Noname manuscript No.
(will be inserted by the editor)

Abstract Picture Generation and Zooming User Interface for Intuitive Music Browsing

Kaori Kusama · Takayuki Itoh

Received: date / Accepted: date

Abstract Today many people store music media files in personal computers or portable audio players, thanks to recent evolution of multimedia technologies. The more music media files these devices store, the messier it is to search for tunes that users want to listen to. We propose MusCat, a music browser to interactively search for the tunes according to features, not according to metadata (e.g. title, artist name). The technique firstly calculates features of tunes, and then hierarchically clusters the tunes according to the features. It then automatically generates abstract pictures, so that users can recognize characteristics of tunes more instantly and intuitionally. It finally visualizes the tunes by using abstract pictures. MusCat enables intuitive music selection with the zooming user interface.

Keywords Musical feature · Clustering · Abstract picture · Zooming user interface

1 INTRODUCTION

Recently many people listen to the music by using personal computers or portable players. Numbers of tunes stored in our computers or players quickly

Kaori Kusama
Ochanomizu University
2-1-1 Ohtsuka, Bunkyo-ku,
Tokyo 112-8610 Japan Tel.: +81-3-5978-5399
Fax: +81-3-5978-5705
E-mail: kaori@itolab.is.ocha.ac.jp

Takayuki Itoh
Ochanomizu University
2-1-1 Ohtsuka, Bunkyo-ku,
Tokyo 112-8610 Japan Tel.: +81-3-5978-5399
Fax: +81-3-5978-5705
E-mail: itot@is.ocha.ac.jp

increase due to the increase of sizes of memory devices or hard disk drives. User interfaces therefore become more important for users to easily select tunes users want to listen to.

Here we think that the procedure to search for the tunes would be more enjoyable, if we develop a technique to interactively search for tunes, not based on metadata but based on features. We usually select tunes based on their metadata, such as titles, artist names, and album names. On the other hand, we may want to select tunes based on musical characteristics depending on situations. For example, we may want to listen to graceful tunes at quiet places, loud tunes at noisy places, danceable tunes at enjoyable places, and mellow tunes at night. Or, we may want to select tunes based on feelings. For example, we may want to select "something speedy/slow" or "something major/minor" tunes based on feelings. We think features are often more informative rather than metadata, to select tunes based on situations or feelings. However, it is not very intuitive if we show feature values of tunes just as numeric characters. We think illustration of features may help intuitive tune selection.

This paper presents MusCat, a music browser featuring abstract pictures and zooming user interface. It visualizes collections of tunes by abstract pictures, based on features, not based on metadata. The technique presented in this paper firstly calculates features of tunes, and hierarchically clusters the tunes according to the features. It then automatically generates abstract pictures for each tune and cluster, so that users can recognize characteristics of tunes more instantly and intuitionally. It finally displays the tunes and their clusters by using the abstract pictures.

We apply an image browser CAT [1] to display a set of abstract pictures. CAT supposes that hierarchically clustered pictures are given and representative pictures are selected for each cluster. CAT places the set of pictures onto a display space by applying a rectangle packing algorithm that maximizes the display space utilization. CAT provides a zooming graphical user interface, which displays representative pictures of all clusters while zooming out, and pictures in the specific clusters while zooming in, to effectively browse the pictures. We call the user interface "MusCat", as an abbreviation of "Music CAT", which enables intuitive music selection with its zooming user interface.

2 RELATED WORK

2.1 Sensitivity

There have been several works on expression of musical features as adjectives. For example, Yamawaki et al. [2] presented that people recognize three pairs of adjectives: heavy - light, speedy - slowly, and powerful - weak, as impression of music, in a part of their study of the correspondence analysis. Similarly, our technique also assigns two pairs of adjectives to musical features.

There have been several studies on the relationship between colors and sensitivity, and our work is on the top of such studies. Color system [9] expresses impressions of colors by arranging them onto a two dimensional sensitivity space, which has warm-cool and soft-hard axes. The color system supposes to place homochromous and polychromous colors onto the sensitivity space: it arranges homochromous colors onto a limited region of the sensitivity space, while it arranges polychromous colors covering whole the sensitivity space. Therefore, we think that impression of music can be adequately expressed by using polychrome colors rather than by homochromous colors.

2.2 Visual Music Representation

Visual user interface is very important for interactive music retrieval, and therefore many techniques have been presented. Goto et al. presented Musicream [3], which enables enjoyable operations to group and select tunes. Lamere et al. [4] presented "Search inside the Music," which applies a music similarity model and a 3D visualization technique to provide new tools for exploring and interacting with a music collection.

There have been several techniques on coupling tunes and pictures for visual music representation. MIST [5] is a technique to assign icon pictures to tunes. As a preparation, MIST requires users to answer the questions about conformity between sensitive words and features of sample tunes or icons. MIST then learns correlations among the sample tunes, and couples the tunes and icons based on the learning results. Kolhoff proposed Music Icons [6], a technique to select abstract pictures suited the tunes based on their features. The technique presented in this paper is also based on the features of tunes, but it focuses on the automatic generation of abstract pictures.

2.3 Image Browser

Image browser is an important research topic, because numbers of pictures stored in personal computers or image search engines are drastically increasing. CAT (Clustered Album Thumbnail) [1] is a typical image browser that supports an effective zooming user interface. CAT supposes that hierarchically clusters pictures are given and representative pictures are selected for each cluster. CAT firstly packs thumbnails of the given pictures in each cluster, and encloses them thumbnail by rectangular frames to represent the clusters. CAT then packs the rectangular clusters and encloses them by larger rectangular frames. Recursively repeating the process from the lowest to the highest level of hierarchy, CAT represents the hierarchically clustered pictures. CAT displays representative images instead of rectangular frames while zooming out, as the initial configuration. On the other hand, CAT displays image thumbnails while zooming in the specific clusters. CAT enables to intuitively search for interesting images, by briefly looking at the all representative images, then zooming into the clusters displayed as interested representative images, and finally looking at the thumbnail images in the clusters.

3 PRESENTED TECHNIQUE

Our technique consists of the following four steps: (1) calculation of features from music media files, (2) clustering of tunes based on the features, (3) generation of abstract pictures, and (4) visualization of tunes by using abstract pictures.

Our current implementation uses acoustic features (MFCC: Mel-Frequency Cepstrum Coefficient) for clustering, because we assumed that acoustic-based division is intuitive to briefly select tunes based on situation, such as "quiet tunes for quiet spaces", or "loud tunes for noisy spaces." It also uses acoustic features for abstract image generation of clusters. At the same time, it also uses musical features for abstract image generation of tunes, because we assume that more information may be needed to select specific tunes from clusters of similarly sounding tunes.

After clustering and abstract image generation, the technique displays the pictures by applying the image browser CAT [1]. We extend CAT so that we can use CAT as a music browser, where we call the extended CAT as "MusCat," as an abbreviation of "Music CAT". Initially MusCat displays all abstract images of clusters while zooming out, and switches them into abstract images of tunes while zooming in. Figure 5 briefly shows the appearance of MusCat.

3.1 Music Feature Extraction

There have been various techniques to extract music features, and some of them have been components of commercial products or open source software. Our current implementation uses features calculated by Marsyas [7] and MIR-toolbox [8]. It may be sometimes difficult to express musical characteristics by single feature value, because features may change gradually or suddenly during a tune. Our current implementation calculates features from randomly selected 15 seconds of the tunes. In the future, we would like to extend our implementation so that we can select the most characteristic features from whole parts of the tunes.

Our implementation uses means and standard deviations of nine bands of MFCC calculated by Marsyas, for clustering and abstract image generation for clusters. We preferred to use Marsyas for MFCC calculation just because it had more variables as results of MFCC calculation.

Also, our implementation uses the following features calculated by MIR-toolbox, for abstract image generation for tunes:
(1) RMS energy (Root-mean-square energy which represents a volume of the tune),
(2) Tempo (Number of beats per minute),
(3) Roll off (Frequency which takes 85% of total energy, by calculating the sum of energy of lower frequencies),
(4) Brightness (Percentage of energy of 1500Hz or higher frequency),

(5) Spectral irregularity (Variation of tones),
(6) Roughness (Percentage of energy of disharmonic frequency), and
(7) Mode (Difference of energy between major and minor chords).

3.2 Clustering

Next, the technique hierarchically clusters music files based on means and standard deviations of MFCC values. We preferred to divide tunes according to MFCC values, because clusters based on acoustic features often denote arrangement or musical situation. This clustering is often good for music selection: we often want to listen to the softly-arranged music at quiet places, or loudly-arranged music at active places. Also, we preferred to apply a hierarchical clustering algorithm, because it can control the sizes and numbers of clusters based on a parameter of similarities or distances among clusters. It is also possible to apply non-hierarchical clustering algorithms such as k-means method, however, we did not prefer them because we need to control other kinds of parameters of non-hierarchical clustering algorithms.

There are many methods of hierarchical clustering for multi-dimensional datasets (e.g., nearest neighbor, furthest neighbor, group average, centroid, median, Ward.) We experimentally applied various clustering techniques for our own collection of tunes, and compared the results by carefully looking at the dendrograms. As a result of our observation, we selected Ward method as a clustering algorithm, because it successfully divides a set of tunes into evenly sized clusters. We preferred to evenly divide the tunes, because our visualization technique represents similarly-sized clusters as similarly-sized rectangles, and therefore all clusters are visually more comprehensively displayed.

We had a small experiment to observe if the clustering result is really intuitive. We asked subjects to listen to several pairs of tunes belonging to the same cluster, and to examine if the pairs are really similar. Also, we asked them to listen to several pairs of tunes belonging to the different clusters, and to examine if the pairs are really different. Unfortunately, result was not really good. Especially, subjects often negatively responded when tunes of particular artists or genres are divided into different clusters.

To solve the problem and improve the impression of clustering results, we applied a constrained clustering method, which divides n tunes using a matrix R calculated by the following equation:

$$R = (1-r)S + rC (0 \leq r \leq 1) \tag{1}$$

Here, S is a matrix consisting of $n \times n$ elements, where the element s_{ij} denotes the similarity of the MFCC values between the i-th and the j-th tunes, taking the range $0 \leq s_{ij} \leq 1$. C is another matrix consisting of $n \times n$ elements, where the element c_{ij} denotes the constrain between the i-th and the j-th tunes, taking the range $0 \leq c_{ij} \leq 1$.

Here, we suppose that each tune has a keyword describing genre or arrangement. We asked subjects to listen to the pairs of tunes annotated by the same keyword, and answer if the pair should be in the same cluster or not. We aggregated the answers and calculated the ratio of the positive answers p_k for the k-th keyword. We then treat that $c_{ij} = p_k$, if both the i-th and the j-th tunes share the k-th keyword, and otherwise $c_{ij} = 0$. Finally, we repeated the clustering process while varying the value of r, observed the clustering results, and subjectively fixed r.

3.3 Abstract Picture Generation from Musical Features

Our technique generates two kinds of abstract pictures to represent tunes, and displays so that users can intuitively select the tunes. One of the abstract pictures is generated from musical features, and the other is generated from acoustic features. Our implementation calculates musical features using MIR toolbox, and acoustic features using Marsyas, as described in below sections. It generates the abstract pictures for each tune, and for each cluster. It calculates the average of feature values for each cluster in order to generate the abstract pictures of the clusters.

We believe the approach is effective as discussed below. First reason is that visual and music words are often related. Actually, many music works depict scenery that artists looked and imagined. Also, some painting artists expressed emotion while listening to the music as abstract pictures [4]. Second reason is that humans have synesthesia [7]. Impression of colors is related to impressions of sound, because some people have perception so called "the colored hearing," which associates colors by listening to the music. Third reason is that impressions of sounds and colors are often expressed by same adjectives. Therefore, we think that music can express through the abstract pictures considering colors to visualize the music.

This section describes our implementation to automatically generate abstract pictures. Currently it is just designed based on our subjective, but we do not limit the abstract pictures to the following design.

3.3.1 Color Assignment

Our technique firstly assigns colors to the objects in abstract pictures. As mentioned previously, the abstract image generation technique using MIR toolbox applies polychrome colors, because the polychrome color arrangement can express impression richer than the monochrome color arrangement. The technique selects colors of abstract pictures based on color image scale [9]. It is a color system that distributes combination of three colors in a two dimensional space, so called "sensibility space", which has the warm-cool and the soft-hard axes, as shown in Figure 1. The technique distributes tunes into this sensibility space, and assigns the colors corresponding in the sensitivity space to the tunes.

Here, let us discuss which features match to the warm-cool and the soft-hard axes. We feel major chords express positive impression similar to bright

Fig. 1 Sensitivity space based on color image scale.

and warm colors, while on the other hand, minor chords express negative impression similar to dark and cold colors. Based on the feeling, we assign Mode is to warm-cool axis. Similarly, we assign Roll off to soft-hard axis. We think listeners often use substitute adjective words such as "light" or "soft" for the impression of music, and often these impression is related to frequency-based tone balances.

Our current implementation places many sample colors onto the sensitivity space. Calculating Mode and Roll off of a tune, our implementation places the tune onto the sensitivity space, and selects the color closest to the tune. Here, let the position of the tune as (m_{wc}, m_{sh}), and the position of a color set as (c_{wc}, c_{sh}). Our implementation selects the color set that brings the minimum distance between the above two positions.

3.3.2 Abstract Picture Design

This section describes an example of design of abstract pictures based on music features. Our design first generates the following three layers, 1) gradation layer, 2) a set of circles, and 3) a set of stars, as shown in Figure 2.

As the first step, MusCat calculates threshold values for each feature by applying the following equation, from the maximum and minimum values of each feature of a given music collection.

MusCat calculates the above thresholds for each music collection, not applying constant values as thresholds. This dynamic threshold calculation approach is especially useful for biased music collections which has particular genre, because it can generates various abstract images from such biased collections.

MusCat assigns RMS energy to the gradation layer. We subjectively designed to represent power, weight, and broadening by the gradation. We evaluated that RMS energy is the most suitable feature for this representation. MusCat generates the gradation on 1/5 of the area of the layer, for the tunes which have especially smaller RMS energy values. Otherwise, it generates the gradation on n/5 of the areas of the layer, according to the RMS energy values.

MusCat assigns Tempo, Spectral irregularity, and Roughness, to the generation of orthogonally arranged circles. We subjectively designed frequency of rhythm by the number of circles, irregularity and variation of music by irregularity of circles. We evaluated that Tempo, Spectral irregularity, and Roughness are the most suitable features for this representation.

MusCat assigns Brightness to the number of randomly placed stars. We expected many of users will associate bright music from the stars, and we subjectively evaluated that Brightness is the most suitable feature for this representation.

After generating the three layers, the technique finally synthesizes the three layers to complete the abstract picture generation.

Figure 3 shows two examples of abstract images generated from musical features. Figure 3(Left) denotes a tune that tempo is moderately rapid (because the number of circles is moderately large), roughness and spectral irregularity are very small (because circles are very regularly sized and aligned), and volume is medium (because graduation is well-balanced). Figure 3(Right) denotes another tune that tempo is rapid (because the number of circles is quite large), roughness and spectral irregularity are large (because circles are very irregularly sized and aligned), and volume is loud (because graduation is biased to a deeper color).

3.3.3 Abstract Picture Generation from Acoustic Features

Muscat generates another design of abstract pictures for clusters. It simply represents average power values of nine bands of MFCC as colored squares. Figure 4 shows an illustration how our technique generates abstract pictures. Our implementation defines nine colors for the bands based on the soft-hard

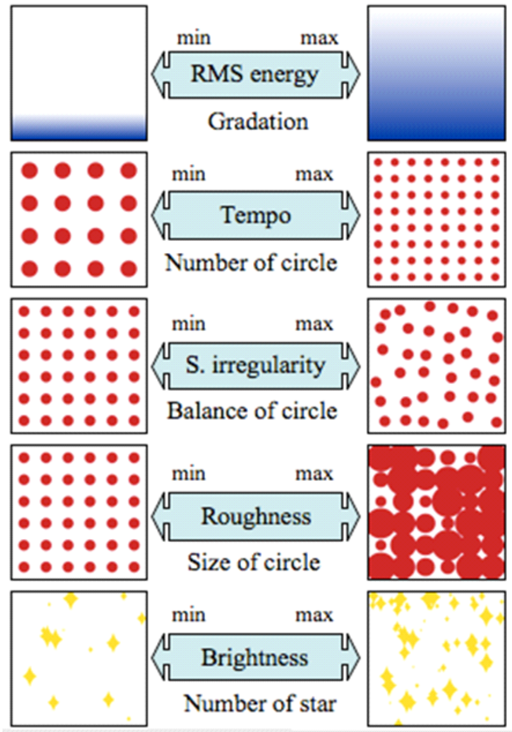

Fig. 2 Automatic generation of three layers of images based on music features.

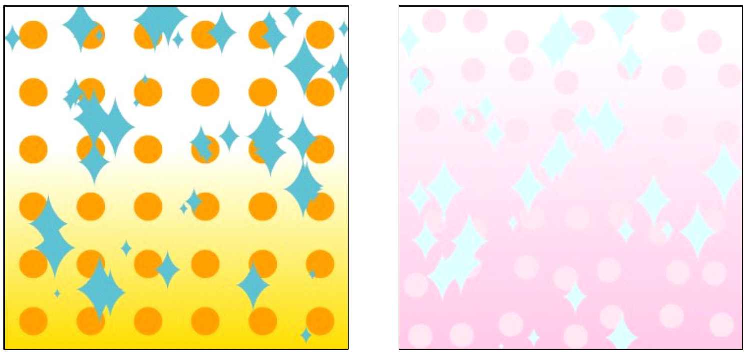

Fig. 3 Examples of abstract images from musical features.

axes shown in Figure 1. While we subjectively designed to represent nine features of MFCC by corresponding colors one-by-one, we employ monochrome colors against abstract images using musical features employ polychrome colors. It assigns softer colors to higher bands, and harder colors to lower bands. It calculates the average values of the mean values of the tunes for each cluster,

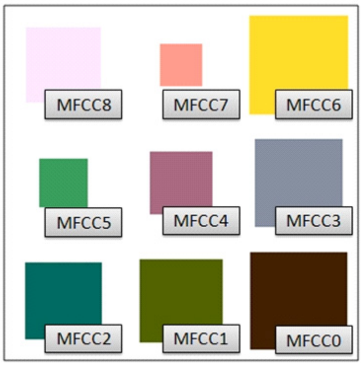

Fig. 4 Illustration of abstract picture generation for clusters.

and calculates the sizes of the colored squares as proportional to the average values. The pictures denote acoustic textures of tunes in the clusters.

4 Examples

This section describes examples using the presented technique. We used Marsyas [7] and MIRtoolbox [8] for feature extraction, and R package for clustering. We implemented abstract picture generation module in C++ and executed with GNU gcc 3.4. We implemented MusCat in Java SE and executed with JRE (Java Runtime Environment) 1.6.

In the following example, we applied 88 tunes which are categorized into 11 genres (pops, rock, dance, jazz, latin, classic, march, world, vocal music, Japanese, and a cappella), provided by RWC Music Database [10].

Figure 5 is an example snapshot of MusCat. Initially MusCat shows all abstract pictures of clusters while zooming out, as shown in Figure 5(Left). When a user prefers a picture and zooms in it, MusCat turns the abstract pictures for clusters to abstract pictures for tunes, as shown in Figure 5(Right). Users can select a tune and play it by double-clicking the corresponding picture. Here, this result shows characteristic clusters (a), (b), and (c). The abstract image of cluster (a) denotes that the lowest frequency band (MFCC 0) of its tunes is respectively large. The abstract images of the three tunes in the cluster (a) have size-varying and un-aligned circles, and respectively more stars. These images denote that the tunes in cluster (a) have loud low and high frequency sounds, and respectively more disharmonic sounds. Actually, two of the three tunes are dance music that have loud Bass Drum beats, and backing of electric disharmonic sounds.

The abstract image of cluster (b) denotes that low and high frequency bands (MFCC 0, 1, 2, 7, and 8) are extremely small. The abstract images of the two tunes in the cluster (b) have aligned circles, and less number of starts. Colors of circles are different between the two images. Actually, the two tunes

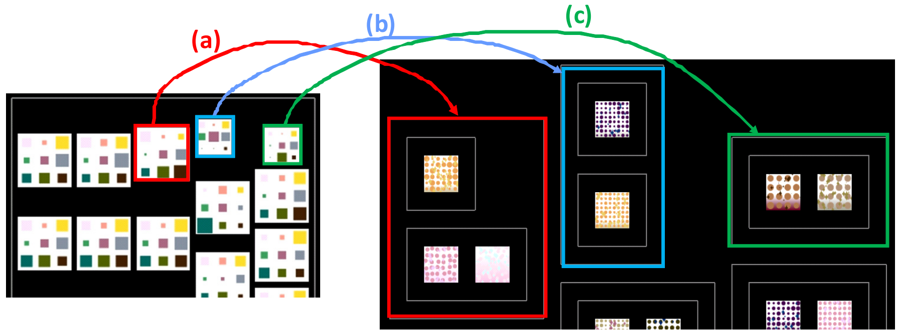

Fig. 5 Example of visualization by MusCat. (Left) MusCat displays abstract pictures of clusters while zooming out. In the cluster (a), the lowest frequency band (MFCC 0) is relatively large. In the cluster (b), low and high frequency bands (MFCC 0, 1, 2, 7, and 8) are extremely small. In the cluster (c), specific two frequency bands (MFCC 1 and 3) are large, but others are quite small. (Right) MusCat displays abstract pictures of tunes while zooming in.

Fig. 6 Example of visualization by MusCat. (Upper)All pictures are generated based on MFCC. (Lower)All pictures are generated based on musical featrures.

are Japanese traditional folk music played by old wood wind instruments, without bass part or high-tone percussions. One of the tunes plays major scale, and the other plays minor scale. That is why colors of two pictures of the tunes are much different. The abstract image of cluster (c) denotes that specific two frequency bands (MFCC 1 and 3) are large, but others are quite small. The abstract images of the two tunes in the cluster (c) have smaller number of well-aligned, equally-sized circles. Actually, the two tunes are slow female vocal songs, and these backings are only piano or Japanese traditional strings. That is why the abstract image of the cluster denotes that specific two frequency bands are large, and the abstract images of the tunes have less number of circles.

As above mentioned, users of MusCat can select clusters of tunes based on acoustic features by specifying the abstract images of clusters, and then select tunes based on musical features. We think this order is reasonable: users can firstly narrow down the tunes based on their situations: for example, quiet tunes for quiet places, loud tunes for noisy places, and so on. They can select features of tunes in the specific clusters based on their feelings.

Though our original concept of MusCat supposes to show different abstract images for clusters and tunes, it is possible to generate abstract images of tunes based on MFCC, as well as those of clusters. Also, it is possible to generate abstract images of clusters based on average of musical features. Figure 6(Upper) shows an example of abstract images of clusters and tunes generated based on MFCC. MFCC-based abstract images of tunes in a cluster are usually very similar, since their corresponding tunes are clustered based on MFCC. Figure 6(Lower) shows an example of abstract images of clusters and tunes generated based on musical features. Feature-based abstract images of clusters are quite different from abstract images of tunes in the cluster, since the musical features are not referred in the clustering process.

In the next example, we applied 50 Jazz tunes containing 5 piano solo, 5 guitar solo, 5 duo, 5 trio, 10 quartet, 6 octet, 2 vocals, 2 big bands, 2 mode jazz, 2 funk jazz, 2 free jazz, and 10 fusion.

Abstract images of three tunes in a cluster indicated by a blue square in Figure 7(Upper-left) denote that lower frequency (MFCC 0, 1, and 2) are loud, and others are small. These all tunes are guitar solo tunes, and we found that actually sounds of lower strings are recorded louder. Figure 7(Upper-right) denotes abstract images of five piano solo tunes indicated by red squares, where three of them are concentrated in the same cluster. Actually, the three tunes are simply arranged with fewer notes, while other two tunes sound more lively. These examples suggest that MFCC-based clustering is effective to gather some kinds of acoustically characteristic instruments or arrangements into a same cluster.

Abstract images of variously arranged same tune are indicated as green squares in Figure 7(Lower-left). The arrangements are piano solo, piano trio (piano, bass, and drums), and piano quartet (piano, bass, drum, and vibraphone). This example suggests that even same tune can be divided according to acoustic features and visually distinguished.

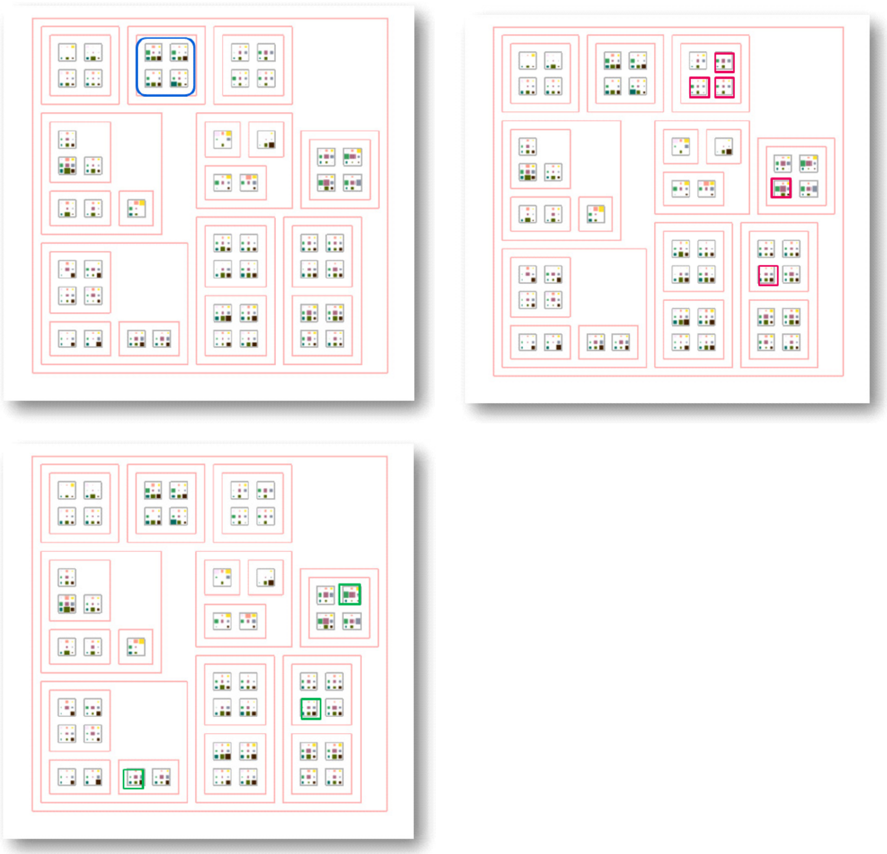

Fig. 7 Example of visualization by MusCat. (Upper-left) Three guitar solo tunes are concentrated in a cluster. (Upper-right) Three piano solo tunes are concentrated in a cluster. (Lower-left) Variously arranged tune is divided into different clusters.

5 EXPERIMENTS

We had a user evaluation with 17 subjects to examine the validity of abstract pictures. All the subjects were graduate and undergraduate students majoring computer science, where several of them had deep knowledge about music while others did not have. We also asked the subjects to play with MusCat and give us comments or suggestions. We showed 14 tunes and their abstract pictures to the subjects. We then asked to evaluate the suitability of the pictures for the tunes by 5-point scores, where ”5” denotes suitable, and ”1” denotes unsuitable. This experiment obtained positive evaluations for five tunes which obtained higher than 3.5 average scores; on the other hand, it obtained relatively negative evaluations for other four tunes which had lower than 2.5 average scores. Figure 8 shows the positively or negatively evaluated abstract pictures. Here, three of negatively evaluated pictures, (f), (g), and (i),

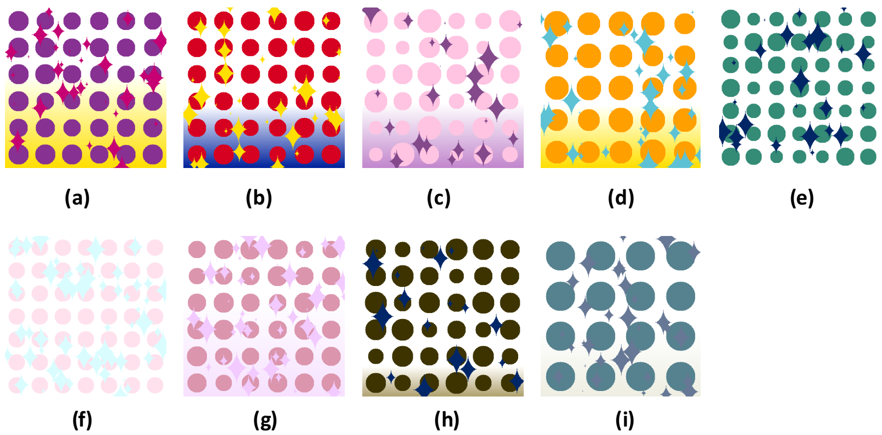

Fig. 8 Positively or negatively evaluated abstract pictures. (a) (e): Positively evaluated. (f) (i): Negatively evaluated.

were generated from features of non-melodious tunes (e.g. Rap music, African music) and therefore it might be difficult to associate particular colors and textures from such tunes.

We then clustered the subjects, according to the similarity of their answers, to determine what kind of preferences of subjects governs the evaluation result. We found that three subjects in a cluster positively evaluated (g), and negatively evaluated (b), against that averages of the evaluation were relatively high. One of the subjects in the cluster actually told that she might prefer purple during the experiment. We also found that four subjects in another cluster negatively evaluated (d) and (e) against the average. The picture was generated from features of a Japanese traditional tune and therefore it might be difficult to associate particular color and texture. Such observation suggests that we may need to customize the feature extraction and abstract image design according to the preferences of users.

Next, we asked subjects to give us any comments during the experiments, to discuss the reason why we could not obtain positively evaluations for several tunes. Subjects commented as follows:

- Colors were dominant for the first impressions of pictures.
- Gradation of pictures was unremarkable because of the color arrangement.
- Some subjects might associate colors of music genre by fashion; for instance, colors of rock were black and white.

We think the above comments are key points to improve the evaluation of users. We will discuss about them more in the future.

We also asked subjects to give any comments on usability of MusCat. Many subjects gave us positive comments that MusCat was useful to use when they wanted to select tunes according to intuition or emotion, especially for unknown tunes. Some of them suggested us that MusCat can be an alternative of shuffle play mechanism of music player software. We also got some constructive suggestions from subjects. Some of them suggested us to indicate the metadata or feature values of tunes they selected, even though they selected the tunes according to the impression of abstract pictures. We think it is interesting to add more effective pop-up mechanism for the selected tunes to indicate such information. Some other subjects commented that they might loss which part they are zooming in. We would like to add a navigation mechanism to solve the problem.

6 CONCLUSION AND FUTURE WORK

We presented MusCat, a music browser applying abstract pictures and zooming user interface. The technique uses features to cluster tunes, and to generate abstract pictures, so that users can recognize tunes more instantly and intuitionally without listening. Following are our potential future work:

Music feature: Our current implementation calculates features from randomly selected 15-second segment of a tune. We would like to calculate features from all 15-second segments of a tune and select the most preferable or characteristic features from the calculation results.

Abstract picture: Our current abstract picture design is just an example, and therefore we think there may be better designs. Some of our subjects pointed that color is more important for the impression of pictures, than shapes and properties of objects. However, we have not yet found the best scheme to assign three colors to gradation, circles, and star. We need to discuss better schemes to assign the three colors. Another discussion is mood-based design of abstract pictures, since current design directly represents feature values.

User interface: Current version of MusCat just plays the music by click operations, and simply indicates text information. We would like to extend the functionality. We would like to develop to show more metadata information of the selected tunes. Also, we would like to develop to play a set of tunes in the selected clusters by one click operation.

Interactive feature selection: Current implementation of musical feature based abstract image generation method always refers every feature values, and therefore the visualization result may complicated for users. It will be easy to visually compare tunes or clusters if users can interactively select interested features and fix values of other features before the abstract image generation. We would like to implement this function and test it.

Scalability: This paper showed a visualization example with less than 100 tunes. We believe that the visualization of music collections helps to provide efficient and effective accesses to large music collections. We would like to apply MusCat to much larger music collections, and test its scalability and reasonability.

References

1. A. Gomi, R. Miyazaki, T. Itoh, J. Li: "CAT: A Hierarchical Image Browser Using a Rectangle Packing Technique," 12th International Conference on Information Visualization, pp. 82-87, 2008.
2. K. Yamawaki, H. Shiizuka: "Characteristic recognition of the musical piece with correspondence analysis," Journal of Kansei Engineering, Vol. 7, No. 4, pp. 659-663, 2008.
3. M. Goto, T. Goto: Musicream: "New Music Playback Interface for Streaming, Sticking, Sorting, and Recalling Musical Pieces," Proceedings of the 6th International Society for Music Information Retrieval, pp. 404-411, 2005.
4. P. Lamere, D. Eck: "Using 3D Visualizations to explore and discover music," Proceedings of the 8th International Society for Music Information Retrieval, pp. 173-174, 2007.
5. M. Oda, T. Itoh, "MIST: A Music Icon Selection Technique Using Neural Network", NICOGRAPH International, 2007.
6. P. Kolhoff, J. Preub, J. Loviscach: "Music Icons: Procedural Glyphs for Audio Files," IEEE SIBGRAPI, pp. 289-296, 2006.
7. Marsyas, http://marsyas.info/
8. O. Lartillot: "MIRtoolbox," http://www.jyu.fi/hum/laitokset/musiikki/en/research/coe/materials/mirtoolbox
9. S. Kobayashi: Color System, Kodansha, Tokyo, 2001.
10. RWC Music Database, http://staff.aist.go.jp/m.goto/RWC-MDB/