Acoust. Sci. & Tech. **22**, 1 (2001)

PAPER

Improved DP matching between a musical score and its performance using interpolation

Takayuki Hoshishiba and Susumu Horiguchi

School of Information Science, Japan Advanced Institute of Science and Technology (JAIST), 1–1, Asahidai, Tatsunokuchi, Nomi-gun, Ishikawa, 923–1292 Japan
E-mail: hoshisi@jaist.ac.jp, hori@jaist.ac.jp

(*Received 18 February 2000, Accepted for publication 12 September 2000*)

Abstract: In order to realize musical performances with human-like emotions by computer, it is necessary to analyze performances using musical score data. The Dynamic Programming (DP) method is often used for matching between a musical score and its performance. However, the DP does not achieve a high matching rate since the ordering of attack of the notes in a chord is not always the same and there are difficult passages of right and left hand performances. Furthermore the score does not define the number of iterations of trills. This paper addresses the process of matching two or more MIDI performance data involving DP along with linear interpolation to maximize the collation between the score and the performance. The improved DP method is also extended to match performances including trills.

Keywords: Computer performance, Piano performance, Matching, Dynamic programming, Linear interpolation

PACS number: 43.75.St

1. INTRODUCTION

With the current development of electronics, the sound quality of electronic musical instruments has been improved. Particularly, the sound of electric piano has been getting realistic. Many people enjoy composing their own music and making the system produce the sound employing virtual instruments on MIDI systems provided with appropriate software. Excellent performances, however, have been recorded generally as sound but not as MIDI performance data, so to extract precise performance information from recorded sound is desirable in the future.

With data from a musical score alone, however, computer only perform mechanically and can not be expected to perform with human-like emotions. In order to make computers perform musically, features required for yielding artistic performance are to be extracted from scores and fed to computers [1, 2]. First of all it is necessary to find precise note-to-sound correspondence between a musical score and its actual performance.

To realize musical performance on computers, many researchers have proposed their own methods. For imitating famous performances, Takami *et al*. [3] proposed a system that generates performance data from acoustically analyzed frequency maps of both performed sound and score data. Dynamic Programming (DP) matching has been employed to generate these maps. Many computer accompaniment systems [4–7] have been proposed for performing along with a human performer. To follow the human performer, DP methods are applied to find note-to-sound correspondence between its musical score and the human performance. Block *et al*. [5] introduced the penalty for three types of errors of the performance; wrong notes, missing notes and extra notes to improve the efficiency of following the human performer. Furthermore, Dannenberg *et al*. [6] dealt with trills and glissandi by converting these into single events. Conventional DP methods are not enough to find precise correspondence between score and performance.

In order to realize musical performances with human-like emotions by computer, it is necessary to analyze past human performances using musical score data. The DP method, however, does not achieve a high matching rate for actual piano performances. The reason may be that in actual piano performance, sounds in a chord don't appear simultaneously, and there are difficult passages to be played with both hands as described in the score. Furthermore, the score does not define the number of iterations of trills.

In this paper, we propose an improved DP matching method between musical score and performance in MIDI data using linear interpolation. Furthermore, the improved DP method is also extended to match the performance including trills so that one note on score data is matched to plural notes on performance data. The matching results of the improved DP method are discussed for twenty piano performances by different pianists. It is shown that proposed matching method is effective for piano performance where right and left hand parts are not played in the order described in the score. The proposed matching method can be applied also for performance with trills where the the number of iterations is not definitely specified.

2. MATCHING PERFORMANCE DATA BY THE DP METHOD

To analyze past human performance data, note-to-sound correspondence must be found precisely between the score data and past human performance data. Since execution time required to find note-to-sound correspondence is huge, it is necessary to automate this process. In this section, the data format of both the score and the performance data is described, and the DP method and the clustering of chords are also explained. The matching performance is also discussed for twenty piano performances of five compositions played by different professional pianists.

2.1. Data Notation of Score and Performance

To simplify the handling of both the score and the performance data, the Standard MIDI File Format is used for the both. In converting from the score to the Standard MIDI File Format, the dynamics (velocity, in MIDI terminology) are held constant and all notes are considered as tenuto, ignoring staccato signs.

Figure 1 shows the first measure of Chopin Op. 10-12 (*Revolutionary*). This score are converted to the Standard MIDI File Format, which is expressed in MIDI text format, as shown in Fig. 2. In the Figure, one line corresponds to one note. The “onset”, “note”, “vel” and “dur” indicate onset time, tone height, velocity and duration of the note respectively. The onset time consists of three numbers; measure ID, beat ID, and the time lag from the beat. In the Fig. 2, the time unit is one 240th of nominal duration assigned to a quarter note. This value can be changed. Notes Ab4 and Eb4 in Fig. 1 are expressed as G#4 and D#4 in Fig. 2 because they are equivalent, respectively, in MIDI representation. In MIDI format, tone height is coded using an integer from 1 through 127 as 64 for A4 incrementing by unity for going up a semitone even for notes modified by b or #. Right hand part and left hand part on the performance data are mixed in the MIDI stream. In this paper, both parts on the score are mixed into one stream, then the correspondence between the score data and the performance data is found.

Figure 3 shows performance data of the score shown in Fig. 1 by a professional pianist. The onset time of the MIDI performance data is defined by time units, since a quarter note is set to 192 time units and the tempo is set to 118 (beats/minute), the unit for describing onset time and duration is 2.65 (milli-second).

Fig. 1 The score of *Revolutionary*.

onset	note	vel	dur
(Right part)			
1|1|000	B5	64	2|000
1|1|000	G5	64	2|000
1|1|000	F5	64	2|000
1|1|000	D5	64	2|000
1|1|000	B4	64	2|000
(Left part)			
1|1|120	G#4	64	0|060
1|1|180	G4	64	0|060
1|2|000	F4	64	0|060
1|2|060	D4	64	0|060
1|2|120	D#4	64	0|060
1|2|180	D4	64	0|060
1|3|000	B3	64	0|060
1|3|060	G3	64	0|060
1|3|120	G#3	64	0|060
1|3|180	G3	64	0|060
1|4|000	F3	64	0|060
1|4|060	D3	64	0|060
1|4|120	D#3	64	0|060
1|4|180	D3	64	0|060

Fig. 2 MIDI text format of the score.

onset	note	vel	dur
1650	B5	90	386
1652	D5	80	390
1654	G5	73	390
1656	B4	87	388
1658	F5	78	390
1776	G#4	80	42
1824	G4	78	32
1860	F4	76	32
1896	D4	74	42
1932	D#4	75	20
1964	D4	63	32
1998	B3	81	32
2028	G3	74	44
2064	G#3	75	24
2096	G3	64	32
2132	F3	81	32
2162	D3	78	44
2196	D#3	77	28
2230	D3	70	36

Fig. 3 Data format of the performance.

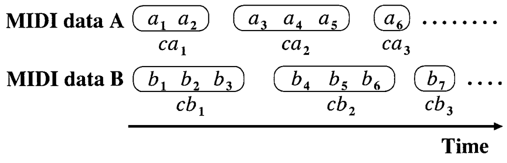


Fig. 4 Result of clustering.

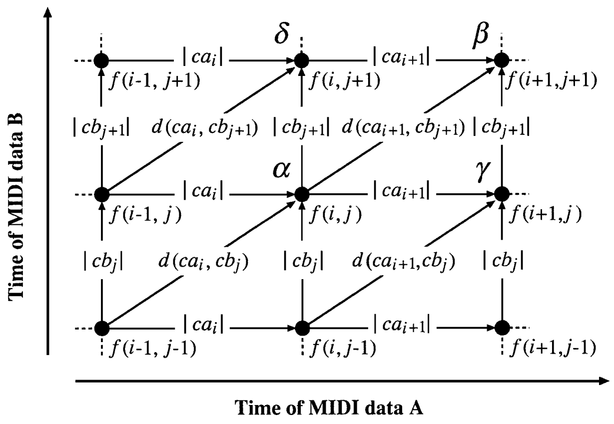


Fig. 5 Dynamic programming method process.

2.2. Clustering of Chords

While searching for correspondence between the score data and the performance data, it becomes necessary to find the match between chords.

In score, notes in a chord appear simultaneously, but in performance this is not usually the case. Furthermore, the ordering of attack of the notes in a chord is not always the same. Thus, the notes that are played in close proximity in time are grouped together to form a cluster (chord). Time window is used as a time-width to form a cluster. The time window for the score is set at 0, since there are no lags. For the performance data, the time window is adjusted so that the number of clusters becomes approximately the same as the number of chords in the score. Specifically, the clustering process is repeated by increasing the time window from 0 until the number of clusters becomes approximately the same as the number of clusters in the score.

In Fig. 4, a_i and b_j indicate the i-th and j-th note of MIDI data streams A and B, respectively and ca_m and cb_n indicate the m-th and n-th cluster of data stream A and B, respectively.

2.3. Dynamic Programming Method

The DP method [8] is used to find the maximum correspondence between the elements of the clusters that are generated from performance data and score data respectively. The cluster distance $d(ca_m, cb_n)$ is defined as the total number of elements (notes) that are missing in the other. And assuming that the clusters can be matched one-to-one, DP cost function $f(m,n)$ is defined as follows:

Table 1 The list of compositions for experiments.

Composition	Notes	Performances
Fantaisie-Impromptu C sharp minor op. 66	3035	3
Revolutionary Etude No. 12 C minor op. 10–12	2080	5
Raindrop Prelude No. 15 D flat major op. 28–15	1518	4
Minute Waltz No. 6 D flat major op. 64–1	1384	3
Heroic Polonaise No. 6 A flat major op. 53	5914	5

$$\begin{cases} f(0,0) &= 0 \\ f(m,0) &= f(m-1,0) + |ca_m| \\ f(0,n) &= f(0,n-1) + |cb_n| \\ f(m,n) &= \min \begin{pmatrix} f(m-1,n) + |ca_m|, \\ f(m,n-1) + |cb_n|, \\ f(m-1,n-1) + d(ca_m, cb_n) \end{pmatrix}, \end{cases} \tag{1}$$

where m and n denotes the cluster ID of ca and cb, respectively and $|ca_m|$ and $|cb_n|$ are the number of notes in the clusters ca_m and cb_n, respectively.

In Fig. 5, at point α, the correspondence between i-th cluster from data A and j-th cluster of data B has been accomplished. From α to β in Fig. 5, $(i+1)$-th cluster of data A and $(j+1)$-th cluster of data B will be matched. From α to γ in Fig. 5, $(i+1)$-th cluster of data A will be skipped and the cost of $|ca_{i+1}|$ will be increased. From α to δ in Fig. 5, $(j+1)$-th cluster of data B will be skipped and the cost of $|cb_{j+1}|$ will be increased. The optimal path with minimum cost is searched backwards.

2.4. The Matching Result by The Clustering and DP Method

Experiments were conducted on twenty performances of five Chopin compositions for piano in MIDI format. Table 1 shows the list of compositions, the number of notes for each composition and the amount of performance data for experiments. In Table 1, the compositions without original title are written in English. All score data were prepared using Henle's original edition (edited by Ewald Zimmermann) and the performance data were published by Yamaha Music Media.

Table 2 shows the result of the match between the twenty performance data and the five score data. In the table, the integer in the performance column corresponds to the player. As this table shows, it is clear that the matching success rate for Op. 28-15 (*Raindrop*) is about 98% and for Op. 10-12 (*Revolutionary*) is about 96%. However, the matching success rate for Op. 66 (*Fantaisie-Impromptu*) is very low at about 80%. The reason for errors may be that the Op. 66 used in the experiment contains sections that are extremely fast and moreover, as shown in Fig. 6, there are difficult passages to be played where the right hand plays sixteenth notes against the left hand playing eighth-note triplets that divide the half note into six parts, causing the performers to play not exactly as notated.

Figure 7 shows the matching results between score notes shown in Fig. 6 and performance data by Player 1. The correct ordering should be C#3, G#4, G#3, A4, C#4, and G#4, but the Player 1, for example, plays C#3, G#4, A4, G#3, G#4, and C#4. The matching results of the third and forth notes, and the fifth and sixth notes are crossing. Since the DP method can't match two notes with different order, the matching rate remains 80%. To improve the matching rate, an improved DP matching algorithm that can match data in different order is proposed.

Table 2 Matching success rates for twenty performances by DP method.

Performance		Total notes	Matched notes
Op. 66	1	3036	2608 (85.90%)
	2	3017	2425 (80.38%)
	3	3033	2425 (79.95%)
Op. 10-12	1	2091	2016 (96.41%)
	2	2080	2002 (96.25%)
	3	2073	1988 (96.38%)
	4	2087	2045 (97.99%)
	5	2086	2039 (97.75%)
Op. 28-15	1	1518	1498 (98.68%)
	2	1513	1494 (98.74%)
	3	1517	1496 (98.62%)
	4	1523	1496 (98.23%)
Op. 64-1	1	1401	1307 (93.29%)
	2	1441	1309 (90.84%)
	3	1432	1349 (94.20%)
Op. 53	1	5943	5517 (92.83%)
	2	6116	5614 (91.79%)
	3	6009	5573 (92.74%)
	4	5925	5385 (90.89%)
	5	6065	5644 (93.06%)

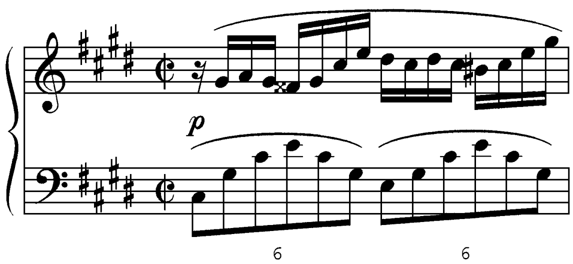


Fig. 6 A passage from the score of *Fantaisie-Impromptu*.

3. IMPROVED MATCHING ALGORITHM USING INTERPOLATION

Since the ordering of attack of the notes in a chord is not always the same even in performance by professional pianists, the clustering of notes is not always successful. Thus, the matching success rate only using the DP method is not high. In this section, an improved matching algorithm is proposed using a table for linear interpolation. The matching result of the improved algorithm will also be evaluated on the same performances.

3.1. Improved Matching Algorithm Using Interpolation

Reordering the notes at a certain interval using linear interpolation improves the performance of the DP method. The following algorithm based on an interpolation table is proposed. Figure 8 shows example of improved matching process between the score data and the performance data shown in Fig. 7. The table depicted in Fig. 8 represents temporal correspondence between notes on the score and those in performances in each step of the procedure.

1. The score data and the performance data are divided into clusters by the method written in section 2.2.
2. The clusters are matched by DP method leaving unmatched clusters out of correspondence.
3. The elements (notes) in a pair of corresponding clusters (chords) are matched leaving unmatched notes out of correspondence.
4. The linear interpolation procedure finds correspondence between notes on the score and those in performance evaluating the results of the improved DP matching. The thick arrows in Fig. 8(4) show corresponding data in every three clusters.
5. Linear interpolation using the table estimates the onset time of score notes which were not matched with the performed notes by the DP method. Mesh notes in Fig. 8(5) will be applied to this linear interpolation.
6. The performed notes are matched to the estimatedly close values. In case multiple candidates are available for the estimated value, the closest note in performance data is selected.

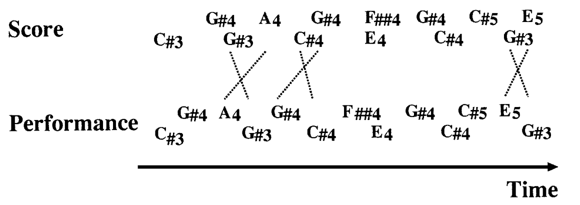


Fig. 7 The performance data by Player 1.

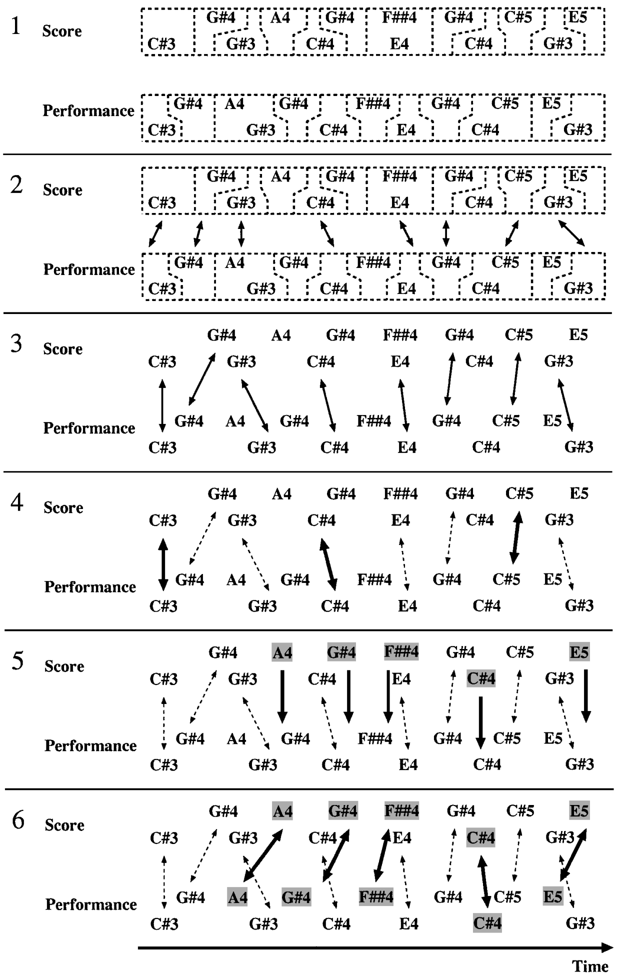


Fig. 8 Example of improved matching process.

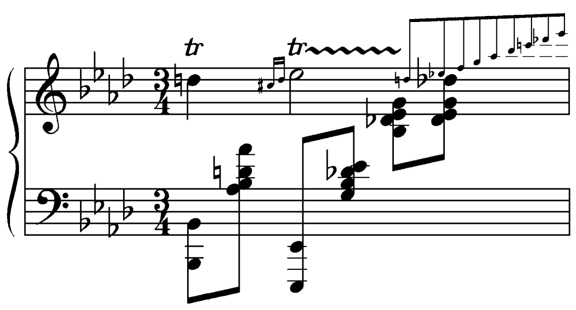


Fig. 9 The score of *Heroic*.

3.2. The Matching Result by The Improved DP Algorithm

Table 3 summarizes matching success rates the same for twenty performances by the improved DP method using interpolation. Comparing Table 3 with Table 2, all of the matching success rates are much improved. It is clear that the performance data for three compositions, Op. 66, Op. 10-12 and Op. 28-15 are almost matched to the score data since the matching success rates are more than 99% except for Player 3 of Op. 66. For Player 3, it is clear from his comments in the liner notes that he used two score versions as references and interpreted them accordingly. The parts of performance which is different from the score is extracted as a performance mistake.

On the other hand, although the matching success rates of Op. 64-1 (*Minute*) and Op. 53 (*Heroic*) are improved from 90% to 95%, the performance data are not matched to the scores completely. The reason for errors is that these two compositions have trills and the score does not define the number of iterations. To dissolve this problem, an extended algorithm for trill matching is proposed.

4. IMPROVED MATCHING ALGORITHM FOR TRILLS

In the score data, a trill is expressed by one note with a trill sign. However, the trill of the performance data is composed of plural notes. Thus, the matching algorithm for the trill has to take account of one-to-many matching.

Table 3 Matching success rates for twenty performances by the improved DP method.

Performance		Total notes	Matched notes
Op. 66	1	3036	3030 (99.80%)
	2	3017	2979 (98.74%)
	3	3033	2957 (97.49%)
Op. 10-12	1	2091	2055 (98.28%)
	2	2080	2072 (99.62%)
	3	2073	2066 (99.66%)
	4	2087	2079 (99.62%)
	5	2086	2079 (99.66%)
Op. 28-15	1	1518	1512 (99.60%)
	2	1513	1510 (99.80%)
	3	1517	1513 (99.74%)
	4	1523	1516 (99.54%)
Op. 64-1	1	1401	1348 (96.22%)
	2	1441	1374 (95.35%)
	3	1432	1379 (96.30%)
Op. 53	1	5943	5742 (96.62%)
	2	6116	5840 (95.49%)
	3	6009	5777 (96.14%)
	4	5925	5642 (95.22%)
	5	6065	5885 (97.03%)

```
  onset note  vel    dur
---------------------------
(Right part)
64|1|000 D5    64    0|220 [tr]
64|1|220 C#5   64    0|010
64|1|230 D5    64    0|010
64|2|000 D#5   64    1|150 [tr]
64|3|150 D5    64    0|010
64|3|160 D#5   64    0|010
64|3|170 F5    64    0|010
64|3|180 G5    64    0|010
64|3|190 G#5   64    0|010
64|3|200 A#5   64    0|010
64|3|210 C6    64    0|010
64|3|220 C#6   64    0|010
64|3|230 D#6   64    0|010

(Left part)
64|1|000 A#1   64    0|120
64|1|000 A#2   64    0|120
64|1|120 G#3   64    0|120
64|1|120 A#3   64    0|120
64|1|120 D4    64    0|120
64|1|120 G#4   64    0|120
64|2|000 D#1   64    0|120
64|2|000 D#2   64    0|120
64|2|120 G3    64    0|120
64|2|120 A#3   64    0|120
64|2|120 C#4   64    0|120
64|2|120 D#4   64    0|120
64|3|000 A#3   64    0|120
64|3|000 C#4   64    0|120
64|3|000 D#4   64    0|120
64|3|000 G4    64    0|120
64|3|120 C#4   64    0|120
64|3|120 D#4   64    0|120
64|3|120 G4    64    0|120
64|3|120 C#5   64    0|120
```

Fig. 10 Data format of the score with trills.

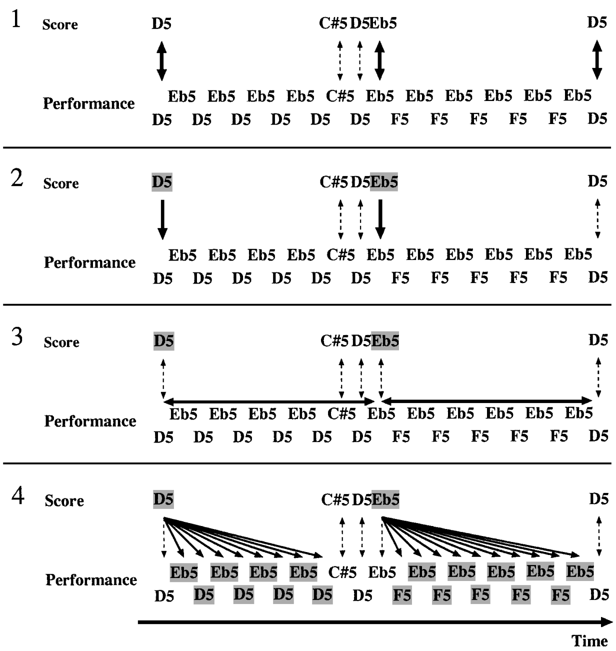


Fig. 11 Example of matching process for trills between the score and the performance data.

Figure 9 shows a piece of the score of Chopin Op. 53 (*Heroic*) which has the trills. For handling the trill, trill information is added to the MIDI text format of the score as shown in Fig. 10.

Table 4 Matching success rates by the extended DP method for trill matching.

Performance		Total notes	Matched notes
Op. 64-1	1	1401	1400 (99.93%)
	2	1441	1424 (98.82%)
	3	1432	1426 (99.58%)
Op. 53	1	5943	5886 (99.04%)
	2	6116	6030 (98.59%)
	3	6009	5933 (98.74%)
	4	5925	5840 (98.57%)
	5	6065	6030 (99.42%)

In this section, an extended matching algorithm for trills will be described. The matching result of the extended algorithm will also be evaluated for the performance data including trills.

4.1. Extended Matching Algorithm for Trills

To realize one-to-many matching between the score and the performance data, the following algorithm is proposed. Figure 11 shows example of matching process for trills between score notes and actual performance data shown in Fig. 9.

1. The linear interpolation for trills finds correspondence between notes on the score and those in performance employing the improved DP matching described in section 3.1.
2. The beginning point of a trill is estimated by linear interpolation based on the foregoing note and the succeeding note neighboring the current note having a trill sign. Meshed notes in Fig. 11 are notes to which the linear interpolation described above is applied.
3. The ending point of a trill is estimated based on the duration of the note having a trill sign on the score in the same way as step 2.
4. All the performed notes which satisfy the condition of the estimated interval are matched to the score note with the trill sign. The number of trill iterations is ignored.

In Fig. 11, the notes (“D5” and “Eb5”) which have a trill sign on the score are correctly matched to plural notes of performance data.

4.2. The Matching Result by Extended DP for Trill Matching

Table 4 shows matching success rates for Op. 64-1 and Op. 53 by the extended algorithm for trill matching. The method improves the matching rates from 95% in Table 3 to 99% in Table 4 for eight performance data with trills.

5. CONCLUSIONS AND FUTURE PROBLEMS

In this paper, a method to improve the matching success rates between a piano score and the actual performance data was presented. An improved method was suggested where linear interpolation was used to modify the data to achieve a high matching success rate in the DP matching of performance data and score data. A method to extend the algorithm to one-to-many matching for trills was also presented. The experiments with twenty performance data of five compositions for piano played by professional pianists showed that the extended algorithm achieved excellent matching success rates between the score and performance data with trills. In the future, it is important to investigate how to extract parameters that define musical individuality by comparing multiple performances of the same composition played by different pianists.

ACKNOWLEDGMENTS

We would like to thank Professor I. Tokuhiro (Kanagawa Institute of Technology) and Professor I. Fujinaga (Peabody Conservatory) for their discussion. Part of this research was supported by Grant-in-Aid for Scientific Research, The Ministry of Education, Science and Culture of Japan.

REFERENCES

[1] T. Hoshishiba, S. Horiguchi, and I. Fujinaga, "Study of expression and individuality in music performance using normative data derived from MIDI recordings of piano music," Proc. Int. Conf. Music Perception and Cognition, 465–470 (1996).

[2] T. Hoshishiba and S. Horiguchi, "Normative performance rules extracted by multiple performance date," Japan-China Joint Meeting on Musical Acoustics, 155–160 (1997).

[3] K. Takami, H. Katayose, and S. Inokuchi, "Extraction of music informations from piano performance," Trans. IEICE **J72-D-II**, 917–926 (1989) (in Japanese).

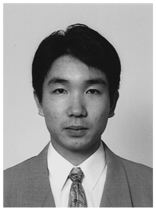
[4] R. B. Dannenberg, "An on-line algorithm for real-time accompaniment," Proc. Int. Computer Music Conf., 193–198 (1984).

[5] J. J. Block and R. B. Dannenberg, "Real-time computer accompaniment of keyboard performances," Proc. Int. Computer Music Conf., 279–289 (1985).

[6] R. B. Dannenberg and H. Mukaino, "New techniques for enhanced quality of computer accompaniment," Proc. Int. Computer Music Conf., 243–249 (1988).

[7] Y. Horiuchi and H. Tanaka, "A computer accompaniment system with independence," Proc. Int. Computer Music Conf., 418–420 (1993).

[8] M. Nagao, *Pattern Information Processing* (Corona Publishing Co., Tokyo, 1988), pp. 134–135 (in Japanese).



Takayuki Hoshishiba was born in Kanagawa, Japan, on April 18, 1971. He received the B.E. degree in Information and Computer Science from Kanagawa Institute of Technology in 1994, and the Ms. Degree in Information Science from Japan Advanced Institute of Science and Technology (JAIST) in 1996. He is presently a PhD. student of JAIST. He is a member of Acoustic Society of Japan (ASJ), Information Processing Society of Japan (IPSJ), and International Computer Music Association (ICMA).

Susumu Horiguchi was born in Shiga, Japan on July 10, 1952. He received a B.E. and Ms. E. and Dr. E. degrees from Tohoku University in 1976, 1978 and 1981, respectively. He joined to Department of Communication Eng., Tohoku University as a research associate in 1981. From 1985 to 1986, he joined to IBM Watson Research Center, NY, U.S.A. as a visiting research scientist. In 1989, he became an Associate Professor, Department of Information Eng., Tohoku University. Since 1992, he has been with Graduate School of Information Science, Japan Advanced Institute of Science and Technology (JAIST) as a chair Professor of Multimedia Integrated System. His research interests include multimedia systems, massively parallel computing and Virtual reality system. During July–August, 1994 and 1997, he was a visiting professor, Center of Advanced Computer Study and Department of Commuter Science, Texas A&M University. Dr. Horiguchi is a member of the Institute of Electronics, Information and Communication Engineers (IEICE) of Japan, the Information Processing Society of Japan (IPSJ), and a senior member, Computer Society, the Institute of Electrical and Electronic Engineering (IEEE).