Music Score Alignment and Computer Accompaniment

Roger B. Dannenberg and Christopher Raphael

INTRODUCTION

As with many other kinds of data, the past several decades have witnessed an explosion in the quantity and variety of music in computer-accessible form. There are primarily two kinds of "music data" one encounters today: sampled audio files, such as those found on compact discs or scattered over the web in various formats, and *symbolic* music representations, which essentially list notes with pitch, onset time, and duration for each note. To draw an analogy, music audio is to symbolic music as speech audio is to text. In both cases the audio representations capture the colorful expressive nuances of the *performances*, but are difficult to "understand" by anything other than a human listener. On the other hand, in both text and symbolic music the high level "words" are parsimoniously stored and easily recognized.

We focus here on a form of machine listening known as music score matching, score following, or score alignment. Here we seek a correspondence between a symbolic music representation and an audio performance of the same music, identifying the onset times of all relevant musical "events" in the audio—usually notes. There are two different versions of the problem, usually called "off-line" and "on-line."

Off-line matching uses the complete performance to estimate the correspondence between audio data and symbolic score. Thus, the off-line problem allows one to "look into the future" while establishing the correspondence. An off-line match can be viewed as an *index* into the performance allowing random access to a recording. Such an index might be used to allow a listener to begin at any location in a composition, such as the 2nd quarter of measure 48, to link visual score representations with audio, or to coordinate animation with prerecorded audio.

We believe off-line score matching will soon enable many new and useful applications. For example, digital editing and post processing of music often requires the location of a particular note in an audio file to be tuned, balanced or "tweaked" in various ways. Score matching allows the automatic identification of such locations, greatly simplifying the process. Another example involves musical sound synthesis, which often relies heavily on audio samples under various conditions such as pitch, dynamic level, articulation, etc. Score matching can be used as a means of automating this arduous data collection process. In a different direction, score matching can provide quantitative information about timing and tempo, aiding the study and understanding of musical expression. In fact, score matching may well be the key to studying many other musical attributes, such as dynamics, vibrato, and tone color, by providing note-based audio segmentation. We anticipate that the future's synthesized music will greatly benefit from this inquiry.

Off-line score matching's real-time cousin, sometimes called *score-following*, processes the audio data on-line as the signal is acquired, thus, no "look ahead" is possible. The goal of score following is to identify the musical events depicted in the score with high accuracy *and* low latency. The application dearest to the hearts of both authors is the accompaniment system,

which generates a flexible musical accompaniment that *follows* a live soloist. While there are many aspects to such a system, "hearing" the live player is clearly one of the necessary ingredients in any solution. Other applications include the automatic coordination of audio-visual equipment with musical performance, such as opera supertitles and lighting effects, as well as real-time score-based audio enhancement e.g. pitch correction, and automatic page turners.

OFFLINE SCORE MATCHING

Perhaps the easiest way to explain the basic challenges and ideas of score following is to begin with a simple example. Suppose we have a simple monophonic (one note at a time) instrument playing from a known score containing only a few notes, say ABCD as shown in Figure 1. Every 50ms or so, we estimate a pitch value from the audio signal. Pitch estimates are not very reliable, but there are many estimates per note, resulting in something like: AAAAAAAAaABABAB BBCBBCBcBBBBBdBBBddCDCCCCCcCCCCDDBDdDDcDBDDDDDD, where lower-case letters denote a lower octave than the one notated in the score.

The task is now to segment this sequence into regions that correspond to the note sequence ABCD. One approach, shown in Figure 1, uses a simple state graph, labeled "Score Model" with four possible states: A, B, C and D. At each step, the model can remain in the same state or advance to the next one. Since the input in this example has 62 pitch estimates, consider all possible paths of length 62 through through the graph. Each path will be scored according to how closely it matches the actual data sequence above. For each symbol, there is no penalty for an exact match, a small penalty for an octave or neighboring pitch error, and a large penalty for anything else. Thus every sequence has a total penalty—the sum of all penalties encountered, and one can define an optimal state sequence as the one that globally minimizes this penalty. Readers are likely familiar with the Viterbi Algorithm, which uses dynamic programming to solve this problem without actually enumerating every path. The result is an optimal state sequence—our score match—such as AAAAAAAAAAAABBBBBBBBBBBBBBBBBBBBCC CCCCCCCCCCCCDDDDDDDDDDDDDDDDD.

While this example is, admittedly, a toy one, the basic ideas can be extended and generalized to cover most of what is encountered in a broad range of music. For instance, polyphonic music can be represented as a sequence of *chords*, where a new chord appears at every point where a note begins or ends. Furthermore, different note lengths can be represented by allocating several graph states for each note and including transition *probabilities* to generate a reasonable length *distribution* for each note.

A direct extension of the state graph idea to *polyphonic* music might replace the pitch estimates with *sets* of pitch estimates. The note sequence of the graph would be replaced by a chord sequence. Unfortunately, multi-pitch estimation is unreliable and still an active research area, so today's score-alignment systems use representations based on the distributions of frequencies in the signal, also known as the spectrum. The spectrum is used because it is easy to obtain from audio and because it can be reasonably predicted from the score.

To predict spectra from chords, we can use some basic knowledge of musical acoustics. Each note in the chord produces a spectrum that can be approximated by a set of harmonics. A collection of pitches is modeled as a superposition of these one-note models. One can then base a penalty function on the distance between predicted and observed spectra, or on a probability model for the data spectrum, given the hypothesized notes of the chord. Another way to predict

spectra is to play the score through a music synthesizer and compute spectra from the resulting audio. This also opens the possibility of audio-to-audio alignment of two different performances.
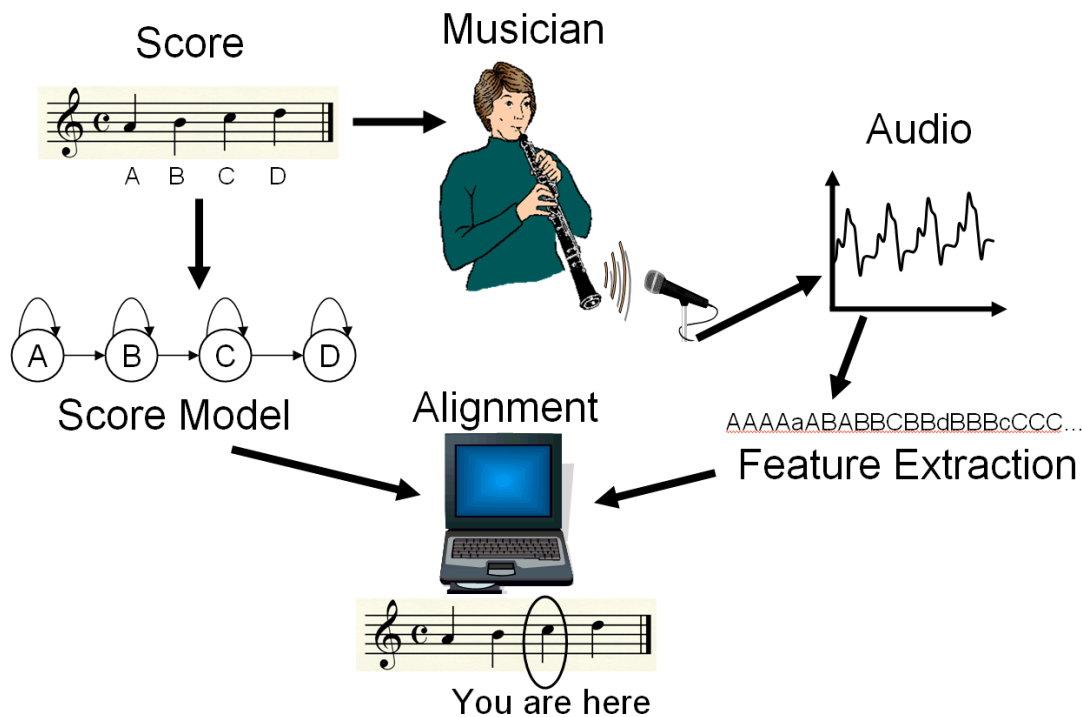
Figure 1. **Audio is analyzed to obtain a transcription of pitch estimates. These correspond to a sequence of states from the score model, which is derived from the symbolic score. An alignment process relates the audio time position to the music score position.**

A popular technique for comparing spectra from scores to spectra in music audio is to reduce the detailed spectrum to a 12-element *chroma vector*, where each element represents all the energy in the spectrum associated with one of the 12 pitch classes (C, C-sharp, D, etc.). The Euclidean distance between chroma vectors is a robust distance metric for score alignment. [6]

Armed with these essential ideas, we have made score matchers that handle realistic audio data, such as that of Figure 2, in which we have indicated the recognized measure locations with vertical lines. The recognition of this audio was achieved by augmenting the notion of "state" discussed above to include both current note and time-varying *tempo*, leading to a simultaneous estimate of both note onset times and the tempo process. [11] The audio file can be heard at http://xavier.informatics.indiana.edu/~craphael/acm with clicks added to mark the measures.
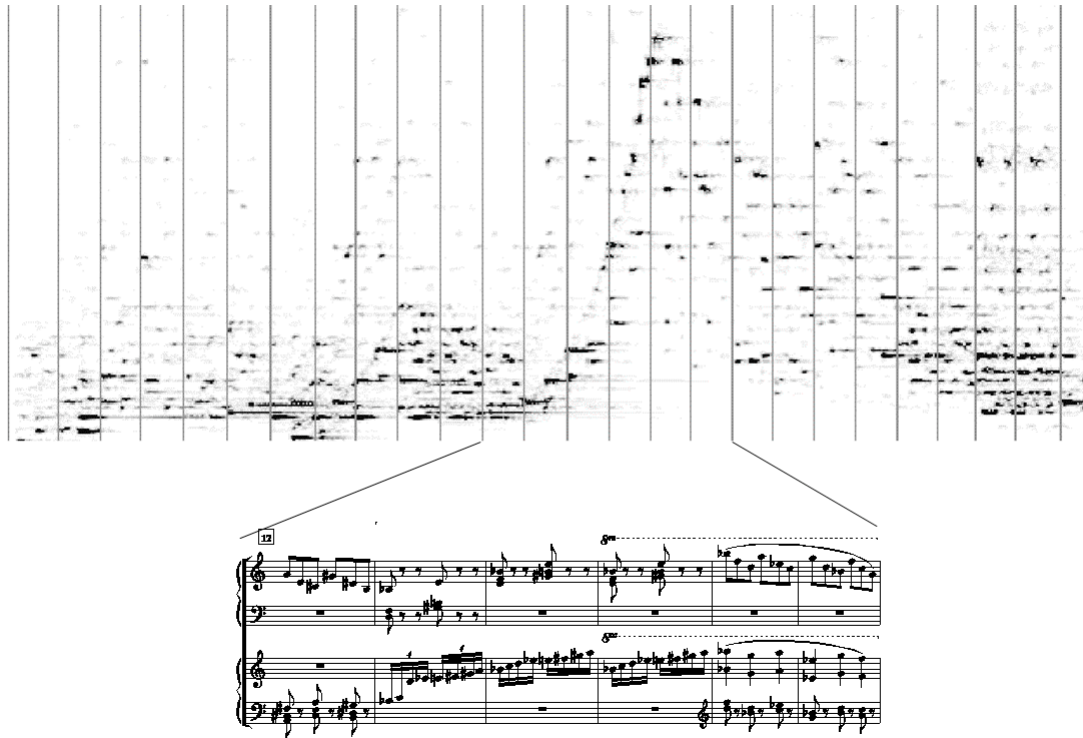
Figure 2. Spectrogram from the opening of *Mercury* from The Planets, by Gustav Holst. The vertical lines indicate the positions of the bar lines in the music as determined by automatic score alignment. Some of the music is shown at the bottom.

On-Line Matching and Computer Accompaniment

The biggest success of on-line score following so far has been computer accompaniment. There are several areas in which we see a promising future for these accompaniment systems. The first is the traditional soloist-accompaniment scenario in which a live player wishes to play with a flexible, sensitive, (and tireless!) accompaniment. While human accompanists provide stiff competition in this traditional domain, there are areas in which accompaniment systems *beat* their human counterparts. The accompaniment system has nearly unlimited technical facility, allowing the coordination of nearly arbitrarily fast notes and complex rhythms. This virtuosity has been exploited in recent music explicitly composed for accompaniment systems by Jan Beran and Nick Collins (http://xavier.informatics.indiana.edu/~craphael/music_plus_one/), and we hope that accompaniment systems will provide an outlet for other composers as well. Perhaps accompaniment systems will someday find a place in jazz and other improvisatory music.

Most computer accompaniment systems are organized roughly along the lines of Figure 3, which shows four fairly independent components: audio feature extraction, score following, accompaniment control, and music synthesis. Note that the input includes machine-readable scores for both the performer and for the accompaniment, while the output is a real-time musical accompaniment. The feature extraction component performs signal processing on the incoming audio to generate a stream of data such as pitch estimates or spectral frames. The score following component matches this sequence to the score to derive an evolving estimate of the current position of the soloist. This is the real-time version of the score alignment illustrated in 0—the

4

score follower provides a running commentary on the soloist's performance ("you are here"), giving estimates of solo event times, delivered with variable latency. The accompaniment control, is the brain of the accompaniment system. Given the commentary provided by the score following, this component integrates the information in the score, musical constraints, and perhaps knowledge distilled from past performances, to determine when to play accompaniment notes. The fourth, music synthesis, generates the actual accompaniment sound, either by means of a note-based sound synthesizer or by resynthesizing prerecorded audio on-the-fly.
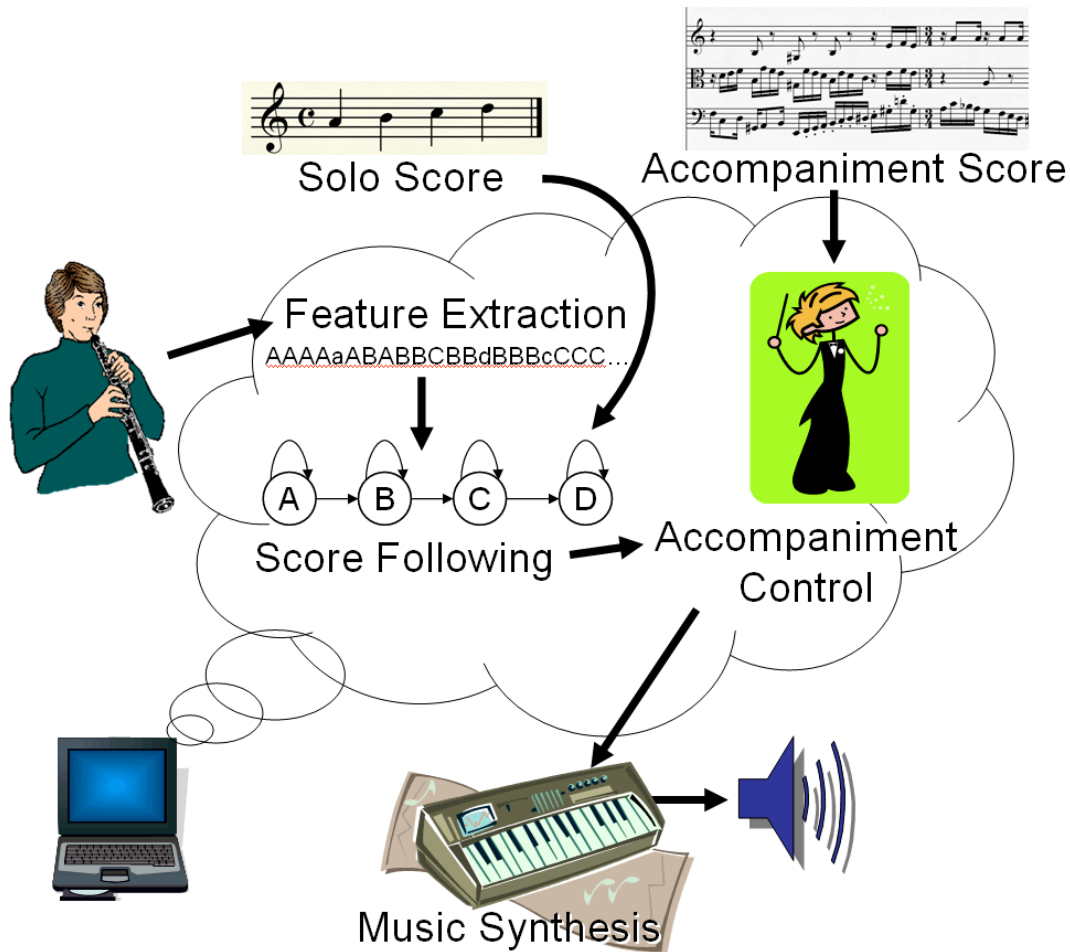
Figure 3.   Structure of a computer accompaniment system.

The importance of the "Accompaniment Control" is paramount and requires some explanation. An accompaniment system, (or person), that tries to achieve synchrony by *waiting* to hear a note, and then *responding*, will always be late, since all musical events are detected with latency. A better approach coordinates parts by *extrapolating* the future evolution based on the detected note onsets, as well as other information. The essential musical extrapolation challenge requires the fusion of several knowledge sources including the musical score, the output of the score follower, as well as past training examples from the soloist and human-played accompaniment. Early accompaniment systems used hand-coded rules to guide this extrapolation. [1, 3] More recently, belief networks address the challenges of automatic learning and extrapolation from incomplete information. [10]

Variations and Alternate Approaches

Many have explored different aspects of the problems and solutions outlined above. The earliest computer accompaniment systems matched note sequences rather than spectral data, reducing data rates and computation. [1, 3, 12] Hidden Markov models have been explored for score following, allowing systems to be trained using rehearsals to follow more accurately and to respond with less latency. [2, 7, 9] Another approach avoids the problem of discrete states by representing score position as a continuous probability density function. [5] (See http://www.cs.cmu.edu/~music/video/CANGIO.MOV for a demonstration.)

Whereas computer accompaniment emphasizes responsiveness and musical accompaniment, the main goal of off-line score matching is accurate alignment, which can be obtained using short audio frames and specialized features for matching. [8]

Current Applications of Score Following

Much research in computer accompaniment has been motivated by contemporary art music. Score following and computer accompaniment have enabled composers to exploit virtuosic performance capabilities of computers and to apply live digital audio signal processing to the sounds of live performers. Computer accompaniment is also used in music education systems, including the Piano Tutor and SmartMusic.

The Piano Tutor

Score following has formed the basis for an intelligent piano tutoring system. The Piano Tutor system [4] was designed to model the interactions that students have with human teachers. The system has a repertoire of about 100 songs that it can select for students. As the student performs the music, real-time score following is used to track the performance and automatically turn pages by updating the display. After the performance, score alignment is used to identify all skipped notes and extra notes. The student's tempo is estimated, and timing errors are identified. Based on this analysis, an expert system gives feedback. For example, a voice may say "watch out, you missed a note here" while a box is drawn around the problem area in the music notation. Score following allows the Piano Tutor to evaluate relatively unconstrained piano performances, resulting in a simpler interface and (we think) a more enjoyable experience for students. (See http://www.cs.cmu.edu/~music/video/pianotutor.mov.)

SmartMusic

SmartMusic (http://smartmusic.com) is a commercial software product for ordinary personal computers. It uses computer accompaniment as a tool for music education and includes accompaniments for over 30,000 pieces of music for wind instruments, vocalists, and beginning string players. Traditionally, music students practice alone and often lack the skills and motivation to practice effectively. SmartMusic challenges students to master pieces from beginning to end. When practicing with accompaniment, students hear their music in the context of rhythm and harmony that is missing from solo parts. The accompaniment also provides a pitch reference so that students are better able to hear when they are out of tune and learn to take corrective action. Only small-scale studies have been undertaken, but they indicate that students who practice with accompaniment improve in performance skill and confidence, motivation, and practice time.

Music Plus One

Music Plus One is a system primarily oriented toward Western art music. In this system, the accompaniment is synthesized from prerecorded audio using a phase vocoder, thus allowing synchronization with the soloist while retaining the rich sonic palette and much musicality from the original performance. This work performs score-following with an HMM and coordinates parts with a belief network (the brain). This latter approach predicts future accompaniment note onsets based on the score-follower, the musical score, and examples of past solo and human-played accompaniment performances. These predictions create a trail of breadcrumbs for the phase vocoder to link. See http://xavier.informatics.indiana.edu/~craphael/music_plus_one for an example.

The Future of Score Matching

The proficiency of on-line and off-line score matching algorithms has improved to a point where many new applications are possible in areas such as editing of digital audio, audio database construction, real-time performance enhancement, accompaniment systems, as well as others previously mentioned or not yet thought of. While music is varied enough to violate almost any assumptions that one might make, these algorithms already perform reliably in some well-established and challenging music domains. We look forward to the many new possibilities that are certain to result from this new technology.

Both authors are especially optimistic about the future of accompaniment systems. While many musicians already use computer accompaniment to make practice more enjoyable and instructive, broader acceptance of this technology requires that we overcome significant challenges. Paramount among these is capturing a deeper understanding of musical aesthetics by the computer. This challenge is especially intriguing since it requires a fusion of expertise from areas that normally do not communicate much.

Computer accompaniment is bound to find commercial applications that will touch the lives of musicians and music lovers. While some express concern that musicians will be *displaced* by these machines, we doubt that this will be the predominant effect. On the contrary, we believe that accompaniment systems will make music more accessible to more people, fostering a wider appreciation and love for music-making. We expect that the effect will be an increased demand for "all-human" music, both from listeners and participants.

References

1. Baird, B. Blevins, D., and Zahler, N. Artificial Intelligence and Music: Implementing an Interactive Computer Performer. *Computer Music Journal 17*, 2, 1993, 73–79.

2. Schwarz, D., Cont, A., and Schnell, N., From Boulez to Ballads: Training IRCAM's Score Follower, *Proceedings of the 2005 International Computer Music Conference*, (Barcelona). International Computer Music Association, San Francisco, 2005.

3. Dannenberg, R. B. Real-Time Scheduling and Computer Accompaniment. In Mathews, M. and Pierce, J. eds. *Current Research in Computer Music*, MIT Press, Cambridge, 1989, 225-261.

4. 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, 1993, 155-173.

5. Grubb, L. and Dannenberg, R. B. A Stochastic Method of Tracking a Vocal Performer. In *1997 International Computer Music Conference*, (Thessaloniki). International Computer Music Association, San Francisco, 1997, 301-308.

6. Hu, N., Dannenberg, R. B. and Tzanetakis, G. Polyphonic Audio Matching and Alignment for Music Retrieval. In *2003 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, (New Palz, NY). IEEE, New York, 2003, 185-188.

7. Loscos, A., Cano, P., and Bonada, J. Score-Performance Matching using HMMs. In *Proceedings of the 1999 International Computer Music Conference*, (Beijing). International Computer Music Association, San Francisco, 1999, 441–444.

8. Orio, N., and Schwarz, D. Alignment of Monophonic and Polypophonic Music to a Score. In *Proceedings of the 2001 International Computer Music Conference*, (Havana). International Computer Music Association, San Francisco, 2001, 155-158.

9. Raphael, C. Automatic Segmentation of Acoustic Musical Signals Using Hidden Markov Models. *IEEE Transactions on PAMI 21,* 4, 1999, 360-370.

10. Raphael, C., A Bayesian Network for Real-Time Musical Accompaniment. In *Neural Information Processing Systems (NIPS) 14*, 2001.

11. Raphael, C., A Hybrid Graphical Model for Aligning Polyphonic Audio with Musical Scores. In *Proceedings of the 5th International Conference on Music Information Retrieval,* (Barcelona, Spain), Universitat Pompeu Fabra, 2004, 387-394.

12. Vercoe, B. The Synthetic Performer in the Context of Live Performance. In Proceedings of the International Computer Music Conference 1984, (Paris). International Computer Music Association, San Francisco, 1985, 199–200.

Roger B. Dannenberg (roger.dannenberg@cs.cmu.edu) is Associate Research Professor of Computer Science and Art at Carnegie Mellon University in Pittsburgh, Pennsylvania.

Christopher Raphael (craphael@indiana.edu) is an Associate Professor in the School of Informatics at Indiana University, Bloomington and an Adjuct Professor in the School of Music.