Melody Retrieval by Self-Organizing Map with Refractoriness from Key Phrase Given by MIDI Keyboard

Hirotaka ICHIKAWA, Manabu GOMADA and Yuko OSANA

School of Computer Science, Tokyo University of Technology
1404-1 Katakura, Hachioji, Tokyo, 192-0982, Japan Email: osana@stf.teu.ac.jp

Abstract—In this paper, we propose a melody retrieval system using self-organizing map with refractoriness. This system can retrieve melodies from the key melody which is given by MIDI keyboard. In the proposed system, the stored melodies are divided into phrases composed of four bars, and the features on pitch and phonetic value are employed. Since the proposed system is based on the self-organizing map with refractoriness, the plural neurons in the map layer corresponding to the input key melody can fire sequentially because of refractoriness of neurons. We confirmed that the proposed system can retrieve correct melodies from the key melody which is given by MIDI keyboard in 98.5% cases.

1. Introduction

Recently, some melody retrieval systems which make use of flexible information processing ability of neural networks have been proposed. The conventional methods for music retrieval generally can be classified into the two groups; (a) retrieval from title, composer, words and so on, and (b) retrieval from a part of melody. As the music retrieval methods from a part of melody, some systems has been proposed[1]–[6]. For example, in the system in ref.[6], although plural melodies corresponding to the key melody, the key melody has to be expressed in a specific format which is called ABC notation using the ASCII character set.

In this paper, we propose a melody retrieval system by self-organizing map with refractoriness which can retrieve melodies from the key melody which is given by MIDI keyboard. In the proposed system, the stored melodies are divided into phrases composed of four bars, and the features on pitch and phonetic value are employed. Since the proposed system is based on the self-organizing map with refractoriness[7], the plural neurons in the map layer corresponding to the input key melody can fire sequentially because of refractoriness of neurons.

2. Features of Melodies

In the proposed system, pitch and phonetic value and keyword (genres of music) are used as features of melodies.

2.1. Features on Pitch and Phonetic Value

In the proposed system, melodies are divided into phrase (four bars), and the features on pitch and phonetic value of sounds in each phrase are defined. Here, consecutive sounds whose pitch are same are treated as a block, and features on pitch and phonetic value are defined per block.

2.1.1. Feature on Pitch

In the proposed system, feature on pitch of sounds is defined per block.

The feature on pitch at the block i in the phrase n of the melody m, $X_i^{P(m,n)}$ is given by

$$X_i^{P(m,n)} = \begin{cases} \dfrac{(P_i^{(m,n)} - P_1^{(m,n)})}{N_{diff}^{max}} + 0.5, & \\ & (1 \leq i \leq N^{(m,n)}) \\ -1, \qquad (N^{(m,n)} < i \leq N_b^{max}) & \end{cases} \tag{1}$$

where $P_i^{(m,n)}$ is the pitch of the sound at the block i in the phrase n of the melody m. In the proposed system, the sound between C_3 and C_7 are used, and 0~48 are assigned to them. N_{diff}^{max} is the maximum of the difference of two pitch and N_{diff}^{max} is set to 96 in the proposed system. And $N^{(m,n)}$ is the number of blocks in the phrase n of the melody m. N_b^{max} is the maximum number of sounds in each phrase, and N_b^{max} is set to 64 in the proposed system.

2.1.2. Feature on Phonetic Value

In the proposed melody retrieval system, (1) length of sounds per block and (2) length of each sound in the block are used as features on phonetic value.

(1) Length of Sounds per Block

The feature on length at the block i in the phrase n of the melody m, $X_i^{L1(m,n)}$ is given by

$$X_i^{L1(m,n)} = \begin{cases} \dfrac{L_i^{(m,n)}}{L^{max}}, & (1 \leq i \leq N^{(m,n)}) \\ -1, & (\ N^{(m,n)} < i \leq N_b^{max}) \end{cases} \tag{2}$$

where $L_i^{(m,n)}$ is the length of the block i in the phrase n of the melody m. In the proposed system, the minimum length (unit length) is set to the length of the demiquaver. And, L^{max}(=16) is the length of the whole note. In the proposed system, the rests belongs to the block including the previous note.

(2) Length of Each Sound in Block

The feature on length of the sound j at the block i in the phrase n of the melody m, $X^{L2(m,n)}_{i(j)}$ is given by

$$X^{L2(m,n)}_{i(j)} = \begin{cases} 0, & (1 \leq i \leq N_b^{(m,n)} \text{ and } j = 1) \\ \dfrac{\sum_{j'=1}^{j} L^{(m,n)}_{i(j')}}{L_i^{(m,n)}}, & \\ (\ 1 \leq i \leq N^{(m,n)} \text{ and } 1 \leq j \leq N^{(m,n,i)}) & \\ -1, & \\ ((1 \leq i \leq N^{(m,n)} \text{ and } N^{(m,n,i)} < j \leq & \\ \qquad N_s^{max}) \text{ or } N^{(m,n)} < i \leq N_b^{max}) & \end{cases} \quad (3)$$

where $L^{(m,n)}_{i(j')}$ is the length of the sound j' at the block i in the phrase n of the melody m, $L_i^{(m,n)}$ is the length of the block i in the phrase n of the melody m, $N^{(m,n)}$ is the number of the blocks in the phrase n of the melody m, $N^{(m,n,i)}$ is the number of the sounds at the block i in the phrase n of the melody m, and N_b^{max} is the maximum number of the blocks in the phrase. And N_s^{max} is the maximum number of the sounds in the block, and N_s^{max} is set to 10 in the proposed system.

2.2. Feature Vector

The feature vector of the melody m is generated by the self-organizing map which trained the features on pitch and phonetic value of sounds as similar as the conventional system[6].

In the proposed melody retrieval system, keywords are used as query in addition to features on pitch and phonetic value. As the keyword, we use the genres of music.

3. Melody Retrieval using Self-Organizing Map with Refractoriness

In this section, the proposed melody retrieval system using self-organizing map with refractoriness is explained.

3.1. Structure

The proposed system has two layers (1) the input layer and (2) the map layer. The input layer is composed of two parts which correspond to features (1) pitch and phonetic value of sounds and (2) genres of music (keywords). The map layer is composed of some modules and each neuron in the map layer corresponds to one of the stored melodies.

3.2. Learning Process

The learning process of the proposed melody retrieval system has two steps; (1) generation of feature vector and (2) learning in the self-organizing map with refractoriness[7]. In (1), the feature vector for each training melodies are generated (See section **2**). And in (2), the generated feature vectors are memorized in the self-organizing map with refractoriness.

3.3. Melody Retrieval Process

The melody retrieval process of the proposed music (melody) retrieval system has three phases;

(1) input of key melody using MIDI Keyboard,

(2) generation of feature vector

(3) melody retrieval by the self-organizing map with refractoriness.

(1) Input of Key Melody using MIDI Keyboard

A user inputs a part of melody. The genre of music also can be input if needed. In the proposed system, key melody is input by MIDI keyboard and the feature on pitch and phenolic value of sounds are extracted from the SMF (Standard MIDI File) as follows:

(a) The length of sounds are extracted from the SMF.

(b) The base length is determined as the average of the length of sounds whose length between l^{min} and $1.25l^{min}$. Here, l^{min} is the minimum length of the sound in the input key melody.

(c) Each sound is assigned to the appropriate note based on the length of each sound divided by the base length.

(2) Generation of Feature Vector

In the proposed melody retrieval system, not only features on pitch and phonetic value of sounds for the input key melody but also its half and twice one are employed as the query.

Step 1 : Generation of Feature Vector on Pitch and Phonetic Value $\boldsymbol{X}^P$, $\boldsymbol{X}^{L1}$ and $\boldsymbol{X}^{L2}$

The feature vectors on pitch and phonetic value of sounds are generated for the input key melody as similar as for trained melodies.

Step 2 : Calculation of Internal State of Neurons in Map Layer

The feature vector $\boldsymbol{X}^{PL(0)}$ composed of $\boldsymbol{X}^P$, $\boldsymbol{X}^{L1}$ and $\boldsymbol{X}^{L2}$ is generated.

$$\boldsymbol{X}^{PL(0)} = \left((\boldsymbol{X}^P)^T, (\boldsymbol{X}^{L1})^T, (\boldsymbol{X}^{L2})^T\right)^T \quad (4)$$

In the proposed melody retrieval system, the feature vectors $\boldsymbol{X}^{PL(1)}$ and $\boldsymbol{X}^{PL(2)}$ are also employed.

$$\boldsymbol{X}^{PL(1)} = \left((\boldsymbol{X}^P)^T, (0.5\boldsymbol{X}^{L1})^T, (\boldsymbol{X}^{L2})^T\right)^T \quad (5)$$

$$\boldsymbol{X}^{PL(2)} = \left((\boldsymbol{X}^P)^T, (2\boldsymbol{X}^{L1})^T, (\boldsymbol{X}^{L2})^T\right)^T \quad (6)$$

When the feature vector on pitch and length of sounds $\boldsymbol{X}^{PL(l)}$ (l=0, 1, 2) is given to the input layer, the internal state of the neuron i in the map layer $u_i^{MAP_F(l)}$ is calculated by

$$u_i^{MAP_F(l)} = D^P(\boldsymbol{W}_i, \boldsymbol{X}^P) \cdot D^{L1}(\boldsymbol{W}_i, \boldsymbol{X}^{L1(l)}) \times D^{L2}(\boldsymbol{W}_i, \boldsymbol{X}^{L2}) \quad (7)$$

where $D^P(\boldsymbol{W}_i, \boldsymbol{X}^P)$ is the similarity for the feature of pitch and it is given by

$$D^P(\boldsymbol{W}_i, \boldsymbol{X}^P) = \frac{1}{1 + \exp\left(\dfrac{d^P(\boldsymbol{W}_i, \boldsymbol{X}^P) - \theta_p}{T^P}\right)} \quad (8)$$

where θ_p is the threshold for pitch, T^P is the steepness parameter, and $d^P(\boldsymbol{W}_i, \boldsymbol{X}^P)$ is the Euclidean distance between the weight vector of the neuron i ($\boldsymbol{W}_i$) and the feature vector on pitch ($\boldsymbol{X}^P$).

And $D^{L1}(\boldsymbol{W}_i, \boldsymbol{X}^{L1(l)})$ is the similarity for the feature of length of sounds per block and it is given by

$$D^{L1}(\boldsymbol{W}_i, \boldsymbol{X}^{L1(l)}) = \frac{1}{1 + \exp\left(\dfrac{d^{L1}(\boldsymbol{W}_i, \boldsymbol{X}^{L1(l)}) - \theta_{L1}}{T^{L1}}\right)} \quad (9)$$

where θ_{L1} is the threshold for length of sound block, T^{L1} is the steepness parameter, and $d^{L1}(\boldsymbol{W}_i, \boldsymbol{X}^{L1(l)})$ is the Euclidean distance between the weight vector of the neuron i ($\boldsymbol{W}_i$) and the feature vector on length of sound block ($\boldsymbol{X}^{L1(l)}$).

$D^{L2}(\boldsymbol{W}_i, \boldsymbol{X}^{L2})$ is the similarity for the feature on length of each sound in block and it is given by

$$D^{L2}(\boldsymbol{W}_i, \boldsymbol{X}^{L2}) = \frac{1}{d^{L2}(\boldsymbol{W}_i, \boldsymbol{X}^{L2}) + 1} \quad (10)$$

where $d^{L2}(\boldsymbol{W}_i, \boldsymbol{X}^{L2})$ is the distance between the weight vector of the neuron i ($\boldsymbol{W}_i$) and the feature vector on the length of each sound ($\boldsymbol{X}^{L2}$) and it is given by

$$D^{L2}(\boldsymbol{W}_i, \boldsymbol{X}^{L2}) = \frac{1}{N^{key}} \sum_{k=1}^{N^{key}} d_k^{L2}(\boldsymbol{W}_i, \boldsymbol{X}^{L2}) \quad (11)$$

where N^{key} is the number of blocks in the input key melody. $d_k^{L2}(\boldsymbol{W}_i, \boldsymbol{X}^{L2})$ is the distance in the block k and between the weight vector of the neuron i ($\boldsymbol{W}_i$) and the feature vector on length of each sound ($\boldsymbol{X}^{L2}$) and it is given by

$$d_k^{L2}(\boldsymbol{W}_i, \boldsymbol{X}^{L2}) = \begin{cases} \sqrt{\displaystyle\sum_{\substack{j: X_j^{L2} \neq -1 \\ and\, j \in C_k^{L2}}} (\boldsymbol{W}_i^{L2}, \boldsymbol{X}^{L2})^2}, \\ (|j : W_{ij}^{L2} \neq -1 \text{ and } j \in C_k^{L2}| \\ \quad = |j : X_j^{L2} \neq -1 \text{ and } j \in C_k^{L2}|) \\ \sqrt{\displaystyle\sum_{\substack{j: X_j^{L2} \neq -1 \\ and\, j \in C_k^{L2}}} \phi_{kj}^1(\boldsymbol{W}_i, \boldsymbol{X}^{L2}) + \sum_{\substack{j: W_j^{L2} \neq -1 \\ and\, j \in C_k^{L2}}} \phi_{kj}^2(\boldsymbol{W}_i, \boldsymbol{X}^{L2})}, \\ \qquad\qquad \text{(otherwise)} \end{cases} \quad (12)$$

where C_k^{L2} is the set of neurons corresponding to the feature vector on length of each sound in the block. And, $\phi_{kj}^1(\boldsymbol{W}_i, \boldsymbol{X}^{L2})$ is given by

$$\phi_{kj}^1(\boldsymbol{W}_i, \boldsymbol{X}^{L2}) = \min_{\substack{j': W_{ij'}^{L2} \neq -1 \\ and\, j' \in C_k^{L2}}} (W_{ij'}^{L2} - X_j^{L2})^2 \quad (13)$$

and $\phi_{kj}^2(\boldsymbol{W}_i, \boldsymbol{X}^{L2})$ is given by

$$\phi_{kj}^2(\boldsymbol{W}_i, \boldsymbol{X}^{L2}) = \min_{\substack{j': X_{j'}^{L2} \neq -1 \\ and\, j \in C_k^{L2}}} (W_{ij'}^{L2} - X_j^{L2})^2. \quad (14)$$

Step 3 : Calculation of Outputs of Neurons in Map Layer

When the feature vector on pitch and length of sounds $\boldsymbol{X}^{PL(l)}$ (l=0, 1, 2) is given to the input layer, the output of the neuron i in the map layer $x_i^{MAP_F(l)}$ is given by

$$x_i^{MAP_F(l)} = \begin{cases} 1, & (i = c \text{ and } u_i^{MAP_F(l)} > \theta_{u1}\ ) \\ 0, & \text{(otherwise)} \end{cases} \quad (15)$$

where the neuron c is the winner neuron whose internal state is maximum, and θ_{u1} is the threshold for internal state of neurons.

Step 4 : Judgment whether One or More Neurons Fire

In **Step 3**, if one or more neurons fire, go to **Step 5**. Otherwise go to **Step 7**.

Step 5 : Update of Threshold θ_{u1}

If there is no neuron which fires in **Step 3**, the threshold for internal state of neurons θ_{u1} is updated by Eq.(16). If $\theta_{u1} = \theta_{min}$ is satisfied, the retrieval is finished. Here, θ_{min} is the minimum of the threshold for internal state of neurons θ_{u1}.

$$\theta_{u1} \leftarrow \theta_{u1} - \Delta\theta_{u1} \quad (16)$$

Then, the outputs of the neurons in the map layer for all feature vectors are calculated again using the updated threshold.

Step 6 : Judgment whether One or More Neurons Fire

In **Step 5**, if one or more neurons fire, go back to **Step 5**. Otherwise go to **Step 7**.

Step 7 : Generation of Feature Vector $\boldsymbol{X}^{PL(l)_k}$

The feature vectors are generated based on the outputs of the neurons in the map layer calculated in **Step 3** or **Step 5**.

When the feature vector on pitch and length of sounds $\boldsymbol{X}^{PL(l)}$ is given to the input layer, the output of the neuron in the map layer $\boldsymbol{x}^{MAP_F(l)}$ can be expressed as the summation of vectors $\boldsymbol{x}^{PL(l)_k}$ which has one element whose value is 1, if $|\boldsymbol{x}^{MAP_F(l)}| \geq 1$ is satisfied.

$$x_i^{MAP_F(l)} = \sum_{k=1}^{M(l)} x_i^{PL(l)_k} \quad (17)$$

$$(|\boldsymbol{x}^{PL(l)_k}| = 1, \quad x_i^{PL(l)_k} \in \{0, 1\})$$

where $M(l)(= |\boldsymbol{x}^{MAP_F(l)}|)$ is the number of elements which is 1 in the output vector $\boldsymbol{x}^{MAP_F(l)}$.

The kth feature vector on pitch and phonetic value of sounds $\boldsymbol{X}^{PL(l)_k}$ is calculated by

$$\text{(otherwise)} X_i^{PL(l)_k} = 2x_i^{PL((l)_k} - 1. \quad (18)$$

Step 8 : Generation of Feature Vector $\boldsymbol{X}^{PL(s(l)_k)}$

Steps 1 ~ 7 are repeated for s-shifted feature vectors, and $\boldsymbol{X}^{PL(s(l)_k)}$ are generated.

(3) Melody Retrieval using Self-Organizing Map with Refractoriness

The feature vector for the key melody $X^{PL(s(l)_k)}$ is given by

$$X_i^{(s(l)_k)} = \begin{cases} X_i^{PL(s(l)_k)}, & (1 \le i \le N^{MAP_F}) \\ X^K_{i-N^{MAP_F}}, & (N^{MAP_F} < i \le N^{IN}) \end{cases} \tag{19}$$

where N^{MAP_F} is the number of neurons in the map layer of the self-organizing map which generates feature vectors on pitch and phonetic value (length) of sounds, and N^{IN} is the number of neurons in the input layer of the self-organizing map with refractoriness for melody retrieval. And X_i^K is the feature on genre of music (keywords) which is given by a user.

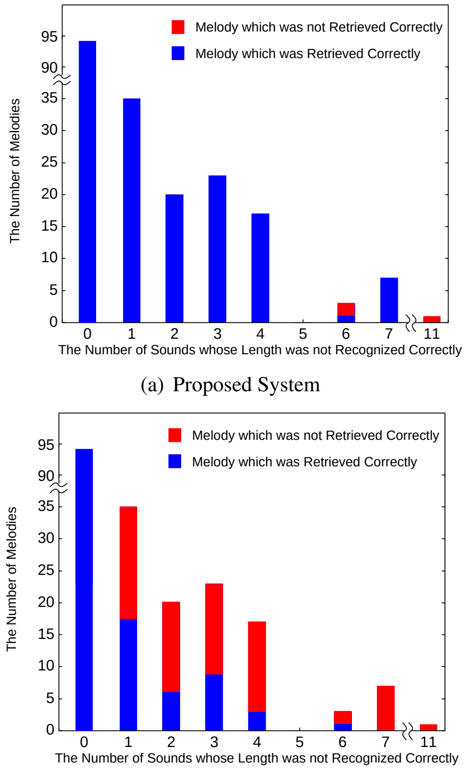
The retrieval process of the proposed melody retrieval system is similar as the conventional melody retrieval system[6]. Since the proposed system is based on the self-organizing map with refractoriness[7] as same as our previous work[6], plural neurons in the map layer whose connection weights are similar to the input feature vector of the input key melody can fire sequentially. As a result, desired plural melodies can be retrieved.

4. Computer Experiment Results

In this experiments, the proposed system which stored 100 melodies were used. Figure 1 shows the retrieval accuracy in the proposed system and the conventional system[6]. In this experiment, 200 keys composed of four bars were given by MIDI keyboard. These are a part of stored melodies. As shown in Fig.1, the features on pitch and phonetic value of sounds can be extracted correctly from 94 key melodies. The features on pitch and phonetic value of sounds extracted from 106 key melodies includes 1~11 notes whose lengths were not correct. The proposed melody retrieval system could retrieve the desired melodies correctly from 197 key melodies. In contrast, the conventional system[6] could retrieve the desired melodies correctly from 128 key melodies. From this result, we can confirm that the proposed system can retrieve correct melodies even when the key melody includes fluctuation.

5. Conclusions

In this paper, we have proposed the melody retrieval system by self-organizing map with refractoriness which can retrieve the melodies from the key melody which is given by MIDI keyboard. In the proposed melody retrieval system, the stored melodies are divided into phrases composed of four bars, and the feature melody of pitch and phonetic value are employed. Since the proposed system is based on the self-organizing map with refractoriness, the plural neurons in the map layer corresponding to the input key melody can fire sequentially because of refractoriness of neurons. And plural melodies to be retrieved can be fired. From computer experiment results, we confirmed that the proposed system can correct melodies from the key melody which is given by MIDI keyboard in 98.5% cases.



(a) Proposed System

(b) Conventional System[6]

Figure 1: Retrieval Accuracy.

References

[1] M. Kataoka, M. Kinouchi and M. Hagiwara : "Music information retrieval system using complex-valued recurrent neural networks," Proceedings of IEEE International Conference on System, Man and Cybernetics, pp.4290-4295, 1998.

[2] T. Sonoda, M. Goto and Y. Muraoka : "A WWW-based Melody Retrieval System," Proceedings of International Computer Music Conference, pp.349–352, 1998.

[3] T. Sonoda and Y. Muraoka : "A WWW-based Melody Retrieval System – An Indexing Method for A Large Database," Proceedings of International Computer Music Conference, pp.349–352, 2000.

[4] T. Sonoda, T. Ikenaga, K. Shimizu and Y. Muraoka : "A Melody Retrieval System on Parallelized Computers," Proceedings of International Workshop on Entertainment Computing, 2002.

[5] T. Hirata, T. Tokuda and Y. Osana : "Melody retrieval by self-organizing map with refractoriness," Proceedings of IEEE and INNS International Joint Conference on Neural Networks, Hong Kong, 2008.

[6] A. Cho and Y. Osana : "Melody retrieval by self-organizing map with refractoriness which has robustness for fluctuation of key input," Proceedings of IEEE and INNS International Joint Conference on Neural Networks, San Jose, 2011.

[7] H. Mogami, M. Otake, N. Kouno and Y. Osana : "Self-organizing map with refractoriness and its application to image retrieval," Proceedings of IEEE International Joint Conference on Neural Networks, Vancouver, 2006.

[8] T. Kohonen : Self-Organizing Maps, Springer, 1994.