MUSICREAM: NEW MUSIC PLAYBACK INTERFACE FOR STREAMING, STICKING, SORTING, AND RECALLING MUSICAL PIECES

Masataka Goto **Takayuki Goto**
National Institute of Advanced Industrial Science and Technology (AIST).
IT, AIST, 1-1-1 Umezono, Tsukuba, Ibaraki 305-8568, Japan
m.goto@aist.go.jp

ABSTRACT

This paper describes a novel music playback interface, called *Musicream*, which lets a user unexpectedly come across various musical pieces similar to those liked by the user. With most previous "query-by-example" interfaces used for similarity-based searching, for the same query and music collection a user will always receive the same list of musical pieces ranked by their similarity and opportunities to encounter unfamiliar musical pieces in the collection are limited. Musicream facilitates active, flexible, and unexpected encounters with musical pieces by providing four functions: the *music-disc streaming function* which creates a flow of many musical-piece entities (discs) from a (huge) music collection, the *similarity-based sticking function* which allows a user to easily pick out and listen to similar pieces from the flow, the *meta-playlist function* which can generate a playlist of playlists (ordered lists of pieces) while editing them with a high degree of freedom, and the *time-machine function* which automatically records all Musicream activities and allows a user to visit and retrieve a past state as if using a time machine. In our experiments, these functions were used seamlessly to achieve active and creative querying and browsing of music collections, confirming the effectiveness of Musicream.

Keywords: Music interface, Music player, Music-collection browser, Query-by-example, Playlist.

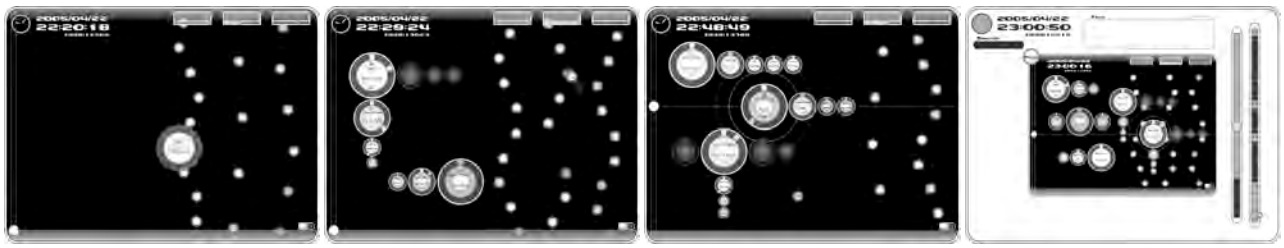
1 INTRODUCTION

Although current music playback interfaces can satisfy user desires like "I would like to hear this song" based on the name of the song or the artist's name, they have not been very strong in functions that satisfy desires like "I want to hear something" or "I want to hear something my way." Such desires, though, will become more common as part of the next-generation music-listening environment as flat-rate, all-you-can-hear music subscription services giving users unlimited on-line access to a million or more musical pieces become widespread. We will soon be hearing users say things like: "If I like a song that I have somehow chosen from a huge music collection, I want to be able to pick out, by myself, other songs similar in mood to that song one after another", or "When I want to hear music in an order that I prefer, I would like to try out a playback order with the same high degree of freedom I have in stacking compact discs (CDs) on my desk and ordering them as I like", or even "I would like to reproduce a song order that I once listened to in the past (such as on a day with good memories) and hear it now."

Functions that can satisfy such desires have been lacking in the past, and with the aim of rectifying this situation, technologies to enable such functions on an individual basis have been developed. However, there are still no interfaces that provide a comprehensive integrated music-listening environment in which functions such as these can be used in an easy and seamless manner. For example, while many effective similarity measures [1, 2, 3, 4, 5, 6, 7, 8] have been proposed to enable users to listen to songs of similar mood, most previous "query-by-example" interfaces for searching similar music only provide functions for listing songs similar to a certain song. No consideration has been given to an operation that finds similar music in a way analogous to a listener pulling one CD after another from a stack of CDs and listening to them as much as one desires. As for the music playback order, many music players already have functions enabling users to specify *playlists* (i.e., lists of musical pieces for playback). The drawback here is that the operations provided by these functions are usually limited to changing the order of pieces and to inserting and deleting pieces in existing playlists; i.e., the degree of freedom is low. In addition, the only way that a user can currently remember the order in which certain musical pieces were listened to at some time in the past is to make a conscious effort to store (save) the playlist at that time for later retrieval (loading).

In this paper, we propose a new music playback interface called *Musicream* ("music" + "stream") that enables all of the above music-listening formats to provide a user with a high degree of freedom. Musicream applies the following functions to create a novel music-listening environment.

1. *Music-disc streaming function*: lets musical-piece entities (discs) stream down one after another on a screen for user perusal and selection.
2. *Similarity-based sticking function*: attaches similar musical pieces to a music disc selected by the user in a manner similar to how two magnets attract each other.
3. *Meta-playlist function*: enables the user to try out various playback orders by making it easy to rearrange groups of musical pieces on the screen in a manner similar to rearranging CDs on a (real) desktop.



Music-disc streaming function: Active and unexpected encounters with interesting musical pieces

Similarity-based sticking function: Controllable and flexible "query-by-example" similarity search

Meta-playlist function: Advanced playlist editing with a high degree of freedom

Time-machine function: Intuitive browsing and perfect recalling of past music listening activities

Figure 1: Four functions of Musicream (demonstration video clips are available at http://staff.aist.go.jp/m.goto/ISMIR2005/).

4. *Time-machine function*: provides a means of moving freely back in time as if using a time machine to return to a point in the past where one was listening to music.

These functions make it possible, for example, to pick out a musical piece (disc) from among many streaming on the screen and then attach other pieces similar in mood to that musical piece, thereby creating a group of pieces. Furthermore, by simply continuing this process, several groups (playlists) of musical pieces can be formed and placed in a blank area on the screen in a manner that decides the playback order of those groups. In this way, users can enjoy music in a much more active manner. Moreover, all operations on Musicream are recorded automatically so that any past state can be returned to at any time. A user can reproduce what he or she was listening to at some point in the past and then continue with other operations from within that state if so desired.

This paper is organized as follows. Section 2 introduces the functions provided by Musicream and Section 3 describes how they are implemented. Section 4 presents the results of experiments on the use of Musicream. Section 5 discusses related research and Section 6 summarizes this paper's contributions.

2 MUSICREAM FUNCTIONS

Musicream provides four novel functions (Figure 1) to satisfy user desires like "I want to hear something" and "I want to hear something my way." Instead of working as independent functions, they operate in a cooperative and comprehensive manner to enable new music-listening formats. Each of these functions is described below.

2.1 Music-disc streaming function

This function presents the user with images of discs, each corresponding to a musical piece in a music collection [1]. These discs flow from top to bottom on the screen one after another, and the user may select a disc and listen to that musical piece. This function is especially useful as a means of encountering various musical pieces by chance as opposed to making a specific request as in "I want to hear *this* musical piece."

The encountering of previously unknown music by chance is not without precedent. People often listen to music broadcast on radio and television or listen to songs or musical pieces on hit charts or recommended by friends. Various studies have been made on music-recommendation or playlist-generation schemes [9, 10, 11, 12, 13, 14, 15, 16] based on collaborative (social) or content-based filtering. If a person were to listen to only musical pieces that have been selected by other people or systems, though, such encounters would tend to be passive in nature. At the same time, it is difficult to use a personal music collection consisting of (tens of) thousands of musical pieces for the purpose of encountering music by chance. While music search methods based on bibliographic (catalog) information such as titles and artist names have been used, they have not considered chance encounters with musical pieces [2]. There are also methods based on folder-based hierarchical classification by music genre, artist name, etc. that allow a user to search from one hierarchical layer to another to refine the search. These methods, however, narrow down searches by promoting hierarchical selections such as "jazz ⇒ bebop," which unnecessarily decrease the possibility of unexpected but interesting encounters with musical pieces.

Our music-disc streaming function enables flexible music encounters not possible through broadcasting, recommendations, traditional searching, or hierarchical structures. Figure 2 illustrates the screen of Musicream with some basic terminology. Each of the three rectangular boxes at the top represents a *music-supply "tap"* that releases small discs one at a time, each disc corresponding to a musical piece. These discs fall straight down at the same speed (since a tap is fairly wide, released discs are spread out crosswise along a tap). The taps are designed so that each releases musical pieces of a different mood, and the rate of release can be adjusted with the sliders above each tap. A falling disc that reaches the bottom of the screen (the *"ground"*) disappears. The user may remove a disc that appears interesting from these streams of falling discs and listen to that musical piece by drag-

[1] In this paper, "music collection" means any set of musical pieces that a user is able to listen to. This would include groups of musical pieces stored on portable music players and personal computers as well as those on flat-rate, unlimited music subscription services.

[2] Although some existing players (such as Apple iPod shuffle) support a random (shuffle) playback function, they do not allow a user to intentionally control the general nature of music encounters, which Musicream does allow.

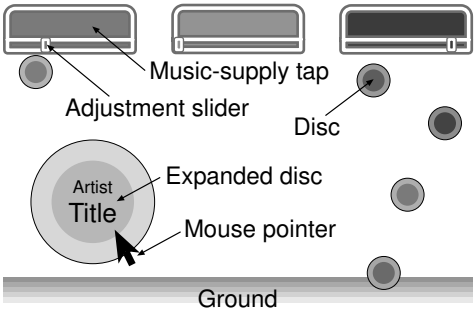


Figure 2: Music-disc streaming function: Discs corresponding to musical pieces stream downwards from three colored taps.

ging the disc using a mouse (or a stylus, touch tablet, etc.). Rolling the mouse pointer over a small disc expands it and displays the title and name of the artist for the musical piece in question, as shown in the figure to the left.

Each tap and disc is given a color that reflects the mood or feeling of a musical piece (each disc falls out of a tap with a similar color). In other words, similarity in color is associated with similarity in musical pieces, which means that a user who likes the musical piece of a selected disc can choose other pieces based on the color of that disc. While using the music-disc streaming function, a user often wants to perform that operation time after time, and this is where the similarity-based sticking function described next comes in.

2.2 Similarity-based sticking function

With this function, the user takes (picks up) a disc previously removed from the streaming discs and touches other discs that are still streaming. This operation has the effect of selectively "sticking" (attaching) discs of similar mood to the original disc one after another. This function may be viewed as a "query-by-example" search for musical pieces, but it is not a search that simply presents a list of similar songs. In our similarity-based sticking function, the user collects musical pieces from streaming pieces according to the user's own choice. The important point here is that musical pieces can be encountered while the user is performing an operation similar to the way people pick up things they like in real life.

Here, the "ease of sticking" has been designed to depend on the similarity between two musical pieces in terms of mood[3]. As shown in Figure 3, two discs with high similarity will stick on first contact whereas two discs with low similarity will not stick unless brought into contact several times. In this way, the range of similarity of musical pieces to be added can be easily adjusted by maneuvering the picked-up disc appropriately through the streams of falling discs.

Sticking discs using this function forms an overlapping series of discs as shown at the left of Figure 4. This is called "*compact mode.*" Clicking the top disc of this series rearranges the discs into a non-overlapping, horizontal series as shown at the right of Figure 4. This is called "*maintenance mode.*" In this mode, moving the mouse pointer over a disc expands that disc to make it the focus of that series (the discs to either side of the disc-in-focus become slightly larger). It is also possible to rearrange disc order in the series or remove a disc from the series, as shown in Figure 5.

[3]Instead of mood, which is used in our current implementation, any similarity based on content-based or collaborative filtering can be used for this function.

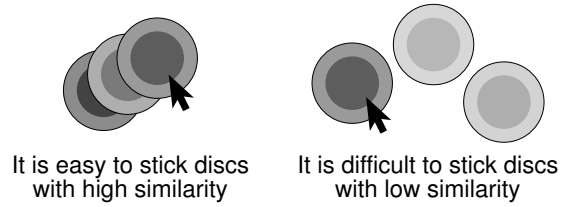


Figure 3: Similarity-based sticking function: The "ease of sticking" depends on the similarity in terms of mood. Similarity in color is associated with similarity in musical pieces.

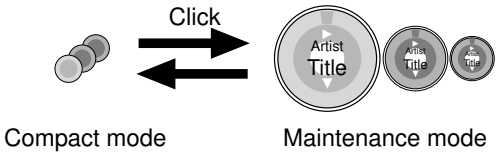


Figure 4: Two disc-series modes.

In maintenance mode, the disc components shown in Figure 6 appear, thus enabling the following functions.

- *Playback control*

  Clicking on the playback button on the disc starts playback of that musical piece. During playback, an animated ripple effect emanates from the disc. If the piece is played until its end, playback automatically changes to the next underlying disc. Clicking again on the button stops playback.

- *Playback position control*

  The playback position slider, whose function is the same as that of playback position sliders provided by ordinary music players, is placed along the periphery (circumference) of the disc. One complete loop in the clockwise direction from the twelve o'clock position corresponds to the length of that musical piece.

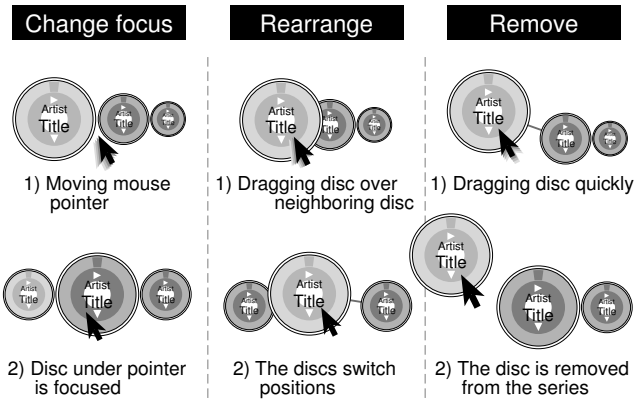


Figure 5: Editing operations for a disc series (playlist).

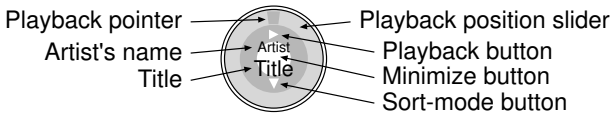


Figure 6: Disc components.

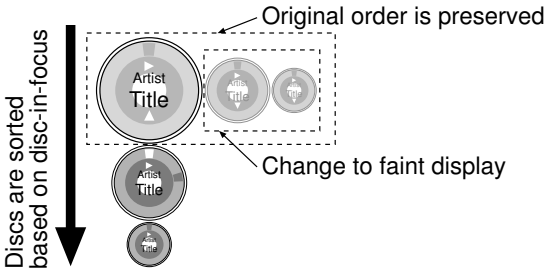


Figure 7: Sort mode: Using the disc-in-focus as a base, other discs in the series descend in order of mood similarity.

Clicking on any position along the slider starts playback from that position in the piece.

- *Minimize*

  Clicking on the minimize button at the center of the disc switches to compact mode.

- *Sort*

  Clicking on the sort-mode button on the disc switches to sort mode as shown in Figure 7. Using the disc-in-focus as a base for sorting, a copy of the other discs in the series appears below that disc in descending order of mood similarity. These duplicated discs disappear upon exiting sort mode.

In Musicream, such a disc series is already a highly functional playlist (i.e., a list of musical pieces that specifies playback order). In existing music players, "zapping" (changing from one musical piece to another in a playlist as the urge arises) in music playback requires the user to perform a two-step procedure. First, the user must (double-)click the title of the desired selection in the playlist, and second, the user must click on the desired playback position on the playback position slider located elsewhere. In Musicream sort mode, in contrast, simply moving the mouse pointer during music playback over other discs in the series automatically starts playing those musical pieces (there is no need for a new playback-start operation). In addition, clicking on the playback position slider on the periphery of a disc right after moving the focus enables a series of operations — musical piece selection without click and playback position specification with a single click — to be performed simultaneously making for the smoothest zapping ever seen.

Similarity-based sticking enables a user to pick out discs from disc streams as desired, create one playlist after another, and leave them on the Musicream screen. Facing these multiple playlists might lead a user to think about a playback order for the playlists. This capability is provided by the meta-playlist function described next.

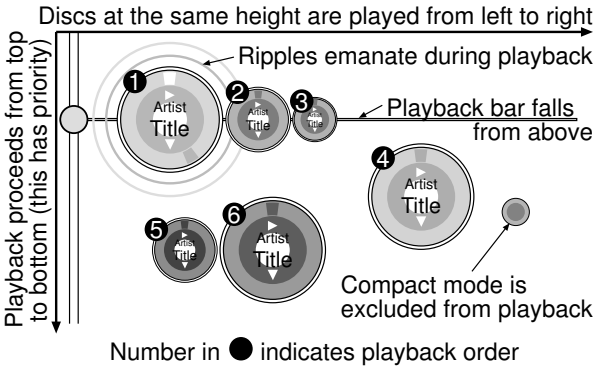


Figure 8: Playback rule of meta-playlist function: The playback bar falls from above and plays playlists in order of contact.

2.3 Meta-playlist function

Similar to rearranging a group of CDs on a (real) desktop, this function enables playlists (multiple series of discs) situated on the screen to be rearranged and their order of playback specified. This is achieved by treating the entire Musicream screen as a meta-playlist and playing back the playlists on the screen in order from top to bottom.

The playback pointer for the entire screen used by this function is simply a straight horizontal line (the "playback bar") as shown in Figure 8. The handle at the left of the playback bar can be used to drag the bar as desired (i.e., upward or downward). Once dragged to a new position, the playback bar starts to drop (automatically moving to the bottom of the screen) and plays back the series of musical pieces in any maintenance-mode or sort-mode playlist (disc series) it comes in contact with. A playlist in compact mode is ignored in this playback.

For this function, only positional relationships in the screen's vertical direction affect playback order. Since horizontal relationships have no affect on playback order, the user can make good use of it to achieve flexible playback control. For example, to consider the best playback order of several playlists through trial and error, the user can arrange those playlists horizontally and then raise or lower each playlist slightly to change their vertical relationship and hear the resulting playback order as the playback bar falls. The user may also arrange playlists according to self-created rules such as placing lively music on the left and peaceful music on the right. In addition, playlists that the user does not presently wish to hear can be placed in compact mode and simply left on the screen — there is no need to delete them to prevent them from interfering with playback.

By allowing free insertion, removal, and rearrangement of groups of musical pieces (playlists) to create meta-playlists, the meta-playlist function enables playlist editing with the highest degree of freedom yet provided. If this type of switching in units of groups were attempted on an existing music player, the operations of selecting and inserting multiple musical pieces would have to be repeated any number of times while remembering group boundaries. Musicream allows groups of musical pieces to be arranged on the screen as individual (small) playlists and allows the user to change their playback order while

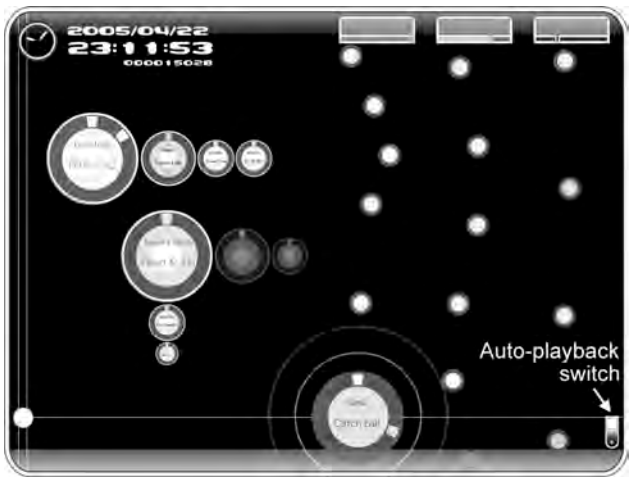


Figure 9: Musicream screen snapshot with the auto-playback switch on: Enabling the auto-playback switch and letting the playback bar fall into the switch activates auto-playback.

preserving their positional relationship in the horizontal direction. This makes intuitive trial-and-error processing possible while using positional memory. Playlist editing in this way is not just a method for changing playback order. It can also be viewed as a creative way of enjoying music with the user becoming actively involved[4].

To facilitate the use of Musicream for playing background music, we also prepared an auto-playback mode that enables listening to musical pieces in succession without requiring user interaction. This mode is entered by turning on the auto-playback switch and letting the playback slider fall into the switch as shown in Figure 9. When playback of the current musical piece ends in this mode, Musicream starts to automatically select and play back a musical piece similar in mood to that piece from among the streaming discs on the screen.

Although existing music players include a random (shuffle) playback function, an abrupt change in mood can easily occur from one song to the next, such as when a wild rock tune follows a romantic ballad. In the auto-playback mode of Musicream, the user is always listening to a new order of musical pieces while continuing to listen to pieces with a similar feel (a mode that allows listening in a format equivalent to conventional random playback is also available).

A user who encounters new music using the music-disc streaming and similarity-based sticking functions, and who creates playlists by trial and error using the meta-playlist function, is most likely to make the act of listening to music by Musicream an everyday activity. As a result, musical pieces played on Musicream will be approximately equivalent to that user's listening experience. Accordingly, if a user would like to know what music he or she was listening to in the past, the user should be able to find out by investigating what music has been played on Musicream. The time-machine function described next makes this possible.

[4]The importance of playlists can be understood from the way that artists carefully determine song order in their albums and from the attention given to WWW sites that present personal playlists (e.g., http://www.artofthemix.org/), for example.

2.4 Time-machine function

This function records all operations performed by the user on Musicream as well as all screen changes so that the user can browse this record of the past and return to an enjoyable point in time whenever desired. This enables the user to reproduce a past Musicream screen (i.e., recall a past listening state) and continue with operations from that screen as if traveling back in time with a time machine. It is also possible to copy and paste a playlist from the past screen onto the present screen.

Reproducing music playback order from the past in this way is difficult to achieve on existing music players. This would require that playlists be intentionally saved for future reference, but this method would only be practical if the user knew exactly what playlists would be important in the future. Musicream has no functions for saving or loading individual playlists because these are not needed as long as the time-machine function is available. However, Musicream allows the user to label the current point in time with a keyword or sentence that can then be used to recall that point later in time.

When the button to use the time-machine function is pushed, two time-travel sliders are displayed as shown in Figure 10.

- *Rewind slider*

  Moving this slider takes Musicream back through time in units of seconds relative to the present. The user can browse the operation history as if he or she was rewinding recorded (video-taped) screen images.

- *Date/time slider*

  Moving this slider specifies a particular date and time (year, month, day, hour, minute) in the past. The slider is colored only for the intervals during which Musicream was operating to make it easier to find and return to a date and time during which Musicream was being used.

The time-machine function makes it even easier to listen to music in the present after deciding on a playback order through trial and error. Because past operations can be recovered at any time, the user need not hesitate to throw away a musical piece picked out through the music-disc streaming function when thinking there might be a better piece available. While it is difficult to intentionally reencounter musical pieces from the past as they stream and fall from the taps in the present, it is easy to return to a past encounter and pick them up there.

We also provide two functions to support the recalling of the past through the time-machine function. Even if specific information, such as date/time, title, etc., corresponding to a past Musicream screen cannot be remembered, these functions enable searching to be done while recalling musical pieces listened to in the past and their screens.

- *Playback history search*

  Starting with a certain musical piece, this function searches for past screens where this piece was listened to.

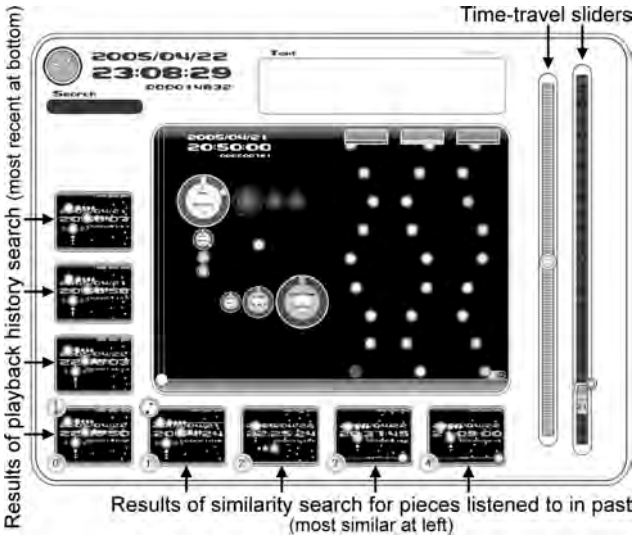


Figure 10: Musicream screen snapshot of the time-machine function: The time-travel sliders consist of a rewind slider (left) for travel back into time in units of seconds relative to the present and a date/time slider (right) for specifying particular dates and times to return to. The results of playback history search and past similarity search are displayed when a disc is clicked.

- *Past similarity search*
  From among all musical pieces listened to in the past, this function searches for past screens where musical pieces similar in mood to the starting musical piece were listened to.

The results of both of these functions are simultaneously displayed by clicking a musical piece in the time machine operation as shown in Figure 10.

3 IMPLEMENTATION

Implementing Musicream requires preprocessing of each musical piece in a music collection and implementation of the Musicream interface which is executed with sound files of musical pieces in the MPEG Audio Layer 3 (MP3) format and a music-catalog file including the results of preprocessing. Although our current implementation supports only MP3 sound files stored on a local hard disk, Musicream can be easily applied to online musical pieces and any music subscription services.

3.1 Preprocessing

A music-catalog file in XML format is generated from MP3 sound files in a music collection. The file includes title, artist name, name of the MP3 sound file, feature vector for computing music similarity, and associated disc color for each musical piece. The feature vector is extracted from each musical piece and the disc color (hue and saturation) is assigned on the basis of the feature vector.

Although we can use any feature vector designed for computing music similarity, in our current implementation we use a 30-dimensional feature vector described in [1], which is obtained by automatically analyzing the mood of each musical piece and is confirmed as effective for genre classification. It consists of the mean and variance of local spectral features (centroid, rolloff, flux, and zero-crossings) across the entire musical piece (eight dimensions); average values of mel-frequency cepstral coefficients (MFCC) across the entire musical piece (ten dimensions); portion of the musical piece occupied by low-energy intervals (one dimension); pitch content features reflecting periodicity in pitch (five dimensions); and rhythmic content features reflecting periodicity in beat (six dimensions). The feature vectors for all sound files are extracted using Tzanetakis's MARSYAS [17], a software framework for audio analysis and synthesis.

The disc color (hue and saturation) is determined from the color circle whose circumference and radius correspond to hue and saturation, respectively. Each musical piece is projected into the circle according to its feature vector. Principal component analysis (PCA) is used to reduce the dimensionality of feature vectors to a two-dimensional vector on a plane. The planar coordinates consisting of the first and second principal components are converted to polar coordinates, and then angle θ is assigned to hue and distance r from the origin is assigned to saturation.

3.2 Implementation of the Musicream interface

The Musicream interface is implemented using Macromedia Flash MX Professional 2004. It takes the music-catalog file as input and provides the four functions of Musicream.

First, for the music-disc streaming function, the hue angle θ is divided into three equal sections of 120 degrees each of which is assigned to one of the three music-supply taps. Any one tap can release only those discs having hue values in the range assigned to that tap. The number of taps can be changed by assigning angle θ in different ways. The interval of disc release can be adjusted by a slider control in a range from 3 to 10 seconds, which is determined so that discs do not overlap when streaming discs are expanded.

Next, for the similarity-based sticking function, the ease-of-sticking between two discs is determined on the basis of the hue angle θ of each disc. If the difference between θ of one disc and θ of the other lies in the range from 0 to 30 degrees, the two discs will stick on the first try. For a difference of 30 to 80, 80 to 130, and 130 and greater degrees, the two discs will stick on the second, third, and fourth tries, respectively.

For playlist sort mode, in which discs in a series are sorted in order of similarity and hung from the base disc, the degree of similarity between two discs is defined in terms of the cosine angle (scalar product) between the 30-dimensional feature vectors of those discs.

Finally, the time-machine function is achieved by continuously recording at one-second intervals the snapshot information needed to reproduce past states (all discs on the screen, each tap's internal state, date and time, etc.). Though it depends on the number of discs on the screen, a memory capacity of about 10 Mbytes is needed to save this information in XML format for one hour of Musicream use (there is a lot of room here, however, for compression).

4 EXPERIMENTAL RESULTS

We operated Musicream using a music collection that included all 315 musical pieces of the RWC Music Database [18][5] and 1572 popular songs which appeared on Japanese hit charts from 2000 to 2004. We found that the proposed interface functioned effectively and that the four Musicream functions working in combination provided an active music-listening experience that was unique. When first using Musicream, users typically tried the similarity-based sticking function after finding a musical piece that they liked in the stream of discs. After becoming more familiar with Musicream, though, instead of first determining whether they liked each musical piece they picked out, users went directly to sticking many other musical pieces to the piece they picked out and then entered sort mode where they would listen to those pieces through "zapping" and then decided which ones they liked. This was because they found that even if they did not like the first piece selected, they could still unexpectedly come across one that they liked through sticking. Furthermore, by listening to a group of musical pieces obtained through sticking, users could understand what kind of music they might like to hear at that time.

To further analyze the advantages of Musicream, we conducted a user study with 27 subjects (16 male, 11 female) who were not familiar with the music collection used here (to this end, we used musical pieces of only the RWC Music Database for this experiment). To evaluate their subjective assessment of Musicream and have the subjects gain a good command of Musicream, each subject was asked to complete a subjective questionnaire after freely using Musicream for five minutes, excluding the time for receiving brief instructions. The questionnaire results indicated that more than 81.5% of the subjects rated each of the four functions as interesting (in particular, 92.6% found the similarity-based sticking function interesting), that 92.6% of the subjects thought the music-disc streaming function was an effective way to encounter unfamiliar musical pieces such as those available through flat-rate, all-you-can-hear music subscription services, and that 96.3% of the subjects wanted to use Musicream in the future. We also found that Musicream was easy enough to use without long training.

We then had each subject use both Musicream and an ordinary music player with conventional playback buttons and a playlist editor on the same three tasks — listening through the "zapping" operation, editing playlists, and finding music — and complete a subjective questionnaire comparing their operation. The number of subjects who rated the zapping operation on Musicream as more convenient was double that who found it less convenient, while the ease of the zapping operation was rated almost equally (Musicream zapping scored slightly better). The results also indicated that 96.3% of the subjects thought that the meta-playlist function on Musicream was easy to use and convenient compared to conventional playlist editing when the playback order were rearranged several times in units of groups of musical pieces. For browsing unfamiliar musical pieces to find a musical piece having a certain mood (e.g., uplifting music), the results indicated that the number of subjects who thought Musicream was more convenient, made music easier to find, and was more enjoyable to use than an ordinary music player was, respectively, 3.6, 3.8, and 5.75 times the number who rated the ordinary music player more highly.

These results showed that Musicream was an effective way, through its four convenient functions, to enable users to browse music collections to find unfamiliar but interesting musical pieces.

[5]Note that Musicream can support operations using a much greater number of musical pieces. It should be highly suitable for all-you-can-hear music-listening environments, like flat-rate music subscription services, as well as personal music collections.

5 RELATED RESEARCH

To go beyond conventional interfaces based on bibliographic information or query-by-example retrieval, several interesting approaches for browsing music collections have been reported.

For example, Tzanetakis et al. [19] developed the "GenreSpace" interface for browsing music collections in a three-dimensional space into which musical pieces are projected according to their similarity; they also developed the "GenreGram" tool for displaying, along with real-time audio input, several up-and-down cylinders corresponding to different genres. Tzanetakis et al. [20] have also developed other interfaces, such as "Sound Sliders" which provides continuous aural feedback of retrieved pieces while a user moves sliders of music properties such as tempo and beat strength.

With an emphasis on visualization, Pampalk et al. [21] reported an interface featuring self-organizing maps (SOMs) that projects musical pieces onto a plane. They used a metaphor of "islands" that represent self-organized clusters of similar pieces. Van Gulik et al. [22] reported an "artist map" interface with the focus on artists and small devices. It enables users to explore and discover music collections on small devices by projecting artists into a two-dimensional space. Artists are drawn as dots in the space so that similar artists are placed close together on the basis of a modified spring-embedder algorithm. Torrens et al. [23] reported visualization techniques where musical pieces are placed in a circle, rectangle, or treemap by using metadata of sound files without analyzing audio signals.

When visual information related to musical pieces is available, a "collaging" technique proposed by Bainbridge et al. [24] is also an effective way to provide leisurely, undirected interaction with a music collection. In Musicream, visual information like jacket covers can also be used effectively by displaying a jacket cover image on or near each music disc when such information is available.

Although most of the above approaches share the same goal of enabling a non-specific music search to satisfy user desires like "I want to hear something", Musicream is the first interface that supports the four functions described in Section 2. In particular, the time-machine function is unique and has great potential. While the concept of time-machine computing itself was proposed for a computer desktop [25], our research is the first to discuss

time-machine computing for music. In Musicream, the ability to reproduce all operations from the past produces an even greater time-travel effect. For example, when a user returns to the past and sticks a series of similar musical pieces that were not previously selected (thereby creating a new future), the musical pieces streaming on the screen revert to those that were streaming at that time. This makes it possible to unexpectedly encounter music that was popular at that point in the past.

6 CONCLUSION

We have described a rich, integrated music playback interface called *Musicream* that enables seamless operation of four novel functions: "*music-disc streaming*," "*similarity-based sticking*," "*meta-playlist*," and "*time-machine*." The first two functions satisfy the user desire expressed as "I want to hear something" and the last two functions satisfy the desire expressed as "I want to hear something my way." The main contribution of Musicream is to provide a novel music-listening environment that enables a user to interact with a huge music collection in active, flexible, and creative ways, which go beyond traditional techniques of music information retrieval.

Although the basic concept of Musicream has great potential, we have not fully exploited it. In the future, for example, we could provide various specially designed music taps such as a tap streaming musical pieces by a specific artist (e.g., an artist name can be typed on the tap), a tap streaming music on a hit chart at a certain time in the past or currently, and a tap streaming music selected by a famous artist or celebrity. It is also interesting to share the screen (i.e., the playlist) of Musicream with friends or anonymous users via the Internet. If they use the same flat-rate music subscription service, as well as seeing the titles on the screen other users can freely listen to those pieces. For the "zapping" operation, we found it useful to color chorus ("hook") sections in the playback position slider (at the disc periphery) as done in SmartMusicKIOSK [26]: these chorus sections can be automatically detected through the RefraiD method [27].

Note that the concept of Musicream is also independent of similarity measures and dimensionality reduction techniques. We plan to use another similarity measure based on content-based or collaborative filtering and another dimensionality reduction technique such as the self-organizing map (SOM). Since Musicream is complementary to conventional music information retrieval techniques, their seamless integration should be the next goal. Future work will also include applying Musicream as an interface for commercial online music subscription services.

REFERENCES

[1] George Tzanetakis and Perry Cook. Musical genre classification of audio signals. *IEEE Trans. on Speech and Audio Proc.*, 10(5):293–302, 2002.

[2] Jean-Julien Aucouturier and Francois Pachet. Music similarity measures: What's the use? In *Proc. of ISMIR 2002*, pages 157–163, October 2002.

[3] Jouni Paulus and Anssi Klapuri. Measuring the similarity of rhythmic patterns. In *Proc. of ISMIR 2002*, pages 150–156, October 2002.

[4] Jonathan Foote, Matthew Cooper, and Unjung Nam. Audio retrieval by rhythmic similarity. In *Proc. of ISMIR 2002*, pages 265–266, October 2002.

[5] Simon Dixon, Elias Pampalk, and Gerhard Widmer. Classification of dance music by periodicity patterns. In *Proc. of ISMIR 2003*, pages 159–165, 2003.

[6] Martin F. McKinney and Jeroen Breebaart. Features for audio and music classification. In *Proc. of ISMIR 2003*, pages 151–158, 2003.

[7] Wei-Ho Tsai, Hsin-Min Wang, Dwight Rodgers, Shi-Sian Cheng, and Hung-Ming Yu. Blind clustering of popular music recordings based on singer voice characteristics. In *Proc. of ISMIR 2003*, pages 167–173, 2003.

[8] Elias Pampalk. A MATLAB toolbox to compute music similarity from audio. In *Proc. of ISMIR 2004*, pages 254–257, 2004.

[9] Upendra Shardanand and Pattie Maes. Social information filtering: Algorithms for automating "word of mouth". In *Proc. of CHI '95*, pages 210–217, 1995.

[10] William W. Cohen and Wei Fan. Web-collaborative filtering: Recommending music by crawling the web. In *Proc. of WWW9*, 2000.

[11] Alexandra Uitdenbogerd and Ron van Schyndel. A review of factors affecting music recommender success. In *Proc. of ISMIR 2002*, pages 204–208, October 2002.

[12] Masoud Alghoniemy and Ahmed H. Tewfik. A network flow model for playlist generation. In *Proc. of ICME 2001*, 2001.

[13] Steffen Pauws and Berry Eggen. PATS: Realization and user evaluation of an automatic playlist generator. In *Proc. of ISMIR 2002*, pages 222–230, October 2002.

[14] Beth Logan. Content-based playlist generation: Exploratory experiments. In *Proc. of ISMIR 2002*, pages 295–296, October 2002.

[15] Jean-Julien Aucouturier and Francois Pachet. Scaling up music playlist generation. In *Proc. of ICME 2002*, 2002.

[16] Beth Logan. Music recommendation from song sets. In *Proc. of ISMIR 2004*, pages 425–428, 2004.

[17] George Tzanetakis and Perry Cook. MARSYAS: A framework for audio analysis. *Organized Sound*, 4(3):169–175, 2000.

[18] Masataka Goto. Development of the RWC music database. In *Proc. of ICA 2004*, pages I–553–556, 2004.

[19] George Tzanetakis, Georg Essl, and Perry Cook. Automatic musical genre classification of audio signals. In *Proc. of ISMIR 2001*, pages 205–210, October 2001.

[20] George Tzanetakis, Andreye Ermolinskyi, and Perry Cook. Beyond the query-by-example paradigm: New query interfaces for music information retrieval. In *Proc. of ICMC 2002*, pages 177–183, September 2002.

[21] Elias Pampalk, Simon Dixon, and Gerhard Widmer. Exploring music collections by browsing different views. In *Proc. of ISMIR 2003*, pages 201–208, 2003.

[22] Rob van Gulik, Fabio Vignoli, and Huub van de Wetering. Mapping music in the palm of your hand, explore and discover your collection. In *Proc. of ISMIR 2004*, pages 409–414, 2004.

[23] Marc Torrens, Patrick Hertzog, and Josep-Lluis Arcos. Visualizing and exploring personal music libraries. In *Proc. of ISMIR 2004*, pages 421–424, 2004.

[24] David Bainbridge, Sally Jo Cunningham, and J. Stephen Downie. Visual collaging of music in a digital library. In *Proc. of ISMIR 2004*, pages 397–402, 2004.

[25] Jun Rekimoto. Time-machine computing: A time-centric approach for the information environment. In *Proc. of UIST '99*, pages 45–54, 1999.

[26] Masataka Goto. SmartMusicKIOSK: Music listening station with chorus-search function. In *Proc. of UIST 2003*, pages 31–40, 2003.

[27] Masataka Goto. A chorus-section detecting method for musical audio signals. In *Proc. of ICASSP 2003*, pages V–437–440, April 2003.