

The Electronic Sitar Controller

Ajay Kapur[✉] ▼ Ariel J. Lazier[▼] Philip Davidson[▼] R. Scott Wilson[†] Perry R. Cook^{▼,‡}
ajay@ece.uvic.ca alazier@princeton.edu philipd@princeton.edu rswilson@ccrma.stanford.edu prc@cs.princeton.edu

Department of Electrical and Computer Engineering[✉]
University of Victoria, Victoria, British Columbia, CANADA

Department of Computer Science[▼] (also Music[♫])
Princeton University, Princeton, New Jersey, U.S.A.

Center for Computer Research in Music and Acoustics[†]
Stanford University, Stanford, California, U.S.A.

ABSTRACT

This paper describes the design of an Electronic Sitar controller, a digitally modified version of *Saraswati's* (the Hindu Goddess of Music) 19-stringed, pumpkin shelled, traditional North Indian instrument. The *ESitar* uses sensor technology to extract gestural information from a performer, deducing music information such as pitch, pluck timing, thumb pressure, and 3-axis of head tilt to trigger real-time sounds and graphics. It allows for a variety of traditional sitar technique as well as new performance methods. Graphical feedback allows for artistical display and pedagogical feedback. The *ESitar* uses a programmable Atmel microprocessor which outputs control messages via a standard MIDI jack.

Keywords

Electronic Sitar, ESitar, Indian String Controller, MIDI Controller, Human Computer Interface, Atmel Microcontroller, *Veldt*, Instrument Graphical Feedback.

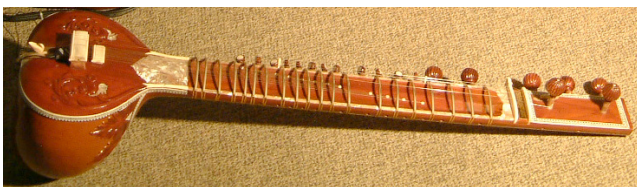


Figure 1. Picture showing traditional Sitar.

1. INTRODUCTION

The sitar is *Saraswati's* (the Hindu Goddess of Music) 19-stringed, pumpkin shelled, traditional North Indian instrument. Its bulbous gourd (shown in Figure 1), cut flat on the top, is joined to a long necked hollowed concave stem which stretches three feet long and three inches wide. The sitar contains seven strings on the upper bridge, and twelve sympathetic strings below, all tuned by tuning pegs. The upper strings include rhythm and drone strings, known as *chikari*. Melodies, which are primarily performed on the upper-most string and occasionally the second string, induce sympathetic

resonances in the twelve strings below. The sitar can have up to 22 moveable frets, tuned to the notes of a *Raga* (the melodic mode, scale, order, and rules of a particular piece of Indian classical music). The sitar is a very sophisticated and subtle instrument, which can create vocal effects with incredible depths of feeling, making it a challenging digital controller to create. [1][2]

Our goal is to use sensor technology to extract musical information from a performing sitar player, deducing gestural information such as pitch, thumb pressure, and 3-axes of head tilt to trigger real-time sounds and graphics. In this paper, we will present:

- The evolution of the technology of the sitar from its origins until the present day.
- The traditional playing style of the sitar, on which the controller is modeled.
- The creation of a real-time MIDI sitar controller, using the Atmel microcontroller.
- The techniques used to map the signals captured from the sitar to sound.
- The creation of a real-time graphical feedback system that reacts to the sitar controller.
- The description of the sitar controller used in live performance.

2. EVOLUTION OF THE SITAR

The precursor of the sitar is known as the *vina*, of the lute family of instruments, which is referenced in Vedic writings as early as the first millennium B.C. Figure 2 (a) shows an early version of the *stick zither vina*, from the 6th and 7th century A.D. From this picture it is evident that the *stick zither* did not have frets, which ancient sculptures suggest evolved in the 10th and 11th Century A.D.[3] Figure 2 (b) shows a primitive type of *vina* instrument whose neck is made out of bamboo. [4]

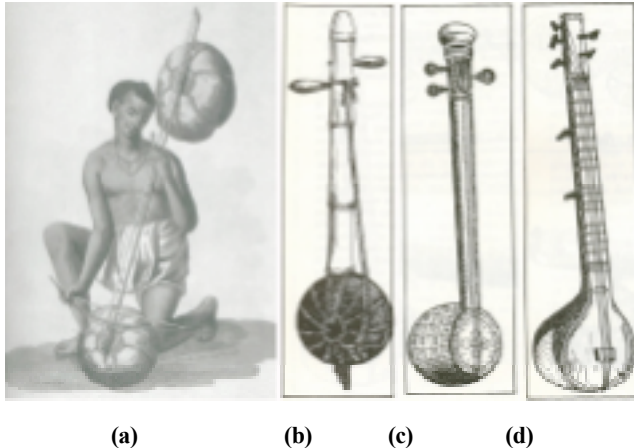


Figure 2. (a) A stick zither vina [3], (b) A vina made of bamboo [4], (c) A sehtar [4], (d) A 7-stringed sitar [4]

There exist several differing historical accounts of the sitar's evolution. Some sources claim the instrument descended directly from the *vina* as performers and builders made small modifications over time as technology and tools evolved. Others claim the similarity between the Middle-Eastern *tambur* and the Persian *sehtar*, which traveled to India during the Muslim occupation of India in 11th century. The name seems to have derived from the Persian *sehtar* (*she*- three, *tar* – strings) shown in Figure 2 (c). In the 18th century, instrumentalist Amir Khusro is credited with adapting the name, as well as reversing the order of the strings, placing the main melody string to the far outside, thus making it easier for the performer to play with the instrument upright. [3] He also improved the sitar by making the frets movable (for fine tuning), by using string to tie the frets down. [4]

In the 18th century, after the innovation of creating a wider bridge, four more strings were added to the sitar, giving a total of seven strings (as seen in Figure 2 (d)). Other improvements include the introduction of metal frets and the *mizrab*, the pyramid-shaped, wire plectrum. In the 19th century, the *tarabdar* style of sitar emerged, which had 9 to 13 sympathetic strings (known as *tarab*) positioned under the frets, as depicted in Figure 1.[3]

In 2003, our team has progressed the sitar to the modern era of computers, adding resistors, capacitors, force sensing resistors, microphones, and ethernet jacks, to enhance traditional technique with the use of a laptop.

3. SITAR TECHNIQUE

It is important to understand the traditional playing style of the sitar to comprehend how our controller captures its hand gestures. In this section, we will define the different parts of the sitar, briefly explain how Indians annotate melodic notes, and describe the basic technique of sitar playing.

3.1 Construction of a Sitar

The gourd section of the sitar is known as the *tumba* and plays the role of a resonating chamber. The flat piece of wood which lays on the front side of the *tumba* is known as the *tabli*. The long column which extends from the *tumba* is known as the *dand* (similar to the neck of a guitar), and is made out of the same material as the *tabli*. This part of the instrument acts as a column resonator. Sometimes, a second *tumba* is put at the *dand* to increase resonance.

The seven main upper strings run along the *dand*, above moveable, curved metal frets, over a bridge (*jawari*) made of ivory or deer horn, and tie together at the *langot*, at the very bottom of the sitar. The sympathetic strings, or *taraf* strings, run below the frets, and have their own separate bridge (*ara*), but still tie together at the *langot*. All strings are made of steel, except for the second upper string (right next to the main melody string), which is made of copper.

3.2 Hindustani Note Annotation

There are seven main notes (*swara*) in Hindustani music: *Shadja* (*Sa*), *Rishab* (*Re*), *Gandhar* (*Ga*), *Madhyam* (*Ma*), *Pancham* (*Pa*), *Dhaivat* (*Dha*), and *Nishad* (*Ni*). These seven notes correspond directly with the western *Do, Re, Mi, Fa, So, La, Ti*, and represent the seven notes in a major scale. Octaves are indicated with a mark as follows: *Sa_o* (lower octave), *Sa* (middle octave), *Sa^o* (upper octave). *Re, Ga, Dha*, and *Ni*, have notes which are half step below, known as *komal*, and are represented as follows: *Re, Ga, Dha, Ni*. The note in between *Ma* and *Pa*, (known as a tri-tone in Western music) is known as *tivra Ma*. There are more notes in between these twelve basic notes, known as *shruti*, which translate to micro-tones. These notes are traversed when transitioning from one main note to another. There is no agreed-upon method of transcribing *shruti*, and proper technique can only be learned in a traditional *guru* (master) to *shishak* (student) tutoring. [2][3][4]

3.3 Sitar Playing Technique

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.[3] The ESitar is modeled based on the Vilayat Khan system.

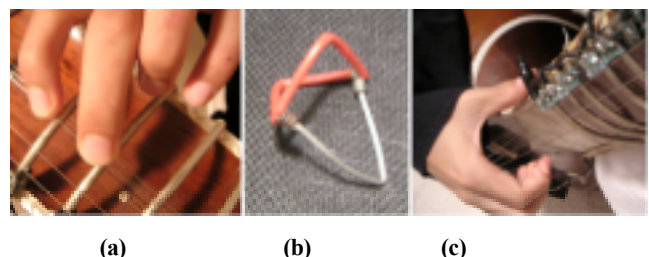


Figure 3. Pictures showing sitar playing technique.

We will only discuss the basic playing techniques of the sitar. 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 3(a), to press the string to the fret for the desired *swara*. 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 (typically around *Sa* and *Pa*). 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 3(b). The right hand thumb, remains securely on the edge of the *dand* as shown on Figure 5(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*. [2][3]

4. THE MIDI SITAR CONTROLLER

The *ESitar* controller captures gestural data from the performer using sensor technology, simple analog circuitry, and a microcontroller.

4.1 The Atmel Microprocessor

The core of the *ESitar's* sensing and communication systems is an Atmel AVR ATMegal6 microcontroller. The low-cost 8-bit microcontroller has eight built-in 10-bit analog to digital converters (ADCs) as well as 24 general purpose digital I/O pins. The microcontroller is housed on an AVRmini development board to provide clocking, serial ports, connector access to the chip, power regulation and programming circuit. The microcontroller program, which reads ADC inputs and transmits MIDI messages, is written in C with the help of the Procyon AVRlib support library. [5]



Figure 4. Pictures showing Controller Box

The Atmel is encased in a controller box as shown in Figure 4, with a number encoders, switches, and knobs, used to trigger events, toggle between modes, and fine tune settings. The sitar and headset are each connected to the main control box via ethernet cable, enabling analog control signals to pass from one device to another with a robust connector feature. The box also has an LCD to display controller data and settings to the performer, enabling him/her to be completely detached from the laptops running sound and graphic simulations.



Figure 5. Pictures showing the network of resistors on the frets of the *ESitar*.

4.2 Gesture Capturing

Our system captures data about fret number, pluck time, thumb pressure, and 3 axes of the performer's head tilt.

4.2.1 Pitch

Figure 5 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 3(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 midi note number. The microcontroller transmits the calculated fret as a MIDI message.

As mentioned above, a performer may pull the string downward, bending a pitch to a higher note (for example play a *Pa* from the *Ga* fret). To handle these instances, we wrote a *pure data*[6] 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 five half steps above that fret.

4.2.2 Mizrab Pluck Time

Pluck time is simply derived from two microphones which are placed on a third bridge above the *ara* (shown in Figure 6(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 MIDI event messages.

4.2.3 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 6(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 MIDI, because this data is rhythmically in time and can compositionally be used for more than just deducing pluck direction.

4.2.4 3-Axis Head Tilt

An accelerometer is attached to a headset in order to retrieve 3-axes of head tilt, as shown in Figure 6(c). We see the head as an easy way to control and trigger different events.[7] We send continuous head data via MIDI 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.

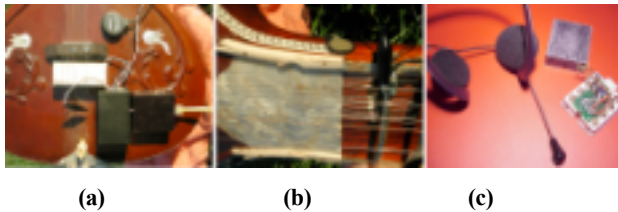


Figure 6. Figure showing Pictures of Gesture Capturing Sensors. (a) Base of *ESitar*, (b) FSR for thumb pressure, (c) Headset with accelerometer chip

5. MAPPING SIGNALS TO SOUND

Once we have control information describing pitch, pluck, pluck direction/power, and head tilt, we must decide how to use this information to create sound.

5.1 Requirements

The *ESitar* produces the natural sound of the instrument as well as control information. Thus, we must take care that the sounds we produce complement the acoustic sound of a classical sitar. This task is possible because of the detailed information available about what is being played. There are two classes of electronic sounds we can produce. The first consists of sounds that function as an extension of an acoustic sitar. One means of extending the sound of the Sitar is to simply process the signal coming through a pickup with standard effects such as delay and reverb. This is not really interesting because there is no need for the detailed information regarding the musician's interaction with their instrument to apply these effects. With very accurate information to describe the actions of the Sitar player it is possible to create effects that reach beyond basic processing of the acoustic sitar sound. When attempting to extend the sound of the Sitar, we must take care that the mapping of control information to sound is done in such a way that the performer can control both the acoustic sound and synthesized sound simultaneously. Such mappings can be considered a coupling of the gestures used to create the acoustic and electronic sounds. Recent research has shown that such mappings have the potential to be more expressive and powerful than one to one mappings between control information and synthesis parameters [8]. The second class of electronic sounds we can produce consists of sounds that function as counterpoint to the sound of the sitar. This class of sounds focuses more on the analysis of control information and the creation of a computerized accompaniment or response, rather than mapping control parameters directly to sound. To interact with such a system, the performer does not try to trigger control information, but instead varies their interaction with the acoustical instrument to implicitly direct the computer-generated sounds. For example, with the detailed information

available, it is possible to generate tabla beats that correctly complement and react to the acoustic sound of the sitar.

5.2 Pure Data Patches

All of the signals generated by the *ESitar* are sent into a computer and captured by Miller Puckette's *pure data* (*pd*) program. This information could be used in many other ways, but we chose to use *pd* because of the ease of capturing and manipulating the control information within the *pd* environment. Below, we describe a number of different *pure data* patches written to demonstrate how such control information can be used.

5.2.1 Slide Sitar

The slide sitar patch was modeled after the sound of a slide guitar. It is a simple module that consists of a bank of oscillating comb filters. The filters have a resonance that is determined by the frequency indicated by the fret information. The control information is used to change the resonance of the filters. We also use thumb pressure to control the amplitude of the resonance oscillation. Here we interpret thumb pressure as a measure of intensity of the performance. The intensity of each pluck can be heard through the amplitude of the filters' oscillations. This is a very simple use of the control information, but such an effect could not be obtained without the detailed information provided by the *ESitar*.

5.2.2 Sympathetic pitch

The sympathetic pitch patch plays rolls of a sampled sitar sound at the pitch indicated by the fret. In this case we mapped thumb pressure to the volume of the roll and head tilt to length of the notes and speed of the roll. Because the sounds produced do not directly correspond to the acoustic sound, this is more of a complementary patch. This is apparent when the performer increases the pressure on the thumb FSR on a beat where the strings are not strummed. In these instances the roll becomes more prominent and can function as a rhythmic replacement for the sitar when the acoustic sound is not heard.

5.2.3 Ring Synthesis and Delay

This patch also takes into account the fret information, setting the frequency of the modulation to the pitch indicated by the frets. This patch produces a distorted sound that would not be possible to create without the accurate pitch information provided by the controller. We also set up a delay system controlled by the head and the thumb. The system allows the musician to capture parts of the output of the ring synthesis module into a loop for later playback and combination with other loops. The head tilt controls which loops are played or filled with new sound, while the thumb controls if or how much sound should be played or stored into the loop buffers.

5.2.4 Analysis/Re-Synthesis

Our last example is a simple analysis/re-synthesis patch. We use fret information and pitch information to identify which partials are generated by the struck note. Simply taking the strongest partials would not lead to the same results because of the sympathetic strings that are constantly resonating. Once the strongest partials are chosen, sine waves with very short envelopes are synthesized which together form a disjointed representation of the acoustic sitar sound. The volume of the re-synthesis is scaled to the sound of the acoustic sitar and then controlled by head tilt. We warp the volume distribution of the detected partials using thumb pressure. When greater

thumb pressure is applied, the lower partials are given more weight and are more easily discernable.

5.2.5 Drone

We also added the capability to control the sound of a drone using head tilt. The drone is created using a bank of fm synthesizers. Each of the synths are tuned to A or G, each with a different set of harmonics. They are also specialized so that head tilt controls the relative loudness of each of the synths. This gives the performer the ability to control which synths are most prominent.

6. GRAPHIC FEEDBACK

We rendered the visuals for the *ESitar* performance using *veldt*[9], as our visual design environment, and we chose to model our visualization on the (name for sitar tablature), the traditional form of melodic tablature for sitar. As the player performs, we read the incoming note/velocity pairs from the Atmel chip to render a stream of swara, which are arranged in a helix as if they are printed on spinning drum of paper. We use a discrete rhythm detection algorithm[10] over a recent history of notes played to estimate a rough beat-per-minute value, which modulates the speed at which the drum rotates so that one measure is displayed per rotation of the drum. Notes played with greater intensity are rendered in a larger, bolder style, emphasizing them within the overall display. Rhythmic patterns are reflected visually as symmetries around the drum.

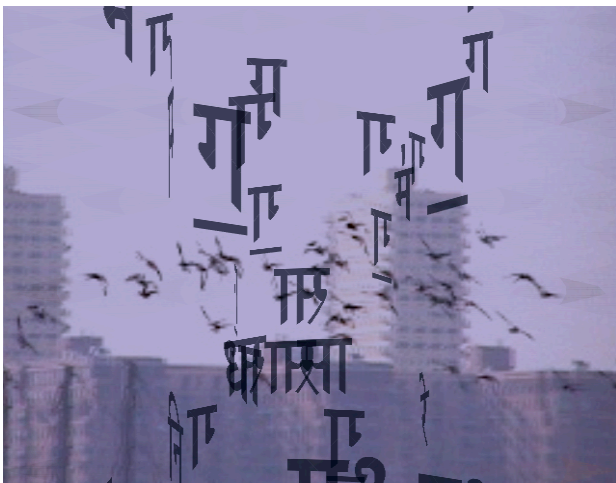


Figure 7. Roll of swara rendered over video stream

As in our audio synthesis methods, we incorporate additional signals measured in the performance to broaden the player's ability to change the scene. We monitor the signals received from two of the three axes from the tilt accelerometers on the headset, as well as the pressure measured from the thumb of the plucking hand, in order to pick up both continuous, low frequency measurements and additional rhythmic cues. The accelerometer values are used to parameterize the blending of several video streams over geometry in background of the video, in correlation with the movement through parameter spaces in our audio synthesis model. The thumb pressure provides a frequent, pulsing measurement in coordination with the plucks of the *ESitar*, and is used to control the

scrubbing of short animation clips – in this example, those of a flower opening and closing.

There are a variety of mappings that could be conceived of with the new range of the measurements that are received from the *ESitar*. In a pedagogical setting, a pre-recorded reel of *swara* could be used as a score against which a student's accuracy could be measured visually, while also drawing their attention to more subtle aspects of their performance.



Figure 8. Animation scrubbing from thumb pressure

7. ESITAR IN LIVE PERFORMANCE

The *ESitar* was premiered at the Listening in the Sound Kitchen Computer Music festival in November of 2003, shown in figure 9. An 8-channel composition “Saraswati's Electro-Magic” was performed. We presented the four different settings, each with different mappings using *pure data* and different visual feedback effects using *veldt* setting discussed above. A performance was our final test that the controller was actually working, and not just some lab experiment to theorize and hypothesize about.

8. CONCLUSION

We have presented a real-time device for sitar performance. The sitar controller captures gestural data from a performer, and uses it to manipulate sounds and visuals. A performer can now use a laptop with a sitar in order to create a multimedia experience for a live audience, using traditional Indian classical sitar technique. Performance based visual feedback forms another means of expression for the performer, as well a pedagogical tool. In the future we plan to use the *ESitar* as a tool to transcribe Indian Classical music.

9. ACKNOWLEDGMENTS

We would like to thank Bill Verplank, Michael Gurevich, Scott Wilson, and Max Mathews for their workshop at CCRMA on controllers and the Atmel microprocessor. We would also like that Asha Kapur, Dolly and Surinder Vohra for bringing a sitar from India to build the *ESitar* with. We would also like to thank Ustad Siraj Khan of the Mewati gharana for his training in the traditional classical theory and technique of sitar performance. Other thanks to Andrew Schloss, Peter Driessen,



Figure 9. Picture showing ESitar used in live performance.

George Tzanetakis, Tae Hong Park, Dan Trueman, Curtis Bahn, Pavan Vohra, Sona Vohra, Meera Kapur, and Arun Kapur, for their support and inspiration.

10. REFERENCES

- [1] Menon, R. R. *Discovering Indian Music*. Somaiya Publications PVT. LTD, Mumbai, India, 1974.
- [2] Vir, R. A. *Learn to play on Sitar*. Punjab Publications, New Delhi, India, 1998.
- [3] Bagchee, S., *NAD: Understanding Raga Music* Ceshwar Business Publications Inc, Mumbai, India, 1998.
- [4] Sharma, S. *Comparative Study of Evolution of Music in India & the West*. Pratibha Prakashan, New Delhi, India, 1997.
- [5] Wilson, S., Gurevich, M., Verplank, B., & P. Stang, "Microcontrollers in Music Education - Reflections on our Switch to the Atmel AVR Platform", *Proceedings of the 2003 Conference on New Instruments for Musical Expression (NIME 2003)*, Montreal, Canada. May 22-24, 2003.
- [6] Puckette, M. "Pure Data: Another Integrated Computer Music Environment." *Proceedings of Second Intercollege Computer Music Concerts*, Tachikawa, Japan, pp. 37-41. 1996.
- [7] Merrill, D., "Head-Tracking for Gestural and Continuous Control of Parameterized Audio Effects," *Proceedings of the 2003 Conference on New Instruments for Musical Expression (NIME 03)*, Montreal, Canada. May 22-24, 2003.
- [8] Hunt, A, Wanderley, M. M., & M. Paradis. "The Importance of Parameter Mapping in Electronic Instrument Design." *Proceedings of the 2002 Conference on New Instruments for Musical Expression (NIME 02)*, Dublin, Ireland. May 24-26, 2002.
- [9] Kapur, A., Essl, G., Davidson, P., & P.R.Cook. "The Electronic Tabla Controller" *Journal of New Music Research*. 32(4), pp.351-360. 2003.
- [10] Dixon, S. "A Lightweight Multi-Agent Musical Beat Tracking System." *Pacific Rim International Conference on Artificial Intelligence*, pp 778-788. 2000.