SHADES OF MUSIC: LETTING USERS DISCOVER SUB-SONG SIMILARITIES

Dominikus Baur, Tim Langer, Andreas Butz
Media Informatics Group
University of Munich (LMU), Munich, Germany
{dominikus.baur,andreas.butz}@ifi.lmu.de, tim.langer@campus.lmu.de

ABSTRACT

Many interesting pieces of music violate established structures or rules of their genre on purpose. These songs can be very atypical in their interior structure and their different parts might actually allude to entirely different other songs or genres. We present a query-by-example-based user interface that shows songs related to the one currently playing. This relation is not based on overall similarity, but on the similarity between the part currently playing and parts of other songs in the collection along different dimensions (pitch, timbre, bars, beats, loudness). The similarity is initially computed automatically, but can be corrected by the user. Once a sufficient number of corrections has been made, we expect the similarity measure to reach an even higher precision. Our system thereby allows users to discover hidden similarities on the level of song sections instead of whole songs.

1. INTRODUCTION

All music is based on repetition on different levels: From the lowest level of sounds in different frequencies to the highest, cultural aspects of genres and trends, every song is contained in an intricate network of repeating segments. One of the best known of these patterns is the verse-chorus form [1] that has been defining for the last half century of popular music and implies inherent repetitive structures, possibly to increase recognition. Nevertheless, certain parts such as the intro, outro or especially the bridge can stand in complete contrast to the rest of the song, sometimes forming a mini-song of their own (and sometimes even digressing along this path and never returning to their origin).

Music recommendation and visualization often relies on an abstract idea of ”similarity” between songs, which is actually a measure for repetition. It is mostly generated by collaborative filtering or content-based measures, but this similarity normally works on the level of whole songs, with a set of related songs based on their averaged closeness. While some systems access songs on a lower level to extract segments, they do so to find the most representative part of the song to, again, do an overall comparison.

In this way, parts of songs with a high inner diversity (as in the bridge parts mentioned above) simply disappear in the similarity measure: While the overall impression of song A might be very similar to song B regarding content and sound, its bridge might be an allusion to a third song C and its outro even closer to another song D, neither of which is reflected in a generalized, one-dimensional similarity value. Query-by-example/humming systems, in contrast, have to rely on these deeper structures within a song, as they mostly have to work with incomplete input. Still, their main use is not to reveal hidden connections between parts of songs but to retrieve the one song that the user has in mind - multiple songs are only displayed because of inaccuracies in retrieval.

In this paper we present our web-based system *Shades of Music* that provides users with an interface to retrieve and discover connections between songs at the level of parts or sections. The user can listen to songs and see which other songs are similar to the currently playing section and in which of their parts. To stay with the example from above: For most of song A, sections from song B are shown as the most similar ones, but during A’s bridge song C and during A’s outro, song D appear. A similarity between these sections is initially calculated using the web service Echo Nest[2], but our system then encourages users to give feedback and improve its classification. In the rest of this paper we present related work in the areas of music user interfaces, then describe our system, the way users can give feedback, and the underlying calculations.

2. RELATED WORK

Query-by-example is an active field of research that aims for retrieving an item with only insufficient information. As the input mostly represents a part of the full item, extracting segments and being able to compare them is an important first step. Older QBE systems for audio mostly worked with symbolic MIDI-files[3], but more recent systems evaluate the actual audio signals. Various attributes, such as note sequences[3], melody[4] (e.g., with Query-by-humming), or beat[5] are used. Since these systems always try to retrieve one specific item, the segmentation is used to create a ranked list of possible candidates. More creative approaches to QBE such as [6] are trying to let

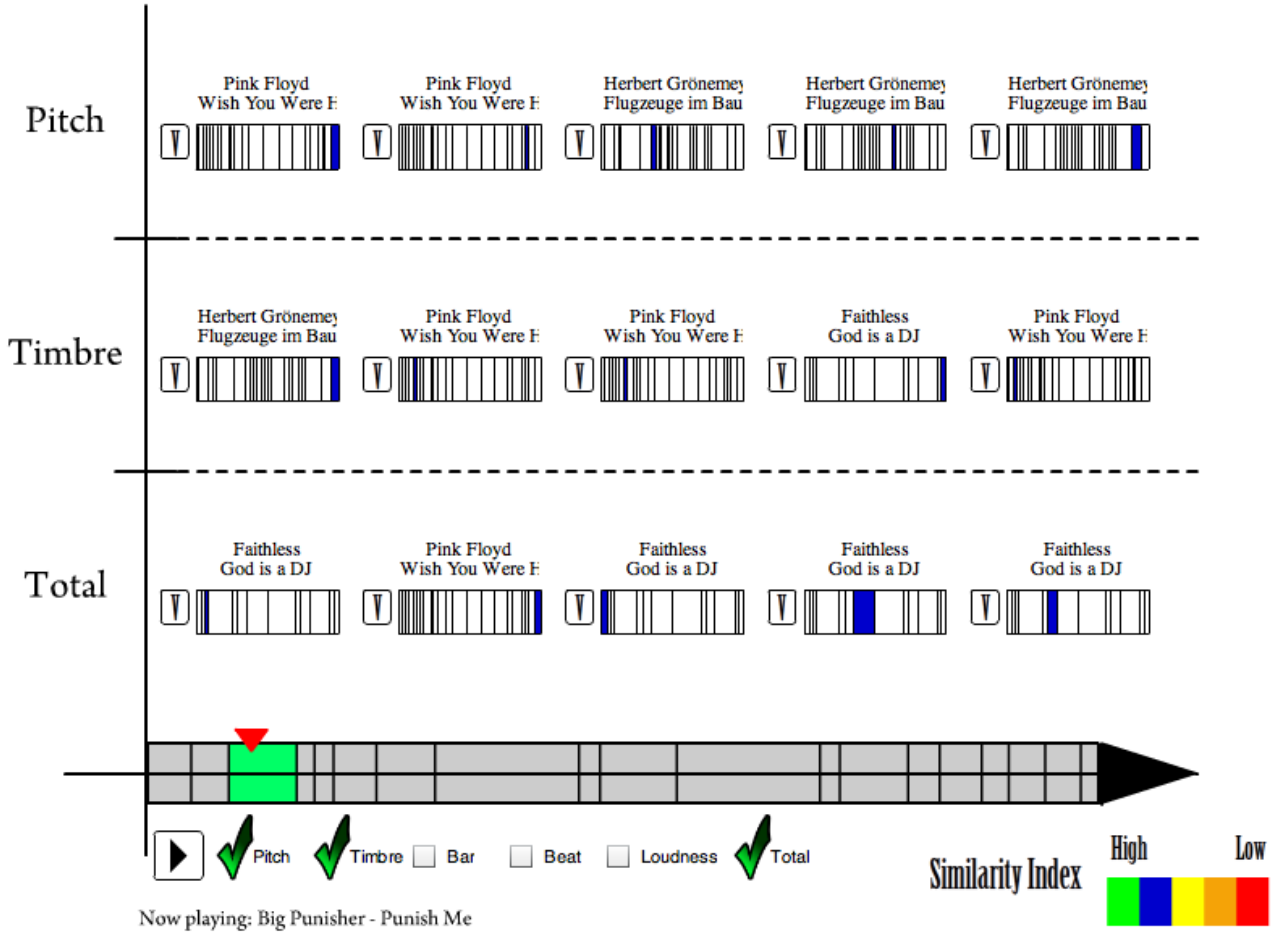

Figure 1. Shades of Music: Listening to a song and finding related songs based on different attributes

the user ”sketch” aspects of a song in various ways as an input to the system. Applications for representing larger music collections follow two courses: One approach is to visualize the collection in a global way, for example using the popular self-organizing maps (Islands of Music [7], but also [8] and [9]) or force-directed layouts[10]. Another way is to display related items based on one currently active item (in principle QBE) as in Musicream[11] or the Expressive Music Jukebox[12].

To make up for the shortcomings in automatic content-based similarity analysis and allow for personalization, user feedback is incorporated in various systems. Recommender systems[13], for example based on ratings [14] or implicit data such as listening histories in the online community Last.fm[15] offer the user suggestions for novel music. Connections between song parts are central, for example, for the music website Who Sampled? [16] whose community adds samples and their origin to the database.

3. SHADES OF MUSIC

Shades of Music is a (prototype of a) web-based service that allows users to listen to songs and find related sections. Based on the currently playing song, related sections of other songs are displayed. Echo Nest does an initial similarity classification, but as the interface collects user feedback this similarity measure becomes more accurate. We implemented Shades of Music with the Ruby on Rails framework on the server side and a browser application based on Adobe Flash on the client side.

3.1 The User Interface

Initially, a list of all songs in her or his collection is displayed. Additional songs can easily be uploaded from the computer or retrieved from online sources. After choosing a song, the application starts to play this song and displays the main interface (see figure 1). A horizontal bar at the bottom represents the current song and its sections. A play head and additional color highlighting show the currently playing section of the song. With the check boxes below the bar, the user can choose the criteria based on which related sections are displayed. Pitch, timbre, bar, beat and loudness plus a cumulative total value are available. For each selected attribute, an additional line of songs is displayed (”Pitch”, ”Timbre” and ”Total” in figure 1). Their order reflects the similarity: The most similar song section appears on the left followed by less similar ones to the right. For each of these sections, the complete song is displayed including artist and title. Each of these songs is

again divided into its subsections with unrelated sections transparent and related sections with a five-step color coding that shows similarity for the current attribute from low to high. Once familiar with this visualization, the user can see at first glance that, for example, the intro of another song is similar to the active section. It is important to note here that the same song might appear not only once but several times: Once among each of the different attributes but also within the list for one attribute if more than one section of the song corresponds to the current section (see "Faithless - God is a DJ" in figure 2). The lists contain only the five most similar songs along that particular dimension. Since the sections of a song are often very similar to other sections of the same song, related sections from the current song are not displayed.

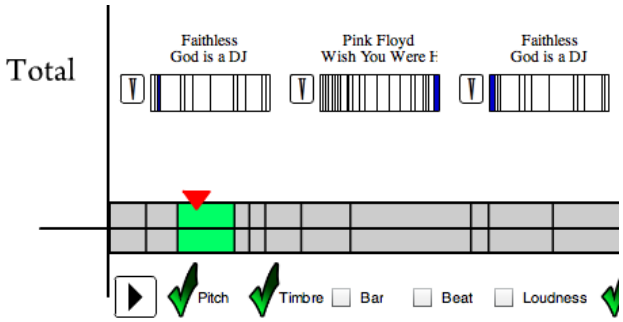

Figure 2. Detail of figure 1: The same song might be represented by several sections

Shades of Music can be used as a web-based radio: If one song is over, the system automatically picks the overall most similar song and starts playing that (which makes the first chosen song the seed song of the playlist[17]). With our similarity metric (see below) being symmetrical, this would lead to two similar songs playing in an endless loop (as the one most similar to the first would in turn have the first song as its top candidate). Therefore, the system only plays each song once. The user can of course also use the system to actively navigate her or his collection: Upon double-clicking one of the suggested songs, the system starts playing it.

3.2 Segmentation

Separating music into relevant subsections is a topic of active research. Methods learned from extracting representative audio thumbnails [1] can also be used to analyse the structure of an audio source [18]. Echo Nest is a web service that provides among others such an analysis for audio data. Besides retrieving meta-data for songs and values such as their current popularity (based on mentioning on webpages), it also performs segmentation and analysis of songs. Details can be found in [19].

One useful feature in our case is the automatic division into longer *sections* of several seconds length (e.g., verse or chorus) and very short *segments* that form short stable elements of a song. For each of these segments, Echo Nest returns a value for variations in loudness plus a chroma vector for pitch and another twelve-dimensional vector for timbre. The pitch chroma vector reflects the relative distribution of the acoustic content along the twelve semitones, while the timbre vector tries to capture the spectral surface of sound in an Echo Nest specific format with weights for twelve basis functions [2].

Additionally, positions of beats and bars for the whole song can also be retrieved. To calculate the similarity between sections of songs, we use the following procedure: First, the positions of segments and sections are retrieved. The longer sections form the basis for the comparison and are displayed in the interface as separate areas. Two sections' beat- or bar-wise similarity is determined by counting the number of beats and bars within a section and comparing these numbers.
For all segments within one section, values for changes in loudness, pitch and timbre are available for a more sophisticated comparison. Variations in loudness can be very easily compared by calculating their difference in decibels. To compare the pitch and timbre vectors, the positions of the vectors within one section are averaged and the resulting vectors compared using the euclidean distance between them.

The final comparison value for two sections is formed by normalizing all five values (beats, bars, loudness, pitch, timbre) and calculating the average difference, which leads to a final similarity between 0 and 1.

This very simple algorithm provides an initial comparison that is sufficient for our purposes, as the given values can be adjusted by the users anyway. Adding weights to the different features could also improve the classification, but since this would need more fine-tuning, at the moment all attributes have the same influence on the final result.

Figure 3. User feedback for one suggestion of the system

3.3 User Feedback

Automatically extracted similarity naturally has its limits. Although the hypothesized glass ceiling [20] for content-based extraction might be circumventable [21], some inherent problems will remain: Especially the issue of personalization is crucial. One user's idea of similarity might completely differ from another's who has a different taste in music or a more sophisticated sense for it. Thus, we are convinced that a metric based on automation is only a first step. For a final classification, user input has to be incorporated into the interface and its underlying algorithms. Last.fm [15] is a prominent example of a robustly classified music library based on user feedback.

3.3.1 Ratings for the automatic suggestions

In its current version, Shades of Music provides a very straightforward mechanism with which the user can correct the suggestions of the system. In the general play view, each song has a button to the left of its icon that makes a small window pop up (see figure 3). Here, the user can rate the suggestion made by the system on a scale from 1 to 5. As songs can appear more than once in the list of suggestions (for example, if the current song section corresponds to the repeated chorus of the other song) and even in several lists (for example for beat and pitch), the user can also criticize certain suggestions while promoting others. This means that the feedback is very specific and doesn't simply rate the computed similarity, but actually the aspect on which it was based.

If the user makes the effort to actually rate a suggestion, this overrides the respective computed value. The user is unable to see the actual internal similarity values and is shown the most similar sections only, so a negative rating always results in a reduction of the calculated similarity (and possibly a removal of the rated section from the list). Therefore, a vote replaces (for the user who made it) the initial similarity calculated by the system for the two sections concerned. The rating of a specific aspect is interpreted as a similarity of 0.0 (1), 0.25 (2), 0.5 (3), 0.75 (4) or 1.0 (5) and stored in the database. If the user votes on the total cumulative value, the rating is used as a factor for all the other attributes, so that their average corresponds with the rating value.

3.3.2 Incorporation of multiple users and feedback

From an initial five-dimensional metric of similarity between sections, the additional user feedback leads for a number of users to a higher-dimensional similarity. In the simplest case, only one user accesses the system and uploads songs from her or his own collection. The system calculates similarity values for existing sections and the user rates these suggestions as replacements for the automatically extracted similarity. In the end, the system reaches an optimal suggestion for this theoretical single user (of course with the overhead of rating millions of section combinations).

As Shades of Music is a web-based system, it is inherently targeting multiple users who all upload their own songs. This is used to reduce the analytical overhead by using meta-data to identify identical songs within separate collections. For these songs, existing classification data can be used. To counter erroneous meta-data, audio thumbnails could also be used for identification as the data is extracted anyway. Previous ratings by other users work as a refinement of the system-generated similarity: All ratings for an attribute of a pair of sections are again converted to a similarity value and, together with the system-generated one, averaged to reach a final value. In this way, we are able to improve suggestions even for new users (as long as they upload existing songs which were already rated by other users). Once a user starts rating suggestions within her or his own collection, these ratings are of course again directly applied (see 3.3.1).

4. SUMMARY AND FUTURE WORK

We presented Shades of Music, a web-based system that lets users discover connections between parts of songs within their music collection. For an exemplary song, a number of similar song sections are displayed, regarding the five attributes beats, bars, loudness, pitch and timbre and an average total. The user can give a rating for a suggestion and thus improve the system's results for himself and others. Informal first feedback showed great potential for the application as especially users with large song collections were curious what connections might be discovered. As a user study for our system should show the merits of the underlying idea and not, for example, the usability of the interface, we plan to open the system for multiple users over a longer period of time and collect our observations. In this way, we will also be able to investigate the value of the integrated rating system.

Extensive testing showed that our prototype also has some shortcomings. First of all, we used a rather simple and not state-of-the-art algorithm for calculating the similarity between sections. When improving this, we would also address the lack of scalability caused by the pair-wise comparison of sections, for example by indexing [22]. With our initial test set of ten songs and an average number of twenty extracted sections, adding one song already leads to a total of 20.000 comparisons (4.000 for each of the five attributes).
The user interface can also be improved in several ways: The representation of the current song as grey section blocks does not help in understanding its structure. Labels with 'verse' or 'chorus' might help, but automation to do that is probably not feasible. Heuristics, such as "repeated sections are a chorus" will most likely be insufficient. Interface elements for labelling could be included to let users do that (and maybe also add the lyrics to the song for additional orientation).

The ways in which users are able to give feedback could also be expanded: Adjustment of section borders or suggestion of new songs (or sections) are only two ideas. Based on our algorithm of averaging all users' votes and the Echo Nest value for novel users we also face the problem of changing suggestions if new votes arrive. To avoid confusing our users with ever-changing suggestions, it might help to only initially use this method and don't update the results every time the interface is launched. Finally, with the generated database of related song sections, additional projects are also feasible: Novel visualizations for a global music collection as a network of interconnected song sections could prove interesting just as clustering the user community ("neighbors" in Last.fm) based on their votes.

5. ACKNOWLEDGMENTS

This work was funded by the University of Munich and the state of Bavaria. We would like to thank the members of the echonest.com forum for valuable feedback and support.

6. REFERENCES

[1] M.A. Bartsch, and G.H. Wakefield: "To catch a chorus: Using chroma-based representations for audio thumbnailing," *IEEE Workshop on the Applications of Signal Processing to Audio and Acoustics*, pp. 15–18, 2001.

[2] The Echo Nest, 2009-05-11, http://the.echonest.com

[3] Y.H. Tseng: "Content-based retrieval for music collections," *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 176–182, 1999.

[4] W.H. Tsai, H.M. Yu, and H.M. Wang: "A query-by-example technique for retrieving cover versions of popular songs with similar melodies," *Proceedings of the International Symposium on Music Information Retrieval*, pp. 183–190, 2005.

[5] A. Kapur, M. Benning, and G. Tzanetakis: "Query-by-beat-boxing: Music retrieval for the DJ," *Proceedings of the International Conference on Music Information Retrieval*, pp. 170–177, 2004.

[6] G. Tzanetakis, A. Ermolinskyi, and P. Cook: "Beyond the query-by-example paradigm: New query interfaces for music information retrieval," *Proceedings of the 2002 International Computer Music Conference*, pp. 177–183, 2002.

[7] E. Pampalk, A. Rauber, and D. Merkl: "Content-based Organization and Visualization of Music Archives," *Proceedings of the tenth ACM international conference on Multimedia*, pp. 570–579, 2002.

[8] F. Morchen, A. Ultsch, M. Nocker, and C. Stamm: "Databonic visualization of music collections according to perceptual distance," *Proceedings of the 6th International Conference on Music Information Retrieval*, 2005.

[9] P. Knees, M. Schedl, T. Pohle, and G. Widmer: "An innovative three-dimensional user interface for exploring music collections enriched with meta-information from the web," *Proceedings of the ACM Multimedia*, pp. 17–24, 2006.

[10] R. van Gulik, F. Vignoli, and H. van de Wetering: "Mapping music in the palm of your hand, explore and discover your collection," *Proceedings of the 5th ISMIR Conference*, 2004.

[11] M. Goto, and T. Goto: "Musicream: New music playback interface for streaming, sticking, sorting, and recalling musical pieces," *Proceedings of the 6th International Conference on Music Information Retrieval*, pp. 404–411, 2005.

[12] F. Vignoli, and S. Pauws: "A music retrieval system based on user-driven similarity and its evaluation," *Proceedings of 6th ISMIR Conference*, pp. 272–279, 2005.

[13] G. Adomavicius, and A. Tuzhilin: "Toward the next generation of recommender systems: A survey of state-of-the-art and possible extensions," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 17, No. 6, pp. 734–749, 2005.

[14] K. Hoashi, K. Matsumoto, and N. Inoue: "Personalization of user profiles for content-based music retrieval based on relevance feedback," *Proceedings of the eleventh ACM internation conference on Multimedia*, pp. 110–119, 2003.

[15] Last.fm, 2009-05-11, http://www.last.fm

[16] Who Sampled?, 2009-05-17, http://www.whosampled.com

[17] E. Pampalk, T. Pohle, and G. Widmer: "Dynamic playlist generation based on skipping behavior," *Proceedings of the 6th ISMIR Conference*, pp. 634–637, 2005.

[18] J. Paulus, and A. Klapuri: "Music structure analysis by finding repeated parts," *Proceedings of the 1st ACM Workshop on Audio and music computing multimedia*, pp. 59–68, 2006.

[19] T. Jehan: *Creating Music by Listening*, PhD Thesis in Media Arts and Sciences. MIT, 2005.

[20] F. Pachet, and J.J. Aucouturier: "Improving timbre similarity: How high is the sky?," *Journal of negative results in speech and audio sciences*, Vol. 1, No. 1, 2004.

[21] T. Lidy, A. Rauber, A. Pertusa, and J. M. Inesta: "Improving Genre Classification By Combination of Audio and Symbolic Descriptors Using a Transcription System," *Proceedings of the International Symposium on Music Information Retrieval*, pp. 61–66, 2007.

[22] R. Cai, C. Zhang, L. Zhang, and W.-Y. Ma: "Scalable Music Recommendation by Search," *Proceedings of the 15th international conference on Multimedia*, pp. 1065–1074, 2007.