

Towards an Intelligent Score Following System: Handling of Mistakes and Jumps Encountered During Piano Practicing

Mevlut Evren Tekin, Christina Anagnostopoulou, Yo Tomita

Sonic Arts Research Centre,
Queen's University of Belfast,
BT7 1NN, Belfast, United Kingdom
{e.tekin, c.anagnostopoulou, y.tomita}@qub.ac.uk

Abstract. Score following has been an important area of research in AI and music since the mid 80's. Various systems were developed, but they were predominantly for providing automated accompaniment to live concert performances, dealing mostly with issues relating to pitch detection and identification of embellished melodies. They have a big potential in the area of education where student performers benefit in practice situations. Current accompaniment systems are not designed to deal with errors that may occur during practising. In this paper we present a system developed to provide accompaniment for students practising at home. First a survey of score following will be given. Then the capabilities of the system will be explained, and the results from the first experiments of the monophonic score following system will be presented.

1 Introduction

A score following system monitors a live performance mainly for the purpose of providing an automatic accompaniment to live performers. The first examples of score following systems were developed by Dannenberg [4] and Vercoe [15] in two different studies and were capable of handling the input from a monophonic sound source. Subsequent score followers were polyphonic and had extended capabilities to handle ornaments such as trills and mordents as well as more widely dispersed pitch clusters such as glissandos [1].

Although score following systems were originally invented for the live performances, they also have potential for providing the core function for tutoring systems and performance evaluation tools. Based on his score following systems, Dannenberg [7] has developed the Piano Tutor, providing guidance to students. The Piano Tutor is an expert system that uses score following techniques to monitor student progress and identify weaknesses. When the students successfully complete a set of exercises, the expert system then advances on to a higher level of exercises. Similarly, Bora's [2] system is also built to evaluate student performances, although it does not provide any further tutoring driven by an expert system [2].

All the above systems rely on MIDI signals as input and they use string matching methods. More recently, new systems have been developed that have the capability to handle audio input captured by microphone: they use stochastic pitch detection algorithms when receiving the input from their sound sources [3], [11], [12], [13]. Their main strength, on the one hand, is the ability to follow the performances that do not depend on the fixed pitches. On the other hand, they are incapable of handling even the shortest of jumps in the performance, which is the main weakness. They also need to be trained through several rehearsal sessions in order for them to be able to adjust the state transition weights to follow the performance precisely.

MuseBook Score, which is marketed as an automatic page-turning system for pianists, is another score following system. It follows the performance by listening to pitch. According to AMuseTec website [16], the system is capable of following the performance and displaying the current note being played by the pianist. However, the system does not offer any accompaniment features in live performance. They acknowledge that it is due to the delay caused during the pitch detection process. Although there is some video demonstration of the running system at AMuseTec website, there is no published information about the techniques used in the product.

The idea for score following systems originates from a very successful product called “Music Minus One” (MMO), which is in short a special recording of musical pieces (piano concertos and violin sonatas are two of the most popular genres) made for an instrumentalist who does not have accompanists during the practice sessions [13]. Normally MMO comes with two versions of the same piece, the one complete performance serving as a guide as to how the ensemble should be worked out, and the other the accompaniment only. It assumes that the performer of the featured instrument plays the MMO recording on the stereo and plays along with the second version of recording, trying to fill in the solo part that is deliberately omitted from the recording. While the concept is excellent, its limitations are obvious. Firstly, MMO does not take into account the fact that the learners cannot perform the piece at the set tempo from the outset; students may need to learn to play slowly first. Secondly, it does not allow the student to stop, make mistakes and repeat challenging sections.

Our score following system is designed to be used by students learning the pieces on the piano. In concert performances the number of errors will be very small and jumps are not normally expected. Based on this assumption, most score following systems that are developed for concert performances focus on the problems related to pitch detection and the identification of ornaments. Our approach deals with significant jumps in the score, repetition, in addition to the sudden changes of tempo, wrong pitches and identification of ornaments that had previously been achieved with some degree of success by the previous scholars in the field.

2 The System

In this paper we describe our system, which is aimed primarily at students engaged in practicing the piano. Its main objective is to facilitate their practice by providing a reliable accompaniment to the students learning at home environment. The system should be able to cope with a large number of mistakes: not only pitch or rhythm but also jumps, repeats and sudden tempo changes that are made deliberately or unconsciously during the practice. It may be worth adding that –ornaments especially

those of technically demanding ones— are often played erratically, and for this reason students might be advised to omit them during the early stages of practicing. The system should be designed to tolerate such interpretative problems.

2.1 Input/Output

For the present work we selected a Yamaha Disklavier Pro series piano, which has a MIDI input and output. The student pianist (user) plays the piece on the piano, and the system provides the accompaniment, which is also played on the same instrument. The system can provide the suitable accompaniment for left hand and right hand practices. The user can also choose between two different tempo options, which will be explained in Section 2.4.

2.2 The Score Following Algorithm

Common Practice Notation (CPN) is a universal notational system representing music in written form. Because it reduces the principal dimensions of music to pitch and duration, it represents music in a very compact way. The continuous musical stream is quantized into discrete notes, resulting in losing some information originally present in music. However, this is not a problem as part of the lost information about the music can be reconstituted through the ‘interpretation’ of trained performers. To help the performers recover this lost information, CPN also makes use of additional expression marks. Thus although the basic unit representation is heavily quantized, different performers can give different performances by interpreting these notational symbols and instructions [8].

Unlike CPN, MIDI does not deal with any expression marks and instructions which are open to interpretation. Instead, all these are explicitly spelt out in the MIDI score (such as trills and crescendos). However, in actual performances, ornaments are seldom played identically, even by the same person. Therefore, it would be impractical to encode a single interpretation of an ornament in the MIDI file as it will deny the player any licence to have his or her performance style. For this reason, the MIDI files the system uses do not include any ornaments, allowing the players to add their own interpretation to the performance. The way the system handles ornaments is discussed in section 2.3.

In our system, the music is represented as a series of events. The score to be followed is represented by the tuple (*pitch,duration*) in every event. To each event a list of accompaniments is attached. This list contains pitch, duration and note_on/note_off messages for the accompaniment to be played within the corresponding event.

The system is a multi-agent system, where agents represent single events of the score representation. Each agent is sequentially linked to the next one in a manner of a linked list. We call this linked list the *score array*. Each agent holds the pitch information and continuously monitors the performance. When a student plays a note, all agents corresponding to the pitch being played are activated to their maximum activation values. When a key is released, the previously-activated agents are deactivated. But the activation values of the deactivated agents are not assigned to the

value of zero; instead, the activation values decrease slowly until they reach zero. In a musical composition, the same notes are used many times, hence many agents being activated at a time. But as the student plays along the piece, a region with high activity appears in the score array. The agent having the highest activation value is picked up as the score location by the system.

There are two different strategies for decreasing the activation values. In the first one, the activation values are decreased every time the user plays a new note. In the second one the activation values are decreased at user defined time intervals (200 milliseconds – 1000 milliseconds). The first option acts like a “pause” key. If the user stops and does not play for a long time, the system resumes playing from the last position when the user starts playing again. However, if the second option is selected, the player has to play continuously or otherwise, the system will reset itself. The first option is more successful at responding to repeats while the second one can pick up new start locations better. The second option is more suitable for concert ready performances.

The communication between the computer and the piano is provided by a USB MIDI interface which has the unidirectional delay of 110 milliseconds. Because of this communication delay, we had to skip a validation step which was originally incorporated in our score following algorithm. In the original algorithm, the system first checked the key pressed by the player before sending the accompaniment data to the piano. If the pressed key was identical with the expected pitch, the system sent the appropriate accompaniment to the piano. However, due to the delay, the accompaniment was heard approximately 230 milliseconds too late. For this reason, in the current system as soon as the next expected note is computed by the system, without carrying out any validation process the accompaniment is sent 110 milliseconds before the player is expected to hit the key.

The system makes use of a *confidence mechanism* which acts like a short term memory. This mechanism enables the system to reliably follow a piece with number of repetitions by reducing the search space. The initial experiments without the confidence mechanism were carried out with pieces by Beethoven, Mozart and J.S. Bach. The music was played with one hand only on the piano and the system was assigned to provide the other hand accompaniment. Although the system performance was satisfactory with Beethoven and Mozart, it encountered some difficulties with the music of Bach. One of the pieces we used was the Prelude in C minor from *the Well-Tempered Clavier, Book I* (BWV 847/1) which is based on the broken-chord motif in various keys. It would seem that the repetition of the identical melodic patterns occasionally caused the system to lose the score position even if the player was not making a jump or an error.

Repetition is an inherent property of music. Many traditional musical forms contain sections that repeat the materials used earlier in the piece. For example, a traditional Rondo would be expected to have a structure of A-B-A-C-A (and so forth), where all As are usually exact repetitions in the same key. For our score following system, this may create an ambiguity. This problem has been resolved by including a confidence mechanism, which increases the weight of a specific section when its prior section is followed successfully. So, for example, when section B is being played, the weight of the second A section is then increased. But when the player plays A, the system picks one of the A sections randomly as it would not have sufficient clues to determine as the weight of all A sections would be the same in this case. As soon as

the player starts playing the next section—either B or C—the system can then decide within a single note which A section it played.

The confidence mechanism artificially increases the weight of the most active region on the score array so that the competing hypotheses—i.e. other possible candidate locations like as given in the previous example—would not be activated as a jump destination. To allow the score follower to react to a jump, this active region with increased weight receives heavier penalties—i.e. the active region receives an instruction to decay its strength faster than the competing hypotheses—than the other possible candidate locations when the player makes a mistake or jumps to another location of the score.

2.3 Handling of Trills and Ornaments

An important issue in performance is ornamentation. It often varies from edition to edition, and different performers may choose different ways of interpreting and executing ornaments. Score following systems should be able to cope with variations in the interpretation.

Dannenberg [5] uses a preprocessor to handle the problems caused by trills and ornaments. The preprocessor which contains a finite state machine is responsible from listening to the input and has two internal stages called “normal” and “trill/glissando”. The score follower, which is referred as the matcher, receives the performance data from the preprocessor. When the preprocessor detects a special signal in the score like a trill or glissando, it changes its internal state to “trill/glissando” and stops sending any data to the matcher. The preprocessor does not change its state to normal and start sending data to the matcher until it receives a longer note or the next note after the trill/glissando is performed near its expected time.

In our approach, as mentioned above, all ornaments are omitted from the stored data representation of the score. This allows students to add their own ornaments to the musical piece; at the same time, it adds greater flexibility by allowing the user to skip ornaments, or play them incorrectly.

The system employs two heuristics in order to be able to identify the ornaments performed by the performer:

- the notes, performed unexpectedly, are around the expected key
- the ornament should end with the expected note

These heuristics deliver successful results in case of unexpectedly performed ornaments (Section 3.3).

2.4 Tempo

One of the most significant issues in autonomous accompaniment systems is the process of beat tracking. The system requires a constant notion of tempo to be able to play in synchronization with the performer. Although in a well-polished performance some tempo variations are frequently introduced by players, during the practice sessions tempo variations often occur at the technically challenging passages, and can be sudden and substantial.

The system implements two contrasting methods of handling the tempo variations: (1) user dependent tempo and (2) forced-tempo. The user has to choose one of these options at the beginning of the practice session.

When practising a piece, some students might play the technically demanding passages slower. In user-dependent tempo, the system instantly adjusts the tempo if the player makes a sudden change.

If the forced-tempo option is selected, the system does not change its tempo suddenly when detecting a sudden tempo change, but it adopts the new tempo gradually. This behaviour forces the student to catch up with the accompaniment. If the tempo changes are minor, caused for example by expressive performance, the system will change its tempo accordingly. The forced tempo option can be useful for the students who feel comfortable with the performance of the musical material and need to push the practised piece to the expected final speed.

To calculate the tempo changes in both cases, weighted averaging is used. In user dependent tempo, the last tempo changes have a higher weight than the previous ones. However, in forced tempo, although the last tempo changes also have a relatively higher weight, the weights are not as distinct as the ones in user dependent tempo.

3 Results

To illustrate the system's performance, we use the Prelude from English Suite in A minor (BWV 807) by J. S. Bach. While the player performed only the right hand part of the piece, the system accompanied the player with the left hand part. Although the right hand part is mostly monophonic, there are a few chordal passages. To be able to run the experiment we edited the piece and performed only the top voice in these passages. The results presented here make use of the confidence mechanism. The activation values are decreased every time the player presses a new key on the piano.

3.1 Experiment 1: Performance without Mistakes

In the first experiment the player simply played through the whole piece from the beginning without making any errors or jumps. The system was able to accompany the player from the beginning to the end without making any mistakes.

3.2 Experiment 2: Performance with Wrong Notes and Jumps

Repeats are the most common type of jumps that one could make during the practice. To be able to test the reaction of the system in case such repeats occur, we prepared a scenario with some sections repeated over and over. These repeats contained some missing/wrong notes with their number decreasing in each repeat.



score performed	d f c f b f c# f d f c f b f c d d f c f b f c f b f a f
expected accompaniment	b - a - g# - - - b - a - g# - a - b - a - g# - a - b - c -
system response	- - a - g# - a - - - a - g# - a - - - b - g# - a - b - c -
	I II III IV

Fig. 1. Excerpt from the Prelude from the *English Suite in A minor* (BWV 807) by J. S. Bach and the results from experiment 2. The highlighted regions marked I, II, III and IV show the difference between the expected accompaniment and the actual system response

Fig. 1 shows the actual score, score performed, expected accompaniment and system response of an example scenario of the practice described above. In this scenario, the student practises this particular bar over and over again until she or he can play the passage without mistakes. This example actually records that this student is making fewer mistakes at each repeat. The regions marked on the table highlight four areas of mismatches between the expected accompaniment and system response, which we discuss below.

When the player starts playing from a random point in the score, the system needs to ‘listen’ for a few notes in order to find the score location. During this time the system is not be able to provide any accompaniment (Region I, Fig. 1).

Region II shows another mismatch. The student here made a mistake, adding a sharp to the c of the score. Although the expected accompaniment would be silent, the system actually plays **a**, which would be the accompaniment to the correct **c** natural note. This is because the confidence mechanism tries to tolerate the error made by the player here and continues to provide an accompaniment despite the error.

The mismatch in Region III is caused by the jump performed by the player. The system requires sufficient clues, i.e. series of pitches, to be able to determine the position on the score.

In Region IV, the player starts repeating the section again. The system starts responding with a wrong note (**b** instead of **a**) after resuming the accompaniment. However the system resumes playing the correct accompaniment within two extra notes.

3.3 Experiment 3: Handling the Trills

In the third and last experiment, we tested the system’s abilities to handle ornaments and trills by adding some trills which were not defined in the score. The results of this

experiment are summarised in Fig. 2. Although the trills were not indicated in the stored data representation, the system was able to continue to provide an acceptable accompaniment, skipping a single note only (highlighted with label I in Fig. 2). Though trills are detected as wrong notes, the heuristics included disable the confidence mechanism temporarily to avoid it changing the weights of other notes so that a jump will not occur. The trills were added at random.

score performed	c b a g# a b c d c a b g a b c a g# a c a b g a f# g# f e d c b c a b a b g# a
expected accompaniment	e d c e a c f b e c d b c b a c e a e c d b c a b a b g# a - d - e - e - a
system response	e d c e a - f b e c d b c b a c e a e c d b c a b a b g# a - d - e - e - a

I

Fig. 2. Excerpt from the Prelude from *English Suite in A minor* (BWV 807) by J. S. Bach and the results from experiment 3. In the first row of the table, the trill marks show the random trills added to the piece by the performer. The difference between the expected accompaniment and system response is highlighted

4. Discussion and Further Work

The above results demonstrate how the system copes with the various situations that can occur during piano practicing, such as repeats, wrong notes and interpretation variations in performing the ornaments. When there are no mistakes in the performance, the system performs accurately. Where trills and ornaments were encountered the system was also able to provide the right accompaniment. In the case of repetitions of the same passage played with a number of mistakes, the system needed one or two successive notes to catch up with the performer. This was because the system needed to resolve the ambiguity caused by the new jump location. This is also what would be expected to be the case with a human accompanist.

In this paper we presented a piano score following system which is designed for piano students practicing on a MIDI instrument. However, instruments with MIDI interfaces are not very common and the existing ones are generally far more expensive than the more traditional instruments. In order to create a system that can be used with a wider array of instruments, a pitch detector can be used.

The polyphonic version of the described score following system is currently under development. One of the important advantages over the monophonic version will be the inclusion of rhythmical information to the score following task. This will allow students to practise a wide range of piano repertory without any restrictions.

The present score following system, which only makes use of the pitch information from the performance, has so far produced promising results. With the added capabilities of polyphonic music support, using duration information for increased robustness and a pitch, the system will be a valuable learning tool for many music students.

References

1. Bloch, J.J., Dannenberg, R. (1985). Real-Time Computer Accompaniment of Keyboard Performances. Proceedings of the 1985 International Computer Music Conference, International Computer Music Association, pp. 279-289.
2. Bora, U., Tufan, S., Bilgen, S. (2000). A tool for comparison of piano performances. Journal of New Music Research, 29, No.1, pp.85-99
3. Cano, P., Loscos, A., Bonada, J. (1999). Score-Performance Matching Using HMMs. Proceedings of the 1999 International Computer Music Conference, International Computer Music Association, pp. 441-444.
4. Dannenberg, R. (1984). An Online Algorithm for Real-Time Accompaniment. Proceedings of the 1984 International Computer Music Conference, International Computer Music Association, pp. 193-198.
5. Dannenberg, R., Mukaino, H. (1988). New techniques for Enhanced Quality of Computer Accompaniment. Proceedings of the 1988 International Computer Music Conference, International Computer Music Association, pp 243-249.
6. Dannenberg, R. (1991). Recent Work in Real-Time Music Understanding by Computer. Music, Language, Speech and Brain, Wenner-Gren International Symposium Series, Sundberg, Nord, and Carlson, ed., Macmillan, 1991, pp.194-202.
7. Dannenberg, R. et al. (1993). Results from the Piano Tutor Project. Proceedings of the Fourth Biennial Arts and Technology Symposium, (March 1993), pp. 143-150
8. Loy, G., Abbott, C. (1985). Programming Languages for Computer Music Synthesis, Performance, and Composition. Computing Surveys, Vol. 17, No. 2, June 1985
9. Orio, N., Lemouton, S., Schwarz, D. (2003). Score Following: State of the Art and New Developments. Proceedings of the Conference of New Interfaces for Musical Expression, NIME, Montreal, 2003, pp. 36-41.
10. Puckette, M., Lippe, C. (1992). Score Following in Practice. Proceedings of the 1992 International Computer Music Conference, International Computer Music Association, pp.182-185.
11. Puckette, M. (1995). Score Following Using the Sung Voice. Proceedings of the 1995 International Computer Music Conference, International Computer Music Association, pp. 175-178.
12. Raphael, C. (1999). Automatic Segmentation of Acoustic Musical Signals Using Hidden Markov Models. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.21, No. 4, pp. 360-370.
13. Raphael, C (2001). A Probabilistic Expert system for Automatic Musical Accompaniment. Journal of Computer and Graphic Stats. vol.10 no.3, pp.487-512
14. Roads, C. (1985). Research in Music and Artificial Intelligence. Computing Surveys, Vol. 17, No. 2, June 1985
15. Vercoe, B. (1984). The Synthetic Performer in the Context of Live Performance. Proceedings of the 1984 International Computer Music Conference, International Computer Music Association, pp.199-200.
16. AMuseTec website: <http://www.musebook.com>