GENERATIVE RHYTHMIC MODELS

A Thesis
Presented to
The Academic Faculty

by

Alex Rae

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in Music Technology in the
Department of Music

Georgia Institute of Technology
May 2009

GENERATIVE RHYTHMIC MODELS

Approved by:

Professor Parag Chordia, Advisor
Department of Music
Georgia Institute of Technology

Professor Jason Freeman
Department of Music
Georgia Institute of Technology

Professor Gil Weinberg
Department of Music
Georgia Institute of Technology

Date Approved: May 2009

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

A system for generative rhythmic modeling is presented. The work aims to explore computational models of creativity, realizing them in a system designed for realtime generation of semi-improvisational music. This is envisioned as an attempt to develop musical intelligence in the context of structured improvisation, and by doing so to enable and encourage new forms of musical control and performance; the systems described in this work, already capable of realtime creation, have been designed with the explicit intention of embedding them in a variety of performance-based systems. A model of qaida, a solo tabla form, is presented, along with the results of an online survey comparing it to a professional tabla player's recording on dimensions of musicality, creativity, and novelty. The qaida model generates a bank of rhythmic variations by reordering subphrases. Selections from this bank are sequenced using a feature-based approach. An experimental extension into modeling layer- and loop-based forms of electronic music is presented, in which the initial modeling approach is generalized. Starting from a seed track, the layer-based model utilizes audio analysis techniques such as blind source separation and onset-based segmentation to generate layers which are shuffled and recombined to generate novel music in a manner analogous to the qaida model.

CHAPTER I

INTRODUCTION

This thesis describes an attempt to create a generative rhythmic modeling system capable of generating musical output in realtime. Our emphasis is on rhythmic and timbral components of music, and two quite different applications are discussed in Chapters 2 and 3. We describe a system which creates novel musical material based upon some initial musical seed; this is accomplished by modeling musical structure in terms of abstractions such as functional groupings of events, and the conditional dependencies between them. In Chapter 2, a generative model of qaida, a traditional north Indian solo tabla form, is presented, along with results of a survey comparing its output to that of a real tabla player. The work in this chapter constitutes the primary result of this thesis — the model is developed to the point of reliably producing aesthetically satisfactory output, as judged by both the author of the work and the survey respondents, and a user interface for realtime operation has been developed. Chapter 3 describes an experimental extension of this work, a generative model of a simplified compositional form found in much rhythm-based electronic music. This additional work is primarily concerned with applying the ideas detailed in Chapter 2 to a style of music built largely on the addition, subtraction, and recombination of vertical layers. Extending our modeling approach to this musical context raises a number of technical challenges, and clarifies some underlying assumptions about musical structures upon which the original model relies.

This work is fundamentally motivated by an interest in exploring computational models of creativity. This topic, which can be generally defined as the design of generative models whose output would be deemed creative when judged by the same

standards as human creativity, holds great interest in its own right. A further appeal is the potential for work in this area to engender new modes of performance, and inspire human creativity. The work presented in this thesis, primarily concerned with developing generative models, has been conducted with an aim of ultimately embedding these models in performance systems. As envisioned, these performance systems will include a range of paradigms: interactivity, in which machine listening is used to influence the output of the computer; manual control, in which a performer is able to manipulate higher-level parameters of the model, perhaps being surprised by the particulars of the results; and fully autonomous, in which the computer acts on its own.

A defining characteristic of any sort of modeling is that the design must always define, whether implicitly or explicitly, a level of detail below which the structure need not have any relationship to system being modeled. This is in fact the essence of what a model is: a structural simplification which approximates something observed in the real world, without emulating its every detail. If a model can convincingly mimic observed reality without mimicking all of the underlying processes, then it is considered successful. The models presented here are not based on the physical structure of the human brain, nor on the network of cognitive processes thought to be involved in human creativity. Rather, they are based on analysis of musical structure and on observations about the process of music-making. They can be seen as models of the processes of musical creativity (or rather, a subset of those processes). One important point which follows from this is that ultimately the only way of making valid judgements of whether or not a model is in fact modeling creativity, or how well it is doing at that task, is by assessing the quality of its musical output. In essence, by presenting a system as modeling realtime creativity, we are suggesting that it should be able to succeed in a performative capacity.

Of course, many systems have been designed for performance and composition

which would not truly be considered models of creativity. A complex and unpredictable interactive music system such as George Lewis's *Voyager* [40] may be undeniably creative, but many such systems are specifically designed for a particular musical context, and function more as living musical pieces than true models of creativity. We can identify a number of factors for a system to be considered to embody a model of creativity: a quality of output which at least some listeners judge to be of a high quality, a substantial degree of autonomy, and enough generalizability that it can be applied outside of a singular musical context. The types of performance systems described above as goals towards which this thesis is working should satisfy these constraints, and as stated, the work described in Chapter 3 is intended to test our system's capacity to generalize. From this perspective, computational models of creativity and musical performance systems, while not necessarily equivalent, are deeply related.

A consistent emphasis in the current work is on designing the systems to be capable of realtime operation. This deserves explanation, as it is not in itself a requirement for a generative model. The first reason for this focus is that in general, the types of musical creativity addressed here are more closely related to improvisational forms than "offline" composition in which a composer is able to view the whole product before it is considered finished, and to make edits to the material. This is not an exclusive focus — the work presented in Chapter 3, for example, is just as readily applicable to offline generation — but is a consideration which has shaped some of the design of the models. In some respects this is necessary in order to properly model the chosen subject matter. The work presented in Chapter 2 models an improvised musical form, and the author's own previous experience creating music similar to that discussed in Chapter 3 usually involved some improvisation, particularly in the construction of larger musical trajectories. The second reason to emphasize realtime generation is the planned future work mentioned above, namely embedding the models

in performance systems. Clearly, any interactive system must be able to generate material in response to realtime input; more generally, it is the author's view that a performing computer system will be best called creative when that creativity happens in realtime. This is not a hard requirement, of course, but helps us to draw the very real distinction between a computer which "performs" by playing back a soundfile and one whose output remains undetermined until the moment before it is played.

This is hardly the first work concerned with generative music modeling, but it distinguishes itself in a number of ways. The assemblage of elements and the architecture of their arrangement is unusual, including the design of components such as the two-stage generation/selection process described in Sections 2.4.2 and 2.4.3, which serves both as an effective generative technique and an ad-hoc model of cognitive processes. Audio analysis techniques are used largely for the purpose of off-line learning, but then brought into a realtime context in which the techniques are applied to material that has not yet been played, which can be thought of as a simple model of a computer listening in its "mind's ear". The work described in Chapter 2 applies computational modeling to an area which, to our knowledge, has yet to be approached in this way. The work described in Chapter 3 represents a novel application of the sorts of syntactical variations developed in Chapter 2 to a very different genre, generalizing some of the components to allow underlying algorithms to remain largely the same. Lastly, a great deal of other work in this general area, reflecting the biases of the Western classical tradition, has focused primarily on pitch-based music, in particular on modeling melodic forms; here we focus on rhythm, and its reliance on timbre.

The core of our efforts focuses on designing the models, but audio generation is obviously a central concern. There are various approaches to the challenge of how to take some musical material represented internally in a computer and output it as sound. These range from printing scores of musical notation to be interpreted

by instrumentalists [31], employing robots to manipulate physical objects such as traditional instruments [70, 69, 59], and of course directly generating audio through synthesis or sampling techniques. The system presented here adopts the approach of sample-based audio generation; in the case of the qaida model, samples occupy known categories of drum sounds, while in the layer-based model, the samples are derived from the audio source upon which the model is built.

While we are attempting to build generative models with a high degree of autonomy, the design of the current project deliberately exposes a certain level of realtime control to a human operator. The motivations for retaining handles of control stem in part from my own history as a performer and composer of electronic music. Firstly, as a stated goal for this work is to lay the groundwork for performative systems, an appealing prospect is to allow the system to be playable even while still in development. In a related vein, control of a generative model should be a qualitatively different experience from that afforded by other forms of musical control. Composers and instrument designers have often experimented with alternate interfaces, including brainwave sensing [41], gestural controllers [67], and extended instruments [42], among others. While many of these innovations are obviously distinguished by their unique solutions to various challenges of physical engineering, a common thread among them is that new modes of control open new avenues of expression, and create a different quality of experience for both performer and listener. This is of course not limited to physical controllers; it applies equally to novel approaches to parameter mapping [6, 33, 71]. An ancillary hope for the current work is that it may eventually contribute something to this tradition. A final point to make regarding these handles of control is that a generative model should ultimately incorporate some awareness of its context, for example the pitch content of a human co-performer's melodic solo, into its generation process; this is in fact necessary for some of the interactive performance applications discussed above, and is a recurrent theme in theories of human creativity discussed in

Section 1.1.2. Many parameters of the models could be retroactively made subject to manipulation based on such data, but building in a number of methods for influencing the model's operation in realtime anticipates this usage.

Essentially, the intent of this thesis work is to create a system which lies in an optimal midpoint between two poles: on the one hand, a fully general system of generative models based in abstract statistical modeling and analysis of music and musical creativity, agnostic to style and aesthetic and capable of autonomous operation, and on the other, an idiosyncratic algorithmic composition or performance tool, tuned to the peculiarities of the creator's taste and needs. Thus, a wide array of tasks are addressed, and an emphasis is placed on finding solutions which are sufficient to the extent that they allow the system to create music. Ultimately, the success of a project pushing towards these goals depends largely on the quality of its musical output.

1.1 Background

The work in this thesis touches on a range of fields. Some relevant background is presented below, but a complete treatment of all related works and issues is somewhat beyond the scope of this chapter; the reader is directed towards the cited works for more detailed information. As is to be expected in interdisciplinary fields, there is often a significant degree of overlap between the areas discussed here.

1.1.1 Improvising Machines

Many systems have been developed which can claim to involve machine improvisation. Three key elements can be identified which distinguish improvisatory systems from other generative or performance-based systems such as non-realtime algorithmic composition or interactive systems built on a set of discrete cues: precise output should not be easily predictable based on previous events, decisions must be made within the time constraints of the musical context, and there is no concept of editing or retracting a previous decision. The current work fits these basic principles,

and for the most part is conceived as an improvisational system. Any improvising system must solve various challenges arising from these constraints; the work of two researchers/composers is presented here briefly.

François Pachet has developed a well-known system called the Continuator [46]. Pachet's program is designed to interact with one human musician, and tries to come up with improvisatory responses to that musician's playing. Briefly, the Continuator segments the stream of input notes into phrases, builds a database from these phrases, and then uses this information to generate a continuation of the latest gesture. The continuation algorithm is based on a prefix tree [54] built from the input phrases, and maintains a variable-order memory, allowing the output to better emulate long-term structure than would be possible using a simple first-order Markov model. While the prefix tree is based directly on the received input, the choice of the output is determined by random draws from the set of possible continuations determined by the tree, weighted by their respective probabilities. These probabilities are determined from the structure of the prefix tree, and thus also representing characteristics of the input. This aspect of Pachet's approach is notable; a recurring technique in creative systems is to determine output by choosing probabilistically from some larger set of possibilities. The current work employs similar techniques, although the processes for defining the larger set of possibilities and assigning probabilities in the choice step are not both based on the same information, and neither one is based on realtime audio input.

Arne Eigenfeldt has done some intriguing work in the vein of machine improvisation. His multi-agent "Kinetic Engine" [24], a semi-autonomous Max/MSP patch, models the interactions between networked improvising agents in terms of musical features such as timbre and rhythmic complexity, and social dynamics such as cooperativeness, allowing shared parameters such as tempo and overall contour to be

controlled globally by a "conductor" agent. This model is implemented as a performance system titled "Drum Circle" [25]. Of note is the way in which modeling the social interaction of rhythmic agents fundamentally precludes the notion of redaction or editing of the results: one agent can make a request to another for cooperative interaction, that is, syncing of various parameters, and the second agent may (or may not) respond affirmatively within a certain period of time. It is worth mentioning that he views his work primarily from the composer's perspective, stating that he regards the musical knowledge and intelligence in his systems as an extension of his own compositional tendencies, rather than as a general model of creativity [26]. In addition, Eigenfeldt has experimented with automated generation of electronica, and even published several recordings under the pseudonym "raemus" [27]. Unfortunately, considering the obvious potential relationship to the work described in Chapter 3, he has published little on this particular direction.

1.1.2 Theories of Creativity

Within the field of psychology, there have been many attempts to characterize the basic nature of creativity, incorporating perspectives and data from a wide variety of sources. This is relevant for the current work as a general framework within which to understand what may be meant by modeling creativity, and should help to elucidate a number of issues involved with meaningful approaches to this task and the evaluations of its outcomes.

The first issue is to define more precisely what is meant by creativity. Many definitions have been proposed, representing different philosophical and research perspectives, often in contradiction with one another. Mihaly Csikszentmihalyi [19] outlined a theory formulating creativity as a concept arising from the interaction of certain elements: a *domain*, such as music or a particular musical genre, the *individual* who produces some possibly creative work, and the *field* within which the work is judged.

One significance of this is that it moves creativity from being a purely individual characteristic, to one largely the product of external interactions; notably, the final determination of whether the individual has been creative rests on the judgement of peers. Sternberg [61] describes a number of theories based in the idea that there are multiple creativities. *Geneplore* [30], for example, models creativity as comprised of a generative phase in which a large set of potential materials, e.g. observations on some topic, or melodic fragments, is amassed, and an exploratory phase in which this set is explored and interpreted. There is notable similarity between this and elements of our system described in Sections 2.4.2 and 2.4.3. One of Sternberg's own theories [62] represents creativity in terms of three processes for finding insights in large quantities of information: selective encoding, combination, comparison. The interaction between these elements acts as a kind of introspection: insights found by filtering information, the first process, are combined to generate new insights, which in turn are compared to previous or distant insights to create yet another insight. This set of processes, he posits, defines a form of creativity. Many of these theories share some relation to Gardner's theory of "Multiple Intelligences," [32], concerned primarily with making the case that intelligence is best viewed not as a singular quality, but as a collection of somewhat independent mental properties; Gardner also addresses creativity, characterizing it as fundamentally concerned with the production of novelty within a domain, similarly to Csikszentmihalyi's approach.

More practical but equally valid definitions have focused on the concept of novelty. From this approach, a common formulation defines creativity as an action or process which produces novel output that satisfies the constraints of context [18]. Addressing the basis for judging whether an artificial system could be considered creative, Pereira [49] identifies the requirements that when given a problem, answers produced by the system should not replicate previous solutions of which it has knowledge, and should apply acceptably to the problem. These are notably similar conceptualizations

of creativity, and share the idea that the existence of creativity can, and should, be evaluated on the basis of the product.

1.1.3 Creativity and Style Modeling

David Cope's long-running project Experiments in Musical Intelligence (EMI) focuses on faithful emulations of styles in the Western classical canon [15, 14]. His approach focuses on extracting typical patterns from a large corpus of works, analyzing those patterns to retain those which encode the main elements of the style, and recombining them to create derivative works [17]. The musical creativity modeled in EMI is generally that of the traditional Western classical composer, that is, composition which does not necessarily happen in realtime, and whose output is a written score.

Cope has written and worked extensively in this field, considering his work to be fundamentally concerned with computational models of creativity. He identifies a number of basic elements which he determines to be central to this task, specifically calling out pattern-matching and recombinance [18]. Much of the work presented in this thesis similarly relies on recombinance as a key process for generating novel material; the technique is particularly central to elements presented as generalizable.

1.1.4 Graphical Models

We borrow some visualization and analysis techniques from graphical modeling, a statistical modeling technique which has in recent years become one of the more commonly used methods in machine learning [45, 1]. Graphical models represent interdependent random variables in a way which is relatively easy to grasp intuitively, potentially reduces computation, and may reveal structure in the model more clearly than other approaches. Jordan[37] describes graphical modeling as a "marriage between probability theory and graph theory.... Probability theory provides the glue whereby the parts are combined, ... the graph theoretic side ... provides both an intuitively appealing interface...as well as a data structure that lends itself naturally to the

design of efficient general-purpose algorithms." The core of this technique, and the most relevant aspect for the current work, is a graphical representation of networks of variables as nodes, with connections between them representing conditional dependencies, or causal relationships. Frequently, complex systems can be easily reduced and visualized in this way. Mathematically, an immediate utility of representation in terms of graphical models is that one need not calculate all possible conditional probabilities when evaluating the network; rather it is possible to represent dependencies in terms of variables "upstream" [45].

Graphical modeling has been applied with some success to a variety of musical modeling and analysis problems. Taylan Cemgil has demonstrated applications ranging from analytical tasks such as tempo and pitch tracking [5] to generative tasks such as automatic harmonization of Bach Chorales[4]. Christopher Raphael has implemented score following[52] using graphical models. The primary appearance of graphical modeling in the current work is as a tool for the initial steps of inspection and model design; however due to their broad applicability and intuitive nature, it is intended that future work incorporate more robust statistical models using this technique.

1.1.5 Music Information Retrieval

Music Information Retrieval (MIR) is the field concerned with extracting information from musical audio. The areas of MIR which are most relevant to this thesis have to do with the extraction of timbral features and the identification of perceptually relevant timing information. There is a large set of timbral features commonly known and used for a variety of applications within MIR. A raft of these features are described in detail in [48], and some are used briefly in Chapter 2. The task of identifying higher-level percepts, however, suffers from the fact that the goals of analysis are defined by human perception. In this situation, objective ground-truths are hard to come by, or

may exist only as approximations of an under-determined concept. Research in music perception and cognition is making inroads into some of the aforementioned problems, identifying unexpected commonalities and regularities in human musical perception, and providing bases for quantitative models [34, 22]. However in many situations, extensive manual adjustment of parameters is required to achieve the desired results. This is noted a number of times in Chapter 3.

The MIR work in Chapter 2 is limited to some simple timbral analysis to provide the model with a richer characterization of some of its constituent material. Chapter 3 presents a more extensive use of MIR; this step results from extending the system to an area not conducive to symbolic representation, necessitating that parsed audio material be made available to the underlying model. A good overview of onset detection, including the algorithms which form the basis of the technique described in Section 3.2.3, can be found in [23]. Beat detection was implemented and tested in the system; ultimately it was rejected due to an inadequate level of accuracy, but for completeness, we include a brief reference to the source of the algorithm used. The "context-dependent" beat tracker developed by Matthew Davies [20] attempts to model the dual human characteristic of "locking in" to a beat while at the same time keeping an open ear by listening for changes in tempo and maintaining two beat period and phase hypotheses simultaneously. He presents his work in the context of developing for real-time musical accompaniment, in part by designing the algorithm for the high computational efficiency required in real-time analysis.

Nick Collins [12] has developed an algorithmic beat tracker and "slicer", essentially an amalgamation of many different musical analysis techniques, some well-known and some innovative; our work borrows some technique and inspiration from his, in particular in the approach to segmentation described in Section 3.2.2, and its validity in the context of the genres of electronic music considered in Chapter 3. Collins has released this software as a plugin for the algorithmic music programming language

SuperCollider [43], named "BBCut2" [11, 13], and has used it extensively in his own performance, something which distinguishes his work from that of many others in the field. Tristan Jehan [35] built an integrated suite of analysis tools for the purpose of "creating music by listening." Situating his analysis within the history of algorithmic composition and technologically aided music-making in general, he states that his work, "inspired by some empirical research on human listening and learning, may be considered the first practical attempt at implementing a 'music cognition machine.' " [35] His approach attempts to combine different modes of analysis, from psychoacoustically inspired signal processing to machine learning of stylistic aspects, creating a more comprehensive representation of musical structure. Again, we draw some inspiration from his approach, and make use of some of his techniques for audio segmentation.

Lastly, MIR is closely related to what is known as "machine listening." This is distinguished from MIR in its general emphasis on incorporating musical intelligence, typically for use within interactive musical systems. It is mentioned here because frequent references have been made to the intended future directions of this thesis work, both as an attempt to properly situate it contextually, and as motivation for a number of design features. Many of the more exciting possibilities afforded by the generative modeling we describe will certainly involve machine listening — for example, our qaida model could be combined with prior work on realtime tabla stroke recognition [8] to create a system which could perform as one half of a percussion duet, trading variations and basing its output on the playing of the human musician.

CHAPTER II

QAIDA MODELING

In this chapter we present a generative model of qaida, a traditional North Indian solo tabla form. This was the first attempt within the current work to develop a functioning generative rhythmic model. Work proceeded from well-known and accepted theoretical descriptions of qaida, and operated mostly on a symbolic level. The motivations for modeling qaida are described further in Section 2.3, but broadly it can be said that this form was a promising starting point due to having a well-described structure, and also due to the relative ease with which a compact representation can capture essential characteristics, further described in Section2.1. Since qaida is a theme-and-variations form, there are also fairly clear limitations on the domain in which the machine creativity is expected to operate.

Further, prior work by the author on automatic classification of tabla strokes in a realtime context [8] had built a foundation of knowledge concerning some of the more low-level aspects of tabla. More closely related to the current work, a highly simplified model of simple rhythmic tabla accompaniment was previously implemented in Java and Max/MSP, and was used in a collaboratively composed piece "Slow Theka" [9], a piece for automated tabla, computer audio processing, and *sarod*, a plucked stringed Indian instrument, which was performed publicly [60]. It seems a natural development to develop a more robust model, and let the machine take center stage.

The qaida model is implemented as a system which is capable of generating new output in realtime, operating largely independently, but allowing for manual control of certain global parameters, which may be manipulated to sculpt a compositional arc. It would be interesting, and quite straightforward given the current architecture,

to connect other sources of control to these parameters, such as physical sensors or audio analysis of another performer, but this is left for later work.

Results indicate that the system produces thematically appropriate and novel material. Informal listening suggest that over moderate time-scales, the machine improvisations continue to maintain a sense of novelty. Upwards of three to four minutes or so, the lack of global changes dampens the effect. This can be overcome by judicious manipulation of the accessible parameters, but longer-term form remains an area for improvement. A formal survey was conducted, discussed further in Section 2.5, and encouragingly, results indicate that computer generated recordings were well received.

2.1 Introduction to Tabla

In order to properly present the qaida model, it is necessary to first give an introduction to the instrument upon which it is played. Tabla's particular timbral and musical characteristics are reflected in many key aspects of qaida and other solo tabla forms. More than a distinctive tone color, the physical properties and playing techniques of tabla lead to an idiomatic style which will be difficult to properly characterize without some understanding of its modes of production.

Tabla is the predominant percussion instrument of North India. Despite relatively recent historical origins [63], it is nearly omnipresent in North Indian classical, folk, film, and devotional music. Unlike Western classical music, Indian classical music makes extensive use of percussion, and while there is no shortage of performances and recordings featuring an unaccompanied melodic soloist, a tabla is present in the majority of cases. Indeed, tabla has come to represent the percussive side of North Indian classical music, and is thus central to the genre as a whole.

Physically, tabla is actually a pair of drums, as seen in Figure 1. It is played

Figure 1: A tabla. The drum on the left is the *bayan*; the drum on the right is the *dayan*.

with the hands and fingers, and each drum is associated with one hand. The right-hand drum, called the tabla or *dayan*, is higher in pitch than the left-hand drum, or *bayan*. Both drums are capable of producing a variety of timbrally distinct sounds, ranging from ringing sounds with a clear pitch to short sharp sounds characterized by a high noise content. There are specific striking techniques for producing each of the different timbres, known generally as strokes. A summary of tabla strokes, limited to those used in the model described in this chapter, is shown in Table 2.1. The compact arrangement of the drum heads and the hand positions for the various strokes allows a skilled player to switch rapidly between radically different timbres [7]. The dramatic effects attainable by juxtaposing various stroke types is heavily exploited by most tabla music.

The basic aesthetic which arises from this is not best seen as one of jarring juxtaposition. Instead, tabla music is built on a discrete recombinant musical vocabulary in which flowing sequences can be built through syntactical development. Individual tabla strokes are conceived of as categorically distinct entities; this is reflected in the predominant naming scheme for tabla, in which the various strokes are named using semi-onomatopoeic syllables. However the real level of categorization is a little

higher, based on the concept of *bols*. Many *bols* simply refer to a particular striking technique, but some refer to short atomic sequences, such as *te te*, two sharper non-resonant sounds played in quick succession, or *te re ki te*, another sequence of non-resonant stroke. Importantly, a *bol* may also be a simultaneous combination of two strokes, one on each drum. *Dha*, for example, is produced by playing the ringing stroke known in isolation as *na* on the *dayan*, and *ge*, another deeper resonant stroke, on the *bayan*. The combination is seen to have qualities coming from its components, but nonetheless to possess a distinct identity. This is significant because within this conceptualization, tabla can be thought of as a monophonic instrument. In fact it is often described as having melodic aspects, occupying a space between melody and rhythm. Note that the notion of melody here has nothing to do with a sequence of pitches, but rather a foregrounded rhythmic sequence with an evolving linear character.

Tabla strokes can be grouped according to their timbral characteristics. The most basic grouping divides ringing resonant strokes from sharp and unpitched strokes, referred to as "open" and "closed", respectively. Both drums can produce strokes of each type. The second division separates low, bass strokes from higher-pitched ones, essentially distinguishing the resonant *bayan* stroke *ge* from the ringing strokes produced on the *dayan*. The right-hand drum is tuned to a very clear pitch, made possible by the application of a thin disk of damping material to the center of the drum head – inharmonic partials are damped and the overall pitch is lowered, the net result being a class of strokes with a harmonic partial structure. The resonant stroke *na* is one of the more common examples. A second clear pitch, approximately one whole tone higher, can be produced on the same drum, and has overall less high frequency content. Resonant strokes played on the *bayan* may also have a clear pitch, but the *bayan* is in general not tuned so precisely, may have a slightly enharmonic spectrum, and most importantly, is manipulated with the palm of the heel to modulate the

Table 1: Tabla strokes used in the qaida model. The drum used is indicated, along with basic timbral information. "Ringing" strokes are resonant and pitched; "modulated pitch" means that the pitch of the stroke is altered by palm pressure on the drum; "closed" strokes are short, sharp, and unpitched.

Stroke name	drum used	timbre
dha	compound	ringing *bayan*
dhe	compound	ringing *bayan*
dhec	*dayan*	closed
dhen	*dayan*	ringing *bayan*
dhin	compound	ringing *bayan* and *dayan*
dun	compound	ringing *bayan* and *dayan*
ge	*bayan*	ringing *bayan*
geM	*bayan*	ringing *bayan*, modulated pitch
ke	*bayan*	closed
na	*dayan*	ringing *dayan*
nec	*dayan*	closed
nen	*dayan*	ringing *dayan*
rec	*dayan*	closed
te	*dayan*	closed
tin	*dayan*	ringing *dayan*
tun	*dayan*	ringin *dayan*
tunke	compound	closed *bayan*, ringing *dayan*

ringing pitch. This is a very expressive technique, and gives tabla a sizable portion of its distinctive sound. The subtleties of *bayan* modulation would be an interesting and productive area to model, but are beyond the scope of the current work.

2.1.1 Theka

When the tabla is in its most common role as a time-keeping accompanist to a melodic soloist, it plays what is known as *theka*. This is basically an improvisation fleshing out an underlying rhythmic cycle, adding interest and variation to an underlying repeating pattern. The basic structure is defined by a sequence of stroke types associated with metrical potision within the cycle, however they are essentially abstract strokes, in that one rarely plays *theka* simply as a literal reproduction of this sequence. Instead, these abstract stroke types can be seen as defining the character of the short improvised phrase to be played within each beat location. For example, a common *theka* using the stroke *dha*, *dhin*, *na*, and *tin*. At metrical position associated with *dha*, one would generally avoid playing *dhin*, as that is the abstract stroke type of a different metrical position.

Leaving aside discussion of the variety of commonly used cycles, or *taals*, and their respective *theka* forms — an extensive topic — we currently limit ourselves to *teental*, the most commonly used cycle. *Teental* is sixteen beats long, with an auxiliary subdivision of four groups, roughly corresponding to 4/4 in Western meter. This is the meter used in the qaidas modeled in the current work. An important characteristic of *teental* is a pattern of closing and opening the *bayan*. The first half of the cycle is played using resonant *bayan* strokes, through the first beat of the second half; from beats ten through thirteen, the player damps the *bayan* with the palm of the left hand, effectively removing all lower frequency content; in the last three beats, from fourteen through sixteen, the player re-opens the *bayan*, signaling the approach to the downbeat.

Theka's relatively simple form can be described using a similar approach to our description of qaida, and certain characteristics such as this pattern of opening and closing the *bayan* are shared with qaida. A basic understanding of *theka* should thus help in understanding qaida form.

2.2 Introduction to Qaida

While tabla appears most frequently in the role of accompaniment, there is a rich tradition of solo tabla performance, in which the tabla takes center stage. In this case, the tabla is usually accompanied by a melodic instrument that plays a repeated figure known as *nagma* which occupies the role of a timekeeper. A solo tabla performance typically involves stringing together a series of different compositional forms, interspersing them with *theka*, and can last anywhere between forty-five minutes and a couple of hours. One of the most prominent compositional forms presented in a solo tabla performance is qaida (sometimes written as *kaida* or *kayda*), essentially a structured semi-improvised theme-and-variations form [68]. The term itself means "rule" or "custom", suggesting a formal underpinning; indeed, it is qaida's tendency to adhere to a set of compositional rules that made it an ideal application of the modeling approach described in this thesis.

The theme upon which a given qaida performance is built is taken as a fixed composition. There are a large number of traditional qaida themes, some quite short, occupying e.g. *taals* of eight or sixteen metrical beats, some much longer. Qaida is basically a cyclic form in that performance takes place in the context of a repeating rhythmic cycle, so the duration of the cycle and the qaida theme must generally match. The passage of time through the cycle is accentuated by a similar pattern of closing or damping the *bayan* as is used in *theka*. This serves both to introduce variation into the material and, as with its use in *theka*, to help the listener follow a longer periodicity and glue the whole pattern together into one identifiable unit.

This pattern may be exhibited within a qaida theme, as well as over the course of a set of variations (described below).

The macroscopic form of qaida follows a fairly simple structure: introduction of the theme, usually at a slower tempo, development of variations at an increased tempo, conclusion. Within the main body, variations are presented in a structured manner: a variation is introduced, the theme is reiterated, the same variation is repeated with closed *bayan*, and finally the theme is played again with closed *bayan*, often re-opening it shortly before the end of the cycle. This alternation of repetition and variation helps to give the qaida a sense of coherence. New material is emphasized by its presentation at the start, and the listener's awareness of the theme is frequently reinforced. Repetition of each variation invites one to hear it as possessing some compositional weight, that is, to hear it as a structural whole, rather than an arbitrary string of strokes. Finally, as mentioned, superimposing the timbral pattern of *bayan* damping groups the whole set together.

2.2.1 Variations

While qaida themes are part of the shared repertoire of solo tabla, and thus learned by the tabla player prior to performance, variations are improvised according to some basic principles. There are a number of known approaches to variation generation. Perhaps the most important guiding principle of qaida variations, however, is a restriction on the material. Only *bols* which appear in the qaida theme may be used in the variations. This is intended to preserved the essential character of the given qaida. Limiting the vast space of possible improvisations in this way immediately imposes some structure; it introduces a simple conditional dependency.

Given this limitation, one common and effective variation technique is to rearrange subsections of the theme. Qaida themes have internal structure, and are often heard as a series of "natural" subdivisions. The partitions correspond to short characteristic

sequences of strokes. This procedure provides a clean, though partial, solution to a central problem in structured creativity, whether by machine or human: how to generate musical material which is novel, yet retains a clear relationship with the source material or context. As all of the qaida partitions can be moved about, and may be repeated, there are a tremendous number of theoretically possible variants on the theme, using just this technique.

Another class of variations is derived from doubling the tempo of a set of subsections. Typically, partitions played at double-time will be repeated consecutively in order to fill the same metrical duration as their original form.

2.2.2 Tihai

To end a qaida, the performer plays what is known as a *tihai*. Briefly, this is particular type of rhythmic figure which stands out from the preceding material, and dramatically emphasizes the end of the cycle (and the qaida). There are many possible *tihai*'s, but the most basic form is comprised of a rhythmic figure, known as the *pala*, which is repeated three times with a short pause between each iteration, and timed such that the last stroke falls on the downbeat of the next cycle. The *tihai* can start at any metrical location within the cycle, so the *pala* is unlikely to line up neatly with regular divisions of the main cycle. The effect is a tension between the internal repetition of the *tihai* (accentuated by the pause), and the progress of the main cycle, resolved by the two coming into phase on the downbeat.

2.3 Why Qaida?

Qaida was chosen as a form to model due to a number of characteristics which lend themselves to formal modeling. Structurally, some aspects of qaida are quite simple. This is not to say that qaida is in any respect a simplistic or even simple form of music; on the contrary, tremendous musical and perceptual complexity is built up through systematic applications of basic principles. Ideally, of course, the deep knowledge

and musicality of the performer is expressed through the music with a subtlety that is hard to imitate on a computer, but many essential characteristics of qaida can be captured through the modeling approach described here.

Specifically, qaida is attractive as a subject for the following reasons. Perhaps most importantly, as qaida is a prominent form in a classical tradition, there is a known and well-developed theory. While traditional approaches to pedagogy in Indian classical music tend to emphasize immersion and a guided process of discovery, there is an equally strong emphasis on gaining knowledge of and respect for fundamentals of the style.

Further, there are canonical forms. It is for example possible to refer to masterful examples of qaida performance that are almost without exception acknowledged as such by the broader community of tabla players. Similarly, it is possible to make judgements of quality, when comparing different performances or recordings, that have some semblance of objectivity. Conversely, different styles of qaida can be understood as being systematically distinguished from each other, belonging to one or another *gharana* (stylistic school) or era. Without these elements of this musical culture, it would be substantially more difficult to parse the complexity of the actual music.

Qaida is essentially a monophonic music. Acoustically, this is a debatable claim, as evidenced by the fact that tabla consists of two separate drums, often played simultaneously. The important point, as described in Section 2.1, is that within the Indian Classical tradition, tabla is predominantly conceptualized as a single stream of timbral syllables. Tabla recitations, in which a fixed composition is recited using the stroke names prior to playing, make full use of this notion; internalizing these vocalizations is often thought to be central to the process of learning tabla. Considering qaida to be monophonic, then, is both consistent with a likely mental representation in musicians, and allows analysis to operate on a manageable entity, a sequence.

Lastly, qaida is easily transcribable, allowing us to begin from symbolic material.

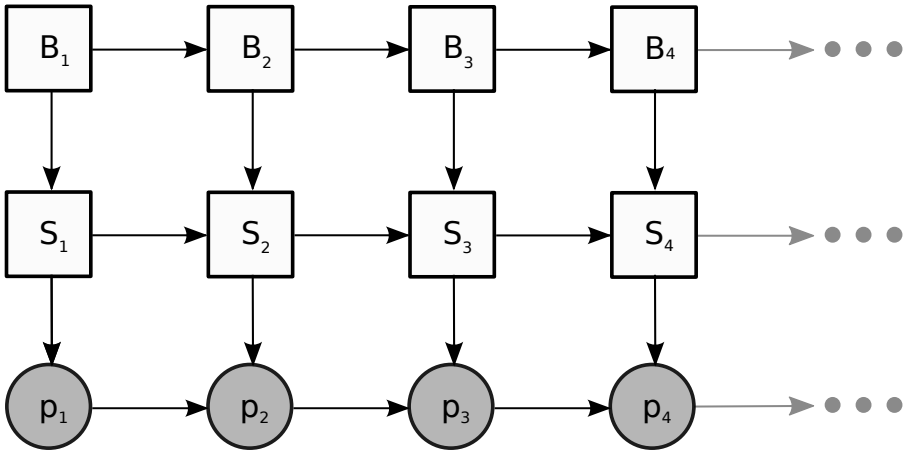

Figure 2: Structure of *Theka* in graphical form. Nodes labeled B represent beat positions within the cycle, nodes labeled S represent the abstract stroke type associated with that position, and the bottom row of nodes labeled p represent the output pattern. Note that the p nodes are shaded circles, to represent that they are observed.

This obviates any requirement for extracting structure from an audio signal, and facilitates "jumping in" to the process of abstract modeling. Of course this does not imply that signal analysis is therefore irrelevant; there are many ways in which machine listening can be combined with a model based on symbolic data, and integrating a variety of those would make interesting developments.

2.4 Methods

The generative model described in this chapter is designed to produce music which follows qaida form as outlined in Section 2.2. The initial step was to revisit the defining characteristics of qaida, undertaking an analysis geared towards designing a model. The abstract representation which we develop here forms the guiding principle of the implementation described later in this section. In this step we employ a visualization and reduction technique borrowed from graphical modeling.

Before addressing qaida, however, we present a similar analysis of *theka*. Describing this simpler form should clarify the subsequent presentation of our qaida model,

but is relevant here also because it was the initial inspiration for qaida modeling. Figure 2 shows a simple graphical representation of *theka*. This diagram may not initially seem to add much, but the key point is that it emphasizes that the final observed output, in the bottom row, is conditionally dependant on an abstract stroke type which is itself dependant on position within the cycle (or equivalently, on the previous abstract stroke type). This led to the development of a simple model of *theka* which was used in the piece "Slow Theka" mentioned in the beginning of the chapter. That model utilized a small bank of possible sequences associated with each abstract stroke type, each with a duration of one beat. One was selected at each beat, producing a reasonable version of *theka*.

An overview of our model of qaida is depicted graphically in Figure 3. One iteration of the main repeating section, between the initial exposition of the theme and the concluding *tihai* is shown. It can be seen that it is similar in style to Figure 2, but the the structure it represents is more complex. The audible output of the four basic stages of presenting a qaida variation are shown, represented by p_1 through p_4 in the bottom row; they are also clearly shown to be dependent on the more abstract "form states" F_1 through F_4. Particularly useful, though, is that this analysis clarifies conditional dependencies of the final output on the time position t and the qaida theme T. The *bayan* opening/closing pattern is represented as a switching state, dependent only on t, while the alternating theme and variation depends both upon t and T. The theme is chosen once, at the beginning of the qaida, while the time progresses both cyclically and linearly. Having clearly identified these relationships, the definition of the precise dependencies occupies rest of the work of building the model.

The most important relationships, of course, are the dependencies of the variation on the theme and the time — defining these, and implementing a system to realize them as audio, forms the bulk of the work presented here. The dual dependencies

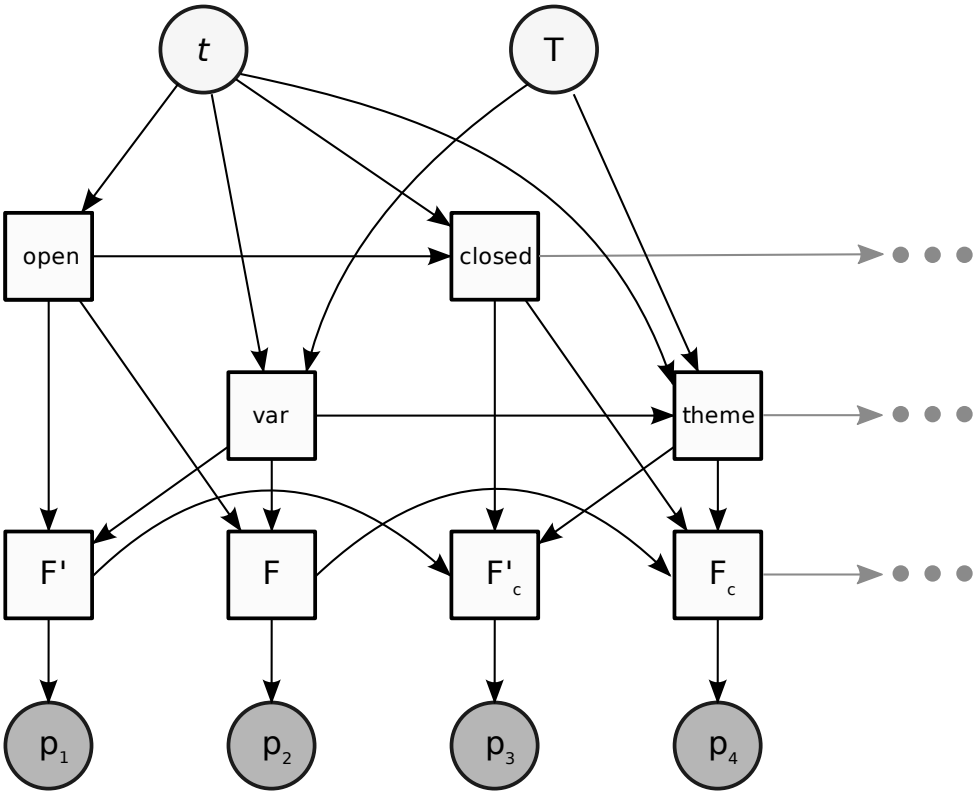

Figure 3: Structure of *Qaida* in graphical form. The nodes labeled t and T represent beat position within the cycle and the qaida theme, respectively. Nodes labeled F represent "form states" (variation *vs.* theme, open *vs.* closed). As in Figure 2, the bottom row of nodes labeled p represent the output pattern.

are modeled separately, as a method for generating style-appropriate variations, described in Section 2.4.2, and a method for choosing a particular phrase to output at a particular time, described in Section 2.4.3.

The method used to generate variations echoes that used by human players. This element of the model can be seen as modeling a *process* of creativity; constrained by its relation to the other elements

The resulting generative model of qaida encodes these structures, and is implemented as a system which generates qaida in realtime, responding to user input. The core of the system was coded in Python [65], relying on the NumPy and SciPy [36] packages for performance intensive computation, and to facilitate manipulation of data structures. Audio output was generated using the Pd-extended 0.40.3 version of Pure Data (Pd) [50, 51, 47]. Communication was handled in realtime between Python classes and Pd using the OSC networking protocol [73].

The Python code handled the process of generating new material based upon the chosen theme. A Pd patch was responsible for controlling the larger-scale form and content of the generated qaida, implementing a simple model of the alternating sequence of theme and variation groups, and sending messages to the Python code requesting variations and specifying a profile of the desired characteristics of the variation.

The basic approach is to build a large database of potential variations through a stochastic process, and to select from that set based on certain criteria. This bears some semblance to technique known in algorithmic composition as "generate-and-test" [53], in which the output of some generative procedure is tested against a set of criteria. However, our method in this work is somewhat different from the standard "generate-and-test" paradigms, in which the criteria are often either constraints, such as fitting the rules of counterpoint, or judgement of the composer. In our case, the criteria are treated more probabilistically, as a basis for the system to make a choice

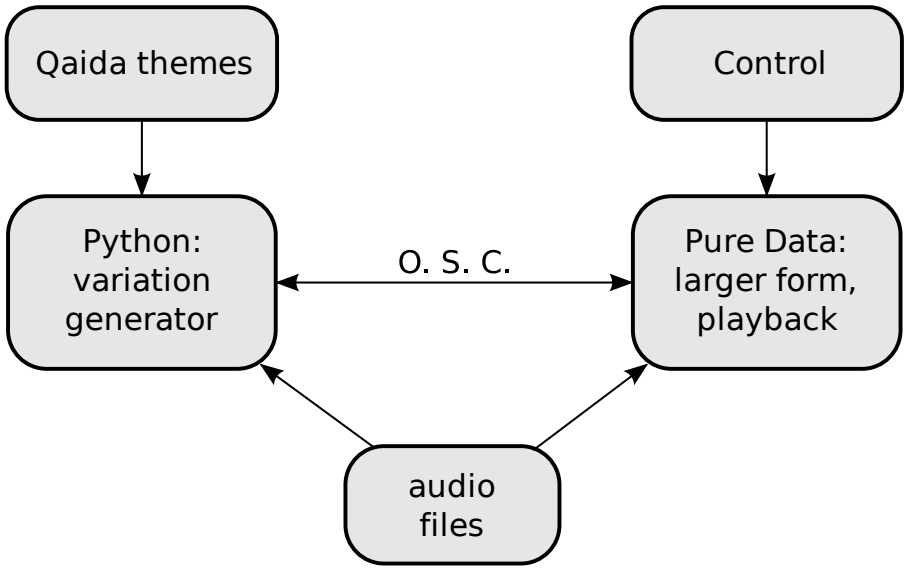

Figure 4: Structure of the *Qaida* modeling system architecture. The two basic components, written in Python and Pd, are depicted; communication between them is handled via OSC.

with a some indeterminacy but weighted heavily towards a desired outcome.

Many of the procedures adopted employ weighted randomness and probabilistic choices as central components. This approach is used frequently for a number of reasons. Firstly, the fundamental nature of creativity and improvisation demands that output not be easily predictable, but have the potential to surprise. This is not to equate randomness with creativity [18], but to emphasize that a certain degree of indeterminacy is central to these domains. Secondly, weighted randomness has a long tradition of use in algorithmic compositional techniques and other creative modeling work, some cited in Section 1.1. Lastly, and importantly, the structure of the model is in the definition of its elements and their mutual dependencies. A property of a generative model of this form, well-known within the generative music community through experience with Markov models, and characteristic of graphical models in general, is that random sampling from the model creates output with the structures which that model describes.

2.4.1 Symbolic Representation

Sequences of tabla stroke were represented symbolically, as pairs of stroke-name and metrical duration. This roughly corresponds to the more common forms of traditional tabla notation, encoding the same basic information. Typically, the primary purposes of tabla transcriptions used by tabla players are pedagogical, mnemonic, or archival, so short and compact representations are the norm. Notation serves as a tool to facilitate detailed study, and most examples consist of excerpted sequences; complete written reproductions of long performances are rare. In our system, the tabla sequence data format has two primary uses: as a format for the long term storage of qaida themes and their subsequent manipulation "behind the scenes", and as the primary information driving the playback mechanism. This minimal representation is appropriate for representing qaida themes, as it includes only the information present in traditional transcriptions – the system cannot "cheat" by reproducing verbatim the expressive timing or sound production found in some carefully selected recording of a qaida theme. On the other hand, exclusion of all manner of subtleties in our symbolic representation creates some ambiguity when it comes to generating satisfactory musical output, giving no indication of *how* the strokes are to be played. Because of this, some effort is made to develop a one-to-many mapping at the last stage, adding some nuances of timbre and amplitude before producing the final audio output.

Consistent with the fact that qaida themes are not themselves improvised, and rarely even composed by the performer, no attempt was made to generate new thematic material. Instead a number of themes were transcribed manually, and annotated with partition bounds. Metrical durations were expressed in fractions of a beat. A bank of of these traditional themes is stored in XML format. The file is loaded from disk at program start-up, and one theme is chosen which remains the only source material for the duration of the qaida improvisation. An example of a qaida theme in this format is shown here, truncated to one half of its full length:

```
<phrase>
    <sequence>
        dhin 0.5
        te 0.5
        na 0.5
        ge 0.25
        tun 0.25
        te 0.25
        ke 0.25
        na 0.166666666667
        ge 0.166666666667
        na 0.166666666667
        dha 0.25
        dha 0.25
        ge 0.25
        tun 0.25
    </sequence>
    <partitions>
        0 5
        5 11
        11 14
    </partitions>
</phrase>
```

No attempt was made to apply any sophistication to the initial choice of theme: it can be specified manually or chosen randomly. This would, however, be worth addressing in future work, as full tabla solo performances are typically comprised of a sequence of many different tabla forms over the course of an entire concert, including several qaidas, with the only break between being a section of *theka*.

2.4.2 Variation Generation

The procedure for generating possible variations on the qaida, together with the process of phrase selection outlined in Section 2.4.3, arguably forms the core of this system. An overview of these components is shown in Figure 5. A mutable bank of phrases is generated from the theme by applying fairly general transformations which are consistent with qaida theory, and then stochastically applying another set of operations to the results of these transformations in order to bias the population towards more stylistically appropriate content.

The size of this phrase database is set by parameter. Clearly, a larger database is preferable as it will contain a greater diversity of material, up until the point at which its contents become redundant. However, the feature extraction and phrase selection processes described in Section 2.4.3 scale with the size of the database, and within the current architecture, they are required to operate in perceptually "zero" time – a delay in processing will result in a delay in audio output. Fortunately they run quickly enough that with a bank of several thousand phrases, there is rarely any perceptually noticeable delay; a bank of two thousand was used during much of the development process, and it was qualitatively found that this size contained sufficient phrase diversity to support varied and novel output.

There are two main transforms used, and a value is stored which represents the relative probability that one method will be chosen over the other. Running in a loop until a bank of the specified size has been constructed, a random value is compared against this probability, and the corresponding method is applied. The most commonly applied method, i.e. the one with the higher probability, is a shuffling of the partitions of the theme, allowing elements to repeat, that is, sampling with replacement from the set of theme-partitions. Phrase partitions are not generally of equal length, so there is no guarantee that a sequence generated by this procedure will have the same total length as the theme (an obvious requirement since it will be played over a rhythmic cycle). A summation is taken over the metrical lengths of the chosen partitions and mismatching phrases are discarded – this can be seen as a crude initial fittingness test. To avoid unnecessary calculation, the number of

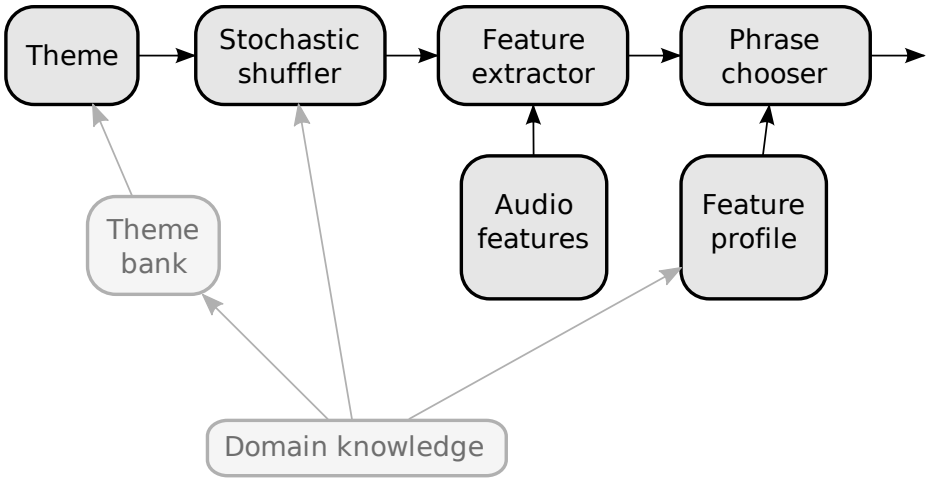

Figure 5: Overview of the *Qaida* variation generating architecture. The theme bank is greyed-out because the choice of theme is made only once, initially. Domain knowledge, specific knowledge about qaida and tabla, is shown being incorporated at specific points in the process.

draws, or partitions in the new phrase, k, is limited to the range within which generated phrases of the required length are possible. Nonetheless, the number of possible sequences which can be assembled from a set of n partitions chosen k times is n^k. The qaida themes in our database are eight beats long, and typically contain between twenty and thirty strokes grouped into six to eight partitions, which range between .5 and 1.5 beats in duration. These numbers make an exhaustive search through this space of variations computationally intractable – a worst case scenario could see us trying to enumerate $8^{16} = 2.81475e + 14$ possible sequences. This was one motivation for separating out the complementary processes of generation and selection. Rather than adopting a brute force search for a phrase of a desired type, the generation process "pushes" material to the selection process.

The less common basic transformation is simply to double the tempo of a randomly selected partition, biased toward the beginning of the phrase. The double time partition is repeated twice in order to occupy the same total length. This procedure is somewhat less probable that the first; best results were obtained setting the parameter close to .1, or adjusting it based on that specific qaida theme. Like the primary shuffling transform, the doubling transform is easily represented in a form that does not require detailed knowledge of the material it is operating on. Both procedures require only that the material be a discrete

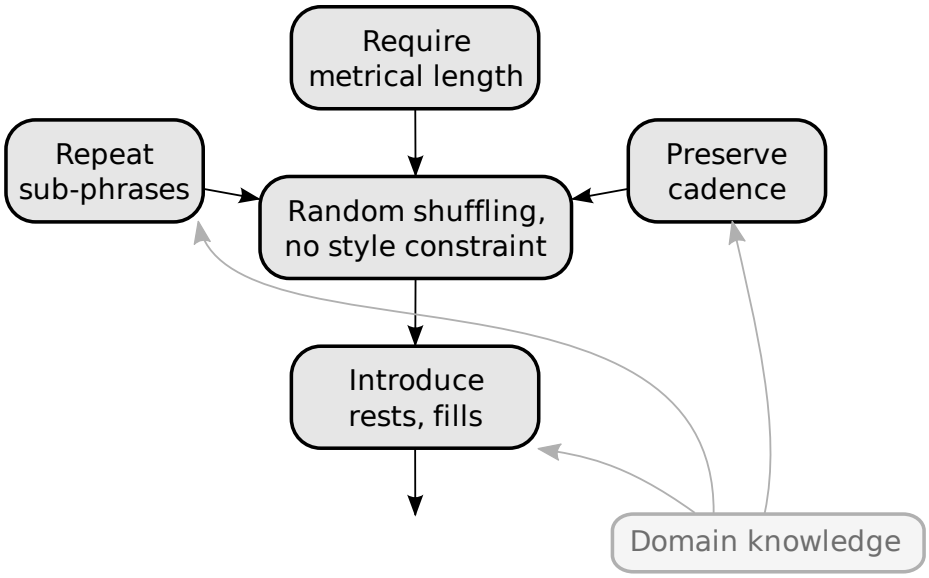

Figure 6: Detail of the *Qaida* variation generating architecture. The reordering routine is depicted here, constrained by metrical length and incorporating domain knowledge in the form of added transformations.

temporal sequence. At the same time, these generative methods are implementations of the two primary qaida variation techniques described in Section 2.2.1, adhering to the basic requirement that the *bols* in the variations be found in the theme,

Another perspective on the re-ordering transform is to see it as a context-switching operation. Each chosen sub-phrase is placed into a different timbral and rhythmic context, altering its basic musical character. It will have new neighbors, timbrally (stroke type) and rhythmically (stroke timings), it may occur at a different overall position in the cycle (e.g. first half vs. second half), and its placement relative to the underlying pulse will likely have changed. New perceptual groupings may occur, as, for example, if several phrases with similar patterns and timbres are placed adjacent to each other, fusing into a single perceptual rhythmic unit. A relatively straightforward, unsyncopated chunk make take on a very different character if placed a half or quarter beat away from the nearest tactus location.

An additional set of four transformations were implemented, with the intention of introducing a bias in the resulting phrase bank, making it more likely to include style-specific elements. They are intended to favor:

- Multiple occurrences of the same partition (non-consecutive repetition)
- Consecutive repetitions of a partition
- Preservation of the final partition (cadence)
- Introduction of short rests, or the omission of strokes

Following the application of one or another of the primary transforms, a set of random numbers is tested as before against a set of parameters corresponding to the probability that each of the first three will occur. The cadence preservation is by far the most likely, with a value of .8 — the others were found to be most effective when set around .1 and .2. These tests are conditionally independent, and somewhat naive, the probability values arrived at by adjusting over repeated listening tests, and are simply intended to represent specific stylistic tendencies. The fourth operation is generally applied at a later stage in the qaida; the introduction of space is essential to breaking the homogeneity which tends to emerge over time, but can also disturb the coherence of a phrase, and so is reserved for use in the more "complicated" sections of qaida development. These operations are shown in context in Figure 6

The phrase bank is described as mutable in the beginning of this section because it is possible to continue to selectively regenerate some fraction, or to reapply the other transformations iteratively. In the current work, this has been implemented in a fairly simple way, and it is easy to deviate too far from musical coherence in favor of novelty. However, this capability suggests interesting avenues for future work, for example in incorporating more intelligence into the probabilistic application of these transformations. One can imagine a system based on the current one in which the phrase bank is continually evolving, representing a distribution of phrases conditioned not just on the stylistic form and the choice of theme, but also on the relative position within the compositional arc of development, on the previous output, and even involving a model of expectation in order to move the contour of density and syncopation towards a climax.

2.4.3 Variation Selection

Selection of a phrase from the bank of variations is the complement to the construction of that bank. This step is performed using input in the form of a feature request. This request triggers a chain of events: phrases in memory are compared against the request, a close match is selected, and finally a single phrase is returned for playback.

Immediately after the phrase bank is first built, features are calculated over each phrase in the set. It was found that a relatively small set of features could provide a surprisingly flexible handle into the character of the returned phrases, though a larger set would no doubt improve the range of performance. The currently calculated features are

- Distribution over each stroke type, by frequency of occurrence
- Distribution over each stroke type, by time (scaled by duration)
- Rhythmic density
- Ratio of open to closed strokes, by frequency of occurrence
- Ratio of open to closed strokes, by time (scaled by duration)
- Spectral centroid
- Spectral spread

Note that these are not all of equivalent dimensionality — rhythmic density, open/closed ratios, and spectral centroid are scalar values, while the distributions over stroke types are vectors. Even at this level, it can begin to be difficult to intuit the relationship between various combinations of values for these features, and the types of corresponding phrases. Futur work should include developing aggregate features which are more intuitive and mutually independent.

For the most part, these are in effect timbral features; they are meaningful because of the fundamental relationship between different stroke types and their timbral characteristics. The spectral centroid and spread, however, require more explanation. The feature itself is

uncomplicated. Spectral centroid is simply the weighted average of the magnitude spectrum, defined as [48]

$$\mu = \frac{\sum_{k=0}^{N-1} kX[k]}{\sum_{k=0}^{N-1} X[k]} \tag{1}$$

where X is the magnitude spectrum, k is the frequency bin number, and N is the length of the spectrum, and spectral spread is the variance of the spectrum:

$$\sigma^2 = \frac{\sum_{k=0}^{N-1} (k-\mu)^2 X[k]}{\sum_{k=0}^{N-1} X[k]}. \tag{2}$$

Spectral centroid is computed on audio, however, and up to this point we have been dealing with symbolic data only. However, the sequences are intended for playback on a known set of sounds, so in this step we calculate average values over a large audio database of segmented tabla strokes which is also used in playback, and calculate the values we would. The net result is that by "looking ahead" to a destination form of the given phrase, we can obtain a quantitative estimate of the hypothetical timbre of the phrase. Timbral features are relevant here not only because of their obvious effect on the character of the resulting sound, but also because basis of tabla's rhythmic vocabulary is the temporal sequencing of contrasting timbres.

The Python object responsible for calculating this feature set also maintains a connection to an OSC server object which is listening for control messages coming from Pd. Aside from handling various commands for initialization such as choosing a starting theme (randomly, or specified by an integer argument) and constructing the phrase bank (no argument passed), the OSC server's most important function is to handle messages requesting playable data, that is, sequences of stroke names and metrical durations. The three types of playable data, requested as needed at the appropriate moment in the qaida, are the qaida theme, a variation, and a *tihai*. Serving up the theme is simply a matter of packing the theme into an OSC message and sending it; *tihai* construction is detailed in Section 2.4.5. Request for a variation is a little more complicated. When Pd sends a request for a variation

phrase, the data in the OSC message consists of a set of feature preferences that describe the desired type of variation, which are passed as arguments to a method in the Python object. Specifically, this can describe any subset of features, specifying three values for each, representing the target value, a relative weighting for this feature, and a "flexibility" measure.

The range of feature values for a given bank of potential variations is largely dependent on the initial choice of qaida theme, as well as on the particulars of how the randomly chosen alterations happen to have occurred on that run. Therefore, the target parameters of the feature request, expressed in the range 0 to 1, are normalized to the range of the current variation bank. For example, one run of the variation generator module produced a bank with a "density" value ranging from a minimum of 4.0 to a maximum of 7.25, leading to a mapping function of

$$xnorm_k = (\max(\bar{f}_k) - \min(\bar{f}_k))x_k + \min(\bar{f}_k) \Rightarrow xnorm_k = (4.0 - 7.25)x_k + 4.0 \quad (3)$$

where x_k is the un-normalized preference for feature k, $xnorm_k$ is the normalized feature preferences, and $\bar{f}_k$ is the vector of values for feature k over the whole bank.

The flexibility parameter functions as a sort of distance metric, defining the width of a Gaussian curve onto which a linear distance is mapped. The Gaussian is centered on the target value, and is used as a look-up table to get the unweighted score for that phrase and feature. This provides a simple way to specify how strict a given feature preference is, independent of the relative weighting for that feature.

After each phrase in the bank of variations is compared to the feature request and a match score calculated, a final choice is made based on this score. Rather than always choose the best match, which would lead to a deterministic output, and require either constant change in the feature requests or frequent regeneration of the phrase bank, the choice is made probabilistically. The two most successful algorithms were to rescale the probabilities to emphasize the higher-scoring phrases, or to take the set of top scorers and make a choice among those based on their normalized probabilities equivalent to setting the probabilities of low-scoring phrases to zero. Again, this procedure serves as a way to

balance the creativity and novelty of the system's output with its responsiveness to the demands of context.

2.4.4 Macroscopic Structure

The macroscopic structure is simpler, and largely deterministic, following the basic qaida form outlined above. Playback is implemented in Pd, and is described further in Section 2.4.6. The patch controls the alternation between theme and variation, requests variations from the Python generator, controls the periodic opening and closing of the *bayan* strokes, and generates the audio. Each stroke type has a set of associated audio samples, and output is generated by selecting randomly from this set, scaling amplitude according to stroke type and duration. An accompanying *nagma* marks the cycle. Feature preferences for the variation requests are specified manually with a set of sliders. Modeling of longer-term structure is minimal, chiefly limited to initial exposition of the qaida theme, allowing tempo transformations, and requesting a *tihai*; the manual controls provided allow a user to take the place of a fuller model. It should be noted, however, that the user need not be highly skilled, or even particularly knowledgeable with respect to tabla or qaida.

2.4.5 Tihai

Qaida form concludes with a *tihai*, described in Section 2.2.2. A minimum *pala* length is defined, with a default of four beats. The *pala* is constructed by selecting and concatenating theme partitions until this minimum length is reached. A simple way of building a more complex *tihai* is simply to increase the minimum length. An optional parameter to scale the durations of the strokes is provided, allowing the *tihai* to achieve a dramatic and virtuosic quality typical of real tabla performance. The *pala* is repeated three times, with a pause inserted between iteration. The start point is determined by the length of the constructed *tihai*, and a short rest is inserted just before. The *tihai* is unmistakable, but this is due primarily to the rests separating the *pala* repetitions; a short pause or other device is necessary to set it off from the preceding material.

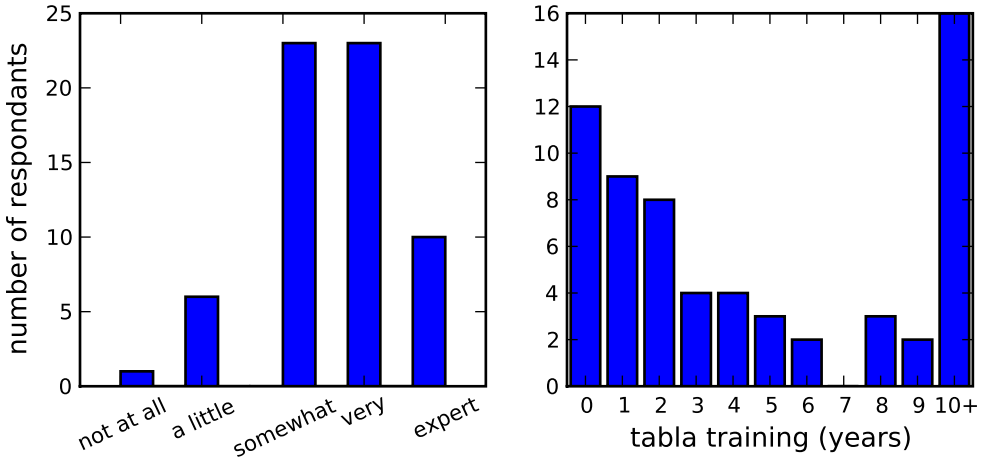

Figure 7: Overview of respondent demographics. The figure on the left shows a histogram of "familiarity with tabla music," and the one on the right a histogram of years of tabla training. It can be seen that both measures indicate that the respondent base is knowledgeable in this genre. In particular, 16 out of 70 were highly accomplished tabla players with more than 10 years of experience.

2.4.6 Audio Output

Synthesis of the generated qaida was accomplished using high-quality isolated samples of tabla strokes, played by a professional tabla player and recorded specifically for this project. Care was taken to obtain several timbrally similar samples for each stroke represented in the qaida theme database. Each playback command caused one of the samples for the given stroke to be selected, helping to achieve a slightly more natural quality. Amplitudes were scaled by durations, to mimic the lighter touch that is generally used when playing fast sequences. For the most part, this scaling was important only in sections involving the fastest sequences, which otherwise sounded notably unnatural; moderate durations (e.g. one beat *vs.* one-half beat) proved less sensitive to this nuance. The quality and consistency of the recordings was reflected in the audio output; the only significant shortcoming remains a lack of *bayan* modulation.

2.5 Evaluation

An online survey was conducted, in which three recordings of generated output were presented along with two recordings by a world-class tabla player, without indication of the origin of the recordings; participants were simply asked to make a series of judgements, unaware that the survey involved comparison of human playing and computer modeling. The survey can be found at `http://paragchordia.com/survey/tablasurvey/`, and the audio clips of both computer-generated output and professional tabla performance can be heard separately at `http://www.alexrae.net/thesis/sound/`, numbered 1–5, the first three being the qaida model's output, as in the results presented here. The recordings of model output were "played" via the user interface implemented in Pd, and were recorded without subsequent editing.

A total of 70 participants responded to the survey. A majority claimed moderate to high familiarity with tabla music, and many reported themselves to be practicing tabla players; distributions of familiarity and training are shown in Figure 7. The mean age was 35.2, with a standard deviation of 12.2. The gender of the respondents was highly skewed: only two (3%) were female. The order of presentation of audio segments was randomized, and participants were asked to rate the examples along several dimensions, with the goal of comparing relative judgements of the computer- and human-generated examples. The judgements were on a scale of 1 to 7, reflecting answers ranging from "very little" to "a lot", except in case of the last two questions, phrased as ranging from "poor" to "excellent". A higher value corresponded to a more favorable judgement. Additionally, respondents were invited to supplement their quantitative judgements with further comments. Results show that the qaida model's output fared quite well in comparison with the professional recordings.

Participants were asked the following questions:

1. To what extent would you say that this recording demonstrates a feeling of musicality?

2. To what extent would you say that this recording demonstrates musical creativity?

3. To what extent would you say that this recording adheres to qaida form?

4. To what extent would you say that this recording is novel or surprising, given the qaida theme?

5. To what extent would you say that the improvisations in this recording are appropriate to the style and the theme?

6. If told that this recording were of a tabla student, how would you rate his/her overall TECHNICAL abilities?

7. If told that this recording were of a tabla student, how would you rate his/her overall MUSICAL abilities?

Analysis of the quantitative data was undertaken using a test for statistical significance corrected for multiple means. Figures 8–14 show results of this analysis, comparing the five audio excerpts for each question. Each figure shows mean values and confidence intervals (using $p < 0.05$) of the judgement scores for each audio segment. A trend is visible in the average values of the data across the examples, showing the computer generated output to be rated slightly lower than the human generated excerpts. However, the differences do not reach statistical significance given the sample size, except in the case of the third generated qaida, which in many cases is rated somewhat lower than the other model outputs.

These results are encouraging: the computer-generated qaida performed quite well in comparison to very high-quality human-played examples. Further, it is worth highlighting two visually apparent trends, even though the differences are slight. Judgements of musical creativity, question 2, are notable, as two of the qaida model's outputs were ranked on par with the human performer. The model fared similarly well on judgements of novelty.

It is also interesting to note from the comments that many respondents remained unaware that three of the examples were computer-generated. One, for example, wrote in response to example 3: "Again this recording demonstrates that the Tabla player has excellent abilities in playing the right drum with crisp tonal quality. Left drum (Baya) needs some improvement as I stated in the first two Qaidas." As with many, this respondent was clearly influenced by the timbral qualities of the playback samples, positively for those which are well-represented by isolated samples, such as *na* and other strokes played on the

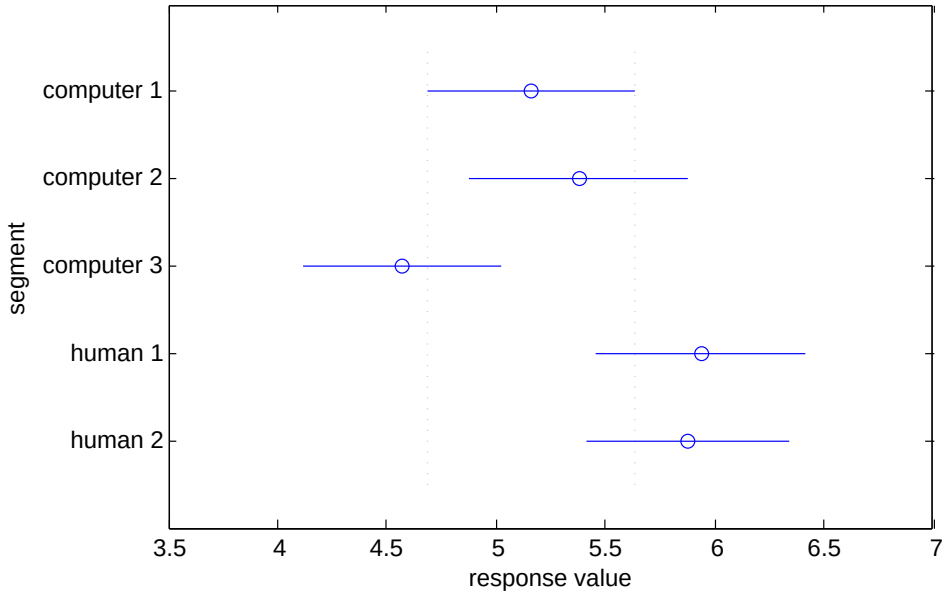

Figure 8: Plot showing mean values and confidence intervals for responses to Question 1: "To what extent would you say that this recording demonstrates a feeling of musicality?" Audio excerpts 1-3 are computer-generated.

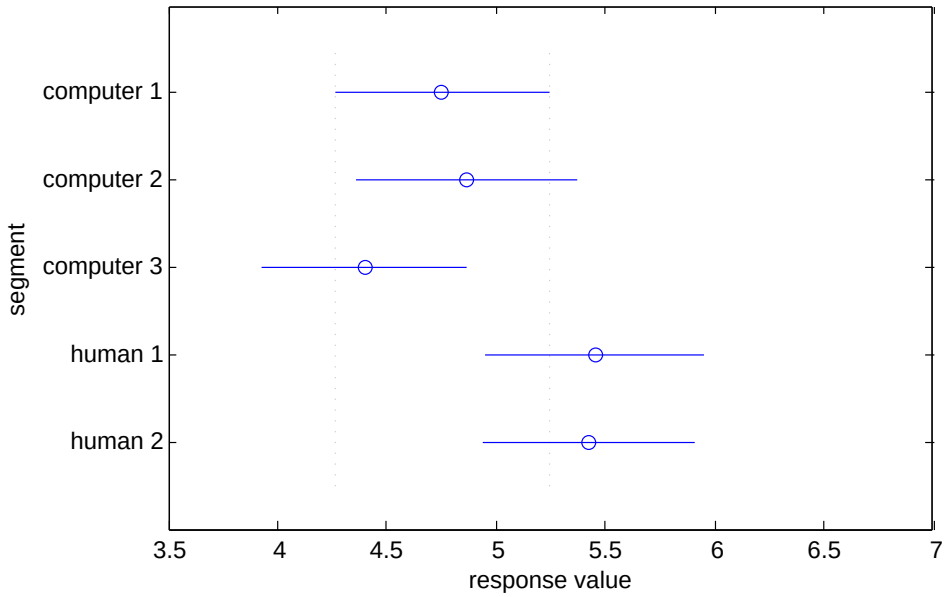

Figure 9: Plot showing mean values and confidence intervals for responses to Question 2: "To what extent would you say that this recording demonstrates musical creativity?" Audio excerpts 1-3 are computer-generated.

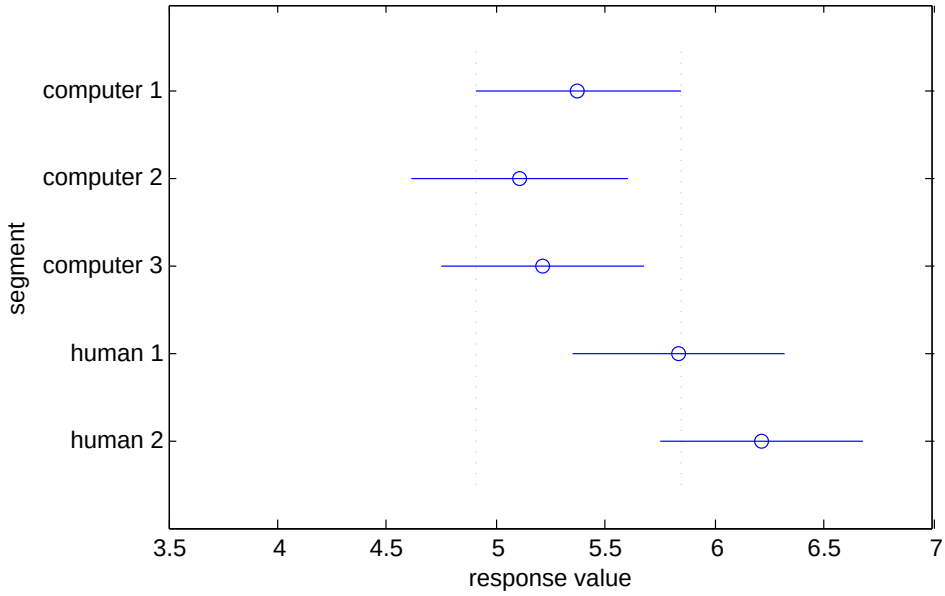

Figure 10: Plot showing mean values and confidence intervals for responses to Question 3: "To what extent would you say that this recording adheres to qaida form?" Audio excerpts 1-3 are computer-generated.

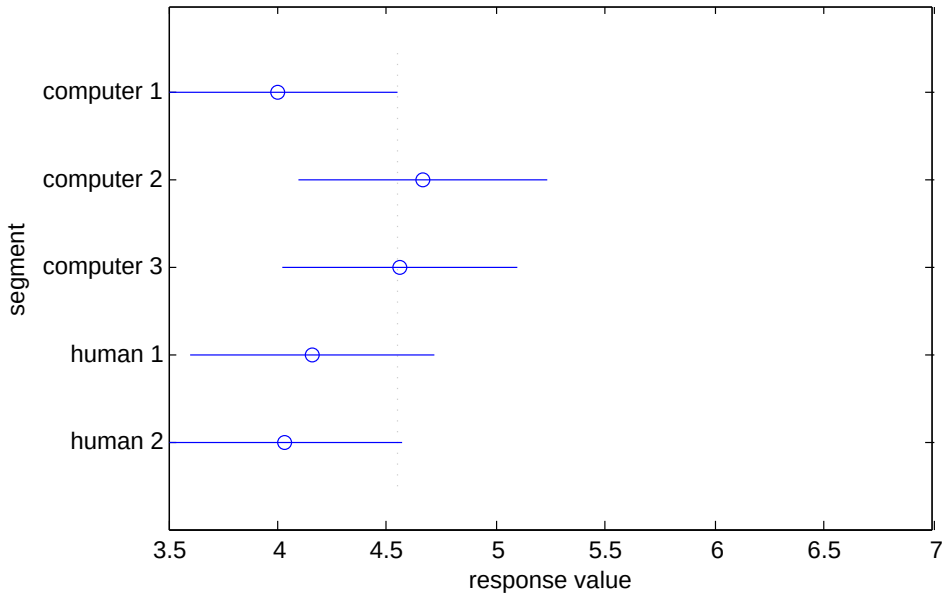

Figure 11: Plot showing mean values and confidence intervals for responses to Question 4: "To what extent would you say that this recording is novel or surprising, given the qaida theme?" Audio excerpts 1-3 are computer-generated.

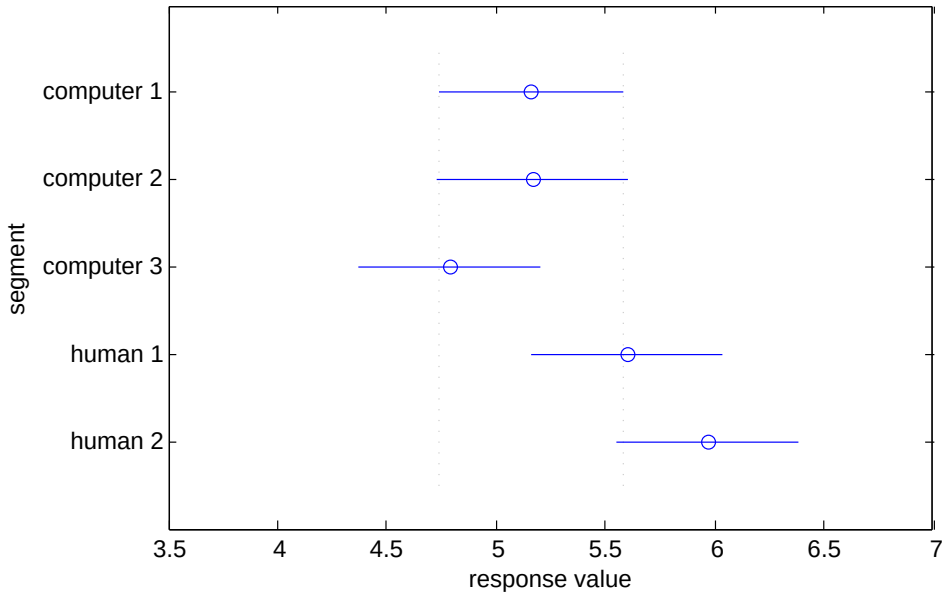

Figure 12: Plot showing mean values and confidence intervals for responses to Question 5: "To what extent would you say that the improvisations in this recording are appropriate to the style and the theme?" Audio excerpts 1-3 are computer-generated.

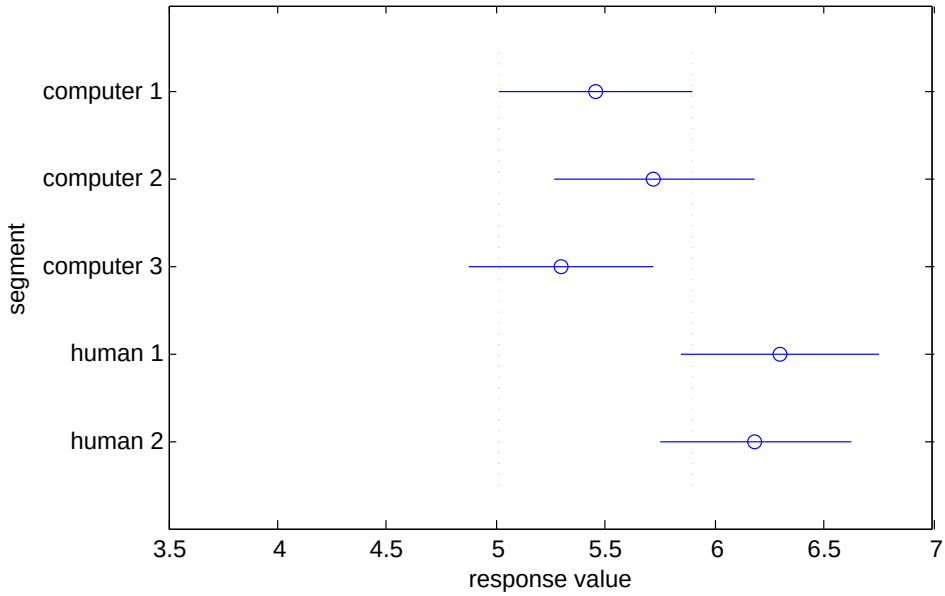

Figure 13: Plot showing mean values and confidence intervals for responses to Question 6: "If told that this recording were of a tabla student, how would you rate his/her overall TECHNICAL abilities?" Audio excerpts 1-3 are computer-generated.

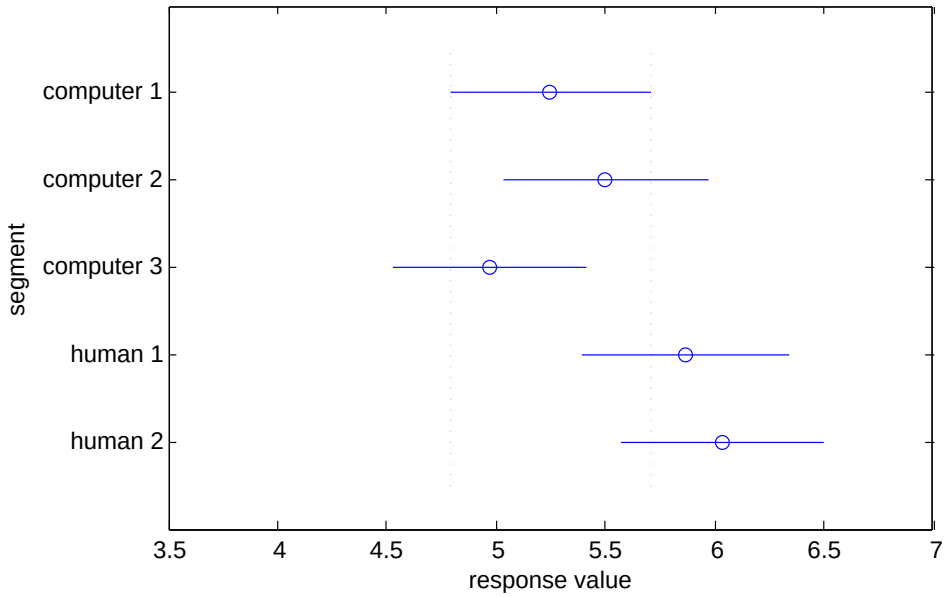

Figure 14: Plot showing mean values and confidence intervals for responses to Question 7: "If told that this recording were of a tabla student, how would you rate his/her overall MUSICAL abilities?" Audio excerpts 1-3 are computer-generated.

right-hand drum, and negatively for the *bayan*, whose pitch would be modulated in truly expressive playing. The same respondent wrote similarly of excerpt 2: "Right drum sounds very musical (has good tonal quality and sounds very crisp). Baya (left drum) playing can be improved a bit in terms of modulation and melodic content." Some comments focused more directly on the style or quality, for example "Good presentation of Purab / Benaras style kayda. Great speed. Nice overall sound" (excerpt 2), and "Very nicely done" (excerpt 3). Only one respondent clearly deduced the origin of the model's output, writing simply "The synthesized nature of this piece limits its ability to be musical."

Criticism was not reserved for the generated recordings. One respondent commented that excerpt 4 "sounded too mechanical and devoid of emotion," and another that "The Tirakitas at the start [of example 5] sound very odd and clumsy!" Most comments for examples 4 and 5, however, were clearly positive.

The results of this survey indicate that the qaida modeling undertaken in this work has been successful in producing improvisatory music which is heard as creative. There is, of course, much work to be done, ranging from addressing deficiencies in playback cited by

a number of respondents, such as the lack of *bayan* modulation, to incorporating a more robust model of sculpting a larger contour. However it is encouraging and quite interesting to see how effective the methods employed in this model have been.

CHAPTER III

LAYER BASED MODELING

In this chapter we describe an experimental attempt to apply the general approach used in modeling qaida to a substantially different musical form. A wide array of subgenres of electronic music variously and imprecisely referred to as "electronica," "glitch," and "IDM", among other names, employ a compositional paradigm which privileges the use of "vertical" layering of patterns over "horizontal" transformation of those patterns as a technique of development. It has been mentioned several times that tabla music is ostensibly monophonic, and the patterns of variation in qaida development described in 2.2.1 are sequential in nature. Cyclic repetition is essential to qaida form, but variations proceed linearly in time, in the continued rearrangement of theme partitions. The form addressed in this chapter, which we will simply refer to as "layer-based," contrasts sharply with this. The difference we are concerned with here is less one of polyphony vs. monophony, as many other genres contain polyphony, than of a manner of development over time. Specifically, the layer-based form is fundamentally based around the use of multiple layers which mesh and interlock but do not themselves change substantially over time; development is built around addition, subtraction, and recombination of elements.

The motivation for the work presented in this chapter is two-fold. On the one hand, the purpose is to make good on previous claims of the potential generality of certain aspects of our approach. On the other hand, taking on a very different musical form allows us to explore limits and assumptions that may have been present in the first case, but not readily apparent. In particular, the model's approach to variation generation was initially seen as closely tied to qaida's sequential development, in which different variations are presented in succession. This chapter is intended to explore the extent to which a similar procedure can be used to in a form lacking that structure. The notion of partitioning the seed material to obtain the working material for musically creative output was initially conceptualized within

the context of qaida, whose traditional theory includes this concept; however applying that technique in a different musical context illustrates that this component of the model is not specific to qaida.

This project presents a number of non-trivial challenges which make it in many ways more difficult than qaida modeling. Unlike qaida, this style does not belong to a long-lasting classical tradition, and is not the subject of a branch of formal music theory. Such theories as do exist are typically not formulated or even known by practitioners — the roots of this branch of electronic music are well outside academia, notwithstanding a recent increase in communication and cross-fertilization between the two worlds. Structural consistencies can certainly be found, but they are better described as conventions than known forms, and the notion of a canonical form is not applicable. Many pieces are essentially impossible to transcribe, since the sounds themselves play a central role in defining the character of a particular piece — a spectrogram may in some sense be considered a reasonable "transcription", but is hardly readable in the manner of a score.

As before, the system presented here requires a seed, a sonic nucleus from which a set of variants is built. Lacking symbolic data, we must start with audio. An important point to emphasize here is that one function of this system is to create new works based upon existing material, or stated differently to allow the model to draw "inspiration" from some musical work. Ultimately, the domain in which that music is represented should not raise an insurpassable barrier to the model's applicability. For this reason, we see introducing a substantial MIR sub-project as fully consistent with the core motivations of the project as a whole. Bringing in audio analysis in this way is challenging, but also attractive. While almost guaranteeing the introduction of errors and noise, it helps to situate the modeling techniques within a broader field, binding computer generation of music to machine listening; additionally, we find many of the constituent problems to interesting in their own right.

It should be made clear at this point that the notion of generality is fairly specific, and refers to the capability of our modeling technique to be applied to different musical contexts and materials. The actual implementation, of course, may require varying amounts

of customization to the stylistic characteristics of the new genre. With the introduction of the capability to act directly on audio, it is tempting to expect, or hope, that the system could thenceforth operate on any soundfile supplied; while interesting results may certainly be possible, this is in general unlikely to be the case, and we make no claims in that direction.

Fortunately, some of the difficulties of this project conversely make other aspects easier. One daunting difficulty in tabla modeling is that the standard of comparison is not only high, but very human. The nuances of timbre and timing which set apart the great players or performances from the good are, frankly, extremely elusive. Music which is made to be played from a recording, or even if "live", from a computer, may be no less profound, but is not built around a human performer, making an automated system less prone to unflattering comparison. Similarly, stylistic flexibility implies a certain degree of permissiveness in terms of what may be considered musically appropriate.

It should be emphasized that the work described in this chapter is experimental. While similarly designed for realtime operation, this system as of yet has no functional user interface, and is operated from the command line. Less effort has been put in to the fine points of audio generation. Finally, there is no formal evaluation as in Chapter 2, for the dual reasons that comparisons of the type used in the qaida survey are much more difficult to arrange, and that the output does not yet have a "polished" feel that one would expect from a human-composed work. This last point is essential to address in machine-human comparison studies, lest respondents deduce a machine origin and be immediately biased by their pre-conceptions on the subject [16].

Finally, the particular choice of this style as a subject for modeling stems largely from personal interest, and from experience making music that falls broadly within this category. It is exciting to bring a radically different approach to this area.

3.1 Introduction to Layer-based Electronica

Before proceeding, it is critical to emphasize the diversity of musical approach contained in this catch-all name. What this chapter focuses on is a particular form found repeatedly in various related subgenres, and not a monolithic characteristic of all music which has been

labeled as "electronica". This form can be more generally seen as a strategic simplification of a more common form or conceptual basis underlying much of this music. This work, however is not an ethnographic study or a work of music theory, so questions of just how prevalent the precise form we address actually is are left without further discussion.

Here we describe some of the stylistic characteristics of electronica which are most relevant to the form modeled here. Readers familiar with the Minimalist movement may note some qualities shared between the two, notably the use of repetition combined with slow changes over time [44]. The music is fundamentally loop-based, a facet which has led other to draw a comparison with the Minimalists [29] As stated, this music is often based around the addition and subtraction of distinct layers [26]. Roughly speaking, a layer is a set of sound events which are grouped in some perceptually relevant manner. Adding or subtracting a group of otherwise unrelated sounds as a unit may be enough to group them in this sense, but typically, a layer is comprised of a timbrally similar set of sounds [3]. If containing more than one distinct timbre, it is likely that the events will be grouped in some other way, such as by forming a gestural unit or phrase, or through reference to a well known combination (notably the "kick-snare" combination of paired low and high frequency percussive events).

Composition over longer time-scales involves drawing slow contours through the additive introduction of layers, and perhaps through gradual changes in timbral parameters [26]. Perhaps the simplest compositional form is a progression from a single layer to a slow "climax" consisting of a sustained dense texture, a mixture of all layers, followed by a deconstruction through subtraction in which layers are removed to return the piece to its starting state. A simple variation on this form would be to subtract layers in a different order, leading to a substantially different end point.

A key feature underlying the layer-based form is a kind of stasis in the material. Once established, a pattern in a given layer is unlikely to change. If it does so, then there are none of the "melodic" characteristics found in the changing patterns of solo tabla; such changes function primarily as shifts in texture. In reality, many works that roughly follow this pattern employ subtle changes in a number of ways, for example by gradual alteration

of timbres, or by introducing a single alteration to an otherwise precisely repetitive layer, thereby creating a peculiar effect by focusing the listener's attention on a detail that in other contexts might seem unremarkable.

In the current work, we address the simplest case, in which layers are considered to be timbrally related sets of events and are not varied once established, and in which longer time-scale form is developed through addition and subtraction of these layers. One additional and interesting complication which is included, however, will be termed "meta-layers". Admittedly an imprecisely defined concept, meta-layers are layers which are formed by subtracting elements from a full layer. The motivation for making this distinction comes from the occasional practice of introducing rhythmic elements which create an ambiguous sense of timing until other timbrally and rhythmically related elements are introduced. Once they are all present, this larger group will fuse perceptually, lending coherence to its components; when this occurs, it suggests that it is this larger set that would be more robustly identified as a layer, thus motivating the introduction of the term meta-layer to refer to those elements which may enter on their own.

3.2 Methods

The initial step in designing the model was a structural analysis similar to that in Chapter 2. A number of key structural elements were determined, reflecting the musical characteristics described above. As with qaida, the final output can be seen as dependent on the initial choice of seed material (analogous to the qaida theme), and the time position. The set of materials constituting the layers is of course dependent on the seed audio, but there is also a mutual dependency among the layers. This represents the common tendency of layers to "interlock," for example by loosely tiling the timbral space over time, not to the exclusion of gaps or silence, but minimizing consistent overlap which would dull the perceptual independence of the layers. The choice of sounding layers at any given moment is determined by the time position, and the expression a layer as complete vs. as a meta-layer depends both on time position and on the choice of layer.

The first implication of this analysis was that generation of musical material and the

structuring of longer forms can be separated. The second was that none generation takes place once the output has started, and thus must take place in the initial setup. A similar approach was taking in the qaida model, but in that case it was neither a necessary nor strict condition — the phrase bank is built at the outset for reasons of computational efficiency, and the system includes methods for altering this bank while the qaida plays. The layer-based model in its current form precludes generating additional material as the music progresses. Modeling such changes would be a natural direction for future work; this would essentially entail developing a more sophisticated model.

The observations regarding the mutual dependencies of layer content, in addition to their collective dependency on the seed material suggested an approach based in source separation. This seems a natural direction to take, given that we are starting with a single section of audio, and wish to end up with multiple layers, but there are other possible approaches, such as linear segmentation and categorization of the segments. An additional motivation to pursue source separation was that many algorithms attempt to estimate a mixture of maximally *dissimilar* components which best represent the signal. The source separation algorithm chosen, described in Section 3.2.1 has the added benefit that its output is easily interpreted as a time-frequency representation — internally, it operates on probability distributions, which have the same mathematical form. These characteristics of source separation align well with the conception of layers as perceptually and timbrally distinct entities.

We approach the generation of new material corresponding each of these layers in a similar manner adopted in Chapter 2. The construction of a compositional arc is accomplished though their addition and subtraction over time; this component is less deterministic than the playback pattern implemented previously. As in the qaida model, the system was coded in Python, with NumPy and SciPy numerical computing extensions. A block diagram of the system architecture is shown in Figure 15. A pre-cut soundfile containing a short musical excerpt, the seed, is loaded, converted to a time-frequency representation, and components thought to correspond to timbral layers are extracted, represented as timbral and temporal profiles. Audio is resynthesized from these components, forming a basis of

separate layers. Each one of these layers undergoes a beat-synchronous segmentation and shuffling process. Each resulting segment is analyzed and again segmented according to detected onsets. These single-event chunks are then put through a filtering function which suppresses a subset of the events. The end result of this is a hierarchically organized and shuffled set of layers, depicted in Figure 16.

Audio was imported from prepared soundfiles, and stereo channels were mixed prior to analysis. First, a time-frequency representation of the signal was obtained by applying the Short Time Fourier Transform [58], defined in the discrete case as

$$\textbf{STFT}\left\{x[n]\right\} \equiv X(m,\omega) = \sum_{n=-\infty}^{\infty} x[n]w[n-m]e^{-j\omega n} \tag{4}$$

where $x[n]$ reresents the audio signal and time index n, m is the time index into the STFT, *omega* is frequency, and w represents a window function. In the definition in Equation 4, the window function serves to both isolate a segment of the signal, and to impose a shape on that segment. In practice, the STFT is computed by splitting the signal into short overlapping frames, and applying a window function, in this case either Hann or Hamming window, and taking the fast Fourier transform (FFT) of the frame. The Hamming window is defined as

$$w(n) = 0.53836 - 0.46164\ \cos\left(\frac{2\pi n}{N-1}\right) \tag{5}$$

where the window is of length N, and the Hann window as

$$w(n) = 0.5\ \left(1 - \cos\left(\frac{2\pi n}{N-1}\right)\right). \tag{6}$$

Frame sizes of $N = 1024$ samples were used, with 50% overlap. The resulting complex valued matrix was then used as input to a source separation algorithm using Probabilistic Latent Component Analysis [57], which yielded pairs of spectral and temporal contributions for each extracted component.

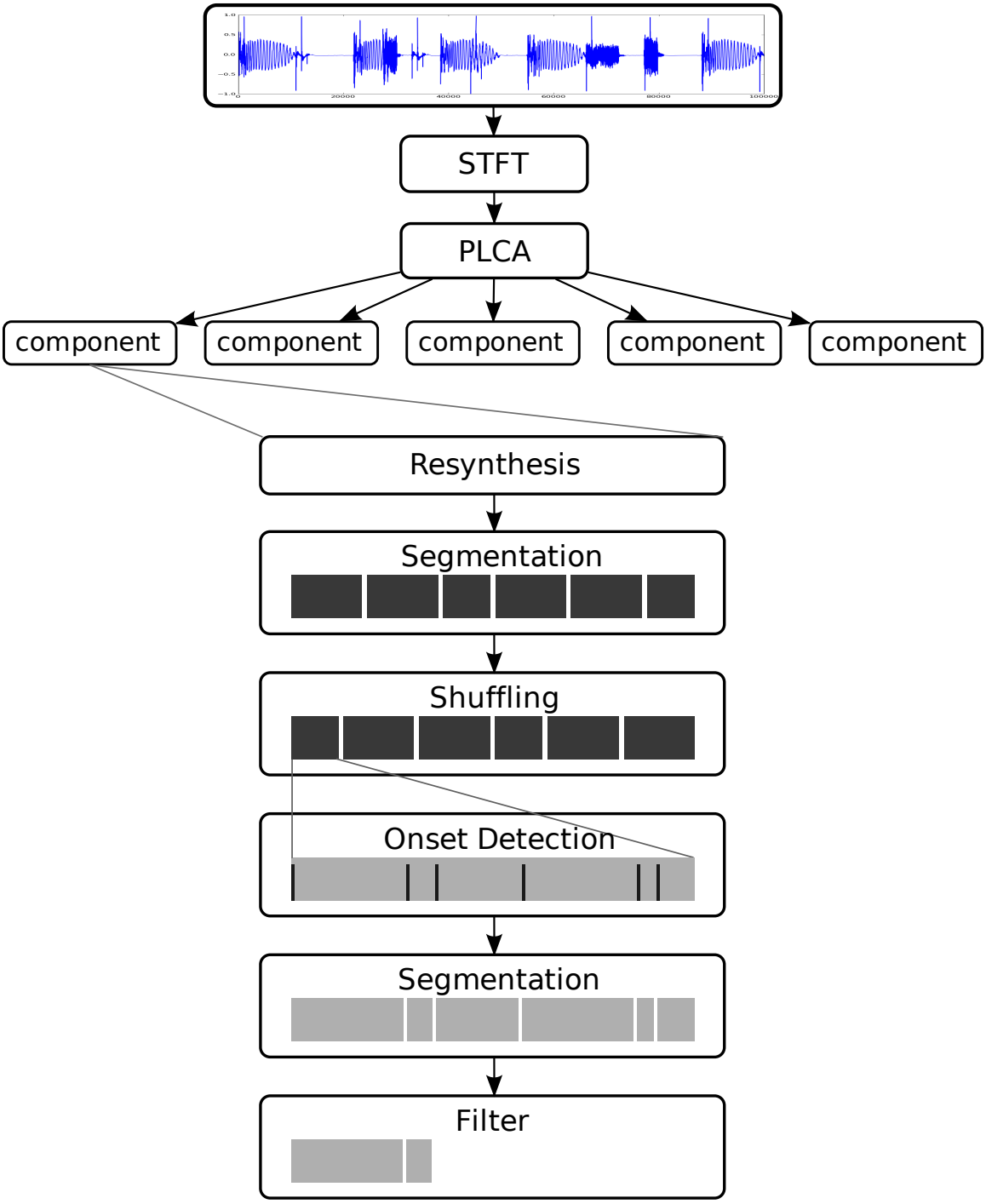

Figure 15: Block diagram of the layer generation system.

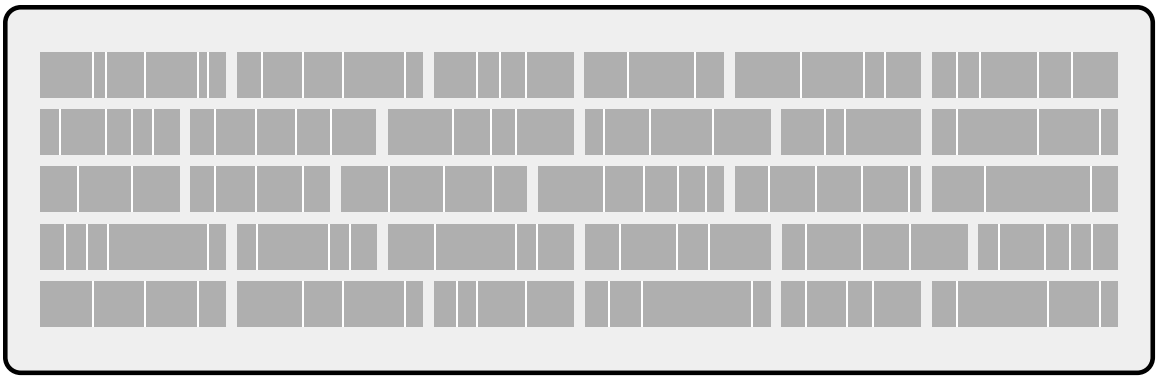

Figure 16: Depiction of the set of processed layers generated from one clip of source material. These layers form the material from which a new piece is constructed. Note that the filtering in the last step is dynamic; all of the material is shown here while only portions will be synthesized.

3.2.1 Source Separation

Blind source separation is the task of analyzing a signal in order to separately recover its components, without any specific knowledge of the components themselves. In audio, this corresponds to "unmixing", in which one seeks to reconstruct the clean signal of each of a number of sounds that have been mixed together into one or more channels of audio. Source separation is a challenging problem, and is an active area of research. A number of techniques have been proposed. One involves using a multichannel audio feed to isolate sounds originating from different spatial locations [72]. Another approach geared towards musical analysis assumes a mixture of pitched instruments, and uses multi-pitch analysis, or the assumption of harmonic partial structures, to extract components [28, 66]. Independent Component Analysis represents the audio in terms of statistically independent components, and does not require specific assumptions (multichannel recording, or mixtures of harmonic spectra) about the material on which it works [39].

We approach the separation of extracting independent layers using a technique developed by Paris Smaragdis, and described in detail in [56]. Known as Probabilistic Latent Component Analysis (PLCA), this technique uses the iterative Expectation Maximization (E-M) algorithm to estimate the timbral profile and relative contribution over time of a set of components which best describe the signal. The input to the system is a spectrogram,

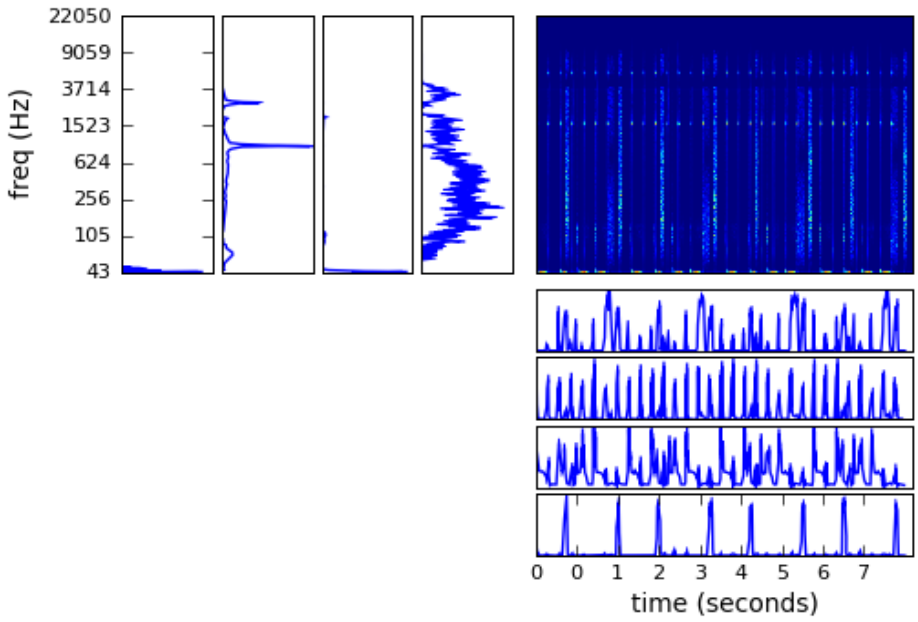

Figure 17: A set of components extracted by PLCA. The four graphs in the upper left describe the magnitude spectra of the components, and the bottom right represents their respective contribution over time. The top right shows a spectrogram of the original mixed sound.

and the output consists of a set of paired magnitude spectra and temporal signals. The system is in fact agnostic to the nature of the input; this version of the algorithm will operate on any 2-dimensional input. The number of desired components is unfortunately required as a parameter for the algorithm, and this remains a limitation on the autonomy of the system. An example output, extracting four components from *Clipper*, a track by the well-known English musical group Autechre, is shown in Figure 17.

The code, ported from Matlab, is implemented in Python, with a C++ external generated using the SWIG wrapper generator.

3.2.2 Partitioning

The components obtained via PLCA were resynthesized with the inverse STFT (iSTFT), using and overlap-add procedure corresponding to the original overlap of the audio frames. One difficulty at this stage is that the PLCA returns a magnitude spectrum, that is, a real-valued sequence, rather than the complex array that is needed to reliably resynthesize

a time-domain signal. This is in general a problem in iSTFT-based resynthesis, as the signal is underdetermined given only the magnitudes [2]. Again following [57], the phase information is simply taken from the original complex STFT of the unseparated signal. This approach generates audio which is, on informal listening tests, reasonably clear of artifacts.

Following this, the set of audio tracks are partitioned into beat-synchronous audio segments. There are many possible divisions, and the motivation for this step, and the choice of partitioning scheme, should be explained. Within the general family of musical styles to which glitch and IDM belong, many genres make extensive use of drum samples known as breakbeats. Breakbeats are sections of recordings, often from older funk or gospel records, in which the all other intrumentation drops out, leaving the drums isolated. With the introduction of relatively cheap samplers in the 1980's and 90's, hip-hop producers began using breakbeat loops as the foundation for their music [38]. Around the same time, producers of dance-driven styles such as techno and house found breakbeats to be fertile source material, and quickly a whole genre known as "jungle" sprang up around the idea of building frenetic and syncopated dance music built from sped-up breakbeats. The two most obvious ways of using this material in a rhythmic style would be to simply loop the entire audio file, or to cut it up into isolated hits and arrange those in a MIDI sequencer, and both techniques have been used extensively across different genres. However, the method that became standard among jungle producers was to cut the breakbeat into assymetric sections, most commonly in a 3 – 3 – 2 pattern of beats [12, 10]. The resulting audio files would typically contain short phrases of several drum hits, and could be rearranged and repeated to form new sequences. This technique gave jungle and the multitude of other subgenres that evolved out of it much of their distinctive character, encouraging heavy syncopation and repetition of short rhythmic sequences, and integrating the quantized timing of a MIDI sequencer with the "human" feel present within the segments.

Beat-tracking was attempted, using a Python implementation of Davies' context-dependent algorithm [21], however it was found that the precise requirements of this type of segmentation meant that the system had a very low tolerance for error in this step — small errors in tempo or phrase estimation could drastically lower the quality of the output. "Off-beat"

phase errors, in which the beat locations are estimated at one half measure from their correct locations, are probably the most common type of error in beat tracking, but it is the smaller errors, which might split a beat-synchronous event between segments, that are more problematic. This is by no means an unsolvable problem; it forms a substantial portion of Nick Collins' PhD thesis [12], but a decision was taken to sidestep this issue by clipping the initial input soundfiles to an integer multiple of their measure length, allowing segmentation to proceed based on ratios of the total duration.

The precise location of the edits is determined by a routine loosely based on Jehan's auto-segmentation [35]. The most recent local minimum in the amplitude envelope is found, and a zero crossing selected. If no sample index fitting those conditions can be found within a window of 50 ms, an amplitude ramp of the same length is applied.

These segments are then shuffled for each track. This step both mimics the production style of breakbeat-based music, and is notably similar to the partition-shuffling described in 2.4.2 that forms a core routine in the qaida model.

3.2.3 Onset Detection

Onset detection was performed using a spectral flux algorithm. This approach involves looking for regions in which the overall change in energy across frequency bins is maximal. A detection function is built from the spectral flux measure, processed to clarify the desired peaks, and onsets are selected using an adaptive thresholding scheme.

The basic input to the onset detection algorithm is a time-frequency representation of the signal. This was obtained by calculating the short-time Fourier transform (STFT) of the signal, defined in Equation 4. After the FFT is computed for each frame, is the magnitude spectrum is computed, and the resulting spectrogram is processed using an adaptive spectral whitening method proposed in [64]. This method seeks to improve detection accuracy by normalizing the magnitude of each frequency bin across time, calculating the normalization factor from a decaying window to provide a moderately local estimate of the range of values for a given bin. This is somewhat similar to multiband dynamic compression, in that amplitudes are scaled according to recent maxima, whose influence decrease over time. An

option is provided to use a weighted average of the original and whitened spectrograms.

Spectral flux is essentially a measure of the overall change between successive frames of spectra. There are a number of possible definitions, two of the most straight-forward being simple Euclidian distance:

$$\begin{aligned} SF(n) &= |X_k(n) - X_k(n-1)|_2 \\ &= \left(\sum_k |X(n,k) - X(n-1,k)|^2 \right)^{1/2} \end{aligned} \tag{7}$$

where $X_k(n)$ represents the value of the spectrogram at the kth frequency bin and nth frame, and positive difference:

$$SF^+(n) = \left| H^+[X_k(n) - X_k(n-1)] \right| \tag{8}$$

where $H^+ = (x+|x|)/2$ is a half-wave rectifier, producing a function which only measures changes in the positive direction[23]. The form used here, however, uses a slightly different, though related, distance metric. The L^p norm is, briefly, a generalization of a certain class of distance calculations, defined as

$$|x|_p \equiv \left(\sum_i |x_i|^p \right)^{1/p} \tag{9}$$

where the value of p defines the norm. Euclidian distance, as in 7 is L^2, while L^1 is a simple summation, commonly referred to as the Manhattan distance, and is the basis for 8 (before half-wave rectification). Following Sapp [55], we set $p = .25$, giving the resulting spectral flux measure:

$$\begin{aligned} SF_p(n) &= |X_k(n) - X_k(n-1)|_p \\ &= \left(\sum_k |X(n,k) - X(n-1,k)|^p \right)^{1/p} \end{aligned} \tag{10}$$

This was informally found to give slightly better results than L^1 or L^2.

The spectral flux detection function is then smoothed by convolving with a short asymmetric kernel comprised of two half-Hanning windows. The widths of each kernel are specified by a single parameter; the "left" half is typically specified to be around 20 ms in length, and the "right" around 150–200 ms. The motivation for this kernel was to preserve the sharpness of attacks, while crudely modeling temporal masking known to occur within 200 ms after a perceptually salient onset, and around 20 ms before [35]. At the same time, this process helps to eliminate spurious peaks that are likely to occurs in noisy regions associated with onsets. This is particularly relevant in the case of detecting onsets in audio resynthesized from PLCA components, as timbral distortion and other artifacts may occur. An exponential decay is then applied to the detection function, and the result is stored separately.

Using a scheme similar to Dixon [23], onsets are identified as those frame indices which satisfy three conditions: the frame must be contain a local maximum in the detection function; the detection function in that frame must not be less than the exponentially decayed copy; the detection function must be greater than some threshold above the local median, calculated on a sliding window. Additionally, onsets are constrained by a debounce time, a short temporal window of 50 ms following a detected onset, within which any subsequent matches to the above three criteria are rejected.

3.2.4 Resegmentation

At this point the audio is again segmented, this time using the detected onsets. An attempt was made to explicitly look for "offsets" in the audio, loosely defined as a point after the onset at which the signal returns to a similar state found prior to the onset. However, better results were obtained by simply cutting the audio into blocks.

3.2.5 Final Representation and Playback

Now in possession of an array of data corresponding to a flexibly playable sequence, that is, a list of start times and associated audio segments, we have one instance of the basic components for a piece. Metalayers are implemented by applying a filter to the symbolic sequence, passing through only a subset of the events. This is applied independently to each

partition in each layer, in order to preserve the rhythmic structure produced by shuffling the partitions.

This set of layers and subcomponents becomes the basic material for a full piece. A contour can be created by following the basic additive procedure of choosing a layer to introduce at every reiteration of the underlying cycle, and similarly by possibly changing the sequence filtering parameter of one or more layers at every loop boundary. The order of introduction is important, of course, and an area of future work would be to apply a more sophisticated algorithm to this procedure; methods could include sorting by spectral centroid or other timbral measure to, for example, follow a global timbral arc, or conversely to ensure a certain degree of relative timbral separation between the currently playing set of layers. It was found that manually choosing the starting layer to be rhythmically consistent, occupying the role of a "click-track" or hi-hat, greatly improved the perceptual coherence of the output. This is of course to be expected, but it should be noted that this device is frequently employed in the style of music we are concerned with here. More sophistication could be achieved by analyzing layers for rhythmic stability, and perhaps delaying the introduction of rhythmically steady elements, allowing for an initial period of uncertainty while resolving it early enough for the rhythmic character to cohere.

Overly direct analogy with the structure of the qaida back-end is not appropriate, due to a number of previously described differences in stylistic form, but it can be seen that this grouping of layers has parallels to both the qaida theme, in that once chosen, the underlying material remains static for the remainder of the piece, and to the bank of variations, in that a compositional form can be constructed through choosing elements from it to enter into the musical output. However, it is also possible to generate a large bank of the layer sets through multiple applications of the shuffling and partitioning procedures. A single group would still be chosen at the outset. Initial experiments in this direction are being conducted, and it is expected that this will greatly improve the system's performance, since it will become possible to implement the "push/select" paradigm used in the qaida generator in order to increase the musicality and coherence of the sonic output.

3.3 Discussion

The work undertaken in this section of the project was successful in a number of ways, but admittedly less so than that presented in Chapter 2. For the most part, the discrepancy lies in the raft of technical challenges involved in working directly with the audio, in particular the process of source separation; further the timbral qualities of the separated layers resynthesized from components estimated using PLCA often include artifacts from the FFT resynthesis, and while the separation is often surprisingly effective, it can require a substantial degree of manual tuning of the parameters. That aspect of the system is not "intelligent"; it is agnostic to the domain of the material it analyzes, and thus is not able to incorporate any understanding of the musical structures or other salient qualities that assist humans in picking apart a mixed signal. Several elements of the generation procedure also have a fairly low tolerance for error, notably in the segmentation by onset; small errors at this step can greatly disturb the quality of the final output. Much more work could also be done to fine-tune the more successful aspects of the system, and introducing an audio processing module at the output would be one obvious improvement.

However, the majority of runs of the algorithm do produce music which is, as hoped, both noticeably related to the seed track, and fairly novel. The simple additive form creates a clearly audible structure, and once that pattern becomes audibly clear, after, say, the second addition, it tunes the listeners expectation in a way that can be quite effective. The basic approach to developing this system is highly similar to that used in the qaida generator, notably in the decision to operate on larger abstracted rhythmic units, and employing operations exploiting rhythmic context switching to generate related new material. The idea of operating in parallel on different levels of grouping and abstraction is common to both as well, for example in separating out the handling of full cycles/loops of material, manipulating partitions, and descending to the event level. Further, both systems incorporate the notion of injecting domain-specific knowledge into the choice of operations at limited specific points in the procedure.

CHAPTER IV

CONCLUSION

Two implementations of a fairly generalizable approach to generative rhythmic modeling have been presented. The systems, designed to model the temporally sequential thema-and-variations tabla solo form qaida and a layer-based sub-genre of electronica, generate musical output which is both appropriate and novel within their respective stylistic domains. They are intended in part to be explorations in computational modeling of musical creativity, and also to comprise core components of future performance systems. The results are generally successful, though more so in the case of qaida modeling. Evaluation of the generative qaida model was performed through a blind online survey, and preliminary results suggest that listeners do perceive the system's output as creative and fitting, with some predictable qualifications.

The work described here in many ways represents an initial step, and there are many promising areas for future work, most of which have been detailed. Generally, this work involves a wide array of disciplines, and so future work will be similarly wide-ranging. Notable areas include modeling expressive *bayan* modulation, building discriminative models based on a large corpus of existing music to incorporate a more robust fittingness measure, and more sohpistication in sound production and reproduction.

REFERENCES

[1] Bishop, C. M., *Pattern Recognition and Machine Learning*. New York: Springer, 2006.

[2] Bouvrie, J. and Ezzat, T., "An incremental algorithm for signal reconstruction from short-time Fourier transform magnitude," in *Proc. Ninth International Conference on Spoken Language Processing (Interspeech)*, (Pittsburgh), 2006.

[3] Butler, M. J., *Unlocking the groove*. Indiana University Press, 2006.

[4] Cemgil, A. T., "Bayesian methods for music signal analysis," October 2006. Slides for tutorial given at ISMIR 2006, Victoria, Canada.

[5] Cemgil, A., Barber, D., and Kappen, H. J., "A Dynamical Bayesian Network for Tempo and Polyphonic Pitch tracking," in *Proceedings of ICANN*, (Istanbul, Turkey), 2003.

[6] Chadabe, J., *Electric Sound*. Prentice Hall, 1997.

[7] Chordia, P., *Automatic Transcription of Solo Tabla Music*. PhD thesis, Stanford University, Dec. 2005.

[8] Chordia, P. and Rae, A., "Tabla Gyan: A realtime tabla recognition system," in *Proceedings of International Computer Music Conference*, 2008.

[9] Chordia, P. and Rae, A., "Slow Theka." `http://paragchordia.com/music/slowTheka`, 2009.

[10] Collins, N., "Algorithmic composition methods for breakbeat science," in *Proceedings of Music Without Walls*, (De Montfort University, Leicester), 2001.

[11] Collins, N., "BBCut2: Incorporating beat tracking and on-the-fly event analysis," *Journal of New Music Research*, vol. 35, no. 1, pp. 63–70, 2006.

[12] Collins, N., *Towards Autonomous Agents for Live Computer Music: Realtime Machine Listening and Interactive Music Systems*. PhD thesis, University of Cambridge, 2006.

[13] Collins, N., "BBCut2." `http://www.informatics.sussex.ac.uk/users/nc81/bbcut2.php`, (accessed March 2009).

[14] Cope, D., *Computers and Musical Style*. Madison, WI: A-R Editions, 1991.

[15] Cope, D., *Experiments in Musical Intelligence*. Madison, WI: A-R Editions, 1996.

[16] Cope, D., "Facing the music: Perspectives on machine-composed music," *Leonardo Music Journal*, vol. 09, pp. 79–87, Mar. 1999.

[17] Cope, D., *Virtual Music: Computer Synthesis of Musical Style*. Cambridge MA: MIT Press, 2001.

[18] Cope, D., *Computer Models of Musical Creativity*. Cambridge MA: MIT Press, 2005.

[19] Csikszentmihalyi, M., *Creativity: Flow and the Psychology of Discovery and Invention*. New York: Harper Collins, 1996.

[20] Davies, M. E. P., *Towards Automatic Rhythmic Accompaniment*. PhD thesis, Queen Mary University of London, Department of Electronic Engineering, 2007.

[21] Davies, M. E. P. and Plumbley, M. D., "Context-dependent beat tracking of musical audio," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 15, no. 3, pp. 1009–1020, 2007.

[22] Deutsch, D., *The Psychology of Music*. Academic Press, 1999.

[23] Dixon, S., "Onset detection revisited," in *Proceedings of the 9th International Conference on Digital Audio Effects (DAFx'06)*, (Montreal, Canada), pp. 134–137, September 2006.

[24] Eigenfeldt, A., "The creation of evolutionary rhythms within a multi-agent networked drum ensemble," in *Proceedings of the International Computer Music Conference*, (Copenhagen, Denmark), pp. 267–270, 2007.

[25] Eigenfeldt, A., "Drum circle: Intelligent agents in Max/MSP," in *Proceedings of the International Computer Music Conference*, (Copenhagen, Denmark), pp. 9–12, 2007.

[26] Eigenfeldt, A., "Encoding knowledge in software performance tools," in *SEAMUS 07*, 2007.

[27] Eigenfeldt, A., "Automated electronica." `http://www.sfu.ca/~eigenfel/research.html#electronica`, (accessed March 2009).

[28] Every, M. and Szymanski, J., "A spectral-filtering approach to music signal separation," in *Proceedings of the 7th International Conference on Digital Audio Effects (DAFx 04)*, (Naples, Italy), pp. 197–200, Oct. 2004.

[29] Fink, R., *Repeating Ourselves : American Minimal Music as Cultural Practice*. Berkeley, CA: University of California Press, 2005.

[30] Fink, R. A., Ward, T. B., and Smith, S. M., *Creative Cognition: Theory, Research, and Applications*. Cambridge, MA: MIT Press, 1992.

[31] Freeman, J., "Graph Theory: Linking online musical exploration to concert hall performance," *Leonardo*, vol. 41, pp. 91–93, Jan. 2008.

[32] Gardner, H., *Intelligence reframed: multiple intelligences for the 21st century*. New York: Basic Books, 1999.

[33] Hunt, A., Wanderley, M., and Paradis, M., "The importance of parameter mapping in electronic instrument design," in *Proceedings of the 2002 Conference on New Instruments for Musical Expression (NIME-02)*, (Dublin, Ireland), May 2002.

[34] Huron, D., *Sweet Anticipation: Music and the Psychology of Expectation*. MIT Press, 2006.

[35] Jehan, T., *Creating Music by Listening*. Phd thesis in media arts and sciences, Massachusetts Institute of Technology, 2005.

[36] Jones, E., Oliphant, T., Peterson, P., and others, "SciPy: Open source scientific tools for Python," 2001–present.

[37] Jordan, M. I., ed., *Learning in Graphical Models*. MIT Press, 1999.

[38] Katz, M., *Capturing Sound: How Technology has Changed Music*. Los Angeles: University of California Press, 2004.

[39] Lee, T.-W., Bell, A. J., and Orglmeister, R., "Blind source separation of real world signals," in *Proceedings of the IEEE International Conference on Neural Networks*, no. 4, (Houston, Texas), pp. 2129–2134, June 1997.

[40] Lewis, G., "Too many notes: Computers, complexity and culture in voyager," *Leonardo Music Journal*, vol. 10, 2000.

[41] Lucier, A., *Statement On: Music for Solo Performer*. Biofeedback and the Arts: Results of Early Experiments, Vancouver, Canada: Aesthetic Research Centre of Canada, 1976.

[42] Machover, T., *Hyperinstruments — A Composer's Approach to the Evolution of Intelligent Musical Instruments*, pp. 67–76. Cyberarts: Exploring Arts and Technology, San Francisco: MillerFreeman, Inc., 1992.

[43] McCartney, J., "Rethinking the computer music language: Supercollider," *Computer Music Journal*, vol. 26, no. 4, pp. 61–8, 2002.

[44] Mertens, W., *American Minimal Music*, pp. 307–312. Audio Culture: Readings in Modern Music, London: Continuum International Publishing Group, 2007.

[45] Murphy, K. P., "Introduction an introduction to graphical models." `http://www.cs.ubc.ca/~murphyk/Papers/intro_gm.pdf`, 2001.

[46] Pachet, F., "The Continuator: Musical interaction with style," *Journal of New Music Research*, vol. 32, no. 3, pp. 333–41, 2003.

[47] "Pure Data — PD community site." `http://puredata.info`, (accessed March 2009).

[48] Peeters, G., "A large set of audio features for sound description (similarity and classification) in the cuidado project." CUIDADO Project Report, 2004.

[49] Pereira, F. C., *Creativity and artificial intelligence: a conceptual blending approach*. Walter de Gruyter, 2007.

[50] Puckette, M. S., "Pure data," in *Proceedings of the International Computer Music Conference*, (Ann Arbor, Michigan), pp. 224–227, International Computer Music Association, 1997.

[51] Puckette, M. S., "Pure data." http://crca.ucsd.edu/~msp/software.html, (accessed March 2009).

[52] Raphael, C., "Aligning music audio with symbolic scores using a hybrid graphical model," *Machine Learning*, vol. 65, no. 2-3, pp. 389–409, 2006.

[53] Roads, C., *The Computer Music Tutorial*. Cambridge, MA: MIT Press, 1998.

[54] Ron, D., Singer, Y., and Tishby, N., "The power of amnesia: learning probabilistic automata with variable memory length," *Machine Learning*, vol. 25, no. 2-3, pp. 117–149, 1996.

[55] Sapp, C., "Mzspectralflux." http://www.mazurka.org.uk/software/sv/plugin/MzSpectralFlux/, (accessed March 2009).

[56] Smaragdis, P. and Raj, B., "Shift-invariant probabilistic latent component analysis," tech. rep., 2007.

[57] Smaragdis, P., Raj, B., and Shashanka, M., "Supervised and semi-supervised separation of sounds from single-channel mixtures," in *Proceedings of the 7th International Conference on Independent Component Analysis and Signal Separation*, (London, UK), Sept. 07.

[58] Smith, J. O., *Spectral Audio Signal Processing, October 2008 Draft*. url-http://ccrma.stanford.edu/ jos/sasp/, accessed March 2009. online book.

[59] Solis, J., Chida, K., Taniguchi, K., Hashimoto, S. M., Suefuji, K., and Takanishi, A., "The waseda flutist robot wf-4rii in comparison with a professional flutist," *Computer Music Journal*, vol. 30, pp. 12–27, Dec. 2006.

[60] "Spark festival of electronic music and arts 2009." http://spark.cla.umn.edu/, Feb. 2009.

[61] Sternberg, R. J., "Creativity or creativities?," *International Journal of Human-Computer Studies*, vol. 63, pp. 370–382, Oct. 2005.

[62] Sternberg, R. J. and Davidson, J. E., "The mind of the puzzler," *Psychology Today*, vol. 16, pp. 37–44, Oct. 1982.

[63] Stewart, R. M., *The Tabla in Perspective*. PhD thesis, U. C. Berkeley, 1974.

[64] Stowell, D. and Plumbley, M. D., "Adaptive whitening for improved real-time audio onset detection," in *Proceedings of the International Computer Music Conference (ICMC'07)*, (Copenhagen, Denmark), pp. 312–319, August 2007.

[65] van Rossum, G. and (eds), F. D., *Python Reference Manual*. PythonLabs, Virginia, USA, 2001. Available at http://www.python.org.

[66] Virtanen, T. and Klapuri, A., "Separation of harmonic sounds using multipitch analysis and iterative parameter estimation," in *Proceedings of IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, (New Paltz, New York), pp. 83–86, Oct. 2001.

[67] Waisvisz, M., "The Hands, a set of remote midi-controllers," in *Proceedings of International Computer Music Conference (ICMC)*, (Burnabey BC, Canada), 1985.

[68] Wegner, G.-M., *Vintage Tabla Repertory*. New Delhi, India: Munshiram Manohalal Publishers Pvt. Ltd., 2004.

[69] Weinberg, G., Godfrey, M., Rae, A., and Rhoads, J., "A real-time genetic algorithm in human-robot musical improvisation," pp. 351–359, 2008.

[70] Weinberg, G. and Driscoll, S., "Toward robotic musicianship," *Computer Music Journal*, vol. 30, pp. 28–45, Dec. 2006.

[71] Winkler, T., "Making motion musical," in *Proceedings ofthe 1995 International Computer Music Conference*, 1995.

[72] Woodruff, J. and Pardo, B., "Using pitch, amplitude modulation and spatial cues for separation of harmonic instruments from stereo music recordings," *EURASIP Journal on Advances in Signal Processing*, vol. 1, 2007.

[73] Wright, M., "Open sound control: an enabling technology for musical networking," *Organised Sound*, vol. 10, no. 3, pp. 193–200, 2005.