Interactive musical editing system to support human errors and offer personal preferences for an automatic piano

Method of searching for similar phrases with DP matching and inferring performance expression

Yuta Hikisaka

Graduate School of Computer Science and System Engineering
Kyushu Institute of Technology
680-4, Iizuka City, Fukuoka, Japan, 820-8502
hiki@mmcs.mse.kyutech.ac.jp

Eiji Hayashi

Department of Mechanical Information Science and Technology
Kyushu Institute of Technology
680-4, Iizuka City, Fukuoka, Japan, 820-8502
haya@mse.kyutech.ac.jp

Abstract

We have developed a system that allows a piano to perform automatically. In order to play music in the manner of a live pianist, we must add expression to the piano's performance. In the case of piano music, there are often 1000 or more notes in the score of even a short piece of music, and the editor must spend a huge amount of time to accurately simulate the emotionally expressive performance of a highly skilled pianist. Therefore, we have developed an interactive musical editing system to edit music more efficiently utilizing a database.

We have analyzed MIDI data regarding the performances of highly skilled pianists in order to observe the stylistic tendencies of their performances. As a result, it was found that phrases having similar patterns in the same composition were performed in similar styles.

Therefore, we developed a system that searches for similar phrases throughout the musical score and infers the style of their performance. Here, we propose a method using DP (Dynamic Programming) matching as a way to search for similar phrases and a method for inferring performance expression.

Key words: *automatic piano, knowledge database, computer music, DP matching*

1. Introduction

We have developed a performance system for an automatic piano. In this system, 90 actuators are installed in the 88 keys and 2 pedals of a grand piano. Those actuators operate key strokes and pedaling to be executed on the piano. (See Figure 1)

Reproducing music with the piano is similar in some ways to reproducing music on the computer. Essentially, variations in tempo, dynamics, and so on are needed to arrange the respective tones in the desired way. However, in the case of piano music, there are 1000 or more notes in a score of even a short piece of music, and the editor must spend an enormous amount of time working with the arrangement in order to simulate the expressions of an actual performance.

Therefore, in this research, we have developed an interactive musical editing system to edit music more efficiently [1].

We have analyzed MIDI data regarding the performances of highly skilled pianists in order to observe the stylistic tendencies of their performances. As a result, it was found that phrases having similar patterns in the same composition were performed in similar styles. Moreover, it was found that the pattern of notes in the score sometimes influences the music's expression.

In this research we developed a system that searches for similar phrases throughout a musical score and infers the style of the performance. We propose a method using DP matching as a way to search for similar phrases. This system converts notes into character strings. As well, it runs DP matching using character strings and calculates the degree of disagreement between these strings. We use these calculations as an index to determine whether the strings resemble each other. Moreover, we designed a method for inferring the performance expression of similar phrases found by DP matching.

In this paper, we describe the results of searching for similar phrases using DP matching and inferring performance expression.

Figure 1: View of the automatic piano

2. Musical Editing Support System

2.1 System Architecture

The structure of the system is shown in Figure 2. The user edits performance information via the user's interface on the computer display (We discuss this performance information in the next paragraph). Also, the user can access a database of musical grammar, the user's preferences, and so on. As a result, editorial work is reduced and efficient editing becomes possible.

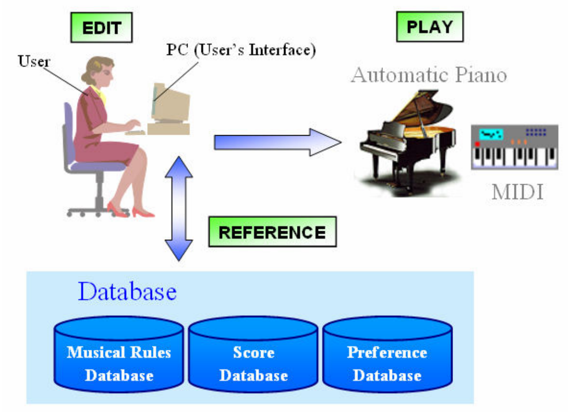


Figure 2: Structure of the editing system

2.2 Format of Performance Information

The automatic piano that we have developed uses a music data structure that is similar to MIDI. We defined performance information, dividing it into two categories, the notes and the pedals.

The note information consists of six parameters involved in producing a tone: "Key (note)", "Velo (velocity)", "Gate", "Step", "Bar", and "Time". "Velo" is dynamics, given by the value of 1～127. "Gate" is the duration of the note in milliseconds. "Step" is the interval of time until the next note, and it also exhibits tempo. "Bar" is the vertical line placed on the staff to divide the music into measures.

The pedal information consists of four parameters: "Key (indicating the kind of pedal, "Damper" or "Shifting")", "Velo (the pedaling quantity)", "Time (the duration of applying the pedal)", and "Bar".

2.3 Automatic Translation with Database

Our system can automatically apply a rough performance expression using a Musical Rules Database and Score Database. (See Figure 3)

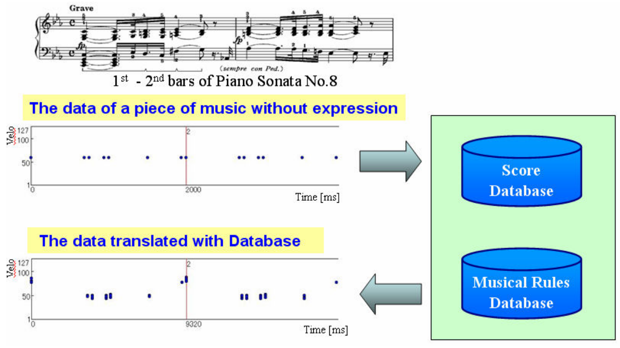


Figure 3: Automatic translation with database

2.3.1 Musical Rules Database

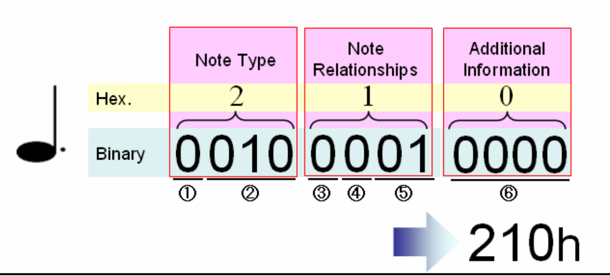
This database has the architecture of musical grammar necessary to interpret symbols in musical notation. This database consists of five tables containing "Dynamics marks", "Articulation marks", "Symbol of Changing Dynamics or Changing Tempo (Symbol that affects the speed of a note or the increase or decrease of the volume)", "Time signature", and "Tempo marks".

This database enables a general performance expression to be applied to music automatically.

2.3.2 Score Database

This database has symbols including notes, time signature, rests and so on in standard musical notation. Symbols were pulled together in the order of bars, and bar symbols are arranged in a time series.

This database consists of three tables, the "Element table" (showing the position of the notes and the composition of the chords), the "Symbol table" (showing the position of the music symbol) and the "Similarity Table" (showing the position of the repetition of the phrase).



①	Note or Rest (0: Note, 1: Rest)
②	Note Value (000: whole note, 001: half note ...etc)
③	Tie (If the note has a tie then this number is 1.)
④	Ornament (If the note has an ornament then this number is 1.)
⑤	The number of dots (Exceptionally, if it is "11" then the note is tuplet.)
⑥	Additional Information The number of tuplets. (Triplets: 0011) The type of an ornament. (Trill: 0101) etc...

Figure 4: Note Value

The element table contains the field “Note Value”. The data in this field indicates the type of note, e.g., a quarter note, a triplet, and so on. “Note Value” is expressed by three hexadecimal numbers. The “Note Value” numbers are shown in Figure 4.

3. Searching for Similar Phrases

As a result of the analysis, it was found that phrases of the same pattern existing in the same tune are performed in a similar expression[2].

This time, we used DP matching to search for similar phrases.

3.1 DP matching

DP matching[3] is a technique used well in the field of speech recognition, bioinformatics and so on. It has a feature that can calculate similarity between two words that are different in a number of characters from each other.

In Figure 5, the route of minimum cost in each point is taken and the route with the lowest cost is assumed to be an optimal path finally. The cost at that time is defined as the distance between patterns. In this system, this distance is handled as a threshold to judge whether the phrases are similar to each other.

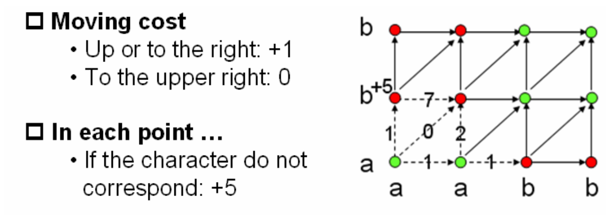


Figure 5: Method of calculating the distance

3.2 Searching with DP matching

In this passage, we describe a method of searching with DP matching. We had to convert a musical score into character strings (Note Pattern) before searching for similar phrases. This process is explained below.

3.2.1 Note Pattern

Our system converts a score into a Note Pattern using Note Value (See Passage 2.3) to perform DP matching. Of the three columns of Note Value, we used the two columns from the left. The system replaces the second column with letters of the alphabet (from G) because a letter should not be the same as the first column with the second column since it expresses one note with two columns. An example of a Note Pattern conversion is shown in Figure 6.

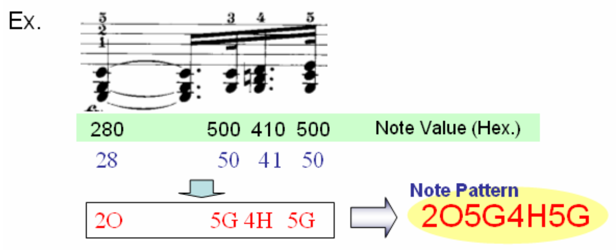


Figure 6: Example of Note Pattern

3.2.2 The Method of Searching

The flow of the similar phrase search is shown in Figure 7.

Essentially two rounds of search processing are performed. The first processing round narrows down the point that may be the point of resemblance in all search ranges. A pattern of the same number as the search phrase is pulled out and the distance between two patterns is calculated using DP matching. If they are in complete accord (distance = 0) then the phrase is stored in the Similar Table. If the distance is lower than the threshold then the phrase is stored in the Temporary Table. In the second round of processing, DP matching is performed again using the phrases in Temporary Table while increasing the number of characters. In other words, the system looks for the most similar phrases in the surrounding phrase. The threshold of this system has been decided by trial and error.

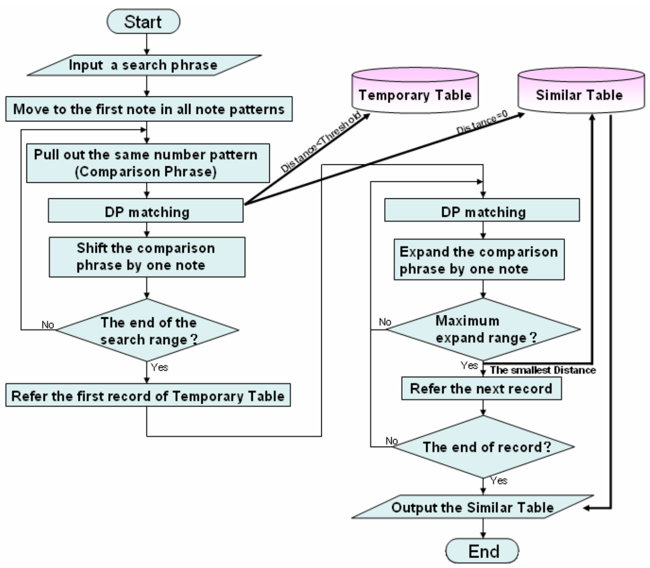


Figure 7: The flow of the similar phrase search
(Search range and expand range are set beforehand.)

3.2.3 Searching Result

An actual search was performed using the search method explained in paragraph 3.2. Part of the result of the search using this system is shown in Figure 8. The horizontal axis of each graph is time (in milliseconds), and the vertical axis is velocity (1 to 127). This graph was made based on performance data by Gerhard Oppitz.

The arrangement of each note also looks like the searched phrase. Especially, it is understood that the performance expression was also very similar in the graph of (a) whose distance is 0. The graph of (b) does not completely correspond. But based on the change in Velo, it seems similar.

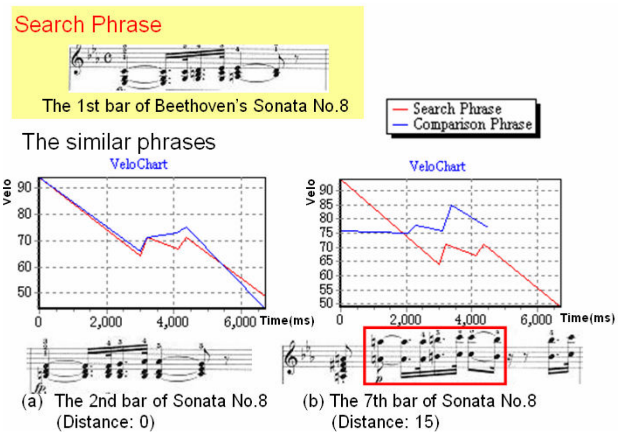


Figure 8: The result of the searching

4. Inferring Phrase Expression System

The similar phrases searched for by the method of the description in Chapter 3 are inferred using the expression of the performance of the search phrase.

4.1 Method of inferring phrase expression

The correspondence of the note patterns is revealed when the distance is minimized in DP matching. Next, the ratio of Velo to the previous sound is calculated in each sound of the searching phrase. Velo of the associated phrase is inferred using a ratio. Thus, the Velo of a similar phrase is inferred. The value of Time uses the value decided by Automatic translation (See Passage2.3).

4.2 Inference Result

The result of the inference using this inference system is shown in Figure 9. The phrase of the seventh bar of Beethoven's Sonata No. 8 was inferred from the first bar, which Gerhard Oppitz performed.

Thus, the inferred phrase was similar to the expression of the performance of the search phrase.

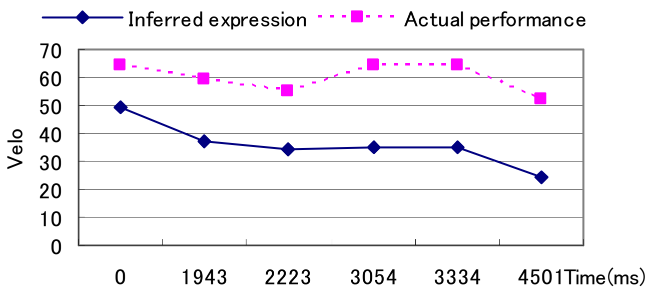


Figure 9: Inferred performance expression

5. Conclusion

We designed a method of searching for similar phrases and inferring the performance expression using DP matching and constructed these as one system.

In the similar phrase search, the system was able to search for similar phrases using DP matching in a short time, and it was even possible to search for phrases whose resemblance might not be immediately apparent.

In the inference of the performance expression, a performance expression was able to be inferred using the best associating by the DP matching from the search.

The performance expression of some similar phrases that exist in the tune can be inferred at the same time using this system. It is regarded that the work efficiency increases by this automatic editing in contrast to editing the piece by hand from the beginning, without sacrificing quality.

In this study, a general performance expression was used for musical symbols such as Crescendo. But, it is thought that it will be possible to infer the user's intentions for such symbols as well in the future.

Reference

[1] Takamatsu Y, Hayashi E, et al, "Study on the Performance Information Edit Support System for an Automatic Piano (*in Japanese*)", Technical Report of Information Processing Society of Japan, MUS-52, pp.111-118, 2003.

[2] Hikisaka Y, Hayashi E, et al, "Interactive musical editing system for supporting human error and offering personal preferences for an automatic piano -A system of inferring phrase expression-", Proc. of the 11th International Symposium on Artificial Life and Robotics, pp 296-299, 2006.

[3] Nakagawa S, "Speech recognition based on probabilistic model (in Japanese)", IEICE, pp18-28, 1988.