directly to a comparison function is not adequate, as melody direction would not be preserved when comparing two similar melodies that lay on different pitches or are played in different scales; and neither would be rhythmic similarities that exist while the same melody is played slower or faster [35]. The solution adopted considers pitch and duration intervals. Authors introduce the notion of unigrams, or the smallest units of melody and rhythm. Each unigram represents the relative pitch and relative duration between two consecutive notes. This approach ensures that the same melody played in various tempos and in different pitches will result the same unigram sequence. The quality of the visualization is, of course, highly dependent on the quality of the MIDI encoding. The focus is on representing the internal structures within a song, and not in comparing different pieces.

Another interesting example is the "The Shape of Song" visualizations by Wattenberg [34]. The visual diagrams display songs as sequences of translucent arches, where each arch connects two repeated, identical passages of a composition, reflecting its structure. MIDI files describe multiple tracks typically associated with different instruments or voices. Each track is analyzed separately to produce the visualizations. Resulting diagrams reflect the unique timing and rhythm characteristics of each MIDI description.

In yet another relevant work, Wing Yin et al. [38] introduce a sophisticated visualization aimed at revealing the semantic structure of classical music pieces, so that even individuals with little background or formal training can gain insight into structural elements and compositional techniques and actually 'understand' the toughts behind the structural arrangements. Authors formulate semantic structure in terms of macrolevel layer interactions, microlevel theme variations and interactions and relationships between themes and layers. Musical structure data are retrieved manually from descriptive essays in a pre-processing stage conducted to recover the relevant structural information on layers and themes.

Also for visualizing music pieces, and also focusing on displaying the chord structure, the Isochords visualization [4] highlights the consonant intervals between notes and common chords in music, conveying information about interval quality, chord quality, and the chord progression. Again, the goal is to offer listeners a means to grasp the underlying structure of a song. Musical events are displayed on a two-dimensional triangular isometric coordinate grid to visually approximate the consonance and dissonance of tones.

The solutions discussed adopt different approaches to handle the problem of visually abstracting aspects of the musical structure. Although they can support comparisons between a few pieces, the goal is to facilitate the analysis or comprehension of single pieces, which is achieved at various levels of sophistication. On the other hand, solutions for visualizing collections concentrate on conveying the similarity relations rather than visually abstracting specific pieces. Combining both types of tasks is typically not a target goal, even though similarity assessment is a relevant issue in computational processing of music and has deserved considerable attention [3, 40].

Other musical data Other kinds of music related tasks have motivated specific visualization tools, for example, tracking media usage [1]. These visualizations depict listening histories as recorded, e.g., in *Last.fm*. Later, Baur et al. [2] introduce the concept of backdrop visualizations of listening records, again obtained from *Last.fm*. A backdrop visualization is one that is not necessarily the central focus of user attention, playing a background role in a certain scenario. Visualizations allow comparing, explaining and discussing the listening histories of two persons, and authors claim they can reveal information on tastes and habits and enrich conversation about music. Such solutions are not in the scope of this work, but they do exemplify the richness of music related data as sources for visualization tools.

Our approach In this work we chose to design a specific feature extractor that captures relevant chord sequences in order to work with a higher level abstraction of the musical structure, as detailed in Section 3. Chord recognition is extensively studied in the field of music information retrieval, but it is typically performed on the audio data, which is converted into a sequence of chroma-based features [14], whereas we recognize chord sequences in the MIDI encodings. Using MIDI as input enables extracting musical abstractions that are meaningful to musically trained individuals, such as the instruments present, melodic contour, chord frequencies and rhythmic density. *JSymbolic*, a software package designed as a feature extractor from MIDI files [19] and widely employed in music retrieval does not serve our purposes, as it does not provide the chord frequencies that for the basis of our approach. Our representation of a music affords creating an icon that summarizes properties not explicitly represented in visualizations built from features extracted from audio signals. Moreover, it allows creating visualizations capable of grouping songs based on their underlying structures and understanding their similarity relations by comparing their visual structures.

As discussed above, existing visualization techniques either focus in computing and highlighting similarities or in identifying and highlighting structures internal to a music. To our best knowledge, our solution is unique in its usage of a single representation both to visually picture a music and to identify similarities between songs. This combination supports certain analysis tasks that can not be conducted with existing visualization solutions.

3 Feature extraction: Chord Recognition

An overview of the steps in the process of obtaining an interactive similarity-based visualization from a music collection is shown in Figure 1.

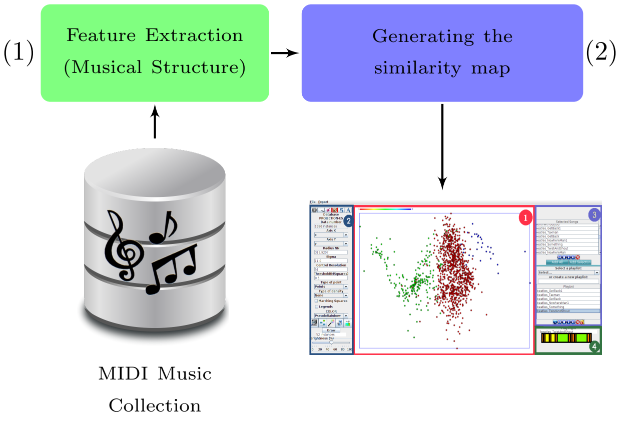


Figure 1: Overview of the steps in generating the music visualizations.

The proposed approach relies on the structure of musical pieces to define a measure of similarity as well as to provide a visual representation of both individual songs and collections. The MIDI files are processed using basic music theory [13] in order to identify music tonality and chords and abstract the global structure of a song. In this section we show how tonality and chords, which correspond to the building blocks of our approach, can be obtained from MIDI files.

Music tonality MIDI files provide information about the compass (small intervals in which a musical piece is split) and the set of notes played in each compass. Using Table 1 it is possible to obtain, from the set of notes, the tonality of a compass. For instance, if a compass contains the notes F# and C# but not G#, D#, A#, E# and B# then the tonality is D major or B minor. If none of the notes in Table 1 appears in the compass then the tonality is C major or A minor. To decide between major and minor tones one should analyze how many sharp (#) and flat (b) notes the compass has. The tonality is major if more sharp notes are present, otherwise, if more flat notes are present then the tonality is minor.

Chords identification Once the tonality has been defined one can identify the chord played in a compass under analysis. For the sake of illustration, suppose the tonality is C. There are a set of possible chords in the harmonic field of C, and we make use of the so-called triads to identify which of those chords correspond to the notes in the compass. Triads are the three main notes that define an chord, for instance, the chord E minor is in the harmonic field of C and its triad comprises the notes E,G, and B. Lookup tables can be built for the chords in harmonic field for each tonality. Therefore, a chord can be identified by taking the tonality of the compass and its three most frequent notes, querying the lookup table for the chord made up by those three notes.

It is important to emphasize that, although we consider all MIDI channels to extract information, the music theory we employ to identify chords is really basic. In fact, there are a multitude of variations as to chord constructions, change of scale and tonality during the execution of musical piece. Some of the contributions reviewed in the literature, e.g., [38, 4] produce more complex mappings focused on other types of user tasks. Still, incorporating all the complexity inherent to music constructions within our general scenario is unfeasible and, as shown in Section 5, the basic theory suffices for reaching informative and coherent results for our purposes. Several sources are available to the reader interested in a deeper understanding of music theory [36, 37, 13, 24].

Armor	**Major tonality**	**Minor tonality**
Bb, Eb, Ab, Db, Gb, Cb, Fb	Cb major	Ab minor
Bb, Eb, Ab, Db, Gb, Cb	Gb major	Eb minor
Bb, Eb, Ab, Db, Gb	Db major	Bb minor
Bb, Eb, Ab, Db	Ab major	fa minor
Bb, Eb, Ab	Eb major	C minor
Bb, Eb	Bb major	G minor
Bb	F major	D minor
	C major	A minor
F#	G major	E minor
F#, C#	D major	B minor
F#, C#, G#	A major	F# minor
F#, C#, G#, D#	E major	C# minor
F#, C#, G#, D#, A#	B major	G# minor
F#, C#, G#, D#, A#, E#	F# major	D# minor
F#, C#, G#, D#, A#, E#, B#	C# major	A# minor

Table 1: Music tonalities.

4 Similarity and Visual representations

The proposed visualization scheme conveys information on the nature of a single music and also on how individual songs in a collection relate with others. It combines an icon representation with information given by a music similarity map. In the following we describe how the music icon and the music similarity map are generated.

4.1 Iconic representation

Each music is depicted as an icon that displays its repeated structural patterns by showing colored block segments that correspond to different patterns. Both the length and color of each block segment map the size of its corresponding pattern.

The icon representing each music is a graphical realization of the structural arrangement of chords along the music. In order to build the icon, the *Horspool* string matching algorithm [16] is applied to identify patterns of chords in a given musical sequence. Subsets, initially of size $\frac{N}{2}$, in which N is the total number of chords, are extracted from the chord sequence. These subsets of chords are the patterns to be searched for by *Horspool*, which scans the sequence for the pattern until it finds one or more occurrences. This process is performed iteratively, at each step decrementing by 1 the size of the patterns to be identified and then searched. Finally, the patterns found in the chord sequence are represented as segment blocks in the icon, associating a different color to each pattern. This process is summarized in Algorithm 1.

Figure 2 shows a simple example of applying Algorithm 1 to a hypothetical chord sequence of size $N = 8$, depicted in Figure 2(a). Initially, the pattern size is set to $n = 4$, the patterns are identified

Algorithm 1 Icon Generation algorithm.

```
Require: Chord sequence $S = \{s_1, s_2, \cdots, s_N\}$ extracted from a MIDI song
Ensure: An icon with colors mapping the chord patterns
for $n = N/2$ to 1 do
  repeat
    Find in $S$ the most frequent pattern $p_n$ of size $n$, considering only chords not already used on longer patterns
    if $f(p_n) > 1$ then {$f(p_n)$ stands for the frequency of $p_n$}
      Assign a color to the chords in $S$ that compose the pattern $p_n$ (different from any previously used color)
    end if
  until $f(p_i) \leq 1$
end for
If no colored chords exist, remove them from $S$ and create the final icon
```

and their corresponding frequencies are calculated, as shown in Figure 2(b). Since all patterns occur exactly once, none is selected. The pattern size is then set to $n = 3$. Figure 2(c) lists the corresponding patterns and their frequencies. Notice that pattern ABC (yellow segment) occurs twice in S. This pattern is selected and after checking that it is contained in an already labeled longer pattern, a color (blue) is assigned to the chords (Figure 2(d)). Figure 2(e) shows the patterns of size $n = 2$ and their corresponding frequencies. Patterns with multiple occurrences (AB and BC) that are contained in patterns already labeled are discarded. Figure 2(f) shows the patterns of size $n = 1$. Only the standard D is selected and labeled with a different color (red), as it has multiple occurrences and is not contained in any pattern already labeled (Figure 2(g)). The patterns that occur only once are not significant and may be removed, as the standard E in this case. Figure 2(h) shows the resulting icon for this example.

We combine the iconic representation of songs and the similarity layout of the collection to derive visualizations that support both observing the structure of individual songs and comparing the structures of multiple ones, as discussed next.

4.2 Music similarity maps

The icon symbolizing a music piece is indeed a graphical realization of the feature vector that represents the music. More precisely, each feature vector corresponds to an array where the entries store the music segments detected during the icon construction. For instance, the feature vector corresponding to the music symbolized by the icon in Figure 2.h is {3,1,3,1}.

Those feature vectors may be interpreted as a regularly sampled signal, or a time series, with amplitudes given by the vector entries, as illustrated in Figure 3. Figure 3.a shows the icon computed for a particular music, and Figure 3.b depicts its corresponding feature vector representation as a signal.

Given a set of songs described by their structural features, we generate a similarity-based 2D spatial layout projecting the feature vectors with the *Least Square Projection* (LSP) [23]. Similarity is assessed with the *Dynamic Time Warping* (DTW) distance function. DTW is a suitable choice because it can compare two feature vectors (signals) while accounting for temporal displacements, thus finding an optimal match between the given signal sequences.

Also worth noticing is that the feature vectors and their corresponding "signals" have different sizes, as each music is represented by an arbitrary number of relevant chord sequences. Although DTW can handle sequences of different sizes, we obtained better results after applying nearest-neighbor interpolation to the smaller sequence, so that both sequences under comparison have the same size. We also use interpolation to fit sequences to a particular size when displaying a detailed list of music icons, in this case for aesthetical purposes, as it will be illustrated in Section 5.

Figure 4 illustrates (in area labeled 1) a layout, or "similarity map", obtained with the above process. The collection is depicted as a point cloud, where each song is represented by a circle colored according to its (known) genre (notice that genre information was not considered in computing the layout). Such layouts provide potentially useful information on the relationships between songs, but point clouds are hard to interpret. For a start, they disclose little information about the nature individual music pieces and why groups are being formed. Moreover, there is considerable clutter.

We combine the similarity layout and the structure icon to deliver a compromise between the point cloud similarity maps, which lack semantic information and suffer from clutter, and simple lists of icons which lack information on similarity relations. This is achieved by displaying the similarity map as a point cloud, and displaying a sequential list of the music icons to detail any user selection on the map.

A	B	C	D	A	B	C	E	D

(a) A hypothetical chord sequence S

ABCD	BCDA	CDAB	DABC	ABCE	BCED
1	1	1	1	1	1

(b) Patterns of size 4 and their frequencies.

ABC	BCD	CDA	DAB	BCE	CED
2	1	1	1	1	1

(c) Patterns of size 3 and their frequencies.

A	B	C	D	A	B	C	E	D

(d) Chord sequences S.

AB	BC	CD	DA	CE	ED
2	2	1	1	1	1

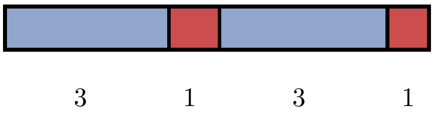
(e) Patterns of size 2 and their frequencies.

A	B	C	D	E
2	2	2	2	1

(f) Patterns of size 1 and their frequencies.

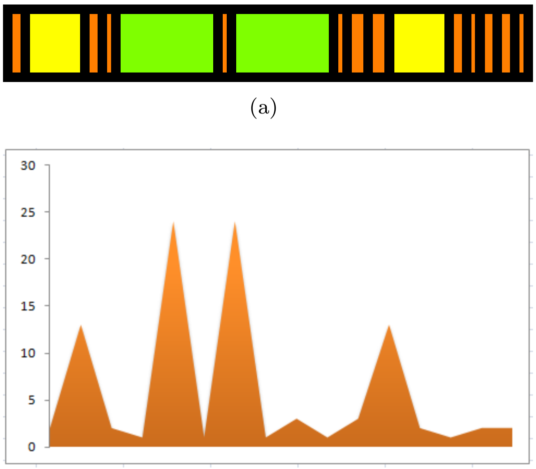
A	B	C	D	A	B	C	E	D

(g) Chord sequences S.



(h) Structure of chord patterns extracted from a song

Figure 2: Steps in the process of creating a music icon.

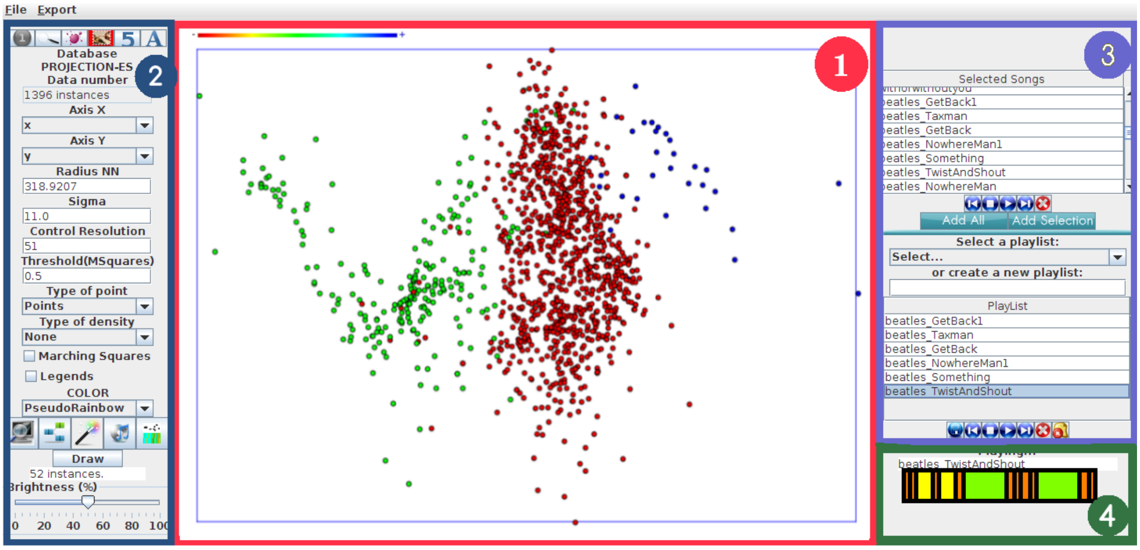


(a)

(b)

Figure 3: Icon of the musical structure of the song “Wind of change”, by Scorpions, and its corresponding feature vector interpreted as signal.

4.3 Visualization System



Database
PROJECTION-ES
Data number
1396 instances
Axis X
x
Axis Y
y
Radius NN
318.9207
Sigma
11.0
Control Resolution
51
Threshold(MSquares)
0.5
Type of point
Points
Type of density
None
☐ Marching Squares
☐ Legends
COLOR
PseudoRainbow
52 instances.
Brightness (%)
0 20 40 60 80 100

Selected Songs
withorwithoutyou
beatles_GetBack1
beatles_Taxman
beatles_GetBack
beatles_NowhereMan1
beatles_Something
beatles_TwistAndShout
beatles_NowhereMan

Select a playlist:
Select...
or create a new playlist:

PlayList
beatles_GetBack1
beatles_Taxman
beatles_GetBack
beatles_NowhereMan1
beatles_Something
beatles_TwistAndShout

beatles_TwistAndShout

Figure 4: Screenshot illustrating the interface and functionalities of the music visualization system.

The representations described have been incorporated into a visualization interface for music collections shown in Figure 4. The visualization system conveys information about the nature of individual songs and also about how songs are related to others in terms of their underlying structure, allowing the exploration of a personal collection. It combines the iconic representation with the similarity map to intuitively visualize relationships and browse the collection at different levels of detail.

The area labeled **1** in the interface shows the interactive similarity map of a collection. Users can interact zooming in/out with the mouse scroll, select individual songs or brush to select groups, using the tools provided in area **2**, which allow, for example, to activate a magnifying glass, brush a selection, or display song titles over the map. It also includes controls for input parameters required by several details-on-demand interaction functionalities.

It is possible, for example, to select a region in the map and handle the corresponding songs as a new data set on a different window, generate the grid-based spatial view, or generate and display a list of the structure icons for a selected subset of songs. Brushing a selection in the map, as indicated by the rectangle, results in details displayed in area **3**: it is possible to play any song by double-clicking on its title or by clicking the play button, move forward or backward in the list by clicking on the corresponding buttons, or remove songs from the list. Furthermore, listed songs (or a subset) may be added to a playlist, which offers similar reproduction functionalities. Area **4** displays the title of the song currently playing

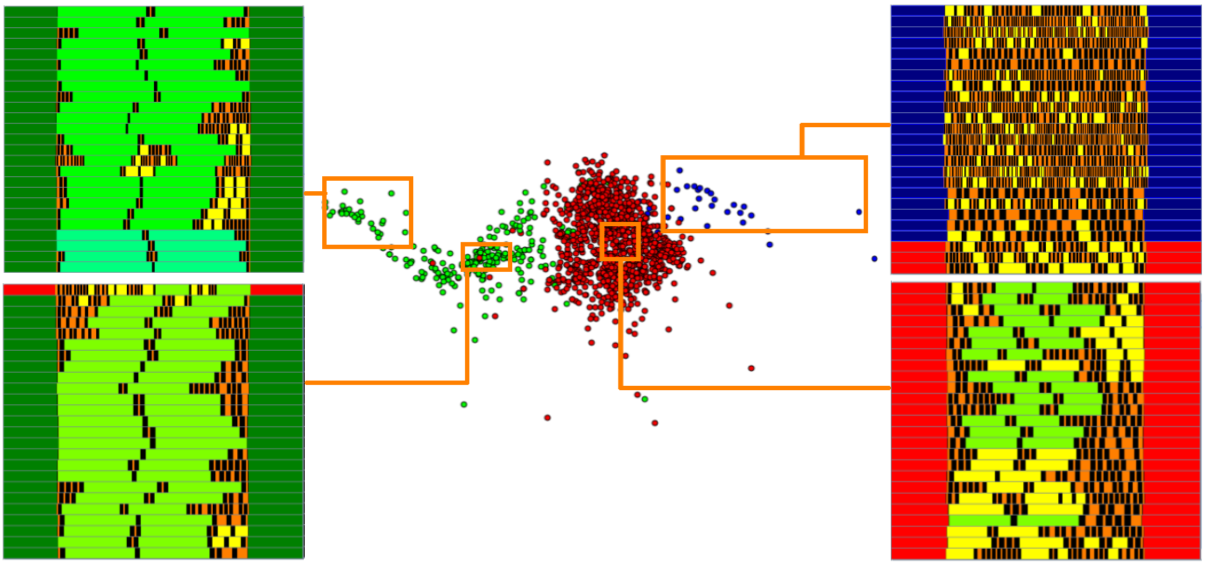


Figure 5: Similarity map of 1, 308 songs, including samples from classical music (blue), pop-rock (red) and latin country (green), and musical icons relative to the selections.

and its corresponding structure icon.

5 Results and Comparisons

Results presented in this section were obtained in an Intel Core i3 CPU M 350 2.27GHz ×4 and 8GB of RAM. The MIDI music files used to illustrate our visualizations have been gathered from multiple Internet sources, resulting in a collection of nearly 1, 400 songs of varied musical genres. Table 2 summarizes the collection and details the distribution of the songs by genre.

Genres	
pop-rock	1, 031
classical music	31
latin country	246
jazz	95
Total	1, 403

Table 2: Contents of the MIDI music collection.

In the following we illustrate some exploratory tasks that can be supported by the proposed visualizations and interaction functionalities.

5.1 Structure versus Genre

The representation based on the musical structure allow us to create visualizations that highlight the underlying similarities and differences of multiple musical genres. Figure 5 shows the LSP similarity map of a collection that includes classical music pieces (31 entries, represented by blue circles), pop-rock (1, 031 entries, red circles) and latin country (246 entries, green circles), comprising a total of 1, 308 pieces. The LSP layout clearly separates the songs from different genres, showing that the proposed feature representation can discriminate those classes of songs. The iconic representation of the highlighted groups depict the underlying structures of each selected music. Notice the similar structure of songs from the same genre, showing the effectiveness of our iconic representation towards enabling a visual comparison of the pieces in a collection.

A similar analysis can be made from Figure 6, which added some jazz pieces to the previous collection (95 entries, shown as the gray circles). The visualization suggests that the underlying structure of classical music and jazz are similar, as they overlap significantly in the layout. This is confirmed in Figure 7, which depict the structure icons for samples from each genre. Interestingly, we discussed those results with a maestro who confirmed that a distinguishing characteristic of both classical and jazz compositions, as compared with pop-rock music, is their lack of a homogeneous structure. Pop-rock songs, on the contrary,

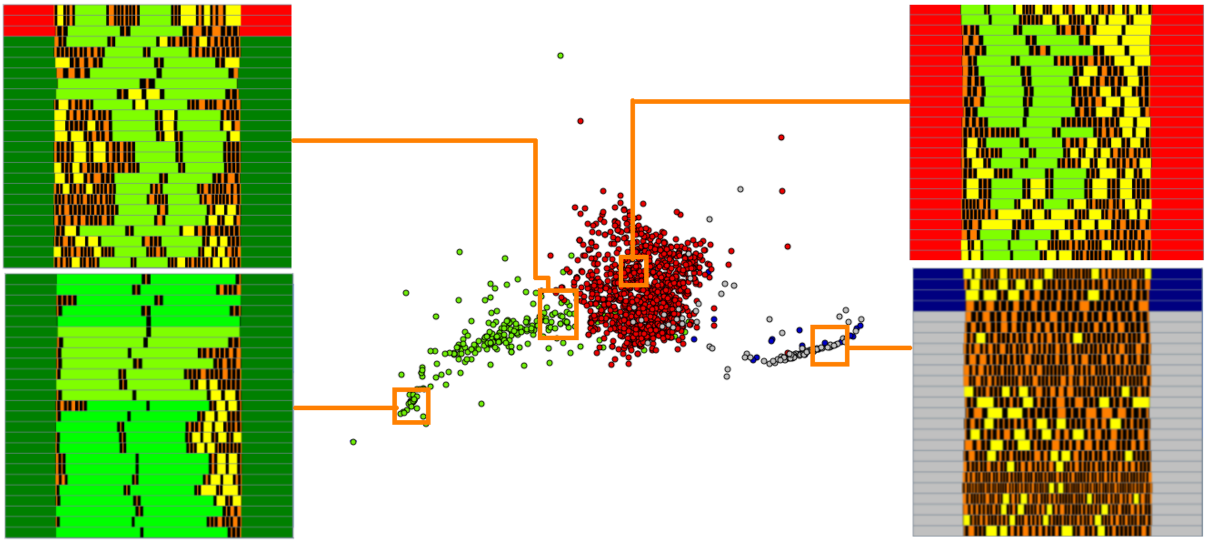
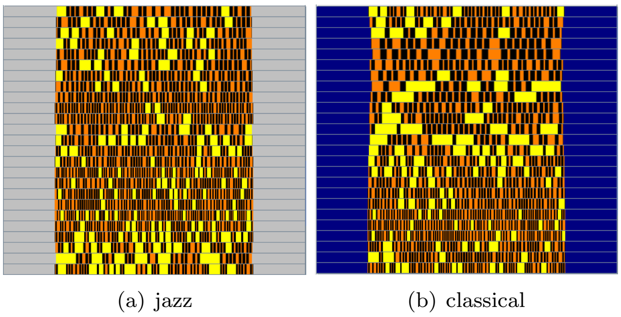


Figure 6: Similarity map of 1, 403 songs, including samples of classical music (blue), jazz (gray), pop-rock (red) and latin country (green), and musical icons relative to the selections.



(a) jazz

(b) classical

Figure 7: Structure icons for samples of jazz and classical songs in the collection.

typically have a small number of well-defined blocks of chords that repeat two or three times throughout the music.

The icons in the detailed views are stacked according to their computed similarities in the layout. Moreover, all selected icons are displayed with the same size, although songs are represented by feature vectors of different sizes. Their signal representation, as discusses in Section 4 are interpolated so that they all have the same size, thus producing a more pleasing visualization, but one might also consider retaining the original sizes in the visualization if such information is deemed useful.

5.2 Identifying Music Versions

Figure 8 shows the icons created relative to several versions of songs by ABBA and Beatles contained in our collection. The MIDI descriptions have been collected from multiple internet sources. For popular songs, these are often created by volunteers and for different purposes, not necessarily complying with specified quality standards. The quality of the outcome is likely affected by varying degrees of user ability, motivation and goals, unlike the high-quality MIDI encodings provided, for example, by recording companies.

It is interesting to compare versions of a music coded in different MIDI files. The icons clearly show similarities and variations, which may be due to the varying quality of the encodings, to using various musical instruments, or yet to different performances of a particular song. The variations are clearly perceptible when the MIDIs are executed.

One can inspect some specific cases. For example, we have two distinct MIDI files encoding the song “Dancing Queen”, by ABBA. Version 1 includes 89 compasses, whereas version 2 has 99 compasses. This introduces variations in the icons because in extracting the feature vectors we generate one chord for each compass. In version 2 the music is played slightly faster, and conversion of the MIDI to the corresponding musical score revealed it is slightly more complex than that of the version 1. Inspecting the corresponding icons one observes that differences exist, which are also discernible when listening to the songs. A similar

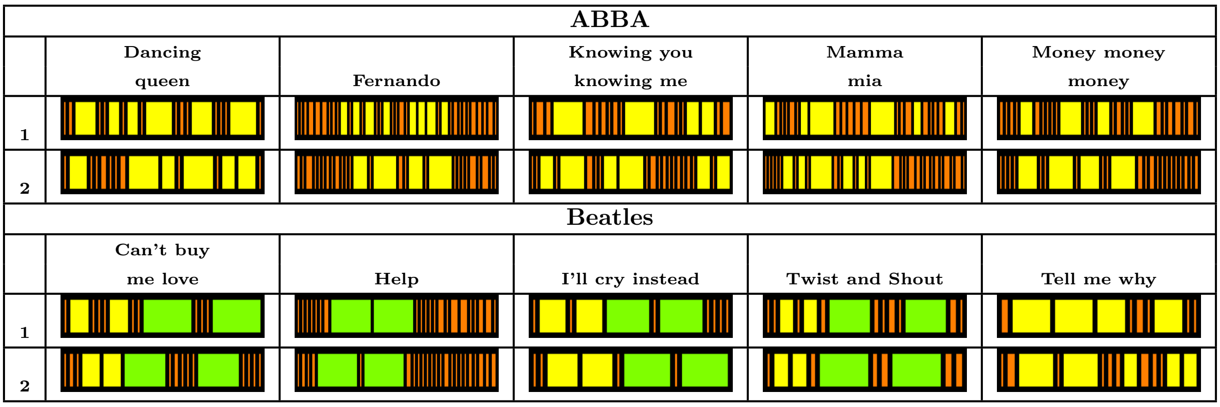


Figure 8: Musical icons reveal differences in multiple performances of a song (distinct MIDI sources).

analysis applies to the two encodings of the song "Fernando". When listening to both versions of this particular song one observes that one is clearly of a better quality. Variations in the encodings of other songs by ABBA, such as "Knowing you, knowing me", "Mamma Mia" and "Money, money, money" also exist, but they are more subtle. For example, MIDI version 1 of "Knowing you, knowing me" is slightly more accelerated and again we found that version 2 has a more complex musical score. Again, variations are noticeable in the icon representations.

Let us consider the encodings of a few songs by The Beatles. Version 2 of "Cant buy me love" again has a slightly faster pace and the endings of the two versions are quite different. Version 1 ends abruptly, whereas version 2 finishes gradually with a smooth melody. A similar case happens with "I'will cry instead". The versions of "Tell me why" have quite similar, but not identical icons. When listening to the songs one notices that the encoding of version 1 uses more instruments and is of higher quality. Version 2 of "Help" has a longer introduction and a faster pace as compared to version 1. In "Twist and Shout" the icons are very similar, and the perceivable differences in their performances lie in the introduction and in the middle, as reflected in the corresponding icons.

5.3 Temporal evolution

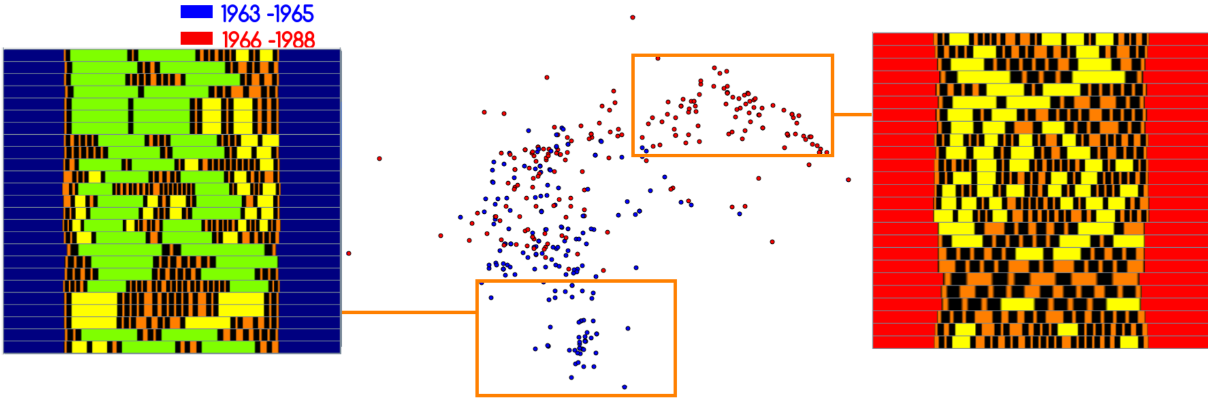


Figure 9: Overview of songs by Beatles in different stages of their production, and detailed views of the structure of songs from each period considered.

Figure 9 shows a projected similarity map of 370 songs by Beatles that have been split into two time windows, one referring to their very early production, from 1963 to 1965 (blue circles), and other relative to songs from 1966 and later (red circles). The figure also details the musical structures relative to both periods, displaying the icons for two selections of songs from each temporal period considered. Their structural patterns are obviously very distinct, confirming the temporal segmentation observed in the spatial distribution of the songs in the similarity map, with a concentration of the older songs in the left area.

This example illustrates how the visual representations can support exploratory investigations on how the production of an artist or band changed along time. We have no knowledge of any other visual tool capable of performing a similar task.

Matching objects	84.9%
Non-matching objects	15.1%
Accuracy	86.83%
Precision	84.85%
Recall	84.92%

Table 3: SVM classification results for data set in Figure 12.

		Classification			
		1.0	**2.0**	**3.0**	**FNR**
Ground Truth	**1.0**	21	0	10	32.26%
	2.0	3	143	97	41.15%
	3.0	21	68	957	8.51%
FPR		1.83%	5.94%	28.08%	

Table 4: Confusion matrix relative to SVM classification.

5.4 Comparing with other choices of feature vectors

In order to attest the effectiveness of the proposed features in discriminating songs based on their structure we derive and compare similarity maps considering other types of features typically obtained from MIDI or from MP3 files. LSP projections were generated for the same subset of $1,308$ songs that includes classical pieces (blue) plus pop-rock (red) and latin country (green). Figure 10 shows the resulting layouts.

The layout in Figure 10.a was obtained considering as feature vectors the histograms of the notes given in the MIDI files [30]. As with the proposed structural feature vectors, DTW was employed in this case to measure similarities. Figure 10.b shows the projection resulting from using statistical properties of the notes as features, namely the mean, standard deviation, uniformity and entropy. The Euclidean distance was employed to compute similarities.

Those statistical measures have been shown useful to quantify some global properties of a music, such as *tempo* [15]. A projected layout of the songs represented by features extracted with *JSymbolic* [19] is shown in Figure 10.c. We extracted 111 features of several kinds, based on instrumentation, musical texture, rhythm, statistics, melody and chords, and compared the resulting vectors using the Euclidean distance. Figure 10.d shows a projected layout obtained using standard features extracted from MP3 audio files, which have been obtained by converting the respective MIDI files.

Finally, Figure 10.e shows the projection of the songs represented by the proposed structural features, which is clearly the most effective alternative to group songs belonging to the same genre, which in this case also correspond to songs with similar structures.

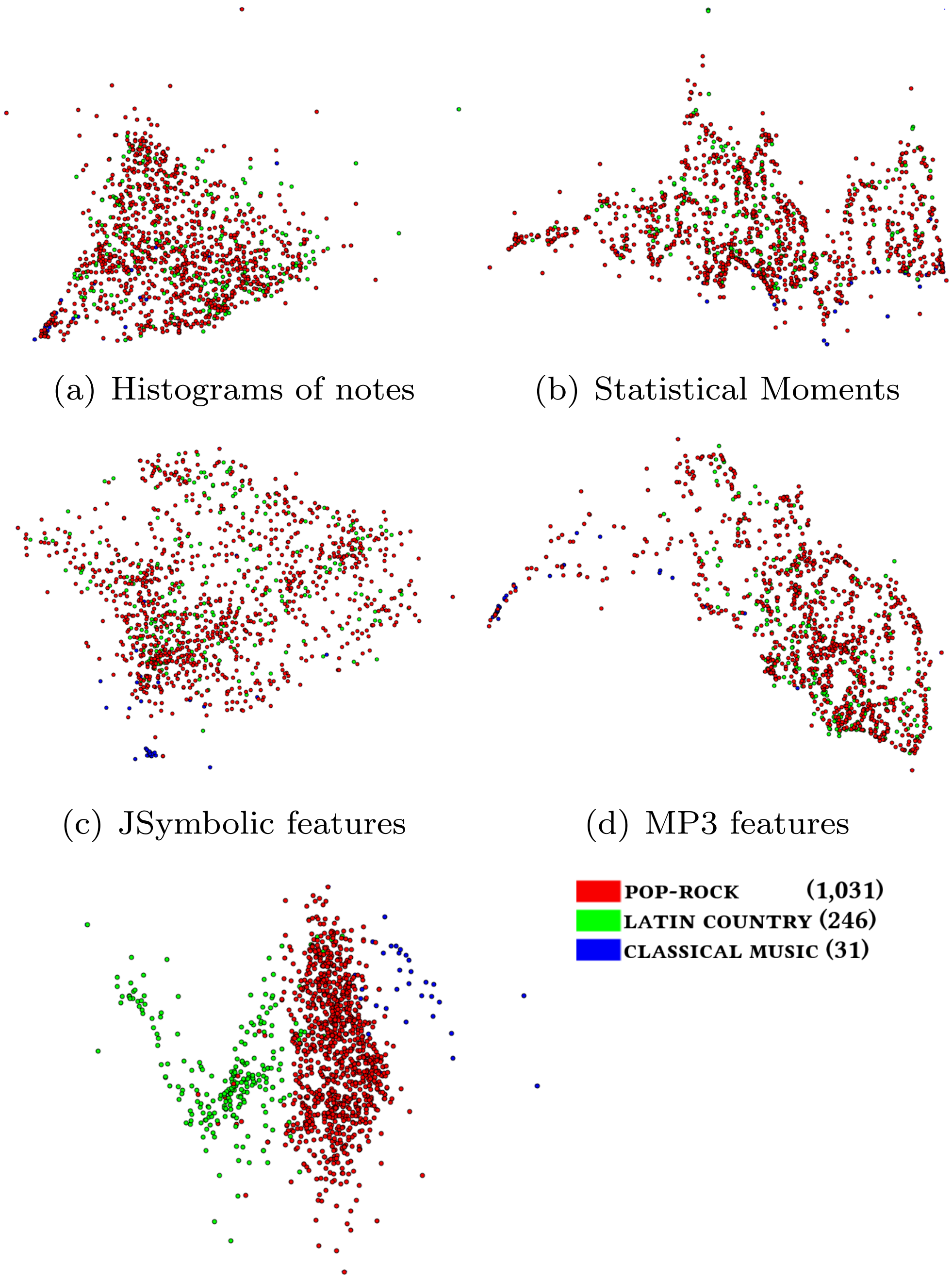
The genre discrimination capability of the feature vectors is confirmed in other visual representations of the collections, such as the Neighbor-Joining tree visualization [39]. The NJ tree shown in Figure 11 depicts the same collection displayed in Figure 6, with songs from 4 genres, and was obtained using DTW as the dissimilarity measure.

We used the *Visual Classification System* (VCS) by Paiva et al. [42] to obtain a classification of the same collection depicted in Figure10, represented with the proposed feature vectors and measuring dissimilarity with the DTW distance. As supported by VCS, we used an NJ-tree representation to visually select a training sample. Best results were obtained with a training set of 370 samples (28.3% of the data set) and the *Support Vector Machine* (SVM) classifier (VCS uses the LibSVM implementation [41]). The SVM parameter settings are as follows: linear kernel with coefficient 0, cost 1, kernel degree 3, tolerance for termination criterion 0.0010, gamma 0, nu value 0.5, and default data normalization. Figure 12 shows in (a) the NJ-tree, and in (b) a mapping of the SVM classification results, as compared to the known classes (ground-truth), where green indicates a song correctly classified and red indicates a classification error. Table 3 summarizes relevant numerical measures and Table 4 shows the confusion matrix.

6 Discussion and Limitations

The comparisons presented in the previous section clearly illustrate the potential of the proposed representation to distinguish songs with different types of structures and highlight similarities and dissimilarities.

Using the same underlying representation to derive a visualization of the songs and to assess similarity between them enables creating a flexible visualization that supports several exploratory tasks and



(a) Histograms of notes

(b) Statistical Moments

(c) JSymbolic features

(d) MP3 features

(e) Structural features

Figure 10: LSP layouts of songs from three distinct musical genres, considering distinct sets of feature vectors extracted either from MIDI or MP3.

navigation on a collection of songs. Example tasks include identifying different versions of the same music, identifying recordings with distinct characteristics and qualities, observing variations across multiple musical genres and temporal structural variations in the production of a band or artist.

We confirmed that the musical structure, expressed in terms of chord repetitions identified along the music, provides an interesting feature to characterize distinct musical genres, and has the added advantage of producing a summary visualization that is simple and easily interpretable.

Scalability is one of the main limitations of the current solution. Layouts resulting from projecting large data sets with multidimensional projection techniques tend to be cluttered, impairing the navigation throughout the data. This issue can be tackled by multiscale approaches and global views of a collection with details-on-demand functionalities, which we shall investigate as future work.

It is also true that the structure as extracted is highly simplified and does not map higher level semantic structures. This is because the feature extraction process proposed relies only on very basic

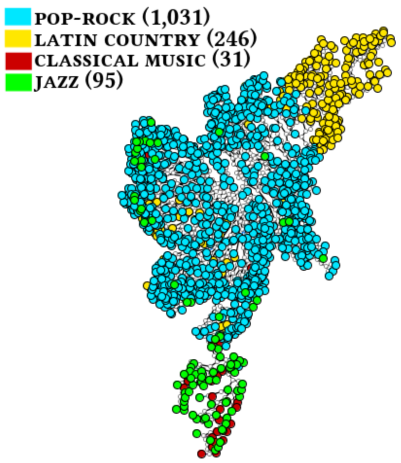
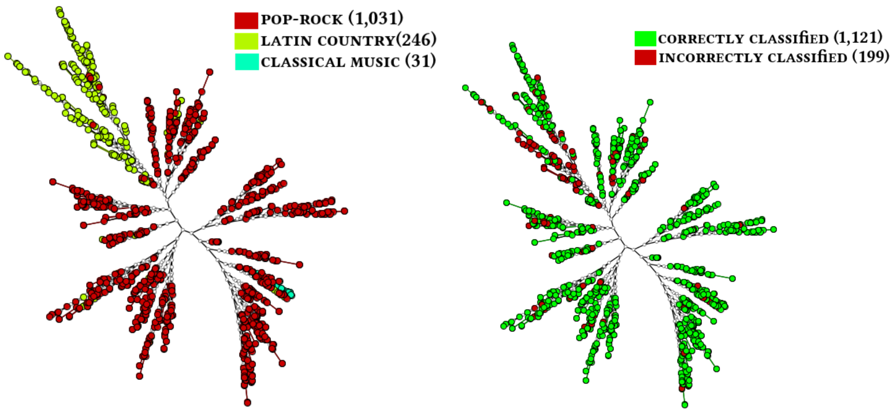


Figure 11: Neighbor-Joing Tree [39] obtained for the collection using the proposed structural feature vector representation, taking the DTW distance as dissimilarity measure. The tree shows an alternative view of the collection. Each circle depicts a song colored by genre. Songs within a branch are more similar than songs in different branches.



(a) NJ-tree

(b) SVM matching to ground truth

Figure 12: NJ trees mapping song genre in (a), and the result of SVM classification in (b), where red indicates mistaken classifications.

concepts of music theory and discards other kinds of potentially relevant information included in the MIDI sources, such as data about rythm and instruments. Still, it does provide an abstraction suitable for navigation purposes.

7 Conclusion

In this work we introduced a music visualization based on a representation of the internal structures within a song. This representation is based on extracting meaningful chord sequences from a MIDI description, which allows creating a feature vector that serves both to encode a visual representation and to compute similarities between songs. We presented several illustrative examples of how the proposed visualizations can assist users in some music exploration tasks that are not supported by existing solutions.

Acknowledgments

The authors acknowledge the financial support by FAPESP, the State of São Paulo Research Funding Agency, Brazil, CNPQ and CAPES, Brazil.

References

[1] Baur, D., Visualizing Media and Music Histories, CHI Workshop- Know Thyself: Monitoring and Reflecting on Facets of One's Life (2010).

[2] Baur, Dominikus and Butz, Andreas and Carpendale, Sheelagh, arcs.fm - A Backdrop Visualization for Music Talk, Eurographics Conference on Visualization, 109–113 (2012).

[3] Berenzweig, Adam and Logan, Beth and Ellis, Daniel P. W. and Whitman, Brian P. W., A Large-Scale Evaluation of Acoustic and Subjective Music-Similarity Measures, Computer Music Journal, 28, 2, 63–76 (2004).

[4] Bergstrom, Tony and Karahalios, Karrie and Hart, John C., Isochords: visualizing structure in music, Proceedings of Graphics Interface 2007, 297–304 (2007).

[5] Cano, P. and Kaltenbrunner, M. and Gouyon, F. and Batlle, E., On the use of fastmap for audio retrieval and browsing, 3rd International Conference on Music Information Retrieval (2002).

[6] Dalhuijsen, L., A graphical representation of digital music libraries using relational and absolute data to rediscover your collection, Leiden University (2011).

[7] Dalhuijsen, L. and van Velthoven, L., MusicalNodes: the visual music library, International Conference on Electronic Visualisation and the Arts, 232–236 (2010).

[8] Faloutsos, C. and Lin, K.I., FastMap: A fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets, ACM SIGMOD International Conference on Management of Data, 24, 163–174 (1995).

[9] Foote, Jonathan, Visualizing music and audio using self-similarity, 7th ACM International Conference on Multimedia, 77–80 (1999).

[10] Frese, Udo, Treemap: An O(log n) algorithm for simultaneous localization and mapping, 4th International Conference on Spatial Cognition: Reasoning, Action, Interaction, 455–477 (2005).

[11] van Gulik, R. and Vignoli, F., Visual playlist generation on the artist map, International Conference on Music Information Retrieval, 520–523 (2005).

[12] van Gulik, Rob and Vignoli, Fabio and van de Wetering, Huub, Mapping music in the palm of your hand, explore and discover your collection, 5th International Conference on Music Information Retrieval, 409–414 (2004).

[13] Harnum, Jonathan, Basic music theory: how to read, write and understand written music, 2nd edition, Sol Ut Press (2004).

[14] Nanzhu Jiang and Peter Grosche and Verena Konz and Meinard Müller, Analyzing Chroma Feature Types for Automated Chord Recognition, 42nd Audio Engineering Society (AES) International Conference on Semantic Audio (2011).

[15] Olivier Lartillot and Petri Toiviainen and Tuomas Eerola, A Matlab Toolbox for Music Information Retrieval, 31st Conference of the German Classification Society (GfKl), 261–268 (2007).

[16] Levitin, Anany, Introduction to the design & analysis of algorithms, Addison-Wesley Reading, MA (2003).

[17] Lillie, A.S., MusicBox: Navigating the space of your music, Massachusetts Institute of Technology (2008).

[18] Meinard Müller and Nanzhu Jiang, A Scape Plot Representation for Visualizing Repetitive Structures of Music Recordings, 13th International Conference for Music Information Retrieval, 97–102 (2012).

[19] McKay, Cory, Automatic music classification with jMIR, McGill University (2010).

[20] Muelder, C. and Provan, T. and Ma, K.L., Content Based Graph Visualization of Audio Data for Music Library Navigation, IEEE International Symposium on Multimedia, 129–136 (2010).

[21] Neumayer, R. and Lidy, T. and Rauber, A., Content-based organization of digital audio collections, 5th Open Workshop of Music Network, 4–5 (2005).

[22] Pampalk, E., Islands of music: Analysis, organization, and visualization of music archives, Vienna University of Technology, Vienna, Austria (2001).

[23] Paulovich, F.V. and Nonato, L.G. and Minghim, R. and Levkowitz, H., Least square projection: A fast high-precision multidimensional projection technique and its application to document mapping, IEEE Transactions on Visualization and Computer Graphics, 14, 3, 564–575 (2008).

[24] Pilhofer, Michael and Day, Holly, Music Theory For Dummies, Wiley Publishing, Inc. (2007).

[25] Rauber, A. and Pampalk, E. and Merkl, D., Using psycho-acoustic models and self-organizing maps to create a hierarchical structuring of music by sound similarity, International Symposium on Music Information Retrieval, 71–80 (2002).

[26] Risi, S. and Mörchen, F. and Ultsch, A. and Lewark, P., Visual mining in music collections with Emergent SOM, Workshop on Self-Organizing Maps, 3–6 (2007).

[27] Schedl, M. and Knees, P. and Widmer, G., Interactive poster: Using CoMIRVA for visualizing similarities between music artists, 16th IEEE Visualization Conference, 89 (2005).

[28] Stavness, I. and Gluck, J. and Vilhan, L. and Fels, S., The musictable: A map-based ubiquitous system for social interaction with a digital music collection, International Conference on Entertainment Computing, 291–302 (2005).

[29] Torrens, M. and Hertzog, P. and Arcos, J.L., Visualizing and exploring personal music libraries, 5th International Conference on Music Information Retrieval, 421–424 (2004).

[30] George Tzanetakis and Andrey Ermolinskyi and Perry Cook, Pitch histograms in audio and symbolic music information retrieval, 3rd International Conference on Music Information Retrieval, 31–38 (2002).

[31] Urbain, J. and Dubuisson, T. and Dupont, S. and Frisson, C. and Sebbe, R. and d'Alessandro, N., Audiocycle: A similarity-based visualization of musical libraries, IEEE International Conference on Multimedia and Expo, 1847–1848 (2009).

[32] Van Den Bergen, Gino, Efficient collision detection of complex deformable models using AABB trees, Journal of Graphics Tools, 2, 4, 1–13 (1997).

[33] Van Liere, R. and De Leeuw, W., Graphsplatting: Visualizing graphs as continuous fields, IEEE Transactions on Visualization and Computer Graphics, 9, 2, 206–212 (2003).

[34] Wattenberg, Martin, Arc Diagrams: Visualizing Structure in Strings, IEEE Symposium on Information Visualization, 110–116 (2002).

[35] Wolkowicz, V.K.J. and Brooks, S. and Keselj, V., Midivis: Visualizing music pieces structure via similarity matrices, International Computer Music Conference, 53–6 (2009).

[36] Wyatt, Keith and Schroeder, Carl, Harmony and Theory: a comprehensive source for all musicians, Hal Leonard Corporation (1998).

[37] Zeitlin, Poldi and Goldberger, David, Understanding music theory, Omnibus Press (2001).

[38] Chan, Wing-Yi and Qu, Huamin and Mak, Wai-Ho, Visualizing the Semantic Structure in Classical Music Works, IEEE Transactions on Visualization and Computer Graphics, 1, 16, 161–173 (2010).

[39] Cuadros, A.M. and Paulovich, F.V. and Minghim, R. and Telles, G.P., Point Placement by Phylogenetic Trees and its Application to Visual Analysis of Document Collections, IEEE Symposium on Visual Analytics Science and Technology, 99–106 (2007).

[40] Aucouturier, Jean-Julien and Pachet, Francois, Music Similarity Measures: What's the Use?, Proceedings of the 3rd International Symposium on Music Information Retrieval, 157–163 (2002).

[41] Chang, Chih-Chung and Lin, Chih-Jen, LIBSVM: A Library for Support Vector Machines, ACM Transactions on Intelligent Systems and Technology, 2, 3, 27:1–27:27 (2011).

[42] Jose Gustavo Paiva and Laura Florian and Helio Pedrini and Guilherme Telles and Rosane Minghim, Improved Similarity Trees and their Application to Visual Data Classification, IEEE Transactions on Visualization and Computer Graphics, 17, 12, 2459–2468 (2011).