# Chord-Cube: Multiple Aspects Visualization & Navigation System for Music by Detecting Changes of Emotional Content

Tatsuki Imai
Faculty of Environment and Information Studies
Keio University
Fujisawa, Kanagawa, Japan
e-mail: t10109ti@sfc.keio.ac.jp

Shuichi Kurabayashi
Faculty of Environment and Information Studies
Keio University
Fujisawa, Kanagawa, Japan
e-mail: kurabaya@sfc.keio.ac.jp

***Abstract*— This paper presents an interactive music search-and-navigation system visualizing musical similarities based on temporal chord progression. A unique feature of this system is a 3D musical space for displaying three types of similarities in musical samples. As a fundamental feature for calculating those features, we employed chord progression in a song because chord progression is one of the most important factor in determining the overall mood of a song. For rendering the content-based relevance with a timeline structure, our system models a typical pops and rock music as a combination of the following chord progression phases: Introductive-melody, Continued-melody, and Bridge. Our 3D visualization space adopts those three chord progression as X, Y, and Z axes. Our system provides an intuitive navigation mechanism over the visualized space by putting a query song in the origin point and showing semantic distance of the inputted song and other songs. Users can utilize this 3D space to find the desired song by putting the his/her favorites song in the origin point and recognizing the semantic distance of the origin point and other songs.**

***Keywords-Music; Recommendation; Visualization***

## I. INTRODUCTION

The change in emotion over time in a song is one of the most important factors in the selection of music to be played on modern mobile music players and smart phones. Especially, young age users will select music according to their location and mood. To support such intuitive and emotionally-based music selection, a player must provide a smart content analysis in order to extract movements of musical elements that have deep effects on human perception.

Current music database systems implemented in online music stores such as iTunes Music Store and Sony's Music Unlimited do not support such perception-oriented retrieval methods, and as users often own thousands of music in the Cloud, such situation makes users difficult to find out their desired songs intuitively even if he/she knows details of the desired music. Furthermore, owing to the temporal nature of music it is difficult to develop an effective music search environment in which users can retrieve specific music samples by using intuitive queries as searching a temporal structure requires the system to recognize the changing features of the contents in a context-dependent manner.

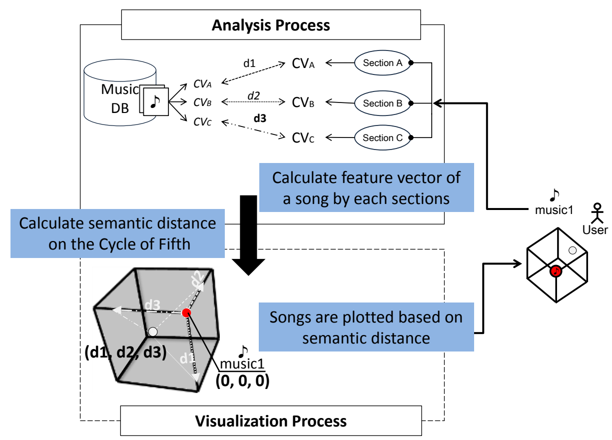


Figure 1. Conceptual diagram of Chord-Cube: Music Visualization and Navigation System within a Chord-Metric Space

Therefore, there is a need to develop a music information retrieval (MIR) method that can reflect the felt by a user as they listens to the music. Such a retrieval environment must have an interactive and navigational user interface that can visualize context-dependent relationships between songs dynamically and according to the user's viewpoint. Whereas traditional MIR systems focus on finding the most relevant song or similar songs by computing similarities or relevance according to extracted features, our system focuses on providing an integrated toolkit with which to compare song in order to create a visualization of implicit interrelationships based on emotional characteristics.

Thus, in order to detect a temporal flow of emotions instilled in a listener, we develop a stream-oriented impression analyzer of chord progression. A unique feature of our system is its "chord-vector space" in which the distance between musical chords can be calculated by analyzing the impressive behaviors of chord progression. By tracing a trajectory of chords within chord-vector space, the system can calculate represent the manner in which the music affects a listener's emotional perceptions. Our system visualizes the impressive relationship between music according to the distance calculated in this vector space.

The core concept of our visualization mechanism is a cubic metric and visual space that uses distances to represent

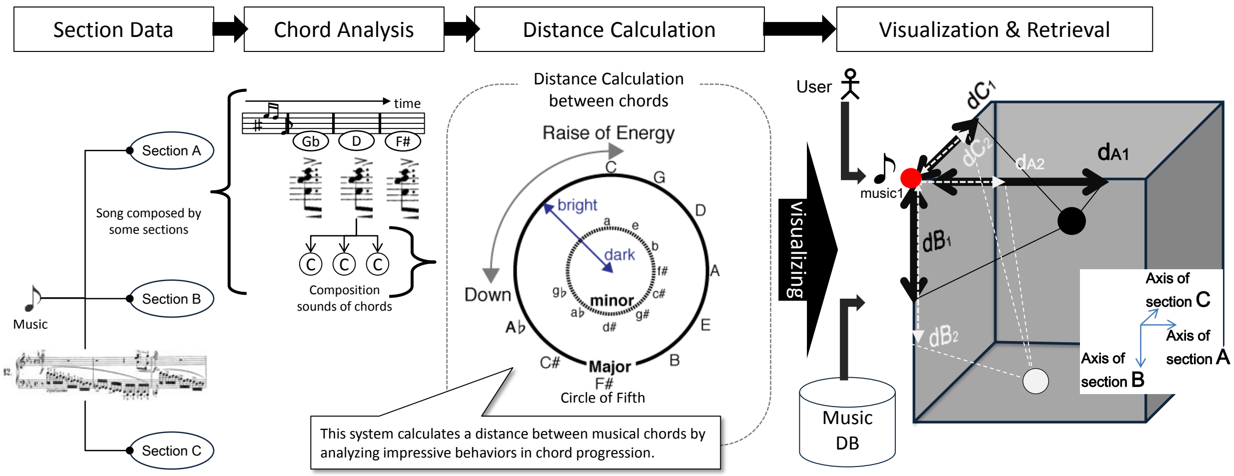


Figure 2. System Architecture of Chord-Cube

the degree of similarity between songs calculated in chord-vector-space by mapping the measured distances into a 3D graphic space for intuitive musical navigation. As shown in Figure 1, each dimension of this graphical space corresponds to the degree of similarity of chords within three respective sets of songs section types: "introductive-melody," "continued-melody," and "bridge-melody." This cube is a three-dimensional object that displays songs as points inside it. By exploiting the above chord-vector space, this system visualizes the distance between songs as a distance between points inside the cube and a vertex of cube.

Our cube accepts an initial song as an origin point in the cube. User can choose any songs as the origin point. The cube system plots other songs inside of the cube, by reflecting the distance between each song and the song at the origin point. Each axis of the cube corresponds to a musical section. For example, the x-axis corresponds to the introductive-melody section, the y-axis corresponds to the continued-melody section, and the z-axis corresponds to the bridge-melody section.

The remainder of this paper is structured as follows. Section II describes related researches of the MIR system. Section III shows an architectural overview of our system. Section IV demonstrates fundamental data structures. Section V defines core functions. Section VI shows prototype implementation of our system. Section VII performs feasibility study. Finally, Section VIII concludes this paper.

## II. RELATED WORK

An MIR system utilizes many aspects of musical data; for instance, fundamental metadata such as genre and artist name, can be set as indexing keys within a conventional MIR system [1]. However, as such fundamental metadata are not sufficient to retrieve music without detailed knowledge of target data, content-based retrieval and advanced query interpretation methods are developed to find music without using fundamental metadata. 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 [2], [3], [4] and chords [5], [6] by utilizing signal frequency analysis methods [7] and power spectrum analysis methods [8]. Overall, 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.

As content-based technologies are very effective in retrieving musical equivalents to inputted queries, they are widely used for copyright protection in online music sharing services. However, common users also want to be able to find new and unknown music more easily, and a method for retrieving music similar but not equal to query would be most helpful in attaining this goal. Several conventional approaches along these lines have already been made, including those by Pampalk et al. [9], who demonstrated an interface for discovering artists, Knees et al. [10], who developed a method of visually summarizing the contents of music repositories, and Stober et al. [11], who reported on an interface that can conduct music searches based on unclearly defined demands.

The most significant difference between our approach and those of the above-mentioned methods is that ours captures emotional transitions; that is, chord-vector space can capture the progression of chords as a trajectory of "how the music sounds" by representing changes of mood in music as a sequence of relevant scores corresponding to 12 types of chords. Based on this, the 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 a 3D visualization space. This visualization method configures a 3D cube around an example query serving as an origin vertex point, displaying each music item according to its relevance score relative to the example query. A user can operate this cube from any perspectives desired.

## III. SYSTEM ARCHITECTURE

As shown in Figure 2, music navigation within the chord-cube system is achieved through integration of music content analysis and relevance visualization. As our visualization mechanism shows a dynamically measured semantic distance between music items rather than a relevance ranking, the visualized music space provides an intuitive interface for users to choose new music samples of interest.

The overall system consists of a distance calculation module and a visualization module. In order to extract chord features of a music sample, the distance calculation module inputs it as a query for analysis. The module computes distances between the chord features extracted from the query and each music item within the database based on a key technology of distance calculation that can measure the distance between two chords based on 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.

In order to make selection of desired music easy, the system displays 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 located 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.

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

1. A user inputs a song as a criterion for finding new songs;
2. The system divides the song's chord progression into component sounds;
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;
4. The inner products between the chord-vectors of each section are calculated to determine the similarities between each of the sections;
5. The relevance of each song is then plotted within a 3D cube in order to present an intuitive visualization of distance between the song at the vertex and the various points in the cube;
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.

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.

TABLE I. COMPONENT SOUNDS DISTANCE MATRIX

| | 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 |

## IV. DATA STRUCTURE

Our system contains four fundamental components: A) chord progression, B) component sounds distance matrix, C) Chord Vector, and D) Visualization.

### A. *Chord Progression*

A chord progression means continuous changes of chords along a time. The system calculates the similarity of songs in terms of their respective chord progressions by using metrics in a chord-vector-space. For each song, the relevant metrics are calculated based on the semantic strengths of chords in terms of their component sounds and the occurrences of chord progression. The module divides chord progressions composed of three or more overlapping sounds into composition sounds and then calculates the correlation between occurrences of each composition sound within a song and the distances of those sounds on a cycle of fifths. The system then constructs a chord-vector space consisting of the calculated 12-dimensional values. By calculating a chord-vector based on each section of a song, comparisons between songs can be made according sectional contents. Calculating the similarity between songs is thus based on an analysis of respective chord progressions composed of three or more sounds elements in order to develop an "impression" of each song. Using a table to store the relationships between chord progression and component sounds (the chord progression component sounds table), the system is able to retrieve occurrences of various component sounds.

### B. *Component Sounds Distance Matrix*

Component sounds distance matrix is a data matrix that stores the semantic distances between sounds on the cycle of fifths as shown in TABLE I. To calculate the similarity between songs based on their component sounds, the system uses this matrix. The values obtained by multiplying the number of occurrences of each particular sound by its respective distance represent the strength of the sounds in the song and constitute the chord vector.

### C. *Chord Vector*

Chord-vector is based on summing the matrix consisting of the products of the semantic distance of each sound on the

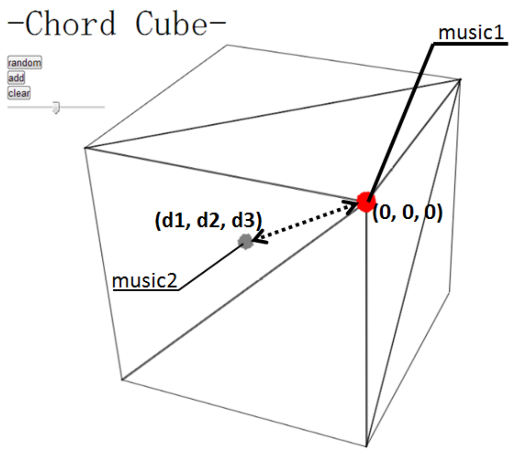


Figure 3. Over view of visualized results

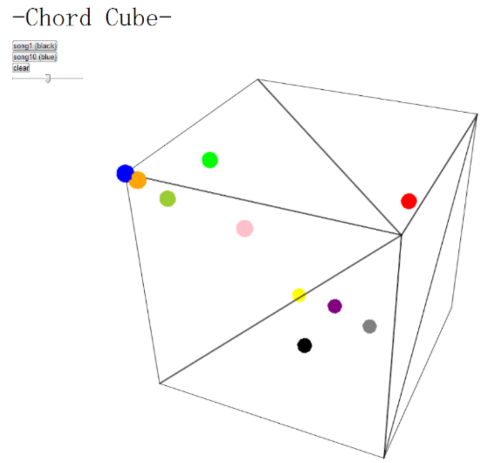


Figure 4. Implementation of the Chord-Cube. Each colored sphere represents a song.

cycle of fifths with the number of occurrences of that sound, as defined by

$$f_{CV}(d,e) := \left(\sum_{i=1}^{12} d_{[i,1]} \cdot e_{[i]},\ \sum_{i=1}^{12} d_{[i,2]} \cdot e_{[i]}\ \cdots,\ \sum_{i=1}^{12} d_{[i,12]} \cdot e_{[i]}\right) \tag{1}$$

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

### D. Visualization

By using the chord-vector, the system compares a user-selected song 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 coordinate of 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.

As the system is able to adopt differing user-input styles, it is able to make comparisons between songs based on varying criteria. Each song can be assigned vector values and allocated a coordinate in the cube based on its correlation to a particular criterion, creating a space that intuitively represents the semantic distance between songs in which the most relevant piece of music is located very close to the origin, while irrelevant items are more remote.

## V. Core Functions

Chord-Vector Calculation: As mentioned 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.

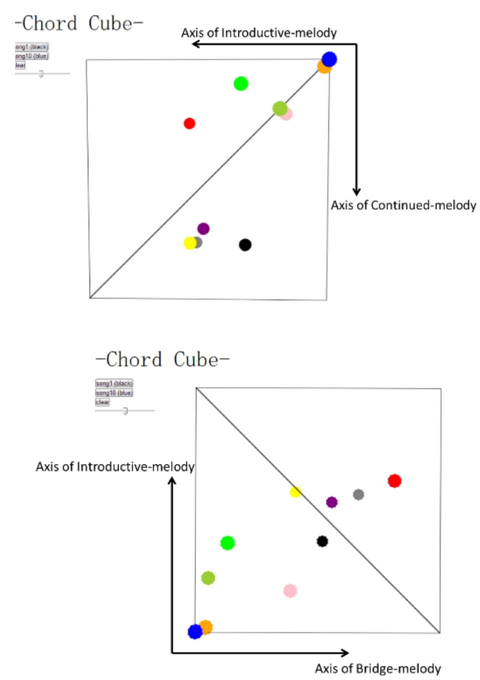


Figure 5. The system allows comparison multiple perspectives.

Chord-Vector Space: 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

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

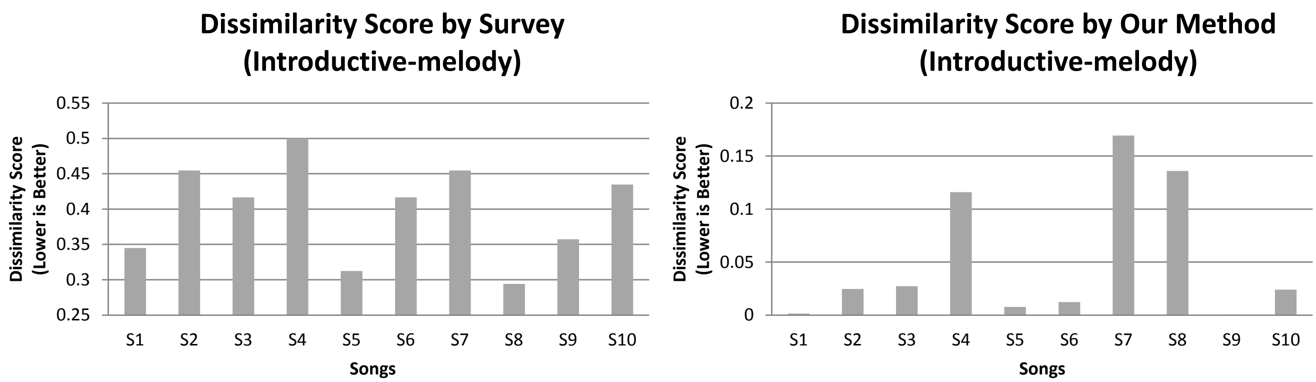


Figure 6. Results of similarity measurement for introductive-melody

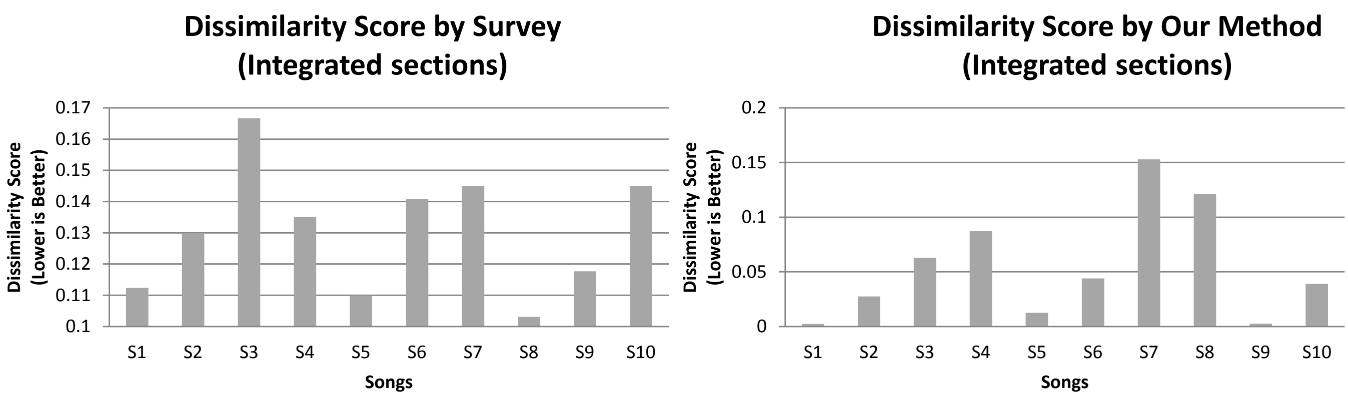


Figure 7. Results of similarity measurement for the Integrated sections

, where CV1 and CV2 are the chord-vector lengths of two different sections.

## VI. System Implementation

Using Three.js Canvas and JavaScript, we implemented a prototype chord-cube system to calculate the similarity of chords by song section, as shown in Figure 4. Using JavaScript, the prototype system can represent an extended library of 3D depictions through a visualization area and an interactive user interface (UI) in which spheres and cubes can be viewed from any angle. One unique feature of this interface is that the user can see comparison outputs from any desired perspective. For example, the upper section of Figure 5 shows a perspective view from the y-axis in which the introductive-melody and continued-melody axes can be seen, representing a musical comparison between these respective sections. Similarly, the perspective in the lower figure represents a comparison between introductive-melody and bridge. As an example of the multiple perspectives viewable in the cube, the dark green and pink spheres display obvious differences between the two figures. In the upper figure, the two songs represented by these spheres have identical similarities to the blue sphere, while in lower figure they are located at differing distances. This is interpreted to mean that the two songs have are similar in terms of introductive-melody and continued-melody, and continued-melody, but difference in terms of bridge-melody.

## VII. Evaluation

### A. *Outline of experimental studies*

In this section, we evaluate the effectiveness of our system by examining its precision in providing musical analyses of input chord data. Our purpose here is to clarify the effectiveness of our method of retrieving songs by means of chord-vector space 3D visualization, and we do this by comparing the results of similarity measurements between calculated results to those of a questionnaire survey submitted to listeners who score points based on the level of similarity they feel in each section. The resulting evaluation of effectiveness comes from comparing the dissimilarities as measured by this scoring method to the distance of each song from the criterion points shown in visualization area.

For the experiment, we established one query song as a criterion and ten other songs as comparison targets. Ten listeners used a 1-to-5 scoring template to evaluate their perceptions of similarity between each comparison song and the criterion by section, and we aggregated the scoring results of each listener and converted them into reciprocal values defined as the dissimilarities by survey. We then

TABLE II. THE SIMILARITY RANKS OF INTRODUCTIVE-MELODY

| Rank | Survey | Score | Our method | Score |
|---|---|---|---|---|
| 1 | s8 | 0.294118 | s9 | 0.00028 |
| 2 | s5 | 0.3125 | 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 III. THE SIMILARITY RANKS OF INTEGRATED OF SECTIONS

| Rank | Survey | Score | Our Method | Score |
|---|---|---|---|---|
| 1 | s8 | 0.103093 | s9 | 0.002429 |
| 2 | s5 | 0.10989 | s1 | 0.002381 |
| 3 | s1 | 0.11236 | s5 | 0.012566 |
| 4 | s9 | 0.117647 | s2 | 0.027525 |
| 5 | s2 | 0.12987 | s6 | 0.044053 |

used these values to calculate the distance within the chord-cube of each target song from the criterion point; 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 the method. In experiment-1, we applied our system to measure dissimilarities of introductive-melody for each music item, while in experiment-2, we measured dissimilarities of integration of introductive-melody, continued-melody, and bridge-melody for each music item.

## B. *Experimental Results*

Figure 6 and TABLE II shows the result of experiment-1. The left-hand side of the figure shows dissimilarity as measured by the manual survey, while the right-hand side shows dissimilarity as measured by our system. It can be seen in TABLE II that the test subjects judged songs s1, s4, s5, s8, and s9 to be highly similar to the query music, while our system retrieved songs s1, s5, s6, s9, and s10 as similar music; thus, the system correctly extracted songs s1, s5, and s9.

Figure 7 and TABLE III shows the result of experiment-2. Again, the left-hand side shows dissimilarity measured by manual survey, and the right-hand side shows dissimilarity measured by our system. By comparing Figure 6 and 7, it can be seen that the surveyed dissimilarity of song s3 drastically increases from experiment-1 to experiment-2, while our system returns identical results for all songs in both experiments. 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. While 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, as s8 has a complex chord progression but a very simple melody. However, the experimental results from the other songs closely follow the results of dissimilarity by survey, clarifying the overall effectiveness of our method for utilizing chord-metric space and 3D visualization.

# VIII. CONCLUSION AND FUTURE WORKS

We proposed a music visualization and navigation system that can provide an intuitive visual retrieval method based in 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. 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.

# REFERENCES


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

[2] R. Type, F. Wiering, and Remco C. Veltkamp, "A Survey of Music Information Retrieval System," ISMIR 2005, pp. 153–160, 2005.

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

[4] R. B. Dannenberg, W. P. Birmingham, G. Tzanetakis, C. Meek, N.Hu, and B. Pardo, "The MUSART Testbed for Query-by-Humming Evaluation," ISMIR 2003, pp. 34–48, 2003.

[5] T. Sonoda, T. Ikenaga, K. Shimizu, and Y. Muraoka, "The Design Method of a Melody Retrieval System on Parallelized Computers," WEDELMUSIC 2002, pp. 66-73 , 2002.

[6] Heng-Tze Cheng, Yi-Husan Yang, Yu-Ching Lin, I-Bin Liao, and Homer H. Chen, "Automatic Chord Recognition For Music Classification And Retrieval," 2008 IEEE International Conference on Multimedia and Expo, pp. 1505-1508, Apil-June 2008.

[7] J. P. Bello, "Audio-based Cover Song Retrieval Using Approximate Chord Sequences: Testing Shifts, Gaps, Swaps And Beats," ISMIR 2007, pp. 239-244, 2007

[8] E. Gómez and J. Bonada, "Tonality Visualization of Polyphonic Audio," International Computer Music Conference 2005.

[9] E. Pampalk and M. Goto, "Musicrainbow: A new user interface to discover artists using audio-based similarity and web-based labeling," ISMIR 2006, pp. 367-370, 2006.

[10] P. Knees, M. Schedl, T. Pohle, and G. Widmer, "An Innovative Three-Dimensional User Interface for Exploring Music Collections Enriched with Meta-Information from the Web," 14th annual ACM international conference on Multimedia, pp. 17-24, 2006.

[11] S. Stober and A. Nürnberger, "MusicGalaxy: A Multi-focus Zoomable Interface for Multi-facet Exploration of Music Collections," CMMR 2010, pp. 273-302, June 2010