PERFORMANCE FOLLOWING: TRACKING A PERFORMANCE WITHOUT A SCORE

Adam M. Stark and Mark D. Plumbley

Queen Mary University of London
Centre for Digital Music
Mile End Road, London E1 4NS

ABSTRACT

We present a technique for following a live performance in the situation where a score is not available. Making use of a local alignment between recent and longer term musical information, we place the present in the context of the past, allowing the prediction of future performance information. By representing music as sequences of beat-synchronous features we reduce the size of the information needed to represent the performance and allow performance following in real-time to occur.

Index Terms**—** Music, Real time systems

1. INTRODUCTION

In recent years, much research has been undertaken into automatic musical accompaniment, in particular *score following* [1]. Score following is the matching of events in a musical performance to notes, or groups of notes, in a score. The positional information provided by this matching process allows for the future of the performance to be determined and a coherent musical accompaniment to be played.

In this paper we address the problem of generating a coherent musical accompaniment to a performance for which no score exists. This scenario happens often amongst human musicians - either no score is available, as in much rock or pop music, or the music may be improvised. This problem, which we will call *performance following*, can be approached by taking advantage of the fact that much music contains repetitions of musical phrases. Human musicians are capable of recognising these repetitions and making predictions of the future performance based upon the past. Here we seek to automate this performance following process.

Our problem is explained as follows: Considering the previous few minutes of a performance and the most recent few seconds, by assuming the music contains some repetition, can we predict the future of the performance by placing our shorter fragment of music in the context of the longer one? If so, then we can 'learn' a piece of music 'on the fly' and possibly play a coherent musical accompaniment, such as a bassline or melody, with no prior knowledge in the form of a score. Furthermore, while there has been recent work on offline automatic acompaniment systems [2], in this work we are interested in real-time approaches.

2. ALGORITHMS FOR SEQUENCE ALIGNMENT

Many, but not all, musical styles can be characterised by the repetition of musical patterns. It is possible to represent these patterns as ordered sequences - either of notes, chords or other features. The automatic extraction of information related to these patterns is useful for many applications including music information retrieval, structural analysis and score following.

There exist several algorithms in the field of computational biology for comparing pairs of sequences. Needleman and Wunsch [3] have presented a technique for the global alignment of two sequences of amino acids. First a score matrix is calculated based upon the similarity of the two sequences. Then a dynamic programming technique is used to calculate all possible pathways through the matrix, followed by a traceback step to find the best alignment. This technique was extended by Smith and Waterman [4] to calculate local alignments - that is the highest scoring alignment of two subsequences of two longer sequences.

The application of sequence alignment algorithms to music has been widespread. Mongeau and Sankoff [5] present a technique for the computation of a value of similarity between two musical scores. Sequence alignment techniques have also been applied in the fields of music information retrieval [6] and structural analysis of music [7] .

2.1. Musical Sequence Alignment in Live Performance Systems

There have been several score following applications of sequence alignment to music. Dannenberg [8] uses pitch estimation of monophonic audio to align a performance to a score using sequence alignment techniques. Several techniques to adapt this system to handle polyphonic keyboard performances were later presented [9]. Pardo and Birmingham [10] present a system that allows a polyphonic MIDI performance to be compared to a lead sheet. Dannenberg and

AS is supported by an EPSRC DTA Studentship and MP is supported by an EPSRC Leadership Fellowship.

Hu [11] present a technique for aligning a performance in the form of polyphonic audio to a symbolic MIDI file. Dixon and Widmer introduced a technique for aligning polyphonic audio recordings of different performances of the same piece of music [12]. This algorithm was later adapted and applied to the real-time case of tracking a live performance [13].

These previous techniques for tracking a live performance have largely used a global alignment between the performance and the score, calculating an alignment matrix step by step over the course of the performance. We do not have a global score to match against and so in this paper we present a system that uses a local alignment. This requires us to compute a new matrix at every step to find the best alignment between a shorter sequence and a longer sequence. The result is that we are able to place recent musical information in the context of the whole performance to detect repetitions of previous musical themes, allowing us to predict the future of the performance, with no use of a score. In order to deal with the computation of a matrix at every step during the performance, we represent the performance beat-synchronously, with a single feature for each beat. This reduces the length of sequences considerably.

3. METHOD

3.1. Beat-Synchronous Sequences

Beat-synchronous sequences are sequences calculated from music that has been segmented at the level of the beat such that each symbol represents information about a single inter-beat interval. There are several advantages to representing music beat-synchronously for our application. Firstly, as we are recomputing an alignment at every step, the representation of music beat-synchronously reduces the number of symbols in each sequence and thereby the required computation time as less calculations are needed to make the comparison between sequences. For example, if we have an audio frame size of 1024 samples at a sampling rate of 44.1kHz, then we need 2583 symbols to represent 60 seconds. However, at a tempo of 120bpm, we need only 120 symbols to represent the same amount of time beat-synchronously. Further to this, beat-synchronous symbol sequences will represent the same musical progression with the same number of symbols regardless of tempo. Finally, harmonic changes often occur at beat locations and so we may lessen the possibility of a symbol that has been calculated from a frame containing a harmonic change as the change is likely to take place at the beginning or the end of the beat-synchronous segment.

In order to create beat-synchronous sequences in real-time, we use a beat tracker combined with some harmonic analysis techniques. We will first present a general explanation of our alignment algorithm for simple monosymbolic sequences generated using chord recognition before discussing the extension to polysymbolic sequences of chroma vectors.

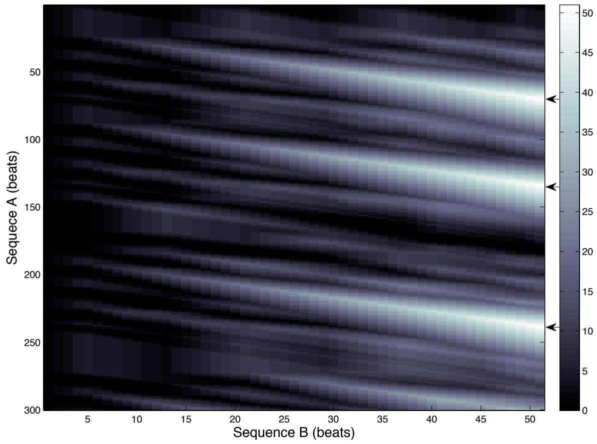

Fig. 1. *The matrix resulting from the comparison of the previous 300 beats of a performance with the previous 50. As can be seen in the final column of the matrix, there are three potential alignments (identified by arrows) for the fragment indicating that the fragment contains a repeated part of the performance.*

3.2. Real-Time Sequence Prediction

We present an adaptation of the Smith-Waterman algorithm for local genetic sequence alignment [4]. Our algorithm takes as input two sequences, $A = a_1, a_2, ..., a_N$ and $B = b_1, b_2, ..., b_M$, where A is a longer sequence of length N and B is a shorter sequence of length $M < N$. The output of the algorithm is the next predicted element of sequence B based upon the local alignment with sequence A. The first step is the calculation of a similarity score matrix, s. For the most simple case, we calculate:

$$s(i,j) = \begin{cases} s_+ & \text{if } a_i = b_j \\ s_- & \text{if } a_i \neq b_j \end{cases} \quad (1)$$

for $1 \leq i \leq N$, $1 \leq j \leq M$, s_+ and s_- are the match and mismatch scores for which we choose $s_+ = 1$ and $s_- = -1/3$. We then calculate the matrix H, with the value $H_{i,j}$ indicating the score for the alignment of two sub-sequences ending in a_i and b_j. We first set all the values $H_{i,0}$, $H_{0,j}$ and $H_{0,0}$ to zero. Then we calculate the rest of the matrix as follows:

$$H_{i,j} = \max \begin{cases} H_{i-1,j-1} + s(i,j) \\ H_{i-1,j} - W \\ H_{i,j-1} - W \\ 0 \end{cases} \quad (2)$$

where $1 \leq i \leq N$ and $1 \leq j \leq M$ and W is the gap penalty for which we choose $W = \frac{4}{3}$, in a similar way to Smith and

Waterman [4]. The gap penalty penalises alignments that contain inserted or deleted symbols.

From the matrix H, an algorithm such as the Smith–Waterman algorithm would find the largest value in the matrix and perform a traceback to compute the alignment. However, we do not wish to compute an alignment, only to identify the next element in the sequence given the values in the matrix. Furthermore, we wish to restrict the alignment to include the rightmost elements of the sequence B as these are the most recent information. As a result, we choose the value

$$t = \arg \max_{1 \leq i < N} H_{i,M} \tag{3}$$

and then predict the next element, $\hat{b}_{M+1}$, of the sequence B to be:

$$\hat{b}_{M+1} = a_{t+1} \tag{4}$$

At each beat, as a new beat-synchronous symbol, c_{new}, is received, we update the shorter sequence B as follows:

$$B = b_2, b_3..., b_M, c_{\text{new}} \tag{5}$$

The prediction is made by comparing sequences A and B and then the new symbol, c_{new}, is added to sequence A in a similar way to equation 5.

Figure 1 shows the resulting matrix of the comparison of the previous 300 beats of a performance with the previous 50 beats. By examining the final column of the matrix, we can see that there are several points at which the fragment has a particularly strong alignment as this fragment is repeated several times. It is possible that a fragment will produce two or more equally strong alignments and so the choice of the earlier or later is a design choice depending upon the application.

4. EXTENSION TO CHROMA VECTOR SEQUENCES

We have seen the implementation of our technique for monosymbolic sequences. In order to consider polysymbolic sequences, we consider sequences A and B as sequences of chroma vectors. A chroma vector is a 12×1 vector whose values represent the energy present in each of the 12 semitone pitch classes found in western music. We achieve this using the chroma calculation technique we developed in [14]. We adapt our algorithm by calculating the similarity score matrix, s, from the inner product of the chroma vectors:

$$s(i,j) = \sum_{n=0}^{P-1} a_i(n) \times b_j(n) \tag{6}$$

where $a_i(n)$ is the nth chroma bin of the ith element of sequence A, $b_j(n)$ is the nth chroma bin of the jth element of sequence B, $1 \leq i \leq N$, $1 \leq j \leq M$ and $P = 12$, the number of bins in each chroma vector. We then proceed, calculating the rest of the matrix as for the monosymbolic version and then predicting the next chroma vector in the sequence.

Song	**Beats**	**Changes**
1. I Saw Her Standing...	84.2% (377/448)	50.0% (28/56)
2. Misery	75.4% (172/228)	45.5% (20/44)
3. Anna (Go To Him)	82.1% (257/313)	58.7% (37/63)
4. Chains	88.8% (270/304)	59.5% (22/37)
5. Boys	93.2% (313/336)	76.6% (36/47)
6. Ask Me Why	73.8% (228/309)	43.8% (28/64)
7. Please Please Me	77.4% (205/265)	55.9% (38/68)
8. Love Me Do	80.4% (270/336)	62.5% (40/64)
9. P.S. I Love You	73.6% (198/269)	47.7% (31/65)
10. Baby It's You	84.5% (240/284)	56.0% (28/50)
11. Do You Want To...	72.7% (141/194)	50.0% (25/50)
12. A Taste Of Honey	56.7% (135/238)	49.1% (54/110)
13. There's A Place	69.5% (178/256)	33.9% (19/56)
14. Twist and Shout	86.9/% (265/305)	82.9% (87/105)
Average	**78.5%**	**55.1%**

Table 1. The results of testing our technique on chord sequence annotations of the album *Please Please Me* by *The Beatles*. The figures are for the number of correctly predicted beats and the number of correctly predicted chord changes.

5. EVALUATION

We assess our technique using hand annotated beat-synchronous chord transcriptions of the songs of the *Beatles* album *Please Please Me*. For each song, we have our system attempt to predict the chord sequence with no prior knowledge.

To measure performance we consider both the percentage of correctly predicted chords and the changes in the ground truth chord sequence, determining the percentage of correctly predicted chords at chord change locations. We choose sequence lengths of $N = 500$ and $M = 20$.

As can be seen in Table 1, the results show that our algorithm was able to correctly predict 55.1% of chord changes across all songs (879 chord changes) despite having no prior knowledge of each piece of music in the form of a score. This result is explained better by Figure 2. This figure shows the ground truth and predictions for the song *Baby It's You*, for which 56% of chord changes were correctly predicted. As can be seen, the system spends the first approximately 110 beats of the song missing chord changes as the system simply outputs the last chord it recognised as it can't find any other alignment. However, as soon as a fragment is repeated, the predictions become accurate. This behaviour is typical for our technique: in general a period of poor performance is followed by much better performance once repetitions in the music have been identified. This has also been the case in informal real-time tests in live performance situations.

In attempting to evaluate performance on this new type of task we are presented with several problems. In particular, in order to produce a beat-synchronous sequence in real-time we require both a beat-tracker and some analysis tech-

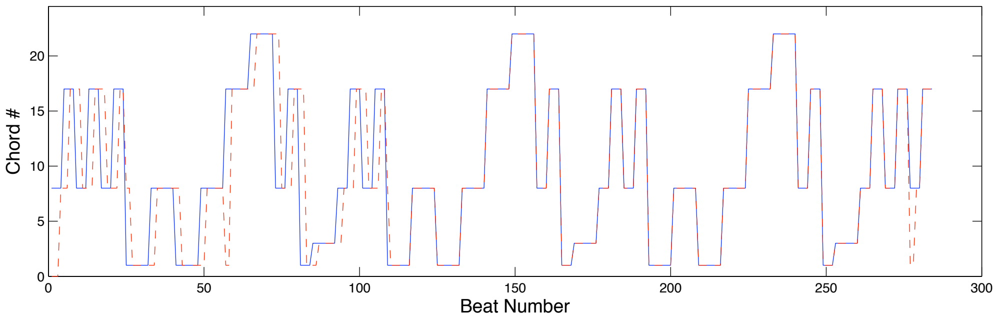

Fig. 2. *The performance of our system on the song Baby Its You. The solid line is the ground truth while the dotted line is the prediction made by the system. As can be seen, after approximately 110 beats the system recognised the repetition of an earlier musical progression and managed to predict the rest of the song almost correctly. Chord numbers 1-12 are C Major through to B Major and 13-24 are C minor through to B minor.*

nique such as a chord detection algorithm. These algorithms will likely not exhibit 100% accuracy and so there will be a compounding of the errors of the analysis techniques and of our sequence alignment technique. While this may well be the case in a real-world situation, it makes it very difficult to assess the performance of our sequence alignment technique independently of the analysis algorithms. For this reason we have evaluated our technique using hand annotated data. Future work will include an extensive evaluation on both symbolic and real-time performance data and a focus on handling errors from analysis algorithms.

The reason for including performance at chord change locations is that much of the data contains repetitions of the same chord for several beats, or even bars. Our experience shows that simply predicting that the chord will be the same as the previous one will achieve a reasonably high score for the percentage of correctly predicted chords overall.

6. CONCLUSION

We have presented a technique for interpreting a live musical performance such that a coherent musical accompaniment may be played, despite the lack of any information in the form of a score. We have shown that musical repetitions can be recognised and so the future of performances can be predicted, creating potential for real-time performance applications capable of improvised accompaniment.

7. REFERENCES

[1] Nicola Orio, Serge Lemouton, and Diemo Schwarz, "Score following: State of the art and new developments," in *New Interfaces for Musical Expression*, 2003.

[2] Ian Simon, Dan Morris, and Sumit Basu, "Mysong: Automatic accompaniment generation for vocal melodies," in *Proc. CHI*, 2008.

[3] Saul B. Needleman and Christian D. Wunsch, "A general method applicable to the search for similarities in the amino acid sequence of two protiens," *Journal of Molecular Biology*, vol. 48, no. 3, pp. 443–453, March 1970.

[4] Temple F. Smith and Michael S. Waterman, "Identification of common molecular subsequences," *Journal of Molecular Biology*, vol. 147, no. 1, pp. 195–197, March 1981.

[5] Marcel Mongeau and David Sankoff, "Comparison of musical sequences," *Computers and the Humanities*, vol. 24, no. 3, pp. 161–175, June 1990.

[6] Ning Hu, Roger B. Dannenberg, and George Tzanetakis, "Polyphonic audio matching and alignment for music retrieval," in *2003 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2003, pp. 185–188.

[7] Roger B. Dannenberg and Ning Hu, "Pattern discovery techniques for music audio," in *Proc. ISMIR*, 2002, pp. 63–70.

[8] Roger B. Dannenberg, "An on-line algorithm for real-time accompaniment," in *Proc. ICMC*, 1984, pp. 193–198.

[9] Joshua J. Bloch and Roger B. Dannenberg, "Real-time computer accompaniment of keyboard performances," in *Proc. ICMC*, 1985, pp. 279–289.

[10] Bryan Pardo and William P. Birmingham, "Following a musical performance from a partially specified score," in *Proceedings of the 2001 Multimedia Technology and Applications Conference*, 2001, pp. 202–207.

[11] Roger B. Dannenberg and Ning Hu, "Polyphonic audio matching for score following and intelligent audio editors," in *Proc. ICMC*, 2003, pp. 27–33.

[12] Simon Dixon and Gerhard Widmer, "MATCH: A music alignment tool chest," in *Proc. ISMIR*, 2005, pp. 492–497.

[13] Simon Dixon, "Live tracking of musical performances using on-line time warping," in *Proc DAFx*, 2005, pp. 92–97.

[14] Adam M. Stark and Mark D. Plumbley, "Real-time chord recognition for live performance," in *Proc. ICMC*, 2009.