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. Some early efforts in score matching made use of a hardware device known as a "pitch-to-midi converter" that parses input audio into notes and issues a MIDI message at each note onset. Suppose a performer is playing the sequence of pitches ABCD, and the pitch-to-midi converter outputs the sequence ABBCD. The challenge is to find a correspondence between these two strings. This problem is solved by the *longest common subsequence* (LCS) algorithm, which finds a sequence of pitches that is common to both strings.

String matching works well when notes can be detected reliably, but in practice, pitch estimation is unreliable. Spurious note onsets and near misses are common. In many cases, a better approach is to estimate a pitch value every 50ms or so, obtaining something like: AAAAAAAAaABABABBBCBBCBcBBBBBdBBBddCDCCCCCcCCCCDDBDdDDcDBDDDDDD, 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. As shown in , we use a simple state graph, labeled "Score Model" with four possible states: A, B, C and D. For every new symbol we examine from the performance, the score matcher can remain at the present state or move to the next state in the sequence, if there is one. Any possible sequence of states 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. The result is an optimal state sequence—our score match—such as

AAAAAAAAAAAABBBBBBBBBBBBBBBBBBBBCCCCCCCCCCCCCCDD
DDDDDDDDDDDDDD

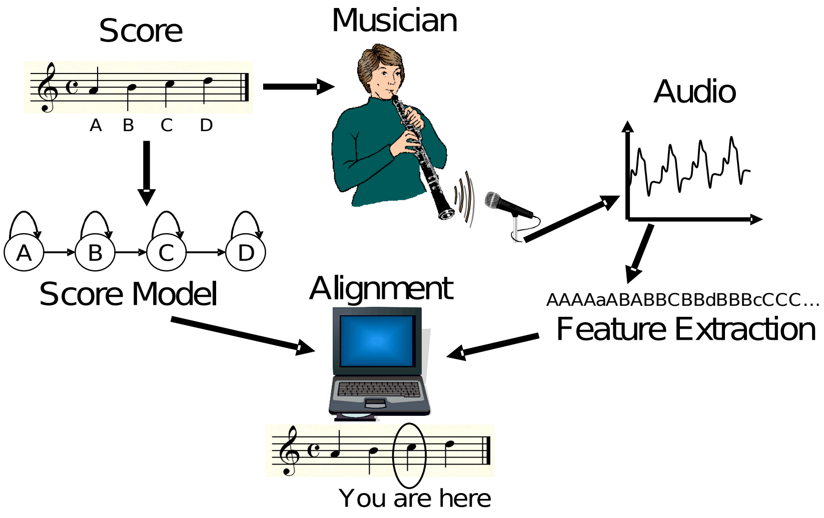

Figure 1.Audio is analyzed to obtain a transcription of pitch estimates. These correspond to a sequence of states from the score model, which is derived from the symbolic score. An alignment process relates the audio time position to the music score position.

While this example is, admittedly, a toy one, the basic ideas are considerably more extensible than they might appear. First of all, the simple “forward-directed” state graph above can be 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*, rather than single notes, in which a new chord appears every time any voice enters, exits, or changes pitch. 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.

Direct extension of the state graph idea to *polyphonic* music would require a multi-pitch estimator in place of the single-pitch estimator—a tall order in a realistic musical scenario. Instead, the pitch sequence in can be replaced with any features that can be derived from both the score and the audio. For polyphonic music, spectral features can be used. One approach generates audio from the hypothesized pitches using a synthesizer, while taking the "distance" between the synthesized and observed spectra as the similarity metric. A more sophisticated method models the observed spectrum through first principles: Music is composed largely of nearly periodic sounds clustering their energy at regularly spaced harmonics. A collection of pitches is therefore modeled as a superposition of these one-note models. One can then base a similarity measure on the distance between templates and observed spectra, or on a probability model for the data spectrum, given the hypothesized notes. A related approach projects each frequency bin of the spectrum onto one of 12 chromatic pitch classes, resulting in a 12-element "harmonic summary" called the *chroma vector*. The Euclidean distance between chroma vectors is a robust distance metric for score alignment. []

Armed with these essential ideas, we have made score matchers that handle realistic audio data, such as that of , in which we have indicated the recognized onset locations with vertical lines. Figure 2 contains a *spectrogram* of an audio recording. Here, the horizontal axis indicates time and the vertical axis represents frequency. Darkness at any particular point indicates energy at that time and frequency. The recognition of this audio was acheived by augmenting the notion of "state" discussed above to include both curent note and time-varying *tempo*, leading to a simultaneous estimate of both note onset times and the tempo process. [] The audio file can be heard at http://xavier.informatics.indiana.edu/~craphael/acmwith clicks added to mark the notes.

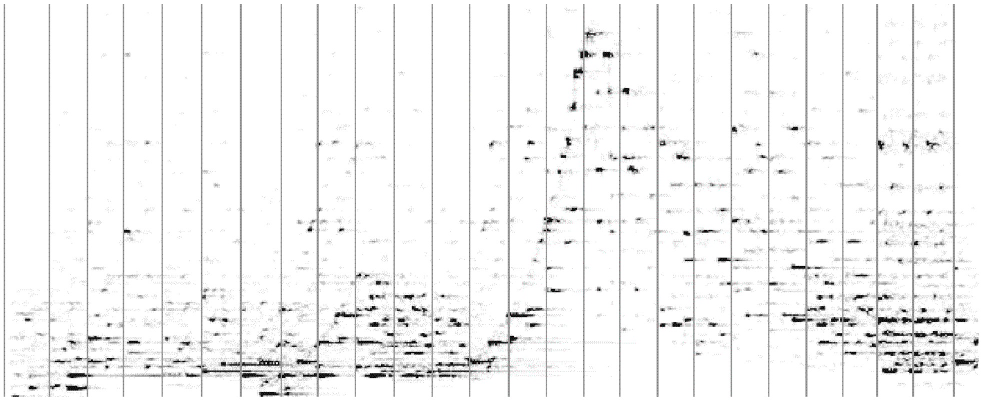

Figure 2.Spectrogram from the opening of *Mercury* from The Planets, by Gustav Holst. The vertical lines indicate the positions of the bar lines in the music shown in , as determined by automatic score alignment. In the accompanying audio file, clicks are added on the downbeats that contain note onsets.

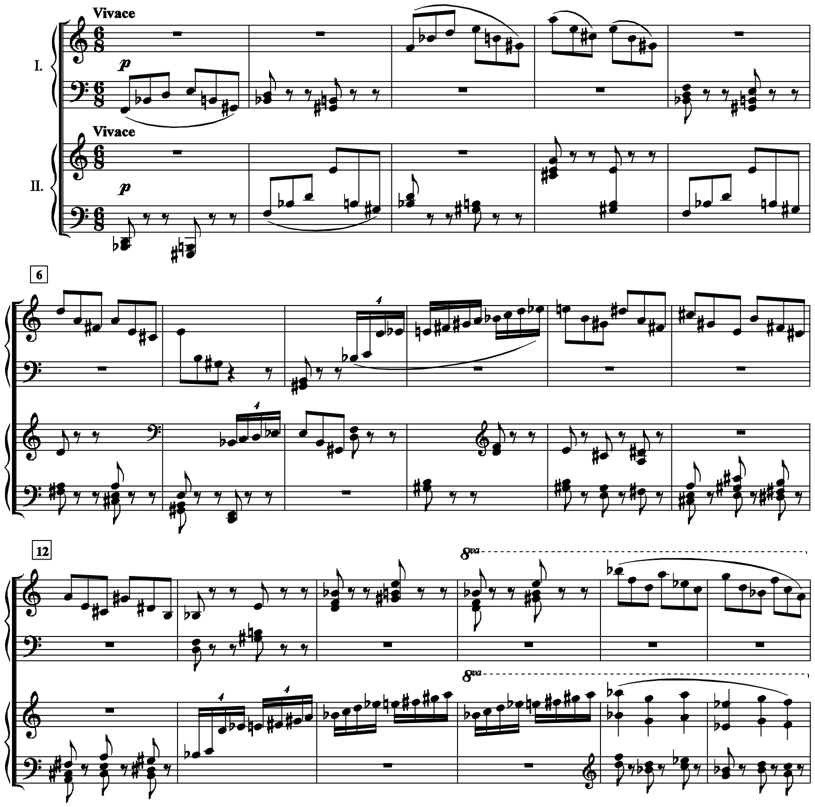

Figure 3.Music from the opening of *Mercury* from The Planets.

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 , 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 score following component "listens" to the processed audio input and derives an evolving estimate of the current position the soloist occupies in the score. This is the real-time version of the score alignment illustrated in . The 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 final component, music synthesis, generates the actual accompaniment sound, either by means of a note-based sound synthesizer or by resynthesizing prerecorded audio on-the-fly.

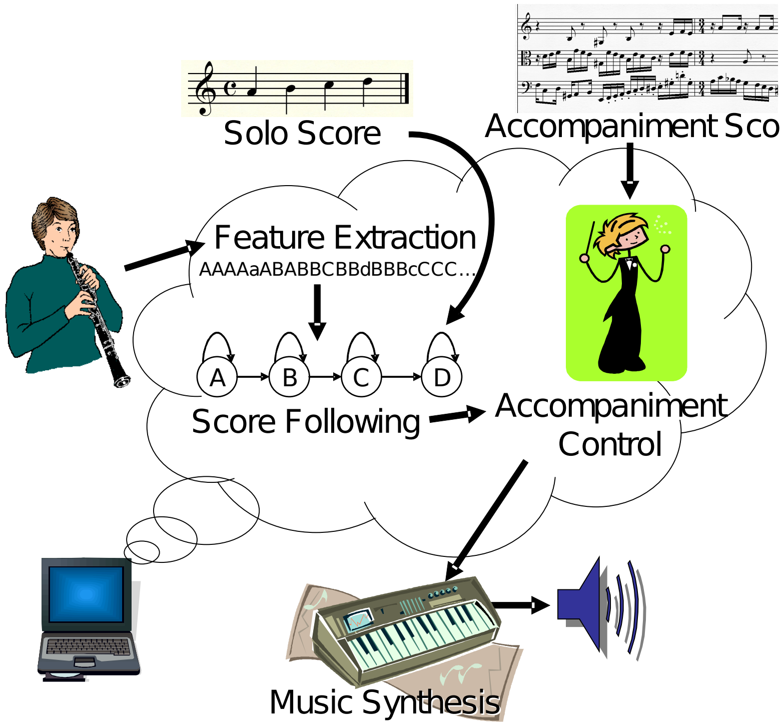

Figure 4.Structure of a computer accompaniment system.

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. [] More recently, belief networks address the challenges of automatic learning and extrapolation from incomplete information. This latter approach is based on a probabilistic model for musical interpretation driven by a hidden time-varying tempo process and sequence of note-by-note timing deviations. The estimated note onset times from the score follower, as well as the currently played accompaniment notes, are the observed variables in the model. The belief network then schedules and reschedules the pending accompaniment note every time new information is observed (e.g. a note detection). Thus when an accompaniment note is finally played it reflects the most current state of knowledge. Furthermore, the model is trained using past performances to better capture and predict the soloist's rhythmic interpretation. []

Probabilistic Score Following and Machine Learning

The string matching algorithm and its variants used for score following require a reasonable note-based segmentation or labeling of the audio data. Not surprisingly, such algorithms degrade significantly in more challenging domains including music with fast notes, vocal music, extremes of register, significant background “noise” from accompanying instruments, polyphony, etc.

One solution is to use probabilistic techniques. In the vocal score follower developed by Lorin Grubb for his PhD thesis, the current knowledge of a vocal soloist’s score position is modeled as a probability density function. [] How can this function be computed? First of all, the function is represented by discrete samples and recomputed periodically, roughly every 60ms. Assuming that there is an estimate of the tempo, we can compute how far the performer will move through the score in 60ms and shift the probability function accordingly. In practice, the tempo estimate is also represented by a probability distribution and the shift is performed by numerical convolution. This has the effect of smearing out the probability distribution as time passes. At the same time, sensors for pitch estimation, note onset detection, and sung vowel detection provide additional information to sharpen the probability distribution around the most likely score position. One attraction of this approach is that it can model tempo and position as continuous variables.

Another probabilistic alternative is to use a hidden Markov model for score following. One strength of HMMs in this domain comes from their automatic trainability. HMM-based score followers can learn to recognize more accurately and respond with less latency by training on past rehearsals. [] This allows the HMM to model the variability in musical timing exhibited by the live player, as well as tuning the output model to a new acoustic environments, such as instrument, room, microphone placement, etc. Both the HMM and the probability density function formulation make it possible to quantify the *confidence* we have in various note onset detections. Thus we only communicate information to the prediction "brain" when we are reasonably certain of its correctness. Examples of an HMM-based score follower in action in an accompaniment system can be heard in a variety of situations at http://xavier.informatics.indiana.edu/~craphael/music_plus_one. A demonstration of Grubb's vocal accompaniment system can be found at http://www.cs.cmu.edu/~music/video/CANGIO.MOV.

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 [] was designed to model the interactions that students have with human teachers. Typically, teachers select music for the student, help the student master the material, and decide when to move on to the next piece. The Piano Tutor 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 determines one problem on which to focus. Feedback is given; 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. With SmartMusic, students are challenged to master pieces from beginning to end. Students can set up practice loops to drill tough passages, and 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.

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 the necessity of the accompanist to create aesthetically satisfying performances in addition to coordinating with the soloist. While a certain amount of this musicality can be piggy-backed by simply following the soloist, many situations require a deeper understanding from the accompanist. This challenge is particularly intriguing since it requires a fusion of expertise from areas that normally do not communicate much.

Popular music performance often involves improvisation and an open structure where sections may be repeated or skipped. Playing this music requires one to listen to chord progressions, rhythmic patterns, and other cues. In the future, perhaps an amateur “garage band” will be able to employ a computer system to play a missing part such as rhythm guitar. This will require advances in automatic music understanding if the computer is to become a truly interesting musical partner.

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. 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.
2. 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.
3. 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.
4. 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.

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

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

7. 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.