

MusiCube: A Music Selection Interface featuring Interactive Evolutionary Computing in Feature Spaces

Yuri Saito*
Ochanomizu University

Takayuki Itoh†
Ochanomizu University

1 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. We think that music recommendation systems based on characteristic of tunes and reflection of users' choices will be more useful, if we can select many tunes based on our purposes.

This poster 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. 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. As a result, MusiCube adequately recommends tunes to users by colors of icons, from the results of interactive evolutionary computing. The users can observe tendency of their music preference from various viewpoints, as if users roll the cubic space studded with tunes.

2 RELATED WORK

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 [1]. 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 [2] tree-based vector quantization (Tree Q) algorithm was [3] 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.

3 USER INTERFACE OF MUSICUBE

3.1 Icon Display

MusiCube displays icons corresponding to the tunes in a cubic space, as shown in Figure 1(Left). Here, four colors of icons denote the following meanings:

Red: Users have already evaluated the tune as matches to their purposes.

Blue: Users have already evaluated the tune as matches to their purposes.

* e-mail: yuri@itolab.is.ocha.ac.jp

† e-mail: itot@is.ocha.ac.jp

Orange: MusiCube is currently recommending to listen to this tune.

Yellow: The tune has not been evaluated or recommended yet.

Meanwhile, we suppose that each tune has various musical feature values. Section 4.1 describes the detail of the musical features. 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.

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 rotation function of XY- and XZ- planes, as shown in Figure 1(Right). Here users can clearly look at the rotation process and the distribution of group turn from a different viewpoint.

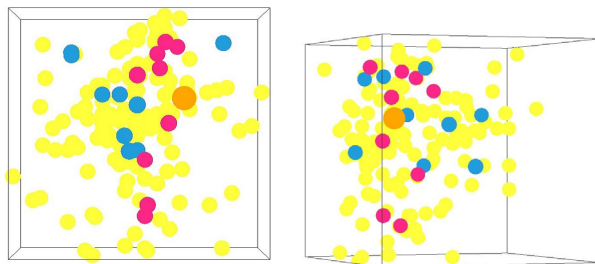


Figure 1: Example. (Left) MusiCube deploys the icons in a cubic space. (Right) Rotation process.

3.2 Music Recommendation

MusiCube expects that users listen to the tunes recommended by MusiCube, and subjectively evaluate them. Our implementation indicates two buttons, "Yes" and "No": we suppose that users press "Yes" button if they match to their purposes, otherwise press "No" button.

MusiCube applies interactive evolutionary computing to select tunes to be recommended which are proper in users' purpose. Figure 2 illustrates the loop of evaluation and recommendation processes. 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.

3.3 Playlist Generation

MusiCube automatically generates a playlist with evaluation of turns.

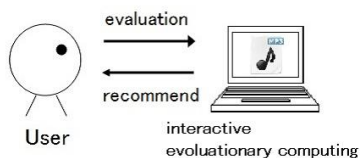


Figure 2: recommendation process.

4 TECHNICAL DETAIL

4.1 Data Encoding

Current our 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 (RMS energy, Low energy, Tempo, Zero crossing, Roll off, Brightness, Roughness, Spectral irregularity, Inharmonicity, Key, and Mode), shown in Table 1, were effective for our purpose. MusiCube has a pre-processing step including normalization of feature vectors of tunes.

Table 1: Musical features.

Feature	Explanation
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.

4.2 Music Recommendation Using Interactive Evolutionary Computing

MusiCube applies Interactive Evolutionary Computing (iGA) in the normalized feature spaces. Figure 3 shows the processing flow of iGA in our implementation. It applies principle component analysis (PCA) to reduce dimensions of the feature vectors before starting iGA, to avoid mislearning.

4.2.1 Initialization and Presentation

MusiCube randomly generates the initial populations.

4.2.2 Evaluation and Selection

Users evaluate individuals (recommended tunes) by just pressing "Yes" button if they match to their purposes, or pressing "No" but-

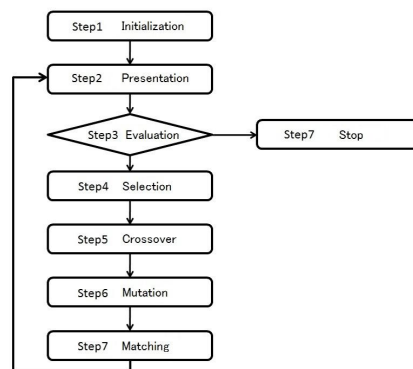


Figure 3: Processing flow.

ton, 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.

4.2.3 Crossover

MusiCube generates two children individuals from a pair of parent individuals.

4.2.4 Mutation

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

4.2.5 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.

$$d = \sqrt{\sum_{i=0}^n (f_i - p_i)^2} \quad (1)$$

f : feature vector of child individual. p : feature vector of a tune.

5 CONCLUSION AND FUTURE WORK

This poster presented MusiCube, a music selection interface based on features of tunes. We point out the following as potential future works: user tests, satisfaction enhancement of recommendation results, and improvement of GUI.

REFERENCES

- [1] E. Pampalk, Islands of Music: Analysis, Organization, and Visualization of Music Archives. Master's thesis, Vienna University of Technology, (2001).
- [2] K. Yoshii, M. Goto, H. Okuno, Hybrid Collaborative and Content-based Music Recommendation. CrestMuse Symposium 2008, pp.27-28 (2008).
- [3] K. Hoashi, K. Matsumoto, N. Inoue, Personalization of user profiles for content-based music retrieval based on relevance feedback, Proc. of 11th ACM Int'l Conference on Multimedia, pp.110-119 (2003)
- [4] O. Lartillot. MIRtoolbox. <http://www.jyu.fi/hum/laitokset/musiikki/en/research/coe/materials/mirtoolbox>.