Score Following: State of the Art and New Developments

Nicola Orio
University of Padova
Dept. of Information Engineering
Via Gradenigo, 6/B
35131 Padova, Italy
orio@dei.unipd.it

Serge Lemouton
Ircam - Centre Pompidou
Production
1, pl. Igor Stravinsky
75004 Paris, France
lemouton@ircam.fr

Diemo Schwarz
Ircam - Centre Pompidou
Applications Temps Réel
1, pl. Igor Stravinsky
75004 Paris, France
schwarz@ircam.fr

ABSTRACT

Score following is the synchronisation of a computer with a performer playing a known musical score. It now has a history of about twenty years as a research and musical topic, and is an ongoing project at Ircam. We present an overview of existing and historical score following systems, followed by fundamental definitions and terminology, and considerations about score formats, evaluation of score followers, and training. The score follower that we developed at Ircam is based on a Hidden Markov Model and on the modeling of the expected signal received from the performer. The model has been implemented in an audio and a Midi version, and is now being used in production. We report here our first experiences and our first steps towards a complete evaluation of system performances. Finally, we indicate directions how score following can go beyond the artistic applications known today.

Keywords

Score following, score recognition, real time audio alignment, virtual accompaniment

1. INTRODUCTION

In order to transform the interaction between a computer and a musician into a more interesting experience, the research subject of virtual musicians has been studied for almost 20 years now. The goal is to simulate the behaviour of a musician playing with another, a “synthetic performer”, to create a virtual accompanist that will follow the score of the human musician. Score following is often addressed as “real-time automatic accompaniment”. This problematic is well defined in [5, 25, 26], where we can find the first use of the term “score following”. Since the first formulation of the problem, several solutions have been proposed [1, 2, 4, 6, 7, 8, 9, 10, 14, 16, 17, 18, 20, 22, 23], some academic, others in commercial applications.

Many pieces have been composed relying on score following techniques. For instance, at Ircam we can count at least 15 pieces between 1987 and 1997, such as *Sonus ex Machina* and *En echo* by Philippe Manoury, and *Anthèmes II* and *Explosante-fixe* by Pierre Boulez.

Nevertheless, there are still some limitations in the use of these systems. There are a number of peculiar difficulties inherent in score following, which, after years of research, are well identified. The two most important difficulties are related to possible sources of mismatch between the human and the synthetic performer: On the one hand, musicians can make errors, i.e. playing something differing from the score, because the musical live interpretation of a piece of music means also a certain level of unpredictability. On the other hand, all real-time analysis of musical signals, and in particular pitch detection algorithms, are prone to error.

Existing systems are not general in the sense that it is not possible to track all kinds of musical instruments; moreover, the problem of polyphony is not completely resolved. Although it is possible to follow instruments with low polyphony, such as the violin [15], highly polyphonic instruments or even a group of instruments are still problematic.

Often, only the pitch parameter is taken into account, whereas it is possible to follow other musical parameters (amplitude, gesture, timbre, etc). The user interfaces of these systems are not friendly enough to allow an inexperienced composer to use them. Finally, the follower is not always robust enough; in some particular musical configurations, the score follower fails, which means that it always needs a constant supervision by a human operator during the performance of the piece. The question of reliability is crucial now that all these “interactive pieces” are getting increasingly common in the concert repertoire. The ultimate goal is that a piece that relies on score following can be performed anywhere in the world, based on a printed score for the musicians, and a CD with the algorithms for the automatic performer, for instance in the form of patches and objects for a graphical environment like *j*Max or *Max/MSP*. At the moment, the composer or an assistant who knows the piece and the follower's favourite errors very well must be present to prevent musical catastrophes.

Therefore, robust score following is still an open problem in the computer music field. We propose a new formalisation of this research subject in section 2, allowing simple classification and evaluation of the algorithms currently used.

At Ircam, the research on score following was initiated by Barry Vercoe and Lawrence Beauregard as soon as 1983. It was continued by Miller Puckette and Philippe Manoury [16, 17, 18]. Since 1999, the Real Time Systems Team, now *Real Time Applications* or *Applications Temps-Réel* (ATR)[1], continues work on score following as their priority project. This team has just released a system running in *j*Max based on a statistical model [14, 15], described in section 3. General considerations how score following systems can be evaluated, and results of tests with our system are presented in section 4.

[1] `http://www.ircam.fr/equipes/temps-reel/`

2. FUNDAMENTALS

As we try to mimic the behaviour of a musician, we need a better understanding of the special communication involved between musicians when they are playing together, in particular in concert. This communication requires a highly expert competence, explaining the difficulty of building good synthetic performers.

The question is: How does an accompanist perform his task? When he plays with one or more musicians, synchronizing himself with the others, what are the strategies involved in finding a common tempo, readjusting it constantly? It is not simply a matter of following, but anticipation plays an important role as well. At the state of the art, almost all existing algorithms are only score *followers* strictly speaking. The choice of developing simply "reactive" score followers may be driven by the fact that a reactive system is more easily controllable by musicians, and it reduces the probability of wrong decisions by the synthetic performer.

What are the cues exchanged by the musicians playing together during a performance? They are not only "listening" to each other, but also looking at each other, exchanging very subtle cues: For example, a simple very small movement of the first violin's left hand, or an almost inaudible inspiration of the accompanying pianist are cues strong enough to start a note with perfect synchronisation. There is a real interaction between musicians, a feedback loop, not just unidirectional communication. A conductor is not only simply giving indications to the orchestra, he also pays constant attention to what happens within ("Are they ready?", "Have they received my last cue?") It seems obvious that considering only the Midi sensors of the musician or the audio signal is a severe limitation of the musical experience.

All these considerations regarding the performer behaviour lead towards a multi-modal model, where several cues of different nature (pitch, dynamic, timbre, sensor and also visual information) can be used simultaneously by the computer to find the exact time in the score.

2.1 Terminology

We propose a new formalisation, and a systematic terminology of score following in order to be able to classify and compare the various systems proposed up to now.

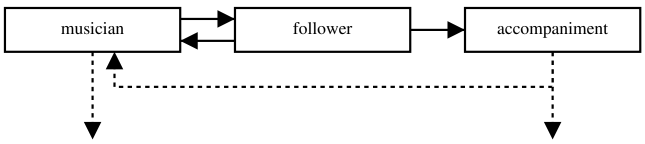

Figure 1: Elements of a score following system. Dashed arrows represent sound.

In any score following system, we find at least the elements shown in figure 1: the (human) *musician*, the *follower* (computer), and the *accompaniment* (also called the automatic performance or electronic part). These elements interact with each other. The role of the communication flow from the musician to the computer is clear, because computer behaviour is almost completely based on human performance. On the other hand, the role of auditory feedback from the accompaniment is not negligible; the musician may change the way he plays at least depending on the quality of the score follower synchronisation.

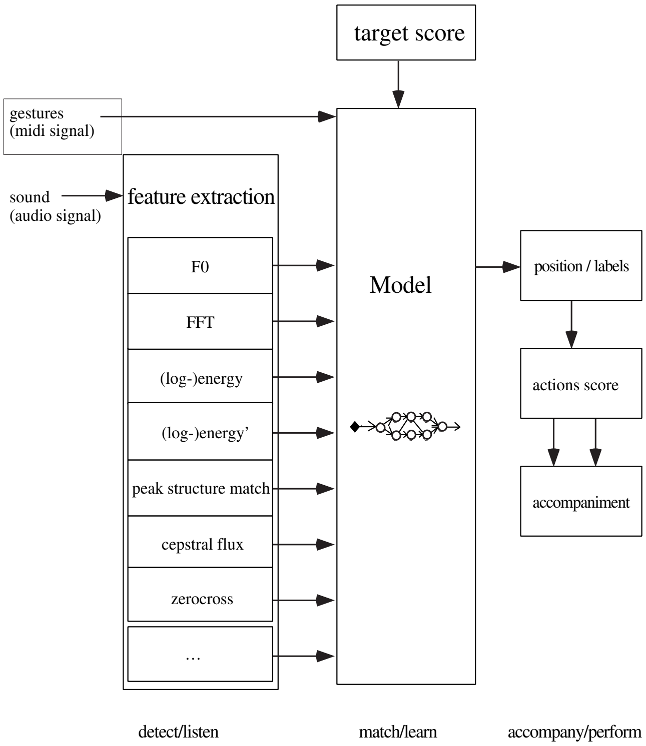

Figure 2: Structure of a score follower.

Figure 2 presents the structure of a general score follower. In a pre-processing step, the system extracts some *features* (e.g. pitch, spectrum, amplitude) from the sound produced by the musician. Each score following system defines a different set of relevant features, which are used as descriptors of the musician's performance. These features define the dimension of the *input space* of the model created from the target score. The *target score* is the score that the system has to follow. Ideally this score is identical to the score that the human musician is playing, even though in most of the existing systems, the score is simply coded as a note list. The question of what kind of score format is used for coding the target score is very important for the ergonomics of the system, and for its performance. We present some possible score formats in section 2.2.

The target score is a sequence of *musical events*, that is a sequence of musical gestures that have to be performed by the musician, possibly with a given timing. These gestures can be very simple, i.e. a rest or a single note, or complex, i.e. vibrato, trills, chords, or glissandi. It is important that each gesture is clearly defined in the common music practice, and its acoustic effect is known.

The *model* is the system's internal representation of this coding of the target score. The model is matched with the incoming data in the follower, while the *actions score* represents the actions that the accompaniment has to perform at some specified positions (e.g. sound synthesis or transformations).

The *position* is the current time of the system relative to the target score. The target score contains also *labels* that can be any symbol, but are usually integer values, giving the cue number of the synthesis or transformation event in the actions score that should be triggered at reception of that label. (That's why often the labels are also called "cues".) The labels can be attached to any of the musical events, for instance to the ones that are particularly relevant in the score, but in general each event can have a label, and also the rests in the score.

According to Vercoe [26], the score follower has to fulfill three tasks: Listen-Perform-Learn. Listening and performing are mandatory tasks for an automatic performer, while learning is a more subtle feature. It can be defined as the ability of taking advantage from previous experiences that, in the case of an accompanist, may regard both previous rehearsals with the same musicians and the knowledge gained in years of public performance. It can be noted that sometimes these two sources of experience may reflect different accompanist's choices during a performance, that are hard to model. Learning can affect different levels of the process: the way the score is modeled, the way features are extracted and used for synchronisation, and the way the performance is modeled and recognized.

There are a number of advantages in using a statistical system for score following, which regard the possibility of training the system and modeling different acoustic features from examples of performances and score. In particular, a statistical approach to score following can take advantage from theory and applications of Hidden Markov Models (HMMs) [19]. A number of score followers have been developed using HMMs, such as the one developed at Ircam [14] and others [12, 21]. In fact, HMMs can deal with the several levels of unpredictability typical of performed music and they can model complex features, without requiring preprocessing techniques that are prone to errors like any pitch detectors or midi sensors. For instance, in our approach, the whole frequency spectrum of the signal is modeled. Finally, techniques have been developed for the training of HMMs.

2.2 Target Score Format

The definition of the imported target score format is essential for the ease of use and acceptance of score following. The constraints are multiple:

- It has to be powerful, flexible, and extensible enough to represent all the things we want to follow.
- There should be an existing parser for importing it, preferably as an open source library.
- Export from popular score editors (Finale, Sibelius) should be easily possible.
- It should be possible to fine-tune imported scores within the score following system, without re-importing them.

The formats that we considered are:

- Graphical score editor formats: *Finale*, *Sibelius*, *NIFF*, *Guido*
- Mark-up languages: *MusicML*, *MuTaTedTwo*, *Wedelmusic XML Format*
- Frameworks: *Common Practice Music Notation* (CPNview), *Allegro*
- Midi

Midi, despite its limitations, is for the moment indeed the only representation to fulfill all these constraints: It can code everything we want to follow, e.g. using conventions for special Midi channels, controllers, or text events. It can be exported from every score editor, and can be fine-tuned in the sequence editor of our score following system. Hence, we stay with Midi for the time being, but the search for a higher-level format that inserts itself well into the composer's and musical assistant's workflow continues.

2.3 Training

One fundamental difference between a computer and a human being is that the latter is learning from experience, whereas a computer program usually does not improve its performance by itself. Since [26], we imagine that a virtual musician should, like a living musician, learn his part and improve his playing during the rehearsals with the other musician. One of the advantages of a score following system based on a statistical model is that it can learn using well-known training methods.

The training can be supervised or unsupervised. Training is unsupervised if it does not need the use of target data, but only several interpretations of the music to be followed.

In order to design a score following system that learns, we can imagine several scenarios:

- When the user inputs the target score, he is teaching the score to the computer.
- During rehearsals, the user can teach the system by a kind of gratification if the system worked properly for a section of the score.
- After each successful performance, so that the system gets increasingly familiar with the musical piece in question.

In the context of our HMM score follower, training means adapting the various probabilities and probability distributions governing the HMM to one or more example performances such as to optimise the quality of the follower. At least two different things can be trained: the transition probabilities between the states of the Markov chain [14], and the probability density functions (PDFs) of the observation likelihoods. While the former is applicable for audio and Midi, but needs much example data, especially with errors, the latter can be done for audio by a statistical analysis of the features to derive the PDFs, which essentially perform a mapping from a feature to a probability of attack or sustain or rest.

Then of course a real iterative training (supervised by providing a reference alignment, or unsupervised starting from the already good alignment to date) of the transition and observation probabilities is being worked on to increase the robustness of the follower even more. This training can adapt to the "style" of a certain singer or musician.

3. IMPLEMENTATION

Ircam's score follower consists of the objects *suiviaudio* and *suivimidi* and several helper objects, bundled in the package *suivi* for *j*Max. The system is based on a two-level Hidden Markov Model, as described in [14]:

States at the *higher level* are used to model the music events written in the score, which may be simple notes (or rests) but also more complex events like chords, trills, and notes with vibrato. The idea is that the first thing to model is the score itself, because it can be considered as the hidden process that underlies the musical performance. By taking into account complex events, e.g. considering a trill as an event by itself rather than a sequence of simple notes, it is possible to generalize the model also to other musical gestures, like for instance glissandi or arpeggios which are not currently implemented.

Together with the sequence of events in the score, which have temporal relationships that are reflected in the left-to-right structure of the HMM, also possible performing errors are modeled. As introduced by [5], there are three possible errors: wrong notes, skipped notes, or inserted notes. The model copes with these errors by introducing error states, or *ghost states*, that model the possibility of playing a wrong event after each event in the score. Ghost states can be used not only to improve the overall performances of the system in terms of score following, but also as a way to refine the automatic performance adding new strategies. For instance, if the system finds that the musician is playing wrong events then it can suspend the automatic performance in order to minimize the effect to the audience, or it can suggest the correct actual expected position in the score depending on composer's choices.

States at the *lower level* are used to model the input features. These states are specialized for modeling different parts of the performance, like the attack, the sustain, and the possible rest, and they are compound together to create states at the higher level. For instance, in an attack state, the follower expects a rise in energy for audio or the start of a note for Midi.

The object *suiviaudio* uses the features *log-energy* and *delta log-energy* to distinguish rests from notes and detect attacks, and the energy in harmonic bands according to the note pitch, and its delta, as described in [15], to match the played notes to the expected notes. The energy in harmonic bands is also called PSM for *peak structure match*. For the singing voice, the *cepstral difference* feature improves the recognition of repeated notes, by detecting the change of the spectral envelope shape when the phonemes change. It is the sum of the square differences of the first 12 cepstral coefficients from one analysis frame to another.

The object *suivimidi* uses a simpler information, that is the onset and the offset of Midi notes. The Midi score follower works even for highly polyphonic scores by defining a note match according to a comparison of the played with the expected notes for each HMM state.

Score following is obtained by on-line alignment of the audio or Midi features to the states in the HMM. A technique alternative to classical Viterbi decoding is employed, as described in [14].

The code that actually builds and calculates the Hidden Markov Model is common to both audio and Midi followers. Only the handling of the input and the calculation of the observation likelihoods for the lower level states are specific to one type of follower.

The system uses the *j*Max *sequence editor* for importing Midi score files, and visualisation of the score and the recognition (followed notes and the position on the time axis are highlighted as they are recognised).

4. EVALUATION

Eventually, to evaluate a score following system, we could apply a kind of Turing test to the synthetic performer, which means that an external observer has to tell if the accompanist is a human or a computer. In the meantime, we can distinguish between subjective vs. objective evaluation:

4.1 Subjective Evaluation

A *subjective* or *qualitative* evaluation of a score follower means that the important performance events are recognised with a latency that respects the intention of the composer, which is therefore dependent on the action that is triggered by this event. Independent of the piece, it can be done by assuming the hardest case, i.e. all notes have to be recognised immediately. The method is to listen to a click that is output at each recognised event and observe the visual feedback of the score follower (the currently recognised note in the sequence editor and its position on the time axis are highlighted), verifying that it is correct. This automatically includes the human perceptual thresholds for detection of synchronous events in the evaluation process.

A limited form of subjective evaluation is definitely needed in the concert situation to give immediate feedback whether the follower follows, and before the concert to catch setup errors.

4.2 Objective Evaluation

An *objective* or *quantitative* evaluation, i.e. to know down to the millisecond when each performance event was recognised, even if overkill for the actual use of score following, is helpful for refinement of the technique and comparison of score following algorithms, quantitative proof of improvements, automatic testing in batch, making statistics on large corpora of test data, and so on.

Objective evaluation needs reference data that provides the correct alignment of the score with the performance. In our case this means a reference track with the labeled events at the points in time where their label should be output by the follower. For a performance given in a Midi-file, the reference is the performance itself. For a performance from an audio file, the reference is the score aligned to the audio. Midified instruments are a good way to obtain the performance/reference pairs because of the perfect synchronicity of the data.

The reference labels are then compared to the cues output by the score follower. The offset is defined as the time lapse between the output of corresponding cues. Cues with their absolute offsets greater than a certain threshold (e.g. 100 ms), or cues that have not been output by the follower, are considered an error. The values characterising the quality of a score follower are then:

- the percentage of non-error labels
- the average offset for non-error labels, which, if different from zero, indicates a systematic latency
- the standard deviation of the offset for non-error labels, which shows the imprecision or spread of the follower
- the average absolute offset of non-error labels, which shows the global precision

There are other aspects of the quality of a score follower not expressed by these values: According to classical measures of automatic systems that simulate the human behavior [3], error labels can be due to the *miss* of a correct label at a given moment, or to the *false alarm* of a label incorrectly given. Based on these two measures it is possible to consider also the number of labels detected more than once, or the zigzagging back to an already detected label.

Again, the tolerable number of mistakes and latencies of the follower largely depend on the kind of application and the type of musical style involved. It can be noted that, for this kind of evaluation, it is assumed that the musician does not make any errors. It is likely that, in a real situation, human errors will occur, suggesting as another measure the time needed by the score follower to recover from an error situation, that is, to resynchronise itself after a number of wrong notes are played. The tolerable number of wrong notes played by the musician is another parameter by itself, that in our system can be experimentally measured through simulations of wrong performances. This aspect is part of the training that can be done directly when creating the model of the score as an injection of a priori knowledge on the HMMs.

4.3 Evaluation Framework

To perform evaluation in our system, we developed the object *suivieval*, which takes as input the events and labels output by the score follower, the note and outputs of the reference performance, and the same control messages as the score follower (to synchronize with its parameters). While running, it outputs abovementioned values from a running statistics to get a quick glance at the development and quality of the tested follower. On reception of the stop message, the final values are output, and detailed event and match protocols are written to external files for later analysis.

We chose to implement the evaluation outside of the score following objects, instead of inserting measurement code to them. This *black box testing* approach has the advantages that it is then possible to test other followers or previous versions of our score following algorithm to quantify improvements, to run two followers in parallel, and that evaluation can be done for Midi and audio, without changing the code of the followers.

However, with the opposite *glass box testing* approach of adding evaluation code to the follower, it is possible to inspect its internal state (which is not comparable with other score following algorithms!) to optimise the algorithm.

4.4 Tests

We have collected a database of files for testing score followers. This database is composed of audio recordings of several different interpretations of the same musical pieces, by one or several musicians, and the corresponding aligned score in Midi format. The database principally includes musical works produced at Ircam using score following (Pierre Boulez *Anthèmes II*, Philippe Manoury *Jupiter*, ...) but also several interpretations of more classical music (Moussorgsky, Bach).

The existing systems that are candidates for an objective comparative evaluation are: *Explode* [16], *f9* [17], *Music Plus One*[2] [23], *ComParser*[24], and the systems described in [2, 9]. This evaluation is still to be done.

[2] `http://fafner.math.umass.edu/`

4.4.1 Audio Following

On our follower, we carried out informal subjective tests with professional musicians on the performance of the implemented score follower together with a jMax implementation of *f9*, a score follower that is based on the technique reported in [17], and a jMax implementation of the Midi follower *Explode* [16], which received the input from a midified flute. Tests were carried out using pieces of contemporary music that have been composed for a soloist and automatic accompaniment.

In *Pluton* for flute, the audio follower *f9* made unrecoverable errors already in the pitch detection, which deteriorated the performances of the score follower. With *Explosante-fixe* the midified flute's output was hardly usable, and lead to early triggers from *Explode*. Our audio follower *suivimidi* follows the flute perfectly.

Other tests have been conducted with other instruments, using short excerpts from *Anthèmes II* for solo violin, with a perfect following both of trills and chords. The different kind of events, that are not directly modeled by *f9* or *Explode*, required ad hoc strategies for preventing the other followers to loose the correct position in the score.

An important set of tests have been carried out on the piece *En Echo* by Philippe Manoury, for a singer and live-electronics. Different recordings of the piece have been used, they were performed by different singers and some of them included also background noise and recording of the live-electronics in order to reproduce a concert situation. The performances of *f9*, which is currently used in productions, are well known: there are a number of points in the piece where the follower gets lost and it is necessary to manually resynchronize the system. On the other hand, *suiviaudio* succeeded to follow the complete score, even if there was some local mismatch for the duration of one note.

Tests on *En Echo* highlighted some of the problems related to the voice following. In particular, the fact that there are sometimes two consecutive legato notes in the score with the same pitch for two syllables, needed to be directly addressed. To this end we added a new feature in our model, the *cepstral flux*, as shown in figure 2. Moreover, new events typical of the singing voice needed to be modeled, as fricatives and unvoiced consonants.

4.4.2 Midi Following

Monophonic tests have been developed for the Midi follower *suivimidi*. The testing of Midi followers is easier because it is possible to change the performance at will, without the need of a performer. In case of a correct performance, *suivimidi* was always perfectly following, and it has been shown to be robust to errors affecting up to 5 subsequent notes, even more in some cases.

Real life tests with the highly polyphonic *Pluton* for midified piano showed one fundamental point for score following: Ideally, the score should be a high-level representation of the piece to be played. Here, for practical reasons, we used a previous performance as the score, with the result that the follower got stuck. Closer examination showed that this was because of the extensive but inconsistent use of the sustain pedal, which was left to the discretion of the pianist, resulting in completely different note lengths (of more than 50 seconds) and polyphony. Once the note lengths were roughly equalised, the follower had no problems, even in parts with a trill that was (out of laziness) not yet represented as a single trill score event. This test shows us a shortcoming of the handling of highly polyphonic scores, which will be resolved by the introduction of a decaying weight of each note in the note match probability.

5. CONCLUSION AND FUTURE WORK

We have a working score following system for jMax version 4 on Linux and Mac OS-X, the fruit of three years of research and development, that is beginning to be used in production. It is released for the general public in the Ircam Forum[3]. Porting to *Max/MSP* is planned for next autumn.

Two other running artistic and research projects at Ircam extend application of score following techniques:

One is a theatre piece, for which our follower will be extended to follow the spoken voice, similar to [11, 13]. This addition of phoneme recognition will also bring improvements to the following of the singing voice.

The other is the extension of score following to multimodal inputs from various sensors, leading towards a more modular structure where the Markov model part is independent from the input analysis part, such that you can combine various features derived from audio input with Midi input from sensors and even video image analysis.

6. ACKNOWLEDGMENTS

We would like to thank Philippe Manoury, Andrew Gerzso, François Déchelle, and Riccardo Borghesi without whose valuable contributions the project could not have advanced that far.

7. ADDITIONAL AUTHORS

Norbert Schnell, Ircam - Centre Pompidou, Applications Temps Réel, email: `schnell@ircam.fr`

8. REFERENCES

[1] B. Baird, D. Blevins, and N. Zahler. The Artificially Intelligent Computer Performer: The Second Generation. In *Interface – Journal of New Music Research*, number 19, pages 197–204, 1990.

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

[3] D. Beeferman, A. Berger, and J. D. Lafferty. Statistical models for text segmentation. *Machine Learning*, 34(1-3):177–210, 1999.

[4] J. Bryson. The Reactive Accompanist: Adaptation and Behavior Decomposition in a Music System. In L. Steels, editor, *The Biology and Technology of Intelligent Autonomous Agents*. Springer-Verlag: Heidelberg, Germany, 1995.

[5] R. B. Dannenberg. An On-Line Algorithm for Real-Time Accompaniment. In *Proceedings of the ICMC*, pages 193–198, 1984.

[6] R. B. Dannenberg and B. Mont-Reynaud. Following an Improvisation in Real Time. In *Proceedings of the ICMC*, pages 241–248, 1987.

[7] R. B. Dannenberg and Mukaino. New Techniques for Enhanced Quality of Computer Accompaniment. In *Proceedings of the ICMC*, pages 243–249, 1988.

[3] `http://www.ircam.fr/forumnet/`

[8] L. Grubb and R. B. Dannenberg. Automating Ensemble Performance. In *Proceedings of the ICMC*, pages 63–69, 1994.

[9] L. Grubb and R. B. Dannenberg. A Stochastic Method of Tracking a Vocal Performer. In *Proceedings of the ICMC*, pages 301–308, 1997.

[10] L. Grubb and R. B. Dannenberg. Enhanced Vocal Performance Tracking Using Multiple Information Sources. In *Proceedings of the ICMC*, pages 37–44, 1998.

[11] A. Loscos, P. Cano, and J. Bonada. Low-Delay Singing Voice Alignment to Text. In *Proceedings of the ICMC*, 1999.

[12] A. Loscos, P. Cano, and J. Bonada. Score-Performance Matching using HMMs. In *Proceedings of the ICMC*, pages 441–444, 1999.

[13] A. Loscos, P. Cano, J. Bonada, M. de Boer, and X. Serra. Voice Morphing System for Impersonating in Karaoke Applications. In *Proceedings of the ICMC*, 1999.

[14] N. Orio and F. Déchelle. Score Following Using Spectral Analysis and Hidden Markov Models. In *Proceedings of the ICMC*, Havana, Cuba, 2001.

[15] N. Orio and D. Schwarz. Alignment of Monophonic and Polypophonic Music to a Score. In *Proceedings of the ICMC*, Havana, Cuba, 2001.

[16] M. Puckette. EXPLODE: A User Interface for Sequencing and Score Following. In *Proceedings of the ICMC*, pages 259–261, 1990.

[17] M. Puckette. Score Following Using the Sung Voice. In *Proceedings of the ICMC*, pages 199–200, 1995.

[18] M. Puckette and C. Lippe. Score Following in Practice. In *Proceedings of the ICMC*, pages 182–185, 1992.

[19] L. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–285, 1989.

[20] C. Raphael. A Probabilistic Expert System for Automatic Musical Accompaniment. *Jour. of Comp. and Graph. Stats*, 10(3):487–512, 1999.

[21] C. Raphael. Automatic Segmentation of Acoustic Musical Signals Using Hidden Markov Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(4):360–370, 1999.

[22] C. Raphael. A Bayesian Network for Real Time Music Accompaniment. *Neural Information Processing Systems (NIPS)*, (14), 2001.

[23] C. Raphael. Music Plus One: A System for Expressive and Flexible Musical Accompaniment. In *Proceedings of the ICMC*, Havana, Cuba, 2001.

[24] Schreck Ensemble and P. Suurmond. ComParser. Web page, 2001. `http://www.hku.nl/~pieter/SOFT/CMP/`.

[25] B. Vercoe. The Synthetic Performer in the Context of Live Performance. In *Proceedings of the ICMC*, pages 199–200, 1984.

[26] B. Vercoe and M. Puckette. Synthetic Rehearsal: Training the Synthetic Performer. In *Proceedings of the ICMC*, pages 275–278, 1985.