Digitizing North Indian Performance

Ajay Kapur[*†], Philip Davidson[†], Perry R. Cook[†♪], Peter F. Driessen[*] and W. Andrew Schloss[∝]

[*]Department of Electrical and Computer Engineering[*] and Music[∝], University of Victoria
akapur@ece.uvic.ca peter@ece.uvic.ca aschloss@finearts.uvic.ca
Department of Computer Science[†] (and Music[♪]), Princeton University
philipd@princeton.edu prc@cs.princeton.edu

Abstract

This paper discusses an evolution in North Indian instruments in the designing of technology to capture gestures from a performing artist. Modified traditional instruments use sensor technology and microcontrollers to digitize gestures, enabling a computer to analyze performance to synthesize sound and visual meaning. Specifically, systems were built to capture data from three traditional North Indian instruments: the tabla (a pair of tonal hand drums), the dholak (a barrel shaped folk drum played by two people), and the sitar (a 19-stringed, gourd-shelled instrument). This paper will discuss how these instruments are modified to capture gestural movement, how these signals are mapped to sounds and graphical feedback, and show examples of the new instruments being used in live performance. The hardware is built to try and preserve the techniques passed down from generations of tradition; however, modified performance techniques with the aid of a laptop are also introduced.

1 Introduction

"Once, a long time ago, during the transitional period between two Ages… people took to uncivilized ways … ruled by lust and greed [as they] behaved in angry and jealous ways, [while] demons, [and] evil spirits… swarmed the earth. Seeing this plight, Indra (The Hindu God of thunder and storms) *and other Gods approached God Brahma* (God of creation) *and requested him to give the people a Krindaniyaka* (toy) *… which could not only be seen, but heard, … [to create] a diversion, so that people would give up their bad ways." (Courtney 1995)* These Krindaniyakas which Brahma gave to humans included the tabla, the dholak, and the sitar.

Tabla are a pair of hand drums traditionally used to accompany North Indian classical vocal and instrumental music. The silver, larger drum (shown in Figure 1(a)) is known as the *Bayan*. The smaller wooden drum is known as the *Dahina*. (Courtney 1995) The dholak (shown in Figure 1(b)) is a barrel shaped hand drum originating in Northern India, with two membranes on either side of the barrel. (Kothari 1968) Two musicians play the dholak. The first musician strikes the two membranes with their left and right hands. The second musician sits on the other side of the drum, striking the barrel with a hard object, such as a spoon or stick, giving rhythmic hits similar to a woodblock sound. (Bagchee 1998) The sitar is *Saraswati's* (the Hindu Goddess of Music) 19-stringed, gourd-shelled, traditional North Indian instrument (shown in Figure 1(c)). It is used to perform *Ragas* (the melodic mode, scale, order, and rules of a particular piece of Indian classical music). (Menon 1974)

(a) (b) (c)

Figure 1. (a) tabla, (b) dholak, and (c) sitar.

The goal of this research is to use sensor technology to extract musical information from a performing tabla, dholak, and sitar player, deducing gestural information to trigger real-time sounds and graphics. The hardware is built to try and preserve the techniques passed down from generations of tradition, however, modified performance techniques with the aid of a laptop are also introduced. In this paper, the following subjects will be presented:

- The technology used to create the gesture capturing systems.
- The creation of the real-time Electronic Tabla controller (ETabla).
- The creation of the real time Electronic Dholak controller (EDholak).
- The creation of the real time Electronic Sitar controller (ESitar).
- Discussions on what was learned from building all three controllers.

2 Technology

This section describes the technology used to create a musical controller. A controller is a device made of different sensors that measures human interaction and converts gestural movement into the digital realm. For example, a mouse is a controller that uses an infrared LED sensor to convert hand movement into x and y coordinates on a computer screen (Brain 1998). A musical controller takes input from a musician and maps it into timbre, rhythm, pitch, or algorithmic parameters that can trigger or control recorded, synthesized, or physically modeled sound (Cook 2001).

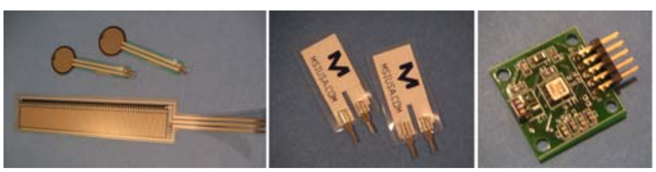
2.1 Sensors

The ETabla, EDholak, and ESitar use a variety of sensors to capture gestural data, including force sensing resistors (FSRs), piezo, and accelerometers.

Force sensing resistors (FSRs) use the electrical property of resistance to measure the force (or pressure) exerted by a user. They essentially are force to resistance transducers. When force is applied, conductivity increases as the connection between conductors is improved. A variety of FSRs are shown in Figure 2(a).

Piezo sensors take advantage of the piezoelectric effect in which mechanical energy is converted to electrical. Electrical charge results from the deformation of polarized crystals when pressure is applied. Piezo sensors are shown in Figure 2(b).

Accelerometers are sensors that measure acceleration using an electrical mass spring system. Accelerometers can easily be used to deduce tilt or rotation in three axes by wiring three components together. An accelerometer mounted on a board designed by Pascal Stang at Stanford University measures rotation in three axes and is shown in Figure 2(c). (Horenstein 1996)



(a) (b) (c)

Figure 2. (a) Force Sensing Resistors, (b) Piezo, (c) Accelerometer

2.2 Microcontrollers

Microcontrollers are small, low-cost computers, designed to accomplish simple tasks on programs loaded in Read Only Memory (ROM). Analog-to-digital converters translate voltage readings from the sensors to bits that microcontrollers can use to deduce musical information. The microcontrollers then use MIDI (Musical Instrument Design Interface) protocol or OSC (OpenSound Control) (Wright, Freed, Momeni, 2003) to send messages to a sampler, sequencer, or laptop to trigger sounds and/or graphics. The ETabla, EDholak and ESitar are built with one of two microcontrollers: The Parallax Basic Stamp IIsx (Parallax 2004) or the Atmel AVR AtMega16. (Wilson et al. 2003)

3 The Electronic Tabla Controller

3.1 Tabla Technique

It is important to understand the traditional playing style of the tabla to see how our controller models its hand movement. Figure 3 is a picture explaining the names of the different parts of the tabla *pudi* (drum head).

Figure 3. Parts of the tabla pudi (drum head): (*chat, maidan*, and *syahi*)

Bayan Strokes. There are two strokes played on the *Bayan*. The *Ka* stroke is executed by slapping the flat left hand down on the *Bayan* as shown in Figure 4(a). Notice the tips of the fingers extend from the *maidan* through to the *chat* and over the edge of the drum. The slapping hand remains on the drum after it is struck to kill all resonance before it is released away. The *Ga* stroke, shown in Figure 4(b), is executed by striking the *maidan* directly above the *syahi* with the middle and index fingers of the left hand. When the fingers strike, they immediately release away from the drum, to let the Bayan resonate with sound. The heel of the left hand controls the pitch of the *Ga* stroke, as shown in Figure 4(c). It controls the pitch at the attack of the stroke, and can also bend the pitch while the drum is resonating. Pitch is controlled by two variables of the heel of the hand: force on to the *pudi*, and the position of the hand-heel on the *pudi* from the edge of the *maidan* and *syahi* to the center of the *syahi*. The greater the force on the *pudi*, the higher the pitch. The closer to the center of the *syahi*, the higher the pitch. (Courtney 1995)

(a) (b) (c)

Figure 4. Traditional strokes played on the *Bayan*: (a) *Ka,* (b) *Ga,* (c) pitch bending *Ga* stroke

Dahina Strokes. There are six main strokes played on the *Dahina*. The *Na* stroke, shown in Figure 5(a), is executed by lightly pressing the pinky finger of the right hand down between the *chat* and the *maidan*, and lightly pressing the ring finger down between the *syahi* and the *maidan* in order to mute the sound of the drum. Then one strikes the chat with the index finger and quickly releases it so the sound of the drum resonates. The *Ta* stroke is executed by striking the middle finger of the right hand at the center of the *syahi*, as shown in Figure 5(b). The finger is held there before release so there is no resonance, creating a damped sound. The *Ti* stroke, shown in Figure 5(c), is similar to *Ta* except the middle and ring finger of the right hand strike the center of the *syahi*. This stroke does not resonate and creates a damped sound. The *Tu* stroke is executed by striking the *maidan* with the index finger of the right hand and quickly releasing, as shown in Figure 5(d). This stroke resonates the most because the pinky and ring fingers are not muting the *pudi.* (Courtney 1995) The *Tit* stroke, shown in Figure 5(e), is executed similar to *Na*, by lightly pressing the pinky finger of the right hand down between the *chat* and the *maidan*, and lightly pressing the ring finger down between the *syahi* and the *maidan*. The index finger then strikes the *chat*, quickly releasing to let it resonate. The index finger strike on the *chat* is further away from the pinky and ring finger, than it is on the *Na* stroke. *Tira* is a combination of two strokes on the *Dahina*, which explains the two syllables of the stroke. It is executed by shifting the entire right hand from one side of the drum to the other. It creates a damped sound at each strike. This stroke is shown in Figure 5(f) and Figure 5(g).

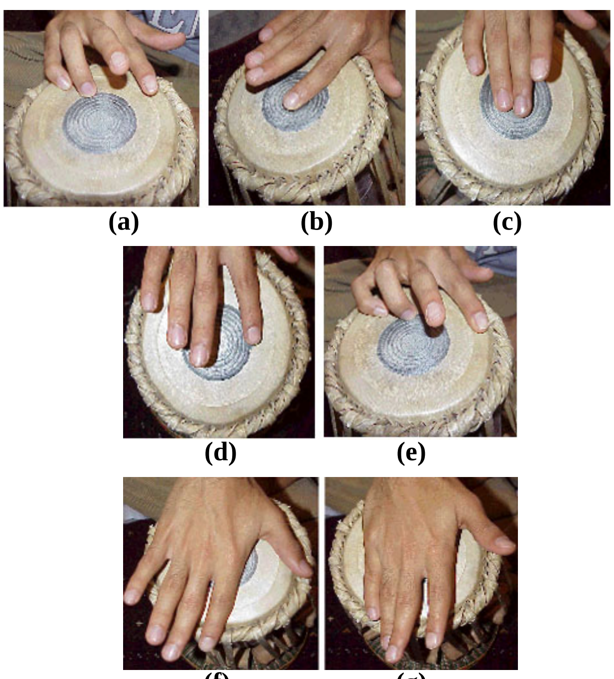
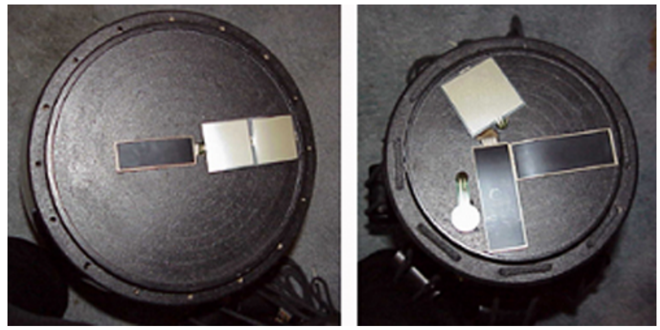


Figure 5. Traditional strokes played on the Dahina: (a) *Na*, (b) *Ta*, (c) *Ti*, (d) *Tu*, (e) *Tit*, (f) and (g) *Tira*

3.2 The MIDI Tabla Controller

The controller is modeled based on the hand positioning and movements of the strokes discussed. The ETabla is decoupled from the computer generated sound source and hence achieves a new versatility while preserving the refined performance practices of traditional play. The ETabla is built using square FSRs, small FSRs, long FSRs (which obtain both force and position in one axis), and the Parallax Basic Stamp IIsx.



(a) **(b)**

Figure 6. FSR placement on Electronic Tabla.

The Bayan Controller. The *Bayan* controller was created using two square FSRs, and one long FSR. Figure 6(a) shows a layout of these FSRs. The top square FSR is used to capture *Ka* stroke events, when a player slaps down with their left hand. If it receives a signal, then the other two FSRs are ignored. The square FSR in the middle captures *Ga* stroke events when struck by the middle and index finger of the left hand. The long FSR controls the pitch of the *Ga* stroke events, using two variables: force and position. The greater the force exerted by the heel of the left hand the higher the pitch. The closer the heel of the hand gets to the *Ga* FSR the higher the pitch. The circuitry allows the pitch to be bent after a *Ga* stroke is triggered.

The Dahina Controller. To implement the *Dahina* controller, four FSRs: two long FSRs, one square FSR, and one small FSR were used. Figure 6(b) shows a layout of these FSRs. The small FSR triggers a *Tit* stroke event. It measures the velocity of the index finger's strike. The square FSR triggers a *Tira* stroke event. It measures the velocity of the hand slapping the top of the drum. If the *Tira* FSR is struck, all other FSRs are ignored. If the *Tit* FSR is struck both long FSRs are ignored. The long FSR on the right in Figure 6(b) is the ring finger FSR, and the long FSR on the left is the index finger FSR. If there is a little force on the ring finger FSR (modeling a mute), and the index finger FSR is struck at the edge of circle, a *Na* stroke is triggered. If the index finger FSR is struck near the center of the circle, a *Ta* stroke is triggered. If there is no force on the ring finger FSR, and the index finger FSR is stuck, then a *Tu* stroke is triggered. When the ring finger FSR is struck with enough force, and not held down, then a *Ti* stroke is triggered. Figure 7 shows a picture of both controllers in their constructed Tabla encasements. The force sensing resistors were placed on top of custom built wood pieces, and covered with neoprene as a protective layer, making the instrument acoustically quiet, and providing a flexible texture for ETabla performance. (Kapur et al. 2003)

Figure 7. Different views of the Electronic Tabla.

4 The Electronic Dholak Controller

4.2 The Traditional Dholak of India

The dholak is a barrel shaped hand drum originating in Northern India. It has two membranes on either side of the barrel, creating higher tones on the smaller end, and lower tones on the larger end. (Kothari 1968) The smaller side has a simple, single layer membrane, where as the larger side has *Dholak masala* (a composition of tar, clay and sand) attached to the inside of the single layer membrane, to lower the pitch and produce a well defined tone. The dholak can be tuned in two ways depending on the type of drum. The traditional dholak are laced with rope, so tuning is controlled by adjusting a series of metal rings that determine tightness of the rope. Modern dholaks have metal turnbuckles which are easily adjusted for desired tone. The dholak is widely used in folk music of villages of India. It is common for folk musicians to build dholaks themselves from commonly available material. They then use the drums in musical rituals and special functions such as weddings, engagements and births. (Sharma 1997)

Two musicians play the dholak. The first musician strikes the two membranes with their left and right hands. There are two basic playing techniques; the open hand method is for louder playing, while the controlled finger method is for articulate playing. There are a few different positions to play the dholak, but the most popular is squatting with the drum in front, the bass head on the left, and the treble head on the right. The second musician sits on the other side of the drum, facing the first musician. They strike the barrel with a hard object, such as a spoon or stick, giving rhythmic hits similar to a woodblock sound. (Bagchee 1998)

4.2 The MIDI Dholak Controller

The design of the Electronic Dholak (shown in Figure 8(a)) is inspired by the collaborative nature of the traditional drum. Two musicians play the EDholak (shown in Figure 8(b)), the first striking both heads of the double-sided drum, and the second keeping time with a *Digital Spoon* and manipulating the sounds of the first player with custom built controls on the barrel of the drum and in software. We further explored multiplayer controllers by networking three drummers playing two EDholaks at two geographically diverse sites.

Finger strikes are captured by five piezo sensors (three for the right hand and two for the left hand) which are stuck directly on the EDholak's drum skins. Sensors are placed in positions that correlate to traditional Indian drumming (similar to tabla strokes described before). The left drum-skin head captures *Ga* and *Ka* strokes, while the right hand drum-skin captures *Na*, *Ta* and *Ti* strokes.

The *Digital Spoon* has a piezo sensor attached to the back of a flat wooden spoon. There is neoprene padding covering the piezo to keep the striking of the *Digital Spoon* acoustically quiet. The spoon player has the option of striking anywhere on the drum, or floor, triggering a audio/visual response, or striking on a linear force sensing resistor (FSR) on the *EDholak Controller Box*, which augments the audio/visual of the spoon strike and the audio/visual instances of all EDholak finger strikes. The *Controller Box* has a long FSR and a knob that the spoon player can use with his left hand to augment all sounds/graphic instances.

All piezo triggers are converted to MIDI by the *Alesis D4* (Alesis 2004) 8-channel Drum trigger box. The *Controller Box* is built using a Parallax Basic Stamp that converts all sensor data to MIDI. When two EDholaks are used in distinct locations, piezo generated MIDI signals are transferred using *GIGAPOPR* (custom built software created for *Gigapop Ritual* performance at NIME 2003) (Cook 2003) and then processed and merged together by an *Alesis D4*.

All MIDI messages get funneled to the *EDholak MIDI Control Software* written for Windows (shown in Figure 8(c)). This software is used by the spoon player to control many parameters of a performance. A user can toggle between a networked performance (two EDholaks sending MIDI messages) and just one. The user can pre-program patches which they wish to use in performance, in order of occurrence, and simply use the mouse to switch between them during a concert. The software also maps the *Control Box* sensors (Knob, and FSR) to different MIDI control parameters such as pitch, sweep, color, pan, and volume, augmenting sounds of piezo MIDI signals.

(a) (b) (c)

Figure 8. The Electronic Dholak.

5 The Electronic Sitar

5.1 Sitar Technique

It is important to understand the traditional playing style of the sitar to comprehend how our controller captures its hand gestures. It should be noted that there are two main styles of sitar technique: Ustad Vilayat Khan's system and Pandit Ravi Shankar's system. The main difference between the styles are that Ustad Vilayat Khan, performs melodies on the higher octave, eliminating the lowest string for the instrument, where as Pandit Ravi Shankar's has more range, and consequently melodies are performed in the lower octaves. (Bagchee 1998) The ESitar is modeled based on the Vilayat Khan system.

A performer generally sits on the floor, in a cross-legged fashion. Melodies are performed primarily on the outer main string, and occasionally on the copper string. A sitar player uses his left index finger and middle finger, as shown in Figure 9(a), to press the string to the fret for the desired *swara* (note). In general, a pair of frets are spaced a half-step apart, with the exception of a few that are spaced by a whole step. The frets are elliptically curved so the string can be pulled downward, to bend to a higher note. This is how a performer incorporates the use of *shruti* (microtones).

On the right index finger, a sitar player wears a ring like plectrum, known as a *mizrab*, shown in Figure 9(b). The right hand thumb, remains securely on the edge of the neck as shown in Figure 9(c), as the entire right hand gets pulled up and down over the main seven strings, letting the *mizrab* strum the desired melody. An upward stroke is known as *Dha* and a downward stroke is known as *Ra*. (Vir 1998)

(a) (b) (c)

Figure 9. Sitar playing technique.

5.2 The MIDI Sitar Controller

The ESitar controller captures gestural data from the performer using sensor technology, simple analog circuitry, and a microcontroller. The core of the ESitar's sensing and communication systems is an Atmel AVR ATMega16 microcontroller. The system captures data about fret number, pluck time, thumb pressure, and 3 axes of the performer's head tilt.

Pitch. Figure 10 shows, an exponentially distributed set of resistors forming a network interconnecting each fret on the ESitar in series. When the left hand fingers depress the string to touch a fret (as shown in Figure 9(a)), current flows through the string and the segment of the resistor network between the bottom and the played fret. The microcontroller digitizes the voltage drop across the in-circuit segment of the resistor network and maps that value to the corresponding fret played, using a look up table. The microcontroller transmits the calculated fret as an OSC message.

As mentioned above, a performer may pull the string downward, bending a pitch to a higher note. To handle these instances, we wrote a *pure data* (Puckette 1996) external which derives pitch using auto-correlation on a piezo pick-up signal, with a lower bound of the fret number, and an upper bound of eight semitones above that fret.

Figure 10. The network of resistors on the frets of the ESitar.

Mizrab Pluck Time. Pluck time is simply derived from two microphones which are placed on a third bridge (shown in Figure 11(a)). These condenser microphones are placed directly under the main melody string and the copper string. The signals from these microphones pass through an analog envelope detector with which we derive pluck time. We also use these two microphones to determine on which string the melody is being played. If the melody is being played on the copper string (which is very rare), the main melody string is not being played. The Atmel sends an event for pluck strike, and an event for which string is plucked via OSC event messages.

Mizrab Pluck Direction. We are able to deduce the direction of a *mizrab* stroke using a force sensing resistor (FSR), which is placed directly under the right hand thumb, as shown in Figure 11(b). As mentioned before, the thumb never moves from this position while playing, however, the applied force varies based on *mizrab* stroke direction. A *Dha* stroke (upward stroke) produces more pressure on the thumb than a *Ra* stroke (downward stroke). We send a continuous stream of data from the FSR via OSC because this data is rhythmic in time and can compositionally be used for more then just deducing pluck direction.

3 Axes Head Tilt. An accelerometer is attached to a headset in order to retrieve 3-axes of head tilt, as shown in Figure 11(c). We see the head as an easy way to control and trigger different events. (Merrill 2003) We send continuous head data via OSC messages. The headset would be a useful addition to almost all other controllers, replacing foot pedals, buttons, or knobs. It is particularly useful in this system as a sitar player's hands are always busy, and cannot use his/her feet as they are sitting. (Kapur et al. 2004)

(a) (b) (c)

Figure 11. ESitar Gesture Capturing Sensors: (a) Base of *ESitar,* (b) FSR for thumb pressure, (c) Headset with accelerometer chip

6. Discussions

Through out the four years of building these controllers our team has experimented with several sensors, microcontrollers, music protocols, sound mappings, and graphical feedback mappings. The approach for all of these areas of research has evolved as we experimented with new approaches and techniques as each new controller was built.

6.1 Sensors

We found that using FSRs on drum controllers did not obtain a quick enough response time, especially for Indian drumming, where finger strikes occur at a very quick rate. However, the gestural footprint obtained about force and position is very useful. Using piezo, we did not obtain enough information about force and position, but the response time was fast enough for live performance. Thus using a combination of the two, as seen in the *digital spoon* of the EDholak is a perfect balance, where timing specific impulses can be triggered immediately and FSR readings can effect the other variables which are not so time dependent, but still expressive.

Using simple circuitry and a few sensors as seen in the ESitar to obtain gestural data of a traditional instrument has also proved to be successful. This way a trained musician in the classical form of the instrument can easily perform with the digitized version and learn new techniques with more ease than with that of an acoustically quiet instrument, which models tradition, but relies on synthesis algorithms or samplers to generate sound.

Further, using sensors on the body, as seen on the headset of the ESitar controller, gathers extremely useful gestures which can be used to modify sound of a performer in real time.

Sensor calibration, which was also programmed for the ESitar in *pure data* was a very important addition, enabling a variety of users to easily set their minimum, maximum and average sensor values for successful mapping.

Finally, sensors are very fragile and break with use over time. Thus designing robust systems for easy exchange of sensors saves a lot of frustration and time during sound checks.

6.2 Microcontrollers

Comparing the Parallax Basic Stamp IIsx and the Atmel AVR Atmega16 as microcontrollers for these instruments, we prefer the Atmel. Programming in C as compared to PBasic (similar to Basic) allows for more complicated algorithms and onboard signal processing, which we would have not even tried to implement on a Basic Stamp IIsx. The fact that the Atmel code can be compiled using *gcc* allows development using Linux, Mac, and Windows. The eight built in 10-bit *Analog-to-Digital* converters, the faster processor speed and other magic functions built in to the Atmega16, also make it a logical choice for more complicated controllers, which use more sensors and require more processing power. (Gurevich, Verplank, Wilson 2003)

6.3 Music Protocols

Recently converting the ESitar from communicating via MIDI to OSC proved very useful for two reasons. First, we no longer had to down-sample our 10-bit *Analog-to-Digital* signals to 8-bit MIDI data streams. Second, as the developing team were at geographically different locations, using the OSC protocol that was optimized for modern networking technology made it easy to send controller data from one location (namely British Columbia, Canada to New York City, USA) to the other using the internet. (Wright, Freed, Momeni, 2003)

6.4 Sound Mapping

Correctly mapping the gestural signals obtained from the three controllers is essential in producing meaningful sound that expresses the performers feelings.

Our first experiments with the ETabla was to trigger physical models of a Tabla (Kapur et al. 2003) using STK Toolkit (Cook 2002). While we were very impressed with results of the sound and expressiveness, the time delays were too great for faster Indian rhythms. The ETabla MIDI signals also trigger sounds on any digital or analog MIDI sampler, such as the one on the R*oland Handsonic* (Roland 2004). Thus an experienced tabla player can easily use his/her years of experience and training to trigger a myriad of different sounds such as congos, djembes, bells, pot drums, and drumsets, using the ETabla. For a beginner tabla player, creating sounds such as a "*Na*" on a real tabla is very difficult and de-motivating. With the ETabla, a beginner can start creating sounds immediately giving them positive feedback to keep working on rhythm, theory, and finger strength.

One observation from the ETabla experiments was that simply triggering a single sound from a MIDI soundbank became very boring during performance. This motivated separating the rhythm making process from the sound production process in the creation of the EDholak. With this controller, one person supplies the beat while another musicians sole purpose is to modify the sounds being triggered, adjusting parameters such as the soundbank, pitch, color, depth, and pan. This proved to be very effective in performance space as deeper emotional content could now be expressed with the controller. Also, having software such as the *EDholak MIDI Control Software* to organize the order of desired soundbanks and different parameters of a sampler makes performances easier and allows the musician to focus on the music rather than memorizing where a specific patch is located.

We took a completely different approach in mapping sounds for the ESitar. The ESitar is the only instrument of the set that is not acoustically quiet when played. Thus we are modifying the sound of the real instrument, using the gestural data deduced from the Atmel chip. We use *pure data* to achieve effects such as ring modulation, delays, additive synthesis, and sympathetic pitch, which overlay on top of the acoustic sound of the sitar during performance. This method seems to fully keep the tradition of sitar performance intact while adding modified techniques using a computer. In the future, we will try to combine this method with physical modeling.

Further, with the ESitar, mapping was successful due to code written to record synchronized audio and gestural data, enabling the user to record a performance and then playback all the data, tweaking parameters to obtain desired effects and sound synthesis. Using *pure data* and other languages of this nature, which allows full control in designing sound patches that can be customized to the performer's dreams, a user can utilize far more interesting synthesis techniques that are impossible to achieve with any sampler/sequencer. With this tool we were able to achieve advanced tasks such as automatic melody seeking, which was programmed to trigger certain events when a pre-determined melody was played, or even loop a sequence of notes if it was performed three times in a row.

6.4 Graphical Feedback

We render graphical feedback for the three controllers using custom built software known as *veldt* (Davidson, Kapur, Cook 2003). *Veldt* is an application that was designed from the ground up for the purpose of visual expression and performance. It receives MIDI and OSC messages from digital musical interfaces and maps them to a system of reactive events in order to generate live visuals, which are rendered real-time using the OpenGL (Segal and Akeley 1993) graphics language. Mappings are flexible: sets of mappings may be arranged and modified during the design and rehearsal process, and triggered by control events during different movements of a performance, and arbitrary text, images, video, and geometric models may be used as source material. (Dixon 2000)

We used *veldt* to generate graphical meaning, such as rhythmic impulses from the ETabla and EDholak, and melodic transcription from the ESitar. The incorporation of visual feedback in concert performance can provide the audience with another means of conceptualizing their experience of it, and an additional means of becoming actively engaged with the artist's performance. Beyond the context of concert performance, our system can also be used in a pedagogical context, allowing a student to receive a variety of feedback and coaching while learning to play these instruments. Figure 12 shows a variety of screen shots of *veldt* being used with the ETabla, EDholak and ESitar.

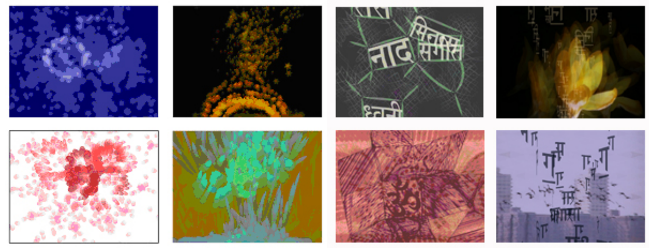


Figure 12. Screen shots of graphical feedback using veldt.

7 Conclusions

This paper presents three new controllers that are modeled after North Indian instruments: the tabla, dholak and sitar. Through our work, we try and respect the centuries of tradition passed on from *guru* (teacher) to *shishak* (student), while enabling a curious performer to experiment with new tools for expression.

In future work, we hope to use different sensors to capture even more performance gestures. We also plan to keep working on mapping the signals to physical model sound synthesizers. We would also like to see the controllers used by professionals in-order to digitally transcribe music, so that it can be analyzed with a computer. Finally, we would like to use machine learning algorithms to help analyze sensor data and predict what a performer is going to do in the future.

8 Acknowledgements

Thanks to Georg Essl, Ari Lazier, Ge Wang, and Scott Wilson for their contributions to this research. Many thanks to Rakesh Kumar Parihast and Ustad Siraj Khan for their training in North Indian classical instruments and theory. Also thanks to Chris Chafe, George Tzanetakis, Bill Verplank, Michael Gurevich, Max Mathews, Manjul Bharagava, Dan Trueman, and Tae Hong Park, for their inspiration and teachings.

References

Alesis. “Alesis D4 Reference Manual.” (2004) Available at: http://www.alesis.com/downloads/manuals/D4_Manual.pdf

Bagchee, S. (1998) *NAD: Understanding Raga Music*, Mumbai, India, Ceshwar Business Publications Inc.

Brain, M. (1998). “How Computer Mice Work” How Stuff Works.Com.

Cook, P. R. (2001) "Principles for Designing Computer Music Controllers," *ACM CHI Workshop in New Interfaces for Muiscal Expression*, Seattle.

Cook, P. R. (2002) *Real Sound Synthesis for Interactive Applications,* A.K. Peters Press.

Cook, P.R. (2003) “Perceiving our Instruments: Psychoacoustics Meets Aesthetics in the Design of New Performance Interfaces,” *Proceedings of IPEM 40,* Institute for Psychoacoustics and Electronic Music, Ghent, Belgium.

Courtney, D. R. (1995) *Fundamentals of Tabla: Complete Reference for Tabla.* Vol. 1, Houston TX, Sur Sangeet Services.

Davidson P., Kapur, A., and P.R. Cook. (2003) “A System for Generating Real Time Visual Meaning for Live Indian Drumming” *Refractory: A Journal of Entertainment Media.*

Dixon, S. (2000) “A Lightweight Multi-Agent Musical Beat Tracking System.” *Pacific Rim International Conference on Artificial Intelligence,* pp 778-788.

Gurevich, M., Verplank, B., and R.S. Wilson. (2003) “Physical Interaction Design.” *Proceedings of the International Conference on Computer Music.* Singapore.

Horenstein, M. (1996) *Microelectronic circuits and devices*. Prentice Hall, second edition.

Kapur, A.., Essl, G., Davidson, P., and P. R. Cook. (2003) The Electronic Tabla Controller. *Journal of New Music Research*. 32(4) pp. 351-360.

Kapur, A.., Lazier, A., Davidson, P., Wilson, R. S., and P. R. Cook. (2004) “The Electronic Sitar Controller” *Proceedings of the 2004 Conference on New Interfaces for Musical Expression,* Hamamatsu, Japan.

Kothari, K.S. (1968) *Indian Folk Musical Instruments,* pp. 43-44, New Delhi, India, Sangeet Natak Akademi.

Menon, R. R. (1974) *Discovering Indian Music.* Mumbai, India, Somaiya Publications PVT. LTD.

Merrill, D. (2003) “Head-Tracking for Gestural and Continuous Control of Parameterized Audio Effects,” *Proceedings of the 2003 Conference on New Instruments for Musical Expression*, Montreal, Canada.

Parallax Inc. (2004) “BASIC Stamp Programming Manual,” version 2.0.

Puckette, M. (1996) “Pure Data: Another Integrated Computer Music Environment.” *Proceedings of Second Intercollege Computer Music Concerts,* Tachikawa, Japan, pp. 37-41.

Roland. “Roland Handsonic HPD-15 Owner’s Manual.” (2004) Available at: http://www.roland.co.uk/

Segal, M. and K. Akeley. (1993) *The OpenGL TM Graphics System: A Specification*, Version 1.0, Silicon Graphics.

Sharma, S. (1997) *Comparative Study of Evolution of Music in India & the West*. New Delhi, India, Pratibha Prakashan.

Vir, R. A. (1998) *Learn to play on Sitar,* New Delhi, India, Punjab Publications.

Wilson, S., Gurevich, M., Verplank, B., and P. Stang. (2003) “Microcontrollers in Music Education - Reflections on our Switch to the Atmel AVR Platform”, *Proceedings of the 2003 Conference on New Interfaces for Musical Expression,* Montreal, Canada.

Wright, M., Freed, A., and A. Momeni. (2003) “OpenSound Control: State of the Art 2003.” *Proceedings of the 2003 Conference on New Interfaces for Musical Expression,* Montreal, Canada.