Spontaneous organisation, pattern models, and music

YON VISELL

Zero-th Studio, Kozada 1, Stinjan, Pula 52000, Croatia
E-mail: yon@zero-th.org
URL: http://www.zero-th.org

Pattern theory provides a set of principles for constructing generative models of the information contained in natural signals, such as images or sound. Consequently, it also represents a useful language within which to develop generative systems of art. A pattern theory inspired framework and set of algorithms for interactive computer music composition are presented in the form of a self-organising hidden Markov model – a modular, graphical approach to the representation and spontaneous organisation of sound events in time and in parameter space. The result constitutes a system for composing stochastic music which incorporates creative and structural ideas such as uncertainty, variability, hierarchy and complexity, and which bears a strong relationship to realistic models of statistical physics or perceptual systems. The pattern theory approach to composition provides an elegant set of organisational principles for the production of sound by computer. Further, its machine learning underpinnings suggest many interesting applications to emergent tasks concerning the learning and creative modification of musical forms.

1. INTRODUCTION

1.1. Structure and spontaneity

Tension between determinacy and indeterminacy provides some of the most compelling motives in music, especially in technological music, where the means are available to construct computational models which readily span the two. Indeterminacy poses an especially interesting problem for the *performance* of computer music, which tends to invite fine, time-varying control, that requires a tremendous amount of data for its specification. An irony contained in the engineering of indeterminacy for such systems, which look at first glance to be hopelessly over-determined, is that uncertainty in any live performance, especially (but not exclusively) those involving direct human intervention, is a physical and inevitable fact which is essential to the character of being live.

One way to avert a possibly disruptive collision between indeterminacy and control is to develop appropriate models for structured uncertainty that can be used interactively in a performance setting. The structure in these models may be formed by design, through a process of variation, or spontaneously, emerging out of a state of apparent disorder. Phenomena in which discrete and unexpected global behaviours arise through symmetry-breaking occur in certain continuum or quasi-continuum physical systems, including ones which are qualitatively well described by statistical models such as that proposed here. For example, a loss of homogeneity in space or time is associated with the breaking of translation symmetry, corresponding to the development of spatially or temporally local variations. In physics, the breaking of a continuous symmetry is typically accompanied by a change of phase. Continuum or quasi-continuum physical models can have a single phase, or they may have several (Cardy 1996). Transitions from one phase to another may be monitored at any time through the values of certain parameters which indicate the scale of the typical fluctuation away from an ordered state; for example, a temperature variable relative to some critical value, or the magnitude of an external magnetic field.

1.2. The statistics of patterns

The machine understanding of patterns in the real world, including those encoded in musical signals, is a primary goal of the computation of perception. Pattern theory is a term coined by Ulf Grenander (Grenander 1976, 1996) to describe a particular statistical approach to the analysis of complex structure in natural signals, including sound, images, and the weather. By contrast with pattern recognition, whose focus has been the classification of signals according to their information content, the main aim of pattern theory is to find families of statistical models which can, through a process of adaptation to observed data, capture the qualities of the patterns which are seen in nature, in such a way that random samples from the adapted models provide the same 'look and feel' as the samples from the natural world (Mumford 2002). Due in part to a common emphasis on the representation of pattern information in natural systems, including sources of variability, pattern models overlap in interesting cases with those of statistical physics, such as were mentioned above, and there is a commonality of algorithms used for their analysis. A characteristic

Organised Sound **9**(2): 151–165 © 2004 Cambridge University Press. Printed in the United Kingdom. DOI: 10.1017/S1355771804000238

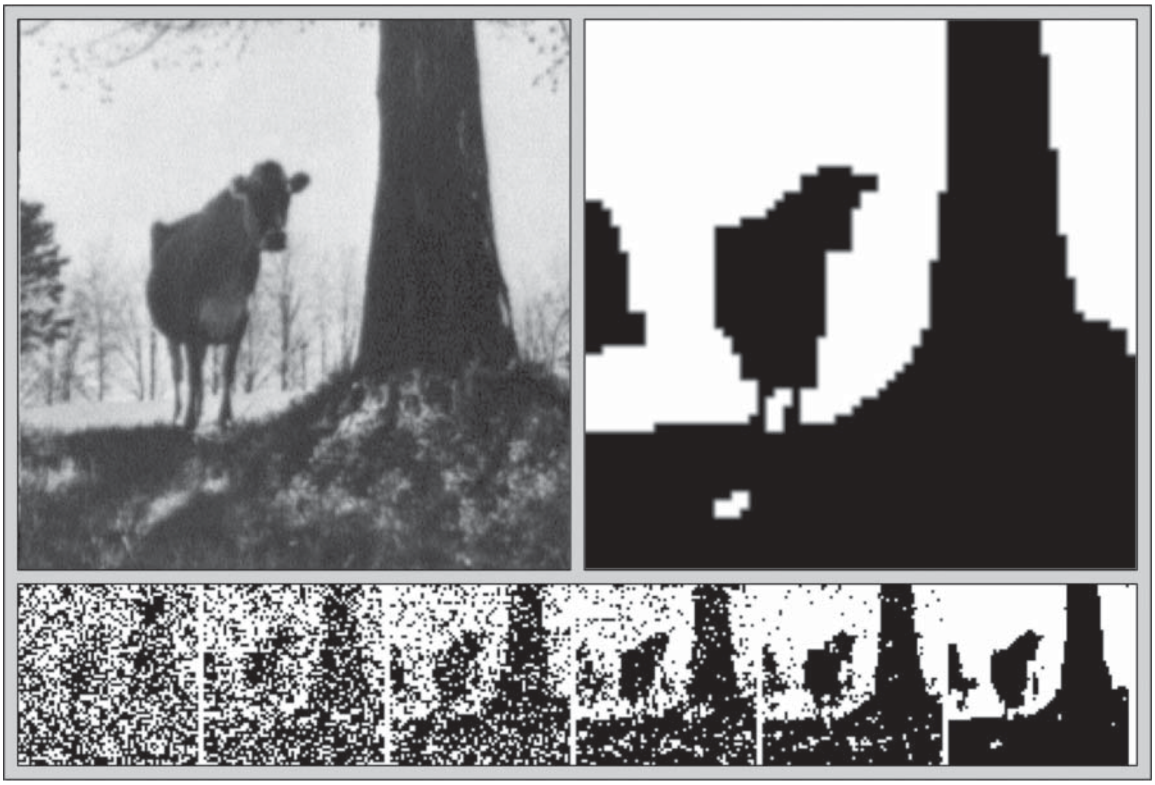


Figure 1. Snapshots of an order–disorder transition in a two-dimensional magnetic Ising Model, as applied to the image of a cow standing in a field (reproduced with permission from Mumford 2002). In the lower row of images, temperature decreases from left to right. At each temperature, a dynamic model exhibiting random fluctuations relative to cow-image dependent properties exists, and above a critical value of the temperature, the model approaches disorder.

example is the two-dimensional Ising Model, which is used in statistical physics to model phase transitions, such as are seen in certain magnets at finite temperature, and in computer vision as a probabilistic model for the segmentation of an image into its constituent shapes (figure 1).

In both pattern-theoretic and statistical physics settings, probability serves to model those uncertainties which are intrinsic to the task and signals, which are too complex to be represented explicitly, or which are irrelevant to the primary phenomena which it is desired to analyse. In this respect, its function is, on one hand, to capture the salient properties of a complex system, and thereby reduce the number of variables. On the other hand, it is to provide a measurement of the accuracy of a given model M relative to a natural signal S which it is supposed to represent. This measurement is frequently cast in the form of a probability functional $P(S \mid M)$ indicating how likely the signal would be to have been produced by the generative model M. A key consideration is the degree to which characteristic features of the signals being studied (i.e. the symmetries, interdependencies and correlations which distinguish them) are captured by the model, once it is adapted to the observed signals. As mentioned above, the comparison of random samples from the distribution $P(S \mid M)$ with the original set of signals may be used to reveal the degree to which these features have been preserved.

1.3. Generative pattern models, naturalness and variation

The foregoing discussion raises issues relevant to the creative algorithmic generation of sensory information. Especially noteworthy is the emphasis which the fields mentioned have placed upon the development, through structural and parametric design, of analytic models which are specifically generative, and capable of synthesising probabilistically varied, natural-feeling sensorial data. The generation of patterns with perceptually natural qualities is closely related to the problem of producing synthetic signals with the nuances that are required to make them seem realistic. A common application of statistical analytic techniques from this area is that of texture generation, which is an important problem in image and graphics synthesis (DeBonet 1997). In sound synthesis, Recht and Whitman (2003) have described statistical audio analysis techniques related to those used in this paper, in application to the steerable generation of perceptually similar families of sound textures.

A related topic – the treatment of variation in pattern models – addresses two issues which are particularly interesting for generative or synthetic art. First, such systems allow, by virtue of their statistical nature, to encode and organise the complex sources of variability, on one or many scales, that are intrinsic to natural signals, and which lend them a realistic quality. Examples include random textural variation or temporal jitter. This role is qualitatively similar to that played by residual modelling in some analysis/synthesis systems used for computer music.

Second, they permit the incorporation of *systematic* causes of deformation or variation in the signal domain. For example, probabilistic contextual models for the time ordering of events (which are typically organised as Markov chains) are employed to account for user-biased local temporal variability in signals, such as speech or music. In automatic speech recognition applications, this organisation permits one to compensate for time-varying rates of speech (Rabiner and Juang 1993), and in synthetic applications to computer music, it can facilitate time-scale modifications of the parameterised sound (Depalle, Garcia and Rodet 1993).

An added advantage of such models is their integration, in addition to parameters which characterise the signal at any point on its domain (which is time, in the case of sound), of hidden variables that may be used to independently control the implicit structure of the signal domain, or in other words the local context which orders signal features. In the case of music or sound, independent modelling of time and parameter-space features is built into the representational structure of many, but not all, systems for composing music. This separation is desirable because it allows time and non-time parameters (for example, spectral content parameters) to be manipulated independently. The range of application and ease with which certain effects may be achieved is thereby enhanced and, in addition, the relationship between model and data features is more transparent than it would be otherwise, because model and data share analogous domains.

The main example exhibiting these features which will be described here (in section 2) is a trainable finite-state parameter sequencing engine structured over a Markov chain. It is a particular adaptation of what is called a hidden Markov model (HMM), a statistical pattern model that has found diverse application in the engineering of systems which deal with time-dependent signals.

1.4. Machine understanding of musical patterns

A great deal of work with analytic statistical models, including the hidden Markov model, has been applied to the understanding of patterns contained in musical signals, on time scales ranging from the composition to the microsound. In data-driven statistical analysis problems, a model is trained on a corpus of musical or sound material which is representative of those attributes which it is desired to capture, such as timbre, compositional structure, or pitch. Certain applications are oriented toward the analysis of musical material for the purpose of simple classification of sounds, parts or compositions. In others, an algorithm is applied subsequent to the analysis which allows one to produce from the model so trained new music or sound material having similar attributes to what was analysed. This is also the approach to musical analysis that pattern theory, as described above, suggests. The production of new examples from a trained model provides an important test for its performance. As Conklin asserts in a review of music generation from statistical models, 'the topics of creative music generation and analysis are in fact highly interconnected, and . . . in principle there is no need to make the classical distinction between analytic and synthetic models of music' (Conklin 2003).

Much of the research to date with data-driven generative stochastic models applied to music has been oriented toward creating computer programmes that are capable of automatically composing in existing styles of music. Partly this represents an approach to mitigating the methodological problems tied up in evaluating the fidelity of automatic composition systems, but the development of systems for composition in historical styles of music (baroque; jazz improvisations) undoubtedly also reflects a desire on the part of researchers to present their work in a way that is stylistically accessible to a wide audience. From one standpoint, it is somewhat surprising that there has not been an explosion of interesting recordings produced from the many systems that have been developed for the simulation of established styles of music. Furthermore, the more interesting creative inquiry – whether an artificial composition system is capable of generating new kinds of music – lies unaddressed by such endeavours. This question lies closer to the spirit which drove earlier innovations in the stochastic approach to music, which were interested in providing new principles that could be applied to the composition of complex music (Xenakis 1971). In addition, experience suggests that the kinds of creative tasks to which humans and computers are well adapted are very different. An example of an exciting emergent application where computers are likely to outperform people is the creation of data-driven audio mosaics from collections of sound material (Schwarz 2000; Zils and Pachet 2001; Lazier and Cook 2003).

1.5. Other related work

Computer music has exploited algorithmic and statistical models since its origin, and since nearly the beginning of the computer age. The use of stochastic models for the organisation, composition and performance of music with computers is by now relatively popular (Zicarelli 1987). Some past and quite recent attempts at the reproduction of compositional styles using data-driven or heuristic artificial intelligence techniques are reviewed, for example, in Conklin (2003) and Papadopoulos (1999).

Continuous density HMMs, such as underlie the system described below, have seen tremendous success in application to problems in time-domain pattern recognition, including speech recognition (Rabiner 1993) and gesture recognition (Yang 1994). Related pattern models such as the Markov random field have been widely used in applications to image texture synthesis (DeBonet 1997) and computer vision (Li 1995).

Following their advance application to automatic speech recognition, hidden variable probabilistic models, such as HMMs, were employed in computer music to analyse musical sounds for subsequent resynthesis. Depalle (1993) utilised sinusoidal partial tracking, while the work of Schoner (1999) was based on adapting weights for a set of Gaussian mixture models. The system described in Donovan and Woodland (1999) uses hidden Markov models for an optimal synthesis of high-quality concatenated speech, and that in Yoshumura *et al.* (1999) takes a spectral approach to HMM-based speech synthesis.

A body of recent work related to score following (Orio and Dechelle 2001) and performance following (Orio 2001; Raphael 2001) exploits trainable, probabilistic hidden variable networks.

A subset of work in musical artificial intelligence (AI) seeks to apply artificial neural networks to composition (Papadopoulos 1999) and synthesis (Hartmann 2003). Other applications of neural networks to music and sound are described in Griffith and Todd (1999). Because these artificial neural networks frequently have a loose perceptual motivation, and resemble some structures through which patterns are learned in the brain, it is tempting to compare them with pattern theoretic tools such as those described here. However, one essential difference is that the standard artificial neural networks do not attempt to model directly the native domain of the signal (time, in the case of sound signals). Consequently, the possibilities for structural refinement based on the analysis of output relative to natural signals are more limited. Some perceptual considerations for AI-based automatic composition systems are described in Purwins, Blankertz and Obermayer (2000).

Finally, HMM methods are attaining substantial application (Birmingham 2002) in music retrieval research. The MPEG-7 multimedia content description framework includes general sound recognition tools. Based on hidden Markov models, the tools are intended to be used for the automatic classification of sounds and for computing similarity metrics between sounds, oriented toward music retrieval applications (Casey 2002).

2. THE SPONTANEOUSLY ORGANISING HIDDEN MARKOV MODEL

Pattern theory and statistical physics serve as important sources of inspiration for the present approach to algorithmic sound. As was mentioned above, HMMs have attained widespread use for the statistical modelling of diverse time-dependent processes. A fusion of this pattern model with stochastic compositional methods is attempted here in an effort to construct a novel interactive algorithmic composition system out of a tool which is also valuable for the analysis of general musical or non-musical sound.

2.1. Overview

The HMM described here is intended as an interactive sequencer for generating patterned data in real time, for consumption by parametric media synthesizers (figure 3). A composer might begin with the model as a statistical sketch of a composition, at the desired level of detail, and improvise with it, either to guide a performance or for data generation in the studio.[1] An illustration of the use of a spontaneously organising HMM as a statistical musical composition is given in figure 2. The input of the model is an excitation source, each excitement of which elicits a chain of note-events. The output is a set of time-varying parameter values intended to specify the sounds that are produced.

The composition is represented by a network consisting of ‘statistical notes’ and weighted connections between them. By ‘statistical note’, it is meant that the note has some properties (e.g. pitch) which are probabilistic, rather than assigned fixed values. The time domain is modelled as a Markov chain, which facilitates the flexible organisation of the statistics of the possible time progressions of sequenced events. Each note is associated with a set of probability densities describing the likelihood for each parameter associated with the note to attain the different values in its range. In addition, there is a fixed duration for each note. It is not normally important which sequence of notes resulted in the parameter curves that are produced; for this reason, the note sequence is said to be ‘hidden’. During performance, the composition is excited and control is exerted by the user. The details are explained in the subsections below. Several applications are described in section 3.

[1] The approach outlined here can apply to any time domain media. I will refer to music for concreteness in what follows.

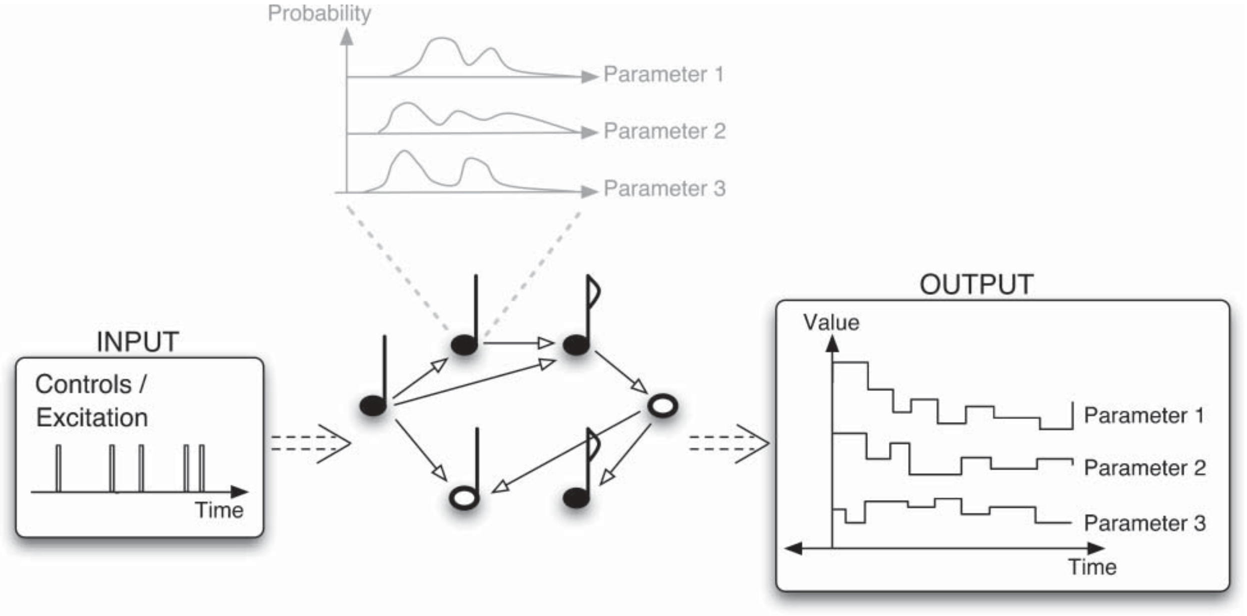


Figure 2. Illustration of the use of the HMM as a statistical musical composition. The score is represented by a network consisting of notes and weighted connections between them. Each note has a duration, as well as a set of probability densities describing the likelihood for each parameter to attain the different values in its range. During performance, the composition is excited and control is exerted by the user, influencing the set of time-varying parameter values that are generated.

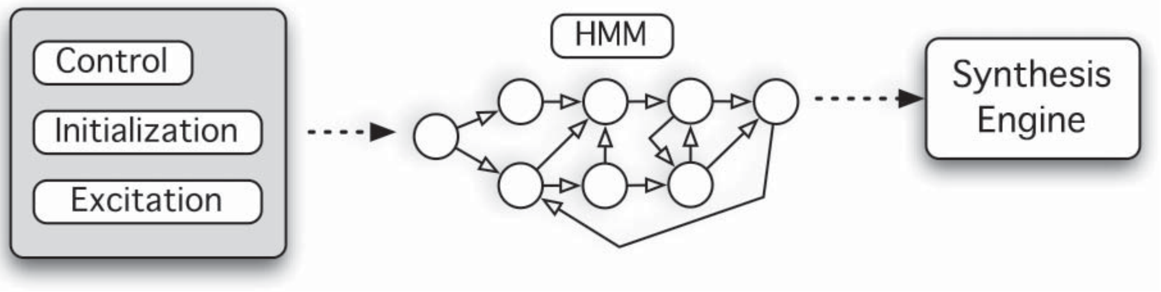


Figure 3. System architecture. Control is afforded through a set of parameters which influence the evolution of a statistical model, and the real-time adaptation of its parameters. The data which is generated is passed to a synthesis engine which is responsible for rendering the time series of parameters as sound or other media.

This system is related to current directions in the use of analytic pattern models in computer music and to the history of stochastic methods for the composition of music (Xenakis 1971). The aim is to facilitate music at time scales spanning the range from microsound (synthesis) to macrosound (composition, performance, installation). Its flexibility of application is intended to be enhanced by virtue of its homogeneity and scalability. A further advantage is the increased modularity suggested by the separation of control and synthesis components, allowing for coupling with synthesis models having different characters or applying to different senses. This complete factoring of synthesis and control models is similar in spirit to that provided in sound server applications, such as the Supercollider 3 language (McCartney 2003).

2.2. hmmm: spontaneously-organising hidden Markov model for Max

The present, spontaneously organising application of the HMM models a distribution for the succession and time of onset of musical events or changes in musical or other performance-related parameters. An implementation in the form of an external for Max/MSP (Cycling'74 2003) may be freely downloaded from the author's website (Visell 2003).

The idea of the continuous density HMM approach to parameter synthesis (figure 4) is that one has a finite-state automaton for the production of a time sequence of parameter vectors. This automaton is specified by primary variables which are capable of modelling the continuous distribution of musical features, and secondary ones that parameterise variability in the time domain. It consists (Rabiner 1993) of the following ingredients:

- A set of K states $S = \{s_1, s_2, s_3, \ldots, s_K\}$ modelling a discrete time progression of events.
- A set of time durations, one for each state.
- A set of transition probabilities, from each state to every other state, $T_{ij} = P(s_j \mid s_i)$.
- An N-dimensional continuous parameter space $\mathbf{V}$ from which the generated sequence of vectors of parameters $\vec{o}_1\vec{o}_2, \ldots$ is to be selected.
- A set of continuous probability distributions $P(\vec{o} \mid s_k)$ for a vector of parameters $\vec{o}$ to be generated by each state S_k.

The Markov property states that the probability for a transition to the next state depends only on the current one. The term 'hidden' refers the fact that the observed sequence of parameters $\{\vec{o}_1, \vec{o}_2, \ldots\}$ is not uniquely associated to the state sequence which produced it, and is normally to be inferred only probabilistically from the state sequence. Because the current application involves sampling from the statistical model, the actual state sequence which produced the observed parameter sequence is available as well.

At each state s, the continuous probability density $P(\vec{o}|s)$ on the N-dimensional space of musical parameters is given the form of a weighted mixture

$$P(\vec{o} \mid s) = \sum_{\alpha=1}^{M} w_\alpha \, G(\vec{o}, \vec{\mu}_\alpha, \Sigma_\alpha)$$

of M normalised Gaussian functions

$$G(\vec{o}, \vec{\mu}_\alpha, \Sigma_\alpha) = \frac{1}{\det(2\pi \Sigma_\alpha)^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\vec{\mu}_\alpha - \vec{o})^t \cdot \Sigma_\alpha^{-1} \cdot (\vec{\mu}_\alpha - \vec{o})\right)$$

The parameters specifying each probability density are:

- M N-dimensional mean vectors, μ_α, $\alpha = 1 \ldots M$
- M N-by-N variance matrices Σ_α
- M weights w_α

The probability normalisation condition requires that the weights at each state sum to one.

Any state may be marked as *non-emitting*, in which case no parameter will be produced by it, allowing it to act as a pause, temporal gap, or rest. Furthermore, a state is allowed to have a duration of zero. A typical application of a zero-duration non-emitting state is to act as a proxy gateway, assigning probabilities for various states in the network to act as an entryway. A state may also be identified as an exit to the network, so that no subsequent transition takes place, thus serving as a possible terminus for a sequence.

Sequences of parameter vectors are produced in real time according to the time delays and connectivity of the directed graph that underlies the model, recapitulating the paths of virtual symbolic tokens through it (figure 5). A sequence begins with the entrance into the network of such a token at any desired state s_i. The

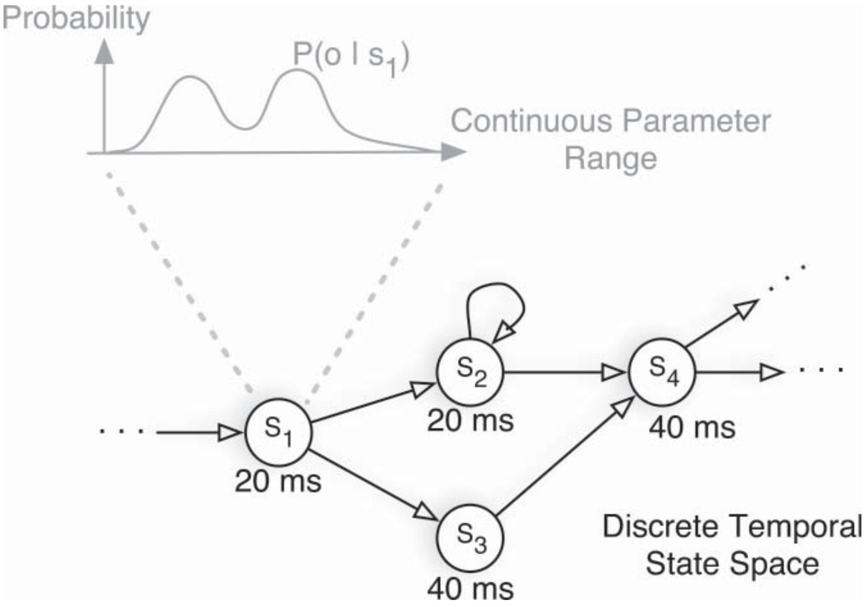


Figure 4. Structure of the graphical, finite-state model for musical parameter synthesis. Time is prescribed as the collection of nodes of a weighted directed graph, each having a fixed duration. Weights assigned to the edges give the probability for a particular transition to occur. Musical parameters at each state are modelled by a probability density on a continuous space of N dimensions.

default entry point is the first state. If the state is emitting, a parameter vector is produced by sampling the continuous probability density function at s_i. After a time given by the duration assigned to the state, a new state s_j is selected, a parameter vector is produced by the state, and the process continues until, for example, an exit state is encountered.

An arbitrary number of tokens in the network may be active at any given time, limited only by user-controlled settings and computational resources, with the result that between one and hundreds or more interwoven streams of parameter sequences may be produced concurrently from a single model instance.

Table 1. The number of each type of model data. K is the number of states, M the number of Gaussian components of each mixture, and N is the dimension of the musical parameter space.

Parameter Type	Number of Values
Mean Vectors	$N \times M \times K$
Variance Matrices	$N \times (N - 1) \times M \times K$
Diagonal Variances	$N \times M \times K$
Mixture Weights	$M \times K$
Transition Probabilities	$K \times K$
State Durations	K

2.3. The assignment of model data

An application of the HMM requires the specification of an initial set of model parameters and properties. The structural scale of the model is determined by the choice of a number of states K, the dimension of the musical parameter space N, and the number M of Gaussian components of each mixture in the probability distribution of each state. The quantity of parameters corresponding to each type of model parameter is listed in table 1.

2.3.1. Variances and independent controls

It is sensible to adopt a simplified, diagonal form for the variance matrices whenever possible. Such a choice significantly improves the convergence of a parameter adaptation algorithm such as that described in the next subsection, by reducing the number of variance components to be adapted by a factor of $N - 1$. Employing diagonal variances is justified under the assumption of decorrelation among the components of the sequenced parameters (Rabiner 1993). For multivariate Gaussian distributions, decorrelation is equivalent to statistical independence. Consequently, if the sequenced parameters are the controls of independent features of a musical process, then the use of diagonal variances is statistically optimal (and it provides an approximation which is more natural to the degree that this is the case). For example, the independent musical features could be the amplitudes of what are desired to be independent sources in a mix, or the values of perceptually independent timbre space

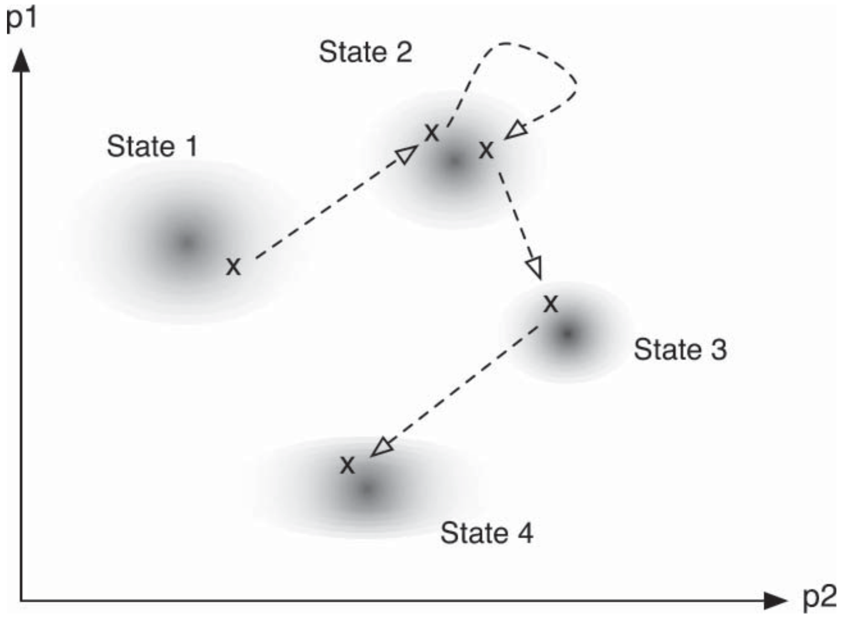


Figure 5. A sequence of parameter vectors, labelled 'x', in a two-dimensional parameter space or subspace, generated from a model composed of single Gaussian mixture states. The dashed line indicates the discrete sequencing path followed by a virtual token in the network of states (i.e. 1 → 2 → 3 → 4). Darker areas are regions of higher probability for each Gaussian distribution.

parameters (Grey 1977). From the analysis standpoint, the MPEG-7 content description format defines standard methods for extracting decorrelated spectral basis functions from analysed sound, based on the Independent Components Analysis (ICA) algorithm. Similar methods were applied to a synthesis-by-analysis of sound textures using characteristic spectral templates in Recht (2003).

2.3.2. Initialisation algorithms

Assumptions about statistical independence notwithstanding, the amount of data required to specify the model can be quite large. In practice, some algorithm or automated procedure is required to fix initial values. Apart from assigning the same data to all states, which may be suitable for applications which involve spontaneous organisation from disorder, initial model data can be chosen by: random assignment, through the use of algorithms computed from state and mixture indices, or using values learned from one or more data examples. The last may come from data files written by an HMM recognition package such as HTK (Young 1994), or through models shared using the general sound specification standard MPEG-7 (Casey 2002).

2.3.3. Temporal structure

The transition data determines the model topology, or collection of possible pathways in time. This should normally be handled separately from the other parameters, because it can have a profound effect on the results that are produced, determining such characteristics as the minimum length of cyclic sequences of states, the maximum length of non-repeating cycles of states, and the local branching characteristics between possible paths. Interesting model topologies range from the highly ordered left-to-right class of models, in which states are organised in a numerically increasing linear fashion, each state connected only to the next N states, to ergodic topologies in which each state is accessible from every other state.

2.3.4. Hierarchical network organisation

A note on hierarchical applications, which model time resolution at more than one scale, may be useful. Such hierarchical structures can be found in many natural and musical processes. They also manifest in general artificial intelligence approaches, such as context-free grammars, which seem to be relevant to perceptual specialisation in music audition (Purwins 2000). A multi-scale network may be defined by assigning sets of nodes a hierarchy of durations, which might be chosen to be on the order of at, a^2t, a^3t, . . ., where a is a parameter determining a characteristic distance between scales. The transition graph that organises such a hierarchy may be constructed with the help of a separate algorithm which assigns the transition probabilities and durations. A more intuitive approach, which is also common in grammar-based pattern recognition applications, is to build a hierarchy from a graph of graphs, mapping the exit states of a given model onto the entry states of another (figure 6). Within software such as Max/MSP, it is easy to iterate the procedure of creating graphs of graphs, so as to build a hierarchical structure of as much depth as is required.

Such flexibility is an asset of pattern models which are built over graphs. As stated in Jordan (1998): 'Fundamental to the idea of a graphical model is the notion of modularity: a complex system is built by combining simpler parts. Probability theory serves as the glue whereby the parts are combined, ensuring that the system as a whole is consistent and providing ways to interface models to data'.

2.4. Adaptation

The goal of adaptation as applied to the spontaneously organising HMM is not, as would be the case in a conventional time-dependent pattern recognition application, the learning of parameters describing representative features of a class of patterns. Rather, it is the interactive refinement of model parameters in a way that is intended to form musical structures in a spontaneous way, eliciting features like repetition, variation and transience. This is accomplished by adapting the model parameters at or near to a state which has produced a given observation to the observed parameters itself (figure 7). In brief, the signal which samples the model is obtained through the finite-state decoding of the network. For each parameter vector that is produced, the model is adapted in a way that increases the similarity between the model and that sampled parameter vector. This proceeds in greater detail as follows.

2.4.1. Training algorithm

First, a transition graph training rate ζ is defined, where ζ is greater than zero and not much larger than one. After a transition from the state s_i to the state s_j occurs, the transition matrix element between them is updated according to

$$T_{ij} \rightarrow T_{ij} + \zeta$$

and the new transition matrix is renormalised, to insure probability conservation.

Once a new parameter vector $\vec{o}$ is sequenced by the model state s, the state is adapted in a way that increases the similarity between the features of which

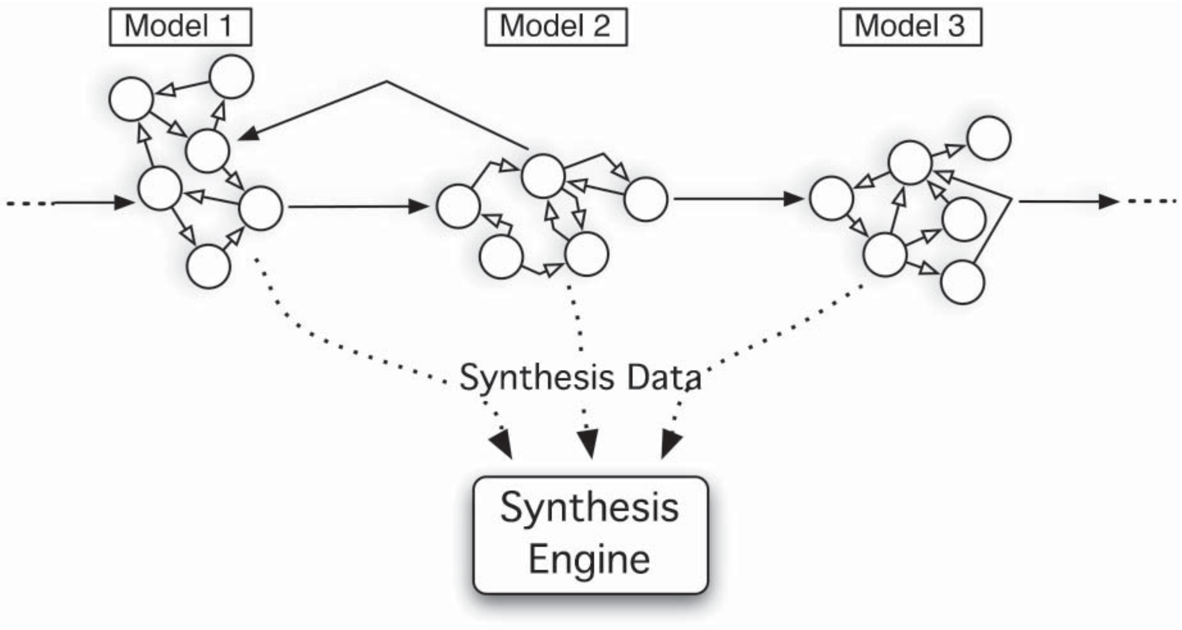


Figure 6. An example hierarchy of structure, consisting of the microscopic level of detail modelled in the structure of each sub-network, and meso- to macroscopic levels of detail determined by the organisation of the network of sub-networks. Solid lines indicate sequencing paths that a token in the network may follow. Dotted lines indicate the routing of synthesised data.

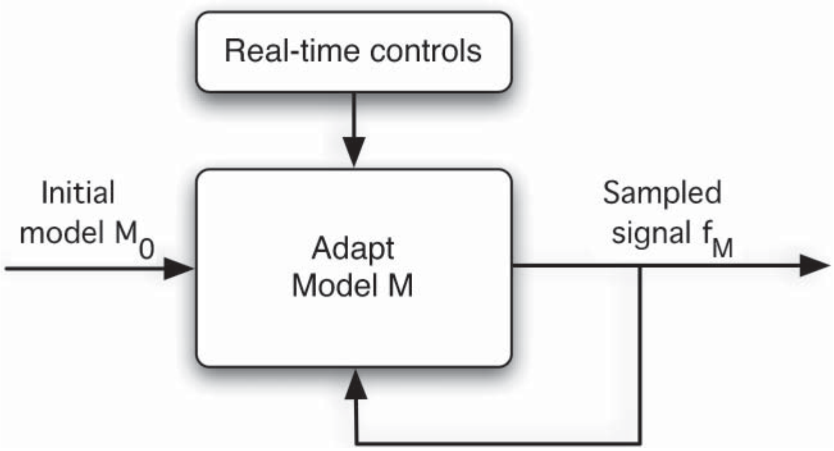


Figure 7. Diagram showing the use of the described model for real-time parameter sequencing, with adaptive feedback.

that state is characteristic and the sampled parameter vector. For this purpose, a standard gradient ascent approximation algorithm is employed (Press 1993). This is done with the mixtures kept *hidden*: although the observed parameter vector is attributable to a single Gaussian component of the probability density at *s*, each mixture is trained in proportion to the likelihood that it produced the observation. The weighted Gaussian quantity

$$L_\alpha(\vec{o}) = w_\alpha G(\vec{o}, \vec{\mu}_\alpha, \Sigma_\alpha)$$

gives the likelihood that the observed parameter vector $\vec{o}$ was produced by the α^{th} mixture at the current state. An observation space training rate λ is defined, where again λ is not much larger than one. The model parameters for each mixture are updated as follows:

$$\begin{aligned} &w_\alpha \rightarrow w_\alpha + \lambda L_\alpha \\ &\vec{\mu} \rightarrow \vec{\mu}_\alpha + \lambda(\vec{\mu}_\alpha - \vec{o})L_\alpha \\ &\Sigma_\alpha \rightarrow \Sigma_\alpha + \lambda(\vec{\mu}_\alpha - \vec{o})(\vec{\mu}_\alpha - \vec{o})^t L_\alpha \end{aligned}$$

(The *t* in the exponent denotes the transpose of the column vector.) Subsequently, the weights w_α have to be renormalised for the given state.

The result is that, for moderate values of the training parameters (on the order of 1.0), the model probability distributions localise in tandem onto some trajectory in parameter space (with caveats that are mentioned below). The transition probabilities at each state sampled by the trajectory become dominated by a single transition, the weight for one component of the continuous probability distribution at each state likewise dominates the others, and the N-dimensional Gaussian distribution for the dominant weight peaks sharply about a particular mean vector. Although the optimal rate of convergence of such a descent algorithm is relatively slow, the radius of convergence is rather large. In practice, the procedure works well without modifications provided N is not too big (say, $N < 11$; see further remarks on this important caveat in section 2.5).

2.4.2. Temporal smoothing

The adaptation mechanism described in the preceding subsection is not, in distinction to algorithms used in recognition applications, based upon training a model with entire sequences of data (an approach which benefits from considering a large set of hypotheses about how the pattern could have been produced by the model), but rather is one which trains states on individual sequenced data vectors that have been produced by them in particular. What is lost, relatively speaking, are temporal correlations between the probability distributions of adjacent states of the model which could otherwise have been learned from the sequenced data. A heuristic approach to bettering this situation is adopted here. Rather than training the state on the data vector $\vec{o}(n)$. produced at time $t = n$, one defines an integrating parameter γ, whose value lies between zero and one. One trains instead using the time-integrated observation vector $\vec{o}'(n)$ given by

$$\vec{o}'(n) = \gamma\vec{o}(n) + (1 - \gamma)\,\vec{o}'(n - 1)$$

Using this procedure, the mixtures of a given state are trained with a decaying time average of the observation vector sequence, with a time constant of $\tau = 1/\gamma$ samples.

As described in the next section, the training process may be influenced through real-time control of various user-accessible parameters which are capable of governing it.

2.5. Control, phenomenology and evolution

All of the statistical parameters defining the model may be modified interactively. For example, the durations of all states might be scaled by the user during pattern synthesis in order to modify the event density. Except for small models, individual control of the parameters at every state would be difficult, due to the number of values which would need to be managed in real time. However, in addition to those local parameters, several high-level interactive controls for initiating and influencing the global model behaviour have been implemented (figure 9). These control options are not oriented toward the gestural control of music, and in that respect are in harmony with other approaches to computer music which concern themselves with providing the performer with a set of tools for controlled improvisational composition in real time (e.g. Ableton 2003). A summary of the interactive control parameters appears in table 2.

2.5.1. Balancing order and disorder

The training rates defined in the previous subsection independently govern the rate at which patterns are localised in time and in the observation space. Working against the adaptive sharpening of parameters, minimum values for the transition probabilities (figure 8), Gaussian mixture weights, and statistical deviations may be fixed. Furthermore, a global order/disorder parameter exists, analogous to temperature in a statistical model such as the Ising model (figure 1). It controls the scale of the statistical deviations of the parameters.

2.5.2. Perturbing patterns

As an interesting result of the tension between training and the minimum limit placed on the variances, once a quasi-equilibrium state is reached, with a temporal pattern of states being repeated in time (or nearly so), a random walk of the variances may be effected. This makes it possible to perform perturbations to an established pattern which are controllable in scale (but not in form), while keeping the time progression of the pattern fixed. Furthermore, if the system has localised reasonably to a pattern, one may deconstruct it without affecting the underlying model parameters. This is done by moving the temperature parameter toward disorder, without training. The original pattern is reconstructed by reducing the temperature to a low value (1.0 is typical).

2.5.3. Controlling temporal event density

There are two basic methods for controlling the level of detail of synthesised parameter contours or events, by influencing the density and occurrence of tokens into the network. In the first, tokens are created at a chosen time or interval, by indicating at each instance the index of the state at which they should enter the network. In the second, a global decay value between minus one and one is set which governs the spontaneous production and absorption of tokens at each state. When the value is positive, it specifies the probability that a token exiting that state will not transition to another, but rather cease to exist (in other words it is a quantity to be subtracted from all model transition probabilities). When it is negative, it determines the probability that a second token will

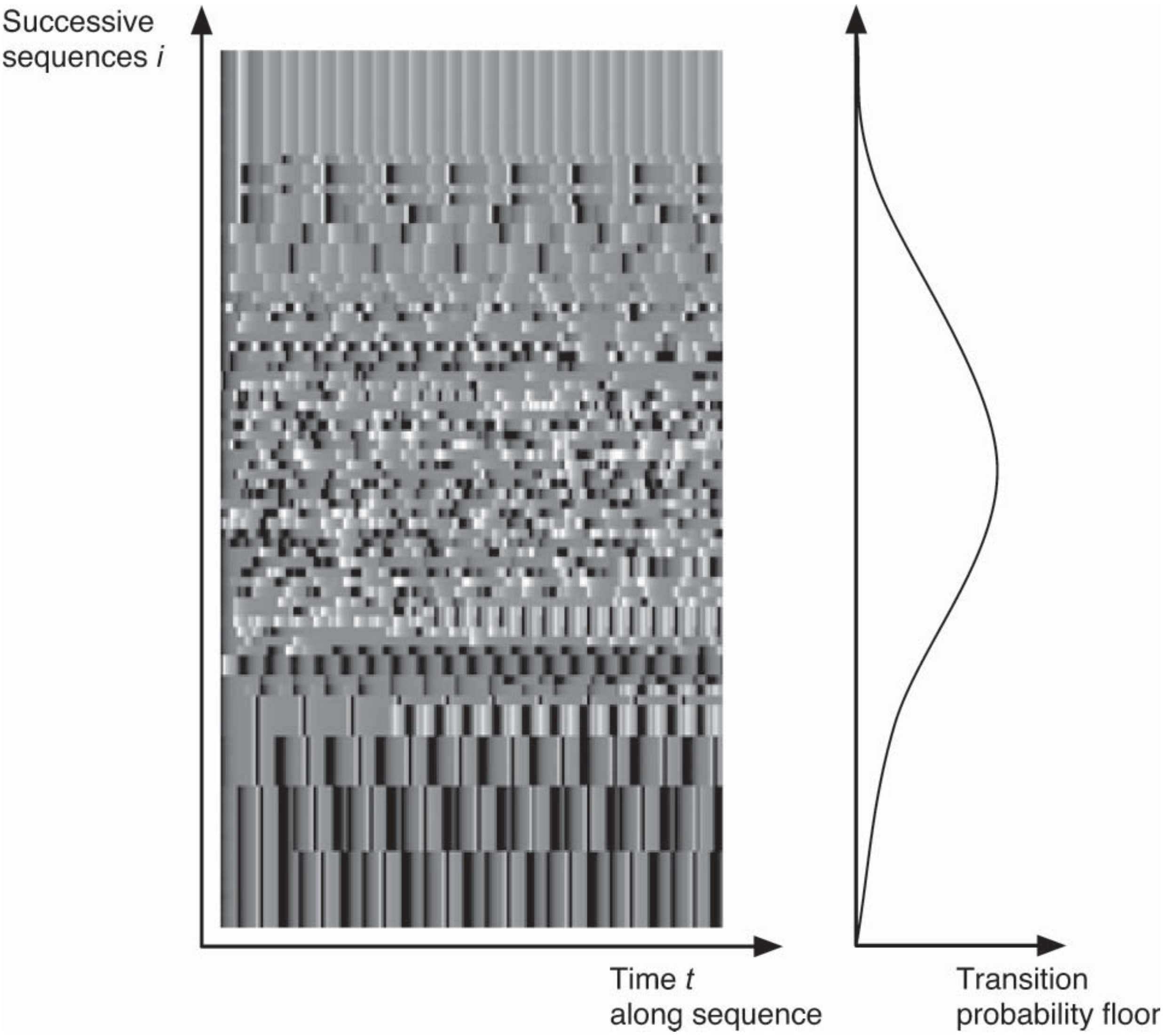


Figure 8. Output illustrating evolution between different ordered patterns, and between ordered and disordered ones, over time, as the transition probability floor is gradually raised and then lowered. Sequences produced by successive tokens are displayed in rows proceeding upward, with time running from left to right in each row. In the plot, value of an observed parameter for this model determines the greyscale value of each pixel in the image.

be spawned to exit the state, in addition to the first. Through these mechanisms, the number of active tokens in the network may be controlled independently of the time scale specified by the durations; and, conversely, durations may be modified without affecting the network density of tokens.

2.5.4. Summary of phenomenology

To summarise, the collection of interesting uses and phenomena relevant to this model includes:

- The creation and interactive execution of statistical compositions, through the sculpting of a model for the synthesis of parameters, which is structured over a directed graph.
- Invocation, evolution and disordering of sequences of N-dimensional parameter vectors.
- Production of interleaved streams of correlated or non-correlated sequenced parameters.
- Spontaneous formation, deformation and destruction of long-range structure.
- Interactive construction and deconstruction of sequences.
- Random walks through paths of sequences in parameter space.
- The generation of variations in time or parameter space of a repetitive sequence through perturbations.
- Introduction through training of non-local interactions in time between feature parameters.

Table 2. List of interactive controls for the model as described in this section (see also figure 9).

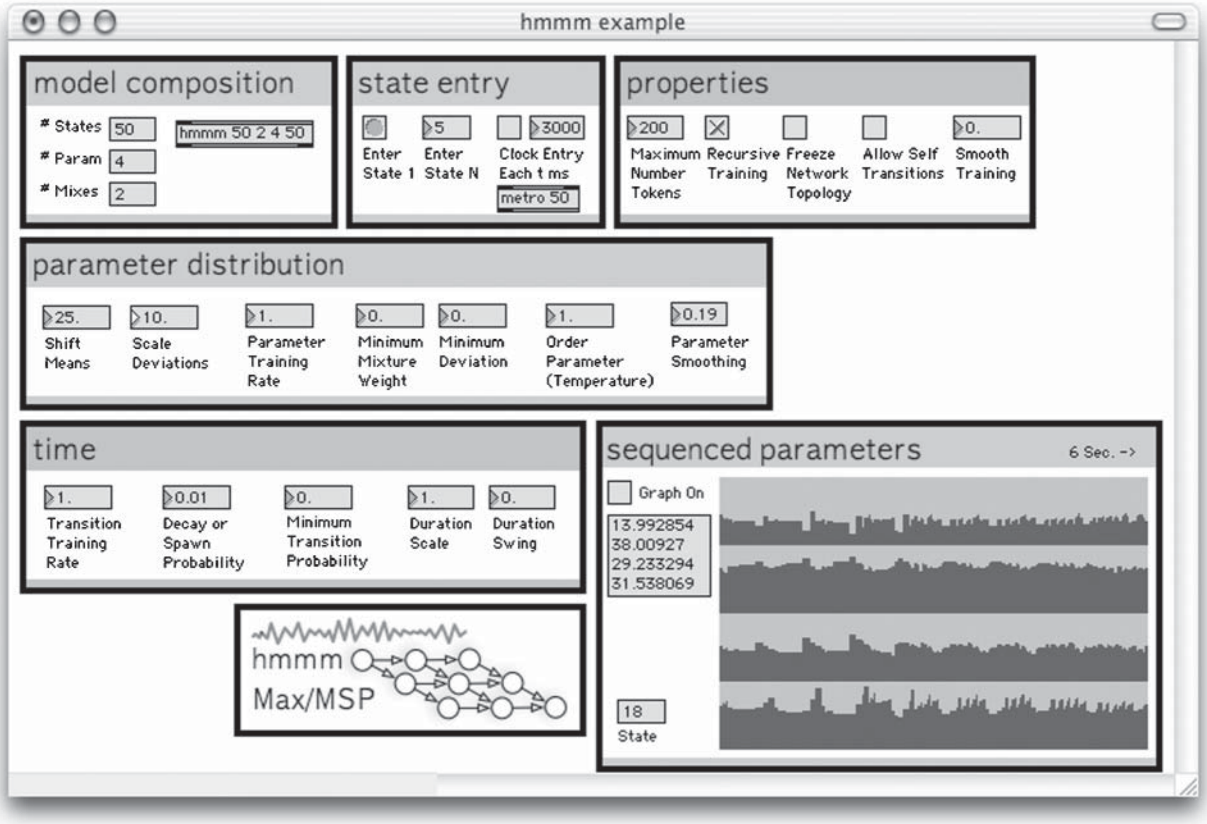
Interactive Controls
Mean Values
Deviations
Mixture Weights
Transition Probabilities
State Durations
Minimum Deviation
Minimum Transition Probability
Temperature
Parameter Training Rate
Transition Training Rate
Token Instantiation
Token Spawn/Decay Rate

- Inter-token interactions produced through heightened token densities and training rates (as one token enters a state that has just been visited by another).

2.6. Practical considerations and caveats

There are a few important practical considerations concerning the operation of the model described. Firstly, a certain amount of practice is required in order to avoid scenarios which could cause the sequencing to destabilise during performance, with sequenced parameter values growing out of control. This typically occurs when the minimum variance is set to a high value coincident with a large value for the observation vector training parameter, a situation which may be encountered while evolving sequences quickly over time. One resolution is to apply a compressive nonlinear mapping to the range of parameters once they are produced. Such a procedure allows for control of the adaptation rate (and consequently of the characteristic size of training corrections) for different regions of parameter space.

Secondly, as mentioned above, there is a limitation on the degree to which correlations may be learned using the time-integrated local training algorithm described here. In the future, it is desired to experiment with training based on sequences or sub-sequences



hmmm example

model composition

# States 50

# Param 4

# Mixes 2

hmmm 50 2 4 50

state entry

● Enter State 1

5 Enter State N

□ 3000 Clock Entry Each t ms

metro 50

properties

200 Maximum Number Tokens

☒ Recursive Training

□ Freeze Network Topology

□ Allow Self Transitions

0. Smooth Training

parameter distribution

25. Shift Means

10. Scale Deviations

1. Parameter Training Rate

0. Minimum Mixture Weight

0. Minimum Deviation

1. Order Parameter (Temperature)

0.19 Parameter Smoothing

time

1. Transition Training Rate

0.01 Decay or Spawn Probability

0. Minimum Transition Probability

1. Duration Scale

0. Duration Swing

sequenced parameters 6 Sec. ->

□ Graph On

13.992854
38.00927
29.233294
31.538069

18 State

Figure 9. A demonstration interface to the Max/MSP implementation of the HMM sequencing software, illustrating the subset of real-time control possibilities which apply to parameters affecting the entire model at once. Further control over subsets of model parameters and behaviour is available by means of commands which modify values individually or in subsets. Sequenced parameters might be passed to an audio or visual media synthesizer running on the same computer, or transmitted to another location using a network data transmission protocol such as OpenSoundControl (Freed and Wright 1997).

of data, using one of several Bayesian probabilistic training methods which have been developed (Rabiner 1993), including variants of the online expectation maximisation algorithm.

Thirdly, there are generic problems with the training of probabilistic models having high-dimensional feature spaces. In the case of Gaussian models in N dimensions, the problem occurs because most of the probability mass for the distribution lies within a shell at a distance approximately $\sqrt{N|\sum|}$ from the peak of the distribution, with the result that, unlike the low-dimensional case, most samples from the distribution do not fall near to the maximum (Bishop 1995). One resolution is to divide the sequenced patterns into statistically separate streams, as is common in large feature space pattern recognition problems (Young 1994). Provision for this is envisioned for the next release of the software.

Finally, the time-domain model which is implied here is a discrete one, while one might in fact prefer to model *continuously* parameterised sound processes. Options for doing so include choosing a fine enough time granularity, employing an interpolation scheme, or some combination of the two. Furthermore, as the current methodology is applicable to continuous time domain Markov processes as presented, for example, in Saul and Rahim (2000), it would be interesting to implement the corresponding spontaneously organising continuum model to explore any novel phenomena that may result.

3. APPLICATIONS

A great advantage of interactive algorithmic control systems such as that described in this paper is that they are not tied to a particular synthesis method, or even to the medium of digital audio. Consequently, only a few application possibilities can be mentioned here. These come largely from experiments, performance, and studio work by the author.

A model with a single state and several Gaussian mixtures is useful in application to the synthesis of quasi-stationary sound processes, including noise, drones and textures, as in Recht (2003). Gaussian mixture models have been applied to data-driven synthesis in cluster weighted modelling (Schoner 1999).

Applications to fundamentally time-varying processes may invite models having anywhere from several states to several thousand states. Some examples sorted according to the time scale native to their application include:

- The sequencing of sound at the scale of gestures, words or phrases.
- The production of sequences of parameterised notes or note-like kernels.
- The generation of data parameterising spectral frames for continuous processes such as sinusoidal, sinusoidal plus noise, or other resynthesis techniques.
- The production of data parameterising streams of synthesised (for example, using formant wave synthesis) or sampled sound grains.

An approach to HMM-based sampled granular synthesis which proves useful is to regard each state of the model as a grain of sound. Parameters specifying the grain – its duration, position or relative position, playback speed, applied filtering, amplitude envelope – are determined probabilistically, according to the continuous distribution at the corresponding state, and the sequence of grains played back is obtained during sequencing from the model topology (which is to say the topology on the collection of sound grains) and transition probabilities. The grain duration may be either linked to the state duration, or independent of it. Sound may be played back from a buffer or soundfile in precisely or approximately the way it is stored in the buffer or soundfile, by associating to it an HMM with a predominantly left-to-right topology and probability distributions which are sharply focused at the correct window shape, duration, and amplitude envelope parameter values. This playback may be interactively deconstructed through the control of the model parameters indicated above.

An exciting related class of applications which has received significant attention recently (Schwarz 2000; Zils 2001; Lazier 2003) is that of the audio mosaic, in which a sound is resynthesised according to its relative similarity to a collection of learned examples, or is generated as a kind of reconstructive synthesis of a body of musical material, allowing a simultaneous microscopic and inter-relational approach to composition or microcomposition. The system described here gives some indication of how such a process can be interactively and creatively steered.

4. CONCLUSION

A spontaneously organising pattern model may also be viewed as an analysis–synthesis statistical model which, absent of external data sources, is trained on itself, by causing its perception to be adapted to self-generated observations. An interesting effect which has been observed in a number of perceptual systems starved of sensory input is that, under certain circumstances, they are capable of synthesising patterns of sensory information from nothing. For example, in the human visual cortex, organically and inorganically evolving geometric shapes called phosphenes can be produced when the retinal image is held extremely constant, and perturbed in some fashion from its uniform steady state, as by pressure exerted on both closed eyes, or when sitting for a prolonged time in total darkness (Tkaczyk 2001). The synthetic model described may in this sense be thought of as analogous to a starved sensory organ. It is capable of generating spontaneously patterned data for synthesis by an appropriate audio or visual synthesis engine.

On the side of current audio technology, while the move was perhaps not primarily directed at applications having an aleatoric or creative bent, the integration of an HMM representation for sound into the MPEG-7 multimedia content description framework (Casey 2002) is very promising for the spread and interoperability of HMM-based sound or other time-domain media sequencing and synthesis applications, such as that described here.

Concerning sources of inspiration for the stunning conception and formulation of stochastic music expressed in his book *Formalized Music*, Xenakis refers to the statistical physics foundations which have influenced models such as that presented in this paper, writing:

> The basic principles of kinetic gas theory, which are described by statistical mechanics, are very simple and very general. They can be found in music as well. (Xenakis 1996)

It is surprising that, forty years after the first publication of *Formalized Music*, the elaboration of new methods for organising sound continues to be enriched by the development of theory and tools for the computation of nature and for the statistical understanding of its perception.

REFERENCES

Ableton. 2003. *Live 3.0*. http://www.ableton.com

Birmingham, W., Pardo, B., Meed, C., and Shifrin, J. 2002. The MusArt music-retrieval system. *D-Lib Magazine* **8**(2).

Bishop, C. M. 1995. *Neural Networks for Pattern Recognition*. Oxford, UK: Oxford University Press.

Cardy, J. 1996. *Scaling and Renormalization in Statistical Physics*. Cambridge, UK: Cambridge University Press.

Casey, M. 2002. Generalised sound classification and similarity in MPEG-7. *Organised Sound* **6**(2).

Conklin, D. 2003. Music generation from statistical models. In *Proc. of the AISB 2003 Symp. on AI and Creativity in the Arts and Sciences.* Aberystwyth, Wales.

Cycling'74. 2003. *Max/MSP 4.2*. http://www.cycling74.com/products/maxmsp.html

DeBonet, J. S. 1997. Multiresolution sampling procedure for analysis and synthesis of texture images. *Proc. of SIGGRAPH 97*, pp. 361–8.

Depalle, P., Garcia, G., and Rodet, X. 1993. Tracking of partials for additive sound synthesis using hidden Markov models. In *Proc. IEEE-ICASSP*.

Donovan, R. E., and Woodland, P. C. 1999. A hidden Markov-model-based trainable speech synthesizer. *Computer Speech and Language*: 1–19.

Dubnov, S., Assayag, G., Lartillot, O., and Bejerano, G. 2003. Using machine-learning methods for musical style modeling. *Computer Magazine*, October. IEEE.

Freed, A., and Wright, M. 1997. Open SoundControl: A new protocol for communicating with sound synthesizers. *Proc. of the Int. Computer Music Conf.* Thessaloniki, Greece.

Grenander, U. 1976. *Pattern Synthesis.* Springer Verlag.

Grenander, U. 1996. *Elements of Pattern Theory*. Johns Hopkins University Press.

Grey, J. M. 1977. Multidimensional perceptual scaling of musical timbres. *Journal of the Acoustical Society of America* **61:** 1,270–7.

Griffith, N., and Todd, P. M. (eds.). 1999. *Musical Networks: Parallel Distributed Perception and Performance*. MIT Press.

Jehan, T. 2001. *Perceptual Synthesis Engine: An Audio-Driven Timbre Generator*. Master's Thesis, Massachusetts Institute of Technology.

Jordan, M. I. 1998. *Learning in Graphical Models*. MIT Press.

Kersten, D., and Schrater, P. 1999. Pattern inference theory: A probabilistic approach to human vision. In R. Mausfeld and D. Heyer (eds.) *Perception Theory: Conceptual Issues*. John Wiley and Sons.

Lazier, A., and Cook, P. 2003. MOSIEVIUS: Feature driven interactive audio mosaicing. In *Digital Audio Effects (DAFx)*. London, England.

Li, S. Z. 1995. *Markov Random Field Modeling in Computer Vision*. Springer Verlag.

McCartney, J. (and contributors). 2003. *Supercollider 3*. http://www.audiosynth.com

Mumford, D. 2002. Pattern Theory: the mathematics of perception. In *Proc. of the Int. Congr. of Mathematicians*. Beijing.

Orio, N. 2001. An automatic accompanist based on hidden Markov models. In F. Esposito (ed.) *AIIA 2001: Advances in Artificial Intelligence*. *Proc. of 7th Congr. of the Italian Association for Artificial Intelligence*, pp. 64–70. Bari.

Orio, N., and Déchelle, F. 2001. Score following using spectral analysis and hidden Markov models. *Proc. of the Int. Computer Music Conf.*, pp. 125–9. La Habana.

Papadopoulos, G., and Wiggins, G. 1999. AI methods for algorithmic composition: A survey, a critical view and future prospects. In *Proc. of the AISB'99 Symp. on Musical Creativity*.

Press, W., Flannery, B. P., Teukolsky, S. A., and Vetterling, W. T. 1993. *Numerical Recipes in C*. Cambridge, UK: Cambridge University Press.

Purwins, H., Blankertz, B., and Obermayer, K. 2000. Computing auditory perception. *Organised Sound* **5**(3): 159–71.

Rabiner, L., and Juang, B.-H. 1993. *Fundamentals of Speech Recognition*. Prentice Hall.

Raphael, C. 2001. A Bayesian network for real-time musical accompaniment. In *Neural Information Processing Systems (NIPS)* **14**.

Recht, B., and Whitman, B. 2003. Musically expressive sound textures from generalized audio. In *Proc. of the 6th Int. Conf. on Digital Audio Effects (DAFx-03)*. London, UK.

Saul, L. K., and Rahim, M. G. 2000. Markov processes on curves. *Machine Learning* **41**(3): 345–63.

Schoner, B. 2000. *Probabilistic Characterization and Synthesis of Complex Driven Systems*. Ph.D. Thesis, Massachusetts Institute of Technology.

Schwarz, D. 2000. A System for data-driven concatenative sound synthesis. In *Digital Audio Effects (DAFx)*. Verona, Italy.

Tkaczyk, E. 2001. Pressure hallucinations and patterns in the brain. *Morehead Electronic Journal of Applicable Mathematics* **1**.

Visell, Y. 2003. *hmmm for Max/MSP*. http://www.zero-th.org/perception/hmmm

Xenakis, I. 1971. *Formalized Music*. University of Indiana Press.

Xenakis, I. 1996. Determinacy and indeterminacy. *Organided Sound* **1**(3): 143–55.

Yoshimura, T., Tokuda, K., Masuko, T., Kobayashi, T., and Kitamura, T. 1999. Simultaneous modeling of spectrum, pitch, and duration in HMM-based speech synthesis. In *Proc. Eurospeech*.

Young, S. J. 1994. The HTK hidden Markov model toolkit: Design and philosophy. Cambridge University Engineering Department document *CUED/F-INFENG/TR*.152.

Zicarelli, D. 1987. M and Jam Factory. *Computer Music Journal* **11**(4): 13–29.

Zils, A., and Pachet, F. 2001. Musical mosaicing. In *Digital Audio Effects (DAFx)*. Limerick, Ireland.