ISMIR 2001 Invited Address

Music Information Retrieval as Music Understanding[1]

Roger B. Dannenberg
Carnegie Mellon University
School of Computer Science
Pittsburgh, PA 15213 USA
1-412-268-3827
rbd@cs.cmu.edu

ABSTRACT

Much of the difficulty in Music Information Retrieval can be traced to problems of good music representations, understanding music structure, and adequate models of music perception. In short, the central problem of Music Information Retrieval is Music Understanding, a topic that also forms the basis for much of the work in the fields of Computer Music and Music Perception. It is important for all of these fields to communicate and share results. With this goal in mind, the author's work on Music Understanding in interactive systems, including computer accompaniment and style recognition, is discussed.

1. INTRODUCTION

One of the most interesting aspects of Music Information Retrieval (MIR) research is that it challenges researchers to form a deep understanding of music at many levels. While early efforts in MIR were able to make impressive first steps even with simple models of music, it is becoming clear that further progress depends upon better representations, better understanding of music structure, and better models of music perception.

As MIR research progresses, the community will undoubtedly find more and closer ties to other music research communities, including "Computer Music," probably best represented by the International Computer Music Association and its annual conference [22], and "Music Perception" as represented by the Society for Music Perception and Cognition [25]. While MIR is not the main focus of either of these communities, there is considerable overlap in terms of music processing, understanding, perception, and representation.

The goal of this presentation is to survey some work (mostly my own) in Music Understanding and to describe work that is particularly relevant to MIR. Most of my work has focused on interactive music systems. Included in this work is extensive research on computer accompaniment systems, in which melodic search and comparison are essential components. Other efforts include beat-tracking, listening to and accompanying traditional jazz performances, and style classification of free improvisations.

Along with the preparation of this presentation, I am placing many of the cited papers on-line so they will be more accessible to the MIR community.

My thesis is that a key problem in many fields is the *understanding and application of human musical thought and processing*; this drives much of the research in all fields related to music, science, and technology. This is not to say that these fields are equivalent, but it is important to understand how and why they are related. The work that I describe here shares many underlying problems with MIR. I hope this overview and the citations will be of some benefit to the MIR community.

2. COMPUTER ACCOMPANIMENT

The general task of computer accompaniment is to synchronize a machine performance of music to that of a human. I introduced the term *computer accompaniment* in 1984, but others terms have been used including *synthetic performer* [26], *artificially intelligent performer* [2] and *intelligent accompanist* [5]. In computer accompaniment, it is assumed that the human performer follows a composed score of notes and that both human and computer follow a fully notated score. In any performance, there will be mistakes and tempo variation, so the computer must listen to and follow the live performance, matching it to the score.

Computer accompaniment involves the coordination of signal processing, score matching and following, and accompaniment generation. Because of the obvious similarity of score matching to music search, I will focus on just this aspect of computer accompaniment. See the references for more detail [9, 12].

2.1 Monophonic Score Following

My first computer accompaniment systems worked with acoustic input from monophonic instruments. The system is note-based: the sequence of performed pitches is compared to the sequence of pitches in the score. Times and durations are ignored for the purposes of matching and comparison, although timestamps must be retained for tempo estimation and synchronization.

Originally, I tried to apply the algorithm from the Unix *diff* command, which, viewed from the outside, seems to be perfect for comparing note sequences. Unfortunately, *diff* does not work here because it assumes that lines of text are mostly unique. This led to the exploration and application of dynamic programming, inspired by longest common substring (LCS) and dynamic

[1] Originally published as: Roger B. Dannenberg, "Music Information Retrieval as Music Understanding," in *ISMIR 2001 2nd Annual International Symposium on Music Information Retrieval*, Bloomington: Indiana University, (2001), pp. 139-142.

timewarp algorithms [24]. To my knowledge, this is the first use dynamic programming for melodic comparison.

Recall that LCS computes a matrix of size $m \cdot n$ for strings of length m and n. An important refinement for real-time music recognition is the introduction of a sliding window centered around the current score position. This reduces the computation cost per note to a constant.

This windowing idea could be used in music search applications, especially to compare long query strings to stored strings. The window only affects the result in cases where the match is poor, but presumably these cases are not of interest anyway.

Dynamic programming algorithms typically look at the final $m \cdot n$ matrix to determine the result, but this is not possible in real time. As a heuristic, my score follower reports a match when a "match score" is computed that is higher than any previous value computed so far. The "match score" is essentially the number of notes matched so far minus the number of notes skipped in the score. This formulation compensates for the tendency to skip over notes in order to find a match.

It is perhaps worth noting that matching the performance (a prefix of the score) to the score is a bit like matching a query, which may be a melodic fragment, to a complete melody. Since a fragment may start and end anywhere in a complete melody, we want to compute the least distance from *any* contiguous fragment of the melody, ignoring a certain (but unknown) prefix and suffix of the melody. Dynamic programming, as formulated for score following, can find this best match with a cost of $m \cdot n$, where m and n are the lengths of the melody and query. (Unfortunately, the windowing idea does not seem to apply here because we do not know where the best match will start in the melody.)

Monophonic score following works very well. The original implementation ran on a 1MHz 8-bit processor that performed pitch estimation, score following, accompaniment, and synthesizer control in real time, and fit under an airline seat in 1984! As a historical note, I suggested in a talk given in 1985 [7] that these matching algorithms could be used to quickly search a database of songs. Unfortunately, I missed the opportunity to mention this in my patent [8] or create the first such implementation.

2.2 Polyphonic Score Following

The logical next step in this research was to consider accompanying keyboard performances, and two algorithms were developed for polyphonic score following [3]. Rather than repeat their full descriptions here, I will simply try to give the main ideas and properties of the algorithms. One approach, developed by Josh Bloch, generalizes the idea of "note" to "compound event." A *compound event* is set of simultaneous note onsets, i.e. a chord. The score and performance are regarded as sequences of compound events, and we are essentially looking for the best match. The quality of the match is determined by the number of pitches that match within corresponding compound events minus the number of pitches that are skipped. This is easily solved using dynamic programming, where rows and columns correspond to compound events.

One problem with the preceding algorithm is that it relies upon some process to group events into compound events. We form compound events by grouping notes whose onsets are separated by less than 50 to 100ms. Another algorithm for polyphonic score following was created that forms compound events dynamically. In this algorithm, score events are initially grouped into compound events, but performed events are processed one-at-a-time. What amounts to a greedy algorithm is used to associate performed notes with compound events. Unfortunately, this algorithm does not always find the optimal match because of the heuristic nature of its grouping.

In practice, both algorithms work very well, failing only in (different) contrived pathological cases. Both use the same windowing technique introduced in the monophonic matcher and therefore run in constant time per performed note. The precision of a MIDI keyboard compared to acoustic input, combined with the additional information content of a polyphonic score, makes computer accompaniment of keyboard performances very robust.

These algorithms could be used for music search, but they rely on matching notes as opposed to something more abstract such as harmony. If an improviser plays correct harmonies but in different rhythms or voicings, the match rating might be low. On the other hand, the algorithm can be used as a sort of *diff* on MIDI files, for example to compare different performances [21, 23] or editions. Another interesting application of this technology is in intelligent piano tutoring systems [4, 6, 13].

2.3 Ensemble Accompaniment

With keyboard performance, the right and left hands are generally synchronized, but this is not so true of ensembles. Following and accompanying an ensemble can be accomplished by following each musician separately and then integrating the results [10, 15, 16]. One of the interesting problems encountered here is that different performers may have more or less relevance at any given time. Usually, performers that have performed a note more recently and that are synchronized with other performers are better sources of timing information. The situation changes constantly in a performance as one part assumes prominence and another plays a background role or rests.

In MIR research, it is common to assume music is a totally ordered sequence of notes or features. It might be useful to consider that, in performance, individuals are not always synchronized. Instead, each performer has a separate notion of time and has a strong goal to produce coherent musical gestures. The synchronization of all these independent lines and gestures is a quasi-independent task performed as each performer listens to the others.

2.4 Vocal Accompaniment

In spite of the success of monophonic and polyphonic matchers for score following, these techniques do not work well for vocal soloists. The main problem is that vocal melodies are difficult to segment into discrete notes, so the data seen by the matcher has a high error rate. Similar problems occur in MIR systems, and a more detailed analysis can be found in Lorin Grubb's thesis [19].

Given that discrete string matching methods cannot be applied to vocal music, Grubb's solution [18, 20] is based on the idea of using probability theory to form a consistent view based on a large number of observations that, taken individually, are unreliable. The probabilistic framework allows the system to be trained on actual performance data; thus, typical performance errors and signal processing errors are all integrated into the framework and accounted for.

The system effectively matches pitch as a function of time to the score, but rather than use dynamic time warping, Grubb's system represents score position as a probability density function. This density function is updated using a model of tempo variation, accounting for natural variations in performed tempo, and a model of pitch observations, accounting for the natural distribution of pitch around the one notated in the score. In addition, phonetic information and note onset information can be integrated within the probabilistic framework [17]. This work forms an interesting basis for MIR using vocal queries.

3. Listening to Jazz

It would be wrong to assume every MIR query can be formulated as a melodic fragment. Similarly, it is restrictive to assume accompanists can only follow fully notated music. What about jazz, where soloists may follow chord progressions, but have no predetermined melody? Working with Bernard Mont-Reynaud, I developed a real-time blues accompaniment system that analyzed a 12-bar blues solo using supervised learning to characterize typical pitch distributions and a simple correlation strategy to identify location [11]. This work also included some early beat induction techniques [1]. It seems unlikely that these techniques will be directly applicable to MIR systems, but the general idea that improvised solos (or even stylized interpretations of melodies) can be understood in terms of harmonic and rhythmic structure is important for future MIR research.

4. Style Classification

An underlying structure of beats, measures, harmony and choruses supports traditional jazz solos. I am interested in interactive improvisations with computers where this structure is absent. Instead, I want the computer to recognize different improvisational styles, such as "lyrical," "syncopated," and "frantic" so that the improviser can communicate expressive intentions to the computer directly through the music, much as human musicians communicate in collective improvisations. This goal led to work in style classification using supervised machine learning [14]. This work has obvious applications to music search where the object is to retrieve music of a certain genre or style. We were able to obtain good classification rates on personal styles using quite generic features obtained from a real-time pitch analyzer. Recognition was based on only 5 seconds of music to minimize latency in a real-time performance.

5. Conclusions

Music Understanding is a critical part of Music Information Retrieval research as well as a central topic of Computer Music and Music Perception. The similarities between score following and style classification to problems in MIR are striking. I hope that this paper will introduce some pioneering work in Music Understanding to a broader audience including especially MIR researchers.

REFERENCES

[1] Allen, P. and Dannenberg, R.B., Tracking Musical Beats in Real Time. in *1990 International Computer Music Conference*, (1990), International Computer Music Association, 140-143. http://www.cs.cmu.edu/~rbd/bib-beattrack.html#icmc90 (note that an extended version of the paper is available online).

[2] Baird, B., Blevins, D. and Zahler, N. Artificial Intelligence and Music: Implementing an Interactive Computer Performer. *Computer Music Journal*, *17* (2). 73-79.

[3] Bloch, J. and Dannenberg, R.B., Real-Time Accompaniment of Polyphonic Keyboard Performance. in *Proceedings of the 1985 International Computer Music Conference*, (1985), International Computer Music Association, 279-290. http://www.cs.cmu.edu/~rbd/bib-accomp.html#icmc85.

[4] Capell, P. and Dannenberg, R.B. Instructional Design and Intelligent Tutoring. *Journal of Artificial Intelligence in Education*, *4* (1). 95-121.

[5] Coda. SmartMusic Studio 6.0, Coda Music Technology, Inc., Eden Prarie, MN, 2000. http://www.codamusic.com/coda/sm.asp.

[6] Dannenberg, R., Sanchez, M., Joseph, A., Saul, R., Joseph, R. and Capell, P., An Expert System for Teaching Piano to Novices. in *1990 International Computer Music Conference*, (1990), International Computer Music Association, 20-23. http://www.cs.cmu.edu/~rbd/bib-ptutor.html#icmc90.

[7] Dannenberg, R.B. Computer Accompaniment (oral presentation), STEIM Symposium on Interactive Music, Amsterdam, 1985.

[8] Dannenberg, R.B. Method and Apparatus for Providing Coordinated Accompaniment for a Performance, US Patent #4745836, 1988.

[9] Dannenberg, R.B. and Bookstein, K., Practical Aspects of a Midi Conducting Program. in *Proceedings of the 1991 International Computer Music Conference*, (1991), International Computer Music Association, 537-540. http://www.cs.cmu.edu/~rbd/subjbib2.html#icmc91.

[10] Dannenberg, R.B. and Grubb, L.V. Automated Musical Accompaniment With Multiple Input Sensors, US Patent #5521324, 1994.

[11] Dannenberg, R.B. and Mont-Reynaud, B., Following an Improvisation in Real Time. in *Proceedings of the International Computer Music Conference*, (1987), International Computer Music Association, 241-248. http://www.cs.cmu.edu/~rbd/bib-accomp.html#icmc87.

[12] Dannenberg, R.B. and Mukaino, H., New Techniques for Enhanced Quality of Computer Accompaniment. in *Proceedings of the International Computer Music Conference*, (1988), International Computer Music Association, 243-249. http://www.cs.cmu.edu/~rbd/bib-accomp.html#icmc88.

[13] Dannenberg, R.B., Sanchez, M., Joseph, A., Capell, P., Joseph, R. and Saul, R. A Computer-Based Multimedia Tutor for Beginning Piano Students. *Interface - Journal of New Music Research*, *19* (2-3). 155-173.

[14] Dannenberg, R.B., Thom, B. and Watson, D., A Machine Learning Approach to Style Recognition. in *1997 International Computer Music Conference*, (1997), International Computer Music Association. http://www.cs.cmu.edu/~rbd/bib-styleclass.html#icmc97.

[15] Grubb, L. and Dannenberg, R.B., Automated Accompaniment of Musical Ensembles. in *Proceedings of the Twelfth National Conference on Artificial Intelligence*, (1994), AAAI, 94-99.

[16] Grubb, L. and Dannenberg, R.B., Automating Ensemble Performance. in *Proceedings of the 1994 International Computer Music Conference*, (1994), International Computer Music Association, 63-69. http://www.cs.cmu.edu/~rbd/bib-accomp.html#icmc94.

[17] Grubb, L. and Dannenberg, R.B., Enhanced Vocal Performance Tracking Using Multiple Information Sources. in *Proceedings of hte International Computer Music Conference*, (1998), International Computer Music Association, 37-44. http://www.cs.cmu.edu/~rbd/bib-accomp.html#icmc88.

[18] Grubb, L. and Dannenberg, R.B., A Stochastic Method of Tracking a Vocal Performer. in *1997 International Computer Music Conference*, (1997), International Computer Music Association. http://www.cs.cmu.edu/~rbd/bib-accomp.html#icmc97.

[19] Grubb, L.V. *A Probabilistic Method for Tracking a Vocalist*. Carnegie Mellon University, Pittsburgh, PA, 1998. http://reports-archive.adm.cs.cmu.edu/anon/1998/abstracts/98-166.html.

[20] Grubb, L.V. and Dannenberg, R.B. System and Method for Stochastic Score Following, US Patent #5913259, 1997.

[21] Hoshishiba, T., Horiguchi, S. and Fujinaga, I., Study of Expression and Individuality in Music Performance Using Normative Data Derived from MIDI Recordings of Piano Music. in *International Conference on Music Perception and Cognition*, (1996), 465-470. http://www.jaist.ac.jp/~hoshisi/public/papers/icmpc96.pdf.

[22] ICMA. http://www.computermusic.org/, International Computer Music Association, 2001.

[23] Large, E.W. Dynamic programming for the analysis of serial behaviors. *Behavior Research Methods, Instruments, and Computers*, *25* (2). 238-241.

[24] Sankoff, D. and Kruskal, J.B. *Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison*. Addison-Wesley, Reading, MA, 1983.

[25] SMPC. http://psyc.queensu.ca/~smpc, Society for Music Perception and Cognition, 2001. http://psyc.queensu.ca/~smpc.

[26] Vercoe, B., The Synthetic Performer in the Context of Live Performance. in *Proceedings of the International Computer Music Conference 1984*, (1984), International Computer Music Association, 199-200.