

Orchestral Musical Accompaniment from Synthesized Audio, by Christopher Raphael

In this paper the author explains their system which is capable of synthesizing a sensitive and responsive orchestral accompaniment to a live musician who is playing a piece of music from a score. The way system works is similar to the learning achieved in series of rehearsals. During the performance the system uses the information collected in training phase, the musical score, as well as on-line analysis of the soloist acoustic signal.

The system contains three components: “Listen”, “Synthesize”, and “Anticipate”.

“Listen” uses a hidden Markov model (HMM) to follow the soloist’s performance through a score and analyze the acoustic signal of her performance and estimate the note onsets. For each note in the score a Markov model is created and chained together from left to right to produce the HMM. It benefits from automatic adaptation to changes in solo performance, such as change of instrument or noises, acoustic trainability, onset time estimation accuracy, fast computation, and etc.

Before producing the re-synthesized output, the start time (in frames) of each chord should be determined. This is done by *indexing* the polyphonic audio in an off-line manner, using a strategy similar to HMM. Because of the polyphonic nature of the data, and variable nature of timber and texture of orchestral data they build the model by hand.

“Synthesize” uses a phase vocoder to play a prerecorded audio file back at different [play] rates without distorting the pitches and ensures smooth phase change in consecutive frames. Inputs of this component are the audio recording of the accompaniment and the index.

“Anticipate”, mediates between “Listen” and “Synthesize” by creating a probabilistic model, a Bayesian network consist of hundreds of Gaussian random variables including observable (e.g. note onset times) and unobservable quantities (e.g. tempo). This phase requires synthesis of three knowledge sources: the musical score, output of “Listen” component, i.e. the note boundaries in solo part, and examples of solo performance for prediction of the onsets.

“Anticipate” begins with a model for rhythmic interpretation, used to build a model for accompaniment and then joins the model of solo and accompaniment. The next phase is the rehearsal phase in which like real rehearsals the system learns the rhythmic interpretation by estimating the parameters of the system which are initially set to a neutral interpretation. After each performance the audio file is parsed off-line and used for training. All the above information, generated and collected during the previous phases, is used to schedule an event for the real-time accompaniment. As soon as an accompaniment note is played it schedules the next event and in case that a solo note is detected after scheduling an accompaniment event, it should be rescheduled.

Due to the nature of the system, there are three types of asynchronous computations. First is the listen component which analyzes the input with a rate of 31 fps and wakes up every time a new frame is ready. Synthesize module should process output with a higher rate of 47 fps and its callback has higher priority. When the audio data is synthesized then the samples are written and played 0.025 second later to overcome the probably delivery problems.

Their system is written in c and tested on a PIV Linux system. Their demonstration at the conference was accompanied by a live performance.