VizMIR: A Cross-media Music Retrieval System Supporting Bidirectional Transformation between Mood-based Color Changes and Tonal Changes in Music

Shuichi Kurabayashi and Yoshiyuki Kato
Faculty of Environment and Information Studies, Keio University
5322 Endo, Fujisawa, Kanagawa 252-0882, Japan
{kurabaya, t10247yk }@sfc.keio.ac.jp

Abstract—**VizMIR is a music retrieval system that provides an intuitive user interface to search for music based on the sentiment that it evokes. The unique feature of this system is a cross-media retrieval mechanism that accepts a sequence of images to describe user requirements for mood transitions within a musical composition. VizMIR has a hybrid metric space for converting color change in images to continuous tonal changes in music, and vice versa. When a user enters a sequence of images as a query specifying the desired changes of mood in music, VizMIR measures the color distance in the image sequence, converts the calculated distance into the distance of the movement of musical tonality, and finds music that has the same or similar tonality movement. To support this bidirectional conversion of distance, we design two metric spaces as topologically equivalent structures, and provide a bridge function that maps a distance measured in the color metric space into one in the tonality metric space. This system enables users to search music by subtly manipulating queries through trial and error, and this is easy to use because images are suited to interactive manipulation. This method is useful in searching for music unknown to the user that evinces a mood satisfying the user's preferences.**

Keywords – Cross-Media Retrieval, Emotion-Aware Search, User Interface.

I. INTRODUCTION

In this paper, we describe the *VizMIR* system [1] and its implementation framework using modern web technologies. Our system provides an intuitive user interface to formulate mood-based queries to find music suited to the user's disposition at the time. This system provides a "bridge" between music and images to enable users to search for their preferred songs by using a sequence of images representing the mood expressed by the desired music. Such cross-media retrieval methods are considered very important for designing user-centered music retrieval systems [2].

With the rapid progress of computing technologies, ever more songs are being digitized and stored in online libraries and on personal devices. Due to their proliferation, portable and personal devices, such as tablet computers and smartphones, are commonly used to listen to music. Such proliferation and diversity of digital media increases the demand for an effective music retrieval system [3]. However, it is difficult to find music satisfying our preferences at any given time because the sentiment expressed by a piece of music varies with its progression. Musical data consists of non-verbal elements that proceed along the timeline of the musical composition. The context and temporal transitions of music deeply effect listeners' emotions. In order to find a musical composition the progression of which corresponds to changes in mood desired by the user, the user typically needs to listen to several parts of a piece of music in repositories, such as online music stores and personal music players. Owing to the temporal nature of music, it is difficult to develop an effective music search environment where users can retrieve specific music samples by using intuitive queries. This is because an effective search through a temporal structure requires that the system recognize the changing features of the content in a context-dependent manner.

To retrieve music intuitively, the concept of the "impression" that a musical composition makes on the listener is of great importance. This is because studies have shown that many users consider their feelings and moods to be among the most significant factors motivating them to listen to music. However, in spite of the fact that young users tend to select music according to their disposition, they are frustrated in their attempts to find and retrieve their desired music in a given mood. This is on account of the absence of technology that allows users to enter visual queries to specify the mood of the desired music and the impression that the user wishes for it to create. In particular, in order to find recently released music, or music that may be unknown to the user that is nonetheless appropriate for his/her mood at a given time, a method to effectively communicate the user's desire for music according to a particular disposition is required [3]. Users need a toolkit that assists them to form their own queries using trial and error. Thus, an intuitive user interface (UI) to effectively communicate the demands of users for music is desirable.

With this objective in mind, we propose an impression-aware music retrieval method that offers a cross-media query model using image files as a medium to describe user demands for mood-based music. It is important to develop a stream-oriented query construction method for music because the content of music as well as the impression it creates on the listener change with time. Our query model interprets the effects of temporal changes in media features, such as tonality in music and color in images. This paper presents a prototype system that carries out web-based music

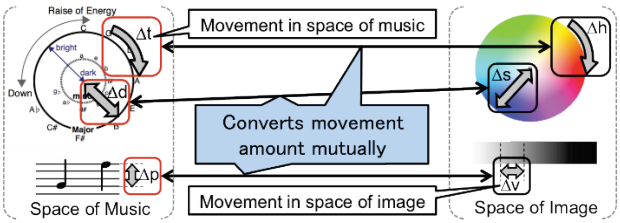

Figure 1. An overview of cross-media retrieval using the delta equation on each media data.

retrieval by considering changes in the mood evinced by the musical composition at hand.

Our design principle for this system is to make it possible for a user to search musical content invisible to him/her by using visible image content. In recent research on synesthesia in psychology, it has become clear that a significant correlation exists between color sense and musical elements [4]. Thus, we think that it is reasonable and beneficial to represent the impression created by musical compositions as sequences of colors in order to design and implement an intuitive user interface for music database systems. This concept is well suited to web-based music retrieval because the web is a visual medium. Thus, a crucial aspect of this system is a cross-media query interpretation method that recognizes how media changes with time by using two metric spaces to calculate the distance between the current and the previous states of the media content in question. For music, we implement a tonality-based metric space. In this tonality space, a song can be modeled as a trajectory in three dimensions. Our system extracts the temporal transition of tonality in a song to analyze emotive transitions occurring with time because tonality is one of the most important factors in determining the overall mood of a musical composition [5][6]. Corresponding to the tonality metric space, we have developed a metric space to compute color distance using the hue, saturation, and value (HSV) [7] color space. HSV is a widely adopted space in image and video retrieval because it describes perceptual and scalable color relationships.

The system transforms invisible changes in the impressions created by music into visible changes in color, and vice versa. Our approach converts a "delta value," which represents distance in each space, between two spaces rather than a feature value itself because the system focuses on how changes of mood in music effect human perception. The most important feature of the two metric spaces is their configuration as topologically equivalent structures (Figure 1). Each axis in the music space has a corresponding axis in the color space. Specifically, tonality is associated with hue, pitch with value, and major/minor with saturation. Thus, a specific distance in music space can be converted into the same distance in color space.

We implement a prototype of our system using HTML5 technologies. The implemented prototype assumes application to online music stores as a front-end user interface. The advantage of this system is an intuitive method for users to edit a query using trial and error depending on their evaluation of the results. The method makes it possible for users to describe changes in impressions created by music, which are difficult to represent directly, as a sequence of images through a visually enhanced user interface, wherein the order of the images represents a change of impression. Thus, our system provides a fundamental framework for implementing the UI of an online music database system.

The remainder of this paper is structured as follows. Section II presents the motivating example of our query processing. Section III briefly summarizes related work in the area. Section IV describes the fundamental concept and the system architecture, whereas we detail our prototype system implementation in Section V. Section VI discusses our feasibility studies, and we offer concluding thoughts in Section VII.

II. MOTIVATING EXAMPLES

In this section, we present two examples motivating our stream-oriented cross-media music retrieval. The first example situation assumes that a user has a lot of music in his/her portable music player and wants to listen to music suited to his/her mood, but does not have an idea of the title and the artist of the type of music in question. In this case, the VizMIR system enables the user to retrieve the desired music by describing moods represented by it using a sequence of images in a trial-and-error method. The user chooses several images from his/her collection of pictures in portable device, and the system find music that creates impressions similar to those created by the selected image sequence.

The second example situation assumes that the user is an illustrator, and wants to make a slideshow of his/her own pieces of illustration in order to seek background music for it. The user has candidates for the slideshow but has no clear idea about the exact composition of the slideshow or the musical piece to serve as background for it. In this case, the user can retrieve music related to changes in the slideshow by revising the order of the candidates, not only for the slideshow but also for forming a query.

III. RELATED WORKS

Conventional music database systems available on Internet use metadata, such as genre and artist name, as indexing keys. As such, fundamental metadata are not sufficient to retrieve music without detailed knowledge of the target data. The music information retrieval (MIR) system is a well-known means of helping users find music by using several intuitive queries [1][8]. However, such approaches cannot be applied to find music that is unknown to the user. Thus, there is a need for a retrieval mechanism for music that users have not heard before [9].

In content-based MIR methods, the system analyzes and extracts several significant features from a musical composition in order to identify equivalent or highly similar music samples in a database. There are several choices of input, such as the user profile-based approach [10], the chord-based approach [11], and the query-by-humming [12][13][14]. Content-based methods are advantageous with regard to ease of input and the ability to generate a large amount of information reflecting the musical content. As

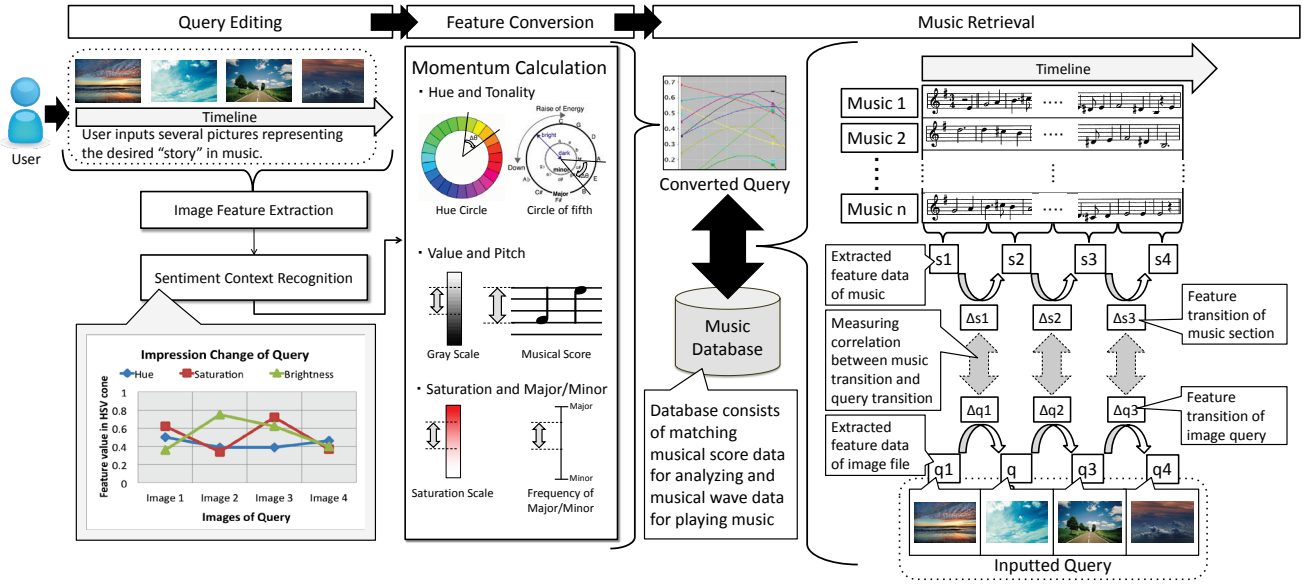

Figure 2. VizMIR system architecture that realizes a visual query construction for retrieving music by converting color changes into tonal changes.

content-based technologies are very effective in retrieving musical equivalents to queries, they are widely used for copyright protection in online music sharing services.

Music visualization systems partially help users find new and unknown music. There are several music visualization systems that utilize the cross-media relationship between color and tonality proposed in [15][16][17][18][19]. An impression-based music visualization method that utilizes the result of a synesthesia study [4] was proposed in [20]. This uses a color sense of tonality to view the harmonic structure of and relationships between important regions in a musical composition. Pampalk et al. [21] proposed an interface for discovering artists by using a ring-like structured visual UI, and Knees et al. [22] developed a method of visually summarizing the contents of music repositories. Stober et al. [23] proposed an interface that can conduct music searches based on ambiguous demands. Research in [24] presents "GlobalMusic2One," a portal site for visualizing songs using a two-dimensional similarity map for explorative browsing and target-oriented finding. Cross-modal media analysis and retrieval methods are proposed in [25][26]. These systems implement user-centered music retrieval by considering user preferences.

It is difficult to create a metric space that can measure temporal changes in heterogeneous media data. In order to design and implement a cross-media retrieval system that considers changes of moods in images and music, it is necessary to develop a new method that transmits feature value bidirectionally between images and music, instead of a knowledge base of music and colors referring to synesthesia.

The most significant difference between conventional approaches and ours is that our system focuses on the development of a method for mood-based cross-media retrieval. Conventional cross-media retrieval methods do not have the capability to detect an impression and a mood in temporal music stream. Our system analyses emotional transitions by capturing the progression of tonality as a function of "how the music sounds." Further, our system allows users to describe their music demands by using an image sequence. Our system is unique in supporting such a cross-media query interpreting mechanism for continuous multimedia data.

IV. System Architecture

Figure 2 shows the fundamental system architecture of VizMIR. The core components of VizMIR constitute four data structures and three functions. The system models the concept of "change in sentiment" by measuring the distance between successive temporal transitions in the media data. Our method succeeds in cross-media retrieval by comparing the results of continual sentiment analysis of music and images. This system provides two delta functions to analyze temporal changes in the media data and to generate sequential values representing changes of mood in them. The system calculates the sentiment-oriented relevance score of music and images by comparing the calculated delta values.

We show an example query in Figure 3. This query represents changes in impression as follows: the brightness (value in the HSV color model) gradually increases and then decreases; the hue (type of color) gradually changes from blue to red; and the saturation (vividness of colors) drastically increases in the middle of the query. The system translates these features of the query into musical features as follows: the pitch gradually increases and then decreases, the tonality gradually changes, and the major/minor changes drastically. The system retrieves music by calculating the relevance score of the query translated into music.

A. Architectural Overview

As shown in Figure 2, the system consists of three main components: 1) a query editor, 2) a feature conversion module, and 3) a retrieval engine.

The query editor is the front-end module of the system. This module provides a set of operations to prepare and

modify image files as a query. For example, the system implements an image-editing operator equipped with several color filters to change the overall impression of the image.

The feature conversion module provides fundamental data conversion functions applied to musical instrument digital interface (MIDI) data and bitmap image data. The feature conversion module generates metadata vectors from the image and MIDI files.

The retrieval engine calculates how the media data change with time by applying distance functions to the generated metadata. The system provides a bridging mechanism between the musical tonality metric space and the HSV color metric space. The bridge converts a distance calculated in the music space into a distance in the image space in order to retain the impression factor from one medium to the other. For example, in the distance conversion mechanism, hue, which is a type of color, corresponds to tonality, which is a type of musical structure. By converting distances between heterogeneous metric spaces, the system realizes cross-media retrieval for stream media such as music. Finally, the retrieval engine compares the set of distance values to calculate the relevance of the music to the image query. In the following sections, we describe in detail the fundamental data structures and functions involved.

B. Image Sequence as a Query

VizMIR accepts an image sequence as a query that represents how the media data changes in mood with time. An image sequence object Q is defined as follows:

$$Q := \langle\langle h_1, s_1, v_1\rangle \ldots \langle h_i, s_i, v_i\rangle\rangle \quad (1)$$

where h_i is the hue data, s_i is the saturation data, and v_i is brightness data in the i-th image of the query. The system converts RGB color values of each image into HSV triples at the pixel level. We adopt the following well-known RGB-to-HSV conversion equation.

$$\mathrm{V} = \max(\mathrm{R}, \mathrm{G}, \mathrm{B}) \quad (2)$$

$$\mathrm{S} = 255 \times \frac{\max(\mathrm{R}, \mathrm{G}, \mathrm{B}) - \min(\mathrm{R}, \mathrm{G}, \mathrm{B})}{\max(\mathrm{R}, \mathrm{G}, \mathrm{B})} \quad (3)$$

$$\mathrm{H} = \begin{cases} 60\dfrac{B-G}{\max(\mathrm{R},\mathrm{G},\mathrm{B}) - \min(\mathrm{R},\mathrm{G},\mathrm{B})} & R == \max(\mathrm{R},\mathrm{G},\mathrm{B}) \\ 60\left(2 + \dfrac{R-B}{\max(\mathrm{R},\mathrm{G},\mathrm{B}) - \min(\mathrm{R},\mathrm{G},\mathrm{B})}\right) & G == \max(\mathrm{R},\mathrm{G},\mathrm{B}) \\ 60\left(4 + \dfrac{G-R}{\max(\mathrm{R},\mathrm{G},\mathrm{B}) - \min(\mathrm{R},\mathrm{G},\mathrm{B})}\right) & B == \max(\mathrm{R},\mathrm{G},\mathrm{B}) \end{cases} \quad (4)$$

We define the metric space of images as the HSV color metric space with three axes: hue, saturation, and value. These three elements are significant factors affecting the impression of the image. Hue represents the differences of color phases, such as red, yellow, green, and blue. In the HSV cone of images, the hue is represented by an angle. The system converts the extracted hue angle in the HSV cone of images into a hue scalar h, which is a value between 0 and 1. Saturation is the vividness of color. In our system, saturation is an average value of the vividness in an image. The system processes this vividness value of an image into a saturation scalar s, which is a value between 0 and 1. Here, 0 and 1 represent the lowest and the highest value, respectively. Value is the brightness of color. Our system calculates the value (brightness) as an average value of color brightness in an image, processes the brightness value of an image into a brightness scalar v, which is also between 0 and 1.

Figure 3. An example impression query consisting of four images

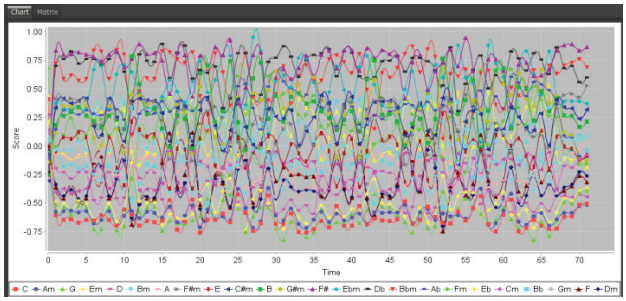

Figure 4. A visualization of tonality change in one music item. The tonality changes with time.

C. MIDI Song Data

Our system uses standardized MIDI data format as the primary data format for storing music. MIDI stores note-on signals and corresponding note-off signals sequentially because it was developed in order to automate keyboard instruments. The system represents a MIDI file F:={$n_1(t, p, d)$, n_2, …, n_k}, where n_i represents the i-th note whose attributes are 1) t: the start time of the note, 2) p: the pitch of the note, and 3) d: the duration of the note. F is a sequential set of k-tuple data.

Our system provides a matrix structure that represents the continuous variation in and distribution of pitch in the target music data. We call the data structure a music pitch matrix. The pitch matrix is a 128 × n matrix, which is given as the data matrix. MIDI specifications define the domain of pitch value between 0 and 127. A musical composition is expressed as a set of m timelines. Each timeline is characterized by a note on information for 0 to 127 pitch level. When the 12th note is on the m-th section, $c_{[12,m]}$ is 1. The pitch matrix P is defined as follows:

$$\mathrm{P} := \begin{pmatrix} c_{[0,0]} & \cdots & c_{[0,n]} \\ \vdots & \ddots & \vdots \\ c_{[m,0]} & \cdots & c_{[m,n]} \end{pmatrix} \quad (5)$$

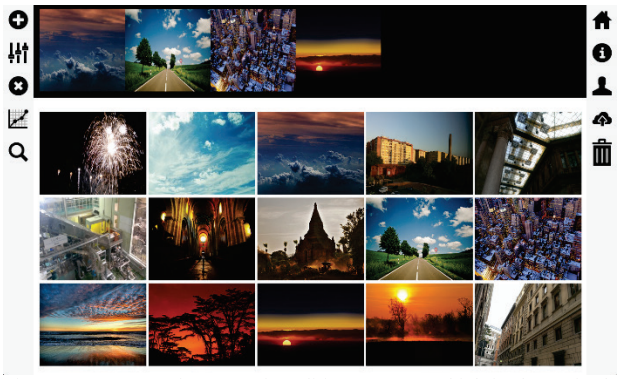

Figure 5. An example screen for editing a query, which is shown in the black area at the top, by utilizing photo stocks shown at the bottom.

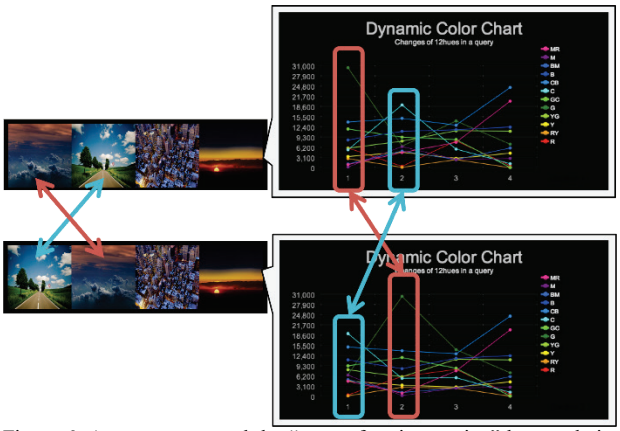

Figure 6. A user can control the "story of an impression" by reordering images. In this case, the impression in the blue rectangle is moved to the head of the story.

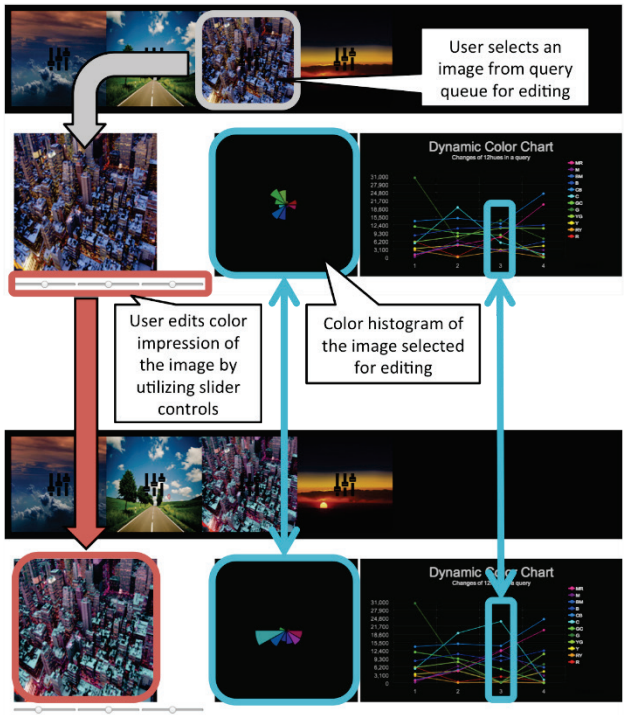

Figure 7. A user can directly apply color filters to images for controlling the impression in a fine-grained way.

where $c_{[i,j]}$ denotes the j-th pitch status at the i-th time duration. We have implemented the MIDI analysis modules for converting MIDI into a musical score-like data structures by using our MediaMatrix system [27], a stream-oriented database management system.

D. Music Data

When VizMIR receives a query consisting of several images, the system divides every song in its database into sections, such that the number of sections is equal to the number of images entered as image query. A music object M is defined as follows:

$$M := \langle\langle t_1, d_1, p_1\rangle \ldots \langle t_i, d_i, p_i\rangle\rangle \quad (6)$$

where t_i is the tonality data, d_i is the deviation data, and p_i is the pitch data, in the i-th section of the music. We define the metric space of the music using three axes: tonality, pitch, and major/minor. These three elements are significant factors effecting the impression of music.

Tonality is the structure of music and is composed of sequential musical notes. There are 24 tones consisting of 12 major and 12 minor tones. As shown in Figure 3, tonality in music changes with time and this causes changes in the impression of the music. Figure 3 shows movements of 24 types of tonality in a song using our tonality analysis function implemented in [27]. In musical theory, a circle of fifths defines the difference or similarity between each pair of the 24 tonalities [5][6]. Each tonality can be represented by an angular value on the circle of fifths. The system processes this angular value into a tonality scalar t, which is a value between 0 and 1. Thus, the system converts the distance measured as the angle of the hue of the image into a scalar quantity representing the angle of the tonality.

Major/minor refers to the deviation of tonality within a music section. The system calculates the deviation of tonality in a music section, and converts the deviation value into a major/minor scalar d. This has a value between 0 and 1, representing the maximum minor deviation and the maximum major deviation, respectively.

Pitch is a value of pitch in a musical score. The system calculates the average of the pitches in a music section, and converts each of the average values into a scalar quantity p. Values of p also fall between 0 and 1, which quantities represent the lowest and the highest pitch, respectively.

E. Image Distance Calculation Function

We design the following three functions in order to calculate a distance in an image sequence query Q:

- The distance in hue between the i-th image and the (i+1)-th image is $\Delta_{hi} := |h_i - h_{i+1}|$, where h is the hue angle in the HSV cone.
- The distance in saturation between the i-th image and the (i+1)-th image is $\Delta_{si} := |s_i - s_{i+1}|$, where s is the saturation coordinate in the HSV cone.
- The distance in value between the i-th image and the (i+1)-th image is $\Delta_{vi} := |v_i - v_{i+1}|$, where v is the value coordinate in the HSV cone.

F. Tonality Distance Calculation Function

We design the following three functions in order to calculate a distance in a music object *M*:

- The distance in tonality between the *i*-th section and the (*i*+1)-th section is $\Delta_{ti} := |t_i - t_{i+1}|$, where *t* is the tonality angle in the circle of fifths.
- The distance in tonality deviation between the *i*-th section and the (*i*+1)-th section is $\Delta_{di} := |d_i - d_{i+1}|$, where *d* is the deviation in tonality.
- The distance in pitch between the *i*-th section and the (*i*+1)-th section is $\Delta_{pi} := |p_i - p_{i+1}|$, where *p* is the pitch.

G. Cross-Media Relevance Calculation Function

The system provides a function to calculate the relevance of the music to the query. The function is defined as follows: $f(a,b) \rightarrow 1 - |a - b|$, where *a* and *b* form a pair of distance changes according to the dual-metrics relation. The system calculates a correlation value for each pair of metrics using this function. The relevance of the music to the image query is represented as follows:

$$\gamma(\Delta q, \Delta m) := \frac{\sum_{i=1}^{n} \frac{s(\Delta_{hi}, \Delta_{ti}) + s(\Delta_{hi}, \Delta_{ti}) + s(\Delta_{hi}, \Delta_{ti})}{3}}{n} \quad (7)$$

where *n* is the number of images entered as part of the image sequence object *M*, as well as the number of divided music sections.

V. Prototype System Implementation

We have implemented a prototype of the proposed system. The screenshots of the prototype, which uses HTML5 Canvas and JavaScript, are shown in Figure 5 – Figure 8. The system implemented consists of three modules: the query editor, the feature conversion module, and the music retrieval engine.

The query editor is the main user interface shown in Figure 5. Users can form a query by selecting four images and by revising the appearance of the query. In Figure 5, a user has selected four images as elements of a query. The system provides two ways for the user to edit the query:

- The first manner is to edit the entire impression of the query by revising the order of the images as shown in Figure 6. The system allows the user to intuitively perform this using a drag-and-drop operation.
- The second manner is to edit the partial impression of the query by changing the color of an image, as shown in Figure 7. The system allows the user to do this by moving the three sliders associated with each of the HSV values.

When the user finishes editing the query, he/she submits the images for the music search. The feature conversion module extracts the semantic color movement of the submitted images and generates a query consisting of a virtual musical feature in order to retrieve music. Finally, the music retrieval engine calculates the relevance of the candidate music with

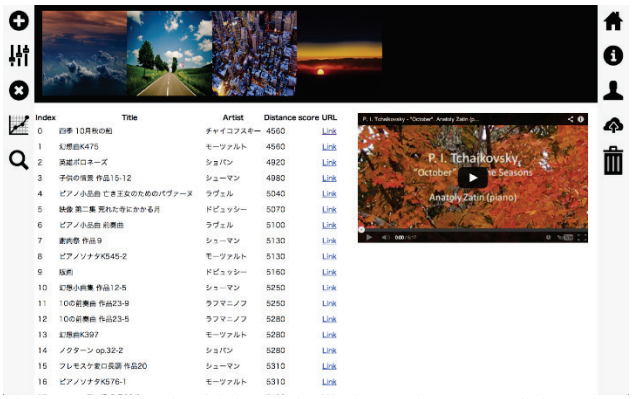

Figure 8. A screenshot of the result set view. When a user clicks an item in the result set, the system opens a video playback screen and plays a video corresponding to the selected MIDI data.

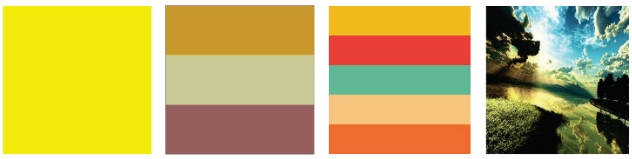

Figure 9. Examples of query components.

the generated query, and shows indexed music as a result according to the calculated relevance score, as shown in Figure 8. If the result does not satisfy the preferences of the user, the user revises the query using the query editor. By repeating the above process, the user searches music through trial and error.

The system has a backend MIDI analyzer. When a user enters a query, the system invokes the MIDI file analyzer with a section number parameter. The system analyzes the MIDI files in an on-the-fly manner in order to extract the tonality transition according to the section number parameter passed as a component of a query. We have implemented the MIDI file analyzer using HTML5 FileReader object and ArrayBuffer object. When finished with the analysis process, the MIDI file analyzer encodes the analysis result into JavaScript Object Notation (JSON) format and passes it to the distance calculation module. This procedure allows our system to share the JSON-encoded figure among multiple web workers to parallelize the distance calculation.

The relevance calculation module compares the queries and the database contents. This retrieval process is parallelized by the web workers application programming interface (API), and the retrieved songs are presented to the user through the search result visualization engine. This system spawns real operating system (OS)-level threads from the web workers API to parallelize the retrieval process. Modern HTML5 technologies enable us to implement complex processes in web browsers. Figure 8 shows a screenshot of an example of the result set and its preview screen. When a user clicks an item in the retrieval results, the system opens a video playback screen and plays a video corresponding to the selected MIDI data. In this case, our system assumes that the MIDI data is a fundamental metadata for a song. Thus, our system stores both a MIDI

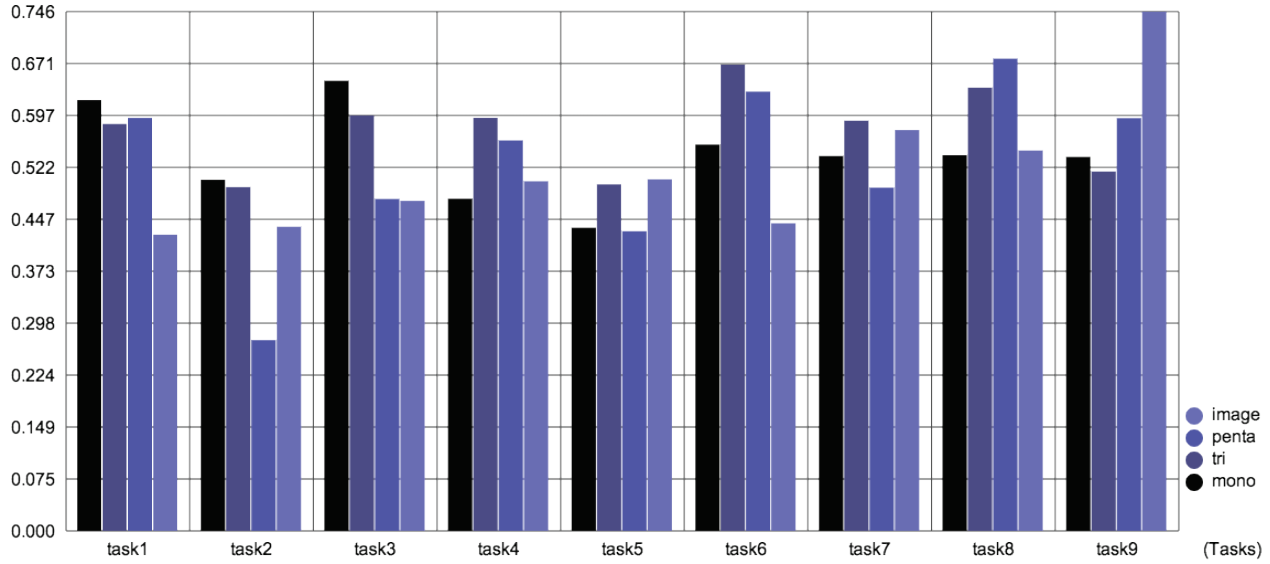

Figure 10. NDCG scores calculated by summing up all subjects, including novices and experts. Simple queries such as mono-color and tri-color images are suitable for simple tasks, and complex queries such as penta-color images and natural images have achieved better results in complex tasks.

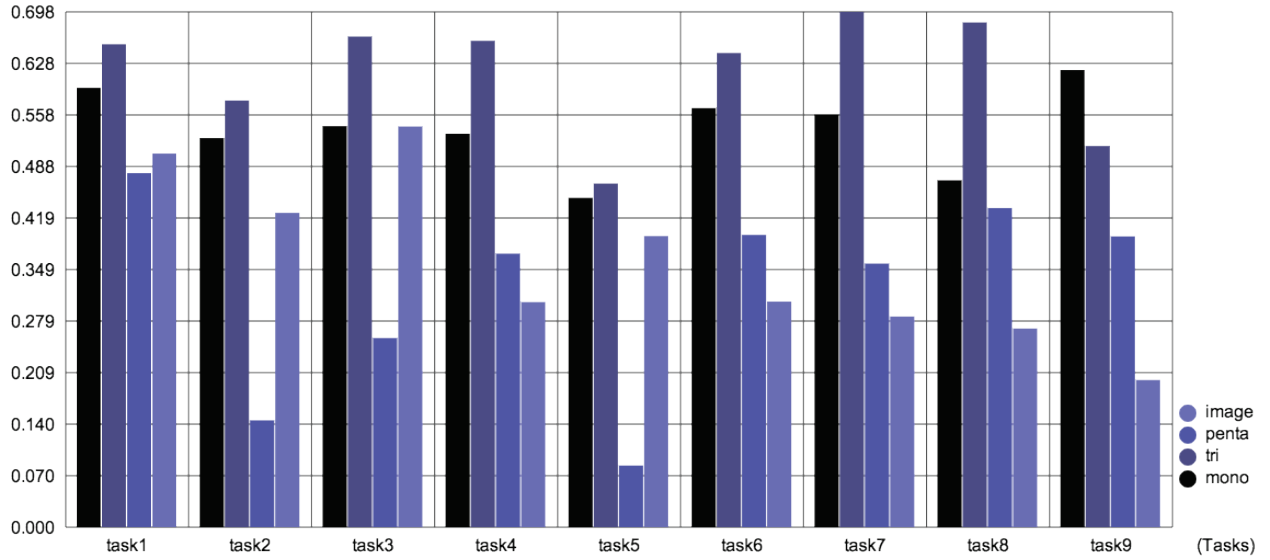

Figure 11. NDCG scores of nine tasks performed by novices. Novices achieved better scores when they used simple queries, such as ones involving mono-color and tri-color images.

file and a corresponding raw media file, such as a video file and an audio file.

VI. EXPERIMENTAL STUDIES

This section details several experiments to evaluate the effectiveness of our VizMIR system when applied to existing 100 western classic music. We prepared for our subjects nine music search tasks that are categorized into three difficulty levels: EASY, NORMAL, and HARD. Each level contains three tasks. The tasks are shown in Figure 9. Table I shows the details of the impression transition of each task in increasing order of difficulty. As can be seen, task 4 is more difficult than task 1, and task 9 is more difficult than task 4. The tasks represent the required impression from the desired music using a combination of 10 kinds of feature words associated with tonality. Subjects form queries by selecting four images according to the requirements of the task and situation at hand. Task 1 requires that the subjects find music that consists of “soft” impressions followed by “gorgeous” impressions. “Soft(C, Db)” means that the impression “soft” corresponds to tonalities “C” and “Db”. In this table, “b” denotes “flat”, and “m” denotes minor tonality.

We asked 24 subjects (13 female and 11 male) to search for music through our system by using the following types of queries:

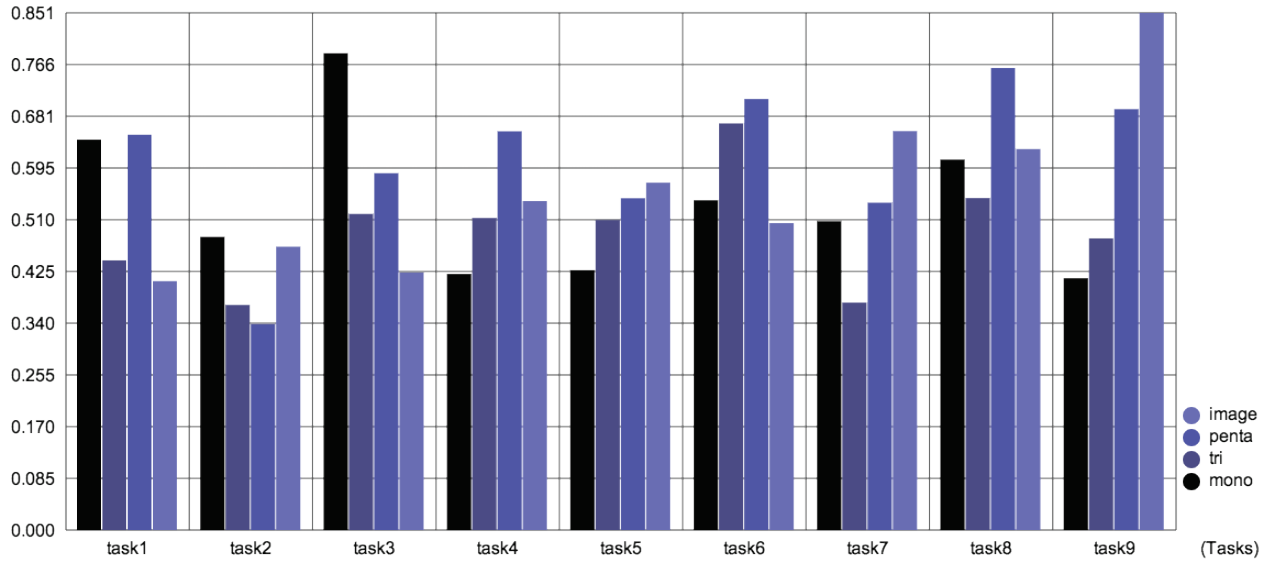

Figure 12. NDCG scores of nine tasks performed by experts. In the case of expert users, queries using full color and penta-color images are more effective when applied to more complex tasks.

- "image" query: the subjects construct a query by using full-color pictures.
- "penta" query: the subjects construct a query by using five-color images.
- "tri" query: the subjects construct a query by using three-color images.
- "mono" query: the subjects construct a query by using a single color image.

TABLE I. THREE-LEVEL SEARCH TASKS

Level	ID	Impression Transition
EASY	1	Soft(C, Db) → Gorgeous(D, Eb)
	2	Mysterious(F#m, Gm) → Solitary(Abm, Am, Abm)
	3	Calm(G, Ab, A) → Plaintive(Bm, Cm)
NORMAL	4	Calm(G, Ab, A) → Powerful(C, Db) → Gorgeous(D, Eb)
	5	Solitary(Abm, Am, Abm) → Sorrowful(Em, Fm) → Weird(Dbm, Dm, Ebm)
	6	Mysterious(F#m, Gm) → Calm(G, Ab, A) → Fresh(B, Bb)
HARD	7	Fresh(B, Bb) → Powerful(C, Db) → Gorgeous(D, Eb) → Soft(C, Db)
	8	Plaintive(Bm, Cm) → Solitary(Abm, Am, Abm) → Sorrowful(Em, Fm) → Plaintive(Bm, Cm)
	9	Soft(C, Db) → Solitary(Abm, Am, Abm) → Calm(G, Ab, A) → Mysterious(F#m, Gm)

Example images used in this experiment are shown in Figure 9. Thus, the subjects translate search tasks into image sequences while satisfying the above constraints.

To evaluate this experiment, we computed the normalized discounted cumulative gain (NDCG) as follows:

$$DCG = \sum_{i=1}^{5} \frac{rel_i}{log_2 i} \quad (8)$$

$$IDCG = \sum_{i=1}^{5} \frac{rel'_i}{log_2 i} \quad (9)$$

$$NDCG = \frac{DCG}{IDCG} \quad (10)$$

where rel_i is the average of survey scores from the test subjects in order of calculated distance, and rel'_i represents the average of the scores in descending order. We have created the correct set by comparing the tonality description in Table I with the automatically extracted tonality metadata from the data set containing 100 music items. Figure 10, Figure 11, and Figure 12 show the NDCG of our system when applied to tasks 1–9. A higher score implies a better retrieval precision. We divided the results according to the musical background of the subjects: novices who have never played music, and experts skilled at playing music.

Figure 10 shows the NDCG of all results of music retrieval for the nine tasks. Overall, simple queries such as those involving mono-color and tri-color images are suitable for simple tasks, and complex queries such as those involving penta-color images and natural images achieved better results in complex tasks. Figure 11 shows the NDCG of music retrieval by novices. It appears that novices achieved better scores when they used simple queries, such as ones using mono-color and tri-color images. Figure 12 shows the NDCG of music retrieval by experts. For these users, full color images and penta-color queries are more effective when applied to more complex tasks. The most important result is the difference in distribution of NDCG scores between the novices and the experts. Novices retrieved music effectively by using three-color images, regardless of the complexity of the task. On the other hand, as shown in Figure 12, experts retrieved music effectively by using full color images when the complexity of a task was

high (e.g., single color images are suitable for a simple task such as task 3, and full color images are suitable for a complex task such as task 9).

These results imply that skill at playing music effects cross-modal sensibility for images and music. Experts can use their musical sensibilities to form queries by using images, and novices find it difficult to detect musical impressions as images.

VII. CONCLUSION

In this paper, we proposed the *VizMIR* system, a cross-media retrieval system for music that can provide an intuitive visual retrieval method. The unique feature of this system lies in its construction of image-based queries to represent the transition in mood within a musical composition. VizMIR has a hybrid metric space for converting color change of images to continuous tonal change of music, and vice versa. We implemented the prototype system by utilizing HTML5 technologies. This implementation system supports the on-the-fly image uploading and configuring in order to create a query. We performed an evaluation of our system using a database of classical music. Experimental results showed that our visually-enriched query model performs well in practice. In the future, we plan to develop a personalized query interpretation and a social-network-based query recommendation system by building on this approach.

REFERENCES

[1] Kato, Y., and Kurabayashi, S., "Cross-media retrieval for music by analyzing changes of mood with delta function for detecting impressive behaviours," In Proceedings of the Eighth International Conference on Internet and Web Applications and Services (ICIW 2013), pp. 236-239, 2013.

[2] Liem, C., Esk, D., and Tzanetskis, G., "The need for music information retrieval with user-centered and multimodal strategies," In Proc. Of the 1st International ACM Workshop on Music Information Retrieval with User-Centered and Multimodal Strategies (MIRUM), pp. 1-6, 2011.

[3] Goto, M., and Hirata, K., "Recent studies on music information processing," Acoust. Sci. Technol., vol. 25, no. 6, pp. 419-425, 2004.

[4] Peacock, K., "Synesthetic perception: alexander scriabin's color hearing," Music Percep. vol. 2, no. 4, pp. 483–506, 1985.

[5] Temperley, D., "Music and probability," Cambridge, MA: MIT Press, 2007.

[6] Krumhansl, C. L., "Cognitive foundations of musical pitch," New York, NY: Oxford Univ. Press, 1990.

[7] Smith, A. R., "Color gamut transform pairs," In Proc. of the 5th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '78), pp.12-19, 1978.

[8] Typke, R., Wiering, F., and Veltkamp, R., "A survey of music information retrieval systems," In Proc. the 6th International Conference on Music Information Retrieval (ISMIR 2005), Univ. of London, 2005, pp. 153-160.

[9] Kuo, F. F. and Shan, M. K., "Looking for new, not known music only: music retrieval by melody style," In Proc. Of the 4th ACM/IEEE-CS Joint Conf. Digital Libraries, (JCDL '04), ACM Press, 2004, pp. 243-251.

[10] Hijikata, Y., Iwahama, K., and Nishida, S., "Content-based music filtering system with editable user profile," In Proc. of the 2006 ACM Symposium on Applied Computing, pp. 1050-1057, 2006.

[11] Bello, J.P., "Audio-based cover song retrieval using approximate chord sequences: testing shifts, gaps, swaps and beats," In Proc. the 8th International Conference on Music Information Retrieval (ISMIR 2007), pp. 239-244, 2007.

[12] Ghias, A., Logan, J., Chamberlin, D., and Smith, B.C., "Query by humming: musical information retrieval in an audio database," In Proc. ACM Multimedia 95, pp. 231–236, 1995.

[13] Dannenberg, R.B., Birmingham, W.P., Tzanetakis, G., Meek, C., Hu, N., and Pardo, B., "The MUSART Testbed for Query-by-Humming Evaluation," In Proc. of the 4th international conference on music information retrieval (ISMIR 2003), pp. 34-48, 2003.

[14] Shifrin, J., Pardo, B., Meek, C., and Birmingham, W., "HMM-based musical query retrieval," In Proc. of the 2nd ACM/IEEE-CS joint conference on digital libraries (JCDL 2002), pp. 295- 300, 2002.

[15] Craig, S., "Harmonic visualizations of tonal music," In Proc. of the International Computer Music Conference (ICMC 2001), MPublishing, University of Michigan Library, pp. 423-430, 2001.

[16] Gómez, E. and Bonada, J., "Tonality visualization of polyphonic audio," In Proc. of the International Computer Music Conference (ICMC 2005), MPublishing, University of Michigan Library, 2005.

[17] Mardirossian, A. and Chew, E., "Visualizing music: tonal progressions and distributions," In Proc. of the 8th International Conference on Music Information Retrieval (ISMIR2007), pp. 189-194, 2007.

[18] Ciuha, P., Klemenc, B., and Solina, F., "Visualization of concurrent tones in music with colours", n Proc. of the 18th International Conference on Multimedia 2010 (MM '10), pp. 1677- 1680, ACM, 2010.

[19] Cooper, M., Foote, J., Pampalk, E., Tzanetakis, G., "Visualization in audio-based music information retrieval," Computer Music Journal, Vol. 30, No. 2, pp. 42-62, MIT Press, 2006.

[20] Imai, S., Kurabayashi, S., and Kiyoki, Y., "A music database System with Content analysis and visualization mechanisms," In Proc. of the IASTED International Symposium on Distributed and Intelligent Multimedia Systems, ACTA Press, pp. 455-460, 2008.

[21] Pampalk, E. and Goto, M., "Musicrainbow: a new user interface to discover artists using audio-based similarity and web-based labeling," In Proc. of the 7th International Conference on Music Information Retrieval (ISMIR 2006), pp. 367-370, 2006.

[22] Knees, P., Schedl, M., Pohle, T., and Widmer, G., "An innovative three-dimensional user interface for exploring music collections enriched with meta-information from the web", In Proc. of the 14th ACM International Conference on Multimedia (MM '06), pp. 17-24, 2006.

[23] Stober, S. and Nürnberger, A., "MusicGalaxy: A multi-focus zoomable interface for multi-facet exploration of music collections", In Proc. of the 7th International Symposium on Computer Music Modeling and Retrieval (CMMR 2010), pp. 259–272, Springer, 2010.

[24] Dittmar, C., Großmann, H., Cano, E., Grollmisch, S., Lukashevich, H., and Abeßer, J., "Songs2See and GlobalMusic2One: two applied research projects in music information retrieval at Fraunhofer IDMT," In Proc. of the 7th International Symposium on Computer Music Modeling and Retrieval (CMMR 2010), pp. 259–272, Springer, 2010.

[25] Mao, X. and Lin, B., "Parallel field alignment for cross media retrieval categories and subject descriptors," In Proc. of the 21st ACM Conference on Multimedia (MM '13), pp. 897-906, ACM Press, 2013.

[26] Zhang, H., Zhuang, Y., and Wu, F., "Cross-modal correlation learning for clustering on image-audio dataset," In Proc. of the 15th ACM Conference on Multimedia (MM '07), pp. 273-276, 2007.

[27] Kurabayashi, S. and Kiyoki, Y., "MediaMatrix: a video stream retrieval system with mechanisms for mining contexts of query examples", In Proc. of the 15th International Conference on Database Systems for Advanced Applications (DASFAA2010), pp. 452-455, 2010.