Interactive High-dimensional Audiovisual Feature Analysis of Live Song Concert Videos

Alexandra Diehl, Melike Çiloğlu, Renato Pajarola;
University of Zurich, Switzerland

Abstract

We present an interactive visualization tool to explore high-dimensional features of audiovisual data extracted from a video archive of live music performances. Our tool presents overviews of data features, similarities between song recordings, and details of the extracted visual and audio features. Features are extracted using neural networks, signal processing techniques, and audio analysis tools. Furthermore, we present a similarity metric to measure how different relevant recordings are compared to other videos. We illustrate our approach via use cases showing initial results to analyze song features, compare songs, identify outstanding songs, and detect song clusters.

Introduction

Music visualization applies music information research approaches to extract meaningful features and then present them through interactive visualization. There is vast related work on music visualization tools that use audiovisual analysis. However, most of the existing visualization tools focus on studio-recorded songs and ignore the effects of a live performance ambiance on the songs. Previous music visualization studies used a limited amount of features to analyze songs. Most studies have relied on notes and chords, and few have analyzed musical features such as tempo, genre, mood, etc. Nonetheless, the literature lacks visualization tools that analyze musical performance videos in detail using a broader amount of audiovisual features, including both video and audio-related features to analyze musical instruments, frequency spectrum, or visual clutter.

We apply state-of-the-art video and audio analysis methods to fill this gap and visualize songs based on a list of relevant live song performance features. We performed a requirement analysis with the help of a domain expert and identified the important features needed for the analysis of live song performances. The selected features were then extracted using neural networks, signal-processing techniques, and audio analysis tools. Furthermore, we present a similarity metric to measure how different a specific song recording is compared to all other videos. Using this measure, it is possible to identify outstanding songs that are significantly different than others or to detect song clusters.

Our main contribution is an interactive visual tool, shown in Figure 1, that integrates live performance features into a set of intuitive, aesthetic, and engaging visualizations. Our visual design provides both an overview visualization and a detailed song analysis view. The overview visualization communicates clusters and similarities between different recordings. The detailed view presents extracted visual and audio features of a song.

Related Work

We divided the related work into (1) antecedents that focus on visualizing song similarities rather than describing each song individually using its features and (2) related work that focuses on describing single songs based on their features.

Visualization of Music Collections

AudioRadar [8] is an exploratory interactive visualization tool for music collections. Using a radar view as a metaphor to visualize a song, a song is added to the center of a circular visualization, and similar songs are placed around it. The comparison is based on four automatically analyzed dimensions, whether the song is melodic or rhythmic, slow or fast, clean or rough, calm or turbulent. The four-dimensional song space is also projected into two dimensions by user-based feature selection. Although AudioRadar allows for a detailed analysis of songs' features, it only presents an overall representation, whereas our tool presents both detailed radial and timeline visual encodings. We also allow for zoom, filter, and details on demand.

Another example is MuVis [5], a visualization tool that utilizes treemaps for browsing and filtering songs to create playlists. The extracted features are tags, track title, duration, genre, song release dates, and fluctuation patterns. The tool provides similarity-based positioning according to the properties of songs which helps users add similar songs to a playlist. It is an interactive tool supporting feature filtering and writing queries to navigate around the treemap easily. Although very powerful to quantify and order songs' features, this work is not thought to analyze temporal patterns, as our solution do.

Walshaw [17] proposed a tool to explore relationships between folk and traditional song collections using a given melodic similarity measure. Datasets with multiple corpora are considered, exploring the similarity metric between each pair of songs. Using these similarity scores, color-coded corpus graphs are constructed using the node-link representation. However, unlike other tools, the implemented visualization is not interactive.

Jorge et al. [14] proposed a visualization that considers similarities between small parts of songs, not comparing songs as a whole. The visualization consists of an overview and detailed view named global and local similarity graphs, respectively. The global similarity graph uses t-SNE to map songs into a 2D space and connects songs according to overall song similarities using Hermite curves. Color coding highlights different artists' covers of the same songs. While this solution presents a powerful similarity analysis, it lacks a detailed temporal analysis and comparison of individual song features.

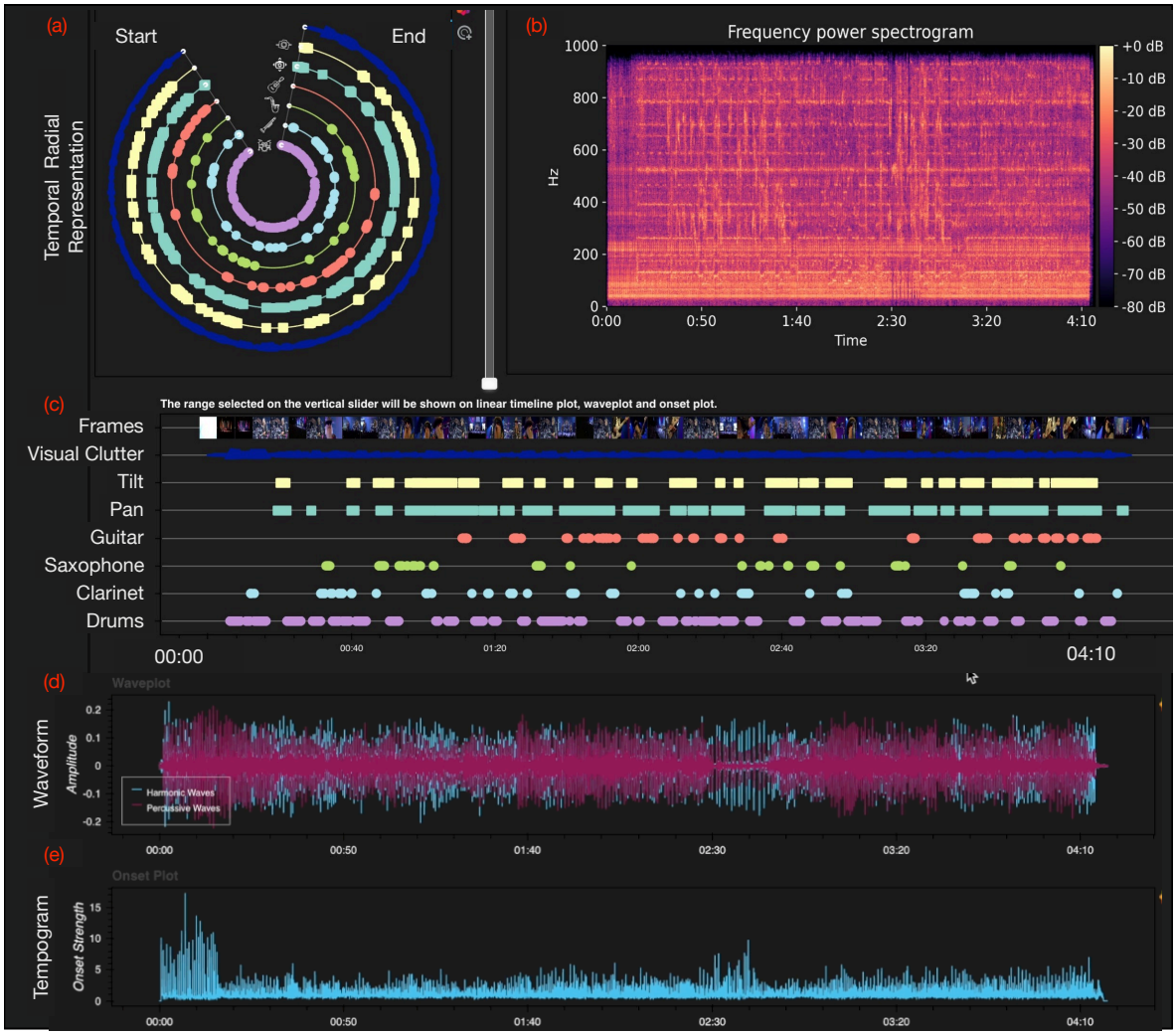

Figure 1. *Screenshot of our solution. (a) presents a radial overview of the temporal distribution of audiovisual features for a selected song, (b) shows the power spectrogram view of the song, and below shows the timeline view displaying the (c) discrete and (d,e) continuous audiovisual features over time.*

Song-based Music and Video Visualization

SongVis [12] is designed to present semantic descriptors of music using icons. Songs are analyzed to extract the semantic features using music information retrieval techniques, and icons are assigned for each feature. For example, depending on the speed of the tempo, a rabbit or a turtle icon is assigned to the song.

Misual [11] uses the waveform of audio as the main source of information, and the visualization uses lighting and shading techniques to make it appear 3D. Hence, the shape's volume encodes the audio signal's power. Repetitions through the song are detected using the mel-frequency analysis and are color-coded, making it easy, e.g., to identify the chorus part. The main goal of the visualization is to help people categorize music, and the design is intuitive and easy to understand. However, the features considered are limited, and the work does not help analyze musical data.

Isochords [1] is an animated visualization that aims for music classification based on structure. It uses MIDI files to extract the musical events and visualizes them in a Tonnetz grid, encoding major and minor chords using the arrow orientation. Isochords allows users to see the musical events while listening to them. Furthermore, examples show how the visual representation of the music of different genres differs clearly. This work is designed as an animation to be used as an ear training aid rather than a multiple-feature analysis tool.

ImproViz [15] focuses on jazz music to analyze improvisations and explore different musicians' harmonic styles. Manual analysis is needed to prepare the visualization and was done for the song "All Blues". In this study, different musicians' harmonic palettes were explored. ImproViz only visualizes one song and considers only the notes while doing so.

Similarity Based Music Collection Visualization

The previous approaches mentioned above either focus on music collection visualization without visualizing individual songs or focus on single-song visualization and ignore similarities between songs. The work of A. Soriano et al. [16] provide visualizations of similarity-based collections and song details. It con-

structs unique icons for each song based on the MIDI information encoding properties of individual notes played in the song. The resulting icons are vertical elements that are composed of multiple bars of different colors and sizes according to the underlying structure of the song. It is possible to see an overview of all songs on a scatter plot, where they are placed according to similarity. Also, it is possible to see the icons of each song closely to get more information about the song. However, even though using MIDI files is a great way to create a similarity metric, the icons created based on them are not very intuitive or easy to understand for inexperienced people.

Feature Extraction

The input of this work consists of individual live song videos, and corresponding metadata, from the Montreux Jazz Digital Project (MJDP) [6, 7] archive. The metadata consists of the date of the concert, in which hall the concert was performed, and the performing artists.

Visual Features

The following visual features were extracted from the video frames using image processing techniques:

Camera motion: Using varying camera motion to control the viewer's attention is widely applied by directors. It is of interest to know how often panning, tilting, and zooming are used throughout a video and whether it reveals a pattern. This is a global feature and also interesting to explore in the context of concert videos. For example, zooms towards the artist or the musical instrument during a solo could be identified as a pattern. Using the optical flow between frames it is possible to calculate the dominant flow angle and magnitude, and predict the camera motion. For camera motion extraction, a motion detection repository on GitHub [3] was used which is based on OpenCV. The method extracts optical flow between two consecutive frames and determines the dominant flow angle and magnitude to decide whether and how the camera is moved.

Visual clutter: The cognitive load for watching a video can be related to visual clutter in the frames of the video. Clutter is a context-free feature that can reveal interesting patterns in different types of movies. To measure visual clutter, a feature congestion method was used. Accordingly, the clutter in an image is correlated to the variance of luminance contrast, color, and orientation of the image. Assuming that if these properties change significantly through an image, the image is cluttered. We used a Python library to measure clutter [10] which implements a feature congestion algorithm. It is possible to individually calculate color, luminance contrast, orientation, and overall clutter values.

Musical instrument detection: In video analysis, one of the most important tasks is to determine focus points, e.g. through applying object detection to the frames to extract the visible objects. In the context of a live song video, it is valuable to extract the musical instruments. For this purpose, a pre-trained convolutional neural network for musical instrument detection was used. The model returns 30 confidence values representing the probability of 30 instruments being visible in a given video frame. Then the confidence values are thresholded so that only the ones with high confidence values are considered. The model we used was trained on 4793 images to identify 30 different musical instruments [9].

Audio Features

The following audio features are considered to be crucial and were calculated using audio and signal processing techniques:

Tempo: The tempo of a song is a powerful tool for composers to convey a feeling to the audience. Knowing the tempo may also help to understand the mood during the concert, audience reaction, and even the genre of the song. Despite the possibility of varying tempo through live performance, one single averaged tempo was calculated in the unit of BPM. We used Lybrosa [13] for the tempo, tempogram, spectrogram, and waveform audio features.

Tempogram: Besides the song's tempo, it would be compelling to see whether the song speeds up or slows down during a live performance. In studio recordings, it is very unlikely to observe varying tempo in jazz, rock, or pop songs. However, for exploratory data analysis of live performances, we decided to consider tempo in a time-dependent manner, based on the initiation point of musical events. These are called *onset* and they are directly related to the tempo of music at a given time.

Power spectrogram: The power spectrogram shows the various frequencies' signal strengths over time. It provides a quick overview of the audio signal and helps people grasp the dominant frequencies. The spectrogram is extensively used and is crucial for music analysis. Even though it is mostly used by area experts, it is easy to understand for the general audience.

Waveform: The waveform describes the amplitude of the audio signal over time. It is the most basic form of audio visualization very intuitive for the general audience, the wave rises as the sound increases and it falls as the sound decreases. In our case, the wave has been separated into two main components, harmonic and percussive waves. Harmonic waves are created by the melodic sounds and percussive waves are created by the beats like hits on drums. This separation helps users analyze which component is dominant over time.

Dominant musical instruments and instrument families: Musical instrument detection in polyphonic songs is still an open challenge for which no definitive solution exists in the literature. We use a pre-trained sound classification model, Essentia [2] to extract the dominant musical instruments and instrument families, mood, and genre detection.

Dominant mood and genre: A common categorization of music is according to the mood and genre of a song. These properties are usually related to the tempo and musical instruments but are not completely determined by them. We used Essentia [2] for dominant mood and genre as well.

Similarity Metric

With respect to comparing different songs based on the above described audio/video features, we decided against the common Euclidean distance and cosine similarity metrics due to the influence of individual attributes' value ranges, possibly dominating the metric, ignoring relative values and not considering

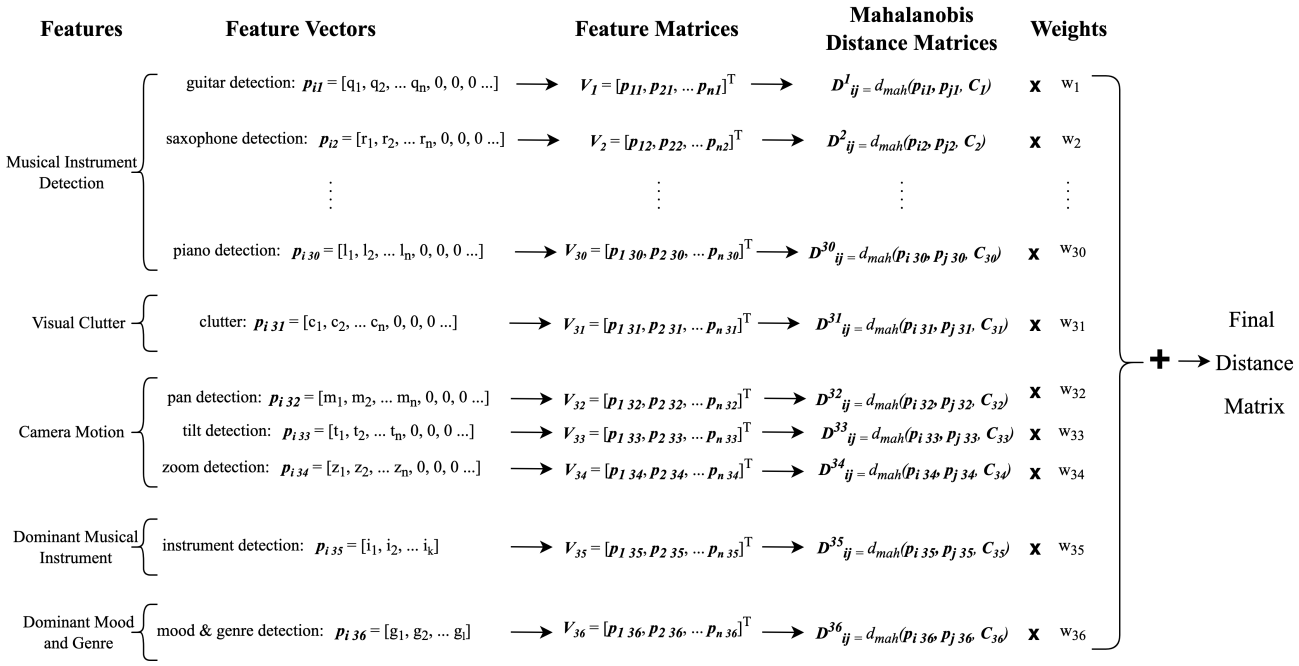

Figure 2. *Overview of the distance metric calculation based on the extracted audiovisual features.*

correlations between attributes. Chebychev and Manhattan distances suffer from the same unit or value range problems and both also neglect possible correlations in data.

The Mahalanobis distance measures the distance between a point and a data distribution. It computes how far a point is from the mean of a distribution in terms of standard deviation along the principal component axes. Hence it is expected to see correlations between different features. Furthermore, important properties are that it is scale-invariant and unitless, which is beneficial as there is no consensus on the units of our extracted features. Most features result in probabilities, but, for example, visual clutter returns scalars.

Given a data point feature vector p and a distribution D with its mean value vector μ and covariance matrix C, the formulation is as follows:

$$d_{mah}(p,D) = \sqrt{(p-\mu)C^{-1}(p-\mu)^T} \tag{1}$$

Some audio/video features were prioritized and structured as follows:

- The musical instruments are captured in a matrix M of size 30 x Number of Frames in the Video, with each entry M_{ij} being the probability of instrument i appearing on frame j, thresholded to 0 if less than 0.5.
- For the camera motions the outcome is a matrix C of size 3 x Number of Frames in the Video where C_{ij} is 1 if motion i is detected on frame j, 0 otherwise.
- The visual clutter feature is a vector of scalars with elements as many as frames in the video
- The outputs of dominant musical instruments and dominant mood and genre features for all videos are vectors of probabilities for each one of the instruments, mood, and genre classes. Overall the lengths of these vectors are the same for each video.

In our implementation, we applied the Mahalanobis distance to individual features and combined the resulting scalars into one distance value. In contrast to flattening all features into one long vector, this makes it easier to interpret the role of each feature on the distance and also the result is simpler to explain to the users.

Note that for some features their vector or matrix sizes are related to the length of the video, but the Mahalanobis distance assumes vectors of equal length. A simple workaround is padding with zeros to make all vectors the same length as the longest video.

The principle to calculate the distance is shown in Figure 2. For musical instrument detection, visual clutter, and camera motion, since the lengths of the vectors are related to the length of the video, zero filling is applied to make the vectors of different videos the same length. For dominant musical instruments, mood, and genre detection, this is not needed as these represent one vector per video. In Figure 2, p_{ik} represents the different feature vectors k for a song video i. In the feature matrices V_k, we collect the feature vectors $k = 1 \ldots 36$ of all data points $i = 1 \ldots n$ (rows).

The covariance matrices C_k needed for the Mahalanobis distance are defined individually for each feature. The calculation of the point-to-point distances is given by

$$d_{mah}(p_{ik}, p_{jk}, C_k^{-1}) = \sqrt{(p_{ik}-p_{jk})C_k^{-1}((p_{ik}-p_{jk})^T}, \tag{2}$$

where p_{ik} and p_{jk} are the i- and j-th data points' k-th feature vectors. For each pair of data points i, j, the distance is calculated for each feature k and combined into one distance matrix D^k_{ij}, where the element ij is the distance between feature vectors p_{ik} and p_{jk}.

Finally, a linear combination of the distance matrices is formed to get the final distance values. The default (average) weights $1/36$ can be adjusted by the user to emphasize certain feature groups. According to the use case, some features might be more important than others, and their roles in the distance can be manipulated.

For visualization purposes, multi-dimensional scaling [4] (MDS) was used to plot songs in a two-dimensional space based on their pairwise similarity values. Furthermore, K-means clustering can be applied to detect song clusters based on their feature similarities. The outcome of the similarity calculation phase is the locations of the performance videos in the two-dimensional Euclidean space and clusters constructed using the K-means algorithm.

Visual Design

Our design choices aim at simplicity, intuitiveness, and attractiveness. Our visual design comprises four main components. Therefore, we chose a radial view that resembles a phonograph record to show the time-dependent features, a linear timeline view that allows the users to explore the evolution of the song's features over time, and a 2D scatterplot visualization that shows songs as dots clustered in space by similarity.

For color coding, we map each feature to a different color as in Figure 6(b). For the musical instruments, the instruments in the same family are represented by similar colors. For example, percussive instruments such as drums and tambourines are both encoded by shades of purple.

Radial Overview

For the overview plot, we designed a radial view resembling a phonograph record as shown in Figure 3 which is endearing for music visualization. We aim to prompt the user to explore the features with an engaging and attractive design. The features visualized are visual clutter, camera motion, and the detected musical instruments. Since all of these features are time-dependent, we re-arranged them into a radial timeline. Radial timeline representations are great for presenting periodic features and are appealing to the human eye. In the context of music visualization using a radial layout aligns with the periodicity of the music. In some cases, radial representations can be difficult to understand for the users. However, since this layer only serves as an overview and will be complemented with a detailed feature-view plot, it is not considered to be a problem.

In our radial plot of Figure 3, simple glyphs (squares, circles, and triangles) are used to differentiate between different features (visual clutter, camera motion, and musical instruments). The radial location of the glyph encodes the time in which the feature occurs (threshold detection). Since the visual clutter is calculated for all the frames, the triangles look continuous and show the continuously varying clutter value per frame. In radial layouts, it is important to clearly indicate to the user where the origin point of time is and in which direction the time flows. To help with this issue, white start and end points were added with additional icons at the starting line to express the features visually. Furthermore, the gap between the beginning and ending points encodes the length of the song so that it is possible to visually compare the duration of different songs.

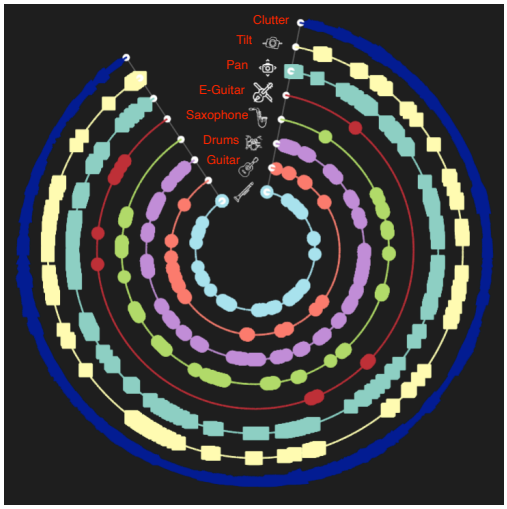

Figure 3. *Phonograph record like radial view of the time-dependent features of a song.*

Linear Timeline

We designed a linear timeline to enable an individual analysis of all features in a comprehensive way, as shown in Figure 1(c). The features to visualize in the timeline consist of visual clutter, camera motion, musical instruments, waveform, tempogram, and power spectrogram. The top section of the timeline view is a straightened horizontal version of the radial overview plot, using the same color encoding for visual clutter, camera motion, and musical instruments to link the radial overview with the other detailed views. The x-axis indicates the time when the feature has occurred. Instead of using icons, the names of the features are placed on the y-axis to be more expressive. The top row visualizes the video frames over time.

Detail Views

For the waveform (waveplot) and the tempogram (onset plot), as shown in Figure 1(d) and Figure 1(e), the colors were selected to be bright and distinct to catch the viewer's attention on the dark background. Again, the x-axis encodes the time, and the legend and axis labels were put to make the visualization easy to interpret. The power spectrogram as shown in Figure 1(b) is commonly used in audio analysis. It displays the strengths of frequencies (y-axis) over time (x-axis) in dB units. For color coding, a perceptually uniform color map was chosen. Since the application background is black, the lightest color was set to represent the highest number because the light colors attract more attention on dark backgrounds.

Similarity-based Overview

The aim of the similarity-based overview is to present multiple songs in a single visualization to allow comparison, as shown in Figure 4. After the calculation of pairwise song similarities, the songs are mapped into two-dimensional locations and clustered using MDS and K-means clustering, respectively. To present the clusters, a simple scatter plot is used, highlighting each cluster by connecting the data points by color-coded lines to the cluster center.

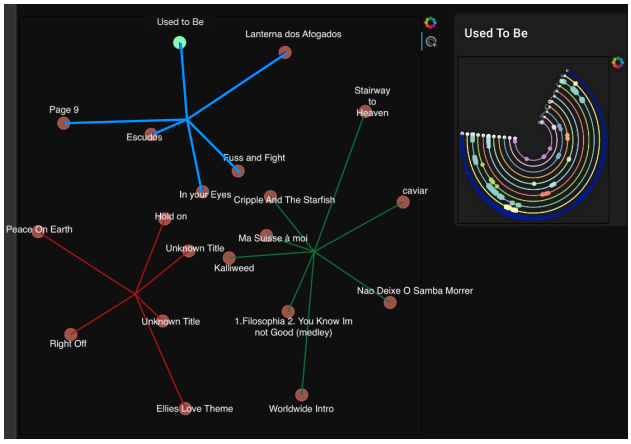

Figure 4. *Similarity-based clustering and display of multiple songs (using MDS and K-means with $K = 3$).*

Use Cases

In this section, some exemplary scenarios are described, referring to the specific visualizations introduced before.

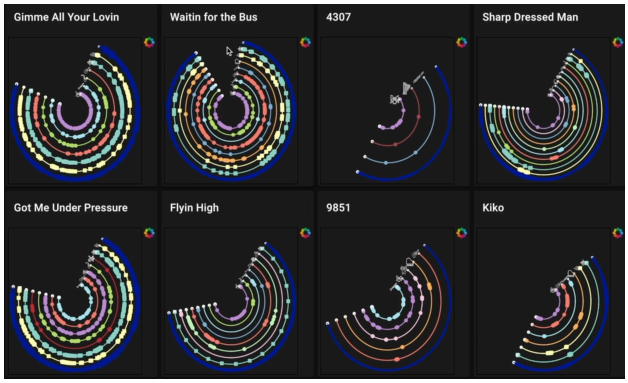

Figure 5. *Song overview page showing the overall song lengths and feature distribution of songs belonging to a specific concert.*

Variations in Length and Musical Instruments

In this scenario, the users might want to compare the overall song lengths and musical instruments played during different songs in a concert. Users can verify whether a set of musical instruments that are played varies during one song. For this purpose, it is possible to use the song overview page (see Figure 5). All songs of a concert are listed with their radial overview plots. Figure 6 shows an example of two different songs' overviews side-by-side. It is possible to compare the length of the songs and the instruments played during a concert by comparing our radial overview plots along with the color-coded legend.

Temporal Variations

Another possible use case is when a user wants to explore the temporal variation of a song such as tempo changes or solos. For this use case, it is possible to analyze the song using our timeline view as shown in Figure 7. The timeline visualization helps to see which musical instruments are playing in which parts of the song. Moreover, the tempo changes are visible via the tempogram plot.

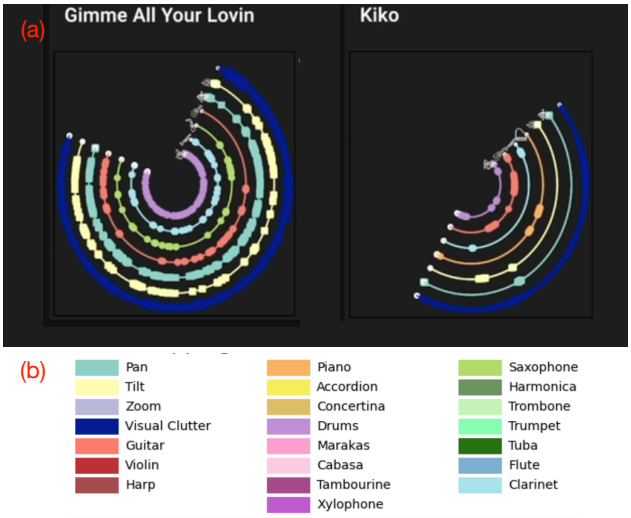

Figure 6. *Visual comparison of two songs. The radial perimeter represents the length of a song. Each line represents an audiovisual feature. (a) compares side by side the two songs "Gimme All Lovin" and "Kiko". (b) shows the color palette assigned to the instruments. The color palette is the same in all views to maintain consistency.*

The waveplot and the spectrogram help to further analyze the audio. To investigate a specific time, for example, a range when a solo is suspected to be happening, the time range slider can be used. Via the slider, a specific time range can be highlighted and visualized which helps to see more details. Finally, the visualized video frames can also be used during the investigation. In Figure 7 the user selected a time range where the guitar is clearly the dominant instrument.

Dominant Instrument and Similarity

This use case explores the effect of the dominant musical instrument on the similarity of songs. A user might want to see how similar songs sharing the same dominant musical instrument families are. Another question the users might ask would be to analyze the effect of the dominant musical instrument on the data clusters. To find the answers to these questions, the similarity-based view can be used (see Figure 4). Users can see the colors in the radial overview representing the dominant musical instrument families and compare the similarities between different songs via the distance between them in the similarity-based overview plot. For example, in Figure 8 the blue cluster shows that in almost all the selected songs the dominant instrument is "Guitar" (pale red). There is only one song with a different color in that group, where the dominant instrument is "Drums" (pale green).

Conclusions

In this paper, we presented an interactive tool for visually analyzing live performance songs and concerts based on audiovisual features. We incorporated essential features that depend on video frames and audio to consider while analyzing live performance song videos. We also proposed a similarity metric to compare different songs based on the Mahalanobis distance and provided several use cases that illustrate the usage of our approach. Future work includes incorporating se-

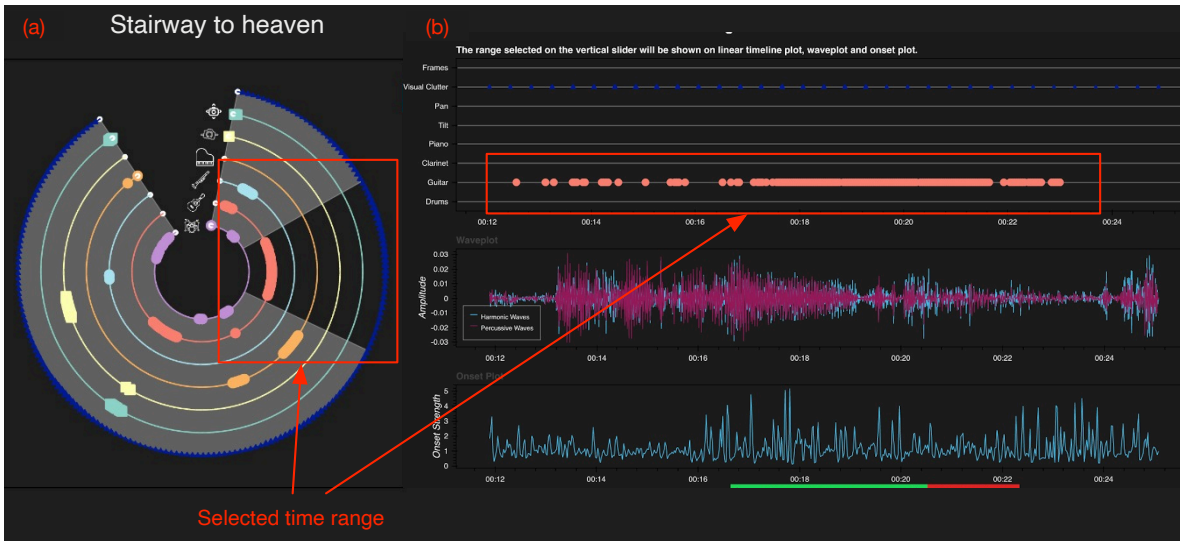

Figure 7. *Overview and detail of a song's features. (a) shows a selected time range in darker grey, showing the appearance of only the "Guitar" instrument in that section of the song. (b) shows three detailed views: the timeline exhibiting in detail the "Guitar" feature distribution over time, and below the corresponding waveplot and the tempogram plot.*

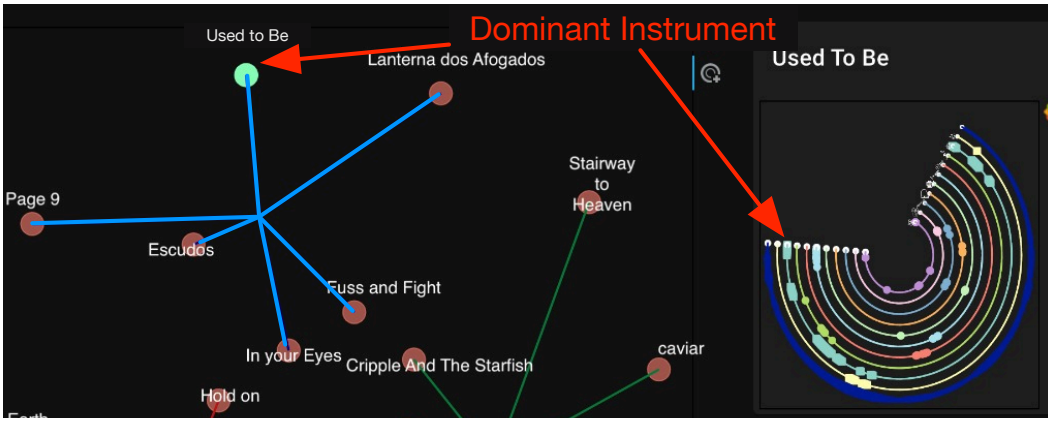

Figure 8. *Similarity-based view of song clusters. The blue cluster shows that in almost all the selected songs the dominant instrument is "Guitar" (pale red). There is only one song with a different color in that group, where the dominant instrument is "Drums" (pale green).*

mantic segmentation of the videos into scenes, solo identification, and a more extensive evaluation with domain experts and the general public. The source code is available publicly at https://github.com/alediehl/EI_2024_MusicVis

Acknowledgements

The Swiss National Science Foundation supports this research through the SINERGIA grant for the interdisciplinary project Narratives from the Long Tail: Transforming Access to Audiovisual Archives (grant number *CRSII5_198632*, see https://www.futurecinema.live/project/, for a project description). We also thank the support of Prof. Barbara Flückiger and the VIAN project team, ERC grant agreement No 670446 FilmColors.

References

[1] T. Bergstrom, K. Karahalios, and J. Hart. Isochords: visualizing structure in music. *ACM Graphics Interface*, pages 297–304, 01 2007.

[2] D. Bogdanov, N. Wack, E. Gómez, S. Gulati, P. Herrera, O. Mayor, G. Roma, J. Salamon, J. Zapata, and X. Serra. Essentia: an open-source library for sound and music analysis. In *Proceedings ACM international conference on Multimedia*, pages 855–858, 2013.

[3] C. Cho. Camera motion detection. https://github.com/chuckcho/camera-motion-detection.

[4] M. Cox and T. Cox. Multidimensional scaling. In *Handbook of Data Visualization*, Springer Handbooks Computational Statistics, pages 315–347, 2008.

[5] R. Dias and M. J. Fonseca. Muvis: An application for interactive exploration of large music collections. *ACM International Conference on Multimedia*, pages 1043–1046, 2010.

[6] A. Dufaux and T. Amsallem. The Montreux Jazz Digital Project: From preserving heritage to a platform for innovation. *Journal of Digital Media Management*, 7(4):315–329, 2019.

[7] EPFL. Montreux jazz digital project. https://go.epfl.ch/mjdp.

[8] O. Hilliges, P. Holzer, R. Klüber, and A. Butz. Audioradar: A metaphorical visualization for the navigation of large music collections. In A. Butz, B. Fisher, A. Krüger, and P. Olivier, editors, *Smart Graphics*, pages 82–92. Springer Berlin Heidelberg, 2006.

[9] Kaggle. 30 musical instruments -image classification. https://www.kaggle.com/datasets/gpiosenka/musical-instruments-image-classification.

[10] A. H. Kargaran. Visual clutter. https://github.com/kargaranamir/visual-clutter.

[11] N. Kosugi. Misual: Music visualization based on acoustic data. *International Conference on Information Integration and Web-Based Applications & Services*, pages 609–616, 2010.

[12] H. Lima, C. Santos, and B. Meiguins. Visualizing the semantics of music. *IEEE International Conference Information Visualisation*, pages 352–357, 2019.

[13] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenbergk, and O. Nieto. librosa: Audio and music signal analysis in python. *Python in Science*, 14:18–25, 2015.

[14] J. H. P. Ono, D. C. Correa, M. D. Ferreira, R. F. de Mello, and L. G. Nonato. Similarity graph: Visual exploration of song collections. *IEEE Conference on Graphics, Patterns and Images*, 2015.

[15] J. Snydal and M. Hearst. Improviz: Visual explorations of jazz improvisations. *ACM Extended Abstracts on Human Factors in Computing Systems*, pages 1805–1808, 2005.

[16] A. Soriano, F. Paulovich, L. G. Nonato, and M. C. F. Oliveira. Visualization of music collections based on structural content similarity. *IEEE Conference on Graphics, Patterns and Images*, pages 25–32, 2014.

[17] C. Walshaw. A visual exploration of melodic relationships within traditional music collections. *IEEE International Conference Information Visualization*, 22:478–483, 2018.

Author Biography

Alexandra Diehl is a senior researcher in the Multimedia and Visualization group at the Department of Informatics of the University of Zurich (UZH), Switzerland. She received her Dip. Eng. in Computer Engineering in 2005 and her Ph.D. in 2016 in Computer Science at the University of Buenos Aires, Argentina. Before, she was a postdoctoral researcher at the Data Visualization and Analysis Group (DBVIS) at the University of Konstanz, Germany. She currently works on topics at the intersection of data science, visualization, and AI.

Melike Çiloğlu is currently working as a junior consultant at a private company in Germany. She received her Bachelor of Engineering in Computer Engineering in 2020 at the Koç University, Turkey, and her MSc in 2022 in Data Science from the University of Zurich (UZH), Switzerland. She wrote her Master's thesis in the Multimedia and Visualization group at the Department of Informatics of the University of Zurich.

Renato Pajarola has been a Professor in computer science at the University of Zürich since 2005, leading the Visualization and MultiMedia Lab (VMML). He has previously been an Assistant Professor at the University of California Irvine and a Postdoc at Georgia Tech. He has received his Dipl. Inf-Ing. ETH and Dr. sc. techn. degrees in computer science from the Swiss Federal Institute of Technology (ETH) Zurich in 1994 and 1998, respectively. His research interests include 3D computer graphics, data visualization and interactive 3D multimedia.