CAIRA – a Creative Artificially-Intuitive and Reasoning Agent in the context of ensemble music improvisation

2nd-year Annual Report, May 2012, Project ID 1002851

Jonas Braasch, Selmer Bringsjord, Pauline Oliveros, Doug Van Nort, Colin Kuebler, Anthony Parks

1 Executive Summary

This report summarizes our work for the second year of the project: CAIRA – a Creative Artificially-Intuitive and Reasoning Agent in the context of ensemble music improvisation. The main goal of our research is to develop an intelligent music agent that can improvise or conduct music live using a cognitive architecture. The CAIRA system uses several stages to simulate the functionality of the auditory central nervous system as shown in Figure 1. A Computational Auditory Scene Analysis (CASA) stage is used to extract the perceptually relevant features of sound. A machine listening algorithm based on a Hidden Markov Model and Empirical Mode Decomposition was designed to learn musical gestures and phrases. CAIRA also uses first-order logic to understand music concepts on a symbolic level, and the agent builds on genetic algorithms to generate new sound material or provide a visual score. Last year, our team worked on all components of CAIRA. In this report, we will first summarize the results for each component and also highlight some of our current early-stage activities that have yet not produced publishable results. The first section will also include an outlook for our third year activities. In the following sections, we will report the findings in greater length based on project-related publications.

1.1 Computational Auditory Scene Analysis

Computational Auditory Scene Analysis (CASA) is a compelling problem in acoustics, because the human auditory system outperforms our best machine listening algorithms. A good machine performance is especially difficult to achieve for music performances due to the complex sound mixture of musical instruments, which overlap in time and frequency. Our preference for reverberant concert venues further adds to the challenging nature of designing a CASA system. Currently, we are tackling this problem two-fold. On the one hand, we developed a scheme that we titled Microphone-aided Computational Auditory Scene Analysis (MaCASA). In this scheme, the individual instruments are captured using closely-positioned microphones to allow acoustic separation of the individual auditory streams. While close-miking techniques are common in sound-recording practice, we extended this technique with an additional calibration microphone, which is necessary to auto-calibrate the levels of the instruments, so CAIRA can understand time-variant musical constellations of the individual musical instruments of an ensemble. The technique will be described in more detail in Section 5. In addition, we were interested to improve humanoid approaches that build on
recording sound using a binaural mannequin (dummy head). Current systems typically use a static head, although most human listeners utilize head movements to improve their ability to extract relevant features of sound. Consequently, we designed a rotatable dummy head with stereoscopic vision that from now on will serve as the front end for CAIRA. The system will be described in detail in Section 2.

1.2 Machine Listening of Sound Textures

The main accomplishments of the reporting period for this area included a thorough investigation of sound texture analysis using Empirical Mode Decomposition (EMD). The investigation was a continuation of the Audio Engineering Society Convention paper which we reported last year. The main finding of the extended project was that our approach of sound texture analysis (the first based on EMD) outperformed MFCC approaches (the current standard for sound texture analysis) in categorizing five different sound textures (crackling fire, typewriter action, rainstorms, carbonated beverages and crowd applause). The paper has been submitted to the Journal of the Acoustical Society of America (JASA) and is now in the second rounds of reviews. Section 3 of this report is based on the renewed submission, which was submitted to JASA on May 19th, 2012.
1.3 Freely Improvising, Learning and Transforming Evolutionary Recombination (FILTER) System

FILTER is sub-system of CAIRA that simulates music improvisation based processes inspired by human intuition. It builds on machine listening, and on intuitive and spontaneous transformations of a human performer’s input in a way that is shaped by learning stylistic trends. Unlike most other existing systems, FILTERs actions are informed by an electroacoustic aesthetic that favors a sound-oriented view on performance output rather than one determined by symbolic music theoretic rules. In short, it operates based on actual sound and not on a representation of music using the MIDI format. The main achievement of this reporting period was to bring the system to a robust level, where it always produces a satisfying experience (as opposed to a system that works really well sometimes, but does not produce interesting or comprehensive results at other times). The performance of FILTER has been documented on Video using the Vimeo platform featuring percussionist Sam Sowyrda (http://vimeo.com/32119052) and accordionist Pauline Oliveros (http://vimeo.com/32120127).

In the near future, we are planning to have a server version of FILTER, where users can access the system remotely and experience its performance first-hand by playing with the system live. Section 4, which is based on a paper given at the NIME 2012 conference, describes our latest achievements for FILTER.

1.4 Simulation of Cognition using Logic-based Reasoning

Significant progress has also been made for the logic reasoning module of CAIRA. Based on the proposed architecture of the last reporting period, we were able to design a functioning prototype that has been demonstrated at the Music, Mind, and Invention 2012 Workshop Workshop and at Rensselaer’s Experimental Media and Performing Arts Center (EMPAC) for a delegation from Disney Imagineering. Over the summer, we will produce video material to document the performance of the current stage of the CAIRA system. In a nutshell, in our demo CAIRA participates in a free music trio, performing as the third trio member using V-Accordion material recorded by Pauline Oliveros, which is then processed through FILTER. Using logic-based reasoning, CAIRA understands different musical configurations and can recognize them live through analyzing musical tension curves of the other two human improvisers (e.g., Braasch and Van Nort). It understands, for example, that Braasch is intending to perform a solo if his tension curve is above a threshold of Van Nort’s tension curve. Based on this insight, CAIRA can either accompany Braasch by performing with a low-tension curve or it can also take initiative by taking a different action (e.g., cutting out Braasch by presenting material that significantly exceeds Braasch’s tension curve). CAIRA also presents a dynamic video score that reflects the different ensemble states. The current system is described in detail in Section 5, which is based on a conference paper presented at the Music, Mind, and Invention Workshop that was recently held at the College of New Jersey. This paper does not yet describe our most recent accomplishment of CAIRA producing different sonic environments based on the different ensemble states (e.g., creating a different acoustical environment for a solo of a particular ensemble member). The environments are defined through an acoustical model that simulates early reflections and late reverberation.
(e.g., simulating a concert hall in a studio-type environment) or multi-channel environmental recordings (e.g., the sound of a forest for the low-tension tutti ensemble state).

Now that we have a functioning prototype, our main goal is to expand the cognitive architecture of CAIRA. Currently, we are looking into integrating the cognitive architecture Clarion, another RPI development by Cognitive Science Prof. Ron Sun into CAIRA or to develop a completely new intermediate-level module that bridges FILTER and the logic-based reasoning module Handle with a Bayesian-based approach that allows CAIRA to make cognitive decisions where a logical proof does not exist or where timely decisions have to be made. In addition, we will extend CAIRA’s ontology to understand the concepts of tonal music theory (to complement the intuitive listening functions of FILTER) and work on a system that allows CAIRA to automatically prove theorems and develop new concept based on the results off-line and apply the results live in the next music session.

We are also working on a new distributed composition, “Trio en Trio”, to explore the performance of CAIRA in a multi-agent scenario. In this piece, a trio of three musicians (Braasch, Oliveros, and Van Nort) will be complemented by a trio of three CAIRA instances (with identical architectures but different personality settings) as shown in Fig. 2. Each musician will be represented by its own trio, performing with two distinct types of instrumental input together with one of the CAIRA instances as a third partner, as follows:

<table>
<thead>
<tr>
<th>Performer</th>
<th>Instrument I</th>
<th>Instrument II</th>
<th>Agent Instrument</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jonas Braasch</td>
<td>saxophone</td>
<td>Moog bass pedal</td>
<td>CAIRA I</td>
</tr>
<tr>
<td>Pauline Oliveros</td>
<td>left V-Accordion hand</td>
<td>right V-Accordion hand</td>
<td>CAIRA II</td>
</tr>
<tr>
<td>Doug Van Nort</td>
<td>GREIS</td>
<td>voice</td>
<td>CAIRA III</td>
</tr>
</tbody>
</table>

Figure 2: Schematic communication scheme for Trio en Trio. Left: Inter-ensemble communication via human musicians. Right: Inter-ensemble communication via CAIRA agents.

Each of the three “trios” will perform using the tension curve model described in Section 5, and then the three encapsulated trios will perform as an overarching trio based on the three encapsulated ensemble tension curves. The interaction between the three encapsu-
lated trios can be varied between agent communication and human communication in order to explore differences in both forms of communication (see left and right graph of Fig. 2). Another focus of the third project year will be a formal evaluation of the CAIRA system. In addition to preference tests, surveys and interviews as stated in the original project proposal, we will also conduct tests to investigate CAIRA’s ability to identify different ensemble constellations. In this test, we will record the states identified by CAIRA over time along with the states identified by the participating musicians (the latter using foot pedals). During the recordings, the responses of each participant (human performers, CAIRA) will not be disclosed to the other two participants. Consequently, the agreement across all three participants will reveal the extent to which the musicians were able to communicate. The comparison between the two human performer results, as well as the results between CAIRA and each of the human performers, will provide further insight into the extent to which CAIRA is able to compete with human performance for this type of task.
A binaural moving dummy head was constructed to serve as the perceptual input apparatus for the CAIRA agent as well as for other research projects in spatial hearing at the communication acoustics and aural architecture research laboratory (CA³RL) at Rensselaer Polytechnic Institute. The human auditory is remarkable in its ability to perceive sound in three dimensions. By analyzing the differences in arrival times and level differences arriving at both ears, the auditory cortex is able to make judgments about the location in space of a sound source located in a given environment. Furthermore, in situations where the location of the source is ambiguous (such as when the source is directly in front of or directly behind the listener), small movements of the head (as little as 15 degrees) can be made to accurately determine the position of the sound source (Perrett 1997). It has also been shown that listener head movement is involved when listeners evaluate other spatial characteristics of the sound, such as reverberance, apparent source width, and listener envelopment (Parks October 20-23, 2011, Kim & Brookes 2007).

From this prior research we have constructed a dummy head that can rotate in the azimuthal plane. Having the capability for movement is our novel approach and a useful one at that. For example, a binaural model that receives inputs from two microphones located
in both ear canals of the dummy head may have a high rate of error in localizing sound sources as compared to human listeners. However, when the likelihood of error is high then the model can send a message to the robotic head and instruct it to rotate and then capture more data. The two sets of data, the data from the original head position along with the data from the rotated position, can be compared and less error-prone conclusions can be inferred from the output of the model in any given auditory scene. The addition of stereoscopic video capture is to provide visual cues to CAIRA, to set markers on individual players and track their movements, gestures, and spatial arrangement.

The dummy head was constructed using a rapid prototyping CNC machine at the School of Architecture’s machine shop to transform a 3D CAD model of a human head into a dummy head made from medium-density fiberboard (MDF). The ears for the head were molded from his own out of silicone. The enclosing box is constructed from MDF as well. A shelf holds the driving motor apparatus, which consists of a stepper motor that operates at 1.8 degrees of precision beltdriving a pulley keyed into a 0.5” diameter linear steel shaft. The shaft is also threaded through a bearing and protrudes from the top of the enclosure, where the head is mounted to a collar keyed into the shaft that screws into the head from its base.

The stepper motor is powered through an Arduino Atmega 328 microcontroller. The microcontroller provides power to the leads as well as control over the motor. The stepper motor control software is written in Processing (a Java-like language) using the AFMotor and AccelStepper libraries from ladyada.net The software is written in such a way that any language or software that can send messages over serial port (the USB of the arduino disguises itself as a serial port through COM3), such as MATLAB, can control the amount of movement of the motor. For example, if MATLAB sends a message over COM3 that reads “F90” the motor will turn the appropriate amount of rotations such that the head moves forward (clockwise) 90 degrees. ‘B’ indicates counter-clockwise motion. Because of the brevity of the messages, MATLAB, along with the primary Max/MSP patch that guides CAIRA, can work together to communicate motion directions to the dummy head using a combination of serial part and open sound control (OSC) messsages.

For the input devices, small electret microphone capsules were wired into each siliconemolded ear. These microphones are then fed into a preamp which in turn feeds its output to an EASERA audio gateway, for further processing and for input into Max/MSP and Matlab (along with any digital audio workstation). For cameras, Apple iSight webcams with 0.25 diameter sensors were used and inserted into the “eye sockets”.
3 Sound Texture Recognition through Dynamical Systems Modeling of Empirical Mode Decomposition\(^1\)

Doug Van Nort, Jonas Braasch, Pauline Oliveros

Abstract

This paper describes a system for modeling, recognizing and classifying sound textures. The described system translates contemporary approaches from video texture analysis, creating a unique approach in the realm of audio and music. The signal is first represented as a set of mode functions by way of the Empirical Mode Decomposition (EMD) technique for time/frequency analysis, before expressing the dynamics of these modes as a linear dynamical system (LDS). Both linear and nonlinear techniques are utilized in order to learn the system dynamics, which leads to a successful distinction between unique classes of textures. Five classes of sounds comprised a data set, consisting of crackling fire, typewriter action, rainstorms, carbonated beverages and crowd applause, drawing on a variety of source recordings. Based on this data set the system achieved a classification accuracy of 90%, which outperformed both a Mel-Frequency Cepstral Coefficient (MFCC) based LDS-modeling approach from the literature, as well as one based on a standard Gaussian Mixture Model (GMM) classifier.

3.1 Introduction

In the realm of sound signal modeling, a great deal of attention has been paid to representing the shape of a sound spectrum and its constituent parts as it moves through time. This work has been motivated by speech research in telecommunications as well as our understanding of perceptual salience in musical tones, dating back to the work of Helmholtz (von Helmholtz 1885). Modeling of the human vocal mechanism has led to the source-filter approach, which represents an audio signal as a slowly-varying filter that is generally excited by some noise or periodic pulse, notably with the linear predictive coding (LPC) method and related variants. An understanding of the perceptual importance of spectral peaks led to a model in which audio is expressed as a sum of sinusoids (McAulay & Quatieri 1986), which was later augmented with a noise component in order to add to the model’s articulatory and expressive power (Serra & Smith 1990). These families of approaches have resulted in a great number of analysis and transformation systems for speech and musical instrument sounds, where it has been shown time and again that elements such as the spectral envelope, sinusoidal partials and even filtered noise components contribute to perception of timbre, pitch and dynamics. That said, there is a large class of audio signals that relate to environmental “background” sounds as well as to many contemporary music practices that are not well represented by these approaches because they lack salient and stationary features such as defined partials, or regular onsets. This class can collectively be referred to as sound *textures*. Sounds of

---

\(^1\)Section based on a submitted journal paper that is currently under review at the Journal of the Acoustical Society of America.
this sort can be described as quasi-random local variations that are possibly quasi-periodic, while having stationarity in regards to global spectral or temporal properties. This class includes sound such as those produced by crackling fire, wind blowing through trees, babbling streams, crowd noise as well as musical textures created through superposition of many short fragments of similar sound sources. Our goal is to develop a system that can learn the underlying structure unique to these different classes of textures, for the purpose of recognizing and differentiating amongst these disparate groups, and ultimately for understanding how humans differentiate between these classes of sounds. To this end we have learned from and built upon several past attempts to capture and describe underlying processes that give rise to sound texture phenomena.

3.2 Overview

The remainder of this paper is structured as follows: we first describe existing literature in sound texture analysis, notably articulating the distinct modeling assumptions and research context (including adaptions of work from the visual texture literature). Then, we present our time/frequency analysis framework based on Empirical Mode Decomposition (EMD), which builds upon this past sound texture research. After this we present the probabilistic dynamical systems framework known as Dynamical Texture Modeling (DTM) that learns the temporal structure of EMD-based features to achieve the ultimate goal of recognition and classification. We conclude with an evaluation of our system, including a comparison to an existing application of DTM to musical classification (Barrington et al. 2010), as well as to a standard technique (Gaussian Mixture Modeling) from the music information retrieval literature (Aucouturier et al. 2005).

3.3 Sound Texture Analysis

A small collection of studies have approached the problem of modeling, analysis and resynthesis of sound textures. Arnaud and Popat (St-Arnaud & Popat 1998) utilized a bank of octave filters for analysis in order to arrive at a set of feature vectors, after which a k-mean probabilistic cluster-based approach was used to encode the most likely transitions between vectors. Resynthesis occurs by means of an excitation filter bank, with temporal sequences of events defined by a binary tree structure created during the analysis stage. In a somewhat similar vein, Dubnov et al. (Dubnov et al. 2002) applied a statistical learning algorithm to the coefficients of a wavelet transform, describing the likelihood of textural structure to move between “grains” of audio over time. In both of these works the idea is to assess the signal properties of some fundamental units, and the statistical nature of these units – including the likelihood of given transitions – as the signal evolves over time. Dubnov et al.’s work was adapted from similar research by one of the co-authors that was used for visual texture synthesis (Bar-Joseph et al. 2001). A large portion of other work in this area has also been informed by visual texture research, including that presented by Behm and Parker (Behm & Parker 2004), wherein the computer graphic notions of ’tiling’ and ’stitching’ are used to create a unique sound texture by recombining small sound grains using a chaos mosaic algorithm (first proposed by the Microsoft graphics group (Xu et al. 2000)) that tends to produce sound sequences with little repetition, and which favors smooth transitions. As with
many applications of visual texture, the goal was to create a great deal of subtle variation from a small set of grains, in order to create realistic background sound textures. This is the same goal in Lu et al. (Lu et al. 2004), whose approach is to analyze Mel-Frequency Cepstral Coefficients (MFCCs) (Logan 2000) and compute a similarity measure between adjacent frames. Unique points in audio are extracted, and these are then used as transition points in order to recombine audio with smooth transitions and maximal variety, favoring little repetition.

These works implicitly focus on a functional aspect of sound texture, interpreting this phenomenon as an ambience or background and extracting structural information that is relatively high-level for the purpose of sound modeling. Other approaches have focused on texture as a quality of a singular sound event, and as such have dealt more closely with signal modeling. This was the case with Athineos and Ellis (Athineos & Ellis 2003), where the authors used a dual time and frequency approach to LPC that worked particularly well for textural sounds composed of micro-transients, such as the sound of fizzing soda and crackling or crumpling sounds, but which was less successful with sounds that were smoother and comprised of an assemblage of more pitch-centric grains. The dual time/frequency cascaded LPC technique was further explored by Zhu and Whyse (Zhu & Wyse 2004), with a desire to concatenate arbitrarily long textures, merging a signal model which can handle micro-transients with an ability to create varying background textures, in this case modeled by time-varying filters applied to noise input.

Our work presented in this paper seeks to combine the best of both worlds from existing approaches: to improve the analysis of sound textures through an approach that deeply considers the signal behavior in the modeling step on the one hand, while also using a statistical learning scheme in order to understand the temporal structure of texture on the other. The latter is important because of the nature of this class of signals: globally stationary yet comprised of small grains of similar objects that are stochastic in nature. The same can be said of visual textures, and so there is little surprise that many methods have been adapted from visual texture research towards the task of sound texture analysis. From a perceptual point of view this also makes sense, as humans tend to process a flock of birds, swarm of insects or a traffic jam – both in the visual and auditory domain – as a singular entity rather than as comprised of many disparate events. From an auditory scene analysis point of view we further know that auditory perception tends to group objects in terms of concurrent modulations, which factors into our choice of time/frequency analysis framework. Much like the aforementioned work from the literature, our system further examines the cross-modal nature of texture by adapting two approaches – EMD and DTM – which have been applied primarily to visual texture analysis. Some basic modeling assumptions are now presented before describing our approach to sound texture classification.

### 3.4 Empirical Mode Decomposition for Textural Analysis

#### 3.4.1 Capturing Temporal Modulations and Fine Structure

In the previous research on sound texture modeling, many authors describe an array of global timbral attributes (e.g., energy output from channels of a filterbank) of a set of fundamental units or “grains”, and how these are distributed over time, as with the work of Arnaud and
Popat (St-Arnaud & Popat 1998) and Dubnov et al. (Dubnov et al. 2002). This speaks to the stochastic-particle nature of sound textures. By contrast, the LPC-based texture models look more closely at the qualitative nature of a singular sound texture by examining the temporal structure of the signal at the local level so as to represent micro-transient phenomena – something that also clearly contributes to perceived sound texture. The approach in this paper acknowledges the importance of both modeling stochastic behaviors of grains while also having a more refined description of a signal’s temporal fine structure. In order to decompose sound signals by virtue of their varying degree of temporal fineness, as well as in terms of their modulatory structure we turn to the use of Empirical Mode Decomposition (EMD) as a time/frequency analysis front-end for the system. The EMD technique is a nonlinear time/frequency analysis method (Rilling et al. 2003) that acts in the time domain in order to decompose a signal into a set of modes, which are separated from the given signal by virtue of their differing levels of temporal fine structure as well as grouping similar amplitude/frequency modulations.

3.4.2 Definition

The EMD technique is signal-adaptive and therefore does not in general lead to a set of orthogonal basis functions. It is defined algorithmically (rather than analytically) by virtue of the following process (Flandrin et al. 2004):

1. Given signal \( x(t) \) identify all of the local extrema (maxima and minima).

2. Interpolate across all maxima to produce upper envelope \( e_{\text{max}}(t) \) and all minima to produce lower envelope \( e_{\text{min}}(t) \).

3. Determine the mean of the two envelopes \( m(t) = \frac{e_{\text{max}}(t) + e_{\text{min}}(t)}{2} \).

4. Subtract this mean from the signal, leaving the local detail \( d(t) = x(t) - m(t) \).

5. Repeat steps 1–4 on the detail signal \( d(t) \) until it satisfies two criteria:

   I. The number of extrema and the number of zero-crossings must be equal or must differ by at most one.

   II. The detail signal is considered as zero mean as determined by some relevant stopping criteria (Huang et al. 1998, Rilling et al. 2003).

6. At this point, the resultant detail signal \( d_k(t) \) is subtracted from the input signal and the process begins again on the residual.

At the conclusion of this process (including the variable number of iterations on steps 1-4, known as “sifting”), the signal \( x(t) \) will be decomposed into a set of intrinsic mode functions (IMFs) \( d_k(t) \), while the final IMF may be considered as the overall signal trend \( T_L(t) \) so that

\[
x(t) = T_L(t) + \sum_{k=1}^{L-1} d_k(t).
\]
The resultant mode functions provide information about temporal patterns in the audio at varying level of detail, as can be seen in Fig. 1. This illustrates the first four IMFs from an EMD analysis of a recording of fizzing, carbonated soda. The first mode captures only the sharpest transient, the second contains a quasi-periodic event while the final mode primarily captures the slower evolution from the transient event. Again, these modes describe the unique amplitude/frequency modulations within the given signal. From a physical point of view, if a signal contains modulations driven by a unified source, then the EMD method might be less successful than linear time-invariant techniques of analysis. Given that the concept of a sound texture suggests an assemblage of sources with varying modulation behavior on a local scale, we have considered EMD as a technique well-suited to the task of recognition and analysis of sonic textures, keeping with the idea suggested by Van Nort (Van Nort 2009a).

While the technique operates purely in the time domain, the fact that it functions as a time/frequency decomposition is intuitive when one considers that low-vs.-high modes generally correspond to more-vs.-less fine temporal detail, and thus high-vs.-low spectral placement. This is far from a strict sub-band filtering, as can be seen in Fig. 2, which illustrates the log-spectrum of those time-domain IMFs displayed in Fig. 1. There is clearly an overlap in spectral content, yet each mode is centered in a different area of the spectrum and contains a different shape and functional width. Therefore, each mode function provides idiosyncratic time/frequency information related to temporal regularity, the nature of transient components and complex inter-modulations – all of which may contribute uniquely to the formation of a sound texture. Therefore, while most sound texture methods focus on global timbral features or specialize to particular temporal behavior, this approach segments a signal along that dimension – local temporal structure – that is most pertinent to perception of texture. In its application to visual analysis the dynamic texture method is applied across a spatial image, while in this case it is applied across a “spectrum” created by the set of intrinsic mode functions. In the world of sound, the EMD technique has been used previously to extract long-term rhythmic structures (Heydarian & Reiss 2007), for detecting respiratory crackle sounds (Charleston-Villalobos et al. 2007), for perceptual audio coding (Khalid et al. 2008) and for audio watermarking (Zaman et al. 2010). To our knowledge has not been applied to sound classification in general or to sound texture analysis and recognition in particular.

3.5 Dynamic Textures

3.5.1 Background

In the case of a still image, a visual texture can be considered as a realization of a stationary stochastic process that is invariant in regard to spatial statistics applied to size and location. In the computer vision literature, the concept of Dynamic Textures has been proposed (Doretto et al. 2003) to describe spatio-temporal motion that may be modeled as a sequence of images, each realized by a dynamical system that is excited by an independent and identically distributed (IID) stochastic process – which is to say that it has stationarity in regards to statistics over time. This treats modeling as a system identification problem, wherein the model parameters are learned from observations of example inputs. This approach, which builds on the autoregressive temporal texture modeling presented by Szummer and Picard
Figure 4: First four IMF functions (top to bottom) for fizzing soda audio input, and original signal, as functions of samples.

(Szummer & Picard 1996), allows for recognition by comparison of the model parameters for new inputs to previous training data, and for synthesis by driving the dynamical systems model with an IID process drawn from a Gaussian distribution.

3.5.2 Mixtures of Textures

The dynamic texture model has been extended by Chan et al. (Chan & Vasconcelos 2008) to modeling mixtures of different dynamic textures over time, an approach that we in turn adapt to audio textures. This is a specific case of a mixture model approach, which is a probabilistic model that represents a data set as being comprised of some number of data subsets, so that each member is considered as possessing a mixture of properties from these subsets. Perhaps the most classic example is the Gaussian Mixture Model (GMM), which assumes that any given input data is drawn from a finite number of gaussian distributions. The GMM is used as a classifier by applying the Expectation-Maximization (EM) learning algorithm (Dempster et al. 1977), which we describe in the next section, to the task of learning the mixture model parameters from the given data set. This has been applied to music classification previously (Aucouturier et al. 2005). When more complex combinations of correlated variables are suspected, one can apply a Mixture of Factor Analyzers (MFA) (Ghahramani & Hinton 1997) approach, again using the EM technique to estimate the underlying model parameters. Chan et al. adapted the MFA technique to the Dynamic Texture model, taking a description of the mixture at regular spatio-temporal increments to provide clustering and motion segmentation of video textures. More recently an offshoot project has applied this approach to automatic musical song segmentation (Barrington et al. 2010), which examines global timbral attributes (MFCCs) and fits them to the dynamic texture model on a frame-by-frame basis. In effect, both of these works take the probabilistic framework found in GMM, MFA and similar
mixture models, and provide them with a temporal description. We in turn adapt this dynamic mixture model approach to the recognition of sound textures through a deeper consideration of the underlying time/frequency representation for this class of sounds based on EMD. In merging EMD and DTM, we combine two key characteristics from existing sound texture analysis techniques: consideration of the importance of temporal fine structure (via EMD) as well as representing signals that possess a global stationarity and local stochasticity (DTM). The use of DTM further adds an explicit representation of temporal evolution.

### 3.5.3 Dynamic Texture System Identification

We now describe the Dynamic Texture modeling processing. The fundamental modeling assumption is that we have a random process $y_t$ having some hidden state vector $x_t$ that is governed by the linear dynamical system (LDS) described by

\[
x_{t+1} = Ax_t + w_t
\]

and

\[
y_t = C x_t + v_t,
\]

where $A$ is the state transition matrix, $C$ the observation matrix and $w_t$, $v_t$ are zero-mean Gaussian white noise processes. This is where the suitability for phenomenon such as textures comes into play, in that it assumes stochastic local temporal properties while being quasi-stationary on a larger temporal scale, leading to its popularity in application to video texture. The LDS model can be fully described by including the initial mean and covariance of the state $\{\mu, P\}$ as well as the covariance matrices for each noise process $\{W, V\}$. If we assume that we have a mixture of $K$ such systems, then the $kth$ set of model parameters are described by
φ_k = \{A_k, C_k, µ_k, P_k, W_k, V_k\}.
(4)

As we assume that any observation sequence of length T, \( Y = \{y_1, \ldots, y_T\} \) is drawn from a mixture of dynamic textures (and thus LDS models), the probability \( p(Y) \) of this given sequence occurring is a mixture of the probability \( p(Y|z = j) \) that this sequence came from the \( j \)th mixture component (call this \( z \)) for each possible \( j \), weighted by a prior distribution for each component \( \alpha = \{\alpha_1, \ldots, \alpha_k\} \). This results in the expression

\[ p(Y) = \sum_{j=1}^{K} \alpha_j p(Y|z = j). \]
(5)

Given a set \( \{Y^i\}_{i=1}^{N} \) of such observations, our goal is to achieve system identification by finding the model parameters \( \phi = \{\phi_1, \ldots, \phi_K\} \) as well as component weighting distribution \( \alpha \) that best describes the data by way of finding

\[
\arg\max_{\phi} \sum_{i=1}^{N} \log[p(Y^i|\phi)].
(6)
\]

In light of our state-space LDS model, the model parameters \( \phi \) may be found using the Expectation-Maximization (EM) algorithm, which was introduced by Dempster et. al. (Dempster et al. 1977). The EM technique is an iterative approach that estimates missing information given the current model parameters, and then computes new parameters given these estimates. In this modeling context, we must find the given assignment of mixture component \( z = j \), as well as the hidden state sequence \( X^i \) that gives rise to sequence \( Y^i \). A complete discussion of the use of EM for estimating dynamic texture mixtures is described by Chan et al. (Chan & Vasconcelos 2008). The key concept is that a Kalman smoothing filter (Grewal & Andrews 1993) is applied for the “E step” in order to estimate the hidden state \( X^i \) given observations \( Y^i \) and assignment \( z = j \), which in turn allows one to directly compute the state covariance matrix \( P_k \) as well as the posterior assignment probability \( p(z = j|Y^{(i)}) \). This allows for the “M step” in which new model parameters are updated for each mixture component.

In summary, EM applied to a dynamic texture proceeds as follows (Chan & Vasconcelos 2008):

1. Choose a set of observations \( \{Y^i\}_{i=1}^{N} \), the maximum number of mixture components \( K \) and the state size \( n \).
2. Initialize model parameters \( \{\phi_j, \alpha_j\} \) for each \( j \).
3. E-step: for each observation \( i = 1, \ldots, N \) and component \( j = (1, \ldots, K) \), input \( Y^i \) and \( \{\phi_j, \alpha_j\} \) to a Kalman filter in order to estimate hidden state \( X^i \), \( P_j \) and \( p(z = j|Y^{(i)}) \).
4. M-step: for each component \( j = (1, \ldots, K) \) use these estimates to compute updated model parameters \( \{\phi_j, \alpha_j\} \).
5. Iteratively repeat the E and M steps until log-likelihood \( \log(p(Y^i)) \) converges.
6. Output model parameters $\{\phi_j, \alpha_j\}_{j=1}^K$

These output parameters may then be compared to features derived from other input test sounds, in order to determine if these belong to one or a number of the learned mixture classes.

3.6 Application: Designing an EMD-LDS Recognition System

As noted, previous applications of dynamic textures have focused on musical song segmentation or clustering and segmenting video. The former utilized a sliding window of audio features, while the latter focused their effort on spatial sampling of a small sub-grids of pixels across a given video source. By contrast, we sample across the “space” defined by a set of intrinsic mode functions (IMFs) created through EMD analysis while also capturing this representation on a moving window basis. In this way we may build our model more directly on the unique IMF functions – each having unique spectro-temporal placement and regularity – where the challenge then becomes articulating this through proper feature analysis.

3.6.1 Defining the Time-Frequency Feature Space

Recall the set of IMFs presented in Fig. 1 for a recording of carbonated soda that fizzes after being poured. The lowest IMF captures a sharp transient while higher modes move from a finer to more coarse level of temporal detail, with decreasing total energy. Depending on the temporal profile, density and magnitude of transients, one might encounter the problem of mode mixing, wherein information is spread across several IMFs, resulting in a less physically-meaningful set of data. We have found that, as suggested by Wu and Huang (Wu & Huang 2004), taking an ensemble-averaged approach greatly alleviates this problem. This consists in defining a “noise-assisted” set of IMFs that are redefined as the mean of a number of trials which result from noisy measurements. In practice, this is achieved by adding a small
amount of unique Gaussian white noise to the signal for each trial, and taking the EMD at each step. The final EMD is achieved by averaging across all trials: the effects of the zero mean, uncorrelated random process are cancelled out while the IMFs are smoothed, thereby reducing or completely removing any mode mixing. We have found that using a noise perturbation with amplitude of 0.2 times the RMS of the given input signal works well, along with applying 100 trials over which to take the ensemble average.

A second challenge in defining the feature space stems from the fact that EMD is signal-adaptive, and so each source will produce a varying number of IMF functions. While the number depends on the specific stopping criteria for the sifting process as well as the signal, through extensive exploration we have found that the maximum IMF size generally does not exceed 20 modes, rarely reaching this number. The question then becomes the point at which IMFs no longer produce information relevant to a sound texture signal definition. Certainly finer details of the lowest modes contribute to micro-transients, but longer quasi-periodic fluctuations are also relevant. Any criterion for choice of mode size to use must consider the temporal scale for modulations in the higher modes as well the amount of energy present in these higher modes (given that the overall level falls off as IMF number increases). For example, in Fig. 6 we see IMF modes 5-8 for the fizzing soda audio that was displayed in Fig. 1. From visual inspection it would appear that these signals contribute unique modulations with a fast enough rate and enough energy to contribute meaningfully to texture definition – as opposed to long-term modulations such as repeated rhythmic structures (Heydarian & Reiss 2007). In order to prove this in a more quantitative fashion we conducted an analysis of IMF vector size as a function of other model parameters, as well as a function of different audio feature extraction techniques. This will be presented in the evaluation section (VII).

3.6.2 Salient Feature Extraction

Figures 1 through 3 suggest a unique temporal and spectral profile for the micro-transient texture of fizzing soda. Other textures, such as the din of a crowd or traffic sound, may have a more uniform temporal profile with the distinguishing characteristic appearing in the spectral profile across IMF functions. In the face of such a disparate class of sounds, we need to extract features that are sufficiently broad while complementing these two sub-classes (that is, micro-transient heavy sounds vs. sounds comprised of an assemblage of some atomic “background” events). Further, the information contained in each IMF represents a dual amplitude and frequency modulated component, and so an exploration of temporal as well as spectral features is appropriate. Our research has explored the hypothesis that EMD itself goes a long way to represent both of these aforementioned sub-classes, as they are characterized by a quasi-repeating yet stochastic fine structure that this technique can decouple from the global profile quite well. In order to augment this analysis with higher-level information, we conducted a set of analyses on the individual IMF functions, resulting in a vector of information for each moment in time. The features chosen were drawn from the literature on music and speech timbre perception, with an emphasis on those most relevant to AM/FM variations. In addition to the features roughness (Daniel & Weber 1997a) (calculated using the Psysound Matlab package (Densil Cabrera & Schubert 2007)), as well as spectral flux (McAdams 1999), spectral variability and compactness as implemented by McEnnis et. al. (Daniel McEnnis and Cory Mckay and Ichiro Fujinaga and Philippe Depalle
2005), which led to lower recognition results during our testing, we now discuss the most relevant set of audio features. In each case, note that \( X_p[i] = 20 \log_{10}(X[i]) \), where \( X \) is the Fourier Magnitude Spectrum of input signal \( x \) and \( i \) is the bin index number.

### 3.6.3 Spectral Centroid (SC)

This classic feature describes the relative balance of the spectral envelope (Peeters 2004) and is strongly tied to musical timbre perception when calculated on harmonic input signals (McAdams 1999). For each frame it is defined as

\[
SC = \frac{\sum_{i=0}^{N-1} i(X_p[i])}{\sum_{i=0}^{N-1} (X_p[i])}
\]

where \( X_p[i] \) is the square value of the magnitude spectrum at bin \( i \), computed using an FFT of size \( N \).

### 3.6.4 Spectral Spread (SS)

This feature describes the variance in the spectrum (Peeters 2004), or in other words the extent to which it occupies the spectrum, around the centroid. It is also related to musical timbre perception in the case of harmonic input signals (McAdams 1999). It is defined as:

\[
SS = \sqrt{\frac{\sum_{i=0}^{N-1} X_p[i](i - SC)^2}{\sum_{i=0}^{N-1} (X_p[i])}}
\]

### 3.6.5 Spectral Flatness (SF)

The spectral flatness (Peeters 2004) describes the general peakedness vs. line-likeness of the spectrum, and is a classic feature from speech processing. It relates to noisiness as well as density of transients. It is defined as

\[
SF = \frac{G}{A}
\]

where

\[
G = \left( \prod_{i=0}^{N-1} X_p[i] \right)^{\frac{1}{N}} \quad A = \frac{1}{N} \sum_{i=0}^{N-1} X_p[i]
\]

are the geometric and arithmetic means of the spectrum, respectively. This measure is constructed so that it is close to 0 for tonal sounds and signals with line spectra, and close to 1 for white noise.

### 3.6.6 Temporal Fine Structure (TFS)

This measure was constructed by the first author (Van Nort 2009a) precisely for the task of measuring micro-transient fluctuations in the class of texture signals. It is defined as
\[ TFS = \frac{\sum_{i=0}^{N-1} (x_{p}[n] - \frac{(x_{p}[n+1]+x_{p}[n]+x_{p}[n-1])}{3})^2}{R} \]  

(11)

where \( R \) represents the overall root-mean-square amplitude of the given input frame, and \( x_{p}[n] \) is the squared-amplitude of the signal at time \( n \). Intuitively, this measure is aimed at removing the influence of the overall trend, and examining only local time-variations in the signal.

### 3.6.7 Mel-Frequency Cepstral Coefficients (MFCCs)

This feature produces a Fourier-based perceptual vector that is widely-used in the speech and music recognition communities (Logan 2000), MFCCs efficiently describes the spectral envelope with a small vector of features. This method was further used in the only other known work on dynamic texture modeling of audio signals (Barrington et al. 2010), and so it is included here for comparison as a benchmark, as the lone non-EMD based technique that uses dynamic texture modeling.

In addition to the above features, we examined larger vectors comprised of pairwise combinations of SC, SS, SF and TFS, in the event that such an augmented feature space would yield additional information that would capture qualities which differed across IMF functions.

### 3.6.8 Modeling and Recognition

In the process of learning model parameters, a set of \( N \) observation sequences for a given sound source are taken by extracting audio frames of size \( T \). The EMD technique is run in order to extract IMFs, the lowest \( L \) of which are preserved (again, for comparison purposes, MFCC was tested as well). Note that the number of expected mixture components \( K \) is not tied to the number of IMFs, and instead is a free variable which is a product of the application and context. In general, if the number of unique classes within the data are known, this would be the value one sets for \( K \). Once this maximum mixture index is chosen, the EM algorithm is iterated over each observation vector, defined by one of the aforementioned feature measures. Therefore, each of the \( N \) observations can be thought of as a matrix of values whose column is a multiple of \( L \), taken over a temporal frame of size \( T \). At the end of the learning process, we are left with an array of model parameters that correspond to each potential dynamic texture mixture. For each input sequence \( Y^{(i)} \), the index \( j \) that satisfies

\[
\arg \max_j \frac{\alpha_j p(Y^{(i)}|\phi_j)}{\sum_{k=1}^{K} \alpha_k p(Y^{(i)}|\phi_k)}
\]

(12)

will map the given \( i \)th fragment of audio to the most likely mixture component. Therefore, at the conclusion of this learning process the input audio can be mapped to a given class from the database by computing likelihoods from equation 12 for each set of learned model parameters, which can be achieved efficiently using the Kalman filter from the EM algorithm.

The overall system diagram which presents analysis, feature extraction, modeling training data and recognizing test input is show in Fig. 7.
3.7 Evaluation

We conducted a systematic analysis of the combined EMD-LDS system for each sound feature, using a wide array of sound sources that describe the two sub-types of texture that we have articulated: sharp micro-transients as well as sound comprised of an assemblage of uniform and brief background sources. Our specific choice of audio data was influenced by the fact that the growing literature on sound texture is sorely lacking in a standardized body of data (something that is very present in the visual texture research). In order for a more coherent comparison, we utilized the same sound types that were employed in the study presented by Athineos and Ellis (Athineos & Ellis 2003). This consisted of five classes of sounds: crowd applause, fizzing/carbonated beverages, fast strokes of a typewriter in use, a rainstorm and crackling fire. The actual sound files from the authors’ site (n.d.) were utilized, with the rest coming from the Freesound database (Freesound n.d.). As this latter source is the result of different users uploading audio content, this guaranteed that the audio sources would arise from disparate recording contexts. A minimum of five different recording sources were used for each texture type, with a minimum sample size of 12 seconds (giving rise to at least 200 examples for each source). Each audio sample was adjusted to a sampling rate of 22.05 kHz.

The tests were all conducted using cross-fold validation, with half of the data used for training and the other half for testing. For each half, the sound files within a given class (i.e. texture type) were mixed between the different sample sources in order to provide a wider variety of training and testing data. The training set was grouped by input class and the various sources concatenated together in one long file, with each individual input training sequence of audio being selected in a running overlapped fashion. As a result, each training sample contained time-shifted elements of audio from the previous example. This technique
Table 1: Data Settings, including feature type, observation size (M), frame (F), hop (h) and state (n).

<table>
<thead>
<tr>
<th>Feature(s)</th>
<th>M</th>
<th>F</th>
<th>h</th>
<th>n</th>
</tr>
</thead>
<tbody>
<tr>
<td>TFS/EMD</td>
<td>2-13</td>
<td>100, 200, 250, 500</td>
<td>2, 10, 25, 50</td>
<td>2-6</td>
</tr>
<tr>
<td>SC/EMD</td>
<td>2-13</td>
<td>100, 200, 250, 500</td>
<td>2, 10, 25, 50</td>
<td>2-6</td>
</tr>
<tr>
<td>SS/EMD</td>
<td>2-13</td>
<td>100, 200, 250, 500</td>
<td>2, 10, 25, 50</td>
<td>2-6</td>
</tr>
<tr>
<td>MFCC</td>
<td>2-13</td>
<td>100, 200, 250, 500</td>
<td>2, 10, 25, 50</td>
<td>2-6</td>
</tr>
<tr>
<td>Pairwise</td>
<td>4-26</td>
<td>100, 200, 250, 500</td>
<td>2, 10, 25, 50</td>
<td>2-6</td>
</tr>
</tbody>
</table>

was used by Barrington et al. (Barrington et al. 2010) for the purpose of song segment boundary detection, and is well-suited to an eventual application of real-time recognition. Concatenating the different classes together into one long audio feature vector poses more of a challenge to the system in the case of examples that were comprised of values that spanned across the boundary between groups. The feature extraction analysis techniques were all standardized so that an input vector of 1024 samples was taken for each, with an overlap of 128 samples. In order to map out the training requirements, we examined frame sizes $F$ that ranged from 100 to 500, as well as hop sizes $h$ from 2 up to 50. This corresponds to training examples that range from 0.5 to 2.5 seconds of audio data, with steps from 0.01 to 0.3 seconds. Further, as there is no analytic framework for choice of IMF size, we examined mode numbers ranging from 2 (the minimum needed for the underlying state-space model) up to 13 (out of a maximum 18 mode functions). For comparison, we also examined the same number of components from the MFCC analysis (note that 13 is a common vector size, and is further used by Barrington et al. (Barrington et al. 2010)). Therefore this choice of the number of IMFs (or number of MFCCs in the case of the comparison test) becomes the observation size $M$ for the LDS model, except in the case of multiple features where $M$ would of course be a multiple of the IMF size.

### 3.7.1 Group-level Clustering

Using data drawn from each of the five given categories, we attempted to see how well the algorithm could accurately cluster an input set of data for mixture size $K=5$. As noted in table 1, training and testing was applied to the respective data sources for observation sizes ranging from $M = 2...13$, except in the case where pairwise features were used, wherein even-numbered observation sizes ranging from $M = 4...26$ were tested. In order to examine the size of temporal fragments that were best suited for modeling, input frame sizes of 100, 200, 350 and 500 were examined, as well as hop sizes of 2, 10, 25 and 50. Aside from the choice of $K$, another free parameter in the modeling step is the presumed size of the underlying state, $n$. We further examined states sizes for $n = 2...6$.

In order to measure the ability to properly cluster all five classes in each situation, we examined the output class assignment for each test input. Looking at each test member for a given texture type, the mode was calculated, and in the event that all five possible output classes appeared as the mode precisely once (one for each input texture type), the trial was considered a success. In other words, if the trial properly spanned all five possible output classes it was deemed successful and counted towards the overall clustering score. This group
cluster score was counted across all 12 observation sizes arriving at a score ranging from 0 to 12 for each combination of $F$, $h$ and feature type, with the most successful settings illustrated in Fig. 8.

Note that the use of the TFS feature stands out as having the highest score, with all 12 observations sizes (i.e., TFS measure based on all IMF functions from 2 to 13) resulting in a proper clustering. This happened for $F = 100$ and both for $h = 2$ and 10. Perhaps most interesting is that the clustering was successful with as few as 2 IMF modes being used for observation, as every other method employed needed at least three and generally four IMF functions for a proper clustering. A number of SC-based settings did well in this experiment, making TFS and SC by far the most successful of all features. The combination of these features were the only one with multiple successful trials, though this did not improve on the performance of the individual features. This itself is worthy of note: both a temporal and a spectral feature extraction of IMF functions proved successful, though a combined spectro-temporal approach did not prove more useful in any appreciable way. Spectral Spread illustrated a few minor successes, while roughness did not fully cluster in any situation and spectral flux performed considerably more poorly than the other features in question. The use of MFCCs provided a full clustering of all five input classes in precisely one case ($F = 100, h = 10, M = 4$). It is included in the chart to illustrate its relative poor performance as compared to TFS, SC and SS.

In addition to the predominance of TFS and SC, note that the majority of trials are for a frame size of 100, corresponding to a time frame of 0.58 seconds. Given the relatively large-scale (when compared to timbral attributes) and quasi-periodic nature of texture phenomenon it is surprising that more time would not somehow yield better results. The analysis also benefitted from a finer hop size, with smaller values leading to better results in general. In terms of cross-group confusion amongst the less successful results, the majority of trials that did not fully cluster for TFS, SC, SS and MFCC were able to map into four classes, with the confusion existing between applause and rain in nearly every case. If we examine the output cluster size in these confused examples, suddenly TFS distinguishes itself further as a successful technique: the number of classes was either four or five in every instance, while every other technique exhibited more confusion, mapping into 3 or 2 classes, or even 1 large output class in some instances.

3.7.2 Measuring Quality of Recognition

While the above represents the overall ability to successfully map the five input texture types into five output classes, this says little about how cleanly each individual group clustered together uniformly. For example, while the combined TFS/SC system performed well overall, there was one setting that exhibited an odd trend: for the case of of $M=4, F=200, h=10$ and $n=2$, the system spanned all 5 clusters, yet only 44% of input sequences properly mapped to a coherent and singular value for a given input group. Alternatively, it might be the case that all members cleanly clustered together in a single class, at the expense of accurately separating out the five given classes. This is illustrated in Fig. 9, which shows an example based on TFS in which 93% of all sequences cluster together for a given input class, yet not all five classes are properly differentiated as the applause and rain sets are perfectly grouped together. Note also that the remaining errors arise in the boundary between groups – a
product of our sliding-window approach to learning. If training examples that spanned this boundary were avoided, then overall quality measures would be significantly higher – but we maintain this approach as we anticipate using this system for real-time learning in the near future.

Therefore, in order to provide an overall measure of quality for the recognition system, the grouping tendency within an input class must also be measured. To this end, we calculated the number of values that were mapped to the output class mode for a given input texture type, and then averaged this value across all five texture groups to get the overall measure. By taking the aforementioned group-cluster score in conjunction with this quality measure paints an overall picture of the recognition system. Taking the 11 system settings from Fig. 8 that produced a proper five-class clustering, we averaged their overall quality measure across all successful $M$ values with these numbers listed in table 2. Also listed are the observation sizes that contributed to the overall value, which gives a sense of the overall effectiveness of the given system and the extent to which any special tuning is required.

From the table we can see that the TFS-based system does quite well in terms of recognition quality, working 90% of the time on average across all IMF sizes in the best case, and 88% in the worst. An example is given in Fig. 10 that shows a TFS-based system setting ($F = 100$, $h = 10$) that produced 90% accuracy while properly spanning all five output classes. The quality is much higher than .90 in all cases except rain, which again maintains some confusion with the applause class. The SC-based system does quite well itself, ranging from .83 to .86. Note that the combined TFS/SC system performs precisely the same as for SC given the same model settings, suggesting that the SC measure dominates the modeling process and that, again, a combined system fares no better. The SS system was very consistent, but not as good overall as the others on the group. The MFCC-based system

Figure 8: Number of different observation sizes that produced all five output classes upon clustering; The label represents analysis type, frame size (F) and hop size (h). In each case state size ($n$) is 2.
Table 2: Most successful recognition quality values including weighted score (W), included range of observations (M), frame (F), hop (h) and state sizes (n) that achieved this score.

<table>
<thead>
<tr>
<th>Feature(s)</th>
<th>W</th>
<th>M</th>
<th>F</th>
<th>h</th>
<th>n</th>
</tr>
</thead>
<tbody>
<tr>
<td>TFS/EMD</td>
<td>.90</td>
<td>2-13</td>
<td>100</td>
<td>10</td>
<td>2-4</td>
</tr>
<tr>
<td>TFS/EMD</td>
<td>.90</td>
<td>2-13</td>
<td>100</td>
<td>2</td>
<td>2-4</td>
</tr>
<tr>
<td>TFS/EMD</td>
<td>.88</td>
<td>4-10,13</td>
<td>100</td>
<td>25</td>
<td>2-4</td>
</tr>
<tr>
<td>SC/EMD</td>
<td>.86</td>
<td>5-13</td>
<td>100</td>
<td>10</td>
<td>2</td>
</tr>
<tr>
<td>SC/EMD</td>
<td>.85</td>
<td>5-13</td>
<td>100</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>SC/EMD</td>
<td>.84</td>
<td>5-11</td>
<td>100</td>
<td>25</td>
<td>2</td>
</tr>
<tr>
<td>SC/EMD</td>
<td>.83</td>
<td>5-8</td>
<td>100</td>
<td>50</td>
<td>2</td>
</tr>
<tr>
<td>SS/EMD (peak)</td>
<td>.83</td>
<td>10-20</td>
<td>200</td>
<td>50</td>
<td>2</td>
</tr>
<tr>
<td>SS/EMD</td>
<td>.77</td>
<td>4</td>
<td>100</td>
<td>10</td>
<td>2</td>
</tr>
<tr>
<td>SS/EMD</td>
<td>.66</td>
<td>5-8</td>
<td>100</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>SS/EMD</td>
<td>.66</td>
<td>5-8</td>
<td>100</td>
<td>25</td>
<td>2</td>
</tr>
</tbody>
</table>

gave considerably weaker results than TFS or SC, yet was ahead of the averaged SS value. Having said this, note that the MFCC-based system only gave a proper five-class clustering in one spurious setting. Therefore, this system cannot be considered as reliable as even the SS system, let alone the much stronger SC and TFS-based ones. For a fairer comparison, the best quality value was taken from the SS data, which performs better than this singular MFCC value. To further understand the flexibility of the TFS/EMD-based system, note that the quality measure was calculated across state values ranging from \( n = 2 \) to \( 4 \). This was the only system whose performance did not fall off drastically and immediately once the state size was increased (where this did not occur until \( n = 5 \)), suggesting that this method captures more complexity in terms of degrees of freedom within the underlying LDS model.

3.7.3 Comparison to GMM

As a second benchmark for the recognition system, and in keeping consistent with the comparison presented by Barrington et al. (Barrington et al. 2010), we ran the aforementioned GMM clustering algorithm. This method was run on the same data set, for the same example inputs and observation sizes (there are no state or hop sizes to speak of with this technique). The result is that the clustering and recognition was much poorer than any of the EMD-LDS texture models. None of the settings achieved a correct clustering across all five classes, meaning that every instance of the GMM technique failed to perform near the level of the above systems. In the case of SS, SF and SC as input data for the GMM, all output cluster sizes were 1 or 2, while MFCC-based input resulted in only a handful of cases that spanned four output classes, with a quality measure of .60. The TFS was only marginally better, with a handful of 4-class instances and a quality measure of 0.63.
3.8 Conclusion

Through an examination of a number of parallel systems based on EMD and dynamic texture modeling, and through comparison to other work, it is clear that the TFS/EMD and dynamic texture model hybrid is a successful texture classification system for the classes that we have examined, by proving its ability to distinguish between the five input types with a recognition close to 90% in the worst case and much better in most instances. Certainly restricting the study to these five classes of sounds does not allow any generalization to all possible sound texture types, as this is simply too expansive and general of a category of signals to make such a claim. At the same time, The given classes of sound under consideration here – applause, rainstorms, fizzing soda, typewriters and crackling fire – are very exemplary of sound textures with their qualities such as dense micro-transients, quasi-repetitive local events and stochastic local detail with a stable global spectro-temporal profile. This along with the fact that the system was trained over hundreds of input fragments from different
recordings speaks to its viability for sound texture modeling.

Further, we have built upon the existing studies related to dynamic texture mixture (DTM) modeling by illustrating the superiority of a hybrid EMD/DTM system that uses appropriate feature extraction, showing that this technique can indeed work at a fine temporal level while accepting data in a running-window fashion. This demonstrates that these methods are well-suited to a variety of applications where texture and noisy musical or environmental sounds need to be recognized, classified and segmented in real time. Our hope is that this algorithm can inspire new directions in the field of sound texture analysis and modeling, drawing on our unique synthesis of the literature in sound and visual texture analysis, and the synthesis of nonlinear time/frequency analysis with linear dynamical systems modeling.

3.9 Notes

This paper is an expansion, with many additional results, of the work entitled “Sound Texture Analysis based on a Dynamical Systems Model and Empirical Mode Decomposition” which was presented at the 129th AES convention in November 2010.
4 Mapping to musical actions in the FILTER system

Doug Van Nort, Jonas Braasch, Pauline Oliveros

4.1 Introduction

Our group, along with researchers in acoustics and cognitive science, have undertaken a project dubbed CAIRA, which stands for the creative, artificially-intuitive and reasoning agent (Braasch, Bringsjord, Kuebler, Oliveros, Parks & Van Nort 2011). This project is devoted to understanding and modeling machine performance from both a top-down, logic-based point of view well-suited to following rules as well as from a more bottom-up and intuitive approach to machine improvisation. Created within this context, and stemming from an earlier pilot project with the same mission, is the Freely Improvising, Learning and Transforming Evolutionary Recombination (FILTER) system. This system places an emphasis on three key concepts: an embodied approach to machine listening, on intuitive and spontaneous transformations of a human performer’s input in a way that is shaped by learning stylistic trends, and finally the system’s actions are informed by an electroacoustic aesthetic that favors a sound-oriented view on performance output rather than one determined by music theoretic rules. Defining the space of possible musical actions for FILTER has been a fluid and integral part of the design process; the test-bed for this work is our trio Triple Point (Van Nort 2010b). The details of the FILTER system related to machine listening and learning are discussed elsewhere (Van Nort, Oliveros & Braasch 2010), while here we focus on the design of the mapping to musical actions that results from continued use in improvisational sessions. A recent piece (Van Nort et al. n.d.) is presented as an example of a musical context for which the system was adapted in response to particular performance demands.

4.2 Motivation and Context

The instrumentation for our improvisational trio Triple Point spans the spectrum of acoustic (soprano saxophone), acoustic modeling (Roland V-Accordion) and digital transformations based on analysis/resynthesis (GREIS system (Van Nort, Oliveros & Braasch 2010)). Through extended technique (saxophone), changing synthesis timbres (V-accordion) or by on-the-fly transformations (GREIS) our style is one in which sources can quickly fuse into a single element or conversely spread into unique, disjointed lines. Playing with source and instrumental identity in a manner that is dynamic and controllable is an important part of our music. Reflecting on the role of the GREIS player as one who listens for distinct lines of musical intention in the sound streams before capturing and transforming these, the design of FILTER was motivated to model this performance practice, adding to the richness of the musical interaction and allowing the possibility of expanding the group into a quartet. In addition to the GREIS system, the Expanded Instrument System (EIS) (Van Nort, Oliveros

---

& Braasch 2010) is taken as a point of inspiration, with its focus on acting as a reactive, mirroring partner that re-presents past sound in a surprising manner in performance. Much like GREIS and EIS, FILTER has proven to be a unique partner that lends its own style and sonic character to our performance endeavors.

4.3 FILTER Overview

The FILTER system was designed to move beyond the notion of extending a performer’s actions through time, towards a system that learns information that is embedded in the low-level structure of the audio stream of its improvising partner. While GREIS and EIS both have a running memory in the form of a recording of the past \( N \) seconds of performed audio on which to make decisions, FILTER encodes not only the waveform but also in parallel the fine structure level of information about the temporal evolution of sound features. In parallel with this low-level information the system catalogs a set of sonic gestures which give semantic meaning to performance actions at the note and phrase level. The details of listening and learning are discussed elsewhere (Van Nort, Oliveros & Braasch 2010), though a brief overview is required in order to understand the parameters that result from this stage as they are mapped to and directly determine the output musical actions.

4.3.1 Listening, Learning

The structuring principles for this aspect of the system are that of listening to gestures and textures. The former can be thought of as foreground actions that have a coherent motion in regards to spectrotemporal parameters, while the latter is the characterization of the overall sound field over a larger time duration (e.g. larger than 5 seconds), in the absence of coherent motion. Listening for and recognizing gestures is based on continuous gesture following (Bevilacqua et al. 2010) applied to a small set of sound features. In contrast to many applications of gesture following applied to sound streams, in FILTER this is based on unsupervised learning as follows: when a transient is detected in amplitude or fundamental frequency, subject to an inter-onset temporal threshold, a new “sonic gesture” is considered to have begun. If this gesture is dissimilar to anything in the current “gesture space” then this may be added as a new member of the space, with an older one possibly being discarded. This aspect of learning can be thought of as the system developing a semantic memory and deciding which are the relevant sonic gestures on the fly, in the non-idiomatic spirit of free improvisation. The output of this process is a continuous likelihood that a given action is related to one of the sonic gestures in the given space, thereby providing a continuous degree of certainty that the system is hearing those sonic gestures that have been internalized within the performance moment.

This learning stage is out-of-time in the sense that each gesture is committed to FILTER’s semantic memory without any temporal ordering between gestures. In parallel with this, the underlying audio’s temporal structure is learned in order to provide an understanding of the temporal regularity and similarity across an entire performance. This episodic memory component of the system is inspired by the audio oracle concept (Dubnov et al. 2007) that underlies the OMax system (Assayag et al. 2006), and FILTER utilizes this project’s Max/MSP implementation of the factor oracle algorithm. From experience we
recognize and leverage the power of this learning algorithm, but also see the shortcoming of requiring a human operator to make contextual choices when using this information for musical playback. We construct the semantic-level gesture/texture listening layer and use this to drive output actions to explicitly address this issue. The episodic and semantic levels of information in FILTER are associated by endowing relevant states with time-stamps for any associated sonic gestures in memory, as well as segment boundaries between areas that are considered to have different textural qualities (Van Nort, Braasch & Oliveros 2010).

4.3.2 Evolving, Mapping

The use of analysis, recognition and structure-learning in FILTER is not towards the end of categorically classifying performer actions in an out-of-time fashion, but rather the goal is to leverage the process of recognition and understanding as it develops, and so the system focuses more on anticipation than on recognition. As between human improvisers it is the feedback and reinforcement of certain sounds or passages, guided by moment-to-moment anticipation, that give them meaning within a given performance context. Further, the design of FILTER is predicated on the idea that this continuous recognition should not lead to output that always exists within the same parameter space, but that this should shift over time in a way that favors novelty and challenges a human performer, though not so erratically that it is perceived as random – a balance of spontaneity and the desire for dialogue. With this in mind, the system output is partially governed by an evolutionary process that acts as mediating layer between the learning elements and the space of output parameters. Building on a previous project (Van Nort et al. 2009), FILTER maintains a notion of continuous, dynamic attention that is tied to a measure of confidence in the gesture listening, defined as:

$$C_n = \delta(m_n - m_{n-1}) \sum_{k=0}^{n} 2^{\frac{-1}{\lambda_k}} (m_k d_k)$$

where $m_k$ represents the value of the maximal likelihood for the $k$th gesture in the gesture space and $d_k$ is the deviation from the average value, also for the $k$th gesture. The binary function $\delta$ is present so that if there is a sudden change in gestural probabilities, the confidence value is zeroed before again rising. This allows the system to follow stable gestures, but also to adapt to a perceived sudden change in musical direction. The smoothness of the confidence measure is tunable by the values $\lambda_k$. In order to allow FILTER to move towards a globally predictable direction while maintaining random elements on a local scale, a genetic algorithm (GA) is used as a layer between gestural listening and the space of possible behaviors. What sets this usage apart from many projects related to evolutionary music (Miranda & Biles 2007) is that this is not an interactive GA implementation wherein the user explicitly rates the goodness of each output – which is a substantial time and attention bottleneck. Rather, the fitness is directly tied to the saliency of the gestural recognition process by mapping the smoothed gesture-likelihood values into the fitness of a member of the GA pool, while the confidence is inversely proportional to the mutation rate. In this way, the “goal” changes as a product of the system’s gestural recognition and confidence. However, if the confidence remains substantially low then the mode of listening for FILTER
changes so that the gesture-based confidence value no longer drives the GA, and instead the texture-based sound features influence the output to system behaviors.

There are two layers of mapping in this part of the system. The first is the association of members of the gesture space, as well as textural categories, into output behaviors. This is achieved by mapping archetypal gestures – defined by their temporal shape or morphology (Peeters & Deruty 2010) – into each member of the GA population on one hand, or an archetypal set of texture features (averaged over a large time window) on the other. This mapping provides a semantic association to begin (e.g. “repeated sharp attacks should give rise to X type of behavior”), which can be thought of as a set of musical values for the system. These ‘value mappings’ are reinforced or lost over time as the parameter space evolves and FILTER is influenced by the style of its improvising partner, though they may be reinstated during the course of performance. The second layer of mapping is the embedding of these GA members in a continuous, higher-dimensional space of possible behaviors through the use of continuous mapping strategies (Van Nort 2009b). In this implementation, a set of N-dimensional population members move within a simplicial complex where each node is associated with a behavior state of the system. The member of the population associated with the currently most salient gesture/texture state is used to interpolate the nodes of the enclosing simplex, determining an output behavior state of the system. This can be thought of as a cloud of possible states that move with a quasi-physical nature as determined by the output of the listening module. In this particular aspect, FILTER shares a similarity with the continuous state-based approach of the Ozone project (Sha et al. 2010). One critical difference is that the current state of the system behavior itself (the so-called “stability” feature) also partially determines the movement in the state-space – a sort of self-reflexivity of the system.

### 4.4 Musical Behaviors

In FILTER, the learned graph-like audio structure is navigated in order to produce sound output. Navigating this structure causes the system to recombine past elements of audio that have varying degree of contextual relevance. By reinterpreting the nature of this “relevance” on the fly and altering the manner of recombination, FILTER moves beyond the aforementioned problem of requiring a human operator. Coupling this with additional sound transformations gives FILTER a set of potential performance re-actions that are conducive to Triple Point’s sound-oriented, free improvisation aesthetic.

The behavior states of the system give a high-level description that is then mapped into musical actions, as well as into internal decision making. As noted in figure 11, this mapping is partially regulated by the relative saliency of gestures (vs. textures) to the current musical context. The continuous behaviors include:

- **Rhythmic-ness**: The likelihood that individual lines will be repeated, as well as the degree of variation within a given repetition.

- **Wildness**: If gesture listening is dominant, the likelihood that the system will mirror the performer by using recent input vs. improvising on disparate regions of past and present input. If texture listening is dominant, the likelihood that the system will draw on past regions that have similar textural sound qualities.
Figure 11: FILTER’s Listening system balances between gesture/texture listening, with the output causing an evolution of members within the interpolated behavior space.

- Stability: The likelihood of possible change in the overall behavior state of the system.
- Sustain: The favoring of sustained vs. short tones or actions.
- Density: If gesture listening is dominant, this affects the size and spacing of output phrases. If texture listening is dominant, the number of overlapping layers of content that are performed at once.

Note that the function of these behaviors changes depending on the listening context. Further, the wildness state determines the likelihood that the system will behave similarly or differently from the player, whether this is “gesturally” (a single, well-defined passage) or “texturally” (layers of quasi-repeated or stretched passages of sound). Therefore the notion of same/different playing has the dual interpretation of drawing on similar/different content, or playing in a similar/different style.

4.4.1 Transformations

In addition to recombining disparate fragments of audio in a manner subject to the given behavior state, FILTER has the ability to time stretch and pitch shift its current musical
output, or each layer individually in the case of denser textural playing. Each of these potential phrases may also be fed into a feedback delay line that is subject to modulation and filtering. These fundamental processes define a variety of musical effects, as determined by the mapping from the higher-level behavior parameters and by which listening context is currently dominant. The degree of similarity in playback style – expressed by the Wildness state – also determines the amount (if any) of pitch shifting, while the degree of sustain influences the amount (if any) of time-stretching applied. The set of transformations defined by feedback and modulation are influenced by both wildness and stability in a cross-coupled fashion.

4.4.2 Spatialization

An integral part of the system’s musical actions is its ability to define spatial gestures that react to the musical context. The system utilizes the virtual microphone control (ViMiC) approach (Braasch et al. 2008), which models sound reflections, dispersion patterns of sound sources and doppler shifts. As such it is very conducive to rapidly moving sound sources around the space in a realistic fashion, where relative positioning between source and speaker output may be controlled. The parameters that are subject to machine control in FILTER include: the set of possible trajectories for each sound source, the reverberation decay time, room size of the spatial model, and the radius, speed and incidence angle of each sound source. These spatial parameters are given equal importance to all other musical actions/transformations in consideration of the overall performance of the system. For example, the textural nature of the output is drastically altered if each improvised line of the system is presented as a different moving source, thereby separating each one spatially. This interaction between spatial gesture and machine actions was the subject of consideration in a recent telematic piece that we presented at last year’s NIME conference, which we now describe as an example application in a real musical context.

4.5 Distributed Composition #1

The presentation of a work involving FILTER and Triple Point is a challenging (yet rewarding) endeavor as it involves multiple complex systems. In Triple Point there is already a sharing of sonic gestures through the capturing and transformation of audio on-the-fly (Van Nort, GREIS) that extend the overall sound scene. Presenting the actions of FILTER so that they exist as a unique contributor to the musical dialogue presents an interesting challenge. In the piece Distributed Composition #1 we embraced this complexity and pushed it further by defining a three-site telematic piece. The title of the piece refers not only to this physical distribution of the human players, but also to the distributed musical cognition between human and machine, as well as the fact that each player had a hand in defining the musical structure – making it a distributed composition in several senses of the word. The FILTER system itself had a hand in composing the structure in that it acted as conductor, determining when a member of the quartet would have the option of playing. This was achieved by adding these cues to the behavior state-space, while an audio matrix determined which input FILTER was improvising on at a given moment. Within these confines, any of the eligible four players were free to improvise. The piece allowed the FILTER system to capture the
GREIS output as well (while both were capturing the remote acoustic players), resulting in a proliferation of certain phrases that were subject to several iterations of musical transformation. The staging and sonic display for all human and machine players was adapted so as to allow for a more coherent musical dialogue in light of this sharing of sources. First, the local and remote human players were presented on stage (see figure 12), and their sound was localized to the stage. At the same time FILTER was only present in the surrounding eight channels of audio. Secondly, the system was populated with a set of musical values such that the particular palette of gestures used by the GREIS player for the piece would lead, with high likelihood, to actions that were considered quite different from the current musical context set by that player. In practice, this often led to the FILTER system performing in a very stable, sustained and spatially distant fashion when the GREIS player was producing sounds that were full of transients. Meanwhile when the system improvised on the content from the acoustic players, the result was often a rapid spatial gesture that moved with a small radius in the center of the space, a “musical value” that was added so as to help subvert the distance one might feel in a telematic presentation such as this.

From the experience of this piece, we feel that intelligent, reactive spatial gestures that are integrated into the musical context are a very fertile area of exploration in the case of telematic performance in particular. Further, our hope is that this piece can serve as an early, novel example in terms of a distributed approach to composition – as a non-hierarchical
mode of engagement and planning between human performers as well as between human and machine performers, where each is potentially located in disparate regions of the planet.

4.6 Conclusions and Extensions

FILTER has proven to be a convincing improvising partner, not only in use with Triple Point but in performance with a number of players on bassoon, piano, cello, violin, electronics, various percussion and other instruments. Allowing the system to be flexible in the sense of redefining the high-level mapping to values and behaviors is key to the system’s musicality, as with any human performer wherein one can discuss musical intentions a priori. At the same time, the fact that the design allows for a considered coupling between analysis, recognition and evolved output parameters is an important part of why the system remains convincing and reliable. While the system is under continued development with its current mission and electroacoustic aesthetic in mind, work is currently under way in a parallel project with colleagues in cognitive science (under the CAIRA initiative) in order to explore the result in the case where FILTER is submitted to decision making that is a product of a logic-based reasoning module which acts on long-term information, related to musical tension.
5 A creative artificially-intuitive and reasoning agent in the context of live music improvisation\(^3\)

Jonas Braasch, Doug Van Nort, Pauline Oliveros, Selmer Bringsjord, Naveen Sundar Govindarajulu, Colin Kuebler, Anthony Parks

Abstract

This paper reports on the architecture and performance of a creative artificially-intuitive and reasoning agent (CAIRA) as an improviser and conductor for improvised avant-garde music. The agent’s listening skills are based on a music recognition system that simulates the human auditory periphery to perform an Auditory Scene Analysis (ASA). Its simulation of cognitive processes includes a cognitive calculus for reasoning and decision-making using logic based-reasoning. The agent is evaluated in live sessions with music ensembles.

5.1 Introduction

Numerous attempts have been made to design machine improvisation/composition algorithms to generate music material in the context of various musical styles (Cope 1987, Friberg 1991, Widmer 1992, Jacob 1996). While Oliveros’ Expanded Instrument System (EIS) acts on audio signals (Oliveros & Panaiotis 1991, Gamper & Oliveros 1998), in most cases these algorithms use a symbolic language, such as the Musical Instrument Digital Interface (MIDI) format, to code various music parameters. For example, Lewis’ Voyager system (Lewis 2000) and Pachet’s Continuator (Pachet 2004) work with MIDI data in order to interact with an individual performer. The system transforms and enhances the material of the human performer by generating new material from the received MIDI code, which may be derived from an acoustical sound source using an audio-to-MIDI converter. (Typically these systems fail if more than one musical instrument is included in the acoustic signal). In the case of the Continuator, a learning algorithm based on a Hidden Markov Model (HMM) helps the system to copy the musical style of the human performer.

Following these traditions, this paper describes an intelligent agent that was developed to perform music improvisations in the context of free music. In order to cope with free music, the agent simulates human listening using standard techniques of Computational Auditory Scene Analysis (CASA), including pitch perception, tracking of rhythmical structures, and timbre and texture recognition (see Fig. 13). It uses a Hidden Markov Model (HMM) based approach to recognize musical gestures and Evolutionary Algorithms to create new material. Recently, the authors have begun to integrate a logic-based reasoning system into the overall architecture for a hypothesis-driven approach (see top-down processes in Fig.13).

The current musical output module of the system consists of presenting audio material that is a processed version of input sound which the agent trains itself on during a given session, or from audio material that has been learned by the agent in a prior live session. The material is analyzed using the HMM machine listening tools and CASA modules, restructured through the evolutionary algorithms and then presented in the context of what is being played live by the other musicians. Alternatively, the agent can also conduct a small ensemble using a graphic score and instructions that are updated live. In the following three sections, the basic architecture of CAIRA will be described, followed by a concrete performance example in Section 5.5.

### 5.2 Microphone-Aided Computational Auditory Scene Analysis (MaCASA)

Stemming from the seminal work of Albert Bregman on the perceptual organization and grouping of sounds (Bregman 1990), a body of work has arisen whose primary goal is the computational modeling of the mechanisms by which humans parse audio streams, grouping percepts into identifiable sound objects. This field of Computational Auditory Scene Analysis (CASA) (Ellis 1996, Rosenthal & Okuno 1998) shares similar goals with the analysis stage of our project for several reasons. Primary among these is that contemporary improvised music often does not structure itself by classical paradigms of musical structure – key, meter, melodic progressions, etc. – but rather by working on the level of sound in a more direct, low-level (from a signal processing point of view) and visceral way.

One of the unsolved challenges in CASA is the robust separation of auditory streams from a complex sound mixture. Unfortunately, sound mixtures of music performances are among the most complex cases, and the typically long reverberation times in concert venues are an additional obstacle for robust CASA performance. To circumvent this problem, we separate the individual instruments electro-acoustically using closely positioned microphones for each participating musician. Additional room microphones can be utilized to automatically calibrate the individual microphone signal levels [e.g., see (Braasch, Peters, Van Nort, Oliveros & Chafe 2011)], which is important when the inter-musician relationships need to be determined from these data.

### 5.3 Gestalt-Based Improvisation Model Based On Intuitive Listening

The artificially-intuitive listening and music performance processes are simulated using the Freely Improvising, Learning and Transforming Evolutionary Recombination system (FILTER) (Van Nort et al. 2009, Van Nort 2010, Van Nort et al. 2012). FILTER uses a Hidden Markov Model (HMM) for sonic gesture recognition and Genetic Algorithms (GA) for the creation of new sonic material. In the first step of this stage, the system continually extracts spectral and temporal sound features. At the same time, onsets and offsets are tracked on a filtered version of the signal, which act as discrete cues for the system to begin recognizing sonic gestures. When such a cue is received, a set of parallel Hidden Markov Model (HMM) based gesture recognizers follow the audio, with the specific number of these being chosen...
as a product of needed resolution as well as processing power. The recognition continually provides a vector of probabilities relative to a “dictionary” of reference gestures. Processing on this vector extracts features related to maximum likelihood and confidence, and this information drives the fitness, crossover, mutation and evolution rate of a GA process acting on the parameter output space (Van Nort et al. 2009).

5.4 Logic-Based Reasoning Driven World Model

5.4.1 Overview

In order to better understand the relationship between bottom-up and top-down mechanisms of creativity, a knowledge-based top-down model complements the bottom-up stages that were described in the previous two sections. CAIRA’s knowledge-based system is described using first-order logic notation (for a detailed description of CAIRA’s ontology see Braasch, Bringsjord, Kuebler, Oliveros, Parks & Van Nort (2011)). For example CAIRA knows that every musician has an associated time-varying dynamic level in seven ascending values from tacit to ff. The agent also possesses some fundamental knowledge of music structure recognition based on jazz music practice. It knows what a solo is and understands that musicians take turns in playing solos, while being accompanied by the remaining ensemble. The agent also has a set of beliefs. For example it could be instructed to believe that every soloist should perform exactly one solo per piece.

One of the key analysis parameters for CAIRA is the estimation of the tension arc, which describes the current perceived tension of an improvisation. In this context, the term ‘arc’ is derived from common practice of gradually increasing the tension until the climax of a

Figure 13: Schematic of the creative artificially-intuitive and reasoning agent CAIRA.
performance part is reached and then gradually decreasing tension to end it. Thus, tension often has the shape of an arc over time, but it can also have different time courses. It is noteworthy that we are not focusing here on tonal tension curves that are typically only a few bars long (i.e. demonstrating low tension whenever the tonal structure is resolved and the tonic appears). Instead, we are interested in longer structures, describing a parameter that is also related to Emotional Force (McAdams et al. 2002).

Using the individual microphone signals, the agent tracks the running loudness of each musical instrument using the Dynamic Loudness Model of (Chalupper & Fastl 2002). The Dynamic Loudness Model is based on a fairly complex simulation of the auditory periphery including the simulation of auditory filters and masking effects. In addition, the psychoacoustic parameters of roughness and sharpness are calculated according to Daniel & Weber (1997b) and Zwicker & Fastl (1999). In the current implementation, CAIRA estimates tension arcs for each musician from simulated psychophysical parameters. Based on these perceptual parameters and through its logic capabilities, the system recognizes different configurations for various patterns; e.g., it realizes that one of the musicians is performing an accompanied solo, by noticing that the performer is louder and has a denser texture than the remaining performers. The system can also notice that the tension arc is reaching a climax when all musicians perform denser ensemble textures. CAIRA takes action by either adapting her music performance to the analysis results, or by presenting a dynamic visual score as described in more detail in the next section. CAIRA can, for example, suggest that a performer should end his or her solo, because it is becoming too long or it can encourage another musician to take more initiative. It can guide endings and help an ensemble to fuse its sounds together.

5.4.2 Tension Arc Calculation

In a previous study, we decided to calculate the tension arcs $T$ from a combination of loudness $L$ and roughness data $R$ (Braasch, Bringsjord, Kuebler, Oliveros, Parks & Van Nort 2011):

$$ T = L^4 + a \cdot R^3, $$

(13)

with an adjusting factor $a$. In this paper, we also suggest including information rate (e.g., as defined by Dubnov (2003), Dubnov et al. (2006)) as an additional parameter for the tension arc calculation. A real-time capable solution was developed to measure the rate and range of notes per 2-second time interval. To achieve this, pitch is measured using the YIN algorithm and converted to MIDI note numbers. Next, the number of notes is counted within a 2-second interval, ignoring the repetition of identical notes. The standard deviation of the note sequence is then determined from the list of MIDI note numbers. Finally, the information rate is determined from the product of number of notes and standard deviation of MIDI note numbers. Practically, we measured values between 0 and 100.

In addition, we measured the number of note onsets, by applying an envelope follower, calculating the rate of change of its output signal and then counting the incidents above a given positive threshold. A refractory period of 20 ms was applied, before the next onset is counted to avoid counting the same onsets multiple times. The tension curve is calculated using the following equation:
\[ T = L + 0.5 \cdot ((1 - b) \cdot R + b \cdot I + O)), \]  

with \( I \) the information rate, and \( O \) the onset rate. Note that all parameters, \( L, R, I, O \), are normalized between 0 and 1 and the exponential relationships between the input parameters and \( T \) are also factored into these variables. The parameter \( b \) is the quality factor from the YIN pitch algorithm. A value of one indicates a very tonal signal with a strong strength of pitch, while a value of zero indicates a noisy signal without defined pitch. The parameter is used to trade off roughness and information rate between tonal and noise-like signals.

Figure 14: Schematic communication scheme for a free music performance. Each musician has to establish individual communication channels to all other musicians and also observe oneself. Dashed lines symbolize MaCASA enabled machine listening.

5.4.3 CAIRA’s Self-Observation

Based on the tension arc data, CAIRA assesses the current state of the improvisatory ensemble, thereby addressing questions of who is playing a solo or whether it is likely that the improvisation will come to an end soon. For this purpose, the agent analyses the relationships between the tension arcs of each musician, including the tension arc measured from CAIRA’S own acoustic performance (see Fig. 14). The robust measurement of individual tension arcs is possible, because each musician is captured with a separate microphone.

Figure 15: Video Still from Configured Night.
5.5 Configured Night

An example of the visual score produced by CAIRA was adapted from an audio-visual work titled Configured Night. The core idea of this piece is based on video footage of night scenery recorded from train rides. The material serves both as visual art work and visual score. A catalog of clips was created for the piece. Each clip starts and ends with a dark sequence, which regularly occurs when filming from a train at night, so that the clips can be arranged seamlessly in any order. For the piece, the various clips are categorized according to visual density, rate of change, object sizes, among others features. Figure 15 shows a few stills from the footage. The top-left figure is a very sparse scene from an Amtrak train ride along the Hudson river, the top-center still is taken from a train ride in Germany with camera focusing on the raindrop-sparkled window. The right image is taken from a train ride in Sendai, and characterized by numerous lights in very symmetrical arrangement. The footage can also be used to blend between different levels of concrete vs. abstract. The piece starts in a randomly selected train station and ends in another one in a different continent.

In our concrete example, CAIRA performs with Braasch (soprano saxophone) and Van Nort (GREIS, Van Nort (2010a)) using sound material from Oliveros (Roland V-Accordion). Prior to the performance, CAIRA’s HMM module was trained on Oliveros’ performance with the trio Triple Point (Braasch, Oliveros, Van Nort). For this purpose, acoustically isolated accordion tracks from a 14-minute clip of a trio session were used to feed the machine learning algorithm. During the performance, this material is transformed and played back based on a dialog with the two live musicians.

Visual leitmotifs exist for each ensemble scenario (e.g., improvisation start, low tension group performance, high-tension group performance, CAIRA’s solo, laptop solo, saxophone solo, improvisation end). Within the high-tension group performance mode, CAIRA also arranges rapid moving video fragments rhythmically to the music performance. While the score is not binding for the live musicians in this piece, it gives insight into the operation of CAIRA, and also provides useful feedback to understand the “intentions” and internal state of the intelligent agent.

5.6 Acknowledgment

The real-time implementation of the CAIRA system was written in Max/MSP utilizing various custom externals and abstractions as well as the FTM, Gabor and MnM packages from IRCAM, externals from CNMAT and Tristan Jehan’s toolboxes (also using their loudness and roughness algorithms for a single-machine, stand-alone version of CAIRA).
References


Khaldi, K., Boudraa, A.-o., Turki, M., Chonavel, T. & Samaali, I. (2008), ‘Audio encoding based on the empirical mode decomposition’, *2008 IEEE PacificAsia Workshop on Computational Intelligence and Industrial Application* (Eusipco), 924–928. **URL:** http://hal.archives-ouvertes.fr/hal-00498439/


**URL:** [http://doi.acm.org/10.1145/1873951.1874221](http://doi.acm.org/10.1145/1873951.1874221)


