Data processing in music performance research: Using structural information to improve score-performance matching

[Heijink, H., Windsor, L, & Desain, P. (2000). Data processing in music performance research: Using structural information to improve score-performance matching. *Behavior Research Methods, Instruments, & Computers*, **32**(4), 546–554]

ABSTRACT — In order to study aspects of music performance, one has to find correspondences between the performance data and a score. Locating the corresponding score note for every performance note, called matching, is therefore a common task. An algorithm that automates this procedure is called a matcher. Automated matching is difficult because performers make errors, performers use expressive timing, and scores are frequently underspecified. To find the best match, most matchers use information about pitch, temporal order, and the number of matched notes. We show that adding information about the musical structure of the score gives better results. However, we found that even this information was insufficient to identify some types of performance errors and that a definition of best match based only on the number of matched notes is sometimes problematic. We provide some suggestions about how to achieve greater improvements.

1 Introduction

Music performance involves highly complex cognitive and motor skills and therefore provides a rich domain for the study of perception and motor control. Music performance in the Western classical tradition involves composers who create scores and musicians who transform those scores into performances. A score specifies which notes are to be played in what order, and how long those notes should be in relation to each other. It also gives indications about tempo, loudness, timbre, articulation, and phrasing. These indications are often vague and open to different interpretations. Moreover, certain score events, like trills, glissandi, and other ornaments, have different meanings depending on the instrument, the character, and the style of the piece. A score offers a wealth of interpretative freedom, and the way in which a performer deals with this freedom may contribute to our understanding of the nature of music, perception, and action. The role of the performer manifests itself in the relation between score and performance, and this relation has therefore been the subject of a considerable body of research (for an overview, see Gabrielsson, 1999, and Palmer, 1997.)

In order to analyze aspects of a performance such as timing fluctuations, the onset times of performance notes have to be related to the notated score rhythms. Likewise, when studying performance errors, the performance has to be compared to the score on a note-by-note basis. Linking performance notes to score notes is commonly called *matching* a performance to a score. The resultant relation is called a *match*, and a computer program that performs this task is called a *matcher*.

In the simplest case of music performance, the score consists of a single voice; that is, it consists of a sequence of notes $x_1,...,x_n$. Every note has an onset time in some discrete unit (a time in seconds would not be appropriate, because a score does not exactly specify tempo or timing), a pitch, and a duration in terms of the chosen time unit. A performance is a sequence $y_1,...,y_m$, where every note has an onset time in seconds, a pitch, and a duration in seconds. Pitches are assumed to be discrete and identifiable. Note that the numbers n and m that represent score and performance length, respectively, are not necessarily equal: Performers make errors.

We will only consider recorded performances that consist of discrete events. This means that the pitch, the onset, and the duration of a note is unambiguous and readily available for every note. A performance recorded as an audio signal does not fall into this category, but a MIDI recording of a performance does. The Musical Instrument Digital Interface or MIDI protocol (International MIDI Association [IMA], 1983; see also Roads, 1996) stipulates a hardware interconnection scheme and a method for data communication. MIDI information is packaged into small messages sent from one device to another. The messages usually carry information about keypresses and knob turnings. We are concerned with messages that consist of start and stop times of notes, which in the case of piano music means the pressing and releasing of keys. Because much of the music performance research focuses on MIDI recordings of piano music, the restriction we make here does not limit too severely the applicability of the matchers we discuss too severely.

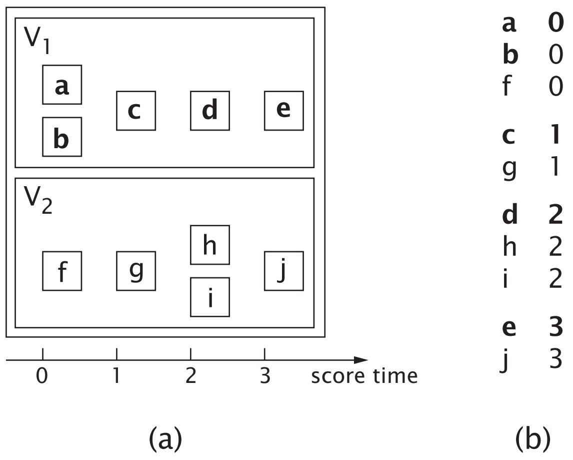

Figure 1: (a) An example of a score structure with two parallel voices. (b) A textual representation of the same structure. V_1 and V_2 stand for voice 1 and 2, respectively. The note names are for identification, they do not refer to pitches. The empty lines in (b) have been inserted to improve readability.

In the case of a single-voiced score, matching a performance to a score is similar to comparing two text strings (see Cormen, Leiserson, & Rivest, 1990, chap. 34) or comparing two DNA strings (see Cormen et al., section 16.3). However, a score rarely just consists of a sequence of notes: More frequently, it consists of a sequence of chords. These chords, commonly called score *events*, consist of one of more notes that should be played at the same time. Not only notes can occur at the same time, but also whole sequences: A score frequently consists of several *voices* that all are sequences of events. In summary, a score consists of one or more voices $V_1,...,V_k$, where each voice consists of one or more events $E_1,...,E_n$. Each event in turn consists of one or more notes $x_1,...,x_q$. In fact, scores can be structured in much more complex ways than this, but we will not discuss those here. Figure 1a illustrates the structure of a score.

In this article, a textual representation of a score will be used, where every line represents one note. On each line is the abstract onset time of the note and its pitch. We will only deal with two-part music (piano music can usually be divided into a left-hand and a right-hand part) and to make clear which notes belong to which part, we will use boldface for the right-hand part. In the remainder of this article, we will refer to parts as *voices*. The onset times make clear which notes belong to which event: notes with the same onset time belong to the same event. Figure 1b shows the textual score representation of Figure 1a.

Matching is a complex task. It is difficult because performers make errors (score notes may be missing in the performance, or performance notes may be missing in the score), performers make use of expressive timing, and scores are frequently underspecified. Furthermore, music usually consists of several parallel streams, so existing string matching algorithms cannot be used.

Underspecification of a score occurs when ornaments are notated ambiguously and it is not clear which notes to expect in the performance. This is not a problem that arises in the pieces we use here. In what follows, we will concentrate on the problems of performance errors and expressive timing.

In music performance, there are many types of errors that occur (Palmer & Van de Sande, 1993). Some errors are hard to detect: For instance, detecting errors in timing or articulation usually requires much general musical and stylistic knowledge. We will focus on errors that are easy to detect automatically, namely pitch errors that occur, for instance, when a wrong key is hit by the performer.

Essentially, there are two kinds of pitch errors: notes are played that are not in the score (*insertion errors*), as shown in Figure 2a, or notes that are specified in the score are omitted (*deletion errors*), as shown in Figure 2b. Some combinations of insertion and deletion errors can be interpreted as *substitution errors*, as in Figure 2c. These errors are very common in music performance, but the interpretation of errors is outside the scope of this article. We will therefore focus on locating them.

An error can be detected even though the note was correct (henceforth called *false alarm*), and conversely, an error may not be detected even though the note was incorrect (henceforth called *miss*). We will refer to a situation in which an error is rightly detected or a note is rightly deemed correct, as a *hit*. A matcher cannot tell the difference between a hit, a miss, and a false alarm. Only by comparing the match found by the matcher with the correct match (for instance, a match made by hand) can the difference be discerned.

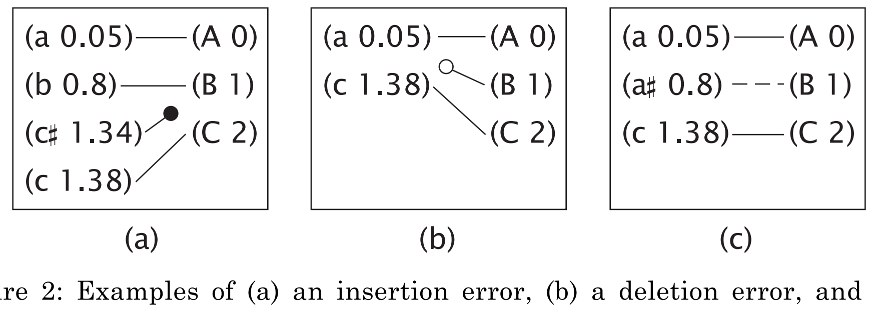

Figure 2: Examples of (a) an insertion error, (b) a deletion error, and (c) a substitution error. In each case, the performance is on the left, the score on the right. Matches and substitutions are indicated by solid and dashed lines, respectively. Insertion and deletion errors are indicated by lines ending in solid and open circles, respectively.

The process of finding errors will now be described. A matcher uses *order constraints* when making a match. This means that when a score specifies a certain temporal order for the notes to be played, this order must be reflected in the performance in a certain way. All the matchers we discuss in this article use the following two constraints.

First, notes in the same score event—that is, notes that should sound simultaneously according to the score—can occur in any order in the performance. If, for instance, a score specifies a chord c, e, g, the performance might be c, e, g, in that order, but the performance could also be e, c, g, or g, e, c, due to motor noise, expressive intentions, or recording artifacts. Second, notes in different events should occur in the order specified in the score. It follows that notes within the same melody cannot be reversed.

The errors in the performance are detected in the following way: A match is made between score and performance, such that the order constraints are satisfied and as many notes as possible are matched. Two notes can be matched if their pitches are equal. All the notes that have not been matched are insertion errors (unmatched performance notes) or deletion errors (unmatched score notes). Because the matchers we discuss try to maximize the number of notes in the match in different ways, false alarms or misses occur in different situations and the number of detected errors varies.

A matcher cannot tell the difference between hits, misses and false alarms; it just finds errors. Every miss or false alarm causes extra manual work to repair the match, so the success rate of the matcher is important. Several authors (see, for instance, Puckette & Lippe, 1992) have suggested that adding some kind of intelligence, such as knowledge about timing or structure, to a matcher may increase this success rate. Similarly, Desain and Honing (1992) have stressed the importance of information about the score structure. In this paper, we will investigate in what way the use of score structure information can improve match results.

2 Different Approaches to Score-Performance Matching

There are many approaches to score-performance matching. Matchers have been proposed by Dannenberg (1984), Dannenberg and Mukaino (1988), Puckette and Lippe (1992), and Vantomme (1995). These matchers are concerned with following a score during a real-time performance and are commonly known as *score-followers*. In these cases, knowing where the performer is in the score is more important than making a qualitatively good match.

Other approaches have been proposed by Honing (1990); Large (1993); Hoshishiba, Horiguchi, and Fujinaga (1996); and Heijink (1996). These matchers are concerned with off-line analyses of recorded performances and the quality of the match is much more important than efficiency.

The three approaches that we will compare in this article are the general strict matcher (GSM, based on a matcher that is part of the POCO system; see Honing, 1990), the general Large matcher (GLM, based on a matcher proposed by Large, 1993), and the structure matcher (SM, based on a matcher proposed by Heijink, 1996). The approaches that these matchers use, their design, and their implementation are described in detail in Heijink, Desain, Honing, and Windsor (2000). For a discussion of the relation between the original versions of the matchers, see Desain, Honing, and Heijink (1997). In this paper, we will concentrate on the behavior of the GSM, the GLM, and the SM. It will be described in the remainder of this section, which is summarized from Heijink et al.

The GSM processes both score and performance front to back and note for note. When a performance note and a score note of the same pitch are compared, the matcher considers only one alternative: a match between the notes. When a performance note and a score note of different pitches are compared, the matcher considers two alternatives: the performance note might be spurious (an insertion error), or the score note might have been omitted in the performance (a deletion error).

Because notes of the same pitch are always matched to each other, some match alternatives are never considered. For instance, when the score contains two repeating notes of the same pitch and the performer plays only one, the match alternative where the second score note is matched is never considered, because the matcher processes the first score note sooner. This approach makes the matcher very efficient, but in cases where an error is made in a context where several notes of the same pitch occur (a situation that occurs frequently in music), the matcher is apt to make wrong decisions and may find more errors than the performance contains.

The GLM solves this problem by considering several alternatives when it encounters a performance note and a score note of the same pitch: a match between the two notes, an insertion error, and a deletion error. In addition to these alternatives, the GLM introduces another: a substitution error. This error is a combination of an insertion and a deletion error and is common in music performance (see Palmer & Van de Sande, 1993). When a performance note is compared to a score note of a different pitch, three alternatives are considered: a deletion error, an insertion error, and a substitution error. The GLM is guaranteed to find the best match that satisfies its note order constraints, even if the performance is barely recognizable.

The SM was designed in order to evaluate the contribution of score structure information. It resembles the GLM as far as the allowed matches are concerned: It considers all possible matches between score and performance that satisfy its order constraints. The matchers differ in the order constraints they use. The SM uses voice information annotated in the score and considers parallel events in different voices to be independent in their timing: This enables voices to shift with respect to each other. The information the SM can use is not limited to voice information: The matcher can be extended to use other structural information like types of ornaments and instrument-specific annotations.

3 Material

We tested the matchers using seven different performances of two different scores. The performances to be matched were five performances of the Etude in c minor (Révolutionaire), Op. 10, No. 12, and two performances of the Fantasie Impromptu, Op. 66, both by Fryderyk Chopin. These two pieces were chosen partly because they were used as examples in Hoshishiba et al. (1996), but also because they are complex pieces that are difficult to perform, they contain several voices, and they are usually played with large expressive deviations. Moreover, the Fantasie Impromptu had a polyrhythmic relationship between left and right hands: The left-hand rhythm is based on a beat divided into three equal parts, while the right-hand rhythm is based on a beat divided into four equal parts. Furthermore, the seven performances are from widely available Yamaha Disklavier disks. Table 1 shows the performances with their performers and disk numbers. The performances were converted to standard MIDI files by playing back the performances on a Yamaha Disklavier grand piano and recording from the MIDI out port. All MIDI information other than note onsets and offsets (such as pedal information) was deleted.

Table 1: Performances Used in Matcher Comparisons

Reference	Performer	Disk Number	Track
Etude 1	not specified	YMM 900202	2
Etude 2	Satoshi Ito	YMM 900148	12
Etude 3	Marc LaForet	YPA 1069E	1
Etude 4	Krystof Jablonski	YPA 1070E	27
Etude 5	Yukio Yokoyama	YPA 1100E	8
Fantasie 1	Yukio Yokoyama	YPA 1100E	5
Fantasie 2	Piotr Paleczny	YPA 1077E	3

NOTE — The performances are available on floppy disks from Yamaha Music Corporation. Etude refers to the Etude in c minor, Op. 10, No. 12; Fantasie refers to the Fantasie Impromptu Op. 66.

The scores were standard MIDI files as well. They were checked by hand and automatically annotated with voice and chord structure using POCO, a workbench that consists of a collection of tools that can be used for the analysis, modification, and generation of musical expression in a research context (Honing, 1990). The voice and chord structure annotations were necessarily incomplete. Annotating a score with complete structure information automatically is beyond reach at the moment; it would require an automated music analysis tool. In our view, the problem of score structure annotation is not part of the matching problem. The structure annotations were checked and completed by a trained musicologist and performer who is familiar with the matching process (L.W.).

The matchers are implemented in Macintosh Common Lisp as part of POCO. The tests were run on an Apple PowerMac G4 with 40 Mb allocated to Lisp. The matchers make large demands on the computer's memory. We limited the scores and the performances to the first eighteen bars of the Etude in c minor and the first twelve bars of the Fantasie Impromptu, in both cases corresponding roughly to one page of sheet music. The reason for this is that with 40 Mb of memory allocated to Lisp, only the GSM was able to match the whole pieces. However, we wanted to compare the differences in behavior of the matchers, rather than the differences in efficiency, and the excerpts we used provided enough situations where the matchers behaved differently. The time required for matching ranged from less than 1 second to 7.5 minutes on the configuration described above.

In order to arrive at an objective criterion for evaluation of the matchers, two trained musicians (H.H. and L.W.) matched the performances and scores by hand, independently of each other. The resulting matches are referred to as *hand matches* (HMs). The HMs for the Etude were made by both and they agreed completely. The HMs for the Fantasie were made only by H.H. The HMs are available on request.

In the seven expert performances, the judges found up to 8 errors in one of the Etude performances (performance 4, 465 score notes and 467 performance notes), and up to 19 errors in one of the Fantasie performances (performance 2, 248 score notes and 247 performance notes).

4 Success rate of the Matchers

The automatic matches were compared against the hand matches to see how successful the matchers were. The total amount of notes in all the scores and performances is 5,642. Of those notes, the GSM misinterpreted 66 (1.2%), the GLM misinterpreted 60 (1.1%), and the SM misinterpreted 8 (0.1%).

In the context of music performance research, the interpretation of the match needs to be 100% correct. Every note that is incorrectly interpreted causes extra manual work to correct it, and therefore, it is worthwhile to improve a match from 98.8% correct (the GSM) to 99.9% correct (the SM).

5 Detailed Analyses of Error Disagreement

We will now look more closely at the causes of the disagreement between the HM, the GSM, the GLM, and the SM. All the errors that were found by some, but not by all the matchers were analyzed to determine the cause of the disagreement. The disagreement between the matches was almost always between the HM and the SM on the one hand, and the GSM and GLM on the other hand.

Disagreement between matchers occurred for two reasons: Voice information that was not taken into account and timing information that was ignored. We will compare the behavior of the matchers in detail by discussing examples of the two causes of disagreement.

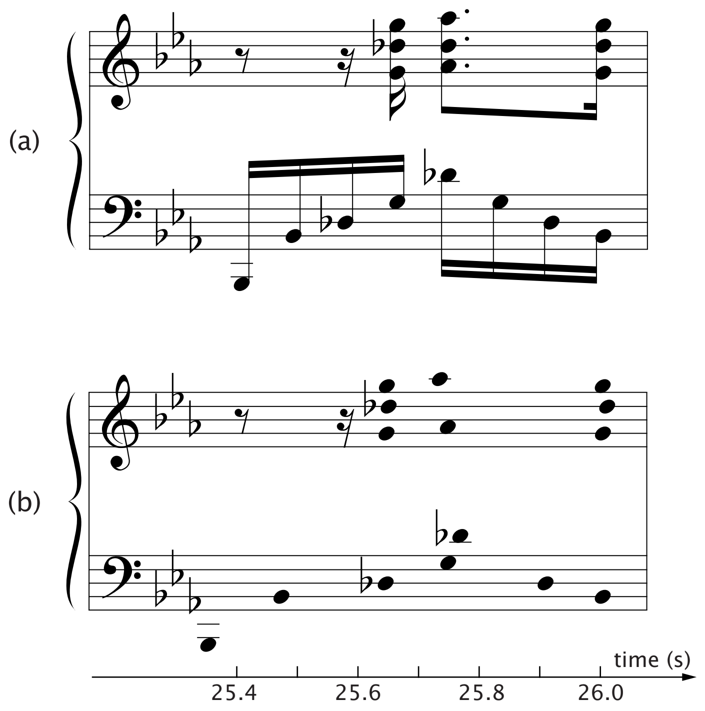

Figure 3: (a) The sheet music score of measure 15, beats 3 and 4 of the Etude in c minor, Op. 10, No. 12 by F. Chopin, and (b) the fourth performance of the Etude (see Figure 4), notated such that the spacing reflects the timing of the performer.

5.1 Voice Information

The first example is taken from performance 4 of the Etude, measure 15, beats 3 and 4. Figure 3 shows the sheet music score for this excerpt, along with a transcription of the performance, spaced according to the performed timing. The textual representations of the score and the performance are displayed in Figure 4. The score times are represented as *bar*:*beat*:*index* (Figure 4, middle column). Because there are four sixteenth notes in one beat, the index goes from 0 to 3.

Middle c is represented by c3. In this case, the result of the SM is correct and therefore equal to the HM; the GLM and the GSM arrive at the same result.

The notes in the right hand (the notes in boldface) run ahead of the notes in the left hand, so that the chord g3, d♭4, g4 at 15:3:3 in the score, as well as the next chord, is about one sixteenth too early. This kind of asynchrony is common in music performance. Moreover, the performance misses the g2 at 15:4:1 and the d♭4 at 15:4:0 in the score. Because this performance of the Etude is very fast (approximately 140 quarter notes per minute) and because the full, pedaled sound of the piano makes the notes of the accompaniment blend into each other, the omitted notes are not noticeable and the timing of the melody sounds expressive instead of erroneous. Therefore, we did not consider the timing of the melody to be erroneous in the hand match. The missed notes are still considered to be deletion errors.

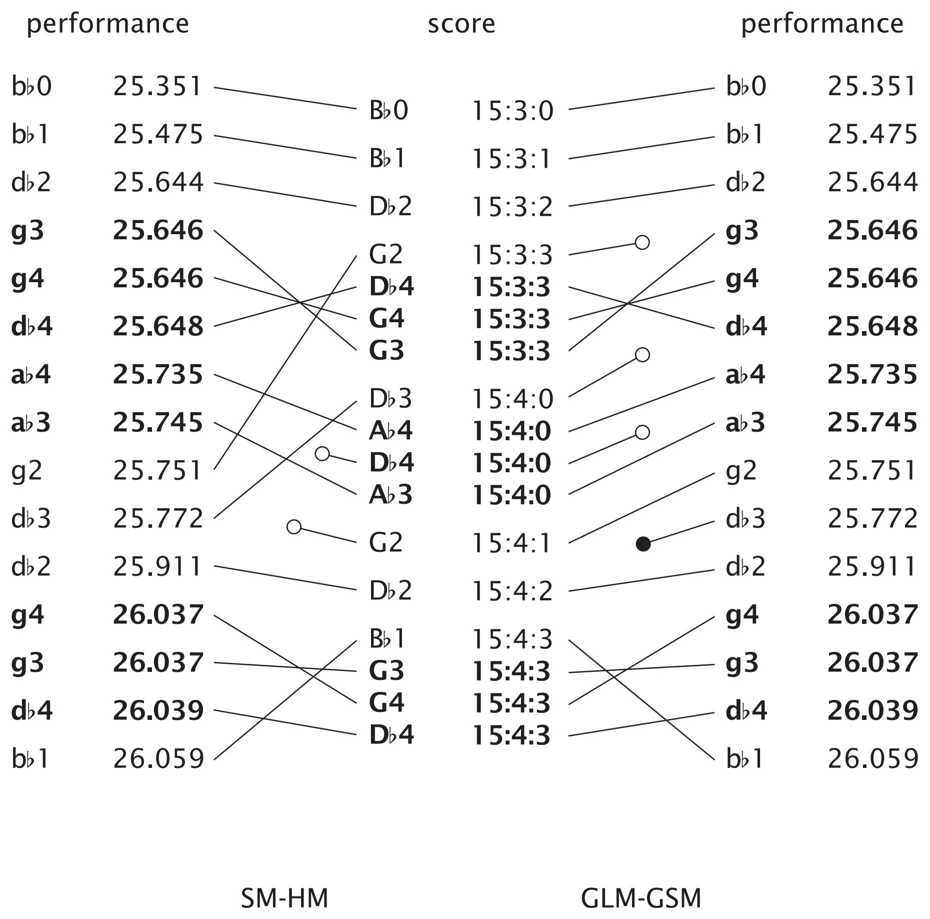

Figure 4: A textual representation of the score in Figure 3 is in the middle. The match between the score and performance 4, found by the general strict matcher (GSM) and the general Large matcher (GLM) is on the right, the match found by the structure matcher (SM), is on the left. The hand match (HM) agrees completely with the match found by the structure matcher. Refer to Figure 2 for an explanation of the lines and circles.

The SM finds the correct match because it only requires notes in the same voice to appear in the order specified in the score. Notes can change order if they are in the same chord or in different voices.

Because the GSM and the GLM consider all notes to be in the same voice, they treat the playing of the second chord (a♭3, d♭4, a♭4 at 15:4:0 in the score) before the g2 at 15:3:3 in the score as an error and either the chord or the accompaniment note has to be excluded from the match. Because all the matchers we discuss regard the match that excludes the fewest notes as the best match, the g2 is treated as a deletion error.

When the g2 at 25.751 in the performance is considered, it can be matched to the g2 at 15:4:1 in the score, but not without violating the order of notes in the accompaniment. This means that the d♭3 at 25.772 in the performance cannot be matched to the d♭3 at 15:4:0 in the score. The decision between matching the g2 or the d♭3 is arbitrary, because the matchers have no concept of timing and the number of notes in the match is equal in both cases. The matchers are specified in such a way that they always choose to match the note that occurs earlier in the performance, so the g2 is matched instead of the d♭3.

This example shows that using voice information is necessary to make the correct decision in cases like this one. It should be mentioned, however, that the structure matcher does not pose any limits on the amount of asynchrony between voices. In this case, the melody was approximately 0.1 sec ahead (one sixteenth note at a tempo of ca. 140 quarter notes per minute), but if the melody were played a whole beat before the accompaniment, the matcher would still match all the notes. Whether this is desirable or not is debatable.

5.2 Timing Information

The second example is taken from measure 10, beats 3 and 4 of the second performance of the Fantasie Impromptu. This example is included because it shows a case where the SM seems to arrive at a better match than the GLM, although the latter one finds the correct match and the SM does not.

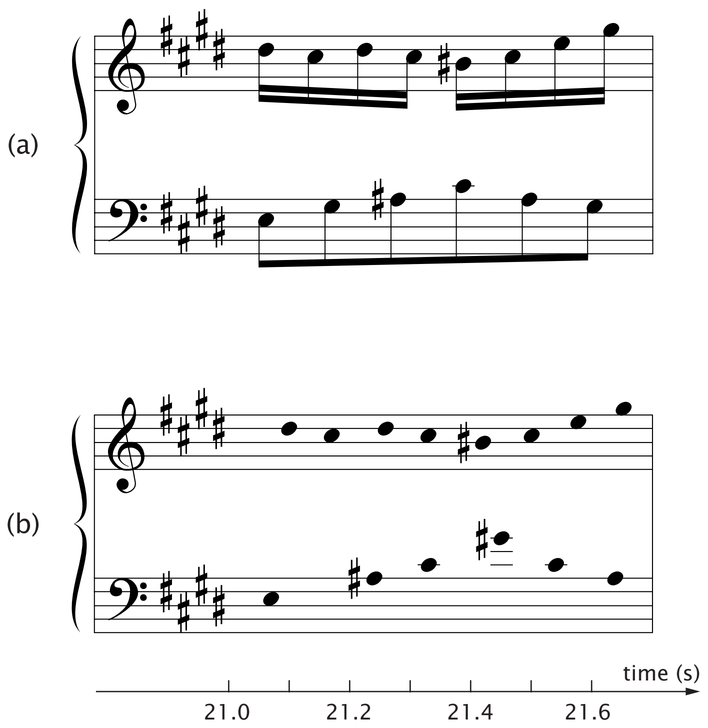

Figure 5: (a) The sheet music score of measure 10, beats 3 and 4 of the Fantasie Impromptu, opus 66 by F. Chopin, and (b) the second performance of the Fantasie (see Figure 6), notated such that the spacing reflects the timing of the performer.

Figure 5 shows the sheet music score and the performance transcription of this excerpt and Figure 6 shows the matches made by the GLM and the SM. The match made by the GSM is not shown, because it is not relevant to the point we wish to make here. The score time representation uses a different grid than in Figure 4, because in this case a beat contains both a triplet and four sixteenths. The index goes therefore from 0 to 11 with the sixteenths at 0, 3, 6 and 9, and the triplet notes at 0, 4 and 8.

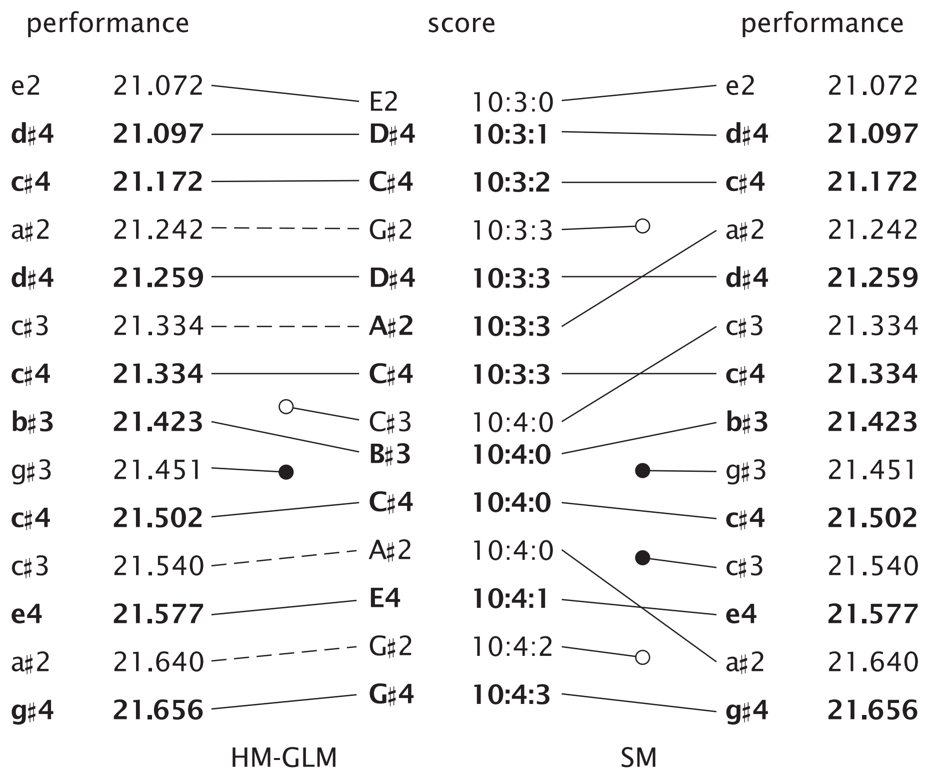

Figure 6: A textual representation of the score in Figure 5 is in the middle. The match between the score and performance 2, found by the structure matcher (SM) is on the right, the match found by the general Large matcher (GLM), is on the left. The hand match (HM) agrees completely with the match found by the general Large matcher. Refer to Figure 2 for an explanation of the lines and circles.

In the situation shown in Figure 6, the performer played the wrong broken chord in the accompaniment: instead of e2, g♯2, a♯2, c♯3, a♯2, g♯2, he played e2, a♯2, c♯3, g♯3, c♯3, a♯2. Because the GLM can interpret combinations of insertion and deletion errors as substitution errors, it finds the correct interpretation of the relation between score and performance. The SM matches more notes, because it allows asynchrony between the two voices.

Both the GLM and the SM try to maximize the number of notes in the match. According to this strategy, the match found by the SM is better than the one found by the GLM, because the match made by the GLM contains 10 errors (a substitution errors is counted as two errors) and the match made by the SM contains only 4 errors.

In making the HMs, we used other information than number of matches and note order information, like timing information, harmonic information and knowledge about errors in general. The GLM found the correct match for the wrong reasons: The errors it found were due to note order violations according to its constraints. It can therefore not be said that the GLM is guaranteed to find the correct match in such cases.

According to the definition of the best match that we used, the SM found the correct match. This example makes clear that the number of notes in the match is not a sufficiently good criterion to find the correct match, but that other information is also necessary. Only one of the matchers we know of, the one proposed by Hoshishiba et al. (1996), uses other information than the number of notes in the match to find the best match. After a match has been found using only pitch and order information, this matcher uses timing information to modify the match and get a better result.

6 Efficiency and Usability of the Matchers

We have seen that the SM is more often correct than the other two matchers. The GSM, however, is much more efficient than either the GLM, or the SM. In this section, we will explain the reasons for this difference.

In the process of searching for the best possible match, the matchers construct a graph that contains partial results. This dynamic programming technique prevents the matchers from having to compute the same result more than once. Because the process of finding the best match consists almost exclusively of building the graph, the number of nodes in the graph is a good measure of the efficiency of the matchers.

The number of nodes in the graph depends on four things: the number of errors detected by the matcher (including false alarms), the place of these errors in the performance, the number of different interpretations of a detected error that the matcher allows, and, in the case of the SM, the number of different voices in the score.

All the matchers consider many possible match alternatives in order to find the one with the fewest errors in it. Every error that is detected increases the number of alternatives that have to be considered and thereby increases the size of the graph.

Table 2: Number of Nodes in Graph and Errors Found per Matcher

	GSM		GLM		SM	
Piece	**Nodes**	**Errors**	**Nodes**	**Errors**	**Nodes**	**Errors**
Etude 1	470	2	3179	2	1849	0
Etude 2	481	7	6223	7	22165	7
Etude 3	486	5	5732	5	9260	3
Etude 4	512	10	12805	10	30985	8
Etude 5	468	1	2484	1	5225	1
Fantasie 1	1595	42	70407	42	20180	14
Fantasie 2	2089	41	55038	41	21039	13

NOTE — GSM, GLM and SM refer to the general strict matcher, the general Large matcher and the structure matcher, respectively. Piece refers to the first column of Table 1.

If an error is detected near the beginning of the performance, the effect on the graph size is small, because the fewer notes seen, the fewer match alternatives to consider. If, however, an error is detected near the end of the performance, the amount of possible alternatives is greater and the effect on the graph size larger.

Finally, the number of possible interpretations of a combination of a performance and a score note differs per matcher. The GSM allows two interpretations for notes of different pitches and only one for notes of the same pitch. The GLM and the SM allow more interpretations of notes of the same pitch: they might represent an undetected error (a miss). The GLM also allows one extra interpretation for notes of unequal pitch: They might represent a substitution error.

The SM allows, in principle, many other interpretations for notes of different pitch. The number of possible interpretations depends on the number of voices in the score and the user-defined structural annotations. It is beyond the scope of this article to explain this in more detail; it is fully described in Heijink et al.(2000).

Table 2 shows the number of nodes in the graph and the number of errors found for each performance per matcher. It is clear that for every matcher, the number of errors increases the number of nodes in the graph. Because the errors occur in different places in different performances, the increase is not always obvious; for instance, the SM finds 14 errors in Fantasie 1 and 13 errors in Fantasie 2, yet the graph size for Fantasie 2 is greater than that for Fantasie 1.

The mean graph size differs greatly between the Etude performances and the Fantasie performances for both the GSM and the GLM. The reason for this is that these matchers do not use voice information. Both the Etude and the Fantasie are two-voiced, but the Fantasie has a more complex relationship between the voices than the Etude does. Therefore, the GSM and GLM find more false alarms in the Fantasie than in the Etude performances. Because the SM does use voice information, the difference in structure between the Fantasie and the Etude does not have an effect on the graph size.

7 Discussion

We have compared three approaches to matching notes in a musical performance with the corresponding notes in the score. Although a simple problem at first sight and an easy task for experienced musicians when visually inspecting the MIDI information, matchers frequently do not find the correct match, especially when the performance exhibits extreme use of expressive timing, when performance errors occur, or when there are underspecified ornaments in the score.

The use of structural annotations in the score is necessary to find the correct match in situations where the score contains more than one voice, which is a situation that occurs frequently in music. We have shown that the GSM and the GLM cannot deal well with voice asynchrony and that the SM can.

Annotating a score with voice information is, however, a time-consuming task to do by hand. Fortunately, it only needs to be done once for each score, after which that score can be used in matching many performances. Automating the annotating process fully would mean developing an automated music analysis tool: At the moment, this is beyond reach. However, we are designing and implementing tools in POCO (the workbench for expression analysis, modification, and generation; see Honing, 1990) to at least help the user to add structural information like bar, phrase, and voice information to a score.

The efficiency of the matchers is still a problem. The GLM and the SM compute much unnecessary information, because when a performance note is compared to a score note of the same pitch, they should be matched to each other in most of the cases. Only by considering the possibility of insertion and deletion errors in these cases are these two matchers always guaranteed to find the best match that satisfies their respective order constraints. We are investigating the possibilities of optimization techniques that may reduce the graph sizes considerably.

The matchers and related tools are part of POCO. POCO is available on an Apple Macintosh platform from www.nici.kun.nl/mmm. There are no plans of porting POCO to different platforms, but work is being done to make POCO and its matching facilities accessible over the World-Wide Web. Progress on this will be reported at the above-mentioned Internet site.

In conclusion, we think that the matchers we discussed in this article are valuable tools in music performance analysis. The SM finds the correct interpretation of the relation between score and performance notes in more than 99.8% of the cases in the pieces we considered. Finding the correct interpretation of an error in all possible cases is beyond the reach of any matcher at the moment, but the use of timing information is probably part of the solution to this problem.

8 References

Cormen, T. H., Leiserson, C. E., & Rivest, R. L. (1990). *Introduction to algorithms.* Cambridge, MA: MIT Press.

Dannenberg, R. (1984). An on-line algorithm for real-time accompaniment. In *Proceedings of the 1984 International Computer Music Conference* (pp. 193–198). San Francisco: ICMA.

Dannenberg, R., & Mukaino, H. (1988). New techniques for enhanced quality of computer accompaniment. In *Proceedings of the 1988 International Computer Music Conference* (pp. 243–249). San Francisco: IMCA.

Desain, P., & Honing, H. (1992). *Music, mind and machine: Studies in computer music, music cognition and artificial intelligence.* Amsterdam: Thesis Publishers.

Desain, P., Honing, H., & Heijink, H. (1997). Robust score-performance matching: Taking advantage of structural information. In *Proceedings of the 1997 International Computer Music Conference* (pp. 337–340). San Francisco: ICMA.

Gabrielsson, A. (1999). The performance of music. In D. Deutsch (Ed.), *The Psychology of Music* (2nd ed., pp. 501–602). San Diego: Academic Press.

Heijink, H. (1996). *Matching scores and performances.* Unpublished master's thesis, University of Nijmegen. (Available in html from www.nici.kun.nl/mmm).

Heijink, H., Desain, P., Honing, H., & Windsor, L. (2000). Make me a match: An evaluation of different approaches to score-performance matching. *Computer Music Journal*, **24**(1), 43–56.

Honing, H. (1990). POCO: An environment for analyzing, modifying, and generating expression in music. In *Proceedings of the 1990 International Computer Music Conference* (pp. 364–368). San Francisco: ICMA.

Hoshishiba, T., Horiguchi, S., & Fujinaga, I. (1996). Study of expression and individuality in music performance using normative data derived from MIDI recordings of piano music. In *Proceedings of the 4th International Conference on Music Perception and Cognition* (pp. 465–470). Montreal: McGill University, Faculty of Music.

International MIDI Association (1983). *MIDI musical instrument digital interface specification 1.0.* Los Angeles: Author.

Large, E. W. (1993). Dynamic programming for the analysis of serial behaviors. *Behavior Research Methods, Instruments, & Computers*, **25**, 238–241.

Palmer, C. (1997). Music performance. *Annual Review of Psychology*, **48**, 115–138.

Palmer, C., & Van de Sande, C. (1993). Units of knowledge in music performance. *Journal of Experimental Psychology: Learning, Memory and Cognition* **19**, 457–470.

Puckette, M. & Lippe, C. (1992). Score following in practice. In *Proceedings of the 1992 International Computer Music Conference* (pp. 182–185). San Francisco: ICMA.

Roads, C. (1996). *The computer music tutorial.* Cambridge, MA: MIT Press.

Vantomme, J.D. (1995). Score following by temporal pattern. *Computer Music Journal*, **19**(3), 50–59.