MIST: A Music Icon Selection Technique Using Neural Network

Mizuho Oda [*]
Graduate School of Humanitics and Sciences,
Ochanomizu University

Takayuki Itoh [†]
Graduate School of Humanitics and Sciences,
Ochanomizu University

Abstract

Thanks to recent evolution of multimedia technology, today we often use computers as music recorders and players. Number of tunes stored in our computers is monotonically increasing, and therefore users often face difficulty while selecting tunes which they want to listen. To solve the problem, we aim visual selection of tunes based on their impression.

This paper proposes a technique for automatically selecting icons for music files. Main problem of the proposed technique is matching of tunes and images based on their impressions, where the images are used as icons. The technique first extracts features from images and tunes, and calculate fitness of sensitivity words using Neural Network (NN). Forming multi-dimensional vectors from the fitness values, the technique calculates Euclidian distances from tunes to images. The technique finally selects the closest images as the most matched images for each tune. We think users can visually recognize impressions of a set of tunes, and easily select preferable tunes, if the icons selected by the technique are displayed in folders of file systems.

1 Introduction

Today, we often listen to the music using computers, thanks to the evolution of multimedia technology. Moreover, the Internet becomes a major media to purchase music. We often play the music downloaded via the Internet on the computers or portable players. Increasing capability of hard disk drives and main memories of computers, we often download huge number of tunes onto our personal computers.

Since several decades ago, we have listened music using electric media such as compact disks or cassette tapes. Operations for selecting tunes were quite simple while using such media, because many of such media record tunes played by the same musician. When a user picks up a compact disk or a cassette tape, the action means that the user has already narrowed down the selection of music. After inserting the media into a player, the user's operation will be only the simple selection of tracks.

Comparing to the operation on the compact disk or cassette tape players, we can freely select music on the computers. We usually select tunes by looking their titles or names of musicians, but sometimes the operation may be troublesome. There are many interface products to assist the selection of music; for example, several vendors provide interface to select tunes according to their mood or impression, such as “Playlist” in iTunes. However, such interface usually displays only character information such as titles and players, so it may be sometimes difficult to intuitively select the tunes from the huge number of the tunes, according to users' preferable moods.

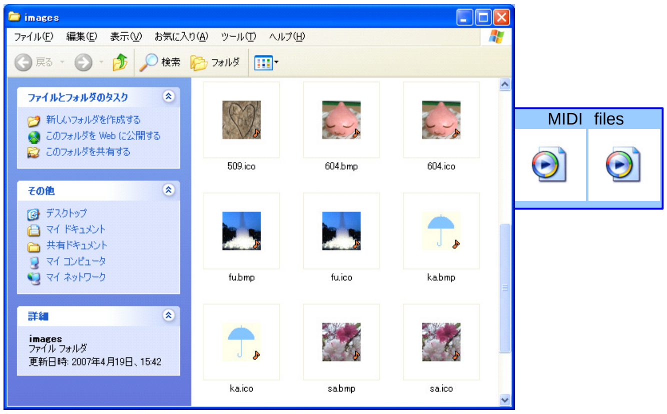

Figure 1: Using arbitrary images as icons.

In this paper, we propose a technique for automatically matching tunes and images based on their impressions. We think the technique can be applied to automatic selection of expressive icons for music files, as shown in Figure 1, and therefore users can easily select the tunes when the icons are displayed in folders of file systems. Here we allow using photographs as icons, because it may make variety of icons wider, creation of icons more flexible, and selection of tunes easier.

In this technique, we prepare multiple sensitivity words [1] to unify the expression of impression between tunes and images. The technique estimates fitness values of sensitivity words from feature values of tunes and images. The fitness values are n-dimensional vectors, where n is the number of sensitivity words, and therefore the technique can match

[*] e-mail: miz-oda@itolab.is.ocha.ac.jp
[†] e-mail: itot@computer.org

tunes and images by calculating distances between them and extracting the closest image for each tune. Here we use Neural Network (NN) for learning relativity between the fitness values and feature values, and apply Euclidean distances for calculating distances between tunes and images.

Here, our technique does not aim maximum matching of image and music. In other words, we do not assume one-to-one matching of image and music. Rather than that, we think similarly impressed tunes must match to the same image. Therefore, our technique matches image and tunes by only calculating distances of fitness of sensitivity words.

Remark that the target of the technique is matching of tunes and images based on personal feeling of users, not based on average of impressions learned from statistics of questionnaire or experiments. We target that the users of our technique will be personal users who store thousands of tunes on their personal computers. Since the technique focuses on personal feeling rather than average feeling, we think effectiveness of our technique will not depend on age, gender, feeling, culture, country, and so on. Title of this paper "MIST" comes from Music Icon Selector Technique (MIST) presented in our previous paper [2], and this paper introduces some improvements against the previous work.

2 Related Work

This section introduces several papers related to the proposed technique, including automatic creation and selection of icons, color selection based on impression of music, sensitivity word selection for music and images, and matching of music and images.

Automatic Creation and Selection of Icons

Semanticons [3] is a technique for automatically synthesizing icons based on the semantics of files, so that users can intuitively understand the contents of the files. However, the method must suppose that the system can extract semantics of files, including messages of images and music, as structured text information. The paper [3] does not mention how to extract the features and impression of media files.

Kolhoff et al. focused on icon selection for music files [4]. Though it only deforms predesigned glyphs according to features extracted from MIDI files. Against this limitation, our technique aims to use every design of icons since we allow using arbitrary pictures as icons.

Color Selection Based on Impression of Music

Gotoh et al. proposed Musicream [5], a music recommendation system that represents a list of tunes colored based on their mood, and automatically places the metaphor of tunes so that similar tunes get closer on the display. Kawanobe et al. [6] also expressed impression of music using 3 colors. Our technique also considers of impression of colors and music, however, the number of colors used for single tune is unlimited, because it allows using arbitrary icons.

Sensitivity Word Selection for Music and Images

Ikezoe et al. proposed a technique for estimating musical impression using sensitivity words [1]. The technique selects sensitivity words extracted from Semantic Differential Method according to impression of tunes, and leads fitness of sensitivity words for the tunes. This technique is relative to our technique because it calculates fitness values of sensitivity words using NN. Takahashi et al. presented a similar approach [7] that leads sensitivity words from music. Our technique is a kind of their extension, because it calculates fitness of sensitivity words both for music and images.

Matching of Music and Images

Ohyama et al. proposed DIVA [8], which automatically arrange tunes based on impression of images. It is relative to our technique because it calculates distances between aranges and images based on feature values and keywords of images. However, DIVA does not consider of features and keywords of music.

Our previous version of MIST [2] first determines relativity between feature values and fitness of sensitivity words, then formalizes their relativity using Response Surface Method (RSM), and finally matches tunes and images by calculating their distances in the space of fitness of sensitivity words. We found that the previous version of MIST was not very useful, because it requires bothering manual preprocess before using it. One of the biggest problems was that we used RSM for learning relativity between feature values and fitness of sensitivity words. RSM is a technique to define quadric surface that passes neighborhood of a set of given parameters x and responses y, by calculating the optimal coefficients β in the following equation:

$$y_k = \beta_0 + \sum \beta_1 x_i + \sum \beta_2 x_i^2 + \sum \beta_3 x_j + \sum \beta_4 x_j^2 + \sum \beta_5 x_i x_j \quad (1)$$

The previous version of MIST required $(n+1)!$ equations to calculate β, if tunes or images has n features. For example, if our implementation extracts 5 features from tunes or images, that means $n = 5$ and therefore we need $(5+1)! = 120$ equations. In other words, we must deal with 120 sample tunes or images for learning. It is very bothering, and therefore not practical. On the other hand, the current version of MIST presented in this paper allows smaller number of tunes or images for leaning, since the current MIST applies NN. Also, the current MIST refers keywords of tunes and images as well as their feature values, against the previous version only referred feature values.

3 Implementation

Figure 2 shows the processing flow of the current version of MIST, which consists of preprocess and actual use steps.

Preprocess step of MIST prepares several sample images and tunes, where we assume that they have their own several keywords. MIST calculates their feature values, where the

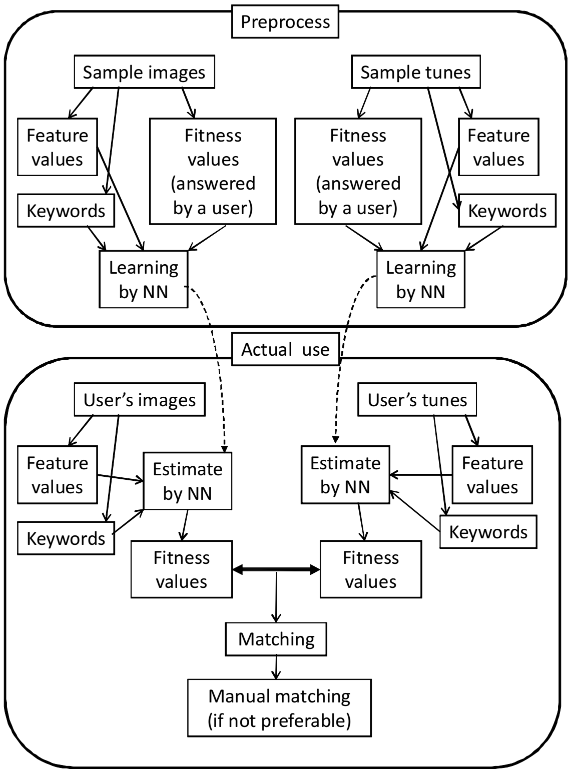

Figure 2: Process flow of MIST.

example of feature values is shown in Table 1. Simultaneously, MIST requires fitness values of sensitivity words answered by a user, where sensitivity words we used are shown in Table 2. MIST then learns relativity among feature values, keywords, and fitness values, using NN.

Actual use step of MIST assumes that users bring their own images and tunes in order, and the images and tunes may have their own keywords. MIST calculates their feature values, and then estimates their fitness values of sensitivity words by NN. It then calculates Euclidian distances between a tune and all images in the space of fitness values, and finally matches the closest image as the icon of the tune. If the selection is not preferable for the user, MIST allows manual matching of his/her preferable image to the tune. When manual matching is done, or the user brings new images or tunes, MIST may return to the preprocess step to update the learning result.

The following sections describe about the processing units of MIST.

3.1 Feature values of Images

MIST assumes that tunes and images have their own keywords. We assume that keywords of images are name of objects shot in the images, such as ”flower” and ”cloud”.

Preparing such images, MIST automatically calculates feature values shown in Table 1. We currently use values of several representative colors for each image as the feature values. Our implementation first applies posterization for the images, to retouch the images using only predefined colors. It then calculates the areas of each predefined color in the images, and specifies several colors that have maximum areas. The current implementation specifies the largest two colors, and uses the six values of the two colors as feature values of the images.

Here, many image recognition techniques apply non-RGB color system, but other color systems such as LUV or YCbCr. These color systems are often preferred because it separates intensity component as an independent variable, such as L in LUV, or Y in YCbCr. It is reasonable to recognize pictures in many cases, for example, to recognize the two pictures which shoot the same scene but only lighting condition is different.

Current our implementation uses YCbCr color system, where Y means intensity, Cb means signal of difference for blue color, and Cr means signal of difference for red color. YCbCr color system is useful since it can be easily converted from RGB color values by the following equations:

$$\begin{aligned} Y &= 0.29891 \times R + 0.58661 \times G + 0.11448 \times B \\ Cb &= -0.16874 \times R - 0.33126 \times G + 0.50000 \times B \\ Cr &= 0.50000 \times R - 0.41869 \times G - 0.08131 \times B \qquad (2) \end{aligned}$$

3.2 Feature values of Music

MIST requires keywords to calculate feature values for tunes, as well as images. We assume to use musical terms, such as ”March” and ”Waltz”, as keywords of tunes.

Current our implementation extracts the following values as feature values of tunes:

- tempo
- average pitch
- number of musical notes per a time
- Ratio of rest
- key

“ Average pitch ” is calculated by dividing the sum of all musical pitch values by number of musical notes.

“ Number of musical notes per a time ”is calculated by dividing the sum of lengths of notes by number of beats, which means average of density of notes. Specifically, if a tune forms four-quarter time, and there are four quarter notes during one measure, the value is 1. Under the same conditions, if there is one whole note during one measure, the value is 0.25. If there are sixteen sixteenth notes, the value is 4.

“ Ratio of rest ”is calculated by dividing the sum of length of rest notes by the length of whole measures.

Currently, our technique uses only one phrase of the music to extract feature values. We assume that tempo or key does not change during the target part of the music used for extracting features.

Table 1: List of feature of tunes and images

images	tunes
Y	tempo
Cb	average pitch
Cr	number of musical notes per a time
	Ratio of rest
	key (moll/due)

Table 2: List of sensitivity words

sensitivity	
bright	dark
weighty	fragile
hard	soft
stable	unstable
clear	murky
smooth	articulate
agitato	tranquil
thick	thin

3.3 Calculation of Fitness Values of Sensitivity Words from Feature Values and Keywords

In the preprocess step, MIST consumes his/her inputs of fitness and sensitivity words for the prepared tunes and images, where it may be convenient to use selection of scales of 1 to 7 or so. Remark that we must use the same set of sensitivity words for tunes and images. Current our implementation applies 8 items of sensitivity words which are also used in [7]. Remark that MIST requires using the same sensitivity words for tunes and images.

MIST then learns relativity between the fitness values and feature values or keywords by using NN, which is one of the famous supervised learning algorithms. NN is categorized into two structures, which are feed-forward NN and cross-coupled NN. MIST uses feed-forward NN, and applies Back Propagation (BP) algorithm for the learning step.

Supposing that input is feature values of tunes or images o, and output is fitness of sensitivity words y, we formalize the function as follows:

$$y = f(\sum_{j=1}^{n} w_j o_j - \theta) \qquad (3)$$

Here w denotes union weights, and θ denotes a constant threshold value. Detailed implementation of NN is described in [9].

The preprocess step of MIST corresponds to optimize the set of w values in equation 3, and the actual use step of MIST corresponds to calculate y values to estimate the fitness of sensitivity words.

As described above, current version of MIST needs a learning step as well as the previous version of MIST; however, in our experiments the current version of MIST could start the learning step with only ten sample images or tunes, though images and tunes had five or six features. One more improvement of the current version of MIST is variety of features. The previous version requires continuous variables as features; however, in the current version can apply binary variables as features.

Figure 3: Original and marked images.

3.4 Calculation of Distances from Tunes to Images

MIST treats the fitness values as high-dimensional vectors, to calculate Euclidean distances between tunes and images, and finally select images for each tune. Here we allow selecting same images for multiple tunes; because the result may show information that there are multiple tunes that have similar impression.

Current our implementation uses eight sensitivity words, so the distance is calculated in an 8-dimentional space. Here we denote the fitness of sensitivity words calculated by NN from an image as g_1, g_2, ..., and g_8. Also, we denote the fitness of sensitivity words calculated by NN from a tune as m_1, m_2, ..., and m_8. The distance from a tune to an image is calculated by the following equation:

$$distance = \sqrt{\sum_{k=1}^{8} (m_k - g_k)^2} \qquad (4)$$

The current implementation calculates distances from a tune to all images, and specifies the image closest from the tune. Fitness values calculated by our implementation of NN vary between 0 and 1, and therefore distances are between 0 and $2\sqrt{2}$.

MIST has a capability to manually replace images if the selection is not preferable for users. We would like to develop a feedback of manual replacement for improvement of learning results, as a future work.

3.5 Addition of New Tunes and Images

Users usually add tunes and images to their computers anytime, so MIST allows adding them anytime. It can calculate the fitness values and select an image for the added tunes anytime. It also can select the added images anytime, if users would like to run the selection process again.

We assume that image retouch tool is useful to make variety of images wider. When retouched images are prepared, we assume that name of retouch methods are used as keywords, and retouch parameters are used as a part of feature values, to effectively reflect the retouch to the selection of images. We applied several retouch methods to create differently impressed images, including binarization, averaging filter, gamma correction, negative, Laplacian filter, and posterization. We tried several other filters such as weighted averaging filter and Gaussian filter, but we did not apply them because we are not differently impressed by the images applied them. We suppose that one reason was images for icons are too small for these filters.

We think that music-specific marks are needed for icons, because MIST has a capability to freely input arbitrary images. We suppose to mark the icons while preparing them. Figure 3 shows original and marked images.

3.6 Visualization system

Many of current window systems assign same icons for files which have the same extension. We do not assume to apply MIST to window systems themselves to fundamentally change the selection of icons. We aim to develop particular applications that can intuitively visualize large number of tunes in the windows of music players or streamers.

By the way, many file system viewers bundled by operating or window systems display only icons of files under the single folder. On the other hand, many users store music files categorizing according to various criteria, such as artists and day of the purchase, and divide into many folders. Thus, if contents those folders could be visualized at the same time, we can effectively use MIST because we can discover similarly impressed tunes in many folders.

It is effective to display music icons in multiple folders in one display space, by using the visualization technique HeiankyoView [10]. Figure 4 shows an example of visualization results with HeiankyoView. It represents files as icons, and folders as nested rectangular borders. At the same time, MIST assigns the same icon to similarly impressed tunes. By using such visualization technique, users can simultaneously discover many similarly impressed tunes in many folders.

We think such visualize system can be applied not only for browsing music stored in the file systems of personal computers, but also as music recommendation systems of large-scale music markets.

4 Experiments

4.1 Preparation for Experiments

We implemented MIST on Windows XP and GNU gcc 3.4, and experienced with 5 examinees.

Here, we prepared multiple MIDI files as tunes. Since our most interest is musical feature values rather than lyrics, we used instrumental classical music. Also, we cut tunes to apply only their beginning parts, because current our implementation does not consider of changes of keys or tempo.

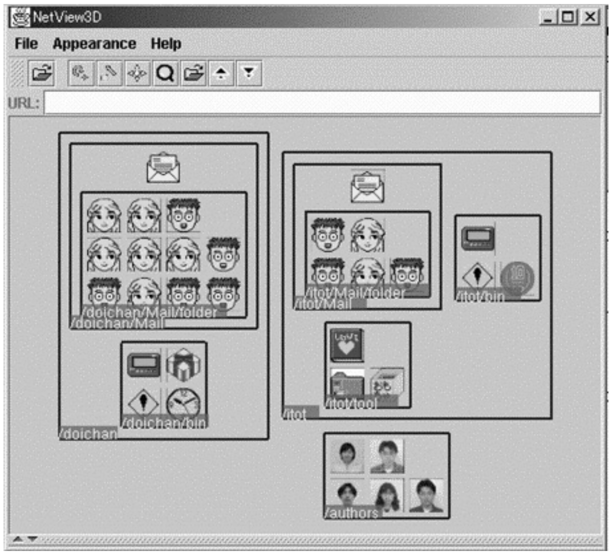

Figure 4: HeiankyoView.

Table 3: Answers of an examinee.

tune	image	distance	satisfaction
1	8	0.3674	Yes
2	8	0.3905	Yes
3	3	0.6616	Yes
4	3	0.6292	Yes
5	3	0.5197	Yes
6	8	0.3451	Yes
7	8	0.6788	Yes
8	8	0.3350	No
9	3	0.2795	Yes
10	3	0.4890	No
11	3	0.3697	Yes

We also prepared multiple BMP files as images. We cut them as 48×48 pixels because we aimed to use MIST as automatic icon selector technique.

4.2 Results

We had the experiments using 10 sample images and 11 sample tunes for learning, and other 10 images and 11 tunes for actual use. Table 3 shows the results of one examinee. This table shows IDs of tunes, IDs of images matched to the tunes, distances between the tunes and images. We asked the satisfaction of matching results to the examinee as “ yes ” or “ no ”. The answers of the examinee are also shown in Table 3.

Here, we define the ratio of satisfaction as the number of satisfied matching results divided by the total number of tunes (11 in these experiments). For example, Table 3 de-

Table 4: Result of 5 examinees.

examinee	ratio of satisfaction(%)	maximum distance
A	91	0.8726
B	82	0.6788
C	64	0.7527
D	55	0.9253
E	45	0.6172

notes that the ratio of satisfaction of an examinee was 82%, and correlation between satisfaction and distances was not strong. Some tunes had large distances to the matched images, but the examinee was satisfied by the matching results. On the other hand, some other tunes had small distances to the matched images, but the examinee was not satisfied.

We had the similar experiments with 5 examinees who listen to the music every day. Table 4 shows the result of the 5 examinees, including maximum distances of the matched tunes and images. It shows that the ratio of satisfaction strongly depended on examinees, and the improvement is one of our future works.

Table 4 also denotes that the maximum distances strongly depend on the examinees. This difference will be reduced by applying normalization of fitness of sensitivity words; however, we think it does not guarantee to improve the satisfaction, especially when the impression of prepared tunes and images are totally different. We do not attempt to minimize the distances by the normalization, but recommend users to freely add preferable images as additional icons.

5 Conclusion

This paper proposed MIST, which matches images to the tunes based on their impressions, for automatic icon selection. Against the previous version of MIST used Response Surface Method (RSM) for learning of relativity between features and fitness, the current version of MIST uses Neural Network (NN) for the learning. We think that MIST makes users easier to select tunes based on their impressions.

As a future work, we would like to use sound files such as MP3, instead of MIDI. Also, we would like to improve the learning process, since we currently use a simple NN program which contains only one middle layer.

We think that we can extend this study to realize the following systems.

Current our implementation uses only one beginning phrase of a tune to extract features, and matches only one image for the tune. If the extended implementation of MIST can match multiple images according to the change of impression along the progress of the tune, we can develop a system to provide animations matching to the tunes. One problem of the extension is that we will often demand a technique that smoothly changes the images according to the smooth change of features of tunes.

Another possible extension is selection of tunes for images Current MIST specifies the closest image for each tune; however, we can inverse the process. There will be some applications of such extension of MIST, such as automatic background music (BGM) selector for images or movies.

Acknowledgements

We would like to thank Prof. Ichiro Kobayashi of Ochanomizu University, for his overall suggestions.

References

[1] T. Ikezoe, Y. Kajikawa and Y. Nomura: “Music Database Retrieval System with Sensitivity Words Using Music Sensitivity Space”, Journal of Information Processing Society of Japan, Vol. 42, No. 12, pp. 3201-3202 (2001).

[2] M. Oda and T. Itoh: “ MIST: A Method for Automatically Selecting Icons from Musical Impressions ”, NICOGRAPH Autumn Conference (2006).

[3] V. Setlur, C. Albrecht-Buehler, A. A. Gooch, S. Rossoff and B. Gooch: “Semanticons: Visual Metaphors as FIle Icons”, Computer Graphics Forum (ERUROGRAPHICS 2005) Vol. 24, Num. 3, pp647-656 (2004).

[4] P. Kolhoff, J. Preub and J. Loviscach: “Music Icons: Procedural Glyphs for Audio Files”, SIBGRAPI’06 IEEE (2006).

[5] T. Goto and M. Goto: “Musicreams: New Music Playback Interface Where Musical Pieces Can Be Streamed, Sticked, and Sorted”, WISS2004, pp.53-58 (2004).

[6] M. Kawanobe and M. Kameda: “Media Conversion between Music and Color Combination Considering Time-series Changing of Musical Impression”, The Journal of the Society for Art and Science, Vol. 5, No. 4, pp.95-105 (2006).

[7] M. Takahashi and I. Kobayashi: “An Approach to Music Recommendation based on Image”, The 69th Information Processing Society of Japan National Convention Report, N3-3 (2007).

[8] K. Ohyama and T. Itoh: “DIVA: An Automatic Music Arrangement Technique Based on Impressions of Images”, SmartGraphics 2007 (2007).

[9] T. Agui and T. Nagao: “Introduction to Image Processing Using Programming Language C”, ISBN4-7856-3124-4 C3055 (2000).

[10] T. Itoh, H. Takakura, A. Sawada, and K. Koyamada: “Hierarchical Visualization of Network Intrusion Detection Data in the IP Address Space”, IEEE Computer Graphics and Applications, Vol. 26, No. 2, pp. 40-47 (2006).