

# MusiCube: A Visual Music Recommendation System featuring Interactive Evolutionary Computing

**Yuri Saito**

Ochanomizu University  
2-1-1 Ohtsuka, Bunkyo-ku  
Tokyo 112-8610, Japan  
yuri@itolab.is.ocha.ac.jp

**Takayuki Itoh**

Ochanomizu University  
2-1-1 Ohtsuka, Bunkyo-ku  
Tokyo 112-8610, Japan  
itot@itolab.is.ocha.ac.jp

## ABSTRACT

We often want to select tunes based on our purposes or situations. For example, we may want background music for particular spaces. We think interactive evolutionary computing is a good solution to adequately recommend tunes based on users' preferences. This paper presents MusiCube, a visual interface for music selection. It applies interactive genetic algorithm in a multi-dimensional musical feature space. MusiCube displays a set of tunes as colored icons in a 2D cubic space, and provides a user interface to intuitively select suggested tunes. This paper presents a user experience that MusiCube adequately represented clouds of icons corresponding to sets of users' preferable tunes in the 2D cubic space.

## ACM Classification Keywords

H.5.2 Information Interfaces and Presentation: User Interface—*Graphical User Interfaces (GUI)*; H.5.5 Information Interfaces and Presentation: Sound and Music Computing—*Systems*; I.2.m Artificial Intelligence: Miscellaneous

## Author Keywords

Interactive Evolutionary Computing, music recommendation

## INTRODUCTION

Thanks to the evolution of multimedia technology, we can store a lot of tunes in music players and personal computers. Today, there are many services and researches on music recommendation, based on meta-data (e.g. title, artist name), annotation of musical score, acoustic information and these combinations.

On the other hand, we often want to select tunes based on our purposes or situations. For example, we may want to play background music at moody spaces (e.g.

room, cafe, and banquet) or while driving a car. We have our own aspect what kind of music is good as background music under such situations, and therefore it is difficult to automatically select tunes to satisfy everyone. To develop such purpose-based or situation-based music recommendation, we think machine learning or optimization techniques are useful to master the tendency of music selection by particular users.

We are often unconscious to musical features during selecting tunes based on purposes and situations. For example, we do not always think of key, tempo, or acoustic textures while selecting background music; however, the selection may have tendency of musical features, such as many of selected tunes are fast/slow and major/minor. It is fun if the tendency of the music selection is visualized. Such visualization makes users understand the tendency, and help their music selection easier.

This paper proposes MusiCube, a music selection interface based on characteristic of tunes considering users' purposes. MusiCube displays a set of icons corresponding to tunes in a cubic space like scatterplots. It assigns one of predefined four colors to the icons, where the colors corresponding to "positively listened", "negatively listened", "being suggested", and "not suggested yet". Each tune has multi-dimensional musical feature values, and users can select two of the dimensions which are assigned to X- and Y-axes of the scatterplots space.

Also, MusiCube applies interactive evolutionary computing, a method to optimize users' subjective evaluations, so that users can interactively input their preferences based on their purposes. It firstly suggests several tunes to be listened to, by switching colors of corresponding icons of the tunes. Users can listen to the tunes and input "positive" or "negative" evaluation. After inputting the evaluations, it selects other several tunes as next generation items, and again switches the colors of icons. Repeating these processes, MusiCube learns preferences of users and finally suggests tunes in a higher positive evaluation rate. It visualizes the distribution of positively or negatively evaluated tunes in the musical feature space, so the users can observe tendency of their music preferences from various viewpoints, and understand which features are dominant to their preferences.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

VINCI 2011, August 4–5, 2011, Hong Kong, China.

Copyright 2011 ACM 978-1-4503-0875-5/11/08...\$10.00.

## RELATED WORK

MusiCube is a kind of music recommendation and visualization software, and its visualization component is somewhat relevant to multi-dimensional data visualization techniques. This section introduces such techniques.

### Music Visualization and Recommendation

Many user interfaces of music collections have been already presented. "Islands of Music" deploys groups of tunes on geographic maps based on psychoacoustics models and self-organizing maps [6]. In fact, users explore tunes on a plane based on dimension reduction of musical features. On the other hand, MusiCube assigns particular features to X- and Y-axes without dimension reduction schemes, to visualize many tunes. As a result, users of MusiCube can understand that which feature is most important on group tunes reflecting purposes of the users.

Also, there have been many content-based music recommendation systems. Three aspect model [9] and tree-based vector quantization (Tree Q) algorithm [2] have been applied to music recommendation and content-based music retrieval. We think that such algorithms can be well integrated with MusiCube to improve the users' satisfaction of music recommendation.

### Multi-dimensional Data Visualization

There have been various multi-dimensional data visualization techniques, including Parallel Coordinates, VisDB, and Worlds within Worlds. Some other techniques apply heatmaps or glyphs to represent multi-dimensional value. Meanwhile, Scatterplots is one of the most popular techniques to visualize multi-dimensional data. Many scatterplots implementations directly assigns two or three of the dimensions to axes of the visualization spaces, while others apply dimension reduction techniques. Scatterplot matrices are often used for overview of scatterplots selecting arbitrary pairs of dimensions; however, it requires very large display spaces if number of dimensions is large. If users do not want to use such large display spaces for scatterplots, they may need to interactively switch the pairs of dimensions to understand correlations between the dimensions. Rolling the Dices [1] is one of the novel techniques to assist the interactive selection of dimensions for scatterplots.

Dimension analysis is helpful to obtain fruitful knowledge from multi-dimensional data visualizations. Sips et al. [3] presented a view selection technique of multi-dimensional data visualization by applying the dimension analysis. Nagasaki et al. [5] presented a correlation-based dimension selection technique for scatterplots-based visualization of credit card fraud data. The correlation-based strategy is also useful to reorder the dimensions and improve the readability of Parallel Coordinates and scatter plots matrices [7].

## SYSTEM DESIGN OF MUSICUBE

This section presents the processing flow and user interface design of MusiCube. Storing a lot of tunes, MusiCube firstly displays them as icons in a 2D musical feature space. It then randomly suggests several tunes to listen to, and evaluate them as "positive" or "negative" according to a user's purpose or situation.

The processing flow of MusiCube is as follows:

1. Calculate feature values of tunes.
2. Initialize the system.
3. Suggest several tunes by switching the colors of icons.
4. Receive user's evaluations, and switch the colors of icons of listened tunes.
5. Conduct the evolutionary computing to the next generation.
6. Repeat 3. to 5.

### Display of Icons

MusiCube displays icons corresponding to the tunes in a cubic space, as shown in Figure 1. Here, tunes have one of the following four statuses, and corresponding icons are colored by one of the following four colors:

- "**Positive**": Users have already positively evaluated that the tune matches to their purposes. Corresponding icons are colored by **red**.
- "**Negative**": Users have already negatively evaluated that the tune does not match to their purposes. Corresponding icons are colored by **blue**.
- "**Being suggested**": MusiCube currently suggests listening to this tune. Corresponding icons are colored by **orange**.
- "**Not yet**": The tune has not yet been evaluated or suggested. Corresponding icons are colored by **yellow**.

Meanwhile, we suppose that each tune has various musical feature values. MusiCube treats the musical features as multidimensional values, and calculates the location of the icons by assigning two of the features to X- and Y- axes, as various scatterplots techniques do.

### User Interface

Right side of the window of MusiCube contains three tabs, as shown in Figure 1. This section introduces user interface widgets featured by the three tabs.

#### Tab(a): play & stop, and evaluation

MusiCube expects that users listen to the tunes suggested by MusiCube, and subjectively evaluate them. Our implementation supposes users click icons colored by "orange" to start playing the suggested tune, because the orange icons correspond to the current generation of genes. Moreover, the tab features four buttons: "Stop", "Next", "Yes", and "No". Play of tunes stops when "Stop" is pressed, and MusiCube suggests

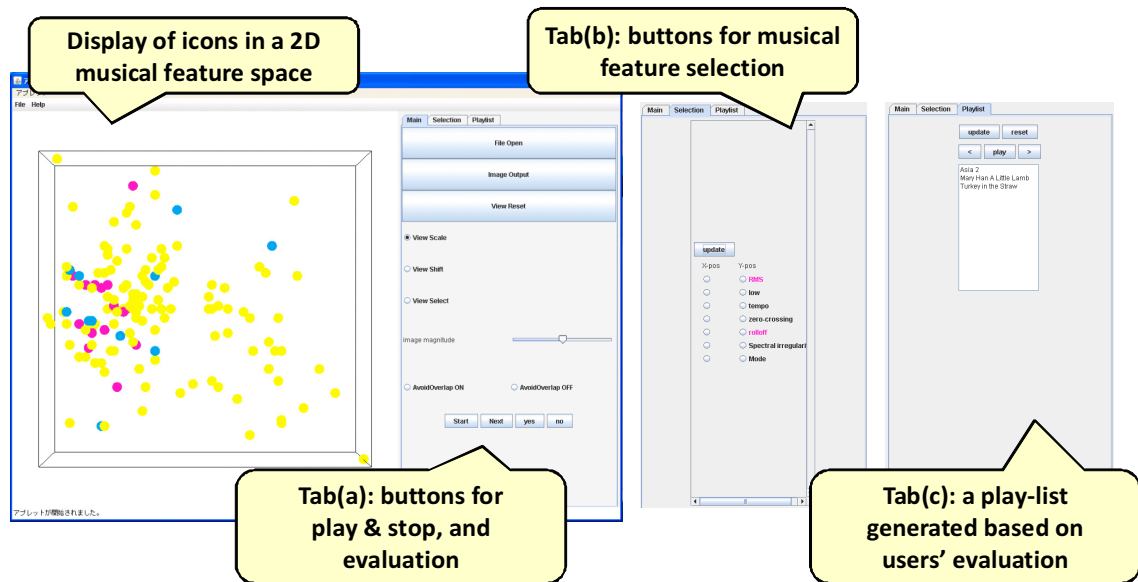


Figure 1. Window design of MusiCube. Left side of the window displays a set of colored icons corresponding to tunes. Right side of the window has three tabs.

another tune when "Next" is pressed. We suppose that users press "Yes" if played tunes match to their purposes; otherwise, users press "No".

MusiCube applies interactive evolutionary computing to select tunes to be recommended which are proper in users' purpose. Repeating these operations, MusiCube will learn the preferences of users, and be able to effectively recommend proper tunes. When sufficient numbers of users' evaluations are collected, users can stop the interactive evolutionary computing. At this moment users can quickly get proper new tunes, by choosing two features and looking at relations between evaluated tunes. Our implementation suggests users the best pair of features by calculating spatial entropy of evaluated tunes.

#### Tab(b): musical feature selection

MusiCube provides a user interface to choose two features to be assigned to X- and Y-axes. Once the users choose the features, MusiCube redeploys the icons by rotating the cubic space along the X- or Y-axes, similar to the rotation mechanism implemented by Rolling the Dice [1]. When a user selects a feature to the X-axis, MusiCube temporarily assigns the selected axis to the Z-axis, and then rotates the cubic space along the Y-axis, so that the XZ-plane turns to the XY-plane. Similarly, when a user selects a feature to the Y-axis, MusiCube temporarily assigns the selected axis to the Z-axis, and then rotates the cubic space along the X-axis, so that the YZ-plane turns to the XY-plane.

Here users can clearly look at the rotation process and the distribution of tunes from a different viewpoint.

At the same time, MusiCube can recommend desirable

pairs of features. Here, we suppose that a pair of features is desirable when red icons are concentrated on the display. When a user selects one of the features for the X- or Y- axis, our implementation of MusiCube highlights several other features that bring visualization results red icons are well concentrated. The technical detail of calculation of concentration ratio will be described below. Also, MusiCube features a button that automatically selects a pair of features for the X- and Y-axes that brings the best concentration ratio.

#### Tab(c): playlist

MusiCube provides a function to generate a playlist based on evaluation of users. We suppose that there are many preferable tunes around a positively evaluated tune. MusiCube has a function of automatic/manual playlist creation. As the automatic playlist creation, MusiCube collects tunes corresponding to red or yellow icons around the clicked icon, and displays a list of the collected tunes, when a user clicks a red icon corresponding to a positively evaluated tune. As the manual playlist creation, MusiCube inserts the tunes one-by-one specified by the users via click operation into the playlist. After creating the playlist, users can interactively play, add, and delete particular tunes in the playlist.

#### TECHNICAL DETAIL

This section describes technical detail of MusicCube, especially on our selection of musical features and implementation of interactive genetic algorithm.

#### Data Structure and Musical Features

We define that MusiCube deals with a collection of tunes  $T = \{t_1, \dots, t_N\}$ , where  $t_i$  is the  $i$ -th tune, and  $N$  is the number of tunes. Also, we define that a tune

as  $t_i = \{v_{i1}, \dots, v_{iM}, s_i\}$ , where  $v_{ij}$  is the  $j$ -th dimensional value of  $t_i$ ,  $M$  is the number of dimensions of the musical features, and  $s_i$  is an integer value which denotes the status of  $i$ -th tune, "positive", "negative", "currently suggesting", or "not yet".

Our current implementation uses features calculated by MIRtoolbox [4]. We had a feasibility study of features applying many sample tunes, and subjectively determined that the following 11 features were especially effective for our purpose.

**RMS energy:** Root-mean-square energy which represents the recommended volume of the tune.

**Low energy:** Percentage of frames whose energy is lower than the average energy.

**Tempo:** Tempo in beats per minute.

**Zero crossing:** Frequency of which the waveform takes zero value.

**Roll off:** Frequency which takes 85% of total energy, by calculating the sum of energy of lower frequencies.

**Brightness:** Percentage of energy of 1500Hz or higher frequency.

**Roughness:** Percentage of energy of disharmonic frequency.

**Spectral irregularity:** Variation of tones.

**Inharmonicity:** Percentage of energy of non-root tones.

**Mode:** Difference of energy between major and minor chords.

MusiCube deals with the feature values normalizing in a range [0, 1].

### Music Recommendation Using Interactive Evolutionary Computing

MusiCube applies Interactive Genetic Algorithm (iGA) in the normalized feature spaces. It applies principle component analysis (PCA) to reduce dimensions of the feature vectors before starting iGA, to avoid mislearning. We experimentally use 4 principle components to reduce the dimensions.

The following is the processing steps of our implementation of iGA.

**Initialization and Presentation:** MusiCube randomly generates the initial populations.

**Evaluation and Selection:** Users evaluate individuals (recommended tunes) by just pressing "Yes" button if they match to their purposes, or pressing "No" button, in the evaluation phase of iGA. We think this user interface is good to ease loads of users' psychological

tasks. Then, individuals evaluated as match to the purposes are defined as parent individuals, in the selection phase of iGA.

**Crossover:** MusiCube generates two children individuals from a pair of parent individuals. The midpoint between two parent individuals defines children individuals for each feature.

**Mutation:** MusiCube generates a random variable for each bit in a sequence. We set mutation probability as 10% in our implementation.

**Matching:** MusiCube selects individuals (tunes) which have the smallest Euclidean distances from the children individuals as the next-generation individuals (tunes), and recommends to users.

### Evaluation of pairs of features

We measure the concentration ratio by Entropy. MusiCube internally divides the display space into  $N_s$  rectangular subspaces, and count the number of red icons  $r_i$  and non-red icons  $q_i$  in the  $i$ -th subspace. MusiCube calculates the sum of Entropy  $E_{sum}$  in the subspaces as follows:

$$E_{sum} = \sum^{N_s} (p_{ri} \log p_{ri} + p_{qi} \log p_{qi})$$

$$p_{ri} = r_i / (r_i + q_i)$$

$$p_{qi} = q_i / (r_i + q_i) \quad (1)$$

Here,  $p_{ri}$  and  $p_{qi}$  are probabilities of red and non-red icons in the  $i$ -th subspace.

### USER EXPERIENCE

This section introduces our user experiences with MusiCube. We implemented MusiCube with Java JDK 1.6.0, and tested on an Lenovo ThinkPad T510 (CPU 2.4GB, RAM 2.0GB) running with Windows XP SP3. We used a collection of 143 tunes selected from RWC Music Database [8], including pop, rock, dance, jazz, latin, classical, march, and folk music. 13 subjects had experimental tests with MusiCube, where all of the subjects were university female students majoring computer science. We asked them the following processes:

1. Listen to the tunes suggested by the system.
2. Press "Yes" if they think the tunes are good to use as background music at a cafe, otherwise press "No".
3. After repeating 1. and 2., answer our questions.

### Examples

Figure 2(Left) and Figure 2(Center) are very similar results of two subjects. The pair of features "RMS energy" and "Roll off" brings the best concentration ratio in the both results. Also, distributions of red and blue icons in the two results are also very similar. These results suggest that the two subjects have very similar selections for background music at a cafe. On the other

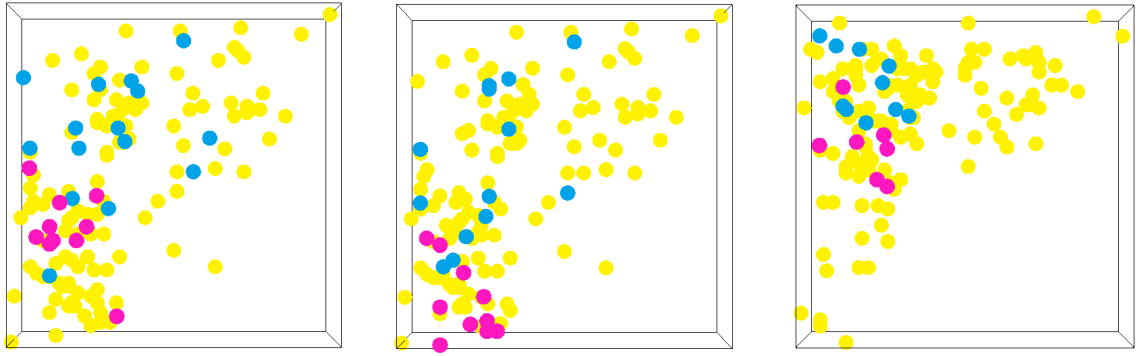


Figure 2. Examples. (Left)(Center) Result of two subjects. Features are "RMS energy" and "Roll off" in the both results. Also, distributions of blue and pink icons are very similar. These denote that preferences of the two subjects are very similar. (Right) Result of another subject. Features are "RMS energy" and "Inharmonicity", and the selected tunes are much different from those of other two subjects.

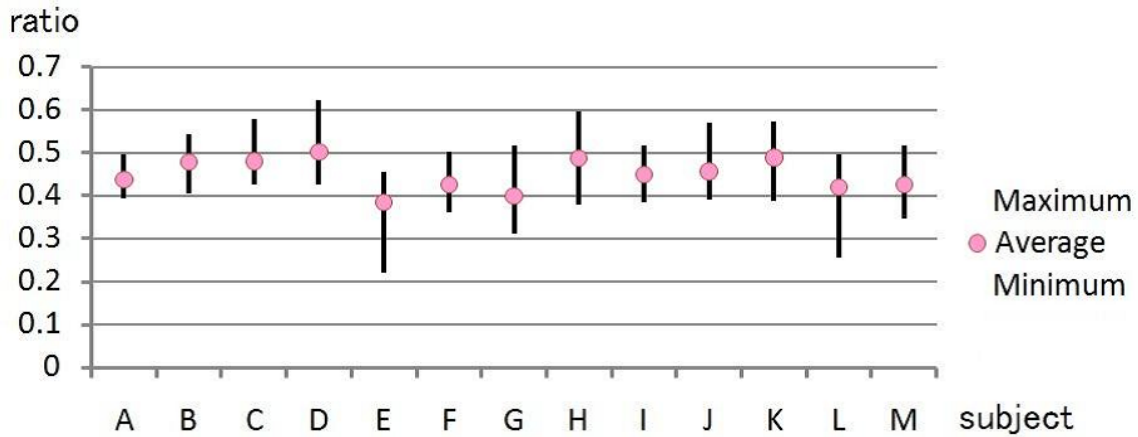


Figure 3. Entropy. Minimum, average, and maximum  $E_{sum}$  values from the results of 13 subjects(A-M).

hand, Figure 2(Right) shows a different result of another subject. The pair of features "RMS energy" and "Inharmonicity" brings the best concentration ratio in the both results. This result suggests that the selection of background music at a cafe depends on the sense of users, and therefore personalized music recommendation interface like MusiCube should be effective for this purpose.

The result of three subjects shown in Figure 2 also denote that icons of positively evaluated tunes are well concentrated on the display if two features are adequately selected. Once many red icons concentrate on the display, users can select other non-evaluated tunes around the cloud of red icons. This interactive selection may be better than the suggestion by iGA, because iGA often selects tunes far from the cloud of the red icons, and it is not always good selections for users. As a summary, we think visualization of music evaluation results by MusiCube is effective from the following two aspects:

**Interactive tune selection:** Users can freely select tunes around the icons which they have already posi-

tively evaluated.

**Notification of musical features:** Users will be notified what kind of musical features affects their selections.

We calculated the sum of Entropy  $E_{sum}$  by Equation 1 with arbitrary pairs of features. We then determined minimum, average, and maximum  $E_{sum}$  values from the results of all subjects. Figure 3 shows the minimum, average, and maximum values of 13 subjects. This result denotes that  $E_{sum}$  value gets significantly small if two features are adequately selected.

### Subjective Evaluation

We asked the subjects to answer the following questions after playing with MusiCube:

- Q1: How many tunes in the created playlist do you agree to play as background music at a cafe?
- Q2: Do you feel MusiCube is effective as a GUI of music recommendation? (5-grade evaluation: 5 is very good, 1 is very poor.)
- Q3: How MusiCube is effective?

**Q4:** How MusiCube can be improved?

We calculated the ratio of number of positively evaluated tunes in Q1 against the total number of tunes in the playlist for each subject. Maximum, average, and minimum ratios were 1.0, 0.7, and 0.27. We think the result is totally good except the ratio of just one subject was under 0.5. Average of the evaluation of subjects in Q2 was 4.15. The result also denotes the effectiveness of MusiCube.

We received the following answers for Q3:

- It is easy and convenient to collect preferable tunes because corresponding icons concentrates very well.
- This kind of visual recommendation is good for passive listeners who are not eager to look for preferable tunes.
- Ratio of preferable tunes in the suggested tunes was surprisingly good.

These comments suggest that MusiCube is subjectively effective for users.

We received the following answers for Q4:

- It may be useful if MusiCube indicates metadata of each tune.
- Overlap of icons may prevent the usability.
- Evaluation may be influenced by the visualization result.

We would like to enhance MusiCube based on the above comments as short-term future works.

#### CONCLUSION AND FUTURE WORK

This paper presented MusiCube, a visual interface for music selection. MusiCube displays icons corresponding to tunes in a 2D space assigning two features to X- and Y-axes. Users can input evaluation for listened tunes, and MusiCube learns their preferences applying interactive genetic algorithm. It displays positively evaluated tunes in a particular color, and therefore users can create a preferable playlist by collecting icons in a region which positively evaluated tunes concentrate.

As short-term future works, we would like to have more user experiments with more tunes and more subjects. Also, we would like to enhance the implementation of MusiCube as discussed in the section of subjective evaluation.

Icon layout is another interest. Our current implementation does not apply dimension reduction schemes because we intended to indicate and notify the features affected to preferences of users. However, it is often better to distribute the icons in a display space applying dimension reduction schemes. We would like to apply a linear discriminant analysis scheme so that we can concentrate positively evaluated tunes while separating from other tunes. Also, we would like to implement a

mechanism to deal with multiple icons overlapped on the display space.

#### REFERENCES

1. N. Elmqvist, P. Dragicevic, and J. Fekete. Rolling the dice: Multidimensional visual exploration using scatterplot matrix navigation. *IEEE transactions on Visualization and Computer Graphics*, 14(6):1141–1148, November 2008.
2. K. Hoashi, K. Matsumoto, and N. Inoue. Personalization of user profiles for content-based music retrieval based on relevance feedback. *Proc. of 11th ACM Int'l Conference on Multimedia*, pages 110–119, 2003.
3. B. N. M. Sips, J. P. Lewis, and P. Hanrahan. Selecting good views of high-dimensional data using class consistency. *Computer Graphics Forum*, 28(3):831–838, 2009.
4. MIRtoolbox.  
<http://www.jyu.fi/hum/laitokset/musiikki/en/research/coe/materials/mirtoolbox>.
5. A. Nagasaki, T. Itoh, M. Ise, and K. Miyashita. A correlation-based hierarchical data visualization technique and its application to credit card fraud data. *1st International Workshop on Super Visualization (in conjunction with the 22nd ACM International Conference on Supercomputing)*, 2008.
6. E. Pampalk. Islands of music: Analysis, organization, and visualization of music archives. *Master's thesis, Vienna University of Technology*, 2001.
7. W. Peng, M. O. Ward, and E. A. Rundensteiner. Clutter reduction in multi-dimensional data visualization using dimension reordering. *IEEE Symposium on Information Visualization*, pages 89–96, November 2004.
8. RWC-Music-Database.  
<http://staff.aist.go.jp/m.goto/RWC-MDB/>.
9. K. Yoshii, M. Goto, and H. Okuno. Hybrid collaborative and content-based music recommendation. *CrestMuse Symposium 2008*, pages 27–28, 2008.