Similarity Graph: Visual Exploration of Song Collections

Jorge H. Piazentin Ono, Débora Cristina Corrêa, Martha Dais Ferreira, Rodrigo Fernandes de Mello, Luis Gustavo Nonato
{jorgehpo, deboracorrea, daismf, mello, gnonato}@icmc.usp.br
Institute of Mathematics and Computer Science
Universidade de São Paulo
São Carlos, Brazil

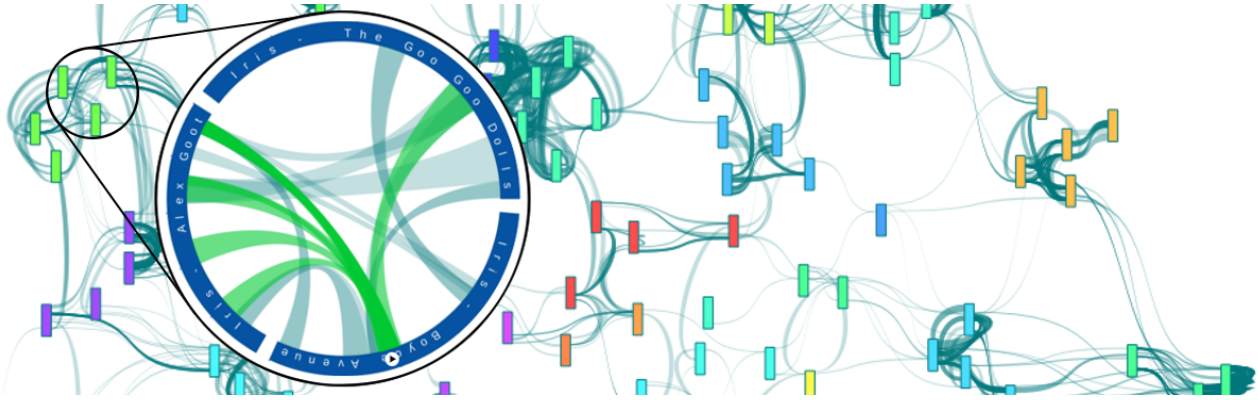

Fig. 1. Visualization of a cover song data set. *Back*: Global Similarity Graph, showing the entire data set. *Detail*: Local Similarity Graph, showing three covers of the same song ("Iris", by "Goo Goo Dolls", "Alex Goot" and "Boyce Avenue"). Rectangles with the same color represent cover songs of the same title.

***Abstract*—Music collections are widely available on the Internet and, with the increasing storage and bandwidth capability, users can currently access thousands of songs, what brings challenges to music organization and exploration. Therefore, there is a growing demand towards automated Music Information Retrieval (MIR) tools for organizing, retrieving and processing music data sets. Recent visualization systems have focused in showing music similarities using audio data and multidimensional projection. Nonetheless, they do not take into account similarities between parts of songs, that is, they compare pieces as a whole, neglecting the aspect that some of them have similarities only in small parts. This work proposes a novel methodology for users to visually explore music collections considering that the similarity can take place only in small parts of the song. Our approach uses MIR to find similar segments between pairs of audio files and a graph metaphor to display the detected similarities.**

***Keywords*-Information Visualization; Music Information Retrieval;**

I. Introduction

The multidisciplinary area of Music Information Retrieval (MIR) aims at automatically describing, investigating, retrieving and organizing music content [1]. MIR approaches are typically characterized by the manipulation of raw data, in an attempt to extract relevant features to account for different tasks, such as song classification, cover songs identification and audio segmentation.

Several approaches have been proposed to segment audio signals in order to improve the interactive aspects of media players, seeking for relevant features that provide suitable song fingerprints as well as detecting song onsets. However, most studies are focused on modeling an individual song without relating it to others present in a data set [2]. This gap motivates our work, in which a new visualization mechanism is proposed to explore segment similarities among different songs. The features provided by our mechanism are useful to support several tasks such as the identification of cover songs, plagiarism detection and mixing.

In summary, our methodology works as follows. First, we extract chroma features from the audio data, and apply a hierarchical segmentation algorithm to locate meaningful audio segments for every song. Then, the comparison of the resulting snippets is performed with the use a robust similarity metric, derived from the Recurrent Quantification Analysis [3]. Finally, the visualization uses a graph-based metaphor to show global similarities among all songs in a database. It also provides an additional possibility to explore local similarities, i.e., the cross comparison of all segments among a small group of songs.

Contribution: The main contribution of this work is the graph-based visualization mechanism, which enables a hierarchical similarity analysis of song collections and the display of information in different levels of detail simultaneously.

A. Related work

Systems that explore the visualization of music collections try to map songs to the visual space in order to represent their similarity relationships. The criterion used to quantify similarity is either established from meta-data (e.g. tags) or from features associated with the audio content. These systems also provide user interaction, the visualization of groups of songs and, in most cases, the creation of playlists. Examples of systems that assess the visualization of music datasets are briefly summarized in the following. The reader is invited to address their respective references for further details.

As examples of systems for the visualization of music collections that are based on meta-data we mention the proposals by Torrens et al. [4], and Dalhuijsen and van Velthoven [5]. Torrens et al. [4] considered meta-data to explore music collections and produce playlists. They employ music tags such artist, composer, year, album and genre to organize songs. The visualizations are based on three different metaphors: circle, box and treemap. In a similar approach, Dalhuijsen and van Velthoven proposed a tag-based visualization tool called Music Nodes, which also supports the management and exploration of music collections [5]. However, differently from the previous proposal, users can associate a song to more than one genre (for example, 40% rock and 60% electronic) as well as establish new tags to proceed with the classification. Albums are positioned in a bi-dimensional space according to a force-based system: albums of the same genre are arranged nearby, while albums from different genres are pushed apart. Users can select and export playlists through textual searches or by using visual selections.

Among the visualization approaches based on audio content are the studies by Pampalk et al. [6] and Paulovich et al. [7]. Pampalk et al. [6] designed a visualization method called Islands of Music, which extracts rhythm characteristics and applies Self-Organizing Maps (SOM) to group feature coefficients in the data space. The motivation for the name "islands" comes from the metaphor produced by the SOM density map in the bi-dimensional space. Paulovich et al. [7] proposed a local multidimensional projection technique called Piecewise Laplacian-based Projection (PLP), and applied it to visualize music collections. JAudio [8] was executed on every song to produce a feature vector with statistical measures of the audio, beat histograms and spectral analysis, which were then projected onto the screen with PLP. The authors also provided interaction capabilities so users can perform selections and create playlists.

II. Similarity Graph

The construction of the Similarity Graph is divided into two main steps: pre-processing of the data set, and the visualization itself. Both steps are described in this section.

A. Pre-processing

As a first stage, our approach extracts chroma features using the Harmonic Pitch Class Profile (HPCP) [9]. HPCP takes the audio signal as input and divides it into consecutive overlapping windows along time. Then, the Fourier Transform is applied on every window to produce frequency-based feature vectors. By appending those vectors, one obtains the spectrogram matrix, in which rows refer to frequencies and columns to time. As a next step, HPCP detects magnitude peaks (local maxima) present in every spectrogram column, which are maintained as the main features. This results in a chromagram that basically represents all harmonic characteristics of a song along time.

The next pre-processing stage is responsible for the hierarchical segmentation of songs, performed using the algorithm proposed by McFee et al. [10]: it looks for repeated patterns by analyzing the Self-Similarity Matrix (SSM), built with the comparison the pairwise distances of column vectors (i.e., harmonical features) contained in a song chromagram. SSM will contain the value 1 at position i, j if column vectors i and j are in the neighborhood of each other (considering a radius ϵ), or zero otherwise. After obtaining SSM, we apply a window mode filter on it to attenuate noise and emphasize diagonal lines. In fact, those diagonals are associated with temporal recurrent patterns in the audio signal [3].

After the normalization procedure, the SSM becomes an adjacency matrix R', which represents a Markov process. Every cell at position i, j in R' corresponds to the transition probability from state i (associated to a column vector in the song chromagram, i.e., the harmonic features produced at a given time window) to state j (another column vector in the chromagram). R' is then added to another binary matrix W, which represents the relationship among consecutive states $(i, i+1)$ and $(i, i-1)$ as follows:

$$W_{i,j} = \begin{cases} 1 & \text{if } |i-j| = 1 \\ 0 & \text{otherwise} \end{cases}$$

A weighted sum is used to combine R' to W, using a weighting factor μ. A suitable value of μ is chosen so that the probability of a Markov process of going to a similar random time (R') is equal to the one of going to an adjacent time (W). The combination of R' and W is made as:

$$R^*_{i,j} = \mu R'_{i,j} + (1-\mu) W_{i,j}$$

Finally, we build a Laplacian matrix from R^* and apply spectral clustering on it to find the repeated segments (or patterns) in a song. The Laplacian matrix L is defined by the following equation, in which D is the diagonal matrix containing the degrees of the nodes in graph R^*:

$$L = I - D^{-1/2} R^* D^{-1/2}$$

Let $Y \in \mathbb{R}^{n \times m}$ be the eigenvectors associated to the m-smallest eigenvalues of L. Then, we apply the K-means clustering algorithm over the rows of Y, having $k = m$. More specifically, n is the number of time windows in a song chromagram and m is the number of segments we want to detect, discarding repetitions. One can build a hierarchical segmentation by varying m. This hierarchy is useful for visualization purposes, once users may be interested in identifying

segments under different levels of detail. In this work, the user may choose which values of m are used, $m = 1...10$ being the default configuration.

Finally, we divide every song chromagram into the respective obtained segments. After segmenting all chromagrams, we then pairwise compare such parts of different songs to cross correlate them. This comparison is conducted by using the Recurrence Quantification Analysis (RQA). RQA is computed on top of the Cross Recurrence Plot (CRP), which builds a matrix in the same way as the SSM does. RQA is composed of several different measures, but we particularly employ the Q_{max}: it computes the longest diagonal line in the CRP and summarizes the temporal similarity between segments in different songs. This approach is commonly employed to study the stability and the trajectory of dynamical systems, what indeed motivated us to apply it on the pairwise comparison of audio segments (for more details we recommend [3]).

B. Visualization

Similarity Graph is an interactive graph-based visualization, in which the user first explores the data in a entire perspective (Global Similarity Graph), and then, in further inspections, the user can select a portion of the songs to be visualized in more details (Local Similarity Graph). Hermite and Bézier splines are used to create the visualizations. For a review of these curve interpolations, see [11].

1) Global Similarity Graph: In order to represent the complete song collection, the songs are projected into the two-dimensional space using a dimensionality reduction technique. The literature agrees that t-SNE [12] results in one of the best 2-D mappings in terms of neighborhood preservation and group segregation [13] [14]; therefore, it is the chosen method to be used in our visualization pipeline.

First, we compute the projection of the song data set, i.e. each song is represented by a vector in $\mathbb{R}^2$. Given the Q_{max} similarity matrix between pairs of songs (segmentation parameter $m = 1$), we calculate a dissimilarity matrix D, $D_{i,j} = (Q_{max_{i,j}} + 1)^{-1}$, and project the songs with the t-SNE optimization scheme with respect to D.

Next, we represent the similarities among pairs of segments graphically. Let S_i and S_j be two segments of length l_i and l_j, respectively. In order to check if two segments are similar, we compute the normalized Q_{max} measure between them and check if it is greater than a threshold. The normalized measure is given by:

$$Q'_{max}(S_i, S_j) = \frac{Q_{max}(S_i, S_j)}{\min(l_i, l_j)}$$

In this work, a threshold of 0.8 was empirically used. More specifically, this constraint imposes that, for two segments to be considered similar, there must be a match in at least 80% of the Cross Recurrence Plot. Since the size of the segments can vary greatly, another restriction is determined: $l_i \leq 2l_j$ and $l_j \leq 2l_i$.

In the proposed visualization scheme, each song is represented by a rectangle glyph, with height greater than width. The top of the rectangle corresponds to the beginning of the song and the bottom, to its ending. The glyph is positioned according to the t-SNE projection.

Similar segments among all songs and their k-nearest neighbors (in the projected space) are connected with a cubic Hermite curve. More specifically, the start of segment S_i is connected to the start of segment S_j and the end of S_i is connected to the end of S_j. Only the k-nearest neighbors are connected to avoid too many crossing edges in the visualization. In our experiments, we used the neighborhood $k = 10$.

The cubic Hermite interpolation is used to represent the segment similarity. With this interpolation, one can specify a curve with predefined endpoints and tangents. In the Similarity Graph, two curves connect a pair of rectangles if there is enough similarity between their segments. The tangents are chosen to be horizontal vectors and their magnitudes depend on the time in which the similarity occurred in the highest positioned song. The closer the endpoint is to the top (0 seconds), the greater is the vector. Fig. 2 shows the two possible configurations for the connection with Hermite curves: if the horizontal distance between boxes is greater than two times the box width (W), configuration (a) is used. Otherwise, (b) is employed. Note that vertical and horizontal reflections of both configurations can occur.

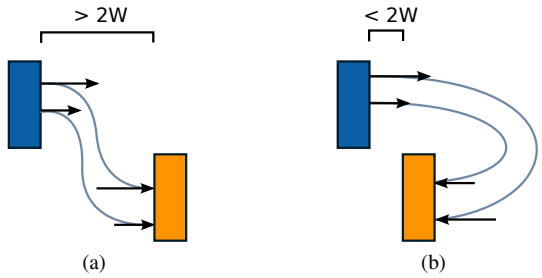

Fig. 2. Hermite curves connecting two songs in the global visualization.

2) Local Similarity Graph: The user may be interested in analyzing a portion of the data set in more details. Thus, the Local Similarity Graph was developed to this end. This visualization displays songs as arcs on the screen and, like the global visualization, connects similar song segments. However, it enables the discovery of the selected songs with two interactive mechanisms: segment audio reproduction and highlighting.

The visualization is developed as follows: first, each song is displayed as an arc within a circle, whose angle sweep is proportional to the duration of the song when compared to the total duration (considering all selected songs). Then, all similar segments of the selected songs are connected by arcs, drawn with two quadratic Bézier curves. The start of segment S_i is connected to the start of segment S_j and the end of segment S_i is connected to the end of segment S_j if S_i and S_j are similar. More specifically, three control points are used for each curve, with the middle control always at the center of the circle. Fig. 3 shows a scheme of how two segments are connected in the Local Similarity Graph. The segment of the top song (blue), starting at A and ending at B, is similar to the

segment of the bottom song (orange), which starts at C and ends at D. Note that the connection curves toward the center of the circle O.

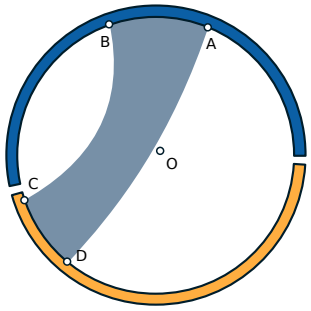

Fig. 3. Bézier curves connecting two songs in the global visualization.

Two interactive mechanisms make songs exploration easier in the Local Similarity Graph: segment play and segment highlight. The user can play segments from the songs of interest, which enables him to recognize why the segments are similar. Since the circular graph layout can have many overlapping edges (arcs), the feature of snippet highlighting facilitates the understanding of the graph: when the user plays a segment, all similar snippets are highlighted, simplifying the graph analysis. These two mechanisms will be exemplified in the next section.

III. Results

The proposed technique was used to visualize a personal cover songs data set. The song collection was developed in the context of the present work and contains a set of cover songs of popular music available on the video sharing platform Youtube. It consists of 5 cover versions of 30 different songs (150 songs in total). This data set was built in order to exemplify our methodology, since we expect that a large amount of cover songs results in a large number of similar segments between them.

Fig. 1 shows the visualization of the data set with the Global and Local Similarity Graphs. Due to the limited size of this article, only a portion of the global graph was displayed. Note that cover songs (represented by rectangles of the same color) are positioned close to each other in the visualization, which suggests that both the dissimilarity metric and the projection technique are suitable for the task at hand.

The user can select a group of songs in the global visualization to explore them in the local graph context. This task is also displayed in Fig. 1, in which three versions of the song "Iris" are compared. Note that some arcs are displayed in a brighter green. These arcs connect segments similar to the song snippet being played, facilitating the understanding of the visualization.

IV. Conclusion and Future Work

In this paper, we proposed a technique for the exploration of music collections that differs from the previous work in the sense that it compares musical segments between songs, as opposed to complete songs, in order to create a visual representation of the data set.

The features present in the Similarity Graph are useful for the understanding and discovery of music collections. They facilitate the exploration of similarities between songs by enabling the identification of similar segments in the data with different levels of details. This is possible due to the hierarchical segmentation employed.

A drawback of this technique, however, is its scalability: we use large glyphs to represent the temporal evolution of songs in the global scheme, thus, the visualization supports only a few hundreds of songs. Future work includes the study of visual metaphors to mitigate this problem. A first approach will be the use of focus + context methodologies, for example, the operations of zooming and fish eye, to enable the simultaneous display of a larger number of songs.

Acknowledgment

This work is supported by FAPESP (#2011/22749-8, #2012/17961-0, #2012/24801-0, #2014/13323-5) , CNPq (#132239/2013-2, #441583/2014-8, #303051/2014-0, #302643/2013-3) and Capes (#7901561/D).

References

[1] N. Orio, "Music Retrieval: A Tutorial and Review," *Foundations and Trends in Information Retrieval*, vol. 1, no. 1, pp. 1–96, 2006.

[2] J. Paulus, M. Müller, and A. Klapuri, "Audio-Based Music Structure Analysis." in *ISMIR*, 2010, pp. 625–636.

[3] J. Serrà, X. Serra, and R. G. Andrzejak, "Cross recurrence quantification for cover song identification," *New Journal of Physics*, vol. 11, no. 9, pp. 1–20, Sep. 2009.

[4] M. Torrens, P. Hertzog, and J. L. Arcos, "Visualizing and Exploring Personal Music Libraries." in *ISMIR*, 2004.

[5] L. Dalhuijsen and L. van Velthoven, "MusicalNodes: The Visual Music Library," in *Proceedings of the 2010 International Conference on Electronic Visualisation and the Arts*, ser. EVA'10. Swinton, UK, UK: British Computer Society, 2010, pp. 232–236.

[6] E. Pampalk, "Islands of music: Analysis, organization, and visualization of music archives," Ph.D. dissertation, 2001.

[7] F. Paulovich, D. Eler, J. Poco, C. Botha, R. Minghim, and L. Nonato, "Piece wise Laplacian-based Projection for Interactive Data Exploration and Organization," *Computer Graphics Forum*, vol. 30, no. 3, pp. 1091–1100, 2011.

[8] C. McKay, I. Fujinaga, and P. Depalle, "jAudio: A feature extraction library," in *Proceedings of the International Conference on Music Information Retrieval*, 2005, pp. 600–3.

[9] E. G. Gomez, "Tonal description of music audio signals," Ph.D. dissertation, Universitat Pompeu Fabra, Barcelona, 2006.

[10] B. McFee and D. P. Ellis, "Analyzing song structure with spectral clustering," in *ISMIR - International Society for Music Information Retrieval Conference*, 2014.

[11] D. Hearn and M. P. Baker, *Computer Graphics, C Version*. Prentice Hall, 1997.

[12] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of Machine Learning Research*, vol. 9, no. 2579-2605, p. 85, 2008.

[13] S. Ingram and T. Munzner, "Dimensionality reduction for documents with nearest neighbor queries," *Neurocomputing*, vol. 150, Part B, pp. 557–569, Feb. 2015.

[14] S. G. Fadel, F. M. Fatore, F. S. L. G. Duarte, and F. V. Paulovich, "LoCH: A neighborhood-based multidimensional projection technique for high-dimensional sparse spaces," *Neurocomputing*, vol. 150, Part B, pp. 546–556, Feb. 2015.