This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2016.2614226, IEEE Transactions on Multimedia

MoshViz: A Detail+Overview Approach to Visualize Music Elements

Gabriel D. Cantareira, Luis G. Nonato, Fernando V. Paulovich

Abstract—A music piece contains a large amount of information represented as a series of instructions corresponding to notes that must be played at specific times. These simple notes are combined to form complex harmonic structures that can be difficult to identify and analyze. Due to its simplicity and straightforward interpretation, music sheets, and piano rolls have been the visual metaphor employed by most music visualization tools to support interpretation. Albeit it can represent all necessary elements to perform a music piece, these metaphors do not explicitly show many of the patterns and structures inherent to music arrangements, such as rhythm progression and harmonic interactions, needing users to create a mental model of them. Moreover, comparing different pieces and visualizing how a particular instrument track relates to the others is an issue not only for music sheet-based techniques but also for most existing music visualization methods. In this paper, we present a novel visualization framework, called *Music Overview, Stability and Harmony Visualization (MoshViz)*, which facilitates the visualization and understanding of music renditions, focusing mainly on the visual analysis of specific musical instruments. Our approach creates a high-level model of music data and highlights structures of interest, enabling a detail+overview visualization to assist users in the task of identifying harmonic and melodic patterns. The usefulness and representativeness of MoshViz is confirmed by a set of user tests which demonstrate that the proposed visual metaphor matches, with a high degree of accuracy, the mental model of different users regarding the recognizable patterns of sounds.

Index Terms—Music visualization, Overview representation

1 Introduction

Music pieces are complex arrangements of notes that, when played at the right times, result in harmonious and melodic songs [1]. However, figuring out the structure and patterns of a music piece is not a trivial task, demanding both knowledge in music theory and time. Building a mental model of a music piece is a time-consuming activity even for experienced musicians. Either by reading the score or listening to a performance of the piece, the data needs to be analyzed sequentially. Several visualization-assisted methodologies have been proposed to support musicians on this complex task [2][3], facilitating the understanding of music structures and how they evolve over time.

Although such methods represent a valuable support for musicians, most common metaphors typically do not provide visual mechanisms to identify patterns and structures inherent to musical data, such as harmonic variations and structural repetitions. This information can be useful when studying music pieces, be it for composition, practice, performance, or improvisation, as well as understanding the relations between instruments and identifying features that can characterize a particular song or musician. The approaches that do offer this sort of information [4][2] usually focus on only one type of pattern, leaving no room for analysis on how these different levels interact. Moreover, such methods are mainly designed to support local analysis, shown in a sequential manner; they do not offer a general view to aid comprehension of context. Some other concepts, such as sound stability, which measures how the music piece moves away from its “tonal center” considering rhythmic and harmonic elements, and level of performance complexity, which refers to an estimative of skill and practice time needed to perform the piece, are also not adequately addressed. Visualization of these concepts would allow users to highlight more efficiently points of interest and grasp details of how a music piece sounds, being a valuable asset when studying a piece or screening through a music library.

In this paper, we propose a novel visualization framework, called *Music Overview, Stability and Harmony Visualization (MoshViz)*, that enables users to visualize structures and patterns in music pieces while providing visual information about aspects such as harmony, stability, and complexity of specific musical instruments. MoshViz creates a high-level model of the musical data and highlights aspects of interest enabling a detail+overview interpretation, thus assisting users in understanding musical structure as well as its harmonic and melodic patterns. The music piece is depicted as a whole, and the users can select specific time frames for further exploration.

While our framework can be used to analyze any music piece, in this paper we focus on visualizing electric guitar performances. Thus, some of its properties and applications are adjusted as such.

In summary, the main contributions of this paper are:

- The definition of a rich framework that presents music structure in many different layers, such as temporal, melodic and harmonic;
- A visual metaphor that can compactly represent music, allowing a faster interpretation while retaining melodic and harmonic elements;
- A novel approach to quantify and visualize musical instability, helping users to identify points of interest.

The remainder of this paper is structured as follows: Section 2 presents existing approaches for song visualization and discusses their main drawbacks. Section 3 details our approach showing how the detail view and the overview are constructed. Section 4 presents some results and use cases. Finally, Section 5 describes conclusions.

G.D. Cantareira, L.G. Nonato and F.V. Paulovich are with the Institute of Mathematics and Computer Science, University of São Paulo, São Carlos, SP, Brazil. E-mail: {gabrieldc,gnonato,paulovic}@icmc.usp.br.

2 Related Work

The most standard approach to visualize musical notes and their progress over time is the piano roll metaphor [1], which consists of plotting all notes on a two-dimensional space where the horizontal axis represents time, and the vertical axis represents pitch. Despite its simplicity and straightforward interpretation, the piano roll metaphor does not allow for easy visualization of relevant information such as patterns of repetition, the overall structure of a music piece, and harmonic interactions.

Different attempts have been made towards defining enhanced visual representations. Smith and Williams [5] propose a 3D version of a piano roll where the z-axis is used to convey extra information such as tracks and timbre variations. The Music Animation Machine, by Malinowsky [6], is an animation-based system which depicts notes as geometric entities which are highlighted when a note is played. The Comp-i system, proposed by Miyazaki *et al.* [7], aims to unveil structures contained in the music piece and offers, in addition to a three-dimensional representation, a cone-tree based model [8] to enable different exploratory scenarios. Although highly informative, allowing the analysis of note concentration and trends, these methods do not provide resources for visualizing the entire music piece and its structures at once or understanding its musical properties at a deeper level.

Watanabe *et al.* [3] devised a system, called BRASS, designed for music learning and performance visualization. BRASS offers a focus+context view of a musical score and allows the visualization of areas with high concentration of notes, “louder” sound sections and sections where notes are played faster. However, melodic information cannot be visualized, which hampers the identification of patterns and themes.

Based on a different concept, the Shape of Song by Wattenberg [4] processes the content of MIDI files to find repetition patterns using a string matching algorithm. These patterns are visualized through translucent arches that connect the repetitive profiles. The goal is to show the interconnectivity and repetition of sequence of notes on a music piece. This technique offers an effective visualization model but is focused only on the display of repeating sections throughout the song, not showing any information regarding what sequences of notes or melodic patterns are being repeated. Wolkowicz *et al.* [9] propose a visualization which aims to map and compare features from MIDI files using similarity matrices. While informative and offering a global view, the model is also focused only on repetition structures.

Ciuha *et al.* [2] proposed a method to visualize concurrent musical tones over a music piece based on color saturation. Colors are assigned to different tones according to their consonance expressed on a circle of fifths. When different notes are played at the same time, a vector-based model is used to stack colors, resulting in saturated colors for consonant notes and gray tones for dissonant notes. This representation is then combined with the piano roll metaphor, allowing users to observe the tonal variation during the music execution. Although this method can be used to show the distribution of tones along different sections, the system does not enable an overview of the whole music piece.

Proposed by Syndal and Hearst [10], the ImproViz method enables the visualization of melodic and harmonic trends of different musicians by analyzing transcriptions of their improvisations. The visual metaphor relies on two principal concepts. The *Melodic Landscape* is a continuous map of played notes that conveys musical phrasing information, and the *Harmonic Pallete* is a visual representation that allows the comparison of notes played with the base chords of a song so as to reveal the trend of each musician in playing a particular combination of notes. The main issue of such representation is the visual clutter when dealing with larger song excerpts.

Some other music visualization methods share similarities with the method we propose in this paper. For instance, the techniques proposed by Mardirossian *et al.* [11], Hayashi *et al.* [12], Bergstrom *et al.* [13], and Chan *et al.* [14] provide additional insight on how chord structure is formed, which are the semantic structures involved and the recognition of roles for different instruments. However, when compared to our approach, they present different goals, such as producing animated visualizations or multi-track score abstractions.

Our framework addresses most limitations of existing techniques, providing the general view of musical information as well as depicting information of particular notes, thus allowing the analysis of music pieces into different levels of abstraction. It supports an overview of a whole song through its main musical properties, such as the temporal structure of the piece, containing chorus, bridges, important themes and unique moments, and a more detailed view of the music elements, such as melody and harmony, scales and intervals, and correlation between notes. Also, it allows the analysis of the predictability of sequences of sounds/notes in songs, rendering a unique analytical capability. The existing techniques do not support so many different levels of information at the same time.

3 Music Overview, Stability and Harmony Visualization (MoshViz)

MoshViz relies on two different views, namely, the overview and detail view, which are coordinated through a linking-and-brushing strategy. Figure 1 presents the main window of the prototype tool we have implemented to test our approach, composed by the overview and detail view representations. The detail view depicts the individual notes played. The layout is an extension of the piano roll metaphor enriched with additional information such as a base chord for each time measure, the time signature, and tempo, and the impact and consonance of each note. We design this framework to support the analysis of the roles played by each note in specific sections of a music piece.

The overview enables the visualization of a song as a whole. In this view, notes are condensed into time signature measures which are the basic units in the visual representation. Each time measure is represented by a box whose size and color reflect the variation of the notes density and pitch in the corresponding time frame. Additional information such as complexity of note sequences, repetition levels, and interval variations are also depicted in the overview through heatmaps, with the intent of generating unique patterns, similar to the metaphor employed by the Literature Fingerprint technique [15].

The two views contain a set of visual features to assist users to identify and explore aspects of interest. Users can visually correlate the information shown in the overview and detail view to easily identifying patterns, high or low complexity regions, changes in rhythm and harmony, among other areas of interest. The visualization is focused on analyzing instructions corresponding to a single instrument in a track (or combination of tracks) of a MIDI file. This track is compared to another track which

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2016.2614226, IEEE Transactions on Multimedia

Fig. 1. Our prototype tool. The main window is split into two parts, overview and detailed view. From the overview, it is possible to conclude that the depicted song presents different structures of repetition with different levels of concentration, while the overview shows how these structures are formed. The section circled in red and marked as (A) highlights a pattern that is repeated in large segments, while the section circled in red and marked as (B) highlights a section with high complexity and note variation. Clicking on a measure the system highlights similar ones in green, making patterns visible.

contains the chord structure of the music piece, such as a bass, base guitar or even all the other instruments combined. From now on, the tracks under analysis will be called "Main Track" and "Bass Track", respectively.

Next subsections detail the functionalities of our framework.

3.1 Detail View

The detail view relies on an extension of the piano roll metaphor. The vertical axis has a pitch range of six octaves; the four commonly observed on an electric guitar fretboard, plus an additional octave for both higher and lower pitches. Notes are represented by circles followed by lines that indicate the duration of the note, drawn on a grid split horizontally according to the time measures and vertically into octaves. The time signature, current tempo, and ID of each time measure (for reference) are also depicted in the background. Figure 2 shows an example of a detail view representation. The circle notes are shown at different sizes, according to a measure we call *Impact*. The *Impact* estimates the influence of each note in how a section of the piece sounds, with larger values indicating the notes that are louder or are played at stronger beats according to pulse stress patterns in music theory [16][17]. The *Impact* is computed according to the following equation,

$$I_n = S_n * V_n \tag{1}$$

where V_n is the velocity of note n extracted from the MIDI message. This value is linearly normalized according to the speed range. S_n is an approximation for the beat strength of note n according to its position inside the time measure, given by

$$S_n = \begin{cases} 1.00 & \text{the note is at the first beat} \\ 0.50 & \text{the note is at the first beat after half measure} \\ 0.25 & \text{otherwise} \end{cases}$$

The values of I_n are normalized throughout the song, assigning $I_n = 1.0$ to the note with highest impact value.

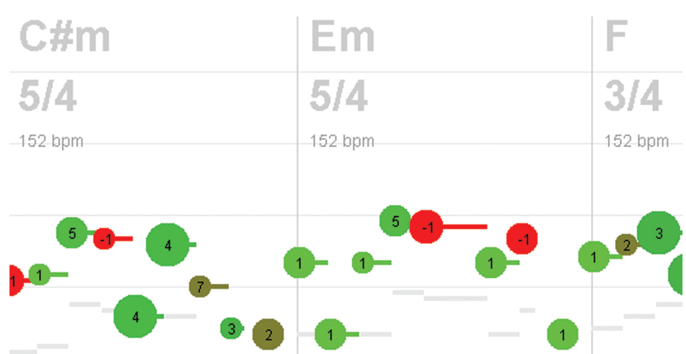


Fig. 2. Detailed view representation. Each note is represented as a circle with size proportional to its impact and the color mapping its consonance, from red (dissonant) to green (consonant). As major and minor chords are recognized, intervals that do not belong to the scale being used are marked as negative.

To explore consonance and harmony, MoshViz estimates the base tone for each time measure from the bass track. Our framework provides two simple mechanisms to estimate the base tone. The first mechanism simply looks for the lowest note played at a strong beat of the time measure in the bass track. That note is then considered as the base tone for that measure, and it is used as a reference when analyzing the notes played on the main track. We call this mode *Unsigned Harmony* because it does not distinguish between major and minor intervals. The second mechanism relies on chords. The note found with the first mechanism is considered the root, and by estimating the harmonic field over the main key from the music piece, the tone is assigned as a major or minor chord. Although presenting some limitations, in particular for pieces with complex harmony patterns, estimating the base tone by chords results in more precise harmony calculations.

Once the base tone of each measure is obtained, the consonance is calculated for each note. The consonance C_n of a note n is computed by obtaining the interval between the note and the base tone. Consonance values range in $[0,1]$ from the less consonant to the most consonant intervals according to how consonance is usually perceived in western music [1]. The harmony and consonance models can, however, be changed to fit different musical styles, employing different scales. These values are then expressed by the color of the note circle, from red to green. The number of the detected interval is also written in the center of the circle.

Besides providing the usual piano roll notation for users who want to have a closer look at the notes being played, the detail view also ties the overview to the note-by-note low-level representation. This happens mainly due to its use of note impact and consonance, which are also used to calculate many of the overview's features (discussed in the next section). For instance, the reason for a time measure to be labeled as unstable in the overview is revealed when the detail view shows a highly dissonant note with a high impact value (a big red note).

Both consonance and overall impact properties are automatically set, but our framework allows users to tweak values to better reflect properties of certain genres or music styles.

3.2 Overview

Visualizing Note Information. The overview contains information about the whole music piece in a compact view. All the notes on the main track of the MIDI file are condensed along with additional information about complexity, structure and important changes in the MIDI meta-events. Each time measure is represented by a colored rectangle with the height proportional to the pitch range of the notes contained in the measure. Given the i^{th} measure M_i, the height of the corresponding rectangle is given by

$$H(M_i) = \frac{\max(M_i) - \min(M_i)}{PitchRange} \times H_{max} \tag{2}$$

where $\max(M_i)$ and $\min(M_i)$ return the maximum and minimum pitches of the notes in M_i, respectively. *PitchRange* is the range of values that the pitches can assume, so that the left side of the equation ranges in $[0,1]$ (in MIDI note messages, the values for pitch are contained in a byte ranging from 0 to 127). H_{max} is maximum height of a rectangle, in pixels. Figure 3 presents an example of an overview representation.

The transparency level of each rectangle is proportional to the number of notes played during that time measure. The more dense a time measure is, regarding played notes, the more opaque is the associated rectangle. The opacity is a value ranging in $[0,1]$ and is given by

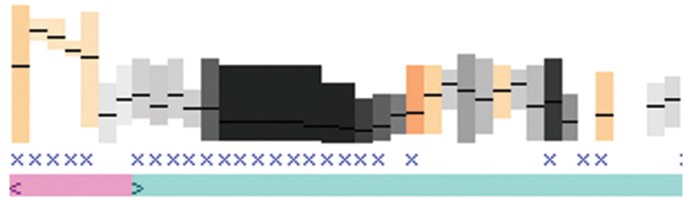


Fig. 3. Overview representation. A rectangle represents a time measure with its height proportional to the pitch interval, transparency representing its density and color mapping its instability. A line inside a rectangle shows the mean value of the notes contained. MIDI meta-events, such as tempo alterations and time signature changes, are represented by glyphs.

$$B_{M_i} = \min\left(\frac{|M_i|}{\alpha}, 1.0\right) \tag{3}$$

where $|M_i|$ defines the number of activated notes in M_i and α the number of notes deemed, by the user, as the maximum considered density, so that, if $|M_i| > \alpha, B_{M_i} = 1$.

A horizontal line is drawn within each rectangle to complement the information related to the pitch range. The vertical position of the line is defined as the weighted average of the pitch of all notes played during the measure, using the duration of the note as weight. Its vertical position L_i is computed as

$$L_i = \frac{\sum_{n \in M_i} F_n D_n}{\sum_{n \in M_i} D_n} \times \frac{H_{max}}{PitchRange} \tag{4}$$

where F_n and D_n are the pitch value and duration of the note n, respectively. The overview's vertical axis range in $[0, H_{max}]$, corresponding to pitch values ranging from 0 to the maximum pitch value defined in *PitchRange*.

From the position of these lines, one can visually convey whether there is a disproportional amount of notes clustered somewhere inside the pitch range.

We also compute the instability value Ins_i of each measure M_i, which estimates the overall unpredictability of the sound at that point of the music piece. The instability is defined considering different aspects, including what notes are played, when they are played, and whether there are changes in the rhythm or tempo. Highly dissonant sounds, sounds that are played at unexpected moments and overall chaotic time structures are known to add unpredictability to a musical segment. Mathematically, our measure for instability is computed as

$$Ins_i = \min\left(\left(\frac{\sum_{n \in M_i}(1 - C_n) I_n}{\sum_{n \in M_i} I_n}\right) + a_i + b_i, 1.0\right) \tag{5}$$

where $a = 0.15$ if there is no note on the first beat and $a = 0.0$ otherwise, and $b = 0.1$ if the time signature changes in the measure and $b = 0.0$ otherwise. The reasoning for using a and b in this equation is to add the perception of unpredictability when time signatures changes or if an absence of sound occurs in moments where it is not expected, such as at the first beat of a measure. The instability ranges in $[0,1]$ and it is set to 1 if it exceeds this range. On the visual representation, an instability threshold controlled by the user discriminates highly unstable time measures. While areas recognized as stable are colored in grayscale with opaqueness according to Equation 3, unstable time measures use the same equation but are colored in shades of red.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2016.2614226, IEEE Transactions on Multimedia

Also, the symbol "X" in blue is placed below a time measure rectangle to indicate a change in time signature and the symbols "<" and ">", in pink and green, respectively, are used to indicate a drop or raise on tempo. Figure 3 illustrates the visual attributes described above.

Visualizing Complexity, Interval Variation and Repetition. Three distinct color bars below the time measures rectangles convey information about the complexity, interval variation and repetition level of each measure. This information is depicted using color scales ranging from white to a different color for each bar, from low to high values, respectively.

In our framework, complexity is the overall estimation of musical clutter, mainly regarding performance. Fast segments with a high density of different notes are considered more complex, usually being harder to play and understand when studying. We estimate complexity of the measure M_i as follows

$$CP_i = \frac{ct_i * bpm_i * cm_i}{\beta} + a \tag{6}$$

where ct_i is the number of unique timestamps for playing notes, cm_i is the number of unique notes played, bpm_i is the current tempo (in bpm), and $a = 0.15$ if there was a change in time signature and $a = 0.0$ otherwise, as time signature changes tend to require attention when studying or practicing a music piece. The parameter β is a normalization factor and represents the product of maximum values for note timestamps, unique notes, and bpm on the entire music piece.

The repetition level RP_i is used for highlighting areas containing repetition patterns so they can be easily recognized. RP_i is equal to the number of times an exact sequence of notes in a measure M_i appear in other time measures, divided by a parameter γ for normalization. The Interval variation IV_i counts how many different intervals are played on the measure M_i, regarding the base tone. It allows for identifying sections with more varied harmony, such as elaborated chords or complex melodies. Figure 4 shows the visual representation of complexity (green heatmap), repetition (purple heatmap), and interval variation (blue heatmap).

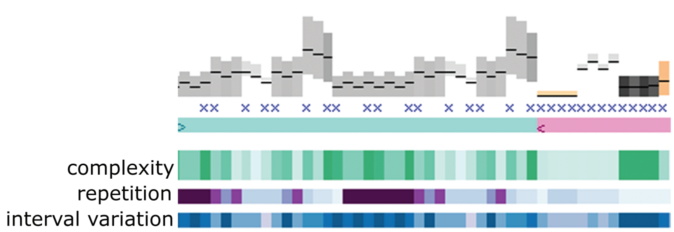


Fig. 4. Visual representation of complexity, interval variation and repetitions. Heatmaps are used below the time measures rectangles to map additional information. Complexity, repetition and interval variation are shown in green, purple and blue, respectively.

The choice of values for the parameters α, β and γ, along with the constants for specific conditions mentioned above, enables a high degree of flexibility to the proposed framework towards users' needs and expectations. The easiest way to set such parameters would be to set them according to the highest value found in the entire music piece. However, this would change the meaning of some features as the visualization would only be useful to compare a song to itself. Absolute references are needed to compare different songs. It is possible to analyze a lot of songs and set the parameters to the highest overall values found, but that may leave the configuration vulnerable to outliers, either from data errors or simply music pieces with atypical values. Our decision was to set empirically the values based on the observation of the behavior of a collection of MIDI transcriptions since these values are only used as static references. We recognize that a statistical modeling of these parameters may improve the results. However, this could be sophisticated enough to render an entire study.

Our framework also enables an interactive mechanism to group measures to simplify the overview. Consecutive measures are combined and represented using only one rectangle, compacting the produced visual representation while maintaining information about flow and patterns in the note sequences. This is further discussed in the next section.

4 Results and Use Cases

In this section, we present and discuss the features of the MoshViz framework through use cases scenarios. We start by analyzing individual songs.

4.1 Overview and detail of individual songs

In Figure 1, the song under analysis is a transcription of *Hangar 18*, by Megadeth. This figure presents information about the solo guitar track. In the remainder of this section, we also use the solo guitar track from MIDI transcriptions for analysis. On the overview representation, it is easy to get insights about the musical structure of this song. It can be seen that some small sequences of similar time measures are sequentially repeated (circled in red and marked as *(A)*), composing larger sequences that are repeated two times during the song. Other information can also be captured from this view. There is a concentration of high complexity time measures at the end of the song (as seen in the green heatmap), followed by a high interval variation (the blue heatmap) and spikes of shapes in the measures that do not match with anything that appears before (circled in red and marked as *(B)*). This suggests that something unique might be happening in this part of the song. By listening to the song, one can notice that this part refers to a guitar solo segment.

For this song, the overview shows few red spots, suggesting a stable-sounding piece. There are almost no time signature variations (the blue "Xs") and the most frequent interval to start a measure, by far, is an octave, so the most consonant possible sound usually sits at the strongest position of the time measures, which matches the discussed concept of stability. By listening to the song, it is possible to notice this stable structure. The song contains over 170 time measures with notes, but only about 70 of those are unique, meaning there is a lot of repetition, which further increases the idea of structure and stability.

To show the capability of MoshViz in capturing features of complex music data, Figure 5 presents the analysis of a transcription of *For the Love of God*, by Steve Vai, a song focused on solo guitar performance. Aside from a few initial repetitions of measures (circled in red and marked as *(A)*), there are no clear patterns in the main guitar track. From the 143 measures, 137 are unique, meaning almost no repetition. The octave is still the most frequent interval to start a measure, but it appears at only 27 out of the 143 of them. For this song, more red segments are identified, indicating a much higher degree of instability than the previous song. Besides, the many pitch variations (shown as the different boxes on the overview and the blue heatmap) suggest a more

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2016.2614226, IEEE Transactions on Multimedia

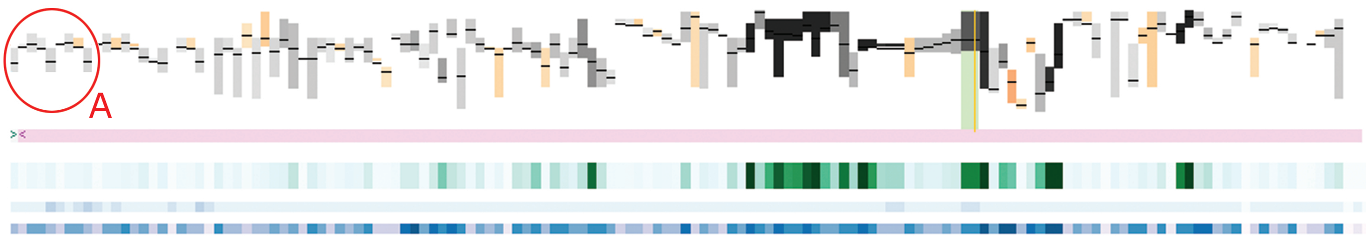


Fig. 5. Variations in complexity and stability of a more complex piece. Few repetitions are observed with a high degree of instability which matches what is expected for a song focused on a solo guitar performance.

melodic piece. The differences in complexity between Figure 1 and Figure 5 can also be easily spotted on the detailed views. The two different presented sections depict a highly different amount of notes with different activation times contained in the time measures of each section. The differences in complexity are confirmed on the complexity heatmap (green heatmap) and the interval variation heatmap (blue heatmap). Listening to both songs, it is easy to confirm these differences.

Notice that these features cannot be easily analyzed with other existing visualization approaches. For instance, the BRASS [3] system and the Shape of Song [4] visual metaphor, although providing an overview, do not support the representation of the melodies (sequences of notes) contained in the song, which holds valuable information, without the focus being applied. On the other hand, the method proposed by Ciuha *et al.* [2] and the Improviz [10] technique, while offering information focused on melody and harmony, do not allow any sort of global view. In fact, the existing methods either lack a general view of the data or only offer views oriented to the observation of a single feature, such as the concentration of sounds or the repetition of motifs and may miss information regarding how these features interact (see Section 2). Differently, our approach allows the observation of features such as repetition structures while also providing some notion of *what* is being repeated, still being able to identify inconsistencies. Also, the detail view provides more information on a local level about the role and importance of each note on the resulting sound, which can be useful whether a user is trying to understand a music piece or is aiming at creating something new on top of it.

When a song is selected for detailed examination, the overview can be used to guide the process. Figure 6 shows the overview generated from another MIDI transcription. The data shown is one of the guitar tracks from *The Evil That Men Do*, by Iron Maiden. The piece contains many repeated measures. From 173 measures with notes on this track, only 67 are unique. By looking at Figure 6(a), it is possible to easily identify many repeating patterns. In Figure 6(b), structures were circled according to the actual content of the song. The areas circled in yellow are the chorus. They are always preceded by the same melody, circled in blue. The verses are shown circled in red while the area circled in purple is a unique solo. The area circled in green is a unique intro, although it is somewhat similar to the melodies circled in blue. It is important to notice that these patterns and structures remain visually recognizable even when there are some changes when a segment is repeated.

Once such patterns are understood, most of the song can be performed or improvised upon. It can be observed that there is an instability focus on a unique section after stable repeated sections (purple area on Figure 6(b)). This is caused by dissonances due to a key change, indicating that this section needs a more careful examination before performance.

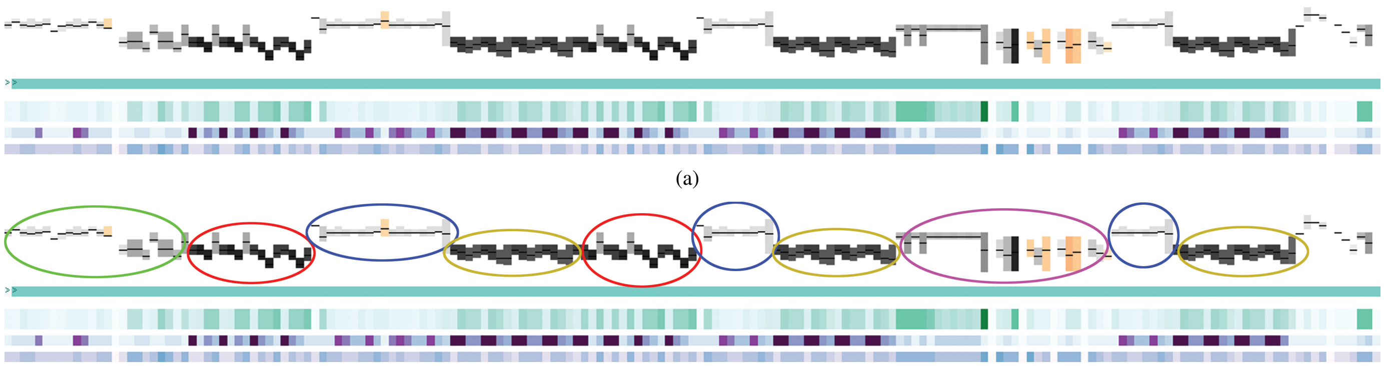
Finally, to show the capability of MoshViz on condensing the overview representation, Figure 7 presents different levels of simplification. This figure shows a transcription of the distorted guitar track from a 23 minute music piece, *A Change of Seasons*, by Dream Theater, containing 633 time measures. To compose this figure, the number of time measures represented by each rectangle (Vs) varies from 1 to 8. Although some fine detail is missed, the main patterns are preserved. Each rectangle can be drawn with as little as a single pixel of length, so this whole piece can be shown using only 640 pixels without loss of information. Applying the simplification up to $Vs = 8$, the image now takes about 80 pixels horizontally, indicating the high degree of compression that can be attained by the overview. This allows a user to have dozens overviews of different songs on the screen at the same time and quickly look for patterns or perform comparisons, as is shown in the next section.

4.2 Comparing songs

Due to the compact visual representation provided by MoshViz, it is possible to get insights about a music piece without the need of sequentially traversing it. This enables the design of different applications, including the fast comparison among various songs towards the identification of pieces with desired or undesired properties.

Figure 8 shows an example of comparison. It presents the visual outcomes of transcriptions of two different songs, *Paranoid*, by Black Sabbath, and *Summer Song*, by Joe Satriani. Although both pieces present similar lengths, their contents are musically very different. For instance, consider the case of a user wishing to study and practice a song to perform it during a future presentation. If the user's goal is to practice something simple and more likely to master quickly, the song in Figure 8(a) is probably a better choice. It has much more repetitions and lower complexity, as shown in both the green and purple heatmaps and in the time measure representation. The song presented in Figure 8(b), however, has many unique segments with higher complexity values, indicating that it might need more practice time for an accurate performance. The overview generated by MoshViz is able to instantly highlight these features, showing information that would otherwise be understood only after listening to both songs entirely or reading their sheet instructions.

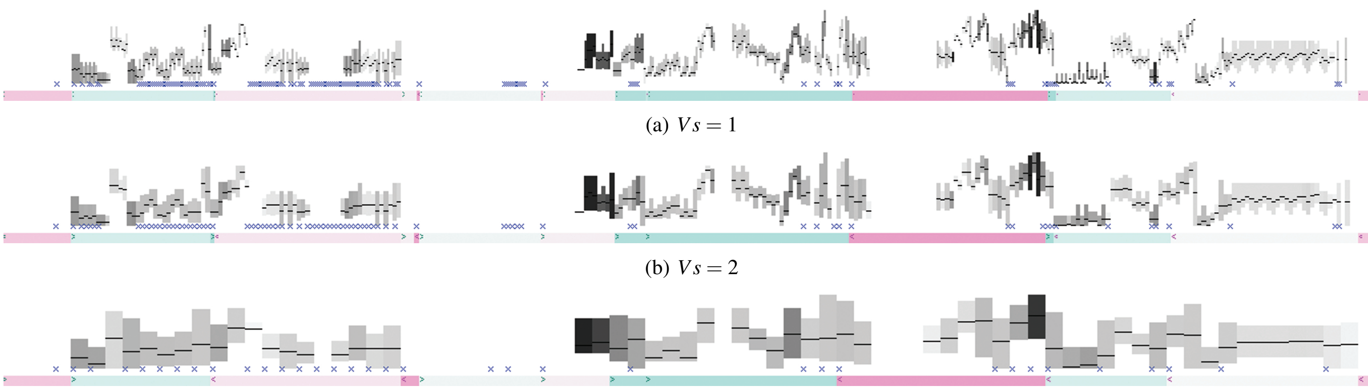
When dealing with a completely unknown musical composition collection, user interest may lie only in particular types of songs. For example, those with stable structures, below a certain



(a)

(b)

Fig. 6. Points of interest on the overview. (b) shows the same song as (a) but with intro (green), verse (red), chorus (yellow and blue), and solo (purple) parts highlighted. After stable repeated sections, one instability spot is identified on a unique section, suggesting that this might need a more careful examination.



(a) $Vs = 1$

(b) $Vs = 2$

(c) $Vs = 8$

Fig. 7. Overview visualization of a 23 minutes long song, containing 633 time measures. (a) 1 measure per rectangle. (b) 2 measures per rectangle. (c) 8 measures per rectangle. Although some fine detail is lost, the main patterns are preserved.

level of complexity or without repeated sections. When assembling a repertoire, this information is important to a musician to decide the role of each song in a performance routine: long unstable segments might get tiresome while a strong chorus can inject energy to the public when played at the right time. By exploring such collection with MoshViz, the answers to some specific questions can be obtained in a more immediate fashion, reducing the user effort if compared to the usual approach of listening and studying entire music collections searching for pieces that match certain constraints.

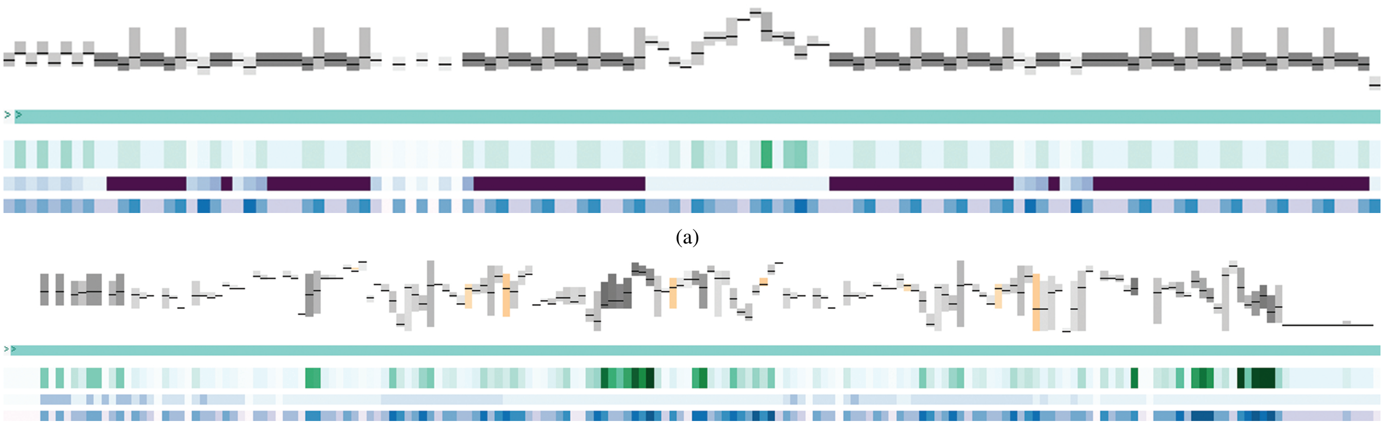
5 Conclusion

In this paper, we presented a visualization framework called *Music Overview, Stability and Harmony Visualization (MoshViz)*, that, not only offers a detail+overview representation of a song but also can represent melodic and harmonic elements. It enables users to grasp the relations between musical properties of note sequences and their impact on the whole piece quickly, helping the process of song interpretation and analysis. The development of the framework was advised and validated by a professional musician.

While existing approaches only employ the loudness or tonality of sounds to estimate their influence, our method also takes into account beat strength, placing each sound under the musical composition's structure. Also, the presented strategy for detecting and visualizing instability is a novelty in music visualization, highlighting unstable sections that would not be shown considering only the consonance.

Our approach, besides being able to support users on tasks that involve studying and analyzing certain music pieces, can also be used to help on the decision of *what* pieces to study, based only on the visual representation, without the necessity of listening to the entire song. Therefore, possible applications include building repertoires for musical performances and practice. Using the pitches of the 12-tone chromatic scale, our method offers an alternative to the standard major/minor chord identification by using the "unsigned harmony" model.

MoshViz has, however, some limitations. One of them is that, although unlikely to happen, two different melodies with similar features can produce similar overviews, leading to erroneous interpretations. Also, the values calculated for instability and complexity are intended to work as guidelines to highlight points of interest. Since the notions of complexity and instability can be subjective and may vary between different users, the same visual representation can be interpreted in various ways. Some of the approaches we use to estimate these values are specific to how consonance and stability are usually perceived in western music. For instance, some music pieces apply strength on traditionally "weak" beats (Syncopation), but as such rhythms are designed to be unexpected, our estimation of instability remains working as intended. Nevertheless, these notions are controlled via parameters



(a)

(b)

Fig. 8. Differences between guitar tracks from two music pieces. (a) *Paranoid*, by Black Sabbath, and (b) *Summer Song*, by Joe Satriani. The first image has clear, regular patterns, while the second is more unpredictable and prone to variation.

that users can change according to a personal point of view.

For future work, some features can still be expanded or improved, such as ways to estimate values of complexity and uncertainty to fit particular views or styles of music. Our base tone recognition mechanism is simple and intended to highlight highly consonant or dissonant intervals. A more robust chord recognition system could improve harmony analysis. Lastly, the grouping done by the overview scaling is not content aware, meaning it does not give priority to blocks or repetition structures when grouping time measures. Better solutions would improve the visual outcome. Additionally, there are still unexplored features in music theory that could be used to enhance our model to detect other important elements in musical composition. A more comprehensive analysis regarding different types of music, such as complex classical compositions, may also give more insights on how to identify musical patterns and structures.

Acknowledgments

We would like to thank the anonymous reviewers for their constructive comments. This work was supported by the São Paulo Research Foundation (FAPESP) (grants #2013/02455-5 and #2011/22749-8) and CNPq-Brazil.

References

[1] M. Hewitt, *Music Theory for Computer Musicians*. Course Technology, CENGAGE Learning, 2008.

[2] P. Ciuha, B. Klemenc, and F. Solina, “Visualization of concurrent tones in music with colours,” in *Proceedings of the International Conference on Multimedia*, ser. MM ’10. New York, NY, USA: ACM, 2010, pp. 1677–1680.

[3] F. Watanabe, R. Hiraga, and I. Fujishiro, “Brass: Visualizing scores for assisting music learning,” in *Proceedings of 2003 International Computer Music Conference*, 2003, pp. 107–114.

[4] M. Wattenberg, “Arc diagrams: visualizing structure in strings,” in *Information Visualization, 2002. INFOVIS 2002. IEEE Symposium on*, 2002, pp. 110–116.

[5] S. Smith and G. Williams, “A visualization of music,” in *Visualization ’97., Proceedings*, 1997, pp. 499–503.

[6] S. Malinowski. (2014) Music animation machine. [Online]. Available: http://www.musanim.com

[7] R. Miyazaki, I. Fujishiro, and R. Hiraga, “comp-i: a system for visual exploration and editing of midi datasets,” in *Proceedings of 2004 International Computer Music Conference*, 2004, pp. 157–164.

[8] G. G. Robertson, J. D. Mackinlay, and S. K. Card, “Cone trees: Animated 3d visualizations of hierarchical information,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI ’91. New York, NY, USA: ACM, 1991, pp. 189–194.

[9] J. Wolkowicz, S. Brooks, and V. Kecelj, “Midivis: Visualizing music pieces structure via similarity matrices,” in *International Computer Music Conference*, 2009.

[10] J. Snydal and M. Hearst, “Improviz: Visual explorations of jazz improvisations,” in *CHI ’05 Extended Abstracts on Human Factors in Computing Systems*, ser. CHI EA ’05. New York, NY, USA: ACM, 2005, pp. 1805–1808.

[11] A. Mardirossian and E. Chew, “Visualizing music: Tonal progressions and distributions,” in *ISMIR*. Austrian Computer Society, 2007, pp. 189–194.

[12] A. Hayashi, T. Itoh, and M. Matsubara, “Colorscore – visualization and condensation of structure of classical music,” in *Information Visualisation (IV), 2011 15th International Conference on*, July 2011, pp. 420–425.

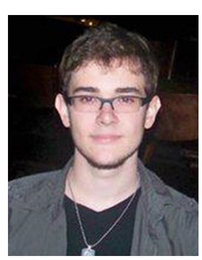
[13] T. Bergstrom, K. Karahalios, and J. C. Hart, “Isochords: Visualizing structure in music,” in *Proceedings of Graphics Interface 2007*, ser. GI ’07. New York, NY, USA: ACM, 2007, pp. 297–304.

[14] W.-Y. Chan, H. Qu, and W.-H. Mak, “Visualizing the semantic structure in classical music works,” *Visualization and Computer Graphics, IEEE Transactions on*, vol. 16, no. 1, pp. 161–173, Jan 2010.

[15] D. A. Keim and D. Oelke, “Literature fingerprinting: A new method for visual literary analysis,” in *Visual Analytics Science and Technology, 2007. VAST 2007. IEEE Symposium on*, Oct 2007, pp. 115–122.

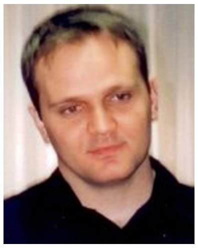
[16] W. Rothstein, *Phrase rhythm in tonal music*. Schirmer Books, 1989.

[17] W. Berry, *Structural Functions in Music*, ser. Dover Books on Music Series. Dover, 1976.



Gabriel Dias Cantareira received the MSc degree in computer sciences from the University of São Paulo, São Carlos/SP, Brazil. He is currently working towards a PhD in computer sciences in the Institute for Mathematics and Computer Science (ICMC), University of São Paulo, São Carlos/SP. His research interests involve computer graphics and information visualization, specializing in visual analytics.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2016.2614226, IEEE Transactions on Multimedia



Fernando V. Paulovich is an associate professor in computer science at the Instituto de Ciências Matemáticas e de Computação (ICMC), at the Universidade de São Paulo (USP), Brazil. He holds a Ph.D. degree in computer sciences from the ICMC-USP, with a period as invited researcher at the Delft University of Technology, the Netherlands. His research interests involve computational visualization, more specifically information visualization, visual analytics and visual data mining. He is a member of the Brazilian Computer Society.

Luis Gustavo Nonato received the PhD degree in applied mathematics from the Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro — Brazil, in 1998. He is a full professor at the Instituto de Ciências Matemáticas e de Computação (ICMC) — Universidade de São Paulo (USP) — Brazil. He spent a sabbatical leave in the Scientific Computing and Imaging Institute at the University of Utah from 2008 to 2010. Besides having served in several program committees, including IEEE Visualization and Eurovis, he was a member of the editorial board of Computer Graphics Forum and the president of the Special Committee on Computer Graphics and Image Processing of Brazilian Computer Society. Currently Dr. Nonato leads the Visual and Geometry Processing Group at ICMC-USP.