COMBINING AUDIO AND GESTURES FOR A REAL-TIME IMPROVISER

Roberto Morales-Mazanares
Dept. Music - CNMAT
Univ. of Calif. Berkeley

Eduardo F. Morales
ITESM - Cuernavaca
Computer Science

David Wessel
Dept. Music - CNMAT
Univ. of Calif. Berkeley

ABSTRACT

Skilled improvisers are able to shape in real time a music discourse by continuously modulating pitch, rhythm, tempo and loudness to communicate high level information such as musical structures and emotion. Interaction between musicians, correspond to their cultural background, subjective reaction around the generated material and their capabilities to resolve in their own terms the aesthetics of the resultant pieces. In this paper we introduce GRI an environment, which incorporates music and movement gestures from an improviser to adquire precise data and react in a similar way as an improviser. GRI takes music samples from a particular improviser and learns a classifiers to identify different improvision styles. It then learns for each style a probabilistic transition automaton that considers gestures to predict the most probable next state of the musician. The current musical note, the predicted next state, and gesture information are used to produce adequate responses in real-time. The system is demonstrated with a flutist, with accelerometers and gyros to detect gestures with very promising results.

1. INTRODUCTION

Skilled and experienced improvisers shape a music discourse in real time by continuously modulating pitch, rhythm, tempo and loudness to communicate high level information such as musical structures and emotion. Interaction between two or more musicians, correspond to their cultural background, subjective reaction around the generated material and their capabilities to resolve in their own terms the aesthetics of the resultant pieces.

During an improvisation musicians are alert, tolerant and communicative. Body expressions also motivate each other to assume different moods and in most of the cases create dramatic contrasts in the development and phrase definition of the material . The almost endless musical structures and diversity in expression that can emerge in a group of musicians from one set and another, is rarely achieved in interactive computer system.

Real-time interactive environments are typically constructed with a limited set of procedures, leaving very little room for any kind evolution during a performance. By set of procedures we mean any data type format, random processes and/or algorithmic method within a predictive output. This constrains human improvisers to gaming the interactive system to generate musically cogent output, a distraction from their own musical potential. We have adopted two strategies to break from these limitations: the integration of gesture and interpreted musical data; and the use of an on-line learning system, GRI, which predicts the next most probable state of the musician from which an adequate audio response is produced in real-time. The system's predictions of the next state are continuously updated during performance considering the actual state transitions of the musician, allowing a gradual adaptation to the current piece as its surface form and "style" evolves. This approach avoids the well known difficulty of previous systems with embed knowledge musical phrases, styles and even recognize different soloist interpretations for specific music passages. Unfortunately, their inefficiency starts when the material is not in their database and/or algorithmic domain. That is, when any performer/improviser defines certain collections of discrete events as musical phrases which might contradict the system.

In the particular application of the GRI system we will describe, we use accelerometers to capture attacks and/or tension in the fingers of the musician, and gyros to detect angular displacements of the flute, for instance to signal the start of a musical event or emphasize some particular notes.

Section 2 describes the proposed approach called GRI. Our approach is contrasted with related work in Section 3. Finally Section 4 gives conclusiones and future research directions.

2. GRI

The idea is to learn a predictive model of what a musician will do in real time. We argue that the information provided from sensors can help to produce more accurate models. In order to produce a companion improviser, GRI follows three main stages. In the first state, information from audio and gestures produced by a musician is used to create a classifier of improvisation styles. In the second stage, a probabilistic transition automaton is learned for each style. The third stage is used during performance, where the current audio and gesture information produced by a musician is used to predict the next most probable state using the previously build classifier and automata. The audio and gesture information with the predicted next state are used to produce adequate output. In the following sections each stage is described in more detail.

In this work we are using music information produced

by a flutist. Two gyros and one accelerometer are attached to the end of the flute. The accelerometer is very effective in detecting tension or attacks in the fingers of the flutist. Fast and strong finger movements are clearly shown by the accelerometer. This is used to distinguished states and help the classifier and the automata to make better predictions. The gyros are used to capture movements with the flute which normally occur (with our flutist) just before starting a new musical phrase or to signal particular notes which are considered relevant by the musician. The gyros are placed orthogonal to each other to capture left-rigth and up-down movements regardless of the orientation of the musician. We plan to include another one in the future to capture twists of the flute. This information is used to produce more adequate audio outputs from a predicted state.

2.1. Learning musical models

As previously mentioned, GRI first recognizes a particular improvisation style of a musician and then predicts the next possible state for each style. This initial classification considerably reduces the state-space and allows faster response times. The musician can still jump between styles during the same musical piece, although GRI will predict a state and produce audio from the previous style. Once GRI recognizes that the musician has change styles it will produced audio according to the new style. This is normally what happens during improvisation performance when one musician decides to change styles, so splitting the system into different styles can be reasonably justified and considerably helps in the performance of the system.

Given a set of music examples and gesture information from sensors of different improvisations styles of a musician, GRI learns a classifier to quickly identify a particular style. In this paper we only considered three different styles: (i) long (with considers long notes), (ii) short (considers mainly short notes) and (iii) erratic (which considers a more "chaotic" style).

The musical and gesture information is used to characterize each state. We tried different classifiers from Weka [11] with continuous and discretized data and considering different attributes. The best results (considering accuracy and simplicity) were obtained using as attributes the current trend of the notes (if the current note is increasing, decreasing or steady with respect to the previous note), a discretized range of the dynamics, a discretized range of the rhythm and a discretized range for the accelerometer. We used an entropy-based discretization which uses MDL as stopping criterion [2] to discretize these values. This representation produces a small number of abstract states from which a very simple classifier can be constructed. We used Weka to construct a classifier from 575 samples with 90% of accuracy using 10-fold stratified cross-validation, which was then directly coded into GRI.

Table 1. Construction of the transition probability automata.

Let $Auto$ be the current automata for a particular style
Given a state transition $A \rightarrow B$
If A or B are new states, add them to $Auto$.
If $A \rightarrow B$ is a new transition
Then set $P(A \rightarrow B) = 1/1$
    if there are no other transitions from state A in $Auto$
    else
        set $P(A \rightarrow B) = 1/(a+1)$
            given that a transitions have been previously observed from A in $Auto$
        set $P(A \rightarrow C) = c/(a+1)$
            for all states $C \neq B$ which have c transitions from A in $Auto$
Else
    set $P(A \rightarrow B) = (b+1)/(a+1)$
    set $P(A \rightarrow C) = c/(a+1)$
        for all states $C \neq B$ in $Auto$

2.2. Learning probabilistic automata

The attributes used for the classifiers were the same used to describe the states in the construction of the probabilistic transition automata. A different automaton is learned for each musical style. The same data used to construct the classifier was used to initially construct the automata.

The transition probabilities between states are updated with each observed transition between states. In this paper, $P(A \rightarrow B) = b/a$ is used to denote the transition probability from state A to state B, where b in the number of times that the transition from state A to state B has been observed and a is the number of times that there has been a transition from state A. A simple mechanism is used to update the transition probabilities:

$$P(A \rightarrow B) = (b+1)/(a+1)$$

if the current transition is from state A to state B, and

$$P(A \rightarrow C) = c/(a+1)$$

for all the other outgoing states C which are different from state B. New transitions are initialized to $1/1$ if there are no previous transitions from state A or to $1/(a+1)$ if there has been previously a transitions from state A. This process is more clearly described in Table 1.

2.3. Producing audio

Once the classifier has been induced and the automata have been constructed, GRI receives musical and sensor information to produce adequate audio responses in real-time. The input information is first discretized and transformed into one valid state. GRI uses the classifier to identify the current style and the automaton to use. The automaton is used to predict the most probable next state.

Table 2. A general overview of GRI

Given a set of music and gestures samples labeled with a particular style:

- Transform the sample set into abstract samples
- Learn a classifier to distinguish between styles
- Learn an automaton for each style according to Table 1

Given music and gesture information during performance

- Transform the information into an abstract state
- Update the observed probabilistic transition from the previous state to the current state (or do nothing if first time)
- Predict the style using the classifier
- Predict the most probable state using an automaton
- Produce audio considering the predicted state, the current note and gesture information

The predicted state, the current note and the information from the gyros are used to produce audio. Once the next state is obtained from the musician, it is compared with the predicted state. The transition probabilities of the used automaton are updated with the real transition according to Table 1. The continuous update of the automata during performance can help to adapt the system to the current improvisation mood of the musician. Similarly, the status of the automata after different performances can be stored and used in future performances. Table 2 has an overview description of GRI.

To produce adequate audio, Max has information of the current note, rythm, dynamics and gyros and receives information from the preditcted state, that is, the predicted trend in note (i.e., if the next note is predicted to be of higher/lower pitch) and the predicted ranges of rythm, dynamics and accelerometer. This information is used to produce adequate audio. GRI has information about intervals, rythms and other musical relations to produce its output. When the flute is not moving, within a certain threshold in the values of the gyros, GRI stops producing audio.

It should be noted that due to its classification process, simple automata structures and representation of states, GRI is able to respond in real-time. In addition, its continual update of the automata during performance allows GRI to adapt its transition probabilities and include states that were not previopsuly considered, making it a very flexible system.

The classifier, the abstraction process and the learning automata are coded in Prolog. Max sends to Prolog the new observed audio and sensor variables. Prolog transforms and classifies the input, predicts the next state and updates the transition probabilities according to the actual observed transition of the musician. Prolog sends to Max a new prediction with the current note. Max takes this information and the information from the gyros to produce adequate audio. Figure 1 illustrates this process.

2.4. Experiments

The classifier was learned with 575 samples which were also used to construct the initial automata. The *erratic* style had 16 states and 36 transitions, *large* had 12 states and 24 transitions, while *short* had 19 states with 84 transitions. GRI is able to update the automata during performance, producing more accurate models over time.

With these initial automata several test were performed. In our experience, after two or three interactions, GRI is able to follow in a very aceptable way a musiciacian.

3. RELATED WORK

Several researchers have also considered having computer music companions during performance. Most of them rely of the user to progam the system and customize the system until it is able to produce interesting musical results (e.g., [8, 7]. Other more recent systems, like Dannenberg's interactive performance system [1] music generation is either hand-coded or trained using supervised learning but it is done to follow the composer's goal instead of the performer's goals. David Wessel's assisting performance setting [10] offer more open-ended improvisations possibilities, however they are still primary focus in uman authored aesthetics. The idea is to organize and control the access of musical material so that a performer can author meaningful musical expperiences on-the-fly. In a more recent wotk, Belinda Thom describes a sistem that is able to customize itself during live performances [9] which is more closely related in spirit to our work. She uses clustering mechanisms to identify different *playing modes* which is similar to our initial classificatin process, and directional Markov chains to generate music. One of the main differences with our work, is that we are also incorporating information about gestures from the sensors which allows to capture movements and tension in the musician that helps to produce better responses.

4. CONCLUSIONS AND FUTURE WORK

We have presented a system called GRI which combines audio and gesture information to produce audio responses in real-time. GRI learns a classifier and probabilistic automata to predict the next most probable state of a musician. The classifier and the abstraction in the representation of states simplifies the task while maintaining rich musical and gesture information, which allows adequate real-time responses.

The use of sensors enrich the representation of states to capture additional features not present in the audio. Accelerometers are adequate to identify tension and strength in the execution of a flutist, while the gyros are good to identify gesture movements with the flute associated with particular music events.

GRI is continuously updating its probabilistic automata during performance which allows to capture, to a certain extend, changes in the mood of the musician.

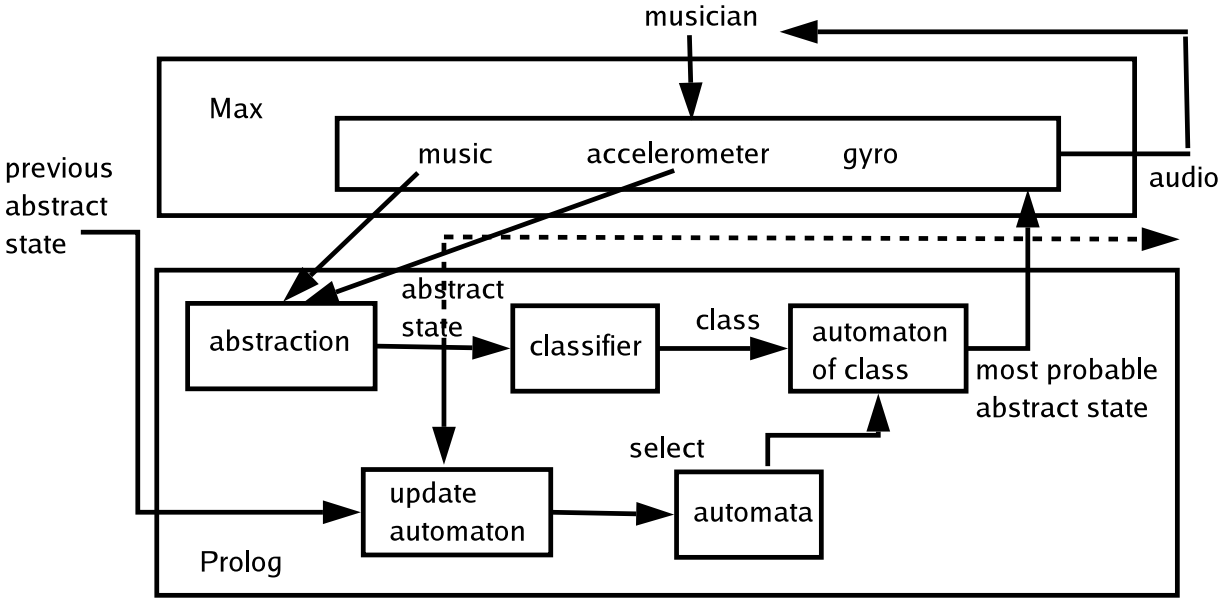

Figure 1. Information flow in GRI.

As future work, we would like to consider richer probabilistic transition models like Hidden Markov Models, Dynamic Bayesian Networks, and probabilistic logic models (e.g., Bayesian logic programs [5] or probabilistic relational models [3]). We would also like to enrich our models with more gesture information.

Acknowledgements

The authors will like to thank Adrian Freed and Rimas Irizarry for providing support on CNMAT's programmable connectivity processor, software and hardware modifications for audio and sensors used in this developement. Part of this research was funded by a grant from UC-MEXUS.

5. REFERENCES

[1] R.R. Dannenberg, B. Thom, and D. Watson (1997). A machine learning approach to musical style recognition. In *Proceedings of the 1997 ICMC*. International Computer Music Association.

[2] U.M. Fayyad and K.B. Irani (1993). Multi-interval discretization of continuous-valued attributes for classification learning. In *Proc. of the Thirteenth International Joint Conference on Artificial Intelligence*, Chambery, France. San Francisco: Morgan Kaufmann, pp. 1022-1027

[3] N. Friedman, L. Getoor, D. Koller and A. Pfeffer (1999). Learning probabilistic relational models. In T. Dean, editor, *Proceedings of the International Joint Conference on Artitificial Intelligence* (IJCAI-99), pages 1300-1309, Morgan Kaufmann.

[4] K. Kersting and L. de Raedt (2000). Bayesina logic programs. In J. Cussens and A. Frisch, editos, *Proceedings of the work-in-progress track at the 10th. International Conference on Inductive Logic Programming*, pages 138-155.

[5] B. Pennycook and D. Stammen (1993). Real-time recognition of melodic fragments using the dynamic timewarp algorithm. In *Proceedings of the 1993 ICMC*. International Computer Music Association.

[6] R. Rowe (1993). *Interactive Music Systems : Machine Listening & Composing*. MIT Press.

[7] B. Thom (2001). Machine Learning Techniques for Real-time Improvisational Solo Trading. In *Proceedings of the 2001 International Computer Music Conference* Havana, Cuba, 2001

[8] D. Wessel and M. Wright (2000). Problems and prospects for intimate musical control of computers. In *HI 01 Workshop New Interfaces for Musical Expression*. ACM SIGCHI.

[9] I.H. Witten and E. Frank (2000). *Data Mining: practical machine learning tools and techniques with Java implementations*. Morgan Kaufmann.