Chord-Cube: Music Visualization and Navigation System with an Emotion-Aware Metric Space for Temporal Chord Progression

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

Tatsuki Imai
Faculty of Environment and Information Studies
Keio University
5322 Endo, Fujisawa, Kanagawa 252-0882, Japan
t10109ti@sfc.keio.ac.jp

Abstract—**In this paper, we propose an interactive music search-and-navigation system, called Chord-Cube, which visualizes musical similarities on the basis of temporal chord progression. Our proposed system offers an interactive navigation mechanism that allows users to find their desired music intuitively, by visualizing music items in a three-dimensional (3D) space. Each axis in this 3D space corresponds to three types of chord progression phases: Introductive-melody, Continued-melody, and Bridge, which are typical structures in pop and rock music. Users can utilize this 3D space to find their desired song by placing their favorite song at the point of origin and obtaining the semantic distance between the input song and other songs. We have conducted two experimental studies, in which we compared our proposed navigation system with the conventional manual trial-and-error manner, to evaluate the extent to which our visual navigation method improves music retrieval. The results of experiments show that our visual navigation method successfully increases the retrieval performance for pop and rock music.**

Keywords—Music, Visualization, Navigation, 3D, Database.

I. Introduction

In this paper, we propose an interactive music search-and-navigation system called the Chord-Cube system and its implementation using modern Web technologies. The Chord-Cube system was originally proposed in [1], and this paper is an extended version of the paper [1]. Our system utilizes traditional music theory including tonality and chord progression, which determines the impression of music, to interpret user's feelings about the music. It employs chord progression in songs as a fundamental feature for calculating similarities among music items because we consider chord progression as one of the most important factors in determining the overall mood of a song. By leveraging this music theory knowledge, we develop an intuitive music navigation method to find the desired music by visualizing music-to-music relationships from the viewpoint of temporal similarities in chord progression. Our proposed system provides an interactive 3D cube that visualizes the relative distance among music items by calculating their similarities.

Music has traditionally been regarded as one of mankind's most important forms of cultural heritage. Concomitant with the rapid advances in computing technologies, many songs are being digitized and stored in online libraries and on personal devices. The proliferation of portable and personal devices such as tablet computers and smartphones has resulted in them being frequently used to listen to music. This proliferation and diversity of digital media has increased the demand for effective music retrieval systems [2]. By enhancing the retrieval capability of music, we believe a wider scope can be provided for the sharing of human cultures.

The change in emotion in a song over time is one of the most important factors in selecting music to be played on modern mobile music players and smartphones. Young people, in particular, select music in accordance with their location and mood. To support such intuitive and emotion-oriented music selection, a player that can utilize smart content analysis to extract the movements of musical elements that have profound effects on human perception is needed.

However, current music database systems implemented in online music stores such as the iTunes Music Store and Sony's Music Unlimited do not support such perception-oriented retrieval methods. Consequently, because users often store thousands of music files in the cloud, it is difficult for them to locate their desired songs intuitively, even if they know the details of the desired music. Owing to the temporal nature of music, developing an effective music search environment, in which users can retrieve specific music samples using intuitive queries, is difficult because in order to search a temporal structure, the system has to recognize the changing features of the contents in a context-dependent manner.

Interactive and visual-oriented search mechanisms that do not use text-based search methods are promising because users often memorize music contents with a spatial metaphor. However, a music information retrieval (MIR) method that can reflect the emotions being felt by users as they listen to the music is needed. Such a retrieval environment must have an interactive and navigational user interface that can visualize context-dependent relationships between songs dynamically and in accordance with the user's viewpoint. For this purpose, we have developed Chord-Cube, which visualizes musical similarities calculated by considering emotive movements in temporal chord progression. Whereas traditional MIR systems [3] focus on finding the most relevant song or similar songs by computing similarities or relevance according to extracted features, our proposed system focuses on providing an integrated toolkit for comparing songs in order to create a visualization of implicit interrelationships on the basis of emotional characteristics.

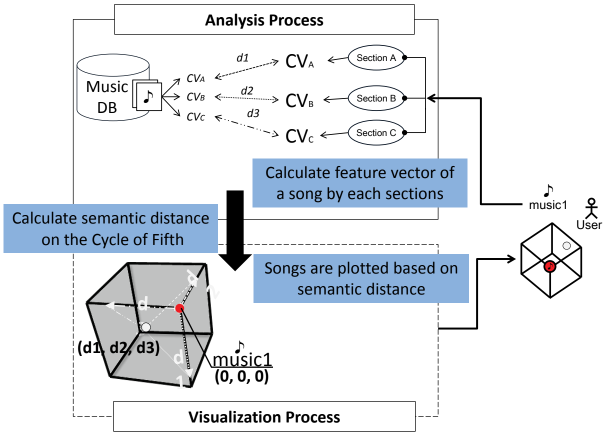

Figure 1. Overview of the Chord-Cube system for visualizing tonality-based distances of songs and navigating users to retrieve their desired song using a query song.

A unique feature of our proposed system is its "chord-vector space," in which the distance between musical chords can be calculated by analyzing the impressive behaviors of chord progression. Our system uses distances, which are calculated in the chord-vector space, to represent the degree of similarity among songs. As shown in Figure 1, each dimension of this graphical space corresponds to a degree of similarity of chords within three respective sets of song section types: "introductive-melody," "continued-melody," and "bridge-melody." Introductive-melody is a beginning of the musical piece. Continued-melody is an interlude between the introduction and bridge section. A bridge is a representative section that expresses the salient feature, and it is one of the most impressive sections in a musical composition.

This cube is a three-dimensional object inside which songs are displayed as points. Our cube accepts an initial song as a point of origin in the cube. Users can choose any song as their point of origin. The system then plots other songs inside the cube by reflecting the distance between each song and the song at the point of origin.

We implemented a prototype of our system utilizing modern HTML5 technologies. The implemented prototype system assumes that it will be applied to the online music store as a front-end user interface. It utilizes WebGL, which is a standardized API for rendering interactive 3D graphics within web browsers without the use of any plug-ins. In the system, a dynamic distance calculation method that applies chord progression data is implemented in JavaScript. Thus, this system provides a fundamental framework for implementing the user interface (UI) of an online music database system.

The remainder of this paper is organized as follows. Section II discusses related research. Section III presents several motivating examples that demonstrate how our system can be utilized to retrieve unknown songs. Section IV gives an architectural overview of the system, Section V demonstrates its fundamental data structures, Section VI defines its core functions, and Section VII outlines its prototype implementation. Section VIII discusses our feasibility studies conducted. Finally, Section IX concludes this paper.

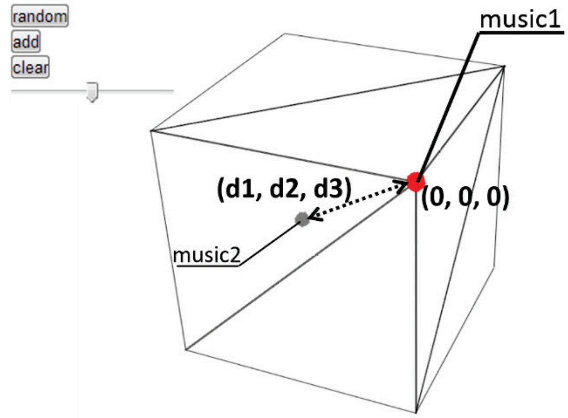

Figure 2. Overview of Chord-Cube visualization.

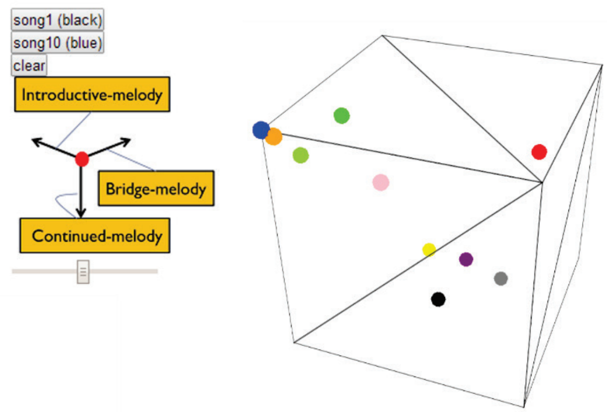

Figure 3. Example of Chord-Cube visualization. Each colored sphere represents a song. A user can operate this cube from any desired perspective.

II. RELATED WORK

In this paper, we present a system architecture that aims to improve the effectiveness of music retrieval approaches by visualizing multi-aspect similarities among songs. Conventional music database systems that are available on the Internet utilize metadata, such as genre and artist name, as indexing keys. However, such fundamental metadata are not sufficient to retrieve music without detailed knowledge of the target data. Consequently, content-based retrieval and advanced query interpretation methods have been developed to find music. In content-based music retrieval methods, a user inputs a raw music file as a query that the system analyzes and extracts several significant features from in order to identify equivalent or highly similar music samples in a database. As an example of the content-based music retrieval method, there are several input materials such as humming [4][5][6] and chords [7][8]. The content-based method has advantages in terms of ease of input and the ability to generate a large amount of information reflecting musical content. Because content-based technologies are very effective in retrieving musical equivalents to input queries, they are widely used for copyright protection in online music sharing services.

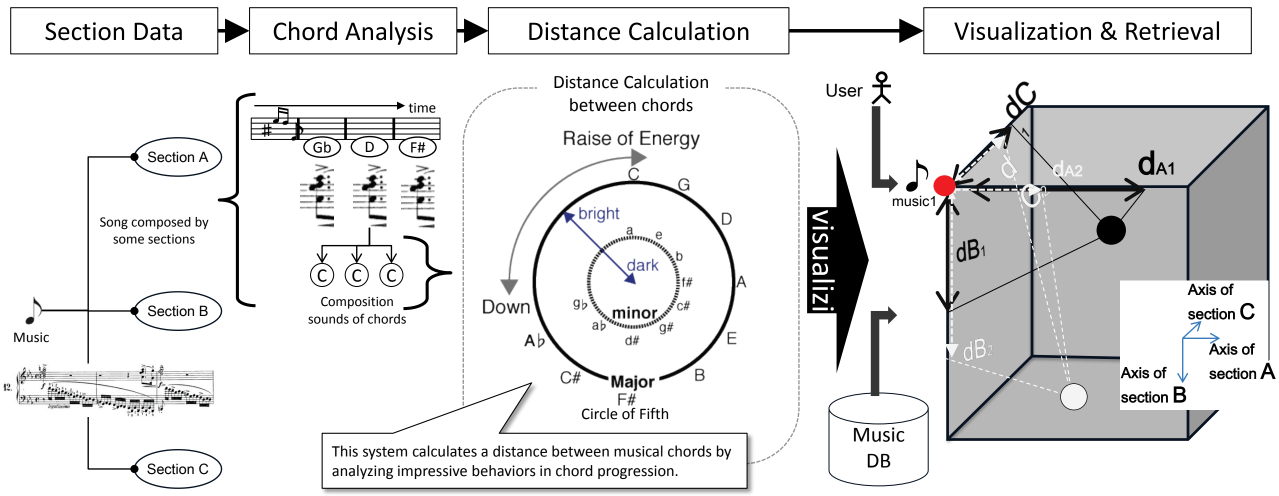

Figure 4. System architecture of Chord-Cube for visualizing tonality-based relevance among songs.

However, ordinary users also want to be able to find new and unknown music more easily, and a method for retrieving music that is similar but not exactly equal to a query would be most helpful in attaining this goal. A wide variety of visualization techniques have been proposed in the context of content-based MIR [9]. For example, Pampalk et al. [10], proposed an interface for discovering artists by using a ring-like structured visual UI, Knees et al. [11] developed a method for visually summarizing the contents of music repositories, and Stober et al. [12] proposed an interface that can conduct music searches on the basis of unclearly defined demands.

Dittmar et al. [13] introduced "GlobalMusic2One," a portal site for visualizing songs by using two-dimensional similarity maps for explorative browsing and target-oriented searches. To enhance the retrieval effectiveness of songs, a music recommendation filter [14] utilizing a user's personal preferences and a method for managing songs in mobile environments [15] has also been proposed. From the aspect of musical structure analysis, several music visualization systems have been developed [16][17][18]. These methods use the color sense of tonality to view the harmonic structure and relationships between key regions in a musical composition. Imai et al. [19] also proposed tonality-based visualization as a means of enhancing the find-ability of music.

The most significant difference between conventional approaches and our approach is that our system focuses on the development of a method for emotion-based music visualization. Conventional visualization MIR methods automatically extract content information from audio signals by applying signal processing and machine learning techniques, whereas our system analyzes emotional transitions by capturing the progression of chords as a trajectory of "how the music sounds." Our system can calculate the evolving distance between two chord vectors as a continuous comparison along a timeline. Another significant innovation delivered by our method is the use of an interactive 3D visualization space. This visualization method configures a 3D cube around an example query serving as an origin vertex point, and displays each musical item according to its relevance score relative to the example query.

III. MOTIVATING EXAMPLE

Our Chord-Cube system is envisioned for use in scenarios such as the following. Imagine that a user has 1,000 songs in his/her smartphone and s/he desires to select songs that are similar in emotion to a specific song in the smartphone. In this case, the user could retrieve a desired song by inputting a sample song and browsing the visualized 3D cube where relevant songs are located close to the input song. As shown in Figure 2, when the user inputs a song, the system positions the song at vertex (0, 0, 0) of the cube. Further, the other songs are plotted within the 3D cube, as shown in Figure 3, indicating the distance between the query song at the vertex and the various points in the cube. Users can then compare songs from various perspectives as follows: similarity in "introductive-melody," similarity in "continued-melody," and similarity in "bridge-melody." Users can rotate this cube to find the most desirable song.

Another scenario that exemplifies the objective of our system relates to the user experience aspect. First, the user selects his/her favorite song in a smart device, such as an iPad or an Android tablet. Then, the system generates the cube showing the relevant songs around the selected song. The user then draws an oval from within which songs are selected and added to the playlist. Such a spatial approach to defining a playlist is effective because of the distance metrics in our cube visualization. Because our visualization mechanism shows dynamically measured semantic distances between music items rather than relevance rankings, the visualized music space provides an intuitive interface for users to choose new music samples of interest.

IV. SYSTEM ARCHITECTURE

A. Architectural Overview

Figure 4 gives an architectural overview of our Chord-Cube system. Music navigation within the Chord-Cube system is achieved through integration of music content analysis and relevance visualization. The overall system comprises a distance calculation module and a visualization module. In order to extract the chord features of a music sample, the distance calculation module inputs the music sample as a query for analysis. The module then computes the distances between the chord features extracted from the query and each music item within the database on the basis of a key distance calculation technology that can measure the distance between two chords according to their respective temporal contexts (i.e., chord progressions). To define the relationship between chord combinations and progressions, we have developed a matrix-based data structure.

B. *Emotive Distance Calculation Using Circle of Fifth*

The system calculates the similarity of songs using the impressive motion defined in Circle of Fifth [20]. The center of Figure 4 illustrates the distance metrics used by the Circle of Fifth [20]. This circle represents the relations of closeness or similarity and distance between tonal elements. In this circle, two adjacent tonalities have similar impressions, but opposite face tonalities have opposite impressions. By tracing a trajectory of chords within the Circle of Fifth, the system can calculate and represent the manner in which the music affects a listener's emotional perceptions. Figure 11 shows a visualization of tonality changing in a music item. The horizontal axis corresponds to timeline, whereas the vertical axis corresponds to the tonality relevance score. This chart shows 12 types of major tonalities and 12 types of minor tonalities. As shown in the chart, one musical composition contains continuous changes in tonality. To detect the emotional changes in music, it is important to trace this tonality behavior. To analyze the change in tonality, well-studied key-finding algorithms, such as the Krumhansl-Schmuckler algorithm [20] and the Temperley algorithm [21][22], can be used. We implemented the Krumhansl-Schmuckler algorithm in the Chord-Cube system.

In order to make selection of the desired music easy, the system displays the calculated distances between samples in a 3D graphical user interface. The visualization module constructs a virtual cubic space consisting of axes corresponding to three music structures typically found in J-pop music: introductive-melody, continued-melody, and bridge. The input query is placed at the origin, while target music items are located within the space according to their respective relevance scores; thus, the most relevant music item is located the closest to the origin, while irrelevant music items are scattered further away.

C. *Visualization Process*

The system performs chord progression oriented music visualization using the following steps:

Step-1. A user inputs a song as a criterion for finding new songs (Figure 5).

Step-2. The system divides the song's chord progression into component sounds (Figure 6).

Step-3. Using a method based on the cycle of fifths, the semantic distances between components are calculated and placed within a feature vector, called the chord vector (Figure 7).

Step-4. The inner products between the chord vectors of each section are calculated to determine the similarities between each of the sections (Figure 8).

Step-5. The relevance of each song is then plotted within a 3D cube in order to present an intuitive visualization of the distance between the song at the vertex and the various points in the cube (Figure 9).

Step-6. Further retrieval can be done by translating another song within the cube to the vertex in order to create a new relevance comparison based on the selected song as the origin (Figure 10).

These visualization mechanisms allow users to retrieve a desired song from an intuitive visual space based on its similarity in chord progression to the reference query song at the vertex.

V. DATA STRUCTURES

Our system contains three fundamental components: A) musical instrument digital interface (MIDI) song data, B) chord progression, and C) component sounds distance matrix.

A. *MIDI Song Data*

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

Our system provides a matrix structure that represents the continuous changing and distribution of pitch in the target music data. We call the data structure a music pitch matrix. The pitch matrix is a 128 by n matrix that is given as the data matrix. MIDI specification defines the domain of the pitch value as zero to 127. A musical composition is expressed as a set of m timelines. Each timeline is characterized by a note on information for zero to 127 pitch levels. When the 12-th note is on in an m-th section, $c_{[12,\,m]}$ is one. The pitch matrix P is defined as follows:

$$P := \begin{pmatrix} c_{[0,0]} & \cdots & c_{[0,n]} \\ \vdots & \ddots & \vdots \\ c_{[m,0]} & \cdots & c_{[m,n]} \end{pmatrix} \qquad (1)$$

where $c_{[i,j]}$ denotes the status of the j-th pitch at the i-th time duration. We implemented the MIDI analysis modules for converting MIDI into a musical score-like data structure by using our MediaMatrix system [23], a stream-oriented database management system.

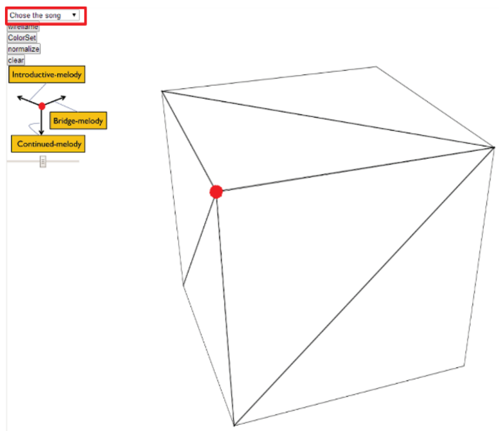

Figure 5. Querying Step-1: A user chooses a song as an origin point.

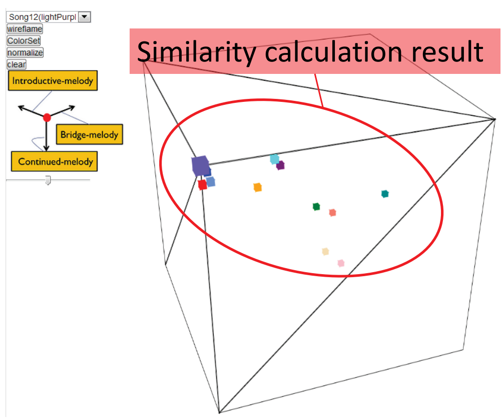

Figure 6. Querying Step-2: System visualizes similarity calculation results

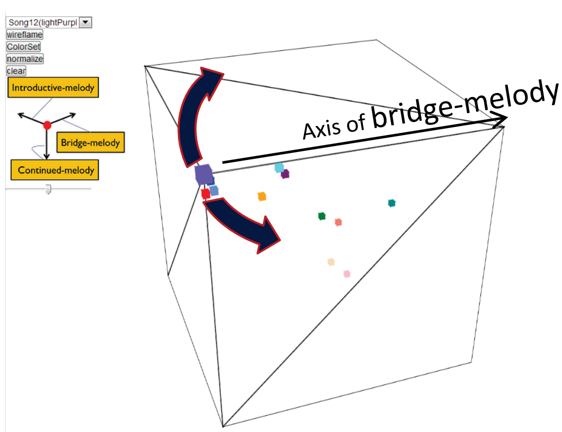

Figure 7. Querying Step-3: User rotates the cube about the axis of the bridge-melody

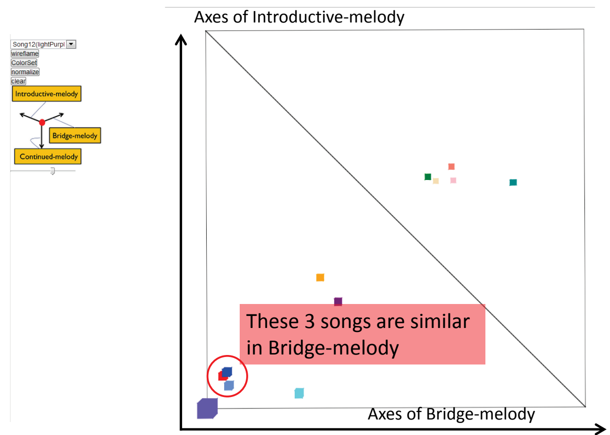

Figure 8. Querying Step-4: User can get information about songs with similar bridge-melody.

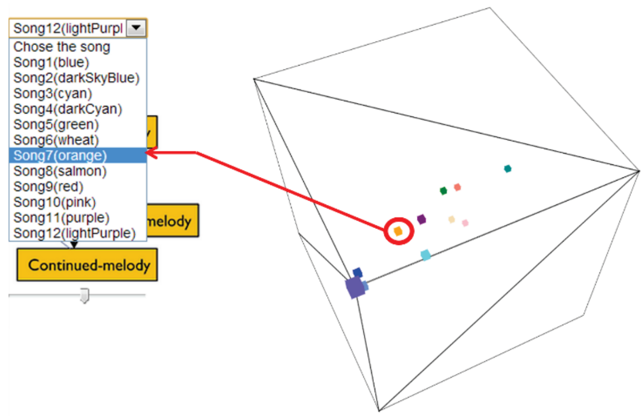

Figure 9. Querying Step-5: User selects another song for new criteria of visualization

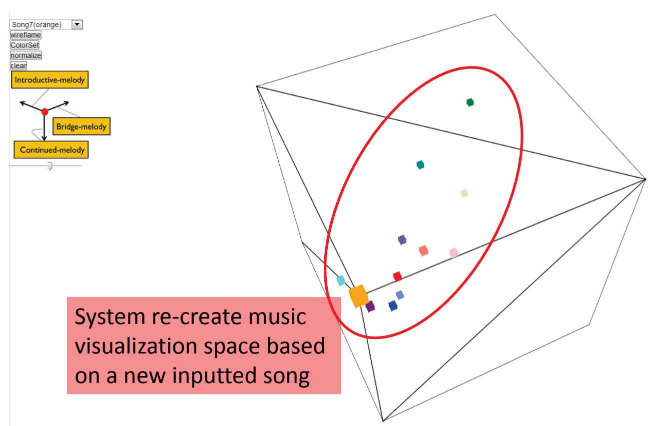

Figure 10. Querying Step-6: System re-creating similarity visualization space on the basis of the new criteria

B. *Chord Progression*

Chord progression refers to continuous chord changes over time. We adopt the concept of tonality, which is a musical system that is constructed by sound elements, such as harmonies and melodies [20]. Different tonalities have different impressions. In one musical composition, tonality changes from section to section. It is important to support this change in tonality in a music database because changes in tonality causes changes in impression.

A music item is modeled as a sequence that consists of chords. More specifically, we define an item Music (M) as a data structure consisting of a sequence of chords (c). Music M_i is defined by the following equation:

$$M_i := \langle c_0, c_0, c_1, \cdots, c_n \rangle \tag{2}$$

where n is the number of chords. A chord is a 12-tuples relevance score, where each tuple corresponds to a specific type of tonality such as C and $C^{\#}$. Therefore, we define a chord (c) as a data structure based on correlation of k-th tonality (v_k). Chord c_j is defined by the following equation:

$$c_j := \langle v_1, v_2, \dots, v_{12} \rangle \tag{3}$$

where v_k corresponds to the k-th tonality; hence, there are 12 values in this vector.

C. Component Sounds Distance Matrix

Chord progressions are composed of three or more overlapping sounds. We call these overlapping sounds "component sounds." We have developed a correlation matrix that defines the movement distance for each combination of tonalities. Figure 12 shows a component sounds distance matrix designed using the Circle of Fifths. The component sounds distance matrix is a 12 × 12 matrix that is given as the data matrix. The size of the matrix corresponds to the number of tonality types defined in the Circle of Fifth. In this matrix, a larger value signifies a stronger correlation. Thus, C and $C^{\#}$ (0.83) are more correlative than C and $D^{\#}$ (0.50). The component sounds distance matrix T is defined as follows:

$$\mathrm{T} := \begin{pmatrix} d_{[1,1]} & \cdots & d_{[1,12]} \\ \vdots & \ddots & \vdots \\ d_{[12,1]} & \cdots & d_{[12,12]} \end{pmatrix} \tag{4}$$

where $d_{i,j}$ denotes a correlation value for the j-th and i-th tonalities.

Our Chord-Cube system uses this matrix to calculate the similarity between songs, based on their component sounds, by multiplying the number of occurrences of each particular sound by its respective distance. As a result, we can obtain a vector representing the strength of the sounds in the song. We call this vector the "chord vector." The system then constructs a chord-vector space consisting of the calculated 12-dimensional values. The system calculates the relevance of two songs by measuring the distance of two chord vectors, which represent how the chords change in each song. In addition, the system can compare songs according to their sectional contents, such as introductive-melody, continued-melody, and bridge-melody, by calculating a chord vector based on each section of a song.

VI. CORE FUNCTIONS

Our system contains four fundamental components: A) a chord detector for converting a pitch matrix into chord progression array, B) a chord-vector generation module for generating a vector data by analyzing the chord progression array, C) a distance calculation module applied to determine the semantic distance of songs, and D) visualization module.

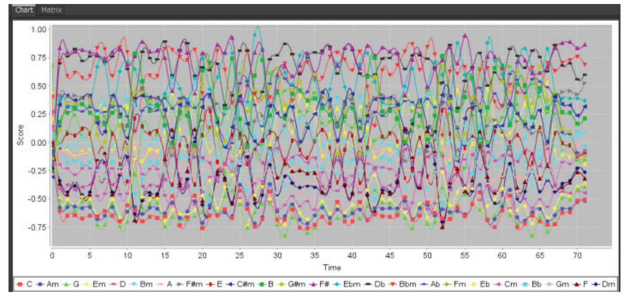

Figure 11. A visualization of tonality changing in one music item. The tonality changes with time.

	C	C#	D	D#	E	F	F#	G	G#	A	A#	B
C	0	0.83	0.33	0.50	0.67	0.17	1	0.17	0.67	0.50	0.33	0.83
C#	0.83	0	0.83	0.33	0.50	0.67	0.17	1	0.17	0.67	0.50	0.33
D	0.33	0.83	0	0.83	0.33	0.50	0.67	0.17	1	0.17	0.67	0.50
D#	0.50	0.33	0.83	0	0.83	0.33	0.50	0.67	0.17	1	0.17	0.67
E	0.67	0.50	0.33	0.83	0	0.83	0.33	0.50	0.67	0.17	1	0.17
F	0.17	0.67	0.50	0.33	0.83	0	0.83	0.33	0.50	0.67	0.17	1
F#	1	0.17	0.67	0.50	0.33	0.83	0	0.83	0.33	0.50	0.67	0.17
G	0.17	1	0.17	0.67	0.50	0.33	0.83	0	0.83	0.33	0.50	0.67
G#	0.67	0.17	1	0.17	0.67	0.50	0.33	0.83	0	0.83	0.33	0.50
A	0.50	0.67	0.17	1	0.17	0.67	0.50	0.33	0.83	0	0.83	0.33
A#	0.33	0.50	0.67	0.17	1	0.17	0.67	0.50	0.33	0.83	0	0.83
B	0.83	0.33	0.50	0.67	0.17	1	0.17	0.67	0.50	0.33	0.83	0

Figure 12. Component sound distance matrix representing distance between each sound based on tonality.

A. Chord Detector

The system provides a fundamental function to convert a pitch matrix into chord progression arrays. The function f_{map} extracts chords by detecting three or more overlapping sounds in the pitch matrix. We define $f_{map}(P_i)$ that inputs a pitch matrix P_i as follows:

$$f_{map}(P_i) \rightarrow M_i \tag{5}$$

where M_i denotes a sequence of chords. The detailed definition of M_i is given in Section V-B, equations (2) and (3).

B. Chord Vector Generation

The system generates a chord vector by summing the matrix consisting of the products of the semantic distance of each sound on the cycle of fifths with the number of occurrences of that sound, as defined by

$$f_{CV}(d,e) := \left(\sum_{i=1}^{12} d_{[i,1]} \cdot e_{[i]}, \quad \cdots, \quad \sum_{i=1}^{12} d_{[i,12]} \cdot e_{[i]} \right) \tag{6}$$

where d represents the distance between the component sounds, while e represents the number of occurrences of each component sound. The chord vector thus generates and stores a correlation between all component sounds in each section.

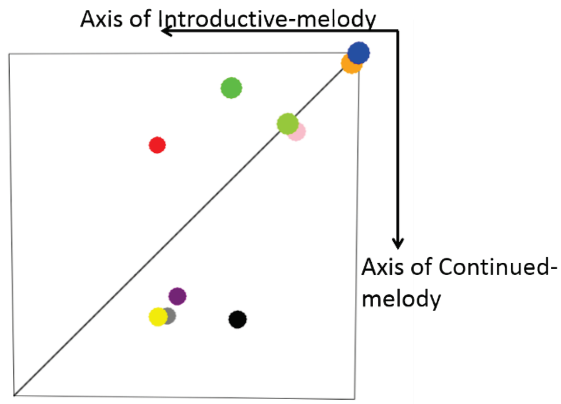

Figure 13. The system facilitates comparison of music items from multiple perspectives. In this case, a user compares from introductive-melody and continued-melody.

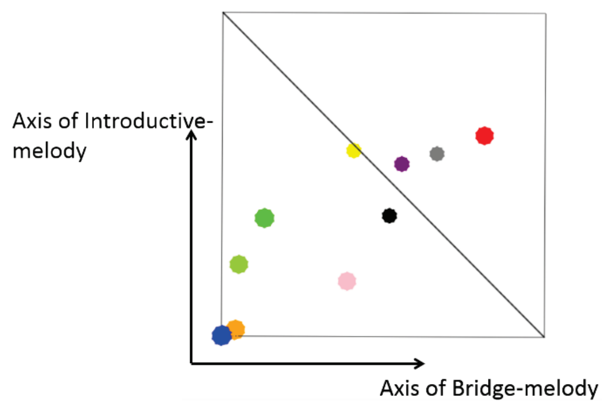

Figure 14. The system facilitates comparison of music items from multiple perspectives. In this case, a user compares from introductive-melody and bridge-melody.

C. Distance Calculation

As stated in the previous section, the chord-vector matrix is derived by multiplying the component sounds distance matrix with the number of occurrences of each sound; this result consists of a 12-dimensional vector representing the strength of each sound within a section. The system compares songs in terms of their representative features encoded in the 12-dimensional distance metric space ("chord-vector space") by their respective chord vectors. Distances between sections are calculated from the inner products of vectors, using

$$f_{distance}(CV_1 \cdot CV_2) \coloneqq \sum_{i=1}^{12} CV_{1[i]} \cdot CV_{2[i]} \quad (7)$$

where CV_1 and CV_2 are the chord vectors of two different songs.

D. Visualization Module

The system utilizes the chord vector to compare user-selected songs to all songs in the music database. Defining each section of music1 (i.e., a user-imported song) as S1a, S1b, and S1c, and of music2 (another song in the database) as S2a, S2b, and S2c, the similarity calculation function distance between S1a and S2a is calculated as d1, the distance between S1b and S2b is d2, and the distance between S1c and S2c is d3. If, on the 3D space consisting of the respective song section type, music1 is located at the origin (0, 0, 0), then the coordinates for music2 can be represented as (d1, d2, d3). Thus, the system can visualize the distances between songs as Cartesian distances in a solid body called the "Chord-Cube," as shown in Figure 3.

The system is able to adopt differing user-input styles; therefore, it is able to make comparisons between songs on the basis of varying criteria. Each song can be assigned vector values and allocated a coordinate in the cube on the basis of its correlation to a particular criterion, creating a space that intuitively represents the semantic distance between songs, and in which the most relevant piece of music is located very close to the origin, while irrelevant items are more remote. Figures 13 and 14 show typical and effective use cases of this system. A typical scenario in which a user compares songs from multiple aspects is depicted. Figure 13 shows a perspective for detecting the similarity by using the introductive-melody and continued-melody. Figure 14 shows a comparison between the introductive-melody and the bridge-melody. It can be seen that there are obvious differences about the dark green and pink spheres between those two figures. In Figure 13, the two songs represented by these spheres have identical similarities to the blue sphere, whereas they are separated in Figure 14. This means that the two songs are similar in terms of introductive-melody and continued-melody, but have different features in terms of bridge-melody.

VII. Web-Based System Implementation

We implemented a prototype of the Chord-Cube system that calculates the similarity between songs and visualizes them in a 3D cubic space. Screenshots of the prototype, which uses HTML5 Canvas and JavaScript, are shown in Figures 5 through 10. Figure 15 details the architecture of our prototype system, which specifically includes the modern HTML5 technologies WebGL API, Web Storage API, and Web Worker API. The system consists of the following three modules: a query input module, a distance calculation module, and a visualization module. We describe these components in detail below.

The main user interface is the visualization module, which uses the HTML5 WebGL API to render a three-dimensional interactive screen. We implemented this prototype system by utilizing three.js (http://threejs.org/), an open-source WebGL wrapper utility library. The implemented system extends three.js to support interactive music data visualization and real-time rendering of the chord-vector space. This 3D UI enables users to compare songs from any desired perspective. Users can view the rendered cube and spheres representing songs from any angle by rotating the cube, zooming in, and zooming out.

When users input a song as a query, the system invokes the MIDI file analyzer implemented in JavaScript. This MIDI file analyzer is implemented using the HTML5 FileReader and ArrayBuffer objects. On completing the analysis process,

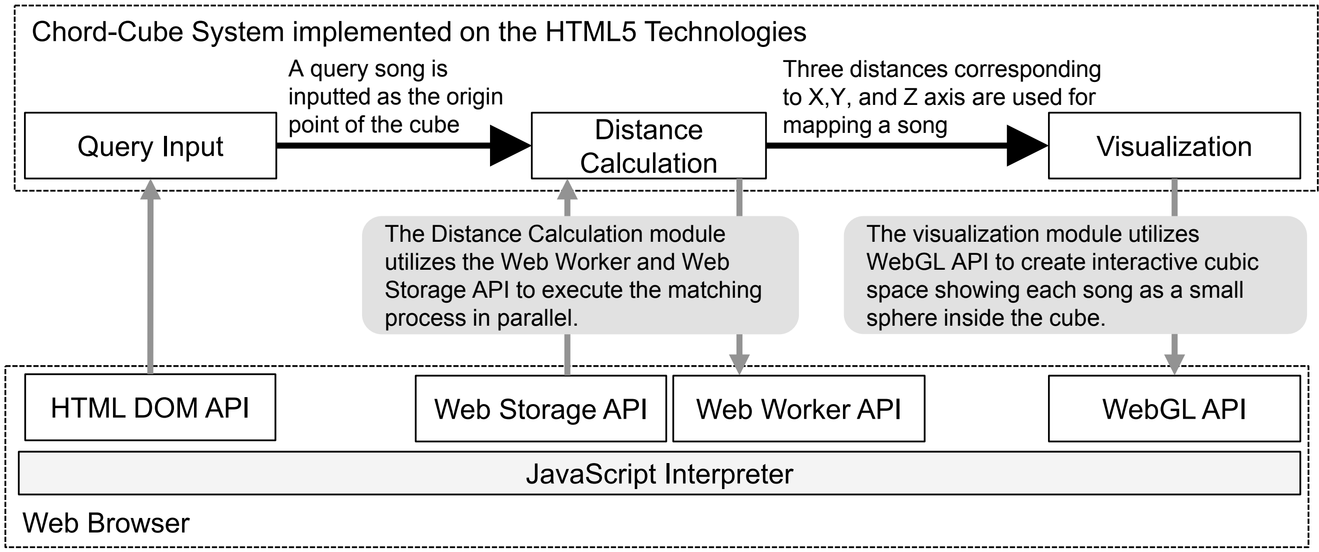

Figure 15. Conceptual view of the query-by-appearance system for style-oriented e-book retrieval using encapsulated editorial design templates for query generation.

the visualization module renders the query song on a vertex of the cube. In addition, 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 execution of distance calculation.

The distance calculation module compares the queries and the database contents. This retrieval process is parallelized by the Web Workers API, and the retrieved songs are presented to the user by the search result visualization engine. This system spawns real OS-level threads from the Web Workers API to parallelize the retrieval process. In this way, modern HTML5 technologies enable us to implement complex processes in web browsers.

After getting a number of users to evaluate the implemented system, we became cognizant of two primary music retrieval use cases. In case-1, the user desires to search for similar songs via the bridge-melody of one song. In this case, the user performs the following music retrieval process:

- Step-1: The user inputs the song that s/he wants to set as the comparison criteria for a bridge-melody.
- Step-2: The system visualizes the similarity calculated based on the input song.
- Step-3: The user rotates the cube on the axis corresponding to the bridge-melody.
- Step-4: The user obtains songs similar in bridge-melody by seeing the visualized results around the axis of the bridge-melody.

In case-2, after the user has found his/her desired song, s/he uses the found song as a query in order to retrieve more songs. This case continues the previous process in case-1.

- Step-5: The user selects a specific song as a new query from the visualized cube.
- Step-6: The system recreates music visualization space based on the new query song.
- Step-7: The user repeats Step-5 and Step-6 until s/he has retrieved enough music items.

VIII. EVALUATION

In this section, we discuss several experiments conducted to evaluate the effectiveness of our Chord-Cube system when applied to existing Japanese Pop songs. We conducted the following two experimental studies: Experiment-1, evaluation of the precision of dissimilarity calculations; and Experiment-2, evaluation of the effectiveness of our visualization. We performed the two experiments by comparing the results of similarity measurements between the implemented system and the results of questionnaires submitted to listeners who awarded points based on the level of similarity that they felt. As preprocessing for the two experiments, we asked 10 subjects (three male and seven female) to create a correct set for each query in Experiment-1 and Experiment-2. The correct set is a data set that stores only items that are considered relevant to a query by test subjects.

A. Experiment-1: Outline of Experimental Studies

Experiment-1 was conducted to evaluate the effectiveness of our similarity calculation precision. For this experiment, we chose one query song as a criterion and 10 other songs as comparison targets. Ten listeners used a one-to-five scoring template to evaluate their perceptions of similarity between each comparison song and the criterion by section, after which we aggregated the scoring results from each listener and converted them into reciprocal values defined as the "dissimilarities by survey." We then used these values to calculate the distance within the Chord-Cube of each target song from the criterion point (the query); this process is called "collection of data dissimilarity." To evaluate the effectiveness of our method, we compared the dissimilarities by survey to the dissimilarities as calculated by our method. In Experiment-1-A, we applied our system to measure dissimilarities of introductive-melody for each music item, whereas in Experiment-1-B,

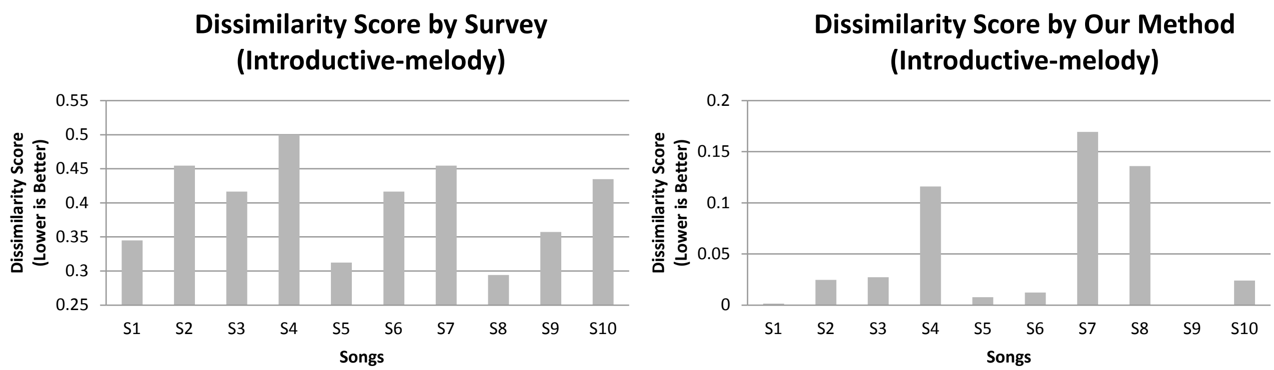

Figure 16. Results of Experiment-1-A: Dissimilarity measurement for introductive-melody.

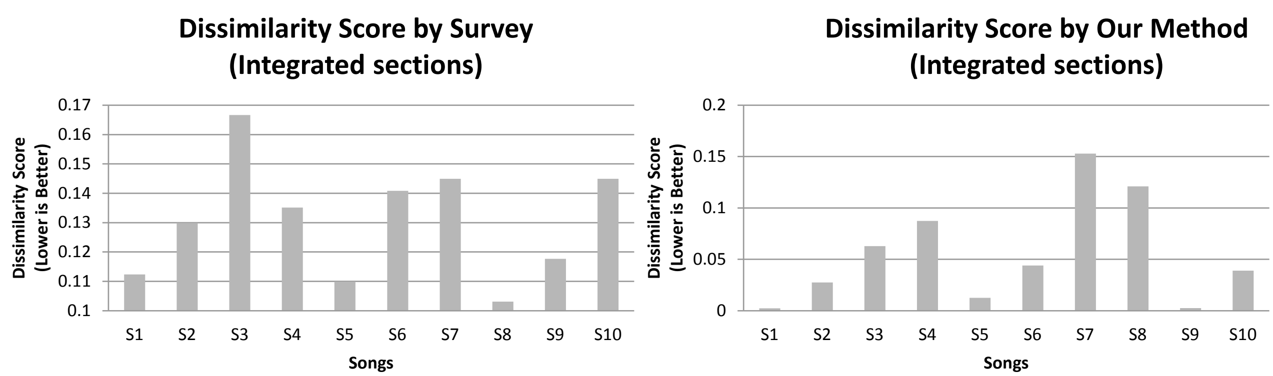

Figure 17. Results of Experiment-1-B: Dissimilarity measurement for integrated sections.

we measured dissimilarities of integration of introductive-melody, continued-melody, and bridge-melody for each music item.

B. Experiment-1: Experimental Results

Figure 16 and TABLE I show the results for Experiment-1-A. The left-hand side of the figure shows the dissimilarity as measured by the manual survey, while the right-hand side shows the dissimilarity as measured by our system. It can be seen in TABLE I that the test subjects judged songs s1, s4, s5, s8, and s9 to be highly similar to the query music, whereas our system retrieved songs s1, s5, s6, s9, and s10 as similar music; thus, the system correctly extracted songs s1, s5, and s9.

Figure 17 and TABLE II show the results for Experiment-1-B. As before, the left-hand side shows dissimilarity measured by manual survey, and the right-hand side shows dissimilarity measured by our system. By comparing Figures 16 and 17, it can be seen that the surveyed dissimilarity of song s3 significantly increases from Experiment-1-A to Experiment-1-B, whereas our system returns identical results for all songs in both experiments. Thus, it can be concluded that our system improves its retrieval precision by integrating a differing evaluation axis into the Chord-Cube visualization space, and thus can effectively display multiple perspectives simultaneously.

The results for song s8, on the other hand, show that some improvements are still necessary. Whereas the survey results judged s8 to be similar to the query music, our system judged it to be dissimilar. We believe that a perceptional gap between the theme melody and the chords progression of song s8 strongly affected the results here, because s8 has a complex chord progression but a very simple melody. However, the experimental results from the other songs closely parallel the results obtained from the dissimilarity by survey, clarifying the overall effectiveness of our method for utilizing chord-metric space and 3D visualization.

C. Experiment-2: Outline of Experimental Studies

In this section, we evaluate the precision of our visualization result by using three types of queries. This experiment clarifies that our approach calculates the appropriate distance between songs. As in Experiment-1, we compared the results of similarity measurements between calculated results and questionnaire survey. For this experiment, we established three query songs as criteria and ten other songs as comparison targets. We have selected three songs from ten JPOP songs randomly. Ten test subjects (three male and seven female) used a one-to-five scoring template to evaluate their perceptions of similarity between each comparison song and the criterion by section. The scoring template is as follows: 0 (completely irrelevant), 1 (irrelevant), 2 (slightly relevant), 3 (relevant), and 4 (very relevant). We consider the ideal ranking as the average of ten results. We then used these scores to

TABLE I. SIMILARITY RANKS OF INTRODUCTIVE-MELODY

Rank	Survey	Score	Our Method	Score
1	s8	0.294118	s9	0.000280
2	s5	0.312500	s1	0.001422
3	s1	0.344828	s5	0.007712
4	s9	0.357143	s6	0.012242
5	s6	0.416667	s2	0.024543

TABLE II. SIMILARITY RANKS OF INTEGRATED SECTIONS

Rank	Survey	Score	Our Method	Score
1	s8	0.103093	s9	0.002429
2	s5	0.109890	s1	0.002381
3	s1	0.112360	s5	0.012566
4	s9	0.117647	s2	0.027525
5	s2	0.129870	s6	0.044053

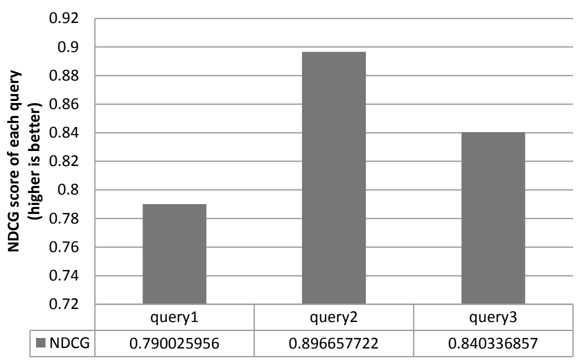

Figure 18. NDCG Scores.

compare with the distance from the origin point in the visualization result.

D. Experiment-2: Experimental Results

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

$$DCG = \sum_{i=2}^{11} \frac{rel_i}{log_2 i} \quad (8)$$

$$IDCG = \sum_{i=2}^{11} \frac{rel'_i}{log_2 i} \quad (9)$$

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

where rel_i are the average survey scores given by the test subjects in order based on ranking of visualized distance, and rel'_i are the average scores in descending order. Figure 18 shows the NDCG of visualization in the Chord-Cube for three queries. A higher score implies a better retrieval precision. From this experimental result, we obtain a value for NDCG that is higher than 0.79 in every query. This result explains the high precision of the visualization result of our system.

IX. CONCLUSION AND FUTURE WORK

In this paper, we proposed the Chord-Cube system, a music visualization and navigation system that provides an intuitive visual retrieval method using chord-metric space. The unique feature of this system lies in its construction of a chord-vector space to extract the transition of emotions within a song as a feature vector. We implemented a prototype system utilizing modern HTML5 technologies. The implemented system supports the chord-metric based similarity between songs according to a user's selected criterion song and visualizes that result within a 3D cube constituted by three evaluation axes. We also performed evaluations of the effects of our system applied to existing J-pop songs. Our experimental results indicate that visually represented search results carry out a practical function.

In future work, we plan to improve the chord-metric space by capturing the direction of chord transitions in order to represent the change in emotional energy through the resulting motion on the cycle of fifth. We are also developing an automatic playlist generation function using the spatial analogy for selecting songs in the visualized cube. The most important future work is to apply our system to raw audio signals, such as MPEG Audio Layer-3 (MP3) format. Our system rely on the score data of music, so we are planning to integrate an existing music transcription system into the Chord-Cube system. In order to enhance the query description scope, multiple songs can be used as a query. We have implemented such a query interpretation method for video retrieval in [24].

REFERENCES

[1] Imai, T. and Kurabayashi, S., "Chord-Cube: multiple aspects visualization & navigation system for music by detecting changes of emotional content," In Proceedings of the Eighth International Conference on Internet and Web Applications and Services (ICIW 2013), pp.129-134, June 23-28, 2013.

[2] Goto, M. and Hirata, K., "Recent studies on music information processing," Acoustical Science and Technology, vol. 25, no. 6, the Acoustical Society of Japan, pp. 419-425, 2004.

[3] Type, R., Wiering, F., and Veltkamp, R.C., "A survey of music information retrieval system," In Prof. of the 6th International Conference on Music Information Retrieval (ISMIR 2005), pp. 153-160, 2005.

[4] Ghias, A., Logan, J., Chamberlin, D., and Smith, B.C., "Query by humming: musical information retrieval in an audio database," In Prof. of the Third ACM International Conference on Multimedia (MM 1995), pp. 231-236, 1995.

[5] Dannenberg, R.B., Birmingham, W.P., Tzanetakis, G., Meek, C., Hu, N., and Pardo, B., "The MUSART testbed for query-by-humming evaluation," In Prof. of 4th International Conference on Music Information Retrieval (ISMIR 2003), pp. 34-48, 2003.

[6] 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.

[7] Cheng, H.T., Yang, Y.H., Lin, Y.C., Liao, I.B., and Chen, H.H., "Automatic chord recognition for music classification and retrieval," In Proc. of the IEEE International Conference on Multimedia and Expo (ICME2008), pp. 1505-1508, 2008.

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

[9] 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.

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

[11] 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 2006), pp. 17-24, 2006.

[12] 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.

[13] 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.

[14] 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 (SAC 2006), pp. 1050-1057, 2006.

[15] Goussevskaia, O., Kuhn, M., and Wattenhofer, R., "Exploring music collections on mobile devices," In Proc. of the 10th International Conference on Human Computer Interaction with Mobile Devices and Services (MobileHCI 2008), pp. 359-362, 2008.

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

[17] Mardirossian, A. and Chew, E., "Visualizing music: tonal progressions and distributions," In Prof. 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," In Proc. of the 18th International Conference on Multimedia (MM 2010), pp. 1677-1680, ACM, 2010.

[19] 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 (DIMS 2008), pp. 455-460, 2008.

[20] Krumhansl, C.L., "Cognitive foundations of musical pitch," Oxford University Press, 1990.

[21] Temperley, D., "The cognition of basic musical structures," MIT Press, ISBN-13: 978-0-262-70105-1, 2001.

[22] Temperley, D. "Music and probability," MIT Press, ISBN-13:978-0-262-20166-7, 2007.

[23] 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, Springer, 2010.

[24] Kurabayashi, S. and Kiyoki, Y., "Impression-aware video stream retrieval system with temporal color-sentiment analysis and visualization," In Proc. of the 23rd International Conference on Database and Expert Systems Applications (DEXA 2012), pp.168-182, Springer, 2012.