TablaNet:
A Real-Time Online Musical Collaboration System for Indian Percussion

by

Mihir Sarkar

Diplôme d'Ingénieur ESIEA
Ecole Supérieure d'Informatique Electronique Automatique (1996)

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning,
in partial fulfillment of the requirements for the degree of

Master of Science in Media Technology

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2007

Author .....
Program in Media Arts and Sciences
August 10, 2007

Certified by .....
Barry L. Vercoe
Professor of Media Arts and Sciences
Thesis Supervisor

Accepted by .....
Prof. Deb Roy
Chairperson
Departmental Committee on Graduate Students

TablaNet:
A Real-Time Online Musical Collaboration System for Indian Percussion

by
Mihir Sarkar

Submitted to the Program in Media Arts and Sciences
on August 10, 2007, in partial fulfillment of the
requirements for the degree of
Master of Science in Media Technology

Abstract

Thanks to the Internet, musicians located in different countries can now aspire to play with each other almost as if they were in the same room. However, the time delays due to the inherent latency in computer networks (up to several hundreds of milliseconds over long distances) are unsuitable for musical applications. Some musical collaboration systems address this issue by transmitting compressed audio streams (such as MP3) over low-latency and high-bandwidth networks (e.g. LANs or Internet2) to constrain time delays and optimize musician synchronization. Other systems, on the contrary, increase time delays to a musically-relevant value like one phrase, or one chord progression cycle, and then play it in a loop, thereby constraining the music being performed. In this thesis I propose *TablaNet*, a real-time online musical collaboration system for the tabla, a pair of North Indian hand drums. This system is based on a novel approach that combines machine listening and machine learning. Trained for a particular instrument, here the tabla, the system *recognizes* individual drum strokes played by the musician and sends them as symbols over the network. A computer at the receiving end identifies the musical structure from the incoming sequence of symbols by mapping them dynamically to known musical constructs. To deal with transmission delays, the receiver *predicts* the next events by analyzing previous patterns before receiving the original events, and synthesizes an audio output estimate with the appropriate timing. Although prediction approximations may result in a slightly different musical experience at both ends, we find that this system demonstrates a fair level of playability by tabla players of various levels, and functions well as an educational tool.

Thesis Supervisor: Barry L. Vercoe
Title: Professor of Media Arts and Sciences

TablaNet:
A Real-Time Online Musical Collaboration System for Indian Percussion

by

Mihir Sarkar

Thesis Committee

Advisor .....

Barry L. Vercoe
Professor of Media Arts and Sciences
Massachusetts Institute of Technology

Reader .....

Tod Machover
Professor of Music and Media
Massachusetts Institute of Technology

Reader .....

Miller S. Puckette
Professor, Music
Associate Director, Center for Research in Computing and the Arts
University of California, San Diego

Acknowledgments

For their help, directly or indirectly, with this thesis, I would like to thank:

First and foremost, Sharmila, my wife, with whom I have shared the joys and pains of going back to study, and who supported me patiently through the several nights I spent solving problem sets, debugging code, and writing this thesis. She helps me reach for my dreams.

Barry Vercoe, my advisor, who gave me the extraordinary opportunity to be here, the freedom to explore and learn and grow, and who points me in the right direction when I ask him to. For this, and more, I am deeply indebted to him.

Tod Machover and Miller Puckette, my thesis readers, for their encouragements, patience, and insightful comments.

Owen Meyers (my office-mate), Anna Huang, Wu-Hsi Li, Judy Brown, Dale Joachim, and Yang Yang (my UROPer) from the Music, Mind and Machine group for their feedback, enthusiasm, and ideas... and the opportunity to share mango lassi.

Brian Whitman, a Music, Mind and Machine alumnus, for easing my transition into the Media Lab and being the link between the group's current and former members, for always being available for pertinent discussions, and for his kindness.

Mutsumi Sullivan and Sandy Sener for their administrative support and help with miscellaneous items (from repairing the tabla set to arranging for the compensation of the user study subjects). In particular, I would like to extend very special thanks to Mutsumi, our Music, Mind and Machine group assistant, for taking care of the practical aspects of this thesis—signature gathering, printing, and submission—up until the last minute, while I was busy wrapping up my thesis in India.

The MIT and Media Lab community for truly being out of this world and continuously striving to make the world a better place.

And finally my parents, whose ideals and education have brought me thus far, who have guided me without ever erecting a barrier, for their love and support.

To those above, and those whom I did not mention here but who played a role in this work, I express all my gratitude for their contribution to this thesis.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview

Motivation

In 1992, I organized a concert in India. I recruited two of my musician friends, both keyboard players like me, and we formed a band. We practiced over the summer and performed at the end of August in a beautiful open air courtyard in front of a jam-packed audience. We played instrumental covers of Western pop songs (The Beatles, Stevie Wonder, Phil Collins) some of which were known to the audience in their Indianized version as Bollywood film songs! As I was living in France at that time, I had brought with me sound and light equipment (smoke machine, black light, strobes and scanners). That year, liberalization and cable operators had just introduced MTV to Indian households, but few had yet been exposed to the kinds of synthetic sounds and light effects that we had in our concert.

As I returned to France, my friends and I were eager to carry on with our collaboration. We even included another friend (a guitarist living in the US) in the process. We exchanged multitrack audio cassettes (like many famous bands are said to have done), conversed over the telephone, sent letters and parcels back and forth. But our subsequent interactions never came close to our experience that summer. A few years later, still eager, with our brand new e-mail accounts, we tried using the Internet to exchange MIDI files. Digital audio streaming was still out of our reach then. However with time, and probably because we never quite managed to collaborate with each other like we had during our face-to-face encounters, our enthusiasm faded out (but our friendship remained).

For many years, the problem was in the back of my mind. I had heard of companies trying to address the issue (under the name *Networked Music Performance*) but was never impressed with their results or approach. When I came to the Media Lab, I decided to take up the challenge and find a novel solution to solve this problem.

Description

The main challenge I am trying to address in this thesis is how to overcome network latency for online musical collaboration. If musicians are to play together in real-time over a computer network such as the Internet, they need to remain perceptually synchronized with one another while data travels from one computer to another. I attempt to tackle this problem by developing a system that is tuned to a particular musical instrument and musical style, here the tabla, a traditional North Indian percussion instrument. The system includes hardware and software components, and relies on standard network infrastructure. Software processes the acoustic signal coming out of the tabla by:

1. recognizing individual drum strokes,

2. transmitting symbolic events (instead of an audio stream) over the network,

3. extracting higher-level rhythmic features and identifying standard drumming primitives from the input data,

4. analyzing previous patterns to predict current events, and

5. synthesizing and playing rhythmic phrases with the appropriate tempo at the audio output.

Contributions

The research presented in this thesis has resulted in the following contributions:

- I implemented a novel approach for real-time online musical collaboration,

- enabled a real-world musical interaction between two tabla musicians over a computer network,

- designed a networked tabla performance system,

- created a tabla phrase prediction engine, and

- developed a real-time continuous tabla strokes recognizer.

This work resulted in a unidirectional playable prototype (from a musician to a far-end listener)—although the system is symmetrical, it has not been tested in full-duplex—and a software simulation environment for testing and demonstration. Preliminary evaluation results show that the system is suitable for distance education and distributed jamming.

1.2 Scope

Context

Several systems have been developed in the past to allow musicians located in different places to play together in real-time or, more generally, to enable various forms of online musical collaboration (refer to Section 2.1). However, notwithstanding specific experimental concerts (usually during musical technology conferences with like-minded computer music buffs), wide-spread success has remained elusive, and many commercial endeavors have died down (although new ones spring up every now and then).

In spite of technological advances in digital audio and networking—the two enabling technologies in this case—most existing frameworks have hit a hard limit due to the inherent latency in computer networks because of protocol and system overhead (buffering, packet switching, etc.).

Much of the literature on networked music performance mentions the speed of light as its main obstacle. Let us perform some basic calculations to verify that claim. The circumference of the earth being around 40,000 km (there are a few kilometers difference between the measurement at the poles and at the equator), the maximum distance between two points on earth is 20,000 km. Taking the speed of light as approximately 300,000 km/s, it takes almost 70 milliseconds for light to travel between the 2 farthest points on the planet. To apply this measure to sound and for comparison purposes, 70 ms represents the time it takes for sound to travel between two points distant by almost 24 meters (340 m/s is the approximate speed of sound in air). This distance is only slightly beyond the size of a regular concert stage or a recording room where musicians are expected to play in synchrony. Most of the latency actually comes from network overheads resulting in speeds of twice the speed of light at best (Chafe, 2003).

The speed of light provides a convenient hard theoretical limit, but if we add the practical constraints of computer networks, transmission delays easily reach hundreds of milliseconds. As an example, here are some average estimates of round trip times using the *ping* tool between the US East Coast (MIT Media Lab network) and the following locations on the Internet:

- another computer on the same network (www.mit.edu): 50 ms,
- a computer on the US West Coast (www.stanford.edu): 132 ms,
- a computer in France (www.polytechnique.fr): 176 ms,
- a computer in India (www.sify.com): 315 ms[1]

[1] interestingly it is rather difficult to find an academic or public office website hosted in India that enables echo request messages (*ping*).

Studies on musician synchronization (see Section 2.1) have shown that musicians need to hear each other within a 20 ms window to remain synchronized. Therefore, creative solutions are required to overcome the transmission delays that we find on regular computer networks.

Barry Vercoe's original idea, NetDuet (undocumented personal communication), gave me the idea of solving the problem of network music performance with a predictive system.

Additionally, personal observations and discussions with musicians suggest that visual contact is not a determining factor in a musical interaction. Therefore, this system assumes that the fact that musicians cannot look at each other while playing together over the network does not significantly affect their collaborative effort—at least, the purpose of their collaboration remains intact (for a caveat look at Section 5.3).

Approach

My approach in this thesis is based on the premise that in order to cancel transmission delays and enable musicians to remain synchronized, we need to predict each musician's intent even before the sound that he or she produces reaches its far-end destination. A realization according to this principle is enabled by technology in the areas of machine listening and machine learning.

This approach implies that for the next musical events to be predicted a few milliseconds in advance, a suitable model of the music should be developed. Therefore the system should be tuned to a particular musical style, and even a particular instrument. In this case, I chose the tabla, a pair of Indian hand drums, not only because of its popularity and my familiarity with it, but also because of its "intermediate complexity" as a percussion instrument: although tabla patterns are only based on rhythmic compositions without melodic or harmonic structure, different strokes can produce a variety of more than 10 pitched and unpitched sounds called *bols*, which contribute to the tabla's expressiveness (see Section 2.2). The tabla is proposed as a case study to prove the feasibility of my approach.

Thus, I propose to develop a computer system that enables real-time online musical collaboration between two tabla players. Although, as I mentioned, the design presented in this thesis is specific to Indian percussions (in particular, the system was evaluated with the tabla in the North Indian *Hindustani* musical style), I expect that the principles presented here can be extended and generalized to other instruments and cultures.

Tabla duet may not the most common ensemble, and it might have been better to choose two instruments that have a more fully developed "duo" modality of playing.

One approach could be to combine the TablaNet system on one side with a streaming audio system (for a vocal or instrumental performance) on the other.

The point of using a real tabla instead of an electronic controller is to enable the audience to participate in a "real" concert at each end of the interaction with musicians playing on acoustic instruments. It could be interesting to add support for music controllers or even hand claps, but not at this stage of the project.

Terminology

The word *stroke* used in this thesis does not stand so much for the gestures that generate a particular sound (by hitting the tabla with the hands in a specific fashion), but designates the resulting sound itself. It should be noted that different musical schools within the North Indian tradition use different gestures (with slightly different sounds) for similar stroke names. This is dealt with by training the (player-dependent) stroke recognition engine for each user.

The term *musical collaboration* encompasses different types of interactions two musicians might be involved in. In the case of two tabla players in a single performance, which is a relatively unusual configuration (there is usually one drum player, or at least one per type of drum, in traditional Indian music ensembles), I consider the following possible scenarios: a student-teacher interaction in an educational setting, and a call-and-reponse interaction in a performance setting. Rhythmic accompaniment, which is the usual role assigned to the tabla player, typically takes place with a melodic instrumentalist or a vocalist, and will therefore not be taken into account in the scope of this project.

In this thesis, I use the terms *near-end* and *far-end*, common in networking—especially in the fields of videoconferencing and echo cancellation—to describe each side of the interaction. The near-end refers to the side where the tabla signal is acquired. The far-end applies to the distant location where the transmitted signal is played back (to another musician).

1.3 Methodology

Preliminary Studies

Before embarking on this research project, I performed two initial studies.

I conducted preliminary work where I demonstrated the concept elaborated in this thesis by sensing vibrations on the tabla drumhead, analyzing stroke onsets, and transmitting quantized onset events and the tempo over a non-guaranteed connection-

less UDP (User Datagram Protocol) network layer. On reception of the events, the receiver would trigger sampled tabla sounds of a particular rhythmic pattern stored in a predefined order. This application was prototyped in the Max/MSP environment and demonstrated in October 2006.

In addition, I developed a stroke recognition algorithm as a final project for a class on pattern classification. This non real-time algorithm, developed in Matlab, was used as the basis for the stroke recognition engine developed in this thesis, and then extended for greater accuracy and implemented in C for real-time performance. The preliminary algorithm was presented in December 2006.

The work described in this thesis was partly published in an article by Sarkar and Vercoe (2007).

Research

The main hypotheses driving this work are:

1. playing on a predictive system with another musician located across the network is experientially, if not perceptually, similar to playing with another musician located in the same room in that it provides as much "satisfaction" to the musicians and the audience;

2. a recognition and prediction based model provides an adequate representation of a musical interaction; and

3. a real-time networked system suggests new means of collaboration in the areas of distance education, real-world and virtual-world interactions, and online entertainment.

The TablaNet system described in this thesis not only develops a solution to address the problem of distant musical collaboration, but also provides a platform, an artifact, to evaluate the previous hypotheses. These hypotheses are evaluated based on subjective tests with users—tabla players and trained listeners.

Technology

This project involves many layers of complexity in terms of the technical fields involved (see Figure 1-1). The red boxes (second and third rows) represent the competencies that were required to develop the system. The blue boxes (fourth, fifth, and sixth rows) represent the underlying technologies.

The mechanical study consisted in carefully selecting the optimal models of vibration sensors, as well as placing and attaching them to the tabla heads. The vibration

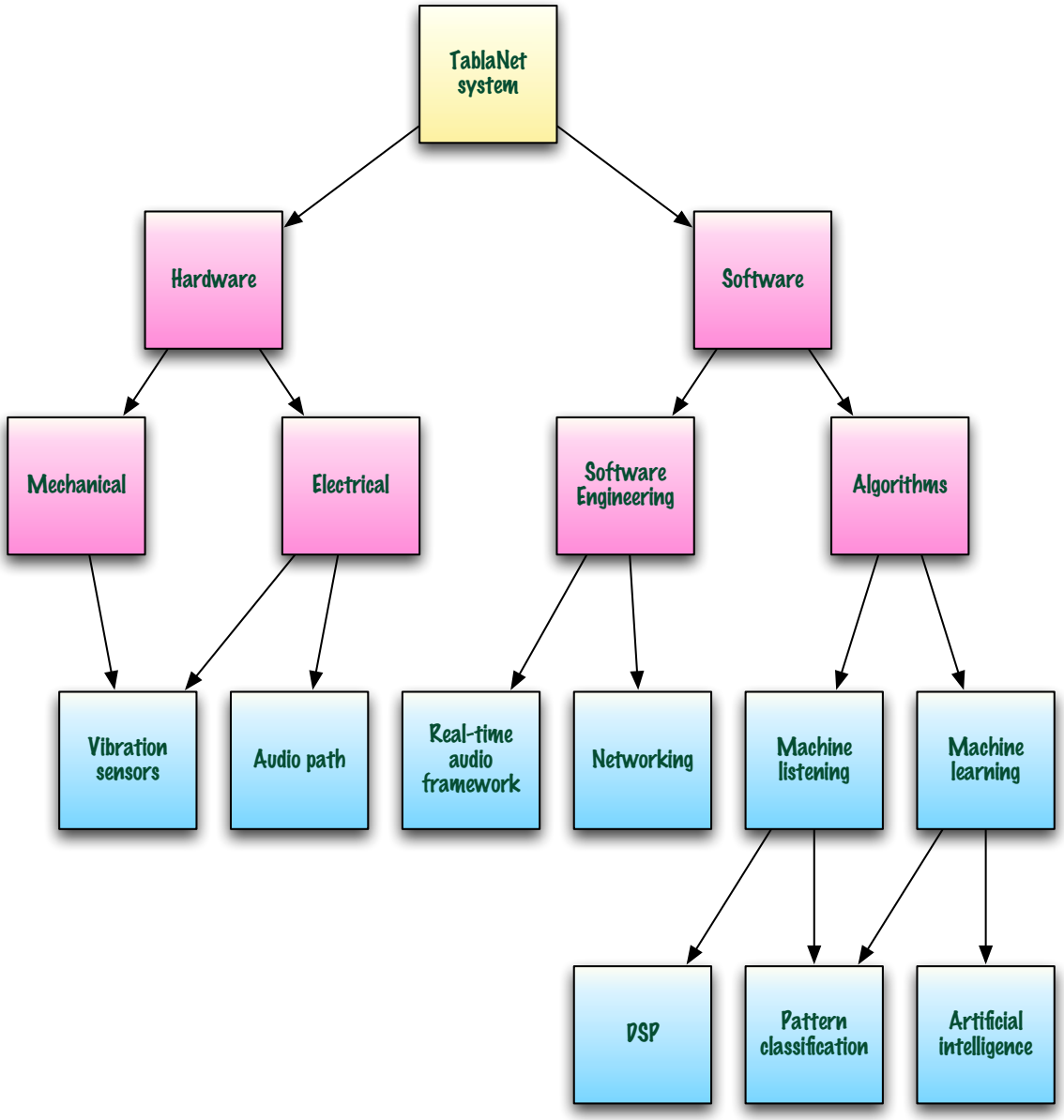


Figure 1-1: Technology employed in the TablaNet system

sensors were also included in the electrical development, which dealt mainly with the audio path originating at the sensors, up until the computer input, and then again from the computer output to the speakers.

The software design and implementation comprised the bulk of this work. I applied standard software engineering techniques to design an audio framework with services to access the default audio device for real-time audio, and provide audio file input and output, and networking capabilities. The algorithms—the main part of the research undertaken in this project—rely on machine listening and machine learning models, which, in turn, are based upon digital signal processing, pattern classification, and artificial intelligence techniques.

1.4 Thesis Outline

Chapter two, *Background*, reviews the literature relevant to this work, in particular networked music performance systems, and offers a description of the tabla in Indian music.

Chapter three, *Design*, describes the TablaNet system, hardware, and software implementation details.

Chapter four, *Evaluation*, discusses quantitative as well as qualitative tests performed on the TablaNet system.

Chapter five concludes this thesis by summarizing its contributions, and suggesting applications and future work.

Chapter 2

Background

This background presentation details two bodies of work relevant to the TablaNet system. First, I survey the literature related to networked musical collaboration, which has been the subject of numerous projects. Then I describe the tabla in the context of Indian music. Finally I review the literature on tabla analysis and synthesis which leads to my work on tabla stroke recognition and phrase prediction.

2.1 Network Music Collaboration

Much work has been done on musical collaboration over computer networks. I organize here the relevant literature into several categories depending on the technology employed, or on the intended application area.

I first describe notable commercial endeavors brought on by the Internet. Then I veer off towards philosophical considerations and proposed roadmaps. I continue with works that describe other types of musical interactions made possible by computer networks. Then follow studies that deal with musician synchronization; studies which have, in some instances, led to the design of architectural frameworks that convey sensory modalities other than audition only. Then I list projects that use a particular technology: from event-based and early MIDI systems to audio streaming with the latest compression algorithms, via phrase looping schemes; I also look for previous work on predictive systems. I follow it up with projects that highlight a particular application area rather than hinge on design elements, such as distance education and virtual studio applications. I end this section with a non-exhaustive survey of major live network music performances that have taken place in the past ten years.

Commercial Endeavors

Since the advent of the Internet, musicians have been looking at online musical collaboration as the next "killer app." In fact, this space has been and continues to be

the source of several commercial endeavors (from the defunct Rocket Network (1998) to Ninjam (2005), Audio Fabric (2007), and Lightspeed Audio Labs (2007), a new startup which just released a beta version of its software).

Rocket Network pretty much started it all. As early as 1998 they provided what their marketing team called "high-end Internet recording studio technology" to audio software companies (like Digidesign or Steinberg), which would bundle it with their popular sequencing software (e.g. ProTools or Cubase) for musicians to use. On closer inspection, it appears that Rocket Network did not provide real-time collaboration, but merely an asynchronous interface to "play locally, then post" (to a central server)—a way to extend the concept of multitrack overdubs beyond the walls of a recording studio by reaching out across the Internet to musicians from all over the world (Townley, 2000). Despite its revolutionary model, the limited bandwidth available on the Internet in the 1990s may have compromised its growth. The company was bought by Digidesign, and its technology (or rather its ideas) was incorporated into ProTools. In the same line, VSTunnel (2005) is a sequencer plug-in that functions much on the same concept as Rocket Network (i.e. non real-time collaboration with a sequencer).

digitalmusician.net (2006) and their Digital Musician Link (DML) plug-in deliver another virtual studio technology based on a peer-to-peer connection between two users. In addition, the website acts as a virtual community portal and message board that helps artists meet each other. Similarly, musicolab (2003) proposes a P2P file sharing network and an upcoming (as of two years ago) community site.

Ninjam, which was developed by the authors of popular programs for the Internet-age like Winamp and Gnutella, proposes a high-latency technique for "jamming" similar to the one introduced by Goto and Neyama (2002). Ninjam works with a central server and distributed clients. Each client, one on each participant's computer, records and sends a compressed audio stream (in OGG Vorbis format) to the server. The server then sends a copy of all the streams delayed by one measure (or one musical interval) back to every participant. In effect, each user plays along with everyone else's previous interval. This is a creative solution to the latency problem, but it sets external constraints on the music (for instance, a drum set and a guitar work well together whereas several melodic instruments improvising together may not). It is also hard to imagine a call-and-reponse interaction on this system (the first musician plays one measure then waits one measure for the other musician to hear his contribution, then has to wait yet another one for the other musician to play his part before finally hearing it back?). As the Ninjam website puts it, it is "part tool, part toy."

In February this year (2007), Audio Fabric released its real-time network jamming platform. According to the company website, the system deals with latency by "utilizing low-delay audio codecs." By targeting musicians within constrained geographical boundaries (coast to coast within the US for instance), Audio Fabric claims

to "achieve one-way audio streaming delays of 40 ms or less across moderate physical distances (...) under optimal system and Internet conditions." They also point out that a delay of up to 40 ms is acceptable for musician synchronization.

In June 2007 (which seems to demonstrate that this is a hot topic!), Lightspeed Audio Labs launched a similar "music-based social networking site." Although few technical details are available on their website, their technology also seems to rely on a proprietary audio codec and optimized streaming techniques. As a starting point, they limit themselves to a geographically-bound area on the US East Coast thereby constraining the network latency to controllable limits. An innovation that distinguishes Lightspeed Audio Labs' product from the competition is its audience participation feature: members of the audience, who, unlike the musicians, can be located anywhere on the planet, can holler at the musicians through their microphone as if they were in the same physical room. Other audience members and even the musicians who are located in the same virtual room can hear them. This is certainly a very interesting feature that is applicable, if not to Western classical music, to rock or pop gigs as well as traditional Indian music where, as we shall see in Section 2.2, the audience expresses its approval with vocal gusto.

Other musical collaboration sites with a social networking twist (or the reverse) include indabamusic (2007) for non real-time collaboration, and eJamming (2007) for real-time performance (with documented lag). Jamglue (2007) and splice (2007)are two other websites where one can upload music and allow others to remix it. This seems like quite a crowd for what may be considered as a niche market.

Although the systems described here propose attractive tools to foster online musical interactions and community building, they fall short of enabling worldwide collaboration in real-time involving all kinds of non-mainstream and non-Western music.

Philosophical Discussions and Roadmaps

Atau Tanaka, a pioneer in the area of network music performance, surveyed the field between 1994 and 1999 (Tanaka, 1999). In his paper, he describes early projects over dedicated ISDN digital telephony lines, and later ones over the Internet. By then, the goal of performing over a network had already given birth to two schools of thought: those that transmit messages or events, and those that send audio streams. Importantly, Tanaka advocates for a new aesthetic taking into account the "acoustic" of the network, "much in the way the reverberation time in a cathedral differs from that of a jazz club." He also observes "an interesting phenomenon of non-congruous simultaneity" due to which the performance may result in a different aural experience at each site by which a single music has "simultaneous multiple interpretations." This remark naturally applies to the the TablaNet system I present in this thesis.

Kon and Iazzetta (1998) establish the convergence of multimedia and networking technologies that enable distributed musical applications, and describe the challenges and opportunities of Internet-based interactive music systems.

In his work on interconnected music networks, Weinberg (2001, 2002, 2005a,b), who has been involved with them since his MIT Media Lab days, discusses their philosophical and aesthetic foundation with a particular focus on social dynamics and group interdependency towards music composition or improvisation. After a historical review of compositional works that highlight the role of interconnected networks, Weinberg describes two novel network music controllers that he developed: the *Squeezables* and the *Beatbug Network*. Furthermore, Weinberg argues that the network architecture influences the way musicians interact. He then goes on to propose various topologies and describes the kind of interactions that each one supports.

Other Types of Interactions

Tod Machover's Brain Opera (1996) is a "musical experience that includes contributions from both on-line participants and live audiences." It uses hardware and software interfaces for musicians of all levels to interact with professional performers (Paradiso, 1999).

With his JamSpace interactive music environment, Gurevich (2006a,b), now at Stanford, proposes real-time jamming over LANs, which are characterized by their low latency (typically around 1 ms according to Chafe et al. (2000)). By enabling rhythmic collaboration between beginner musicians over a network, Gurevich emphasizes group interactions while preserving anonymity.

Researchers have found other creative ways to interact musically over the Internet, for example by converting transmission delays into reverberation. The SoundWIRE team from Stanford's CCRMA is notable for using the Internet's "acoustics" (the transmission delay transformed into reverberation) not only to gauge the network's performance by "listening" to it (Chafe et al., 2000; Chafe and Leistikow, 2001), but also to create, following Tanaka's suggestion, music that matches the environment it is played in (Chafe et al., 2002; Chafe, 2003).

As an alternative to the Internet, some projects use ad-hoc wireless mesh networks to obviate the issue of latency and provide low cost computer platforms (e.g. the $100 laptop) with a networked version of CSound for children to collaborate musically with each other (Vercoe, 2006).

Puckette developed a PureData opcode to detect percussion strokes and re-create them over a network for his Lemma improvisatory jam sessions which were part of the Global Visual Music Project.

Studies on Musician Synchronization and Frameworks

Few research groups have extensively studied issues concerning musician synchronization with regards to network performance. Chris Chafe from the SoundWIRE group at CCRMA studied the effect of delay on rhythmic accuracy (Chafe et al., 2004). The experiment involved a pair of subjects placed in isolated rooms who were asked to clap together. A variable delay (ranging from 0 ms to 77 ms) was introduced in the sound heard from the other subject. The results show the existence of an ideal delay of 11.5 ms that preserves ensemble accuracy. A shorter delay tends to result in an increase in tempo, whereas a longer delay slows down the performance.

An explanation for this phenomenon may be found in the work of Povel and Okkerman (1981): their experiments demonstrate that the variation of time interval between two tones equal in all respect affects the perception of the accent (and therefore that of the beat) in a rhythmic sequence.

A multidisciplinary team at the University of Southern California (Chew and Sawchuk, 2004; Chew et al., 2004, 2005) has conducted several experiments on realistic musical interaction (involving real ensembles rather than subjects in a non musical setting) where a reduced physical presence has been explicitly taken into account. Their studies led to the development of a synchronous collaboration framework called Distributed Immersive Performance (DIP), which transmits MIDI, high-fidelity multichannel audio (enabling spatial sound localization), and video streams over the Internet2 (Sawchuk et al., 2003). Experiments around the DIP platform have refined some of the findings of the CCRMA team. In particular, it was found that latency tolerance depends on the tempo and rhythm of the piece as well as the timbre of the instruments used, which seems to confirm the existence of time and amplitude thresholds in human beat detection as described by Povel and Okkerman. Another experiment with a duo of professional pianists showed that the musicians "tolerated" delays of up to 50 ms. However, if each musician heard himself or herself with a delay equal to the simulated latency from the other musician, delays were tolerable up to 65 ms (with practice).

By using various combinations of instruments, Nishibori et al. (2003) demonstrate the importance of timbre, and in particular of the amplitude envelope attack, for delay identification: as expected, their results show that percussive sounds, like snare drum and piano, are the most sensitive to time delays.

Another team from Helsinki University of Technology studied the importance of tactile feedback in latency tolerance (Mäki-Patola and Hämäläinen, 2004a,b) not so much for network music performance but for physical model-based musical controllers. A Theremin was used for the experiments. It was found that subjects detected the presence of latency when it was above 30 ms. The latency in this situation is the delay between the causal gesture and the resulting sound. It was also shown that age plays a role in detecting latency. In a review paper by Mäki-Patola (2005), the various

factors that play a role in latency perception and tolerance (e.g. timbre, familiarity with a piece or an instrument) are discussed.

Inclusion of Other Modalities

The network music performance (NMP) system described by Lazzaro and Wawrzynek (2001) is implemented between remote sites in California, which keep the network latency within reasonable limits. However, and this is a significant departure from previous approaches, the authors propose, in order to restrict packet loss while sending a combination of MIDI, structured audio and streaming audio, to encode and transmit an abstract representation of the musical gestures, rather than sending the resulting audio signal by itself.

Along the lines of the Immersive Performance project, Cooperstock, from McGill University, researched the importance of multiple modalities in networked audio (Cooperstock et al., 2004). In addition to visual sensory data through video, the author emphasizes the importance of multichannel audio as well as low-frequency vibrations in an immersive environment suitable for videoconferencing or network music.

Architectural Proposals and Design Choices

The AES white paper proposed by Bargar et al. (1998) makes the case for audio and music applications on the Internet2 backbone (1996), and sets forth several network applications including music performance and music education.

Papers by Fober et al. (2002, 2001) address real-time transmission over high latency networks and rendering of time-ordered events such as the ones found in musical structure. The authors propose a mechanism to compensate for latency variations and clock frequency differences (clock drift) that is an improvement over existing clock synchronization protocols.

Bouillot (2003) presents a synchronization algorithm that compensates for time-varying network latency. The distributed concert architecture that he describes imposes that the musicians, who may be located at various venues, play slightly in advance of the audio feedback in order to compensate for the delay. Therefore the need for a constant delay. In this architecture, the collated audio streams are mixed by a sound engineer and projected to a single audience which is located in a traditional physical space.

A collaborative of MIT Media Lab students (Jehan et al., 2004) described the use of OpenSound Control (OSC) as an underlying communication technology for their interactive multimedia projects. OSC was originally developed at CNMAT (UC

Berkeley) and was subsequently used as a communication protocol for the majority of event-based network music performance systems (as a successor to the less powerful and less flexible MIDI format). A paper on the state of the art of OSC (Wright et al., 2003) proposes a tutorial overview and lists several implementations made available.

Quintet is an "interactive networked multimedia performance environment" that lets five musicians play together over a network (Hajdu, 2007). By using digital event-based instruments (MIDI synthesizers and sequencers, sensors) and playing compositions tailored to the limitations of the Internet as a medium, the system combines composition and improvisation with a conductor-based approach to synchronize control streams produced by the musicians and enable live musical collaboration.

Event and MIDI-based systems

One way to limit the end-to-end delay is to limit the size of the data packets transmitted over the network. That is the philosophy followed by ResRocket, the precursor to Rocket Network. From 1994 onwards, the team developed a "technology that allowed stacking of MIDI files by multiple parties across the Net to make a joint musical effort" (Townley, 2000). As we saw in our survey of commercial websites, this concept is still in use today. In 1997, shortly before Rocket Network was launched, the company even experimented with real-time jamming via MIDI.

In 1998, Dutch researchers proposed a browser plug-in to display musical scores on the web and enable MIDI-based live jam sessions where all the clients connect to a central server (Eliens et al., 1997). The system is described but no evaluation data is given.

Phrase Looping Systems

In 2002, Goto and Neyama proposed Open RemoteGIG, a distributed jam session system which pioneered the "one-phrase delay" concept that inspired Ninjam. Based on the assumption that certain types of music have repetitive chord progressions and constant tempo, the remoteGIG system transmits MIDI events over the Remote Music Control Protocol (Goto et al., 1997).

Researchers at Yamaha (Nagashima et al., 2003) describe several network music interaction projects based on a Global Delayed Session (GDS), similar to Goto's concept (i.e. the system delays every player's contribution so that each 4-measure phrase is synchronized).

Another team from Japan improved Goto's system by transmitting audio streams rather than MIDI and called it Mutual Anticipated System (MAS) (Yoshida et al.,

2005). They also developed a system that allowed fluctuating tempo for more expressive performances (Yoshida et al., 2004).

Interestingly most of this research is based in Japan, with the exception of Ninjam's implementation. Coincidence or could there be a cultural reason?

Audio Streaming-based Systems

To overcome the limitations of sound resynthesis at the receiving end, several systems stream audio instead of transmitting messages across. Most approaches use a perceptual compression scheme similar to the one defined in the MPEG 1 audio standard (Brandenburg and Stoll, 1994).

In 2000, researchers demonstrated high-quality streaming of 12 channels of uncompressed PCM audio between McGill University in Canada and the University of Southern California in the US over the high-speed Internet2 network (Cooperstock and Spackman, 2001).

Chatwani and Koren (2004) from Princeton University investigated a practical approach where channel parameters were tuned to improve the perceivable quality of compressed audio streams over cell-phone networks.

Gu et al. (2004) describe a networked music performance system that uses AAC audio compression to enable rehearsals over a local area network. Their approach, however, may not be scalable to wide area networks because it makes the explicit assumption of low latency characteristics.

These systems avoid delay overheads above network latency either by using PCM audio, which requires no decoding, or by employing audio compression techniques with reasonable algorithmic complexity. However one common problem faced by all these systems regardless of their data format is the fact that there is no guarantee of service on the Internet, meaning that applications have no control over the network routing and priority schemes, and are basically tributary to external network conditions: even if variable latency is not an issue, no provision is usually made for network congestion peaks, which may increase transmission times by many factors above average values.

Predictive Systems

CCRMA's Chris Chafe, always interested in network musical performance, did some early research about the prediction of solo piano performance (1997). There he describes his attempt at modeling human aspects of a score-based musical performance such as note timing and velocity. Covariance analysis is used to predict closeness

between a candidate interpretation and stored ones, thus enabling anticipation when confronted with a delayed transmission.

Distance Education

As early as 1999, Young and Fujinaga describe a system for the network transmission of a piano master class by MIDI over UDP. In a similar work, Vanegas (2005) proposes a MIDI-based pedagogical system that brings together a music teacher and a student via the Internet. Although he does not propose a solution against network latency, he acknowledges that the problem exists even with a MIDI-based system.

Virtual Studios

In her master's thesis at the MIT Media Lab, Lefford (2000) tackles cognitive and aesthetic issues surrounding network music performance in the particular context of a producer recording an overdub from a distant musician.

Networked Music Performances

Kapur et al. (2005) trace a history of interactive network performance, propose network optimizing strategies, and describe the technology, tools and framework used in the Gigapop Ritual network performance between McGill University in Canada and Princeton University in the USA (Kapur et al., 2003b). Kapur documents a 120 ms latency. The strategy to counteract the delay is to have a "leading" site (located at Princeton) while distant performers (at McGill) "follow." The followers end up listening to the incoming audio streams and "react" rather than "initiate." This experiment illustrates the fact that the amount of delay that causes synchronization problems depends on the musical structure and style.

Chew, Kapur, and Weinberg all document several distributed live concerts that have taken place in the past. Most of them however are to be considered as experimental because they are usually one-time events produced to demonstrate a certain technology.

Recently, the SoundWIRE team performed a series of 4-way concerts that involve four international venues. These concerts use JackTrip, a multichannel multi-machine high-quality (uncompressed) audio streaming program over Internet2 developed at CCRMA.

2.2 The Tabla in Indian Music

Indian music encompasses an extremely wide variety of musical traditions and styles. Considering the focus of the TablaNet system, I shall present in this section aspects of the tabla and its place within North Indian classical music that are relevant to the technical discussions that follow.

North Indian Music

North Indian music, in particular instrumental music which does not have the obstacle of language, has spread all over the world, in some way following Indian professionals and students in their travels. However this phenomenon can also be attributed, in part, to the inclusion of Indian musical elements in Western popular music (since the Beatles), as well as the generalized trend of internationalization in music, and the increased awareness and interest in non-Western music by the West.

For many centuries, Indian music was performed solely in temples and transmitted within families from one generation to the next. Music was patronized by the nobility and not to be performed at public venues. However, after India gained its independence in 1947, there was a radical change. With the disappearance of the *zamindar* class (landlords), musicians started to perform in public concerts, which middle class families would attend. And it became common for children from those families to learn and practice music and dance (Ruckert, 2004).

Although colleges of music have made their appearance since the past century, many students continue to learn from a *guru*. Music education is done, even today, through oral repetition. A sort of call-and-reponse pattern in which the student hears and then repeats basic exercises until mastering them. This training process, which educates the ear as well as the voice or the fingers often lasts for many years before the student learns his or her first composition.

One way to categorize Indian classical music is by the presence of rhythm: one type is free, and the other is metric. In this thesis, I am of course primarily concerned about the music that is metric, called *nibaddh* (bounded).

In his introductory book on the music of North India, Ruckert (2004) establishes three entry points by which to learn *about*[1] Hindustani music:

- the devotional component, which permeates most if not all activities of life, including music,
- fixed compositions and improvisations, or the balance between "preservation and creation", and

[1] notice the emphasis here: I mean "learn about Hindustani music," rather than "learn Hindustani music."

- the verbal syllable, used to name both pitches and elements of rhythm.

Rhythm in North Indian Music

This section introduces concepts of Indian drumming and the related terminology.

Verbal syllables are used, in various forms, as building blocks for rhythmic phrases. The *jatis*, found originally in Carnatic music, entered North Indian music through dance performances. A concatenation of syllables like *ta ka di mi* where each syllable stands for one beat, they represent basic rhythmic units of one to nine beats.

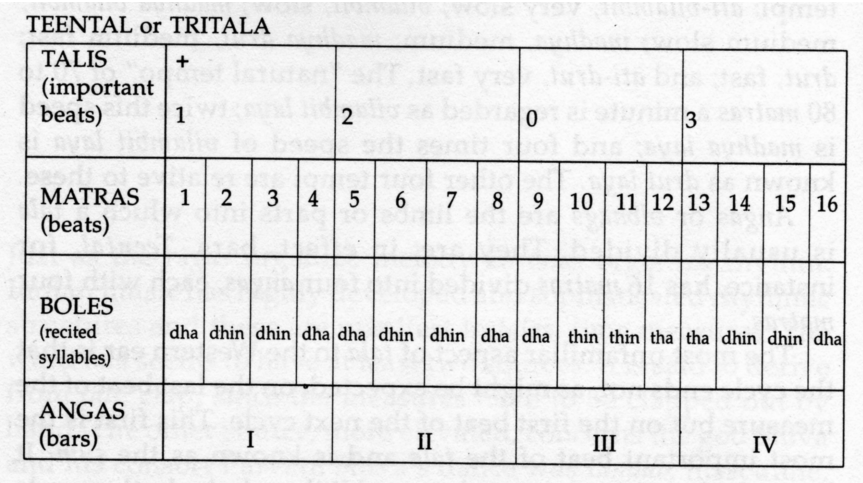
The *chhand*, basically the pulse (accents), can be considered as the meter of successive measures. It can either be regular, or irregular (e.g. 2 beats followed by 3, then by 4). The *laya* is the tempo. It is based on a relative speed of approximately 70 to 80 beats per minute (for the slow laya). Medium speed is twice as fast, and the fast speed doubles it again. The beats themselves are called *matras*. Based on the previous numbers, a typical matra has an approximate duration of $\frac{3}{4}$ to $\frac{6}{7}$ of a second (Reginald and Massey, 1996)

Ruckert also talks about poetic meters and a mnemonic device taught to him by Swapan Chaudhuri, a well-known tabla performer, to remember them: *ya - maa - taa - raa - ja - bhaa - na - sa - la - gaam*. Syllables with one 'a' are short, and those with two ('aa') are long. "Legal" combinations of meters can thereby be remembered—for instance: *short - long (ya - maa); short - long - long (ya - maa - taa); etc.*

The concept of *tala* is central to rhythm in Indian classical music. Talas organize rhythm much like *ragas* organize melody. They are rhythmic cycles that group short and long measures (*vibhags* or *angas*). In theory, there are 360 talas which range from 3 to 108 beats, although only 30 or 40 are in use today. Divisions between each section are marked by claps *talis* and waves of the hand (called *khali* which means "empty"). Arguably the most common tala nowadays is the 16 beat *tintal*, which literaly means "three claps" (see Figure 2-1). It is comprised of four vibhags (measures) of four matras (beats). The + sign indicates the *sam* ("equal"), the first and most important beat. Importantly, a rhythmic cycle does not end on the last beat of the measure but on the first beat of the next cycle. The *talis* numbered 1, 2, and 3 indicate claps, whereas the one denoted 0 indicates a khali.

As we can see in Figure 2-1, each beat corresponds to a *bol*, a syllabic name for each type of drum stroke. The pattern of syllables that identify the tala is called the *theka*. It represents a template for each tala which is otherwise defined by its number of beats and grouping pattern.

Counting exercises (*gintis*) are common for students of Indian percussions. Because of their seeming irregularity, they built up a sense of tension until their final



TEENTAL or TRITALA

TALIS (important beats)	+ 1				2				0				3			
MATRAS (beats)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
BOLES (recited syllables)	dha	dhin	dhin	dha	dha	dhin	dhin	dha	dha	thin	thin	tha	tha	dhin	dhin	dha
ANGAS (bars)	I				II				III				IV			

Figure 2-1: Rhythmic structure of Tintal (Reginald and Massey, 1996)

resolution on the sam, the very last beat.

A *kayda* is a composition which is based on a combination of thekas with variations on them. The *tihai* (meaning "one-third") is a central feature of rhythmic compositions. It consists in a repetition, three times, of a particular phrase. It usually denotes a break or a conclusion. The length of the tihai is usually calculated so that it finishes exactly at the downbeat on the sam, after several background repetitions of the underlying rhythmic cycle.

It should be noted that beyond accenting the pulse, the drummer "plays a fundamental role in the intricate counterpoint of the raga's performance" (Khan and Ruckert, 1991). In most instances, the drummer maintains the theka with minor ornamentations while the melodic soloist improvises, and then a drum solo may take place during which time the soloist repeats the main line of the composition in the background. Thus there is always one musician who keeps a steady beat while the other one improvises.

The literature also mentions "duels" between two expert tabla players where musicians battle in a call-and-reponse fashion and each one tries to outdo the other in invention, stamina, and repertory.

The Tabla

> *I think there are more excellent tabla players today than ever before, and they are doing great new things with the tabla!* – Zakir Hussain

Hand drums are essential to Indian music, and they are truly a living tradition with many students and practicing musicians worldwide. The tabla is the main drum in North India. It is comprised of a pair of drums (see Figure 2-2). The right-hand

Figure 2-2: A tabla set

Table 2.1: Tabla strokes

Bol	Na	Tin	Ga	Ka	Dha	Dhin	Te	Re	Tat	Thun
Dayan (right drum)	✓	✓			✓	✓	✓	✓	✓	✓
Bayan (left drum)			✓	✓	✓	✓				

drum is called the *dayan* and the left-hand drum is called the *bayan* (as seen from the player's perspective). One principal acoustic feature of the tabla is its ability to contrast open (*khula baj*) and closed (*bandh baj*) sounds on the bayan, which is important in recognizing thekas and has probably contributed to the widespread success of the tabla.

It must be noted that the position of the drums (left and right) are described here from the point of view of the player, not the audience. Interestingly, the literature varies on the adopted convention, probably depending on whether the author is a tabla player or not.

Table 2.1 lists some of the most common bols and indicates how they are played.

2.3 Tabla Analysis and Synthesis

In this section, I review the literature related to tabla sound analysis and synthesis. Interestingly, there is some overlap in the list of researchers mentioned in the survey on Network Music Collaboration (Section 2.1) and the one in this section. However, apart from some, like Kapur, most of the overlapping researchers seem to have worked on the two areas independently.

This section starts by introducing studies on the recognition of spoken tabla bols, and follows with works on the recognition of tabla strokes (also called bols—I make an arbitrary distinction in the section titles here)

Recognition of tabla bols

Chatwani (2003) developed a computer program based on linear predictive coding (LPC) analysis to recognize spoken bols. Samudravijaya et al. (2004) also implemented a bol recognizer with cepstrum based features and an HMM. They report above 98% recognition on a limited dataset.

Patel and Iversen (2003) presented an interesting study where they performed an acoustic and perceptual comparison of spoken and played tabla bols. They found that spoken bols do indeed have significant correlations in terms of acoustical features (e.g. spectral centroid, flux) with their gestural counterpart, and that untrained listeners were able to match syllables to the corresponding drum sound. This provides strong support for the symbolic value of tabla bols in the North Indian drumming tradition.

Recognition of tabla strokes

In 2003, Gillet and Richard presented a paper on tabla stroke recognition often cited in later research. Their approach follows three steps: stroke segmentation, computation of relative durations (using beat detection techniques), and stroke recognition. Transcription (stroke recognition) is performed with a Hidden Markov Model (HMM). The advantage of their model is its ability to take into account the *context* during the transcription phase (e.g. the bols *Te* and *Ti* correspond to the same gesture, but the choice of the syllable depends on the neighboring bols). They integrated their model into a graphical environment called TablaScope.

Chordia, who completed his PhD thesis at CCRMA on tabla transcription, implemented a system that segments and recognizes tabla strokes recorded during real performances (with large datasets). He extended the work of Gillet and Richard by trying different classification algorithms, including neural networks, decision trees, and a multivariate Gaussian model. He acknowledges player-dependent recognition rates of up to 94%.

Other researchers have attempted to classify drum sounds other than the tabla (Herrera et al., 2002, 2003). After investigating various features and various classification methods (k-nearest neighbor, kernel density estimation, canonical discriminant analysis, and decision trees), they arrive at a list of about twenty useful features, and report classification rates above 90% with specific taxonomies (identification of the drum sound category (e.g. "resonant", or "closed") rather than the original sound source).

Tindale et al. (2004) evaluated how computers identified different playing techniques on single percussion instruments (i.e. different strokes on the snare drum).

Temporal and spectral features were extracted and fed to a neural net. The researchers reported around 90% recognition accuracy. In 2005, Tindale published a survey of related work in drum identification and beat detection; most of the studies they mention is presented in this section. Later work by the same authors describes percussion sound recognition using pattern recognition techniques (ZeroR, Gaussian, k-nearest neighbor, and neural nets).

Van Steelant et al. (2004) used Support Vector Machines (SVM) to recognize percussive sounds.

Tabla acoustics

The first modern account of the tabla's tonal quality can be found in Nobel laureate C.V. Raman's work (1920; 1935). In his two papers, Raman mentions that the tabla (like the mridangam, a longitudinal South Indian percussion instrument with two heads—similar to the tabla's—opposite each other) differs from other percussion instruments, which usually produce inharmonic overtones, in that "it gives harmonic overtones having the same relation of pitch to the fundamental tone as in stringed instruments". In fact, he highlights the importance of the first three to five harmonics which are derived from the drumhead's vibration modes in the instrument's sound. He describes how the position of the fingers and the type of stroke on the drumhead excite the membrane along some of its nodes and add different harmonics to the sound.

Although Bhat (1991) specifically studies the *mridangam*, he develops a mathematical model of the membranes' vibration modes that could well be applied to the tabla. In particular, his model explains the harmonic overtones found in the mridangam, which account for its tonal quality, when its membranes are excited by particular strokes.

Malu and Siddharthan (2000) confirmed C.V. Raman's observations on the harmonic properties of Indian drums, and the tabla in particular. They attribute the presence of harmonic overtones to the "central loading" (black patch) in the center of the dayan (the gab). This black patch is also present in the bayan, but there it is placed asymmetrically. They solve the wave equation for the tabla membrane and identify its vibration modes.

Tabla sound synthesis

Essl et al. (2004) applied their theory of banded waveguides to highly inharmonic vibrating structures. After initially discussing software implementations of simple percussion instruments like the musical saw, glasses and bowls, they present a tabla model.

Tabla controllers

This section is concerned mostly with systems that convert musical gesture into sound in the realm of percussion instruments.

In 1995, Hun Roh and Wilcox developed (as part of Hun Roh's master's thesis at MIT) a system for novices to discover tabla drumming. A rhythmic input tapped on a non-specific MIDI controller is mapped to a particular tabla phrase using an HMM. The tabla phrase that corresponds to the drumming pattern is then played back with the appropriate bol sounds.

Kapur (2002); Kapur et al. (2003a, 2004) proposed the ETabla (along with the EDholak—the dholak is another Indian percussion instrument—and the ESitar). The ETabla identifies strokes by capturing gestures with Force Sensing Resistors (FSRs) placed on a non-acoustic tabla controller head. Strokes are recognized using a tree-based classifier. The ETabla allows traditional as well as new performance techniques and triggers sounds (synthesized with Essl's banded waveguide model) and graphics.

Beat tracking

The seemingly simple human skill of identifying the beat in a piece of music is actually a complex cognitive process that is not yet completely understood. In fact, less experienced musicians sometimes find it difficult to identify the "correct" beat, tapping instead at twice the speed, or twice slower, compared to professional musicians. Similarly, beat tracking proves to be a difficult task for computers.

Allen and Dannenberg (1990) propose a survey of artificial beat trackers. Then they describe their system which performs more accurately than algorithms based on perceptual models by adding a heuristic function that simultaneous evaluates various interpretations of a single piece.

More recently, Dannenberg (2005) took a more intuitive approach to music analysis by combining beat location and tempo estimation with elements of pitch tracking and even genre identification to simulate the "holistic" aspect of human auditory perception and provide additional constraints to the problem of beat tracking. He reports improved results.

Goto took a path parallel to Dannenberg's. In 1995, Goto also worked on a multiple-agent architecture for beat tracking: his system evaluated multiple hypothesis simultaneously. In 2001, he used three kinds of musical analysis (onset times, chord changes, and drum patterns) to identify the hierarchical beat structure in music with or without drums.

In his MIT Media Lab PhD thesis, Jehan (2005) makes use of a perceptually

grounded approach to onset detection. He computes and sums the first order difference of each spectral band and ignores transients within a 50 ms window (because they fuse into a single event). By smoothing the resulting function, he obtains peaks that correspond to onsets. He observes that onsets also correspond to local increase in loudness. Armed with his onset detection algorithm, Jehan proceeds to his beat detection and tempo estimation phase for which he uses a causal and bottom-up approach (based on signal processing rather than a-priori knowledge).

Representation of tabla grammar

Bol Processor (BP) (1982) was a computer program that analyzed sequences of bols (in textual form) and that evolved a grammatical model that could be used to generate new improvisations (Bell and Kippen, 1992; Kippen and Bel, 1992). Used for the ethnomusicological study of North Indian tabla improvization (Kippen and Bel, 1994), BP was followed a few years later by Bol Processor 2 (BP2) (Bel, 1996). BP2 enabled MIDI (and later CSound) output along with finer control over pitch bends and bol modulations. Both BP and BP2 use a symbolic (rather than a purely numeric) representation of musical structure. Bernard Bel, the author of the Bol Processor software family, emphasizes the importance of representation, which he illustrates with a quantization scheme that, unlike quantization methods found in commercial sequencers, allows polyrhythmic expressions. Bel's another major contribution is a machine learning model that infers tabla grammars by constructing finite-state automatons from examples. Although Bel's work provides a solid framework for my project (see Section 3.5, it delves into a whole new field in the area of linguistic representation of rhythm that I did not get a chance to explore. Nevertheless I was inspired by some of the concepts elaborated by Bel, in particular the use of textual representation. Bel recently added Bol Processor 3 to his suite (2006).

Wright and Wessel (1998) created "a computer-based improvisation environment for generating rhythmic structures based on the concept of tala from North Indian classical music." Their system is particularly interesting to me because they aim to provide a "musical common ground" to musicians who collaborate with each other. The system output is (indirectly) controlled by a musician-operator (by selecting basic rhythmic building blocks, time-scaling them, and then scheduling them). The "generated material is free and unconstrained, but fits into a rhythmic structure that makes sense to the musician(s)". This last statement agrees with the stated goals of my project. It should be noted that Wright and Wessel constrain their experiment to the 16-beat tintal.

Commercial Applications

In the past ten years, Indian music has seen the widespread adoption of electronic sound generators for instruments like the *tampura* (background drone), the sruti box,

which sets the reference tone for a performance, and also the tabla. To my knowledge, two companies (Radel and Riyaz) have developed electronic tabla devices. These boxes come with a variety of presets that produce quite realistic (but static) rhythmic phrases to accompany amateur instrumental or vocal performances.

Swarshala and Taalmala provide a software environment for personal computers for learning, playing and composing tabla performances. Their method of sound generation is undocumented, but Swarshala provides several control parameters (pitch bend, etc.) for individual sound that rule out sample playback.

Chapter 3

Design

3.1 System Architecture

The TablaNet system developed during the course of my research resulted in a prototype that includes basic functionality for training, recognition, and prediction. I programmed a software simulation environment for testing the system and evaluating the hypotheses formulated in Section 1.3.

I used the following resources to build this project:

- a tabla set (from Prof. Barry Vercoe's collection),
- microphone, pre-amplified mixing console, audio components and cables,
- piezoelectric vibration sensors with various characteristics,
- audio speakers,
- my Mac PowerBook laptop for development and evaluation, and
- Xcode, the integrated development environment for Mac OS X.

The TablaNet system architecture is described in Figure 3-1. At the near-end, a pair of sensors (one for each drum) captures the strokes that are played on the tabla. The signals from both the sensors are mixed and pre-amplified, and sent to the Analog-to-Digital converter on the near-end computer. After processing the input audio signal, the computer sends symbols over the network to a far-end computer installed with the same software. The receiving computer interprets the events transmitted over the network and generates an appropriate audio output. The system is symmetrical and full duplex so that each musician can simultaneously play, and listen to the musician at the other end.

Timekeeping happens independently at each end. The system's goal is to synchronize the tabla output with the beat extracted from the incoming tabla rhythm

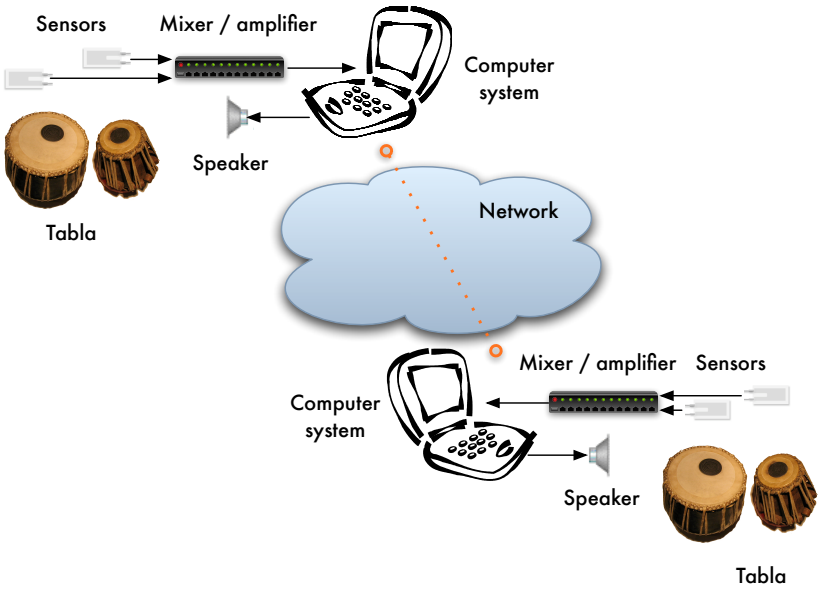


Figure 3-1: The TablaNet system diagram

rather than with the actual far-end source.

When one tabla player starts playing, the system at the other end waits until it picks up the correct rhythmic structure (usually after one or two cycles) to play back audio. The system has a configurable error detection threshold (e.g. start playing after two cycles of 16 beats and once the error rate is lower than three consecutive strokes).

The symbols are transmitted to the far-end computer over a non-guaranteed connectionless User Datagram Protocol (UDP) network layer. The UDP protocol has been shown to perform better than TCP over IP for this application. Currently the system simulates the network channel so no additional information is given on the network configuration and transmission.

3.2 Hardware Setup

The TablaNet system, although mostly software-based, relies on important pieces of hardware. To avoid feedback from the speakers (which output far-end tabla sounds) into a microphone, I use piezoelectric vibration transducers that act as contact microphones. The sensor films are pasted directly on the tabla heads with double-sided tape. The sensor outputs are fed into a pre-amplified mixer so that the resulting monophonic signal can be connected to the microphone input on the target computer. The reason for this is that many low-cost computers (e.g. the $100 laptop) may only have a monophonic input. The fact that the sounds coming from both

drums are mixed is not an issue because strokes that are played on the right drum (dayan) are distinct from those played on the left drum (bayan), and from those played on both drums simultaneously (see Table 2.1). Moreover, because the bayan produces lower pitched sounds than the dayan, the sound of each drum can be separated in the spectral domain in spite of some overlap.

Piezoelectric sensors generate an electric charge in response to mechanical stress (e.g. vibrations). Contact microphones that are made of piezoelectric material pick up vibrations through solid materials rather than airborne sound waves, and convert them into an electric signal similar to that of a microphone output.

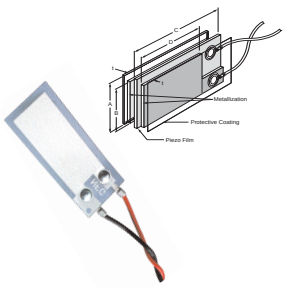


Figure 3-2: Piezoelectric film sensor element (from Measurement Specialties, Inc.)

I used piezo film sensors with lead attachments from Measurement Specialties, Inc (Figure 3-2). Their catalog proposes a variety of lengths and thicknesses. Additionally, the films can either be plain (with a thin urethan coating to prevent oxidation) or laminated (with a thicker polyester layer which develops much higher voltage than the non-laminated version when flexed but happens to be less sensitive to high frequencies probably because of higher inertia). The lead attachments provide wiring from the sensors, which is useful because high temperatures during soldering can damage the films. I experimented with the following elements: non-laminated (DT series) of length 1.63 cm, 2.86 cm, and 6.72 cm (in both 40 and 64 μm thicknesses); laminated (LDT series) of length 1.63 cm, 2.86 cm, 6.72 cm (205 μm thickness).

I also tried different double-sided tape thicknesses. The thiner films worked better but were found to be fragile (their silver ink coating would come off with the tape) and therefore had to be changed after being pasted and then removed a few times from the tabla heads. The thicker films were more robust in that way.

Each computer has an external or built-in amplified speaker to play the audio output estimated from the other end.

3.3 Software Implementation

This section describes the software design and implementation, and provides an overview of the external libraries used.

The computer program at the near-end runs the code to extract features from the audio input, and to classify tabla strokes based on those features. The application then transmits the data (symbols representing the recognized strokes and their timing) to the far-end computer over the Internet or an IP network. The receiver reassembles the packets, and generates a tabla phrase in real-time based on the events received up to that point in time. The software is written in C (GCC) with some offline processing implemented in Matlab.

The C code is developed in Apple's Xcode integrated development environment. I make use of the following free, cross-platform, and open source third-party libraries:

- PortAudio (MIT license, compatible with the GNU GPL)
- libsndfile (released under the terms of the GNU LGPL)
- FFTW, the Fastest Fourier Transform in the West, developed at MIT (GNU GPL license)

PortAudio is a multi-platform wrapper for real-time audio input/output. It provides a convenient way to access platform-specific audio devices through a callback interface.

libsndfile is an audio file wrapper that offers a read/write interface to WAV, AIFF and other types of audio files.

FFTW was used for FFT computations.

In addition, mathematical computations within my code use the math.h standard library.

Audacity is a stand-alone open-source audio recording and processing package that was used to record audio data in WAV or AIFF format and visualize its spectrum.

The software is implemented as two processes, one called TN_Tx (TablaNet Transmitter) that runs on the near-end computer, and one called TN_Rx (TablaNet Receiver) on the far-end computer.

The analysis modules, which convert the incoming audio signal into symbols, are collectively called the Transmitter. On the other side, the Receiver contains the modules that listen to the network and convert incoming symbols back into an audio signal

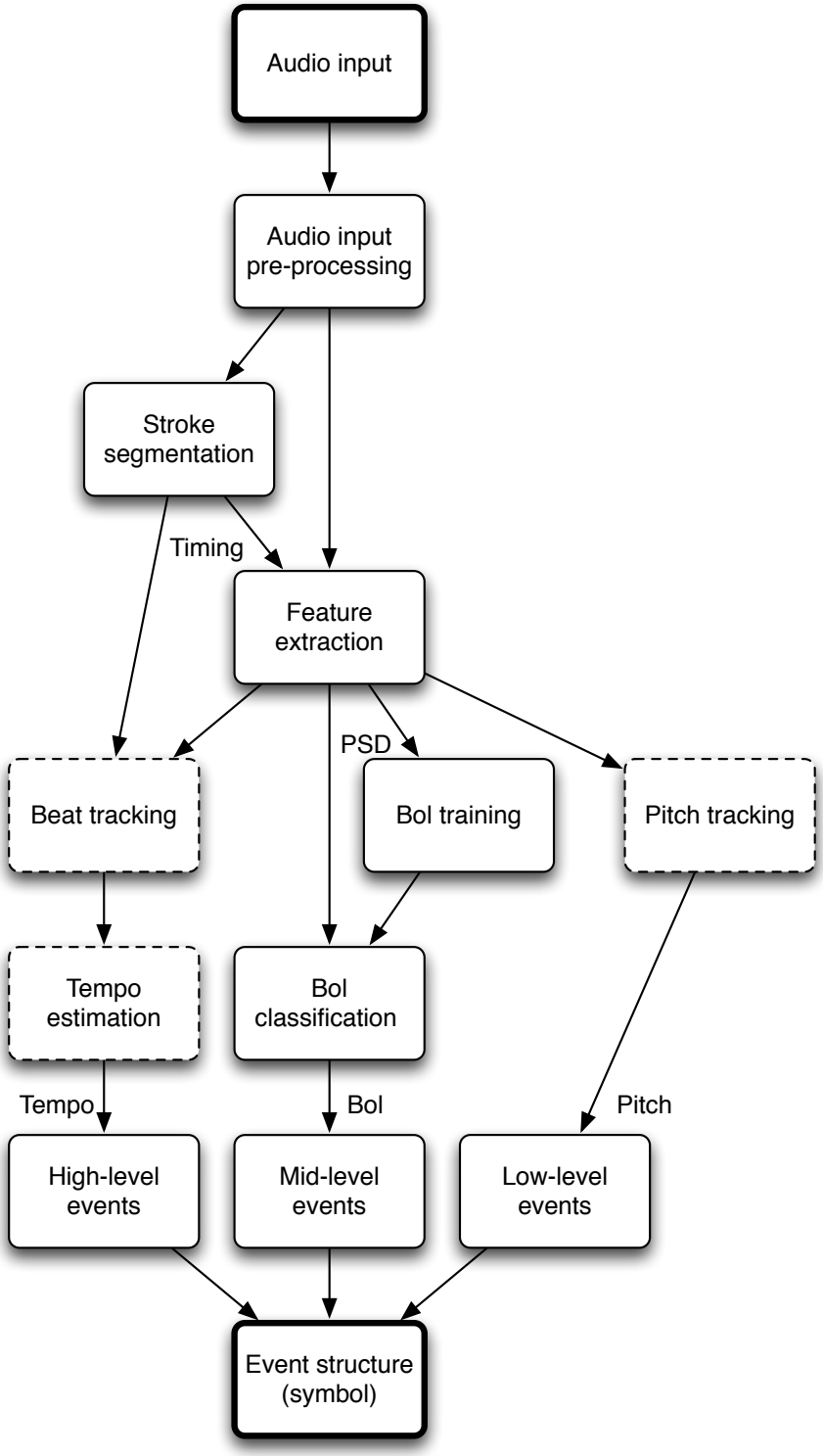


Figure 3-3: Transmitter software block diagram

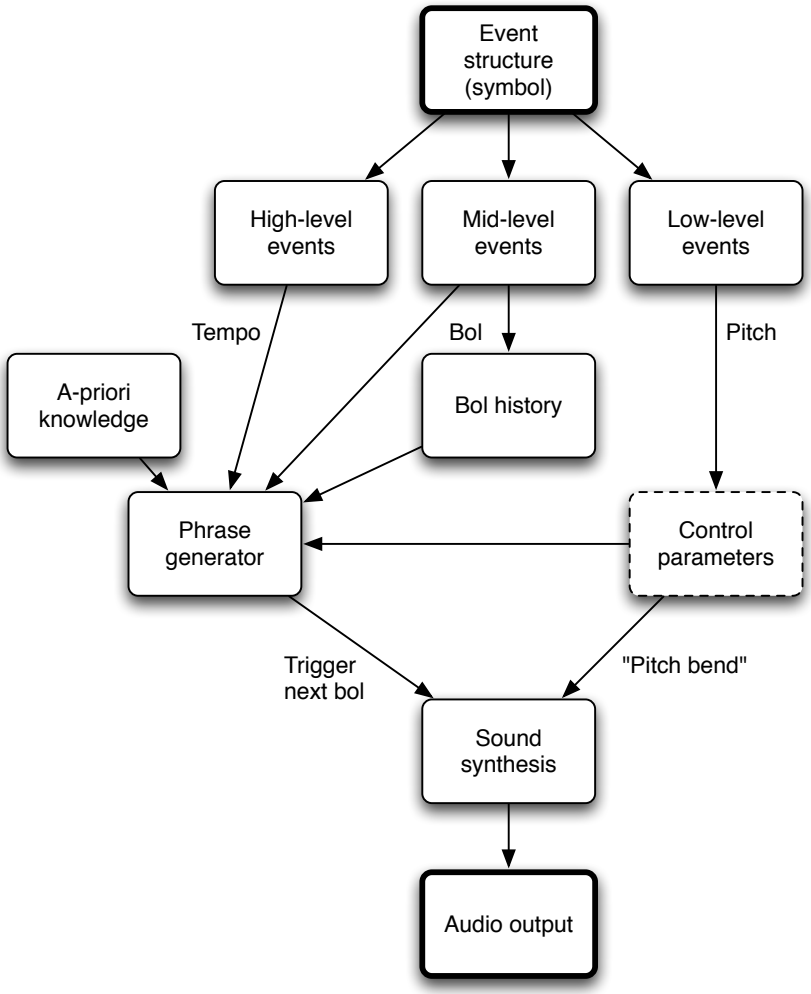


Figure 3-4: Receiver software block diagram

for playback. Figures 3-3 and 3-4 are high-level functional diagrams that represent the tasks undertaken by the Transmitter and the Receiver. Both Transmitter and Receiver are present on the near-end and the far-end computers as the system operates symmetrically in full duplex. The boxes with dashed lines are not fully implemented in the current version of the system.

The software block diagram of the Transmitter is presented in Figure 3-3. In the first step, the audio input is preprocessed (buffered, and converted from a time-domain to a frequency-domain representation). The onset detector is based on an envelope follower, and individual drum sounds are segmented into frames. Then, audio features are extracted from each frame and combined to form a feature vector of reasonable size. The features consist of spectral domain components for bol recognition, pitch tracking for general tabla tuning and occasional pitch slides on the bayan (not implemented in the current version of TablaNet), and tempo data computed from the onset timings (only timing differences are considered in this version). The bol classifier runs on each frame that contains a stroke. The identified inter-related events that make up the incoming audio stream are combined into a data structure, and sent asynchronously over the network through a callback mechanism. The Transmitter subjects the raw audio signal to perceptually-motivated digital signal processing algorithms for bol classification, tempo estimation, and pitch tracking.

The Receiver, on the other hand, operates only on symbolic data. When the event data structure reaches the Receiver through the network, the various categories of events are demultiplexed. Individual bols influence the tabla phrase generator, which estimates the most probable rhythmic pattern (i.e. the next stroke and its timing) to be played locally. This module keeps track of the previous sequences of bols with a logging mechanism. Tempo changes and pitch variations also contribute to the dynamic adaptation of the computer's generative model to the far-end musician's playing style. *A-priori* knowledge, in the form of "grammatical rules" for the tabla also constrain the phrase generator and its prediction. A hierarchical structure with a weighing scheme is used to represent the different levels of a tabla performance structure (i.e. the stroke level, the phrase level, and the composition level). Then the phrase generator triggers the appropriate sample in the sound synthesis engine at the appropriate time. Low-level events such as pitch variations control parameters of the sound synthesis engine.

The software architecture relies on a set of audio input buffers to perform various computations. The output buffers (at the Receiver side) are straightforward so I do not discuss them here. The first set of input buffers are inaccessible by the application directly. They are managed by the audio device driver. PortAudio offers an encapsulation of these low-level buffers by copying their content asynchronously to an application-level buffer through a callback function. This first application buffer is a 3-second ring buffer (with a write pointer and a multiple read pointers) that collects the audio samples provided by PortAudio whenever new data is available. After an initial delay to make sure that the ring buffer has started to fill up, a syn-

chronous polling loop checks that new audio data is available in the ring buffer, and copies a 512-sample frame into a circular array of eight 50% overlapping buffers. A 512-sample length window is applied to each buffer, and the snippet of time-domain signal contained in it is converted to the frequency domain. This set of buffers checks for the presence of an onset (see Section 3.4 for more details on the onset detection algorithm). When an onset is detected, the samples are copied from frame n (where the onset was detected) and frame $n + 2$ to another buffer of length 1024 for further processing (again see Section 3.4 for additional details from this point onwards).

Let us discuss the choice of buffer sizes and their relation with musical structures such as pitch and tempo. With a 44.1 kHz sampling rate, the 512-sample onset detection buffers with 256 sample overlap have a duration of approximately 6 ms, which is the quantization step for stroke detection. The 2048 samples contained in the onset detection buffer array correspond to around 48 ms, which is the system's smallest stroke interval. Considering that, at the fastest speed, tabla players produce one stroke every 80 to 100 ms, this buffer duration provides ample room even at high playing speeds. On the pitch detection front, the two 512-sample buffer in frequency domain allow for the detection of frequencies as low as $\frac{44100}{512} \approx 86$ Hz, which is fine for tabla sounds. It must be noted that the buffer lengths, although they provide enough data for low-latency digital signal processing algorithms, are too short for human perception (in terms of rhythmic structure or pitch resolution).

The current version of the system does not have a graphical user interface. However I experimented with the GTK+ toolbox and consider providing a GUI based on it for easier access to the training, recognition, and prediction modules, and related parameters. In particular, the GUI could provide an interface to save and load sessions and thereby retrieve historical data for a particular user.

3.4 Tabla Stroke Training and Recognition

The problem of tabla stroke training and recognition is basically one of supervised learning and pattern classification. This section defines the problem and identifies its constraints, proposes a solution, and then describes each of the components of the solution that was implemented.

Design Principles

The tabla stroke recognition engine is similar in concept to a continuous speech recognizer although it does not implement advanced features of state-of-the-art speaker-independent speech recognizers (such as dealing with co-articulation between consecutive phonemes). Furthermore, I impose external constraints that further limit the similarities between the two.

Phonemes are the basic units of speech. Speech recognizers extract features from phonemes, and then usually train and use a Hidden Markov Model (HMM) (Rabiner, 1989) which maps a sequence of phonemes to a word. Then, words are assembled into sentences, and homophones can be distinguished based on the grammatical context.

In the same way, tabla bols can also be combined to form word-like multi-syllabic bols (e.g. *TiRaKiTa* or *DhaGeDhiNa*) that usually fit within one beat. However, in the TablaNet system, I do not consider multi-syllabic bols as one compound bol, but rather treat them as a concatenation of multiple bols (with an associated timing, a fraction of a beat).

Most modern speech recognizers are continuous and speaker-independent. Although the recognizer in the TablaNet system is also continuous (it recognizes sequences of bols within rhythmic phrases, not necessarily in isolation), it is player-dependent. There are two reasons for this: the first is that for a recognizer to be user-independent, it would have to be trained on a statistically significant amount of data and I did not collect such a large dataset; the second one is that, depending on the musical school (*gharana*) the player comes from, some bol names may correspond to different strokes.

The stroke training phase corresponds to a supervised learning scheme where labels are provided. At the beginning of the session, the tabla player plays each bol three times in a continuous sequence. The player is asked to play at "his or her own speed" (a speed he or she is comfortable with) so that it resembles a real performance (e.g., with partial overlaps–the decay of a stroke merging with the attack of the next stroke). I have limited the system to 8 bols. Since the model is trained for each player, it does not matter which bols the player chooses. However the system was tested with the following bols: *Ta Tin Tun Te Ke Ge Dha Dhin*.

The part of the system design presented in this section benefited from the most attention and development time. I was able to perform preliminary studies (implementation of an offline stroke recognition in Matlab, and user listening tests) which provided some insights into the problem and informed the design principles presented here.

The tabla stroke recognition presented in Section 2.3 mentions both time domain and frequency domain features that have been used successfully. I also tested various features in my preliminary test, including zero crossings (ZCR), power spectral density (PSD), and Mel-frequency cepstrum coefficients (MFCC). The latter are widely used in speech recognition, but didn't perform well in my experiment with tabla strokes. In my early tests, PSD (512 bins reduced to 8 with principal component analysis) performed the best. Therefore, I chose to use spectral density-derived values for my feature vector. Details follow in the coming sections.

The insight which led to the previous modification is that the strokes evolve in time. The attack transients contain almost all frequencies, and then the spectrum settles down, with some strokes exhibiting their tonal property (i.e. their fundamental frequency F0 and harmonic overtones are visible in their spectrogram). Dividing the time window into smaller frames helps take this time-varying information into account.

As far as the recognition method is concerned, the literature describes the following approaches for tabla stroke recognition:

- hidden markov models (HMM),
- neural networks,
- decision trees,
- multivariate Gaussian (mvGauss) model,
- k-nearest neighbor (kNN),
- kernel density (KD) estimation,
- canonical discriminant analysis (which is really a dimensionality reduction method),
- support vector machines (SVM),
- expectation-maximization (EM), and others.

This list actually covers most of the machine learning algorithms!

In my preliminary study, I compared the performance of three pattern classification techniques: k-nearest neighbor, naïve Bayes, and neural nets. In the current

system, I chose to implement the k-nearest neighbor (kNN) algorithm not only because it had performed the best in my study, but also because it displayed results close to human perception. In fact, the confusion matrix (refer to Section 4.1) showed that strokes that are difficult to distinguish for humans (actually more so for beginner tabla players than for expert musicians) posed similar difficulties for a kNN-based machine recognizer.

Whereas naïve Bayes builds a model and computes parameters to represent each class (each stroke), kNN is an example of instance-based learning where the feature vectors extracted from the training dataset are stored directly. Although it has drawbacks such as its memory footprint and computational complexity, the simplicity of my requirements (eight bols, three eight-dimension feature vectors per stroke) make this choice well suited for my application.

Algorithm Implementation

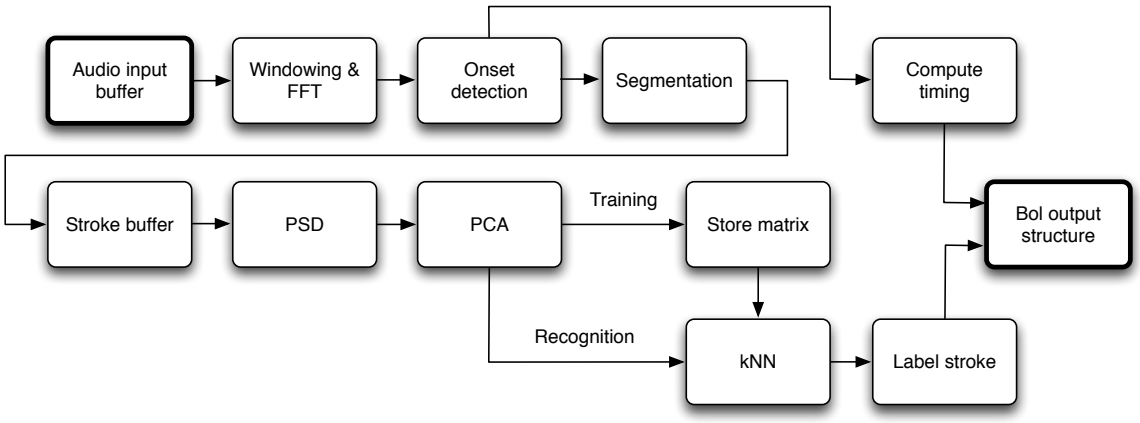


Figure 3-5: Tabla stroke training and recognition block diagram

Figure 3-5 shows the data and control flow in the tabla stroke training and recognition module. The audio input buffer provides audio samples either from the real-time audio input driver or from an audio file. A Hamming windows is applied to the frame to limit frequency leakage from the Fast Fourier Transform (FFT) that follows. The choice of a Hamming window compared to other windowing functions is arbitrary. Then an FFT is performed on the time-domain frame, followed by an onset detection algorithm which is applied to consecutive frequency-domain frames. On detection of a stroke onset, the stroke's timing information is captured on the one hand, and, on the other, a segmentation process ensues, where two non-overlapping frames of a stroke are stored into a buffer (in frequency domain). Each frame's spectral density is computed, and the resulting concatenated vector is the stroke's feature vector. Principal component analysis (PCA) is applied to the feature vector in order to make its

dimension tractable considering the limited amount of training data. In the training phase, the reduced feature vectors are stored in matrix form and stored in a file for future reference. In the recognition phase, the matrix previously stored during the training phase is retrieved and used in the kNN algorithm. The algorithm outputs the label of the recognized stroke. Then, both the timing information from the stroke onset, and the stroke label are placed into a structure for transmission.

The initial symbol structure contains the bol label and timing (number of frames since last stroke). Additional data could later include the stroke amplitude or power information (to extract accent information for instance), and its (time-varying) pitch contour.

Stroke Detection

I present here details of the windowing function, the fast Fourier transform (FFT) implementation, the onset detection, the stroke segmentation, and the stroke timing computation.

The initial buffer frame size is 512 samples (see Section 3.3 for a discussion on buffer sizes). A Hamming window (Equation 3.1) is applied to the samples in the frame.

$$w[n] = 0.54 - 0.46 \cos \frac{2\pi n}{N-1} \tag{3.1}$$

where w is the function output, n, the sample number, and N, the frame size.

The windowed audio input frame is computed by Equation (3.2).

$$x[n] = y[n]\, w[n] \tag{3.2}$$

where x is the window output, and y is the input frame.

The reason for applying a window to the input frame is to prevent leakage during the FFT computation on a finite-length sequence. For more information on leakage and windowing, refer to Oppenheim and Schafer (1975).

The window is applied to half-overlapped buffer frames so as to avoid missing a transient attack close to the edges of a frame and therefore ignored because multiplied by a small value.

Once the window is applied to each frame, an FFT is performed on the frame. The fast Fourier transform is an efficient implementation of the discrete Fourier transform (DFT). The DFT is a discrete time-domain to discrete frequency-domain Fourier analysis transform applied to signals of finite length. In lay terms, the DFT expresses

a time-domain signal as a sum of sinusoids of varying amplitude and phase (in the frequency domain). In mathematical terms, the DFT equation appears in (3.3).

$$X[k] = \frac{1}{N} \sum_{n=0}^{N-1} x[n]\, e^{\frac{-2\pi j}{N} kn} \qquad k = 0, \ldots, N-1 \tag{3.3}$$

where X is the frequency-domain transform of the time-domain input signal x.

The DFT satisfies our requirements for feature extraction because the data that it provides lets us analyze the contribution of each frequency band in the tabla stroke input signal. Since the input signal $x[n]$ is real-valued, its output satisfies Equation (3.4).

$$X[k] = X^*[N-k] \tag{3.4}$$

where X^* denotes the complex conjugate of X; and the indices are module N.

In particular this means that the DC component (at $k = 0$) and the Nyquist component (at $k = \frac{N}{2}$) are real-valued, and only half of the other elements are required to be evaluated and stored (the other half can be recomputed from them).

The FFT is computed with the FFTW software package (see Section 3.3). Its output is non-normalized, meaning that it is not scaled by $\frac{1}{N}$ as in Equation (3.3). This does not matter because we primarily use the FFT result for comparison purpose between audio frames or feature vectors.

The onset detection problem found in the TablaNet system is a simplified version of the generalized onset detection problem, which is still an active research problem. I take advantage of the following assumptions: the audio input only contains tabla strokes (especially when using vibration sensors which do not pick-up ambient sounds), and spurious transients (due to electrical noise, or fingers touching the sensors for example) are of short duration compared to tabla strokes. Therefore I am able to simply check for an increase in energy between the current frame and the previous one, and a slight drop at the next frame (when the stroke reaches its steady state). I do not need to use more advanced perceptually-motivated techniques here.

A preliminary study of onset detection, where I had used a simpler technique of detecting an increase in time-domain amplitude, performed very poorly, with either a high amount of false positives or false negatives, depending on the threshold value.

The total energy of the frame is computed in Equation (3.5).

$$E_{frame_i} = \sum_{k=0}^{N-1} |X^2[k]| \tag{3.5}$$

where E_{frame_i} is the energy of frame i.

Then the system compares the energy of the current frame E_{frame_i} with the energy of the previous frame $E_{frame_{i-1}}$. If $E_{frame_i} \geq 3\,E_{frame_{i-1}}$, we verify that the next frame sees a decrease in energy: $E_{frame_{i+1}} \geq E_{frame_i}$. If both conditions are satisfied, the system reports an onset. The factor 3 in the first condition was chosen based on empirical observations. This algorithm performed with close to 100% recognition accuracy (based on ad-hoc testing).

Once an onset has been detected, the current frame and the next non-overlapping frame are stored in a frequency domain stroke buffer for further processing (i.e. stroke segmentation).

At this point, the timing of the current stroke (or rather its onset) is computed by counting the number of frames since the last onset. Based on a 256-sample overlap between frames (at 44.1 kHz sample rate), the stroke quantization has a resolution of slightly less than 6 ms, which, according to my experiments, appears to be sufficient for the musicians who have used the system (see Section 4.2 and user comments in Appendix A). The relative timing (the "delta") of each stroke is computed.

Feature Extraction

Feature extraction is performed on frequency-domain frames of length 1024. In my preliminary study, I treated each stroke as a stationary random process (i.e. I did not distinguish between the attack phase, the steady state, and the decay). I computed the PSD using the periodogram method. Since I now treat the signal as time-varying (the first frequency-domain analysis frame corresponds to the attack, and the second frame corresponds to the steady state), I am not able to use the PSD formula (because it is non-stationary). Therefore I treat the sequence as periodic, compute its DFT to make a discrete spectrum, and then evaluate its spectral density (Equation 3.6).

In this case, the windows don't overlap: the middle portion (transition between noisy attack and steady state) is not taken into account. Moreover, frames are used for comparison purpose, not time-domain reconstruction.

$$\Phi_{frame_i}[w] = \left| \frac{1}{\sqrt{2\pi}} \sum_{-\infty}^{\infty} x[n] e^{-jwn} \right|^2 = \frac{X[w]\,X^*[w]}{2\pi} \tag{3.6}$$

where Φ_{frame_i} is the energy of the current frame i, and w is the radial frequency.

In practice, however, I do not scale the result by 2π, and use frequency bin indices rather than radial frequency indices (as in Equation 3.7).

$$\Phi_{frame}[n] = X[n]\,X^*[n] \tag{3.7}$$

Φ_{frame_i} is computed for each of the two frames extracted from each stroke, and then concatenated, resulting in a 1024-length feature vector.

Dimensionality Reduction

The feature vector obtained in the previous section presents a problem for efficient classification. Its length, 1024 elements, is too large in comparison with the dataset (8 bols with 3 training examples each). This is a problem with reference to the "curse of dimensionality": the data density is too low within the high-dimensional feature space (Bellman, 2003).

A solution to this problem is to reduce the number of dimensions (and thus elements) in the feature vector. I do so by selecting an orthogonal linear combination of the "most relevant" dimensions in the feature space (relevant in a representation sense–dimensions with the largest variance–not necessarily in a classification sense). The feature vectors are then projected onto a new coordinate system. Used because of its relatively simple implementation (compared with dimensionality reduction techniques optimal for classification), PCA works well enough for classification purposes in this case according to my preliminary studies. PCA involves an eigenvalue decomposition. It does not have a fixed set of basis vectors: its basis vectors depend on the data set.

Figure 3-6 shows the PCA algorithm. Additional information and mathematical derivations can be found in Duda et al. (2000).

In the current system, the algorithm's first five steps (one-time computations for each training set) are performed offline in Matlab and saved to a file, which is then used by the C program for real-time dimensionality reduction in the test data.

First, the training data is organized into an $M \times N$ matrix where the M training vectors are arranged as column vectors and the N rows correspond to the observed variables. In this case, I have $M = 3 \times 8 = 24$ training data vectors. In the first algorithmic step, the empirical mean vector is computed (each vector element corresponds to the mean in a particular dimension). Then the mean vector is subtracted to each column of the observation matrix so that the training data is centered around the origin of the feature space. In the second step, the covariance matrix is computed from the centered observation matrix. At this point, PCA involves an eigenvalue decomposition. I use Matlab's numerical functions to compute the M eigenvectors and eigenvalues of the covariance matrix. Eigenvectors and eigenvalues are stored in their respective matrix and ordered according to decreasing eigenvalues while keeping the eigenvector–eigenvalue correspondence (same column number). Finally, a set of L basis vectors is chosen, starting from the eigenvectors with the largest associated eigenvalue (dimensions with the largest variance). The optimal value for L is discussed in the evaluation section (4.1). This concludes the offline selection of basis

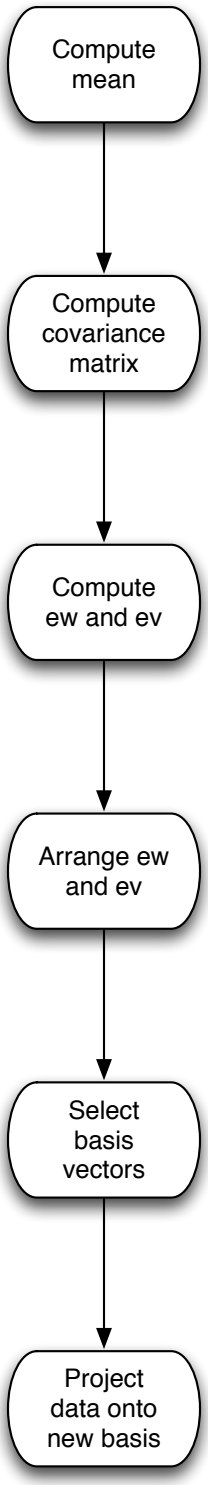


Figure 3-6: Principal Component Analysis algorithm

vectors dependent on the data set. The last step, the projection of the data onto the new basis vectors, is performed in real-time after the stroke segmentation phase, whenever an onset is detected.

k-Nearest Neighbor

The k-nearest neighbor algorithm (kNN) classifies new strokes based on the distance of their feature vector with the stored and labeled feature vectors of the training dataset. kNN is an example of "lazy learning" where no model of the data is pre-established—specific instances of the data are compared—and computation happens at classification time. kNN is used here because of its simplicity and intuitive approach to classification, and in spite of its run-time computation complexity.

The k parameter indicates the number of neighbors the test data point is compared with. A majority vote will select the output label among the k nearest neighbors. I tried out different values for k (see Section 4.1). The distance measure used here is the Euclidian distance (Equation 3.8).

$$d(a_i, b_i) = \sqrt{\sum_{i=1}^{n} (a_i - b_i)^2} \tag{3.8}$$

where a and b are the points whose distance d is evaluated in n dimensions, and i is the dimension index.

In the software implementation however, since we use the Euclidian distance for comparison purposes, the square root is omitted. And given the small set of training data, the algorithm's computational complexity does not come in the way of real-time behavior.

3.5 Tabla Phrase Prediction and Synthesis

As opposed to the previous tabla recognition problem, the tabla phrase prediction problem is one of unsupervised learning. In this section, I present the issue along with background information, propose a solution, and give an example that illustrates its implementation.

Unsupervised learning here means that there is no explicit teacher to label each class or category. The system tends to dynamically form groups of the input patterns during the system's lifetime (or during a rehearsal session). To solve the prediction problem we look at the literature on estimation. We are primarily concerned about an appropriate representation for the historical (input) data, and a suitable model for the a-priori knowledge and the algorithm.

Design Principles

In this section I mainly discuss the tabla phrase prediction engine. I first look at the literature on rhythm perception and musical expectation. Then I briefly look at technical solutions to the prediction problem. Finally I use the insights offered by some of the findings to develop a model that will respond to the specificities of the tabla.

Jackendoff and Lerdahl in their 1996 landmark "Generative Theory of Tonal Music" (GTTM) for Western classical music discuss the importance of grouping (i.e. music segmentation) and meter (i.e. the alternation of strong and weak beats) in the perception of musical structure. They acknowledge the fact that meter is culture-specific (for instance syncopation is often avoided in Western music), explaining that small integer ratios are easier to process than complex ones. However as we saw in Section 2.2, Indian music makes liberal use of syncopation. They also talk about the importance of expressive timing (i.e. timing that is slightly "off") in the value of interpretation.

Narmour (1999) talks about hierarchical expectation. He uses the idea of repetition to explain the notion of style in a way that shapes cognition, and enables us to recognize elements of style, by knowing what to expect within a style that we are familiar with.

Clarke (1999) proposes a survey of research in rhythm and timing in music. In particular he mentions the substantial work of Paul Fraisse in this area. Fraisse makes a distinction between temporal phenomena under 5 seconds, which contribute to the perception of time, and longer events, which lead us to reconstruct temporal estimates from the information stored in memory, and which also contribute to our sense of expectation. In fact, he goes on further about our perception of time, and explains that events that occur approximately within 400 to 600 ms of each other (relative intervals between events) lead to our sense of grouping (although, according to Povel and Okkerman, other characteristics like amplitude and pitch may contribute to the phenomenon of auditory grouping). On the other hand, long durations (above 600 ms) make us aware of the "passage of time." Finally, Fraisse also ties rhythm perception with motor functioning (the fact that we literaly "tap the beat").

Snyder (2000) also categorizes events of different durations: under $\frac{1}{32}$ s, we have event fusion, between $\frac{1}{16}$ s and 8 s, we experience melodic and rhythmic grouping, and above 8 s, we perceive the form. Further, he defines some common terms like "beat" (a single event in time, an onset, or equally spaced temporal units), "pulse" (a series of beats that occur at the tempo), and "accent" (the "weight" or quality of each beat—strong or weak). Snyder emphasizes the importance of metrical hierarchy and the smaller subdivisions of beats within the tempo. He also indicates that the meter can be seen as a temporal schema that requires "closure": we need to know where the downbeat is (from context or from means other than temporal). In fact, my user

experiments (see Section 4.2) showed that the tactus is particularly difficult to catch in Indian music—especially when the context (e.g. the tala, the first downbeat) is not clear.

These studies address music perception and cognition, some based on studies of the brain, but the majority are based on Western classical music. It is therefore noteworthy that there are actually a few relatively recent studies specific to Indian music (see Clayton (2000) and Berg (2004)).

Most of the studies described here emphasize the importance of hierarchical structure in the perception of rhythm, and the fact that intervals between events are of utmost important to distinguish individual events (fusion), from grouping (e.g. rhythmic phrases), and form (or composition-level structure).

On the technical side, Russell and Norvig (1995) present various approaches to learning in the context of artificial intelligence. In particular they mention the following three key elements:

- the components to be learned,
- the feedback available to learn those components, and
- the representation used for components.

Russell and Norvig describe unsupervised learning as a type of learning which has no a-priori output. To be effective, an unsupervised learning scheme is required to find "meaningful regularity in the data." They also advocate an approach that combines prior knowledge about the world with newly acquired (or current) knowledge.

In practical systems, the problem of prediction has often been solved with Kalman filters, which are like hidden Markov models (HMM) except that the hidden state variables are continuous instead of being discrete. HMMs are statistical models that are considered to be the simplest dynamic Bayesian network (DBN). They are widely used in speech recognition and other time-varying pattern recognition problems. During the training phase, hidden (model) parameters are estimated from observable parameters (e.g. the sequence of phonemes that make up a certain word). The model parameters are then used for analysis during the pattern recognition phase (see Rabiner (1989) for a speech recognition application example).

The Kalman filter is a recursive filter that estimates the state of a dynamic system from partial information. The next state (in discrete time) is predicted from the previous state and control parameters. The filter generates the visible output from the hidden state.

Although Kalman filtering and hidden Markov models are powerful tools, I chose another mechanism for the TablaNet prediction engine. For one, Kalman filters operate on continuous hidden state variables, whereas the sequence of tabla strokes that produce rhythmic phrases are discrete events. As for HMMs, the model is trained on a defined set of sequences (or words, or phrases). Moreover integrating new instances within the model is time consuming (for real-time behavior).

Although the prediction problem seemed particularly hard to address at first, I was inspired by the relative simplicity and the promising results of Chafe's statistical model for the prediction of solo piano performance (1997). This gave me hope.

I would like to remind the reader here of the model developed by Bel in his Bol Processor (see Section 2.3). Based on a textual representation of bol sequences along with grammar rules for improvisation, Bel's approach inspired me to develop an alternative model for phrase prediction.

My solution resides in dynamically building a generative grammar. The sequence of recognized bols (labels) is sent from the Transmitter to the Receiver as ASCII characters (the eight bols are represented by characters a to h). These characters are contained within a structure which also contains the relative timing of the current stroke in relation to the previous stroke. The Receiver stores the string sequence as an historical account of the performance and, if applicable, generates a new grammar rule (or updates an existing one) to predict the most likely next stroke the next time around. A string matching algorithm runs for various string lengths to account for the hierarchical structure of rhythmic patterns. Additionally, timing and bol prediction are constrained by the output of a decision tree that contains a-priori knowledge about tabla performance. The decision tree is not currently implemented. Although simple, this approach offers a convenient and simple way to evaluate my hypotheses about the TablaNet system.

I tried in this preliminary design to keep the constraints to a minimum. For instance, there is no higher-level beat tracking or tempo estimation. One reason for this is that these problems are even more complicated for Indian music than they are for Western music because of the pervasive use of syncopation, among other things. Therefore the system trains itself with the previous events in an ad-hoc fashion.

If this previous description is somewhat abstract, the following section describes the algorithm and is followed by an example that illustrates it.

Algorithm Implementation

The tabla phrase prediction algorithm is a non-metric method that makes use of a simplified formal grammar with production rules and grammatical inference. For more information on non-metric methods, please refer to Duda et al. (2000).

This algorithm uses an unsupervised learning method with no reinforcement. There is no feedback loop between the output playback and the incoming symbol stream. Output errors do not affect the algorithm. Instead, it learns from the incoming symbolic data so that it can perform better the next time.

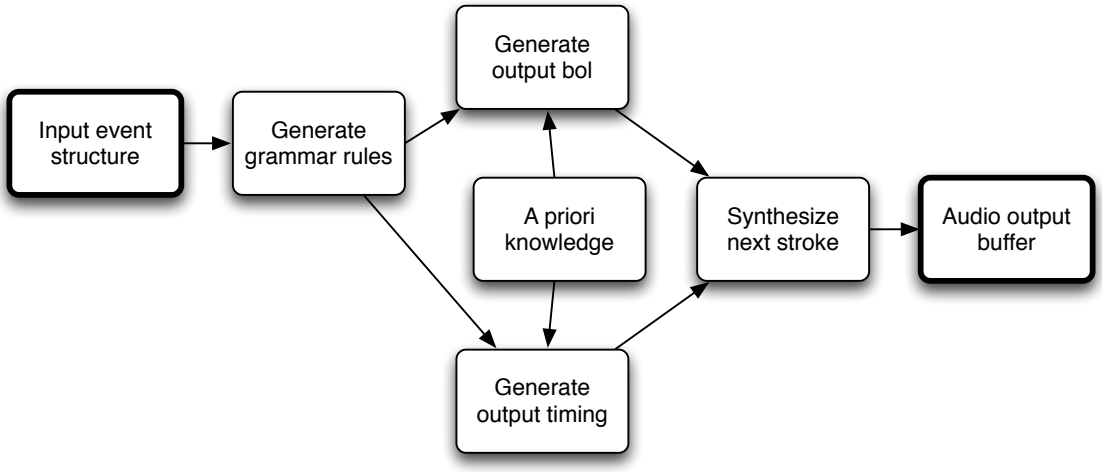


Figure 3-7: Tabla phrase prediction and synthesis block diagram

The input event structure contains the bol name and its relative timing (delta). The algorithm first generates a new grammar rule based on the incoming bol. Production rules have a minimum of two characters on the left. In case of conflict, a longer sequence is searched in the logged incoming stream, and two new longer production rules are generated instead of the previous one. Both the bol label and its relative timing are stored on the right of the production rule. Following this, the algorithm searches for the largest rule containing the incoming stream, and outputs the corresponding bol with its associated timing. We note that timing and bols are highly interdependent: as mentioned earlier, some compound bols have short interval times that are a fraction of the beat.

In case of conflict in the output bol timing, two new larger production rules are generated, in the same way as described previously. To make the process of expanding production rules, a linked list containing pointers to the logged input string sequence is stored along with each production rule.

A problem to be aware of is that recognition errors might introduce errors in the production rules. I deal with this by computing the edit distance between the input string and the string templates in the production rules. The equivalence threshold is fixed at 10% of the string length (based on the recognition error rate).

As a proof-of concept, the algorithm is currently implemented with table lookup.

The algorithm's performance could be improved by using hash tables.

One disadvantage of the system presented here is the fact that it always produces an error the first time a new rhythmic pattern is encountered. However, this could be corrected by implementing a decision tree that contains hard-coded rules specific to North Indian drumming. The rules could be used to set constraints for the output to "make sense in tabla music". Another reason to use a decision tree could be to infer heuristics about the context (e.g. which tala the rhythm is in) to further constrain the output.

Example

This section presents an example (based on Tintal) for the generative grammar. The X represents a random or unknown stroke.

Table 3.1: Example of the tabla phrase prediction algorithm

Step		**Sequence of strokes**
0.	Rules:	`0`
1.	Input:	`a`
	Output:	`X X`
	Rules:	`0`
2.	Input:	`a b`
	Output:	`X X X`
	Rules:	`0`
3.	Input:	`a b b`
	Output:	`X X X X`
	Rules:	`a b -> b`
4.	Input:	`a b b a`
	Output:	`X X X X X`
	Rules:	`a b -> b`
		`b b -> a`
5.	Input:	`a b b a a`
	Output:	`X X X X X X`
	Rules:	`a b -> b`
		`b b -> a`
		`b a -> a`
6.	Input:	`a b b a a b`
	Output:	`X X X X X X b`

Continued on next page

Step		Sequence of strokes
	Rules:	`a b -> b`
		`b b -> a`
		`b a -> a`
		`a a -> b`
7.	Input:	`a b b a a b b`
	Output:	`X X X X X b a`
	Rules:	`a b -> b`
		`b b -> a`
		`b a -> a`
		`a a -> b`
8.	Input:	`a b b a a b b a`
	Output:	`X X X X X b a a`
	Rules:	`a b -> b`
		`b b -> a`
		`b a -> a`
		`a a -> b`
9.	Input:	`a b b a a b b a a`
	Output:	`X X X X X b a a b`
	Rules:	`a b -> b`
		`b b -> a`
		`b a -> a`
		`a a -> b`
10.	Input:	`a b b a a b b a a c`
	Output:	`X X X X X b a a b X`
	Rules:	`a b -> b`
		`b b -> a`
		`b a -> a`
		`a a -> b | c (remove)`
		`b a a -> b | c (remove)`
		`b b a a -> b | c (remove)`
		`a b b a a -> b | c (remove)`
		`X a b b a a -> b`
		`a a b b a a -> c`
11.	Input:	`a b b a a b b a a c c`
	Output:	`X X X X X b a a b X X`
	Rules:	`a b -> b`
		`b b -> a`
		`b a -> a`
		`a c -> c`
		`X a b b a a -> b`
		`a a b b a a -> c`
12.	Input:	`a b b a a b b a a c c d`

Continued on next page

Step		Sequence of strokes
	Output:	X X X X X b a a b X X X
	Rules:	a b -> b
		b b -> a
		b a -> a
		a c -> c
		c c -> d
		X a b b a a -> b
		a a b b a a -> c
13.	Input:	a b b a a b b a a c c d d
	Output:	X X X X X X b a a b X X X X
	Rules:	a b -> b
		b b -> a
		b a -> a
		a c -> c
		c c -> d
		c d -> d
		X a b b a a -> b
		a a b b a a -> c
14.	Input:	a b b a a b b a a c c d d b
	Output:	X X X X X X b a a b X X X X X
	Rules:	a b -> b
		b b -> a
		b a -> a
		a c -> c
		c c -> d
		c d -> d
		d d -> b
		X a b b a a -> b
		a a b b a a -> c
15.	Input:	a b b a a b b a a c c d d b b
	Output:	X X X X X X b a a b X X X X X a
	Rules:	a b -> b
		b b -> a
		b a -> a
		a c -> c
		c c -> d
		c d -> d
		d d -> b
		X a b b a a -> b
		a a b b a a -> c
16.	Input:	a b b a a b b a a c c d d b b a
	Output:	X X X X X X b a a b X X X X X a a

Continued on next page

Step		Sequence of strokes
	Rules:	a b -> b
		b b -> a
		b a -> a
		a c -> c
		c c -> d
		c d -> d
		d d -> b
		X a b b a a -> b
		a a b b a a -> c
17.	Input:	a b b a a b b a a c c d d b b a a
	Output:	X X X X X X b a a b X X X X X a a X
	Rules:	a b -> b
		b b -> a
		b a -> a
		a c -> c
		c c -> d
		c d -> d
		d d -> b
		X a b b a a -> b
		a a b b a a -> c
		d b b a -> a
18.	Input:	a b b a a b b a a c c d d b b a a b
	Output:	X X X X X X b a a b X X X X X a a X b
	Rules:	a b -> b
		b b -> a
		b a -> a
		a c -> c
		c c -> d
		c d -> d
		d d -> b
		X a b b a a -> b
		a a b b a a -> c
		d b b a -> a
19.	Input:	a b b a a b b a a c c d d b b a a b b
	Output:	X X X X X X b a a b X X X X X a a X b a
	Rules:	a b -> b
		b b -> a
		b a -> a
		a c -> c
		c c -> d
		c d -> d
		d d -> b

Continued on next page

Step	Sequence of strokes

	`X a b b a a -> b`
	`a a b b a a -> c`
	`d b b a -> a`

This example shows the output of the table phrase prediction engine for a Tintal (16-beat) input. Slightly after the beginning of the second cycle (beat 18), the model has a sufficient set of production rules to predict accurately the next set of strokes. If the input changes, the algorithm will diverge for a while until it builds a new set of rules or updates the existing ones.

Tabla Sound Synthesis

Once the output bol and its timing are estimated, a stored buffer of sampled sound is copied onto a circular output buffer for playback.

Chapter 4

Evaluation

The system has been evaluated on the following criteria:

- tabla strokes recognition rate, and comparison with existing systems,
- tabla phrase prediction rate, and
- output realism and audio quality by performers and listeners based on a statistical perceptual assessment.

4.1 Quantitative Analysis

This section deals with objective error rates by comparing the actual algorithm outcome with the expected outcome for the recognition task, and then for the prediction task.

Tabla Stroke Recognition

This section computes the tabla stroke recognition rates for various algorithm parameters and system parameters. With the optimal configuration and a selection of 8 bols, the TablaNet recognition algorithm reaches a discrete player-dependent recognition rate above 95%, and a continuous recognition rate above 87%. These results are lower but comparable with perceptual recognition rates for an intermediate tabla player.

The tabla set that I used was tuned at C4 (260 Hz) for the dayan (right drum), slightly above typical tuning values (generally around G3), and between $A^{\#}2$ and B2 (120 Hz) for the bayan (left drum).

Much like speech recognition evaluation, tabla stroke recognition rates are computed by counting the number of insertions (spurious strokes or false positives), deletions (undetected strokes or false negatives), and substitutions (erroneously recognized strokes). The stroke recognition algorithm runs every time an onset is detected in the input signal. Since the onset detector performs close to 100% (tabla strokes

are relatively simple to detect—at least at intermediate paying speeds—because they have a definite attack), the recognition algorithm runs every time a stroke occurs. Therefore insertions and deletions are almost null. Hence we simply count the number of substitutions, and compare it with the total number of strokes presented to the algorithm.

In my preliminary study, where the features and the classification algorithm were selected, I performed the following procedure on discrete tabla strokes: training with leave-one-out validation (leave one stroke out of the training set, train the system on the remaining strokes, and test the recognition of the left out stroke; perform this procedure with each stroke in the training set, and then average the recognition rate), and then testing on a completely different set of data. In this training, since most of the algorithm structure was derived from the preliminary study and careful observations of the raw input data (in time domain and frequency domain), I performed the training phase (no validation) followed by the testing phase (for both discrete strokes and continuous strokes).

The training set is composed of a labeled sequence of 3 consecutive occurrences of each of 8 bols (Ta Tin Tun Te Ke Ge Dha Dhin). There are 24 data points in the training set. The discrete testing set, with two such sequences played by the same player at a different time, contains twice as much data (48 data points). The continuous testing set contains a tabla phrase (Tintal) played twice. This phrase does not contain all the trained bols (Dha Dhin Ta Tin)—half of them actually—but gives a good idea of how the system performs in a real-world situation.

The first set of testing was performed for parameter tuning. Figure 4-1 plots the evidence curve for various k values (the k of the kNN algorithm) with other parameter values fixed to 1024 for the frame size and 8 for the number of feature vector dimensions.

Since $k = 3$ performed the best, I used that value in the final algorithm. I also used this method to look for the optimal FFT size N (Figure 4-2). This figure shows that the algorithm performs better for larger values of N (reduced to 8 dimensions). However, this also means that the frame size has to be increased to accommodate increased FFT frequency bins. This comes at the expense of shorter time domain frames which allow for a faster succession of strokes. Also, the number of frames is kept constant at two, to account for the attack which contains almost white noise, and the steady state which, in some cases, have a clear tonal or resonant quality.

In the final implementation of the algorithm, I chose $N = 512$ which seems to be a good trade-off between the length of the time slices and the recognition rate.

Finally, I plotted the evidence curve for the number of dimensions (at the output of PCA) to find the optimal number of dimensions for the PCA algorithm (see Figure 4-3). The curve shows that 8 is the optimal value, which matches with my preliminary

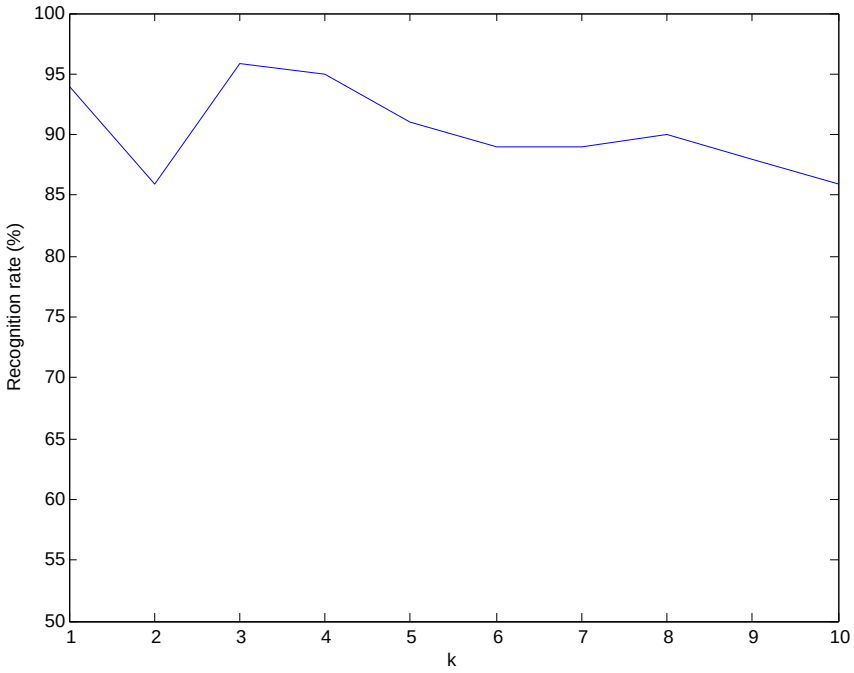


Figure 4-1: Evidence curve (discrete strokes) for varying k

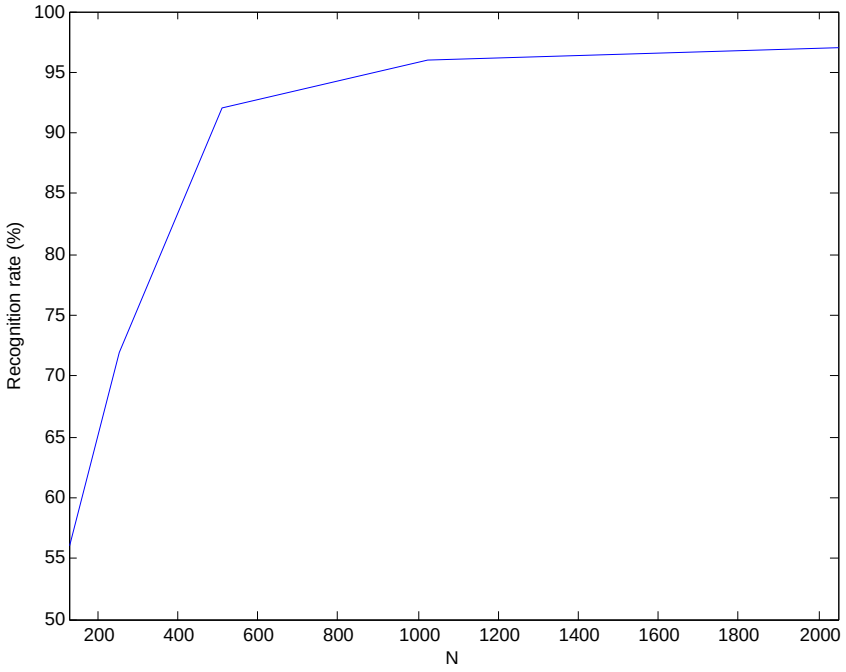


Figure 4-2: Evidence curve (discrete strokes) for varying N

findings.

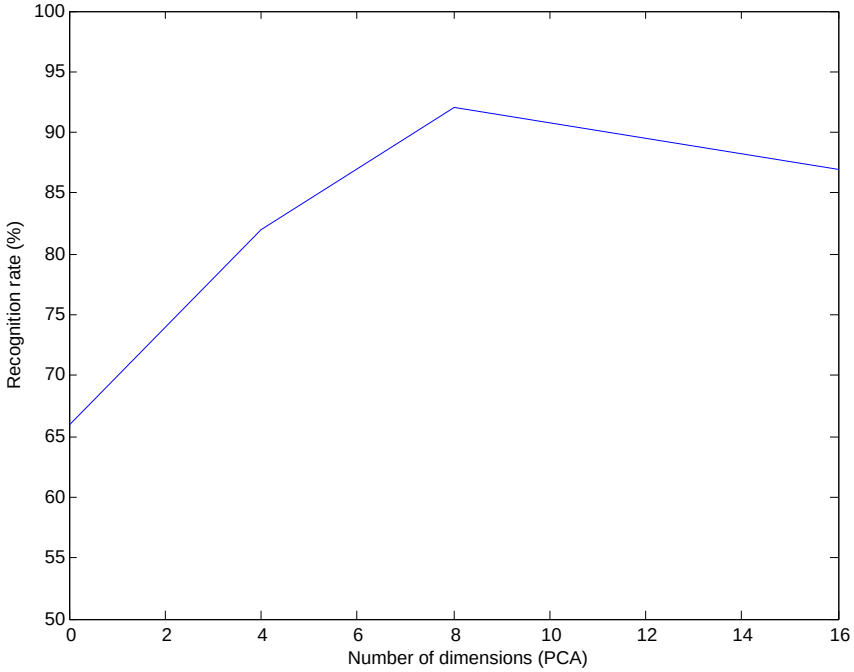


Figure 4-3: Evidence curve (discrete strokes) for varying number of dimensions (PCA)

The set of parameters that were selected for the algorithm for further user tests were the following: $k = 3$, $N = 512$, and number of dimensions (PCA) = 8. The stroke recognition algorithm with these parameters was applied to a subset of the data recorded from four users (among the eight who participated in the study). The four sets of user data were selected manually for their characteristics (in particular their clean and consistent strokes). The plot presented above represent the averaged values over these four data sets (each data set contains one training set, and two testing sets of the same sequence of strokes). The best results using the parameters mentioned above was 92% (the peak at 95.5% was observed with $N = 1024$).

The previous tests were performed using audio data captured with a microphone. Since the later user tests were performed with piezoelectric sensors, I wanted to evaluate the impact of using contact microphones on the recognition rate. The same set of four users mentioned previously were asked to play the same sequence of strokes on a tabla set fitted with the sensors. The data was recorded as a WAV file in the same way as had been done with the microphone input. The recognition rate with the vibration sensors (non-laminated) was 90.6%, slightly lower than the microphone input, but nevertheless comparable. An evaluation with laminated sensors was not performed with the same set of users; instead, I ran the test on data that I collected with me playing. The recognition rate (89.3%), although not obtained in the same way as in the previous controlled study, gives a rough performance comparison between the two types of sensors.

Table 4.1: Recognition rate for tabla players of various levels

Musicianship level	Beginner	intermediate	Advanced
Recognition rate (%)	87.5	90.6	91.7

Table 4.2: Confusion matrix for automatic stroke recognition

	Ta	***Tin***	***Tun***	***Te***	***Ke***	***Ge***	***Dha***	***Dhin***
Ta	**83**	0	17	0	0	0	0	0
Tin	0	**50**	33	0	0	0	0	17
Tun	0	17	**83**	0	0	0	0	0
Te	0	0	0	**67**	33	0	0	0
Ke	0	17	0	17	**66**	0	0	0
Ge	0	0	0	0	0	**87**	13	0
Dha	0	0	0	0	0	17	**50**	33
Dhin	0	0	0	0	0	17	17	**66**

Finally, I tested the discrete recognition rate between three sets of users: beginners, intermediate, and advanced. I classified each user in one category after evaluating their tabla playing skills (see Section 4.2). The results are reported in Table 4.1.

The discrete stroke recognition rate for the advanced player is the highest, which is predictable, because of the consistency of the stroke between the training session and the test sessions. Then come the recognition rates of the intermediate players and the beginner players.

Once the studies with discrete tabla strokes had been performed, I tested the system on continuous tabla phrases (on two cycles of Tintal as mentioned previously). This study was performed only with the advanced tabla player. Recognition results (with the same training sequence of discrete strokes as previously) reached 87.5%. This raw result may not be significant because of the small sample size, but it gave me the confidence that the system performed almost as well with continuous tabla phrases.

To evaluate the performance of my system, I compare it here some results achieved by other researchers whose work has already been introduced in Section 2.3. Gillet and Richard (2003) report recognition results of up to 85.6% using 5-NN on a database of 1821 strokes. The better results demonstrated by my method can be explained by three factors: the more sophisticated feature vectors extracted from the input data, the limited set of tabla strokes considered, and the much smaller set of testing data as compared with Gillet and Richard's database. Their best results are obtained using an HMM model (93.4%). Chordia (2005) reports a recognition accuracy of up to 93% with neural nets on an even larger data set.

Table 4.2 describes the machine recognition accuracy by indicating the correspondence between the ground truth (strokes that the players were asked to play) on each row, and the recognized strokes after speaker-dependent training and recognition on each column. The horizontal labels represent the ground truth labels, and the vertical axis, the recognized labels. The point here is to compare the recognition algorithm with the performance of a human listener (see Section 4.2 for human perceptual results). It is interesting to note that much confusion happens within classes of strokes (e.g. Dha and Dhin are both resonant bols with the dayan and bayan playing at the same time, similarly Ta, Tin, and Tun sound alike when played out of context, and Te and Ke are both closed bols that sound very similar although Te is played on the dayan and Ke on the bayan).

The bol recognition algorithm can be improved by taking the context into account (e.g. language modeling as described in Gillet and Richard (2003)). There could also be a feedback loop between the recognition and the prediction engine to make sure that the recognized bol falls within a category of "legal" bols based on the preceding bols (I don't propose to take following bols into account to avoid causality issues). However, the system models a 10% recognition error in the prediction engine, which makes sure that the tabla phrase fits within the constraints of tabla grammar.

Tabla Phrase Prediction

The tabla phrase prediction algorithm has been evaluated for Tintal, the most common tala (see Figure 2-1 in Section 2.2). In Figure 4-4 I plot the algorithm's cumulative prediction error rate from the first stroke until the 32nd stroke. This graph is based on the algorithm example presented in Section 3.5. The algorithm runs here with a constant tempo, and one stroke per beat so that I can identify issues related to bol prediction rather than beat keeping issues, which are anyway related considering the algorithm's learning pattern.

The first six strokes are completely random and therefore the error rate is at its peak. As the algorithm learns about regularity in the data, the error rate starts to decrease. Then around nine strokes, as the higher-level structure of the theka emerges, the prediction breaks down again, and the error goes up until the tala's 16 beat. When the algorithm recognizes a recurring pattern through the repetition of the 16-beat cycle, the error rate decreases again, and continues to do so until the pattern undergoes another unexpected change.

With such high error rates, it is difficult to know whether this approach is satisfying to tabla musicians who might be either very demanding about the algorithm's output performance, or who might be content with the beat keeping ability of the algorithm instead. This issue will be discussed in the qualitative evaluation section which comes next.

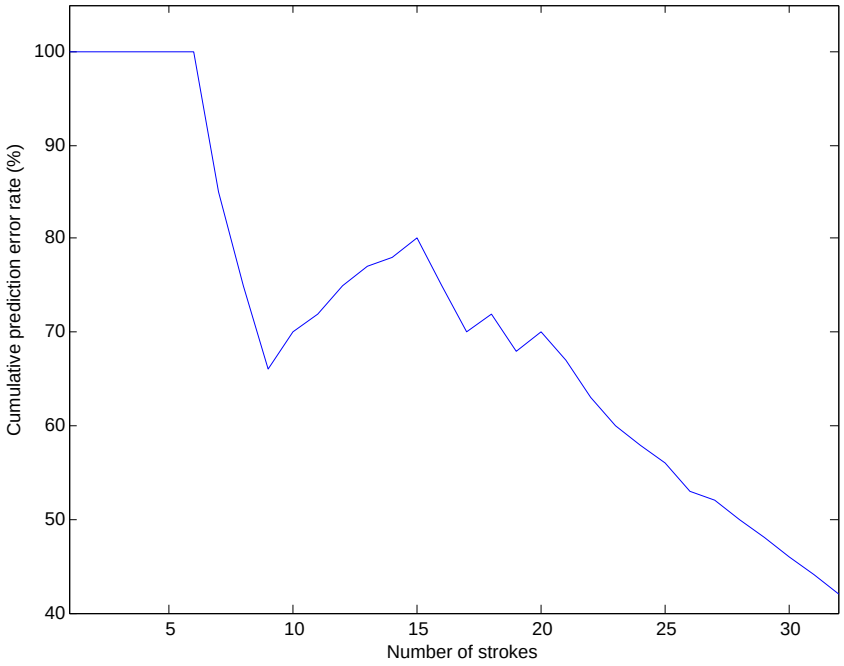


Figure 4-4: Evolution of the prediction error rate during Tintal

In the current scenario, the algorithm is expected to predict the next single stroke. However in extreme cases, supposing the tabla plays four strokes per beat at 80 beats per minute, each stroke lasts approximately 187 ms. If the tabla players play combination beats (a rapid succession of four bols), each stroke lasts less than 50 ms. High latency networks over long distances (algorithmic delay might be negligible compared to the latency of packet switching) implies that the algorithm should be able to predict several strokes in advance. In this situation, I expect the current algorithm to break down much more rapidly as the error rate struggles to decrease.

One way to improve the prediction output realism would be to increase the number of tabla playing rules that the output should abide by, while limiting the historical input data to constrain the algorithmic delay. Even though this method would not decrease the error rate, a set of well-designed (i.e. musically informed) constraints would ensure that the system performs more like a human player and less like a sequencer. In any case, even though the historical data helps decrease the error rate, its main purpose is to convey a certain "style of playing" that emerges through the balance between variety and repetition.

4.2 Qualitative Experiments

This section describes subjective user tests involving tabla players and trained listeners, and the results of these tests.

Method

This section gives an overview of the test procedure (shown in detail in Section A.2) and the data set.

Eight subjects participated in the study. Most of them were from the Boston area, including MIT and Harvard. There is a relatively large number of tabla players in this area, however most of them were out on vacation during the evaluation period. Many expressed their interest in this project and asked whether I would be conducting studies again during the academic year.

Subjects were recruited by e-mail (see copy of the advertisement in Section A.1) and were given gift coupons for their participation. Needless to say, the participants in the study had a favorable frame of mind towards the project.

The detailed study protocol is presented in Section A.2. I discuss here the rationale behind the study.

After asking the users to rate themselves as a tabla player (beginner, intermediate, or advanced), I subjected them to a series of tabla playing exercises so that I could rate all of them on a common basis (i.e. technique ("cleanliness" and consistency of their strokes, etc.), knowledge of various talas). Following this, in the first part of the study, I asked each one of them to play some sequences of bols and thekas to train the recognition algorithm and evaluate its performance (both with discrete bols, and with continuous phrases). In the second part of the study, after a short break, I asked them to play the role of an audience member and answer some questions based on what they heard so that I could evaluate their response to the tabla prediction engine.

The tests included what I like to call a "Musical Turing test" where participants were asked to distinguish a rhythmic sequence produced by a human player from a sequence generated by the computer.

In this Turing test, each rhythmic phrase presented to the user is chosen randomly among the following possibilities:

- a digital audio recording of a real tabla performance,
- phrases generated using a sequencing software and tabla sound synthesizer,

- phrases resulting from a recorded input to the recognizer which triggers tabla samples, and
- phrases generated from an input to the recognizer followed by the prediction engine output.

For the purpose of this evaluation, the TablaNet system had a limited set of functionality. The evaluation was performed under constrained conditions:

- medium tabla playing speed,
- unidirectional system with no network transmission/reception,
- no pitch consideration, no background *tanpura* (drone instrument)—although this could make the session much more interesting and reminiscent of a real-world situation,
- microphone instead of sensors.

As informed listeners (trained in playing the tabla), I asked the participants not only to evaluate the flow and naturalness of the sequences of strokes (e.g. variety, quantization), but also the quality of the audio output. The user study combined tests in a laboratory setting (playing or listening to bols out of context), as well as tests that were conducted in a setting propitious to musical exchange.

Results

Based on my analysis of the subjects' playing skills, I had the following distribution of tabla players: four beginners, three intermediate players, one advanced player. Questionnaire responses are included as an appendix (Section A.3) to my master's thesis. This section compiles some of the study results that are available in their raw form in the appendix.

To evaluate the confusion matrix for machine recognition presented in the previous section against human perception, I asked the users to name discrete strokes that I played to them ("blind test"). I do not present the results in the form of a confusion matrix here, but I highlight my findings. Most listeners, including advanced players, had a difficult time distinguishing between Te, Re, and Ke. Although they correspond to different musical gestures, these three bols sound indeed very similar. Intermediate players (and beginners) also had difficulties between Dha and Dhin, and Tun and Tin. In addition, beginners were sometimes confused between Tin and Te, Ge and Dhin, Tun and Ta, or Dha and Dhin. These observations show the importance of experience in ear training, but also the importance of context to help in recognizing strokes. Halfway in the session, subjects underwent a short training where they were told which audio stream (bol sound) corresponded to which label. Then, when presented with a sequence of bols (as opposed to individual bols), most players (including

some beginners) could identify the bols with very little mistakes. In particular, the compound bol *TeReKeTe* could be easily identified (because of the musical expectation associated with it, and the minute difference in spectrum between each stroke), whereas when taken individually, the bols *Te*, *Re* and *Ke* were difficult to distinguish one from another.

To evaluate the TablaNet audio quality, I asked users to participate in a "blind test". I played tabla phrases using either synthesized tabla sounds from a tabla software sound synthesis, or tabla samples that I collected and asembled into meaningful rhythmic phrases using the TableNet software, and asked them to rate the ones they prefered. At the exception of one user, they all chose the TablaNet output.

Another experiment consisted in asking users to identify the number of beats, and if possible, the name of the tala and the sequence of bols from a rhythmic phrase that was played to them. This task was particularly difficult for beginners. Intermediate players could distinguish between open and closed bols, but were confused as to their specific names. The advanced player was able to perform the task perfectly, probably operating with some kind of template matching since there was no additional context that was presented to her.

When asked to predict the next bol halfway through a rhythmic sequence, again out of context, beginners had a very hard time keeping the beat while simultaneously listening to the bols and extracting the phrase's structure. Intermediate players tended to tap the beat with every stroke instead of grouping them, but some did predict the next bol correctly. In fact, some would guess the phrase's structure correctly and would predict the repetition of a pattern correctly, but would not be able to guess if there was a variation in the structure. The advanced player, however, was able to perform this task brilliantly, not only predicting the next stroke, but also guessing the correct number of beats and the tala from limited information (half a rhythmic structure). Some beginners were able to do the same after three cycles or more were played.

Analysis

This section discusses the qualitative results presented previously.

The confusion matrix presented in section 4.1 matches the recognition capability of most human tabla players, suggesting that the spectrum is only part of the information that is used by the human auditory system to recognize tabla strokes. Humans as well as machines seem to be able to distiguish mostly between categories of bols (closed, open or resonant) rather than specific bols, while the cognitive system and upper-level processes that get honed with experience and continued exposure to tabla sounds and rhythms extract specific bol values from contextual information and from the relative differences of consecutive bols (in spectrum and timing).

It was interesting to note that in many instances, when playing a sequence of bols, beginner players had some difficulty in producing a clean, steady sound. It took them one or two more strokes to feel relaxed and confident enough with their technique. This affected the recognition model because of the lack of consistency in the sound of certain strokes depending on where they were played in the sequence. A specific stroke recognition algorithm could take this information into account and modify its stroke recognition model based on the position of the stroke.

The findings in this study highlight the importance of "active listening" in tabla and rhythmic (or musical) education in general. Players with more experience benefited from a body of knowledge that shaped their expectation. In this context of education, an automatic tabla transcription system which displays bol names along with their audio counterpart in a performance can prove to be a useful tool for active listening and learning.

It is noteworthy that beginners had a difficult time with synthesized sounds, not being able to name them as accurately as sampled or recorded strokes. Probably for this reason, most beginners preferred sampled sounds to synthesized sounds compared to more experienced tabla players.

Finally, as a comparative rating of the stroke recognition system, it is useful to note that it performed best with intermediate players. Beginner players played some strokes inconsistently so the sound would vary between various instances, including playing combined strokes (left hand with the right hand, like Dha) with a slight asynchrony, resulting in two onsets instead of one. Advanced players, on the other hand, had the tendency to embellish their playing with ornamentations that did throw the recognition and prediction engines off guard. This is definitely one aspect that I had not taken into account while designing the system.

Overall, the user studies were one of the most enjoyable and educational parts. In the future, I would recommend doing user tests much earlier in the process to be able to incorporate their feedback in the system design.

4.3 Discussion

This section addresses how the results of the quantitative and qualitative evaluations support the hypotheses of this research.

As a reminder, the main hypotheses of this research are:

1. playing on a predictive system with another musician located across the network is experientially, if not perceptually, similar to playing with another musician located in the same room in that it provides as much "satisfaction" to the musicians and the audience;

2. a recognition and prediction based model provides an adequate representation of a musical interaction; and

3. a real-time networked system suggests new means of collaboration in the areas of distance education, real-world and virtual-world interactions, and online entertainment.

In particular, the recognition engine supports distance education by letting students hear a particular *bol* or stroke, and see its name displayed on the screen. In fact, this feature can even be used in the context of face-to-face learning where this tool can provide much needed support to students, including those with mild learning disabilities for whom multi-sensory data can greatly enhance the learning experience.

Interestingly, the recognition engine is more adapted to beginners who see a need for it even without network transmission (they are more tolerant to delay), whereas the prediction engine is closer to what advanced players require, especially for tight timing synchronization (more training leads to better prediction).

I also show that the prediction engine supports musical call-and-response interactions better than existing latency-ridden or phrase-looping systems.

Chapter 5

Conclusion

5.1 Contributions and Technical Relevance

This thesis presents a novel way to overcome latency on computer networks for the purpose of musical collaboration by predicting musical events before they actually occur on the network. I achieved this by developing a system, called TablaNet, that is tuned to a particular instrument and musical style, in this case the Indian tabla drum. I wrote software that recognizes drum strokes at one end, sends symbols representing the recognized stroke and its timing over the network, and predicts and synthesizes the next stroke based on previous events at the other end. What do evaluation results demonstrate?

How do the main hypotheses that drove this work fare with regards to evaluation results?

1. Playing on a predictive system with another musician located across the network is experientially, if not perceptually, similar to playing with another musician located in the same room in that it provides as much "satisfaction" to the musicians and the audience;

2. a recognition and prediction based model provides an adequate representation of a musical interaction; and

3. a real-time networked system suggests new means of collaboration in the areas of distance education, real-world and virtual-world interactions, and online entertainment.

The work described in this thesis led to the following contributions:

- I implemented a novel approach for real-time online musical collaboration,
- enabled a real-world musical interaction between two tabla musicians over a computer network,
- designed a networked tabla performance system,

- created a tabla phrase prediction engine, and
- developed a real-time continuous tabla strokes recognizer.

The TablaNet system, which was developed as an artifact to support the research presented in this thesis, consists in a unidirectional prototype that includes hardware and software components. It is suitable for testing and demonstration purposes.

This thesis itself contains a technical description of the system (architecture, design choices, implementation details), and a discussion of quantitative and qualitative evaluation results that support (or disprove) my research hypotheses. Preliminary studies suggest that although playing on a system such as TablaNet with a musician located across the network does not necessarily provide an experience identical to playing with a musician located in the same room, it creates new opportunities for learning and entertainment.

The hard theoretical limits (i.e. speed of light, networking system overhead, algorithmic delay) imposed by the physical constraints of large geographical distances demonstrate the need for systems that provide creative solutions to the latency problem. Different applications may have different requirements. The solution set forth in this work is based on the needs of musicians to remain synchronized with each other while playing together, in spite of the distance.

5.2 Applications and Social Relevance

With a system such as TablaNet, we can imagine that it would be possible for an instructor living in a city to teach music to children in villages who may not have regular access to a teacher otherwise.

It is a well-known problem in India and other countries with a large rural–urban divide that it is often difficult to find instruction for music and the arts, or even some more practical skills, beyond those of the local tradition. This applies as much to villages as it does to cities: classical musical instruments may be difficult to come by in rural areas, while cities may have limited access to folk culture from villages even nearby. In fact, this applies even more to areas with different musical traditions (e.g. between the North and the South of India; or what if I live in the middle of Iowa and want to learn the *mridangam*—a South Indian percussion instrument?) Although it is absolutely necessary to preserve local traditions, one way to keep them alive is in fact to make them live and evolve. With people being increasingly mobile and "connected", even in rural areas (especially in countries like India), communication services over data networks are becoming ever more relevant, both socially and culturally. Therefore they can be counted upon as a possible means to sustain indigenous artistic traditions.

With Western economic hegemony (villages in India may not have running potable water but most of them have a Coca Cola stand) permeating into the cultural realm (the influence of MTV mentioned in Section 1.1), it is often feared that local traditions are in danger of becoming extinct if not preserved and perpetuated through teaching and practice while keeping up with modernization. As shown in Section 2.2, Indian music is primarily based on an oral tradition, so a system such as TablaNet, even if it cannot completely replace a live teacher–student interaction, can at least substitute some in-between sessions with online rehearsals. In addition, local traditions may benefit from increased exposure to different, more complex, maybe more sophisticated, musical styles or practice elements (e.g. the use of more elaborate *talas* in contemporary classical Indian music, rather than the ubiquitous *tintal*) by evolving and therefore staying alive.

The $100 laptop developed by One Laptop Per Child (OLPC) seems like an ideal platform to implement and distribute the TablaNet system: the built-in wireless mesh network capability can enable children to play music with each other at a distance through an ad-hoc network as well as through the Internet.

Apart from distance education, preliminary testing also shows that the system is suitable for distributed online "jamming", especially in the context of musical call-and-response interactions, which are very well established in Indian music, and therefore carry relevance in entertainment (two amateur musicians wanting to "jam" together from their respective homes, or a live network music performance between musicians over long distances). Collaborations never heard before are thereby made possible.

5.3 Future Work

A disclaimer should be made here to my statement in Section 1.2 about visual interaction between musicians. It must be noted that a traditional Indian music performance does not actually rely only on auditory cues. For instance, audience members display their appreciation using non-verbal speech sounds and body gestures (specifically hand and head movements). And musicians sharing the stage do glance at each other on occasions. However, I suggest that these modalities do not contribute so much to musician synchronization as to an "excitement factor" resulting from a live stage performance. Therefore, I maintain that the study conducted in this thesis, which deals mainly with the synchronization aspect of networked music collaboration, does not suffer from leaving other modalities out. Nevertheless it would be interesting to research the role of other communication channels in a musical collaboration, including the role of audience interaction, and study how they can be transmitted to distant protagonists.

In fact, it had been suggested that as a preliminary study, I study the role of

visual contact between musicians, in particular between tabla players either in the context of education, or during a rehearsal or performance. Time constraints did not allow me to perform the study, but it would be rather interesting to conduct such an experiment, without any latency involved, by placing a screen between two tabla players and asking them to play together, and compare their experience with one where they would be normally facing each other.

Further work on the TablaNet system itself includes the implementation of the network message passing protocol and a graphical user interface that would allow users to run the system on their own. Additionally, instead of using sample playback for the sound synthesis engine, a physical model (waveguide synthesis) of a tabla drumhead would be a significant improvement. This would enable a fine level of control mimicking playing techniques involving, for instance, pitch slides. Also, the current model does not account for differences in tuning. The stroke recognizer is trained for particular tabla sounds tuned to a particular pitch. And by playing back pre-recorded samples, the system at the output does not account for different pitches at the input. If not using a physical model for sound synthesis, an intermediate solution would be to add a pitch tracker at the input, and provide a phase-vocoder-based harmonizer at the output to tune the sample output to the required pitch. In fact, I experimented with this in CSound (it was easy to program and it worked great), but I did not implement it in the current version of the software.

Various improvements in the recognition and prediction algorithms could possibly be achieved by developing a machine listening model based more closely on a human auditory perception model rather than statistical machine learning. For instance, by using a Constant-Q Transform (CQT), as proposed by Brown (1990), the recognizer may have access to acoustic information relevant to the human ear more satisfactorily than by using a Fast Fourier Transform (FFT). Similarly, a beat tracking algorithm based on the findings of Povel and Okkerman (1981) could account for more accurate tempo estimation.

Finally, I propose to extend the system by enabling more than two tabla players to play together. An Internet-based percussion ensemble! It would also be interesting to support other types of musical instruments, in particular melodic ones. This could lead to a true Indian musical performance where the tabla accompanies a distant vocalist or a solo instrumentalist.

I hope that this thesis will provide a foundational work for researchers who wish to further the principles presented here to other instruments and musical styles. It was my wish to document the user studies by producing video segments to illustrate various usage scenarios of the system in action (e.g. rhythmic accompaniment, call and response). Unfortunately, lack of time prevented me from doing so, but I am confident that I will be able to cater to this in the near future.

Appendix A

Experimental Study

A.1 Study Approval

The following pages contain the forms submitted to the MIT Committee On the Use of Humans as Experimental Subjects (COUHES), as well as their approval notices.

A.2 Study Protocol

This section contains the protocol documentation used during the subjective evaluation of the TablaNet system.

A.3 Questionnaire Responses

This section contains the anonymous responses to the questionnaire of the users who participated in the study.

Bibliography

audiofabric, 2007. URL http://www.audiofabric.com.

digitalmusician.net, 2006. URL http://www.digitalmusician.net/.

ejamming, 2007. URL http://www.ejamming.com/.

indabamusic, 2007. URL http://www.indabamusic.com/.

Jamglue, 2007. URL http://www.jamglue.com/.

Lightspeed Audio Labs, 2007. URL http://www.lightspeedaudiolabs.com/.

Lemma. URL http://visualmusic.org.

Max/msp. URL http://www.cycling74.com/products/maxmsp.

Musicolab, 2003. URL http://www.reggieband.com/musicolab/.

Rocketears, May 2004. URL http://www.jamwith.us/.

splice, 2007. URL http://www.splicemusic.com/.

Swarshala, 2007. URL http://www.swarsystems.com/SwarShala/.

Taalmala, 2005. URL http://taalmala.com/.

Vstunnel, 2005. URL http://www.vstunnel.com/en/.

P. Allen and R. Dannenberg. Tracking musical beats in real time. *Proceedings of the 1990 International Computer Music Conference*, pages 140–143, 1990.

A. Barbosa. Displaced Soundscapes: A Survey of Network Systems for Music and Sonic Art Creation. *Leonardo Music Journal*, 13(1):53–59, 2003.

A. Barbosa and M. Kaltenbrunner. Public sound objects: a shared musical space on the web. *Web Delivering of Music, 2002. WEDELMUSIC 2002. Proceedings. Second International Conference on*, pages 9–16, 2002.

R. Bargar, S. Church, A. Fukuda, J. Grunke, D. Keislar, B. Moses, B. Novak, B. Pennycook, Z. Settel, J. Strawn, et al. AES white paper: Networking audio and music using Internet2 and next-generation Internet capabilities. Technical report, AES: Audio Engineering Society, 1998.

B. Bel. A symbolic-numeric approach to quantization in music. *Third Brazilian Symposium on Computer Music*, 5–7 August 1996.

B. Bel. Bol processor: Using text representation for rule-based musical analysis, composition and improvisation. Presentation slides, seminars in Graz (Austria) and Edinburgh (UK), October 2006.

B. Bell and J. Kippen. Bol processor grammars. *Understanding music with AI: perspectives on music cognition table of contents*, pages 366–400, 1992.

R. Bellman. *Dynamic Programming*. Courier Dover Publications, 2003.

E. Berg. *An Analysis of the Perception of Rhythmic Structure in Music in Free Rhythm with a Focus on North Indian Classical Alap*. PhD thesis, Ohio State University, 2004.

R. Bhat. Acoustics of a cavity-backed membrane: The Indian musical drum. *The Journal of the Acoustical Society of America*, 90:1469, 1991.

N. Bouillot. Un algorithme d'auto synchronisation distribuée de flux audio dans le concert virtuel réparti. *Proceedings of the conférence française sur les systèmes d'exploitation CFSE*, 3, 2003.

K. Brandenburg and G. Stoll. ISO-MPEG-1 audio: a generic standard for coding of high-quality digital audio. *Journal of the Audio Engineering Society*, 42(10): 780–792, 1994.

J. Brown. *Calculation of a Constant Q Spectral Transform*. Vision and Modeling Group, Media Laboratory, Massachusetts Institute of Technology, 1990.

P. Burk. Jammin'on the Web-a new Client/Server Architecture for Multi-User Musical Performance. *ICMC 2000*, 2000.

J.-P. Caceres. Soundwire, 2007. URL `http://ccrma.stanford.edu/groups/soundwire/`.

C. Chafe. Statistical Pattern Recognition for Prediction of Solo Piano Performance. In *Proc. ICMC, Thessaloniki*, 1997.

C. Chafe. Distributed Internet Reverberation for Audio Collaboration. In *AES (Audio Engineering Society) 24th Int'l Conf. on Multichannel Audio*, 2003.

C. Chafe and R. Leistikow. Levels of Temporal Resolution in Sonification of Network Performance. *Proceedings of the International Conference on Auditory Display*, pages 50–55, 2001.

C. Chafe, S. Wilson, R. Leistikow, D. Chisholm, and G. Scavone. A Simplified Approach to High Quality Music and Sound Over IP. In *Proc. Workshop on Digital Audio Effects (DAFx-00), Verona, Italy*, pages 159–163, 2000.

C. Chafe, S. Wilson, and D. Walling. Physical model synthesis with application to Internet acoustics. *Acoustics, Speech, and Signal Processing, 2002. Proceedings.(ICASSP'02). IEEE International Conference on*, 4, 2002.

C. Chafe, M. Gurevich, G. Leslie, and S. Tyan. Effect of Time Delay on Ensemble Accuracy. In *Proceedings of the International Symposium on Musical Acoustics*, 2004.

A. Chatwani. Real-Time Recognition of Tabla Bols. Princeton University, Senior Thesis, May 2003.

A. Chatwani and A. Koren. Optimization of Audio Streaming for Wireless Networks. Technical report, Princeton University, 2004.

E. Chew and A. Sawchuk. Distributed Immersive Performance. In *National Association of Schools of Music*, 2004.

E. Chew, R. Zimmermann, A. Sawchuk, C. Kyriakakis, C. Papadopoulos, A. François, G. Kim, A. Rizzo, and A. Volk. Musical Interaction at a Distance: Distributed Immersive Performance. In *Proceedings of the MusicNetwork Fourth Open Workshop on Integration of Music in Multimedia Applications, September*, pages 15–16, 2004.

E. Chew, A. Sawchuk, C. Tanoue, and R. Zimmermann. Segmental Tempo Analysis of Performances in User-centered Experiments in the Distributed Immersive Performance Project. In *Sound and Music Computing*, 2005.

P. Chordia. Segmentation and Recognition of Tabla Strokes. In *Proc. of ISMIR (International Conference on Music Information Retrieval)*, 2005.

E. Clarke. Rhythm and timing in music. *The Psychology of Music*, 2, 1999.

M. Clayton. *Time in Indian Music: Rhythm, Metre, and Form in North Indian Râg Performance*. Oxford University Press, 2000.

J. Cooperstock and S. Spackman. The Recording Studio that Spanned a Continent. In *Proc. of IEEE International Conference on Web Delivering of Delivering of Music (WEDELMUSIC)*, 2001.

J. Cooperstock, J. Roston, and W. Woszczyk. Broadband Networked Audio: Entering the Era of Multisensory Data Distribution. *18th International Congress on Acoustics*, April 4–9 2004.

COUHES. URL `http://web.mit.edu/committees/couhes/`.

R. Dannenberg and P. van de Lageweg. A System Supporting Flexible Distributed Real-Time Music Processing. *Proceedings of the 2001 International Computer Music Conference*, pages 267–270, 2001.

R. B. Dannenberg. Toward Automated Holistic Beat Tracking, Music Analysis, and Understanding. In *ISMIR 2005 6th International Conference on Music Information Retrieval Proceedings*, pages pp. 366–373. London: Queen Mary, University of London, 2005. URL `http://www.cs.cmu.edu/ rbd/bib-beattrack.html`.

R. Duda, P. Hart, and D. Stork. *Pattern Classification*. Wiley-Interscience, 2000.

A. Eliens, M. Welie, J. Ossenbruggen, and B. Schoenhage. Jamming (on) the Web. *WWW6 / Computer Networks*, 29(8-13):897–903, 1997. URL `http://www.cs.vu.nl/ eliens/online/papers/www6/`.

G. Essl, S. Serafin, P. Cook, and J. Smith. Musical Applications of Banded Waveguides. *Computer Music Journal*, 28(1):51–63, 2004.

D. Fober, Y. Orlarey, and S. Letz. Real time musical events streaming over Internet. *Web Delivering of Music, 2001. Proceedings. First International Conference on*, pages 147–154, 2001.

D. Fober, Y. Orlarey, and S. Letz. Clock Skew Compensation over a High Latency Network. *Proceedings of the International Computer Music Conference*, pages 548–552, 2002.

O. Gillet and G. Richard. Automatic Labelling of Tabla Signals. In *Proc. of the 4th ISMIR Conf.*, 2003.

M. Goto and R. Neyama. Open RemoteGIG: An Open-to-the-Public Distributed Session System Overcoming Network Latency. *IPSJ Journal*, 43(2):299–309, 2002.

M. Goto, R. Neyama, and Y. Muraoka. RMCP: Remote Music Control Protocol—Design and Applications—. *Proc. International Computer Music Conference*, pages 446–449, 1997.

S. Gresham-Lancaster. The Aesthetics and History of the Hub: The Effects of Changing Technology on Network Computer Music. *Leonardo Music Journal*, 8:39–44, 1998.

X. Gu, M. Dick, U. Noyer, and L. Wolf. NMP-a new networked music performance system. In *Global Telecommunications Conference Workshops, IEEE*, pages 176–185, 2004.

M. Gurevich. JamSpace: designing a collaborative networked music space for novices. *Proceedings of the 2006 conference on New interfaces for musical expression*, pages 118–123, 2006a.

M. Gurevich. JamSpace: a networked real-time collaborative music environment. *Conference on Human Factors in Computing Systems*, pages 821–826, 2006b.

G. Hajdu. quintet.net, 2007. URL `http://www.quintet.net/index.asp`.

S. Hang. The Synchronous Approach to Real-Time Internet Musical Performance.

M. Helmuth. Sound Exchange and Performance on Internet2. *Proceedings of the 2000 International Computer Music Conference*, pages 121–124, 2000.

P. Herrera, A. Yeterian, and F. Gouyon. Automatic classification of drum sounds: a comparison of feature selection methods and classification techniques. *Proceedings of 2nd International Conference on Music and Artificial Intelligence, Edinburgh, Scotland*, 2002.

P. Herrera, A. Dehamel, and F. Gouyon. Automatic labeling of unpitched percussion sounds. *114th AES Convention*, 2003.

J. Hun Roh and L. Wilcox. Exploring Tabla Drumming Using Rhythmic Input. In *CHI'95 proceedings*, 1995.

R. Jackendoff and F. Lerdahl. *A Generative Theory of Tonal Music*. MIT Press, 1996.

T. Jehan. *Creating Music by Listening*. PhD thesis, Massachusetts Institute of Technology, 2005.

T. Jehan, D. Overholt, H. Solís Garci, and C. Vaucelle. Real-time distributed media applications in lans with osc. In *Proceedings of the 1st Open Sound Control conference*, Berkeley, July 2004.

S. Jorda. Faust Music on Line: An Approach to Real-Time Collective Composition on the Internet. *Leonardo Music Journal*, 9:5–12, 1999.

A. Kapur. The Electronic Tabla. Master's thesis, Princeton University, 2002.

A. Kapur, G. Essl, P. Davidson, and P. Cook. The Electronic Tabla Controller. *Journal of New Music Research*, 32(4):351–359, 2003a.

A. Kapur, G. Wang, P. Davidson, P. Cook, D. Trueman, T. Park, and M. Bhargava. The Gigapop Ritual: A Live Networked Performance Piece for Two Electronic Dholaks, Digital Spoon, DigitalDoo, 6 String Electric Violin, Rbow, Sitar, Table, and Bass Guitar. In *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME), Montreal*, 2003b. URL `http://gigapop.cs.princeton.edu/`.

A. Kapur, P. Davidson, P. Cook, P. Driessen, and A. Schloss. Digitizing North Indian Performance. In *Proceedings of the International Computer Music Conference*, 2004.

A. Kapur, G. Wang, P. Davidson, and P. R. Cook. Interactive Network Performance: a dream worth dreaming? *Organised Sound*, 10(03):209–219, 2005.

A. Khan and G. Ruckert. *The Classical Music of North India: The Music of the Baba Allauddin Gharana as taught by Ali Akbar Khan, Volume 1.* East Bay Books, distributed by MMB music, Saint Louis, Missouri, 1991.

J. Kippen and B. Bel. Modelling Music with Grammars: Formal Language Representation in the Bol Processor. *Computer Representations and Models in Music, Ac. Press ltd*, pages 207–232, 1992.

J. Kippen and B. Bel. Computers, Composition and the Challenge of "New Music" in Modern India. *Leonardo Music Journal*, 4:79–84, 1994.

F. Kon and F. Iazzetta. Internet music: Dream or (virtual) reality. *Proceedings of the 5th Brazilian Symposium on Computer Music*, 1998.

I. Kondo, K. Kojima, and S. Ueshima. A study of distributed jam session via content aggregation. *Web Delivering of Music, 2004. WEDELMUSIC 2004. Proceedings of the Fourth International Conference on*, pages 15–22, 2004.

D. Konstantas. A Telepresence Environment for the Organization of Distributed Musical Rehearsals. *Objects at Large" edited by D. Tsichritzis, Technical report of the University of Geneva*, 1997.

C. Latta. Notes from the NetJam Project. *Leonardo Music Journal*, 1(1):103–105, 1991.

J. Lazzaro and J. Wawrzynek. A case for network musical performance. In *Proceedings of the 11th international workshop on Network and operating systems support for digital audio and video*, pages 157–166. ACM Press New York, NY, USA, 2001. URL `http://www.cs.berkeley.edu/ lazzaro/nmp/`.

M. Lefford. Recording Studios Without Walls. Master's thesis, Massachusetts Institute of Technology, 2000.

T. Mäki-Patola. Musical Effects of Latency. *Suomen Musiikintutkijoiden*, 9:82–85, 2005.

T. Mäki-Patola and P. Hämäläinen. Effect of Latency on Playing Accuracy of Two Gesture Controlled Continuous Sound Instruments Without Tactile Feedback. *Proc. Conf. on Digital Audio Effects, Naples, Italy, Oct*, 2004a.

T. Mäki-Patola and P. Hämäläinen. Latency Tolerance for Gesture Controlled Continuous Sound Instrument Without Tactile Feedback. *Proc. International Computer Music Conference (ICMC)*, pages 1–5, 2004b.

S. Malu and A. Siddharthan. Acoustics of the Indian Drum. *Arxiv preprint math-ph/0001030*, 2000.

Y. Nagashima, T. Hara, T. Kimura, and Y. Nishibori. GDS (Global Delayed Session) Music. *Proceedings of the ICMC 2003–boundaryless music*, pages 291–294, 2003.

E. Narmour. Hierarchical expectation and musical style. *The psychology of music*, pages 441–472, 1999.

Y. Nishibori, Y. Tada, and T. Sone. Study and Experiment of Recognition of the Delay in Musical Performance with Delay. Technical report, IPSJ Tech. Rep. 2003-MUS-53, 2003.

A. Oppenheim and R. Schafer. Digital Signal Processing. Englewood Cliffs, 1975.

J. Paradiso. The Brain Opera Technology: New Instruments and Gestural Sensors for Musical Interaction and Performance. *Journal of New Music Research*, 28(2): 130–149, 1999.

A. Patel and J. Iversen. Acoustic and Perceptual Comparison of Speech and Drum Sounds in the North Indian Tabla Tradition: An Empirical Study of Sound Symbolism. *Proceedings of the 15th International Congress of Phonetic Sciences (ICPhS)*, 2003.

D. Povel and H. Okkerman. Accents in equitone sequences. *Percept Psychophys*, 30 (6):565–72, 1981.

L. Rabiner. A tutorial on hidden Markov models and selected applications inspeech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

C. Ramakrishnan, J. Freeman, and K. Varnik. The architecture of auracle: a real-time, distributed, collaborative instrument. *Proceedings of the 2004 conference on New interfaces for musical expression*, pages 100–103, 2004.

C. Raman. Indian musical drums. *Proceedings of the Indian Academy of Sciences*, 1: p179–188, 1935.

C. Raman and S. Kumar. Musical drums with harmonic overtones. *Nature*, 104:p500, 1920.

F. Ramos, M. de Oliveira Costa, and J. Manzolli. Virtual studio: distributed musical instruments on the web. *Proc. of IX Brazilian Symposium on Computer Music, Campinas, Brazil, August*, 2003.

Reginald and J. Massey. *The Music of India*. Abhinav Publications, 1996.

R. Rowe. Real Time and Unreal Time: Expression in Distributed Performance. *Journal of New Music Research*, 34(1):87–95, 2005.

R. Rowe and N. Rolnick. The Technophobe and the Madman: an Internet2 distributed musical. *Proc. of the Int. Computer Music Conf*, 2004.

G. Ruckert. *Music in North India: Experiencing Music, Expressing Culture*. Oxford University Press, 2004.

S. Russell and P. Norvig. *Artificial intelligence: a modern approach*. Prentice-Hall, Inc. Upper Saddle River, NJ, USA, 1995.

K. Samudravijaya, S. Shah, and P. Pandya. Computer Recognition of Tabla Bols. Technical report, Tata Institute of Fundamental Research, 2004.

M. Sarkar and B. Vercoe. Recognition and Prediction in a Network Music Performance System for Indian Percussion. In *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME), New York*, 2007.

A. Sawchuk, E. Chew, R. Zimmermann, C. Papadopoulos, and C. Kyriakakis. From remote media immersion to Distributed Immersive Performance. In *Proceedings of the 2003 ACM SIGMM workshop on Experiential telepresence*, pages 110–120. ACM Press New York, NY, USA, 2003.

E. Schooler. Distributed music: a foray into networked performance. *Network Music Festival,(California), pp.–, Sept*, 1993.

N. Schuett. The Effects of Latency on Ensemble Performance. *Undergraduate Honors Thesis, Stanford CCRMA, May*, 2002.

C. Shanhabi, R. Zimmermann, K. Fu, and S. Yao. Yima: a second generation continuous media server. Published in the. *IEEE Computer Magazine*, pages 56–64, 2002.

B. Snyder. *Music and memory*. MIT Press Cambridge, Mass, 2000.

J. Stelkens. peerSynth: A P2P Multi-User Software Synthesizer with new techniques for integrating latency in real time collaboration. *Proceedings of the 2003 International Computer Music Conference*, 2003.

A. Tanaka. Network Audio Performance and Installation. *Proc. Intnl Computer Music Conf*, 1999.

A. Tindale. Annotated Bibliography-Drum Transcription Models. 2005.

A. Tindale, A. Kapur, G. Tzanetakis, and I. Fujinaga. Retrieval of percussion gestures using timbre classification techniques. *Proceedings of the International Conference on Music Information Retrieval*, pages 541–4, 2004.

J. Townley. Rocket network, March 2000. URL `http://smw.internet.com/audio/reviews/rocket/`.

D. Van Steelant, K. Tanghe, S. Degroeve, B. De Baets, M. Leman, and J. Martens. Classification of percussive sounds using support vector machines. *Proceedings of the annual machine learning conference of Belgium and The Netherlands, Brussels, Belgium*, 2004.

R. Vanegas. Linking Music Students and Music Instructors Via The Internet. 2005.

B. Vercoe. Erasing the Digital Divide: Putting your Best Idea on the $100 Laptop. Keynote lecture, WORLDCOMP'06, Las Vegas, June 2006.

G. Weinberg. *Interconnected Musical Networks–Bringing Expression and Thoughtfulness to Collaborative Music Making*. PhD thesis, Massachusetts Institute of Technology, 2001.

G. Weinberg. The Aesthetics, History, and Future Challenges of Interconnected Music Networks. In *Proceedings of the 2002 Computer Music Conference*, 2002.

G. Weinberg. Interconnected Musical Networks: Toward a Theoretical Framework. *Computer Music Journal*, 29(2):23–39, 2005a.

G. Weinberg. Local Performance Networks: musical interdependency through gestures and controllers. *Organised Sound*, 10(03):255–265, 2005b.

M. Wright and D. Wessel. An Improvisation Environment for Generating Rhythmic Structures Based on North Indian" Tal" Patterns. *International Computer Music Conference, Ann Arbor, Michigan*, 1998.

M. Wright, A. Freed, and A. Momeni. Open sound control: State of the art 2003. In *Proceedings of the New Interfaces for Musical Expression Conference*, pages 153–159, Montreal, 2003.

A. Xu, W. Woszczyk, Z. Settel, B. Pennycook, R. Rowe, P. Galanter, J. Bary, G. Martin, J. Corey, and J. Cooperstock. Real-Time Streaming of Multichannel Audio Data over Internet. *Journal of the Audio Engineering Society*, 48(7/8), 2000.

M. Yoshida, Y. Obu, and T. Yonekura. A Protocol For Remote Musical Session with Fluctuated Tempo. *Proceedings of the 2004 International Conference on Cyberworlds (CW'04)-Volume 00*, pages 87–93, 2004.

M. Yoshida, Y. Obu, and T. Yonekura. A Protocol for Real-Time Remote Musical Session. *IEICE Transactions on Information and Systems*, 88(5):919–925, 2005.

J. Young and I. Fujinaga. Piano master classes via the Internet. *Proceedings of the 1999 International Computer Music Conference*, pages 135–137, 1999.