

VISUAL OVERVIEWS FOR SHEET MUSIC STRUCTURE

Frank Heyen

Quynh Quang Ngo

Michael Sedlmair

VISUS, University of Stuttgart, Germany

{frank.heyen, quynh.ngo, michael.sedlmair}@visus.uni-stuttgart.de

ABSTRACT

We propose different methods for alternative representation and visual augmentation of sheet music that help users gain an overview of general structure, repeating patterns, and the similarity of segments. To this end, we explored mapping the overall similarity between sections or bars to colors. For these mappings, we use dimensionality reduction or clustering to assign similar segments to similar colors and vice versa. To provide a better overview, we further designed simplified music notation representations, including hierarchical and compressed encodings. These overviews allow users to display whole pieces more compactly on a single screen without clutter and to find and navigate to distant segments more quickly. Our preliminary evaluation with guitarists and tablature shows that our design supports users in tasks such as analyzing structure, finding repetitions, and determining the similarity of specific segments to others.

1. INTRODUCTION

Common music notation can be considered as a special visual encoding to convey music, including instructions on how to perform it. Despite its compactness and detailed information, a music sheet is hard to analyze for novice musicians [1]. Moreover, it contains lots of information that is hard to display at once without visual clutter or getting too small – getting an overview is tricky. When pieces contain repeating sections such as a chorus, much information is redundant. Even with abbreviations that denote repetitions, such as a double bar with colon, da capo, or al segno, a complex structure can lead to tedious navigation.

Recent work [1–3] strove to enrich notation to better convey music-theoretical information and patterns. In this paper, we instead focus on quickly gaining an overview of structures such as similarities and repetitions. This overview is meant to support learning, or teaching a music piece, as musicians often have to remember which segment of a piece they have to play when and how often, information that can be obscured in classical sheet music notation. According to the visualization principle “eyes beat mem-

ory” [4], we argue that the current notation leaves room for further optimization.

Toward this goal, we propose to visually enrich sheet music by mapping calculated similarities among segments of the sheet music, such as sections or bars (measures), to colors. To fit the notes of a whole piece onto a screen while remaining legible, we propose compact alternative encodings to common music notation that allow for overview and easier navigation without having to scroll or change pages. This work focuses on guitar tablature of Western music, which is easier to represent compactly and often features more repeating parts than other kinds of sheet music. However, we argue that our general method of mapping similarities to color can also help with other kinds of music. We conducted a preliminary qualitative evaluation through pair analytics with four guitarists. The results indicate that our design supports tasks such as summarizing structure, finding repetitions, and analyzing similarity.

In summary, we contribute 1) the exploration of novel representations of sheet music for easier overviews, specifically a method for mapping similarity among components of a music sheet to color, and 2) a preliminary pair analytics study with four guitarists. We further provide source code and a web app where users can try their own sheet music in MusicXML [5] at visvar.github.io/sheetmusic-overviews.

2. RELATED WORK

Similarity in music concerns many dimensions such as cognition, perception, tempo, pitch, and more. Therefore, existing metrics use different approaches, including a continuous representation of notes [6], a geometrical metric [7], shapes of curves [8], and a graph-based metric for harmony [9]. Janssen et al. [10] evaluated melodic similarity metrics using human annotators and a survey [11] defined eight criteria for symbolic melodic similarity. The overall aim of the above work is to query pieces in a database. In contrast, our work focuses on supporting the structural analysis of a single piece, by visualizing similarities within it. While our design allows integrating any metric, its main purpose is demonstrating how visualization can support sheet music analysis. We thus use a simpler symbolic metric to instantiate our design.

There is a broad range of music-related visualization [12, 13], including structure [14–22]. However, some visual encodings, such as one based on Tonnetz [21], re-



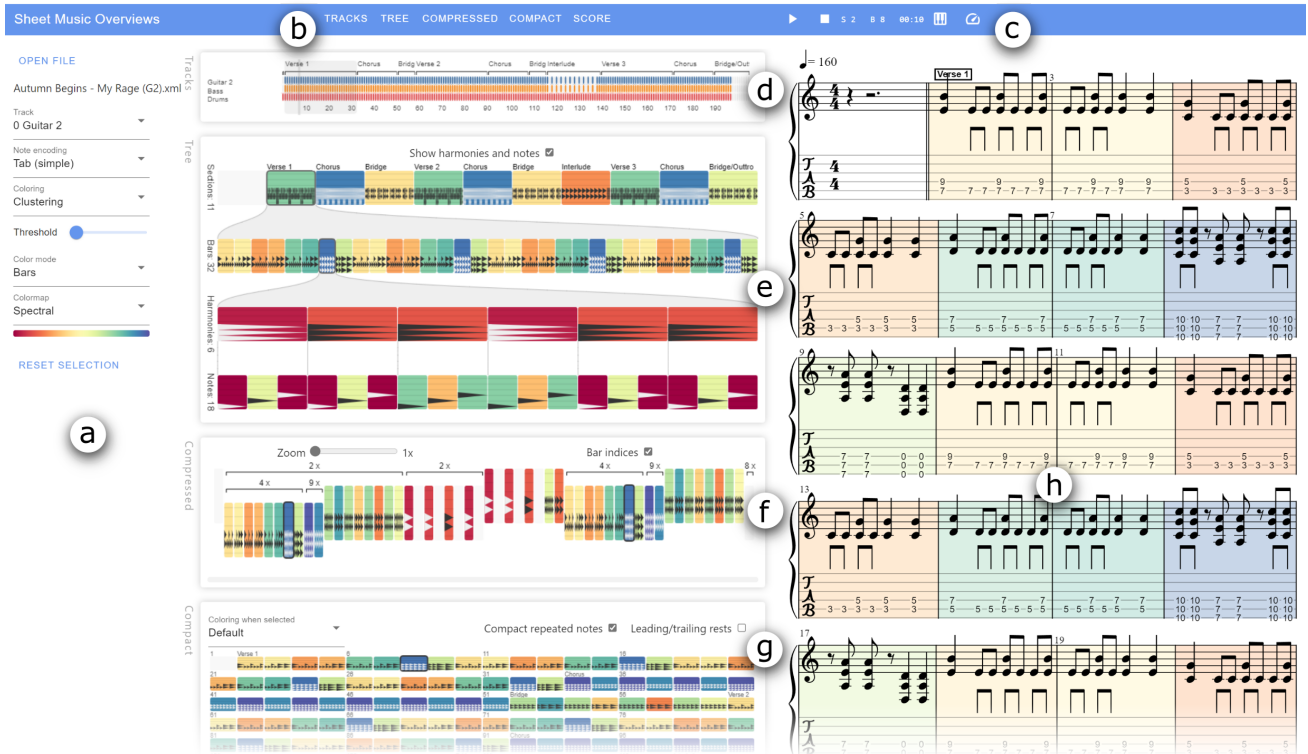


Figure 1: Screenshot of our design with all views (cut): a) data and visualization options, b) view selection, c) player, d) instrument/track overview, e) structure hierarchy, f) compressed repetitions, g) compact sheet, and h) complete score.

quire knowledge of music theory. Similar to our approach, *MoshViz* [17] focuses on visual analysis in an overview-detail fashion, but without considering perceptual or cognitive aspects. Our design allows encoding any similarity metric (mathematical or perceptual) with colors, to make sheet music easier to understand. Closest to our work is a structure visualization that uses dimensionality reduction (DR) to map audio features to color [18], which inspired us to design a similar mapping for sheet music.

Augmented sheet music adds visual components to common music notation to increase expressiveness. Related work augments a music piece with radial note histograms, to facilitate analyzing harmonic patterns [2], or visualizes rhythm through color-coded sunbursts [1]. Miller et al. [3] combined both approaches, but do not address supporting performance preparation tasks. Only little research supports instrument learning and composition. Bunks et al. [23] use color for reference keys on a tabular layout to support jazz improvisation. Others augment sheet music with lines and ellipses to support error detection in composition [24] or pianists in identifying mistakes while practicing [25, 26]. In this work, we also support learning by aiding music reading before and during practice.

3. DESIGN

We first introduce the tasks we want to support with our approach. Then, we describe how we compute similarities and map them to colors and how we represent sheet music visually (Figure 1).

3.1 User Tasks

Our overall goal is to improve the efficiency of reading sheet music, by sparing users the need to search for certain segments or memorize patterns. We want to reveal potentially interesting patterns that are hard to infer from the bare sheet music itself but could be helpful for better understanding or practicing a piece. More specifically, we want to support the following tasks: (T_1) understand the *overall structure* of a piece, (T_2) detect repetitions, which means to spot *where* something repeats *how often*, and detect *repetitions nested within* repetitions, (T_3) *compare* multiple segments regarding their similarity.

3.2 Color Mappings

Similarity metrics. Our approach works with any metric that takes notes and returns a scalar similarity score. We compare non-overlapping segments of the piece, which can be bars, pre-defined sections read from the MusicXML file, or the result of an automatic segmentation (the latter is not implemented).

In our related work section, we discussed existing similarity metrics for symbolic music. Some of these metrics do not support polyphony or require complete scores or additional annotations (such as chords) or assumptions on musical meaning. Metrics that are based on western tonal harmony [27, 28] would also not generalize to various cultures and genres. Therefore, we designed the following simple but robust algorithm: First, the notes of a segment

are sorted by their start time and those with equal start time by pitch ascending. Mapping each note to its pitch then results in one sequence of integers for each segment. We then compute a similarity matrix by calculating the *Levenshtein distance* [29] for all possible pairs of segments, equals the minimum number of pitches one would have to insert, delete, or replace, to transform the first sequence into the second.

We further compute similarities between all sets of notes that have the same start time, which we refer to as harmonies. For these sets, we only use the notes' pitch class (disregarding octaves) and compute the *Jaccard index* [30], which equals the ratio of intersection over union.

Mapping. Once we have a similarity matrix, we can create a color mapping that respects these similarities. We explored three alternative methods that use either a one-to-many comparison, dimensionality reduction, or hierarchical clustering. Figure 2 shows an example for our similarity-based color mapping, Figure 3 summarizes our different mapping strategies.

The first method colors bars by their similarity to a selected bar. This selection is made by the user or automatically when playing the piece, where the currently played bar is selected. To obtain colors, we linearly map the similarities to a color scale. Another mode only colors bars that the metric considers identical to the selected one, allowing users to quickly spot where and how often it repeats.

Our second method uses dimensionality reduction (DR), a method commonly used to transform data from a high-dimensional space to a lower-dimensional one. Usually, the target space is two- or three-dimensional, such that data points can be shown on a screen. We instead project onto a one-dimensional space that we can then linearly map to a color scale. As we do not have concrete positions in a space, but only the similarities between them, we chose *multi-dimensional scaling* (MDS) [31] that accepts a similarity matrix as input. Furthermore, MDS optimizes the computed projection to preserve these similarities, leading to a coloring optimized for these.

As an alternative to MDS, we designed a method that clusters similar segments together and then assigns each cluster one color such that similar clusters have similar colors. Using our similarity matrix, we compute *hierarchical agglomerative clustering*, which gives us a binary tree. We then sort the tree's leaves from left to right, as leaves that are closer together are more similar, and map them in this order to a color scale. Compared to the above method using MDS, the resulting colors are easier to distinguish but represent similarities less accurately. Using a similarity threshold, users can steer the number of clusters and therefore colors, to choose a trade-off between detail and overview.

Color scales. Research on perception proposed a range of color scales specifically designed for visualization. Since there are different irreconcilable goals, no scale is appropriate for all tasks. While multi-hue scales such as rainbows have been criticized [32], they have been shown to work well for some circumstances [33–35]. For users

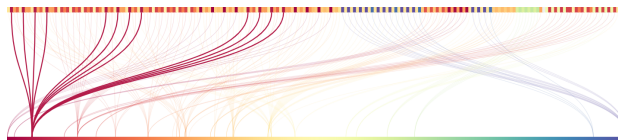


Figure 2: Example of similarity-based color mappings. *Top:* rectangles represent all bars of a piece, from left to right in the order they occur. *Bottom:* a color scale. Curves connect each bar to its color. In this figure, the curves of a single color are highlighted through a stronger opacity to show how they connect to identical bars.

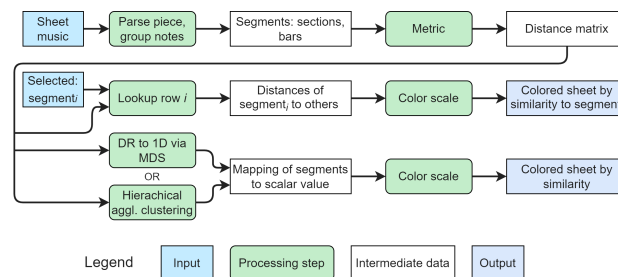


Figure 3: We compute similarity-based colors for extracted segments via direct comparison, dimensionality reduction, or clustering. A *segment* can be any sequence of notes in the piece, such as a bar or a pre-defined section. The *selected segment* is chosen by the user to compare it to all others.

with a color vision deficiency, scales with fewer hues and, therefore, less discernible colors can be used, such as *cividis* [36]. When color is used to compare different values or intervals, a color scale needs to accurately represent similarities between values. For this task, single-hue scales or interpolations between two hues are appropriate but further reduce the number of discernible colors. Although the number of distinguishable colors is small, there are enough for our use case, as the number of different segments in a piece is limited. Since colors are distributed by similarity, indistinguishable colors should only be assigned to very similar segments. To accommodate different user needs, we choose a broad multi-hue scale (spectral) as default but also provide more accessible ones; for direct comparison, we choose a single hue scale (blues) (Figure 4).

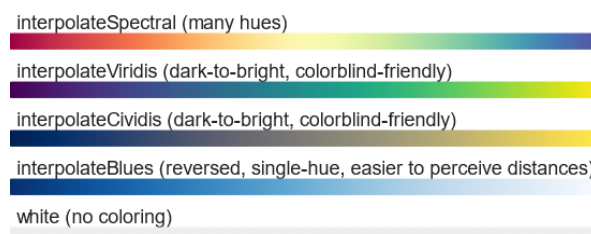


Figure 4: Some of the included color scales, taken from D^3 [37]. The choice depends on the current task and individual limitations of the user's color vision.

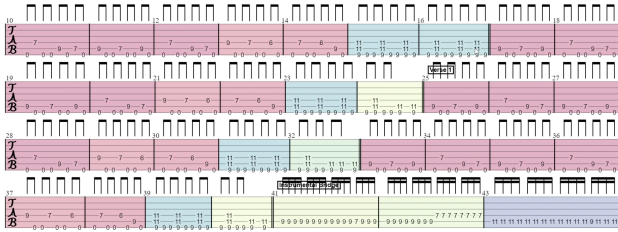


Figure 5: A pattern of repetitions with different endings. Bar 16 is colored blue, 24 and 40 are the same yellow, and bar 32 is green.

3.3 Visual Encodings

Layout enrichment for common music notation. We designed several visual encodings to address different tasks and reveal different kinds of patterns. The most straightforward encoding is to display the full sheet music as the common notation that is familiar to musicians and represents the complete information. We enrich this display by adding colored, semi-transparent rectangles on top of the segments, for example, one for each bar (Figure 5). The reduced opacity makes colors brighter than in other views but improves the readability of the notation. This coloring helps more quickly see where a bar repeats (T_2), as the user only has to compare bars with similar colors (T_3). Even when two different bars were assigned similar colors, this process allows for ruling out many others. This encoding suffers from the same limitations as non-colored sheet music. Due to its highly detailed nature, fitting the complete piece on the screen at once would lead to small and illegible visuals. Therefore, we designed simplified, filtered, and compressed alternatives, which we explain in the following paragraphs.

Note display. In most views, we represent notes by blocks that are positioned horizontally by start time and have a width proportional to the note’s duration, to visually indicate timing and rhythm (Figure 6). The first mode displays the notes as triangles in a piano roll, allowing it to represent music for any instrument, but less readable than other encodings. A second mode displays guitar tablature, where each row stands for one string, and the third adds fret numbers for more detail. We focus primarily on guitar tablature in this work, but new encodings resembling other instruments’ common notations could extend our approach. Depending on their size, our encodings become hard to read but still reveal coarse patterns more clearly than detailed notation. In order to show the whole piece at once (T_1), without filtering or compression, we display the full score with the above encodings (Figure 1g).

Hierarchy. Most music pieces have a hierarchical structure in the form of sections such as verse and chorus, spanning multiple bars, each with none to a few notes, which might be grouped in harmonies (notes played at the same time). We visualize this structure as a tree, where users can select a node to show only its children in the level below (Figure 1e). This representation supports gaining an

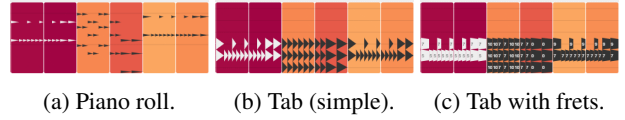


Figure 6: Note display: a) Piano rolls can represent any music but lack additional information. b, c) Tablature either simplified or with fret numbers. Notes are drawn in black or white depending on the background’s luminosity.

overview (T_1) and allows navigating the sheet music more conveniently. The colors are level-specific, such that they only represent similarities within, not between, the levels. Notes have their own color map that is not based on similarity but still allows to spot repetitions or patterns such as sequences of increasing pitch (T_2).

Compressed multi-level repetitions. Music pieces might have another hierarchical structure regarding repetitions when a repeating segment contains repeating sub-segments ($T_{1,2}$). Similar to data compression, this allows us to create a more compact representation, by only displaying a repeating segment once and annotating it with its number of repetitions. Doing this recursively results in a tree where each leaf is a bar of the music piece, and each inner node contains the following information: A pre-fix child, a repeated child with its repetition count, and a post-fix child, where pre- and post-fix can be empty. The visual encoding we chose for this data structure uses our compact note encoding (Figure 6) for the leaves and brackets for the inner nodes (Figure 7). Numbers above the bars denote the index of their first occurrence, allowing to spot recurring ones that are farther apart.

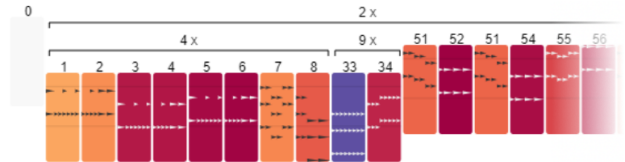


Figure 7: Our compressed view shows repetitions as nested blocks (cut). Note, that bar 51 appears two times, as it equals bar 53.

Workflow and interaction. We envision musicians using our interface primarily while learning a new piece, where they first get an overview and then take a closer look at the detailed sheet music. During navigation, reading, and playing, they could use overviews as ‘minimaps’ that provide context for what they are currently focusing on. As all views are linked, clicking on a bar in any view allows users to highlight or jump to a certain bar in all other views.

4. EVALUATION

We chose a pair analytics setup over a comparative user study, as most related work focuses on different tasks or data, which does not allow fair comparison. Furthermore,

instead of quantitative usability ratings and time measurements, we were more interested in qualitative feedback on *how* guitarists would use our design and *what limitations* they encounter. In pair analytics [38], designers and participants collaboratively analyze the participants’ data, allowing designers to evaluate a design without needing to teach participants how to use it, saving them time and ensuring they use all features appropriately.

Our participants (P1–P4) have experience with reading guitar tablature. P1 has played guitar for 15 years and regularly teaches it and P2 has played drums for 5 years and guitar occasionally. P3 and P4 have played guitar for 16 and 12 years. P1 and P3 have full color vision, P2 and P4 have a slight red-green deficiency. All but P3 were familiar with visualization. We met with each participant for roughly 1.5 hours. After an introduction to our interface, we looked at guitar tablature of their choice together, encouraging them to use different features and think aloud.

Our participants found the coloring generally helpful: “I have played classical pieces with 8 or 12 pages ... you searched, with your teacher, made annotations with a pen ... ‘here it’s that part again’ ... if it’s only black-on-white, you’re blind at some point” (P3). They were able to detect various patterns: “the color indicates a new segment” (P1) (T₁), “always two bars one note, two bars the other note, ...” (P1) (Figure 8). One interesting example was a pattern where the same segment was repeated four times, with a different ending each time, except for the fourth that equals the second (Figure 5, T_{2,3}).



Figure 8: A simple alternating pattern with bars that repeat as *AABB* multiple times.

In some cases, our current similarity metric did not work well: “here, I’m sure it’s different, but the color is not quite different ... if you check the color carefully, I think you can see it” (P1). We proposed coloring annotated sections of the sheet music by their similarity: “That is already very useful, because ... when I’m [teaching] and want to show segments, then I always have to mark them by hand. This is doing it for me” (P1). While we currently colorize either by section or bar, P3 suggested alternatively coloring by sequences of bars that occur together multiple times, such as riffs or motifs. Interestingly, our coloring allows users to quickly guess the effort needed to learn a piece: “The colors show what you already learned” (P3), “For me it looks like I practice this purple bar ... and then I practice these yellow and green bars and then I can play 90 percent of the song” (P1) (T_{1,2}). P4 suggested ignoring bars with only a single note or chord when coloring, as these are less interesting. Instead, they proposed coloring differently transposed versions of the same pattern more similarly and allowing users to manually adjust the color of a set of identical bars.

We also compared coloring via DR versus via cluster-

ing. When trying clustering, P3 first pondered “I think [coloring via clustering] is easier to understand ... as you can really see that it’s different” but then concluded that “it’s difficult, both have pros and cons ... now if I would search [by color], it would be more difficult to spot” (T_{2,3}). P4 suggested to additionally vary the color’s brightness for different segments in the same cluster, to also reveal similarity within clusters (T₃).

Most patterns were visible with all color scales, although less clearly with those using fewer hues, so users with color vision deficiencies can also benefit from our design. Even though P4 has a slight red-green deficiency, they wanted to use the default *spectral* scale for most of the study. When trying out other scales, they told us that it indeed makes a good default, as it has fewer hues than rainbow scales. P4 further preferred *viridis* over *cividis*: “here I see better”. When turning of colors, P3 was astonished: “here you see how white it is! When looking at colors for so long, you see for the first time how ugly white it is”.

The simplified and full tablature encodings still reveal characteristics: “These are power chords, right?” (P2), “This is a power chord on the second fret ... and that should be A minor” (P3). For our hierarchy view, P4 found that “the tab encoding doesn’t help much, the simplified tab and piano roll work much better”. Especially, since with the piano roll “you can see well which bars have similar notes” (P4) (T₃).

In our hierarchy view, clicking on different sections allows users to quickly compare them: “Main riff and verse is almost the same, it’s labeled as ‘main riff’ because it is ... without singing” (P1) (T₃). This also shows a drawback of sheet music, where repetition signs often apply for all instruments at once, so if one repeats and another does not, the first instrument’s notes will show up redundant. During the comparison, we found that P1 labeled the sections incorrectly, as one had a few more bars that actually belonged to the following section: “we found that we labeled it wrong, that’s good!” (P1) (T₃). Switching between different sections allows comparing them: “here’s again a verse, but a little different ... back there, this bar is repeated ... this chorus is much longer” (P3) (T_{2,3}). During our study with P4, we saw that a whole bar consisted of almost only the note E, as indicated by identical color, except for a single note with different color – “I also wouldn’t have noticed that [without color]” (P4). As we only support highlighting sections and bars for now, P3 missed being able to click on single harmonies and notes to highlight them in other views.

The compressed encoding (Figure 7) was P1’s favorite: “This is the feature I’m most excited about for showing people the song structure because this is something I just can’t do with a [music notation software]” (T₁). For playing a song with students where each plays a different instrument of the piece, it “would be great if everyone would have something like this” to have a compact summary of the part they should play. Our participants proposed features we could add to this view, such as reducing or disabling nested repetitions: P1 found “it would

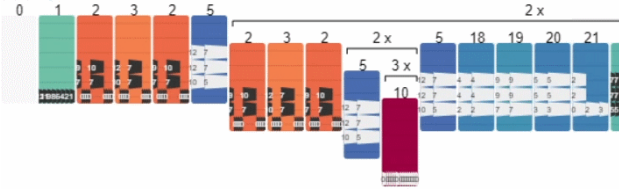


Figure 9: Our algorithm might not create the same structure as a musician: The sequence 2, 3, 2, 5 appears twice in a row, but since the following bars of the second appearance form a longer repetition, the sequence was included there instead.

be great if you had the option to simplify it” and P2 suggested adding a slider for the compression level. They also told us that this view could be extended to support annotation: *“That would be great, to be able to go here and annotate some things”* (P1). Our compression sometimes leads to unexpected results as it tries to find the longest repetitions, which might not be how musicians would compress a piece: *“I would expect that this is in here two times, but somehow it’s here – it makes sense, it’s just two different ways of describing it”* (P1) (Figure 9). P3 and P4 did not consider the compressed view useful (*“I would not use compressed much, maybe once when first looking at a new song”* (P3)) and P4 proposed merging it with the compact view by optionally drawing repeated segments only once with the brackets used in this view.

As our compact sheet music view shows the whole piece at once, it allows users to spot global patterns, such as how often a bar appears ($T_{1,2}$): *“The compact view shows how things repeat”* (P4), *“If you go here and go for [the color mode] ‘Identical’, you can see that there is a lot of this”* (P1). P4 wished for an alignment, to have repeating patterns exactly below each other: *“it could auto-align it for me”*. We then asked what they think about manually inserting line breaks, where to P4 responded: *“this would be the most important to me – if this should help me learn or read or play, I have to be able to customize and save it”*.

When asked for general feedback, P1 told us they *“like it a lot, because it’s always hard to see what is similar to something else ... I think it’s very important that you [know] not just what is played, but also if there is a connection to other segments”* ($T_{1,2,3}$). P1 wished to be able to directly compare bars or sections (T_3): *“That would be great if you could select two and then see the difference because I always click [back and forth]”*.

When asked for use cases, our participants told us they would use our interface for learning, for example by playing along. Both P3 and P4 further imagined using our compact view as a cheat sheet during a performance: *“[For songs with chords and simpler rhythm] you could print this and give it to someone ... and they could play the song ... or you use a tablet”* (P3). P4 hid all views but tracks and compact, thereby maximizing the latter: *“like this, I see the whole song at once ... convenient as a memory aid. Assuming I know the song already ... I see how often I have to*

play everything” ($T_{1,2}$). As another use case, P4 wished to be able to see multiple instruments at once to be able to compare them visually.

5. LIMITATIONS AND DISCUSSION

We mainly focus on guitar tablature, which is easier to represent compactly and often features more repeating segments than other kinds of sheet music. However, we argue that our general method of mapping similarities to color can also help with other kinds of music. With new, specialized similarity metrics and note encodings, our approach could support non-western kinds and even music without discrete notes, as long as a piece can be segmented. Our example design for guitar tablature and with a simple similarity metric allowed us to stay within a reasonable scope and matched our own musical expertise.

Human color vision is limited, even more with color vision deficiencies. Our approach can add value compared to non-colored sheet music for everyone, although accessible color scales reveal less detail. In our study, some patterns were clearly visible while some were harder to spot – still, they were easier to spot than without any color. Coloring by similarity works well for pieces with a few different segments that are repeated, as fewer colors are necessary, but will not work as well for others.

As our current approach depends on dimensionality reduction and clustering, it inherits the limitation of these techniques, such as distortion and artifacts. We chose MDS and hierarchical agglomerative clustering to preserve similarities as well as possible, but other algorithms or approaches might further reduce these limitations.

In our current interface, the participants missed being able to directly compare two selections of bars, align bars automatically or through line breaks, and assign custom colors and labels. Our evaluation only included four participants. While such a number is typical for pair analytics, real-world acceptance can only be evaluated through longitudinal field studies, where a larger number of users regularly use a product in their daily life.

6. CONCLUSION

We designed multiple methods to ease the detection of repeating structures in sheet music. Our evaluation provided a first qualitative indication of the effectiveness of our approach. Therefore, we are confident that extensions to our design can turn our work into a helpful tool for musicians.

Future work includes further similarity metrics and visual encodings better suited for different tasks, sheet music characteristics, instruments, and music genres. Adding labels and exporting them would allow musicians and teachers to save and share their results. Showing multiple instruments of a piece at once would allow comparing them, for example, to quickly see where two guitars play similar notes. We plan to let more musicians actively use our design during learning, playing, and teaching over months to test real-world usage and acceptance longitudinally.

7. ACKNOWLEDGMENTS

This work was funded by the Cyber Valley Research Fund and by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Project-ID 251654672 – SFB TRR 161, project A08.

8. REFERENCES

- [1] D. Fürst, M. Miller, D. A. Keim *et al.*, “Augmenting sheet music with rhythmic fingerprints,” in *5th Workshop Visualization for the Digital Humanities (VIS4DH)*. IEEE, 2020, pp. 14–23.
- [2] M. Miller, A. Bonnici, and M. El-Assady, “Augmenting music sheets with harmonic fingerprints,” in *Proc. ACM Symp. Document Engineering (DocEng)*. ACM, 2019.
- [3] M. Miller, D. Fürst, H. Hauptmann *et al.*, “Augmenting digital sheet music through visual analytics,” *Computer Graphics Forum (CGF)*, vol. 41, no. 1, pp. 301–316, 2022.
- [4] T. Munzner, *Visualization analysis and design*. CRC press, 2014.
- [5] M. Good *et al.*, “MusicXML: An internet-friendly format for sheet music,” in *XML Conf. and expo*. Cite-seer, 2001, pp. 03–04.
- [6] R. Valle and A. Freed, “Symbolic music similarity using neuronal periodicity and dynamic programming,” in *Mathematics and Computation in Music (MCM)*, T. Collins, D. Meredith, and A. Volk, Eds. Cham: Springer, 2015, pp. 199–204.
- [7] W. B. de Haas, F. Wiering, and R. C. Veltkamp, “A geometrical distance measure for determining the similarity of musical harmony,” *Int. Journal of Multimedia Information Retrieval (IJMIR)*, vol. 2, no. 3, pp. 189–202, 2013.
- [8] J. Urbano, J. Lloréns, J. Morato *et al.*, “Using the Shape of Music to Compute the Similarity between Symbolic Musical Pieces,” in *Int. Symposium on Computer Music Modeling and Retrieval (CMMR)*, 2010, pp. 385–396.
- [9] F. Simonetta, F. Carnovalini, N. Orio *et al.*, “Symbolic music similarity through a graph-based representation,” in *Proc. of the Audio Mostly 2018 on Sound in Immersion and Emotion*. ACM, 2018.
- [10] B. Janssen, P. van Kranenburg, and A. Volk, “A comparison of symbolic similarity measures for finding occurrences of melodic segments,” in *16th Int. Society for Music Information Retrieval Conf. (ISMIR)*, 2015.
- [11] V. Velardo, M. Vallati, and S. Jan, “Symbolic melodic similarity: State of the art and future challenges,” *Computer Music Journal (CMJ)*, vol. 40, no. 2, pp. 70–83, 2016.
- [12] R. Khulusi, J. Kusnick, C. Meinecke *et al.*, “A survey on visualizations for musical data,” *Computer Graphics Forum (CGF)*, 2020.
- [13] H. B. Lima, C. G. R. D. Santos, and B. S. Meiguins, “A survey of music visualization techniques,” *ACM Comput. Surv.*, vol. 54, no. 7, 2021.
- [14] F. Watanabe, R. Hiraga, and I. Fujishiro, “Brass: Visualizing scores for assisting music learning,” in *Proc. International Computer Music Conference (ICMC)*, 2003.
- [15] M. Wattenberg, “Arc Diagrams: Visualizing structure in strings,” in *IEEE Symp. Information Visualization (INFOVIS)*. IEEE, 2002, pp. 110–116.
- [16] J. Li, “Music analysis through visualization,” in *Proc. of the Int. Conf. on Technologies for Music Notation and Representation*, 2016, pp. 220–225.
- [17] G. D. Cantareira, L. G. Nonato, and F. V. Paulovich, “MoshViz: A detail+overview approach to visualize music elements,” *IEEE Trans. on Multimedia*, vol. 18, no. 11, pp. 2238–2246, 2016.
- [18] J. Savelsberg, “Visualizing music structure using Spotify data,” in *Extended Abstracts for the Late-Breaking Demo Session of the 22nd Int. Society for Music Information Retrieval Conf. (ISMIR EA)*, 2021.
- [19] M. Müller and N. Jiang, “A scape plot representation for visualizing repetitive structures of music recordings,” in *13th Int. Society for Music Information Retrieval Conf. (ISMIR)*, 2012.
- [20] A. Hayashi, T. Itoh, and M. Matsubara, “Colorscore – visualization and condensation of structure of classical music,” in *2011 15th Int. Conf. on Information Visualisation (IV)*, 2011, pp. 420–425.
- [21] T. Bergstrom, K. Karahalios, and J. C. Hart, “Isochords: Visualizing structure in music,” in *Proc. Graphics Interface*. ACM, 2007, p. 297–304.
- [22] J. Snyder and M. Hearst, “ImproViz: Visual explorations of Jazz improvisations,” in *Extended Abstracts on Human Factors in Computing Systems (CHI EA)*, ser. CHI EA ’05. ACM, 2005, p. 1805–1808.
- [23] C. Bunks, T. Weyde, A. Slingsby, and J. Wood, “Visualization of tonal harmony for jazz lead sheets,” in *EuroVis 2022 - Short Papers*. The Eurographics Association, 2022, pp. 109–113. [Online]. Available: <https://doi.org/10.2312/evs20221102>
- [24] R. D. Prisco, D. Malandrino, D. Pirozzi *et al.*, “Understanding the structure of musical compositions: Is visualization an effective approach?” *Information Visualization*, vol. 16, no. 2, pp. 139–152, 2017.

- [25] S. Asahi, S. Tamura, Y. Sugiyama *et al.*, “Toward a high performance piano practice support system for beginners,” in *Asia-Pacific Signal and Information Processing Association Annual Summit and Conf. (AP-SIPA ASC)*, 2018, pp. 73–79.
- [26] M. Hori, C. M. Wilk, and S. Sagayama, “Piano practice evaluation and visualization by HMM for arbitrary jumps and mistakes,” in *2019 53rd Annual Conf. Information Sciences and Systems (CISS)*, 2019, pp. 1–5.
- [27] W. B. De Haas, M. Rohrmeier, R. C. Veltkamp *et al.*, “Modeling harmonic similarity using a generative grammar of tonal harmony,” in *Proc. of the Tenth Int. Conf. on Music Information Retrieval (ISMIR)*, 2009.
- [28] W. B. De Haas, J. Rodrigues Magalhães, R. C. Veltkamp *et al.*, “HarmTrace: Improving harmonic similarity estimation using functional harmony analysis,” in *Proc. of the 12th Int. Conf. on Music Information Retrieval (ISMIR)*, 2011.
- [29] V. I. Levenshtein, “Binary codes capable of correcting deletions, insertions, and reversals,” in *Soviet Physics Doklady*, vol. 10, no. 8. Soviet Union, 1966, pp. 707–710.
- [30] P. Jaccard, “The distribution of the flora in the alpine zone,” *New Phytologist*, vol. 11, no. 2, pp. 37–50, 1912.
- [31] J. B. Kruskal, “Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis,” *Psychometrika*, vol. 29, no. 1, pp. 1–27, 1964.
- [32] D. Borland and R. M. Taylor II, “Rainbow color map (still) considered harmful,” *IEEE Computer Graphics and Applications (CG&A)*, vol. 27, no. 2, pp. 14–17, 2007.
- [33] K. Reda and D. A. Szafir, “Rainbows revisited: Modeling effective colormap design for graphical inference,” *IEEE Trans. Visualization and Computer Graphics (TVCG)*, vol. 27, no. 2, pp. 1032–1042, 2021.
- [34] K. Reda, A. A. Salvi, J. Gray *et al.*, “Color nameability predicts inference accuracy in spatial visualizations,” *Computer Graphics Forum (CGF)*, vol. 40, no. 3, pp. 49–60, 2021.
- [35] M. Wattenberg, F. B. Viégas, and K. Hollenbach, “Visualizing activity on Wikipedia with chromograms,” in *Human-Computer Interaction – INTERACT 2007*. Springer, 2007, pp. 272–287.
- [36] J. R. Nuñez, C. R. Anderton, and R. S. Renslow, “Optimizing colormaps with consideration for color vision deficiency to enable accurate interpretation of scientific data,” *PLOS ONE*, vol. 13, no. 7, pp. 1–14, 2018.
- [37] M. Bostock, V. Ogievetsky, and J. Heer, “D³: Data-driven documents,” *IEEE Trans. on Visualization and Computer Graphics (TVCG)*, vol. 17, no. 12, pp. 2301–2309, 2011.
- [38] R. Arias-Hernandez, L. T. Kaastra, T. M. Green *et al.*, “Pair analytics: Capturing reasoning processes in collaborative visual analytics,” in *44th Hawaii Int. Conf. System Sciences (HICSS)*, 2011, pp. 1–10.