2010 2nd International Conference on Information and Multimedia Technology (ICIMT 2010)
IPCSIT vol. 42 (2012) © (2012) IACSIT Press, Singapore
DOI: 10.7763/IPCSIT.2012.V42.6

A Content-based Music Retrieval System Using Genetic Algorithm

Xingkai Han[1] and Baiyu Cao[2]

[1]Department of Computer Science and Technology, Harbin Institute of Technology
[2]Shenzhen Graduate School, Shenzhen, China
xingkai.han@gmail.com, baiyu.cao@gmail.com

Abstract—In this paper, we proposed a significant matching method for content-based music retrieval using GA (Genetic Algorithm). The goal of this method is to search for a model which can boost the matching rate between input query and music database. Based on the selected MIDI files which are generated automatically as our music database, we used the Genetic Algorithm approach to evaluate the matching effect for melody segments. In our research, pitch tracking and dynamic threshold note segmentation were utilized to build a template on melody for queries. In addition, a melody contour alignment was presented aiming to correct the humming differences among individuals. Also, we carried out the computation of similarity in our experiment with dynamic time wrapping. As experimental results illustrates, the method better the retrieval accuracy to around 90 percent, which can satisfy the common need.

Keywords- Music Retrieve, Melody Representation, Contour Alignment, Template Matching.

1. Introduction

Recently, with the rapid development of Internet, the collection of audiovisual data has increased dramatically. Also, to access multimedia content through Internet is getting increasingly interest. Therefore, to afford the users an easy access of the huge number of music data, a well manageable multimedia database structure is of high importance. The traditional way of marking them literally has, gradually, not been the best interest of music retrieval. As a result, content-based music information retrieval has become a heat focus in the field of signal processing, pattern recognition, even information retrieval.

The content-based music information retrieval problem can be divided into two sub problems: in the first place, how to transform the query aiming to enhance the retrieval precision. Secondly is how to match the query with melodies in database in order to strengthen the retrieval efficiency. It is well acknowledged that the former is always related with converting a query into temporally note segments or into frame-fixed pitch track, whereas the latter concentrates on the similarity measurement. See [1] for an overview of music information retrieval systems.

In recent years, Asif Ghias presented this concept firstly [2]. Then, Lie Lu [3] added the difference of neighboring pitch and the length of note to the construction of melody. Researchers in [4] describe beat-IDs in which they utilize average beat period. Nowadays, most approaches involving in MIR (music information retrieval) pay attentions to the representation of query by note or by pitch sequence, and to the match method including string matching [5], hidden Markov models [6], or dynamic programming [7].

Though they have got some thrilling achievements, enormous development has not been made as the instability of music signal and the complexity of music analysis. Some key points are listed as follows: the extraction of music features, the representation of melody, and the matching method between query and database. Therefore, in our research, we extracted the music features satisfactorily through fundamental tone extraction and dynamic threshold-setting. Furthermore, we made great progress and tried to tackle the

matching problems by GA and dynamic time wrapping. Accordingly, we find those methods proposed are high efficiency.

2. Musci Retrieval System

As we know, music is a discrete note sequence which changes as the time does. However, we always feel the music is a whole. As the Gestalt theory says, people's feeling has proximity, continuity, similarity. Accordingly, this theory illustrates the perception of a certain type note. Consequently, we run our experiment with the melody, which is the main perceptive characteristic of music. Significantly, the melody contour is the description of the pitch changing, while the pitch is determined by fundamental frequent. Hence, we can describe the melody contour through extracting fundamental frequent and selecting proper model.

In our proposed system, we define the standard template and humming input template, which are made up of our model. To be more specific, the standard template composes of both the numbered musical notation and fundamental frequent. In other words, we adopt the relation of both of them. While on the other hand, the input template consists of the pitch sequence extracted from humming segments by the user. Moreover, both the template has the similar shape after normalization. Our research is performed based on this characteristic.

Figure 1 shows the flow diagram of a content-based music retrieval system based on GA.

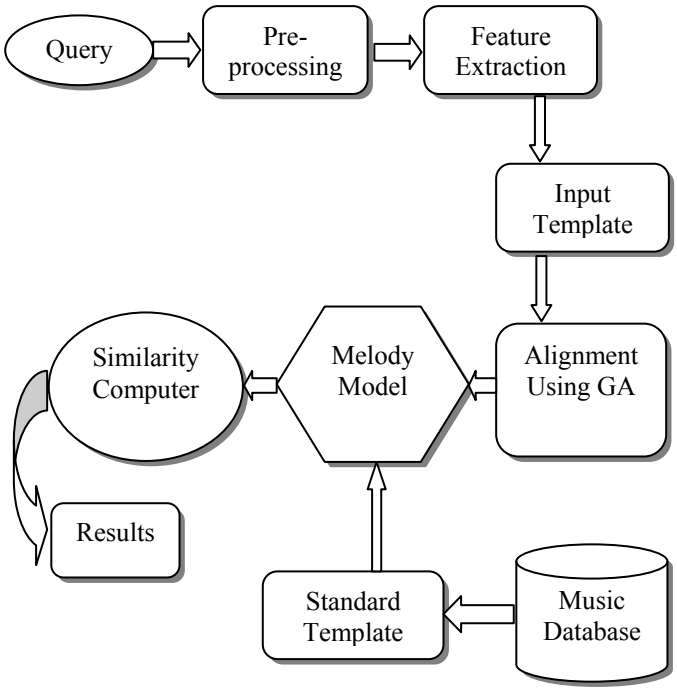

Figure 1. The framework of the MIR system

3. The Algorithm of the Melody Contour Alignment Based on GA

It is universal accepted that the range of frequency produced from human is from around 50Hz to about 3200Hz, while that made by music generally varies from approximate 16Hz to just below 7000Hz, that is to say it changes from C2 to A5 in terms of note. Obviously, the ranges are different [5]. As the different sound of each user has, the different fundamental frequent will evidently be. If we normalize both the input template and standard template simply, the retrieval results to a great extend will inaccurate as such, we have neglected some details.

In order to cope with this problem, we propose a method constructing a template, which replaces the input template to match. This template can be obtained by gaining on the input template in the frequency range of standard template. So we choose the Genetic Algorithm to search this template owing to it has the advantage of searching global optimal solution without a priori knowledge.

To begin with, we define input template

$P = \{p_1, p_2, p_3 .... p_i .. p_n\}$, where p_i denotes a note while n represents the number of notes. Then, we search the imminent template $Q = \{q_1, q_2, q_3 ... q_i, ... q_n\}$ in the frequency range of standard template. We use cosine value to compute the similarity between both two templates so that we can find the optimal template. The equation is as follows:

$$Sim(P,Q) = \cos(P,Q) = \sum_{i=1}^{n} \frac{p_i}{\sqrt{\sum_{k=1}^{n} p_k^2}} \times \frac{q_i}{\sqrt{\sum_{k=1}^{n} q_k^2}} \tag{1}$$

3.1 Encoding with Chromosomal

We use a chromosomal to represent an imminent template, the length of which can change along with the length of template. In our paper, we adopt decimal base to denote the chromosomal, and we list the table1 which shows the relationship between note and frequency.

Table1. The relationship between the note and frequency..

	Do	Re	Mi	Fa	Sol	La	Si
Low pitch	130	146	164	174	196	220	247
Middle Pitch	261	293	330	349	392	440	494
High Pitch	523	587	659	698	784	880	988

We use the note fundamental frequency as our standard pitch template, and there are 21 notes varying from "Do" to "Si". In each chromosomal, we select a frequency reflecting to the note from Table1 so that if there are n notes the length of chromosomal is n . As well as this, we mark the encoding of chromosomal to be $G = \{g_1, g_2, g_3 ... g_i ... g_n\}$.

In the first place, generate the chromosomal groups randomly, which can include any frequency from table1. Then, replacing Q with one chromosomal and computing the similarity between P and Q. Repeat this step. Finally, rank all the chromosomal by the similarity. After that, we operate them with selection, crossing, and mutation in order to reserve the best chromosomal, which is the imminent template we want

3.2 Fitness Degree Function

As the definition of GA, the optimal chromosomal should have the maximum fitness degree. Therefore, we adopt the cosine function as the fitness degree function.

$$F(s) = Sim(P,Q) = \cos(P,Q) \tag{2}$$

3.3 Generate the Next Generation

1) Generate L chromosomal randomly.

2) Selection. We select the chromosomal with highest fitness to the next generation directly. And the others are chosen to the next generation by sample randomly with the probability formula

$$P_s(S_i) = F(S_i) \Big/ \sum_{j=1}^{L} F(S_j) \tag{3}$$

$F(S_j)$ is the fitness degree function, L is the number of chromosomal. After selection, we remain the better one. While in contrast, we remove the worse one. The total number keeps the same L

3) Crossing. The crossing operation performs between two chromosomal. Randomly, choosing the reflecting part of them and exchanging each other, respectively. In proposed research, we adopt the double points crossing. In other words, we select two points in each chromosomal and exchange the part between the two points of each.

In this step, new chromosomal can be generated.

4) Mutation. Mutation means choosing a chromosomal K randomly, in which a position is also selected randomly. Then, replace the value in this position with a new random number. Significantly, we should compare the fitness degree of new chromosomal K' with that of K in order to copy the higher one to the next generation.

Figure 2 shows the steps of GA

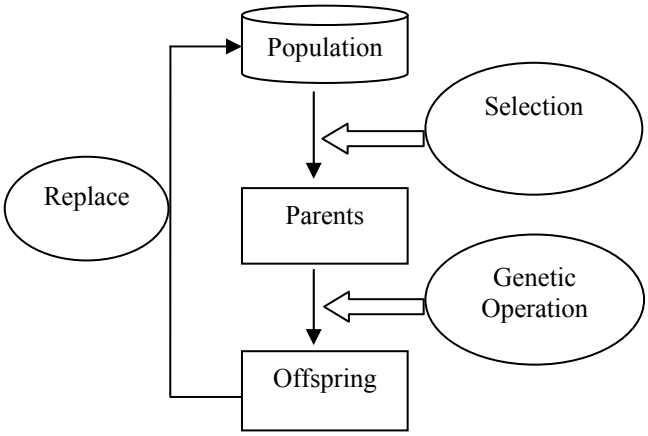

Figure2. Flow chart of Genetic Algorithm

4. Template Matching Algorithm

There are some matching algorithms nowadays including Euclidean distance, DTW, Cosine etc.

Euclidean distance has been recognized to be a practical method, but it demands the two wav shapes are nearly similar. For instance, if only one peak does not locate at the right site, they are not similar. Whilst by contrast, DTW supports the stretching of melody contour in the time domain. It can align easily without matching the point one by one. DTW allows the appearance of some errors, because it compares the wav shapes not the points.

We measure the distance of DTW using the dynamic programming based on the accumulative distance matrix (ADM). The accumulative distance matrix between input template P and standard template S is

$$r(i,j) = d(p_i, s_j) + \min\{r(i-1,j), r(i,j-1), r(i-1,j-1)\} \tag{4}$$

Obviously, DTW can not search the accurate solution when the some templates' contours look like each other.

In our proposed research, we add weight to the measurement method. Firstly, the user should hum a whole sentence at least which is the input template. Meanwhile, dividing a song into some whole sentences stored at the database which viewed as the standard templates. Finally, the matching formula is

$$\left|L_s - L_n\right| \le 3 \tag{5}$$

where L_s is the number of note in standard template while L_n denotes the number of note in input template. Then we just take the standard templates which satisfy this formula into account.

The advantage is that it can lead to high similarity when neglecting the difference among notes in terms of some errors of note segment.

Combine the two algorithms together, and we achieve the final method of similarity measurement

$$S = w_1 S_v + w_s S_D \tag{6}$$

where S_v represents the similarity computed by Euclidean distance while S_D means that of DTW. It is noteworthy that w_l and w_s stand for the weight of S_v and S_D, respectively. And

$$w_l + w_s = 1 \tag{7}$$

According the way of similarity measurement mentioned above and the rule of $N-best$, we reserve 3 of the best results as our final results

5. Experiments and Analysis

In our experiments, there are 4350 songs in our database. Those songs are music files monaural MIDI. The number of standard pitch templates is 7590. Every song reflects several different templates. And the input music segment's sample rate is 44100 Hz, and the quantization bit rate is 16 bits.

At the beginning, we filter the input music with frequency range from 60 Hz to 3500 Hz. The filter is first order digital filter $H(z)=1-\mu z^{-1}$, $\mu=0.98$. Then, strengthen the input signal, and flame the input segment with Hamming Window, the length of which is 20ms while the overlap part is 10ms.

In our experiment, one of the input numbered musical notations is "2475272346". We depict this input template in Figure3.

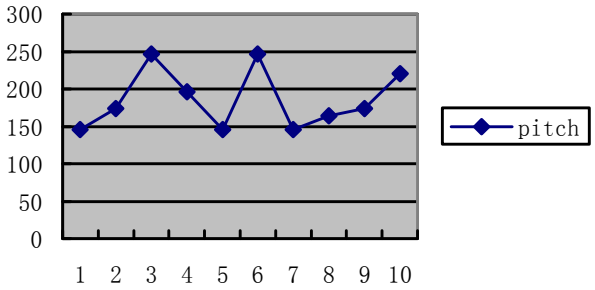

Figure3. The input template before alignment

The input of Genetic Algorithm is the fundamental frequency as well as the number of notes of the input template. Here, we define the length of chromosomal is 10, the size of population is 40, and the number of generation is 100. The flow chart is as follows:

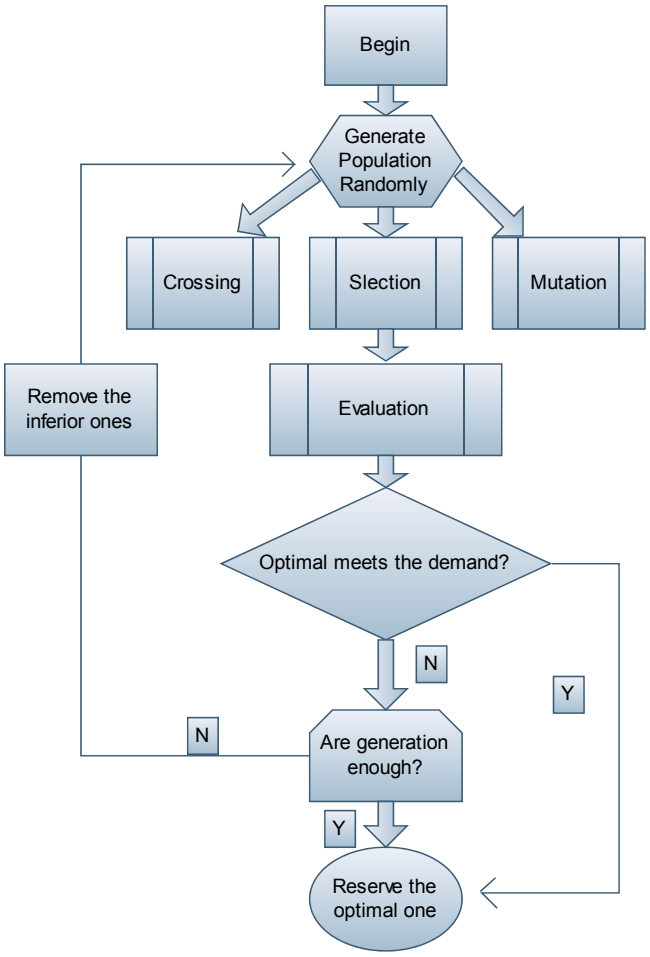

Figure4. The flow chart of our experiment

In our experiment, we use Sum-Squared Error to measure the similarity error between input template and standard template. The result shows in figure5

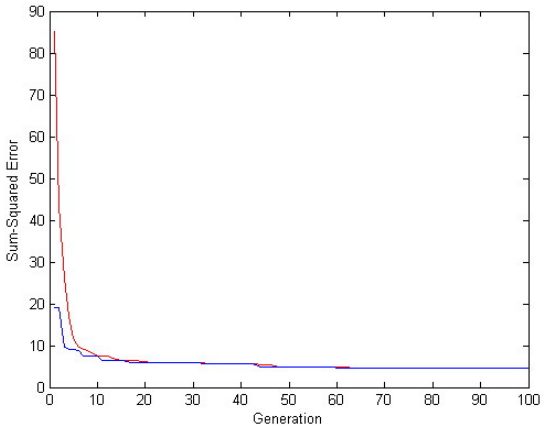

Figure5. The relationship between the Sum-Squared Error and Generation

From this char we can clearly see that, the error falls all the time. More specifically, the error decline dramatically to just below 10 at the 10th generation. Then, it fluctuates during 10 to 20. Notably, the similarity degree gets the trend of convergence after the 20th generation. Over the next 80 generations, the results remain stable.

The figure6 illustrates that the imminent template has higher similarity to standard template than the input template.

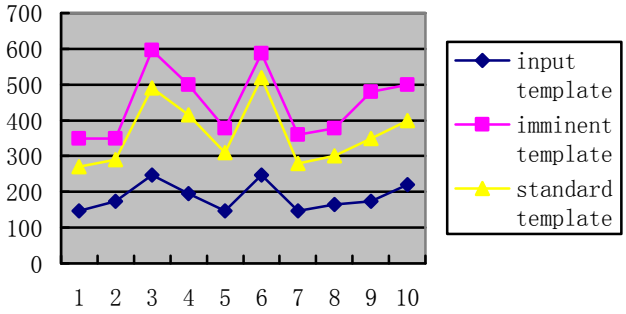

Figure6. Compare the 3 templates

The figure7 shows the situation after normalization to those 3 templates.

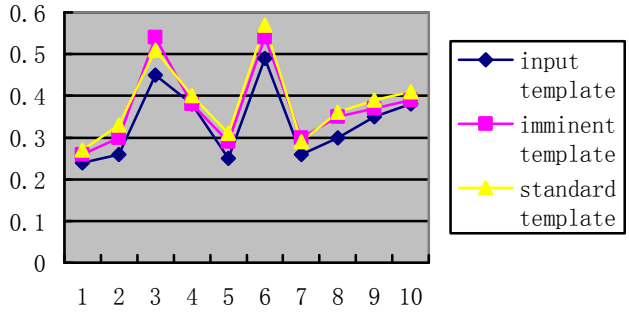

Figure7. Compare the 3 templates after normalization

As can be seen from Figure6 the imminent template nearly coincides with the standard template. It illustrates the input template can be revised to be more standard after Genetic Algorithm. More precisely, the difference of some pitch can be neglected.

We choose 20 melody segments recorded by 3 men and 3 women as the input music and remain the top X results. The retrieval results are listed in table3. We can see the recognition rate reach up to 90% when $X = 3$, which is pretty well. Even when $X = 1$, and it can arrive at 80%.

Table3. The Retrieval Results

Top X	The Number of music	Retrieval Time (s)	Recognition Rate (%)
1	16	1.12	80
2-3	18	1.20	90
4-10	19	1.10	95
10-20	19	1.12	95

6. Conclusion and Future work

To sum up, we have conducted great efforts on the template generating and matching in our research. Also, we search the optimal imminent template by Genetic Algorithm, which perform pretty well in dealing with the humming errors. Importantly, we mix the DTW and Euclidean distance together, and fulfill the automatic matching of template. Actually, the proposed method can improve the robustness and adaptability of the system.

However, we just go about the experiment with small corpus. In future, when confronting with the large music database, hopefully we can build the index automatically just in case the standard templates explosion. As well as that, we should take effective measures to research the interface for the user.

7. Acknowledgment

This investigation was supported in part by the National Nature Science Foundation of China and the National 863 Program of China. Also, the authors would like to Qingcai Chen for their valuable suggestions in preparing this paper.

8. References

[1] R. Typke, “Music Retrieval based on Melodic Similarity”, Ph.D. thesis, University of Utrecht, 2007.

[2] ASIF G, JONATHAN L, DAVID C, et al.: “Query by humming-musical information retrieval in an audio database”, proceedings of the Third ACM International Conference on Multimedia. San Francisco, USA, 1995.

[3] CAI R, LU, L and ZHANG H J: “Using structure patterns of temporal and spectral feature in audio similarity measure”, proceedings of the Eleventh ACM International Conference on Multimedia. Berkeley, 2003.

[4] Kirovski, D., Attias, H.: Beat-ID: “Identifying Music via Beat Analysis”, Proceedings of Multimedia Signal Processing. (2002) 237-240.

[5] K.Lemstrom, “String Matching Techniques for Music Retrieval”, Ph.D. thesis. University of Helsinki, 2000.

[6] C.Meek,W.Birmingham, “Applications of binary classification and adaptive boosting to the query-by humming problem”, in Proc.3rd International Conference on Music Information Retrieval 2005..

[7] J.-S. R. Jang, C.-L. Hsu, and M.-Y. Gao, “A query-by singing system based on dynamic programming”, in Proc. International Workshop on Intelligent Systems Resolutions, 2000.

[8] PARDO B.: “Music information retrieval”. Communication of ACM 2006, 49 (8): 29-31.