Multi Input Sensing Table

MIST

By

James McCloskey

Advisers

Professor Virginia de Sa, Professor Ryan Kastner, David Medine

Collaborators

Roger Jennings, Danny Anderson, Alric Althoff

Multi-Input Sensing Table: A Sensor Fusion for Drumming

James McCloskey*, Danny Anderson*, Roger Jennings*, David Medine*
University of California, San Diego
San Diego, CA 92093, USA
jmcclosk@ucsd.edu, d8anders@ucsd.edu, roger.0718@gmail.com, dmedine@ucsd.edu

ABSTRACT

Optical sensors are the most common method for sensing touch input on multi-touch tables. This project's approach was to build a large multi-input sensing table with multiple sensors to collect optical, pressure, and acoustic data. The data was aggregated and processed to create a computer music application using the table's sensors. This application demonstrates the utility of accurate temporal and spatial resolution data and showcases the novelty of the multi-input sensing table.

ACM Classification Keywords

H5.2 [Information interfaces and presentation]: User Interfaces.—Interaction styles.

General Terms

Design, Human Factors

Author Keywords

Interactive Tabletops, Pressure, Multi-touch, Sensing, FTIR, Acoustics

INTRODUCTION

Multi-touch tables have been a hobbyist reality for years, with various methods of display and input control. Most input methods revolve around the use of frustrated total internal reflection (FTIR) [1]. Infrared light beamed into the acrylic surface of the table from all sides will mostly reflect inward off the top and bottom edges of the surface, called total internal reflection. When a finger touches the screen, the infrared light bounces downward to an infrared camera. Other methods use lasers or diffuse infrared light, but FTIR is an economical and viable method. Capacitive touch screens, such as the touch screen in most smart phones, are not cost-effective at a large scale, with even a small 22" capacitive multi-touch display sold by 3M for $1,500 [2]. Using the FTIR method, the cost of the input method is only for the acrylic surface, infrared light-emitting diodes, and a camera.

In the multi-input sensing table (MIST) explored here, we go beyond the typical multi-touch table approach which uses only a camera for input, by adding pressure (resistive) and acoustic (piezoelectric) sensors to the table. The goal of this project is to show the utility of the additional sensors. Although many applications have been written using multi-touch input [3], no applications were found which have been written to use a multi-touch input in addition to pressure and acoustic information. This project was to create a drumming application user interface, which combines the multiple sensor sources. The drum pad application, Bapp, showcases the ability of the multi-input sensing table (Fig 1). The acoustic sensors can locate a touch on the screen at much higher frequency than the FTIR approach. The pressure sensors can relay audio volume information to the drumming application. The FTIR is used for overall setup in the application. This multiple sensor fusion results in a novel human computer interaction, which relays more information on every touch of the surface.

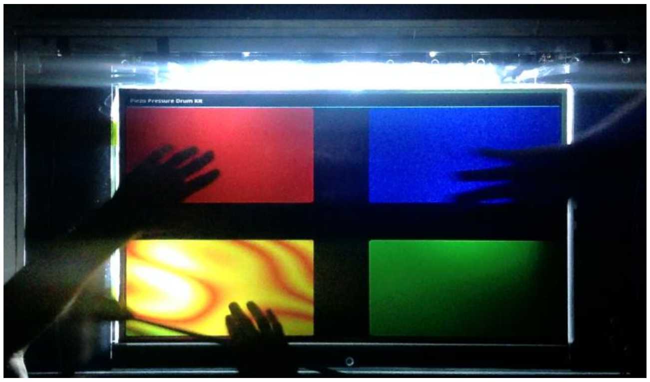

Figure 1: Bapp drum pad

RELATED WORK

The work presented in this paper builds on previous multi-touch tables that incorporated piezoelectric acoustic sensors and resistive pressure sensors.

Recently we have seen the emergence of table top displays incorporating acoustic sensors. Lopes et al. demonstrate acoustically enhanced touch sensing [9]. They identify specific audio signatures to combine sound and touch data in order to eliminate false positives in optical multi-touch setups. Harrison et al. demonstrated an acoustic based input classification approach called TapSense [9]. They trained a support vector machine to classify different types of finger touch inputs including pad, tip, knuckle and nail taps.

These acoustic sensors that are used on table top displays are typically piezoelectric sensors. Tindale et al. give an overview of nine different sensor strategies currently employed for developing a percussion interface [11]. They

ITS'12, November 11–14, 2012, Cambridge, Massachusetts, USA.

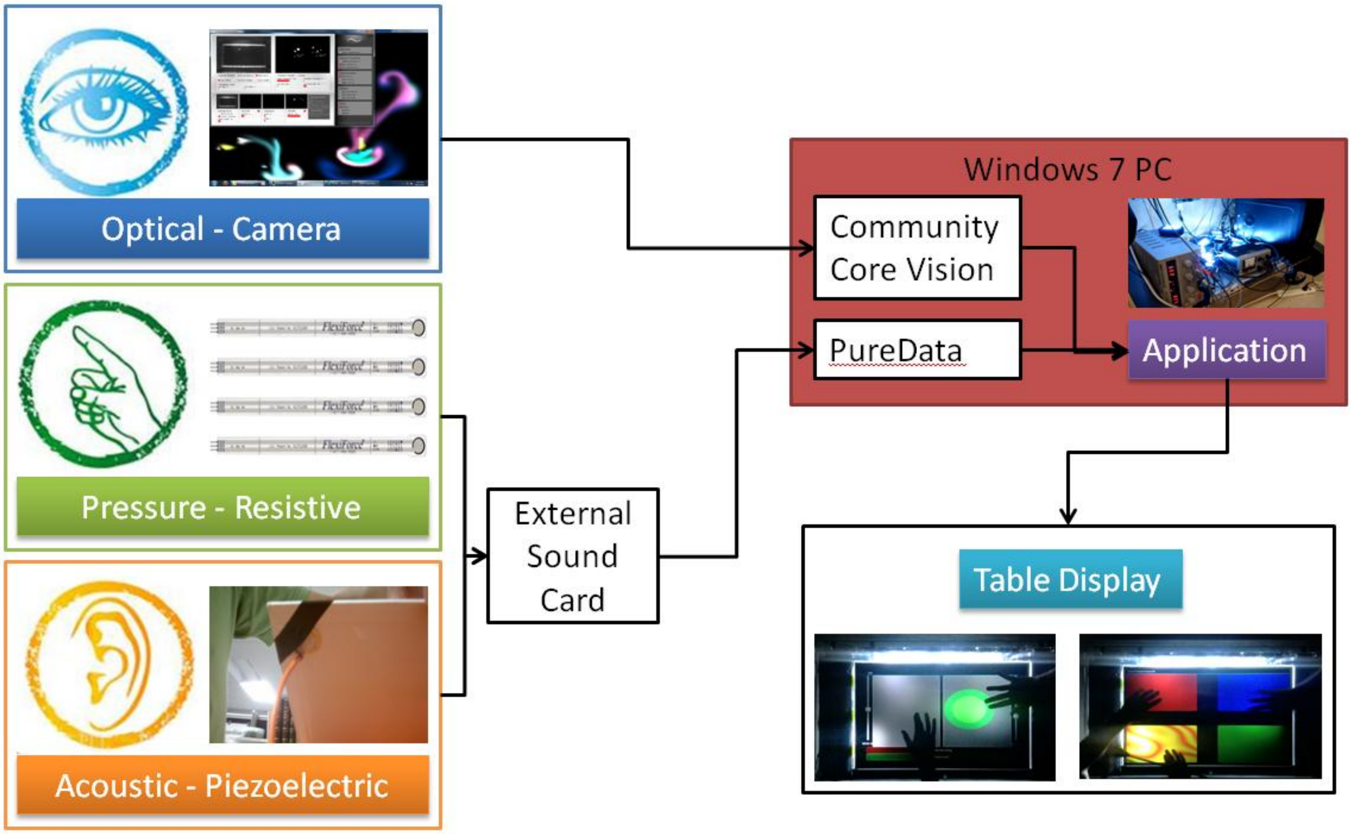

Figure 2: Multi-Input Sensing Table Overview

identify the bandwidth of a percussionist as being a major consideration when designing a drum surface. They cite that the world record for the fastest single stroke roll is currently 1199 strokes per minute which translates to approximately 20 Hertz. Hence the sampling rate must be high enough to capture each drum onset.

Resistive pressure sensors detect and measure a relative change in force or applied load. Heo and Lee demonstrated the feasibility of force gestures for small touch screen devices [12]. These force gestures also took advantage of a multi sensor array. Mandalapu and Subramanian demonstrated single sensor pressure based zooming and showed how pressure sensors can be bi-directional and not just unidirectional.

MULTI-INPUT SENSING TABLE

The multi-touch table backlight array emits visible light from underneath the table and illuminates an LCD screen display. The top surface of the table is ½” thick acrylic that is surrounded by one hundred twenty-eight infrared light-emitting diodes, pointing inward. The infrared light bounces around inside this acrylic with minimal loss; however, when an item contacts the acrylic surface from above (i.e. a finger), the infrared bounces off this object and downward. A camera with an infrared filter is placed below the surface to detect the refracted light when an object touches the surface. The camera is a FireFly MV, capturing video of 640x480 pixels at 60 frames per second.

The set up discussed above is standard for many multi-touch tables, but the MIST table adds four pressure (resistive) sensors and four acoustic (piezoelectric) sensors mounted at the corners of the 1m by 0.5m surface (Fig 2). The piezoelectric sensors are originally purposed as simple speakers (i.e. a microwave beep). A standard lightweight resistive sensor is used. The resistive sensor works by transmitting a change in potential as its material is deformed under pressure. These eight sensors feed into an external soundcard, the AudioFire Echo 12.

Beyond this custom hardware, a typical desktop computer interprets the sensor data, processes the input to drive software, and sends the video signal to the table’s display. Community Core Vision (CCV) [4], an open source computer vision project, parses the camera data, and then sends touch events using the Tangible User Interface Objects protocol. The TUIO packets are sent using the User Datagram Protocol to the local host on a determined port. A Window 7 Quad-Core PC runs CCV and controls the display. The PureData [5] program (PD) parses the other sensors and sends UDP packets to the local host on other determined ports. The application uses the Kivy [6] Python framework to take these three UDP ports as input to create a drum pad application, which is displayed on the table's LCD screen.

Piezoelectric Sensors

The four piezoelectric sensors are placed in the corners of the table, past the viewable area of the display. The piezoelectric sensors detect incident sound waves, which run through the inside of the acrylic surface following a tap of the screen (Fig 3). The output of the four sensors feeds into the high-bandpass filtering, external soundcard. The Echo 12 sends this data via firewire to the controlling PC.

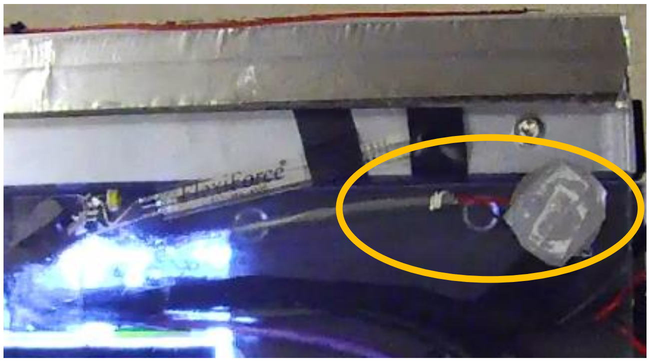

Figure 3: Piezoelectric sensor attached to the surface

Theoretical Analysis

The following is an analysis of how well these sensors can locate a touch on the table given that the sample rate of the piezoelectric sensors is 44.1 KHz. The speed of sound within Acrylic is 2,870 m/s [7]. The table measures 1m wide by 0.5m tall, which allows solving for the distance of sample granularity using the formula relating distance, speed, and time.

$$\text{Distance} = \text{Velocity} * \text{Time}$$

$$0.065\ (\text{m}) = 2{,}870\ (\text{m/s}) * 1 / 44100\ (\text{s})$$

Thus, the smallest distance that can be detected to differentiate two touches is 6.5 cm. Stated differently, any two touches within a 6.5 cm concentric ring around a sensor will be detected in the same sampling interval. Considering the ability of the sensors to detect a touch on either side of the table, a central divider of 8 cm should be sufficient to split the table into two pads with adequate accuracy. Hence two dividers of 8 cm can be added running horizontally and vertically through the center of the table. Thus, detecting four quadrants using the acoustic sensors is feasible. Each sensor feeds into a wave onset detection algorithm and transmits an identifying signal when it detects an incident wave. To avoid sending a signal twice for one tap, after sending a packet these sensors should avoid sending a new packet until the wave has passed all sensors. The length of the timeout can be calculated using the table's diagonal length of 1.118 m.

$$\text{Time} = \text{Distance} / \text{Velocity}$$

$$0.00039\ (\text{s}) = 1.118\ (\text{m}) / (2{,}870\ [\text{m/s}])$$

As the number of detectable regions increases past four regions, the computation becomes more complex and unreliable. Using many sensors could be used to triangulate a touch, but using multiple sensor detections of a wave to locate a touch would introduce extra latency in order to wait for the touch to be detected by all sensors. Using four quadrants is a simple algorithm where the first sensor to detect a wave locates the wave as being within its quadrant.

Practical Analysis

The four quadrant detection using piezoelectric sensors can well discriminate between the four regions when the table is struck. The dividing zones between the four regions have a more ambiguous result when struck, as predicted. Using two regions gives more accurate detection, as each sensor operates differently.

Some non-uniform properties of the sensors can be compensated by software. Three of the piezoelectric sensors are of the same origin, but the fourth is using slightly different technology. Also, the power circuit for the fourth sensor is on another board than the other three sensors. Furthermore, the sensors are not placed exactly in the same places in the four corners of the table. However, within the PureData processing of the sensor data, some tunable threshold values have been experimentally optimized to improve the definition of the four quadrants. Unfortunately, any appreciable latency differences among the sensors cannot be adjusted within our PureData processing. An alternative method which uses the amplitude of the incident waves would likely not be a suitable replacement to adjust for diverse latencies, as the sensors do not display uniform reactivity to a tap of the table.

The tunable values within PureData include the minimum incident wave peak for onset detection, as well as the minimum velocity of the wave. The process for tuning these values is to tap in the center of a region and observe which region was reported having been touched. If the region reported is incorrect, the incorrect region's tunable values are raised. This tuning greatly improved the accuracy of quadrant detection, but using two drum pads has a more reliable result. However, at times the piezoelectric sensors can be unresponsive to tuning for unknown reasons.

Among more complicated models that could be used to model the location of a touch, a concentric moiré pattern model would result in finer granularity of localization, along with increased computational complexity and latency. Another method would be to model the reflected sound waves at the sides of the acrylic surface. When drumming fast, this effect will sometimes result in an incorrect detection of another region. A machine learning technique could also be used for finer granularity of touch localization using the piezoelectric sensors in further work.

Pressure Sensors

Mounted underneath the four corners of the acrylic surface of the table are resistive strain sensors. The acrylic is resting directly on these sensors and nothing else, so any pressure applied to the surface goes directly through these elements (Fig 4). The resistive effect relies on a changing electromagnetic potential through a deforming material when pressure is applied. The pressure sensor reports an absolute value for pressure.

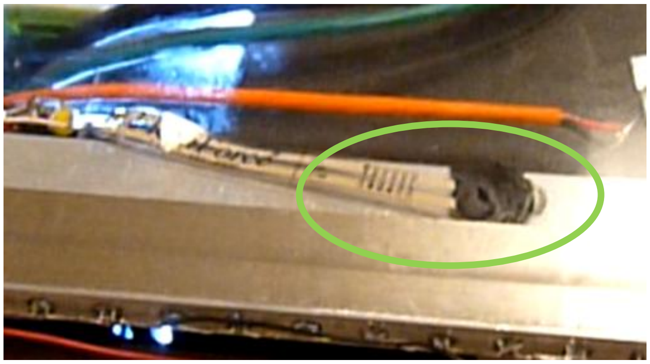

Figure 4: Pressure sensor attached to the surface

Theoretical Analysis

The pressure sensors and related hardware on the table have some shortcomings. The stressed elastic material in the pressure sensor takes time to return to its initial shape after a touch. This fact makes the set of piezoelectric sensors more suited for locating quick touches than the resistive sensors. Another drawback is that the pressure sensor data feeds into the computer via an AC audio bridge. Since the pressure sensors report a magnitude, which is a slowly changing value of pressure, this AC path with a high frequency filter greatly reduces the utility of the sensor.

The pressure sensor will not give the magnitude of applied pressure because of this AC path to the PC, but rather the derivative of the pressure value. In the ideal set up, upon touch detection, the pressure sensors would supply a concrete value for how much pressure is applied at that touch. However, with the current set-up, the sensor provides a value for how quickly the pressure changed. This can be used as a rough estimate of the pressure value, but the usefulness of the sensor is severely hampered by the AC interpretation of the signal. To greatly increase the utility of the pressure data, a new DC interface should complement the current hardware.

Practical Analysis

PureData is used once again for interpreting the data of the pressure sensors. When the onset of a wave is detected in the aggregated pressure sensor data, the height of the peak of the wave is transmitted in a UDP packet to the application. Then this transmission times out for a brief tunable interval, as the release of the touch may appear as another wave onset. Unfortunately, the AC audio interface does not allow reliable detection of when the surface is held down, as this is an absolute magnitude signal; however, the above method does result in some differentiation between harder and softer taps of the screen using padded drumsticks.

At times the pressure data can be unpredictable. Occasionally a very hard tap of the surface will relay the same value as a light tap. Overall this method gives a rough estimate of the impact strength, but can be greatly improved with a change of hardware.

UI Frameworks

Although the multi-touch table project has run a variety of sample programs found online, custom development has relied upon two approaches. The first approach is using OpenFrameworks' C++ API, and the second approach was a PureData program. The C++ approach had been explored more fully and would have been the natural choice, but many other frameworks were compared before diving into extensive development.

TUIO.org provides references to many multi-touch frameworks, among which the PyMT project seems most concise. In a prior course in Programming Languages, Python has shown its brevity, ease of expression, and large development community, which has supplied many open source Python bindings for powerful tools. The major advantage to using PyMT is its amazingly concise programs: 115 lines of code for a Google Earth Multi-Touch interface, and 116 lines for a Mandelbrot Fractal Explorer.

Although PyMT was referenced from TUIO.org, the Kivy framework has replaced the PyMT project. Kivy has matured far past PyMT, and even includes cross-platform support for iOS, Android, OSX, Windows, and Linux. This portability is a huge advantage as the development machines run OSX while the table's controlling PC is running Windows. Given that Python is known for being cross-platform and various cross-platform audio libraries exist, Kivy was chosen over OpenFrameworks. Although OpenFrameworks' API can be used for development on OSX, Windows, Linux, iOS, and Android, the deployment to each device requires code changes and recompilation. Another advantage of Kivy is its abstraction of UI styling into another ".kv" file (kv is to py as CSS is to HTML). This further increases the brevity of Kivy code, and is syntactically convenient.

Audio Libraries

Various audio libraries purport to be cross-platform, but upon using them, a lofty barrier of dependencies arises. Audio libraries we analyze for their ease of use and cross-platform appeal include athenaCL, Csound, improviser, pyao, pyFluidSynth, pygame, SDL_mixer, and pyo [8]. Among these choices, Csound and pyo have the lowest barrier for entry in terms of dependencies required. pyo was the initial decision based on code brevity, which was the main point in Kivy's favor as well. Although typically concise code and high-level abstractions result in a

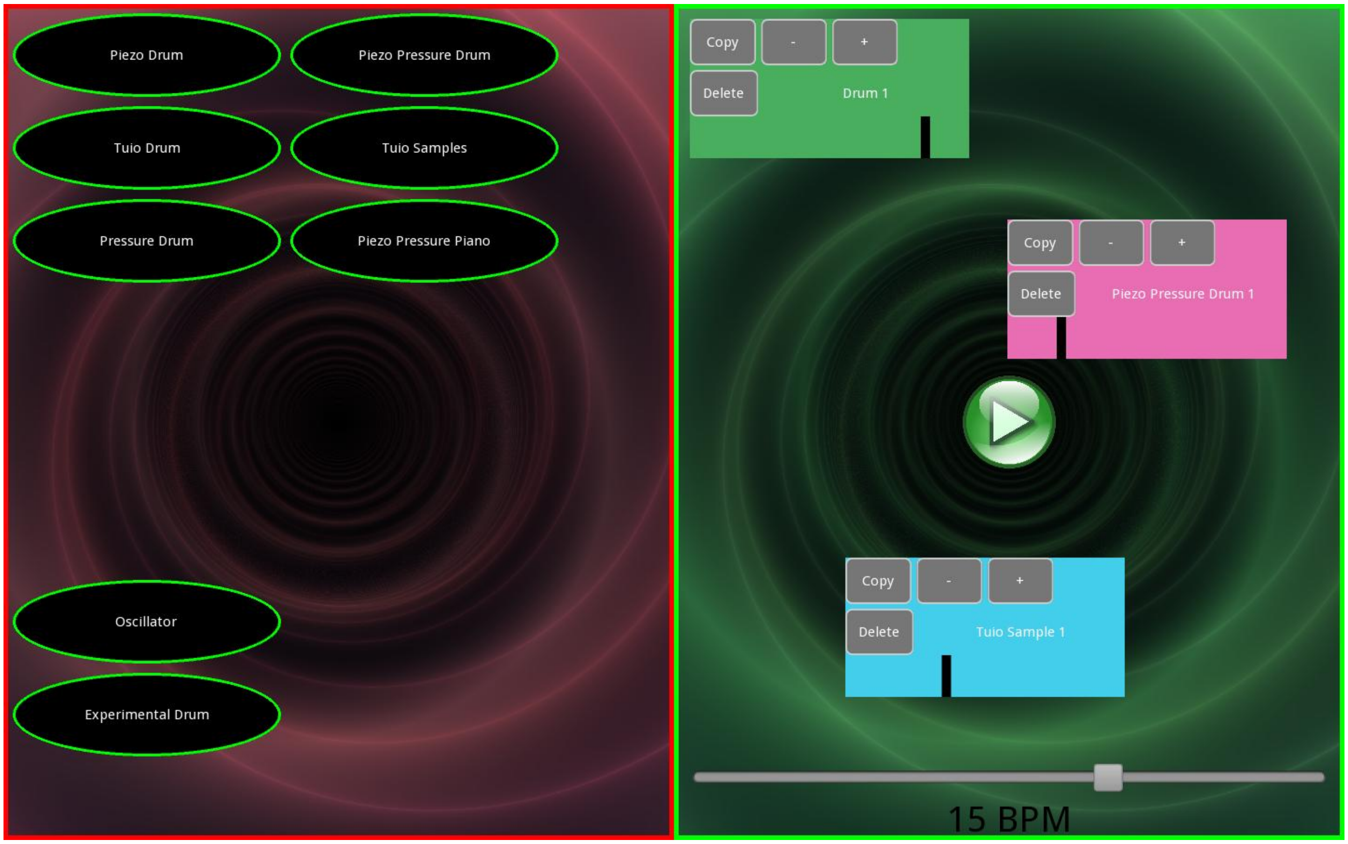

Figure 5: Bapp UI

performance loss, key sections of the Kivy and pyo code are compiled into C for a great performance boost. In fact, Kivy's graphics library uses C code to access OpenGL in a surprisingly well-performing, cross-platform implementation.

After some development, the final audio framework choice is Kivy's built in audio framework, which uses a GStreamer backend. Initially pyo was chosen for its ability in generative synthesis. Consequently, using pyo resulted in an enormous scope of development to include not only sensor fusion but also implementing complex computer music algorithms. To reduce the scope of the project, audio file playback suffices for drum pad sounds, using Kivy's built in audio capabilities. Another advantage to using Kivy's audio toolkit is its seamless compilation for other architectures, as opposed to cross-compiling pyo for the ARM architecture used by iOS and Android devices.

Application Overview

The Multi-Input Sensing Table application is a computer music program, demonstrating the viability of multi-sensor fusion for human computer interaction. The app uses the acoustic sensors to respond to high frequency taps of felted drumsticks. The pressure sensor data controls the volume of created sounds. The camera sensor is used for application set up and control. Multiple implementations of some features use different sensors to compare the methods of input.

The application is separated into two halves: creation and playback (Fig 5). Pressing a button on the left half of the screen will bring up a drum pad to record. After recording a beat, the dialog will automatically be dismissed after 5 seconds of inactivity, and a sound item representing the recorded beat is added to the main screen. Dragging this sound item to the right half of the screen will play the sound item in a loop, the length of which can be controlled.

Creation

The left side of the screen holds various options for sound creation: Tuio Drum, Tuio Samples, Piezo Drum, Pressure Drum, Piezo Pressure Drum, and Piezo Pressure Piano. After pressing a sound creation button, a modal popup displays for recording the beat. Upon creating the sound item, a visual representation of the sound item is added to the creation side of the screen. The sound items can be rotated and zoomed using the standard, intuitive multi-touch gestures.

In addition to the sound items above, some generative sound items are implemented using the pyo audio backend, although they can be quite odd sounding. The first generative sound available is "Drum." Choosing a left and right frequency customizes the drum, and then a beat is input. The second sound is "Oscillator." An oscillator is customized with the base frequency of the sound, and the oscillation frequency.

The Tuio Samples sound item uses the TUIO input to choose between 16 sound samples. These samples come from Daft Punk's "Harder, Better, Faster, Stronger."

The four drum kit sound items differ in the sensors used for input, but are very similar except for the sounds created by each. All these drum sound items share the same appearance. The Tuio Drum solely uses the camera's TUIO touch information to locate a touch on the screen and trigger a drum hit. The Piezo Drum uses the piezoelectric sensors' to perform four-quadrant detection to trigger sounds. Two-region detection with the piezoelectric sensors can be seen in the Piano Sound Item. Finally the Piezo Pressure Drum uses the piezoelectric sensors for touch localization, and the pressure sensor data to control the volume. The Pressure Drum uses only the pressure sensors for volume modulation and tap detection.

The Piezo Pressure Piano is another example of multi-sensor fusion. The piano keys are controlled using TUIO, while the rest of the sensors control two drum pads on the opposite side of the table. Using the four drums and piano shows the appeal of multi-sensor fusion for human computer interaction, as opposed to current methods of input. The piezoelectric sensors respond much quicker, and the pressure sensors allow much finer control of input. The TUIO is best when fine-grained input localization is required. The multiple implementations of drum pads using different sensors also shows the use cases best handled by each sensor.

Playback

After creating a sound item with any of the methods available on the left half of the screen, a sound item appears on the main application screen. These sound items can be dragged, copied, deleted, increased or decreased in volume, resized, and rotated. When a sound item is dragged onto the playback side of the screen, it begins playing in a loop. An animation shows the sound item's percentage of completion. The cycle will begin anew according to the beats per minute set in the bottom right of the screen. A sound item will not be clipped by a fast tempo, but will start again at the beginning of the next interval when the sound item is not currently playing. A metronome can be used to hear a tone at the start of each tempo period.

CONCLUSION

The additional sensors are useful, as shown by the higher frequency input allowed with the piezoelectric sensors. The piezoelectric sensors discriminate properly between four quadrants approximately 80% of the time when properly tuned. In addition, when only required to delineate two regions, the piezoelectric detection accuracy rises to around 90%. The maximum input frequency of the piezoelectric method outstrips the TUIO under experimentation. Given these advantages, the piezoelectric sensors are justified as an addition to the MIST. The pressure sensors provide some usable data, but would be far more interesting with a DC coupled interface to the controlling computer.

FUTURE WORK

Some optimizations can be employed in future work to increase the usefulness of the sensor data. Pressure sensor data should not pass through a soundcard so that the pressure data can be a useful magnitude reading instead of the pressure's derivative. A more complex algorithm could be used to increase the localization of the piezoelectric sensors, such as a concentric moiré model or a model of wave reflection at the edges of the table. Higher quality pressure and acoustic sensors would surely report more meaningful data with less variability than among the current sensors.

REFERENCES

1. NUI Group. Multitouch Technologies. 2009. nuicode.com/attachments/download/115/Multi-Touch_Technologies_v1.01.pdf
2. 3M Microtouch Display M2256PW – 22" Multi-touch Display. 3M. June 5, 2012 www.3mselect.co.uk/p-1945-3m-microtouch-display-m2256pw-22-multi-touch-display.aspx
3. TUIO Application Showcase. Martin Kaltenbrunner. June 5, 2012 tuio.org/?showcase
4. Community Core Vision. NUI Group Community. June 5, 2012 ccv.nuigroup.com
5. Pure Data – PD Community Site. Institut für Elektronische Musik und Akustik. June 5, 2012 puredata.info
6. Kivy: Crossplatform Framework for NUI. The Kivy Open Source Community. May 15, 2012 kivy.org
7. Velocity Materials Speed of Sound through Materials Velocity of Materials. BAMR. May 15, 2012 www.bamr.co.za/velocity of materials.shtml
8. pyo – Python digital signal processing module. Olivier Bélanger. May 15, 2012 http://code.google.com/p/pyo/
9. Lopes, P., Jota, R. and Jorge, J. Augmenting Touch Interaction Through Acoustic Sensing. *In Proc. ITS '11. 53-56.*
10. Harrison, C., Schwarz, J. and Hudson S. E. TapSense: Enhancing Finger Interaction on Touch Surfaces. In *Proc. UIST '11*. 627-636.
11. Tindale, A., Kapur, A, Tzanetakis, G., Driessen, P., Schloss, A. A comparison of Sensor Strategies for Capturing Percussive Gestures. *In Proc. NIME '05. 200-203.*
12. Heo, D., Lee, G. Force Gestures: Augmenting Touch Screen Gestures with Normal and Tangential Forces. *In Proc. UIST '11 621-626.*

*This is joint work between James McCloskey, Danny Anderson, Roger Jennings and David Medine. Many setbacks were encountered and overcome. The software was initially tested on an older table without the piezoelectric and pressure sensors, and this did not allow as much testing on the Multi-Input Sensing Table as hoped. Over the entire span of the project, the piezoelectric and pressure sensors have been in various states of disrepair and required maintenance. Fortunately David Medine was able to help with reprogramming the PureData sensor processing as the sensors became operational. Roger Jennings helped greatly in setting up the sensors. James McCloskey did the bulk of setting up the table. Danny Anderson wrote the Bapp application. The main problem over the project's duration was working with the constant changes to the table, and trying to perform useful testing with the set of currently working components. Many optimizations were attempted to improve the sensors' operation such as modifying the current and voltage levels, tuning and calibration within CCV, and parameter tuning in PureData. All challenges to developing the software (including hardware set up) were overcome, and the result is functional and clearly shows the useful fusion of multiple sensors.

A final version of this paper will be submitted to the ACM 2012 Interactive Tabletops and Surfaces conference.
Author order may be different in the final submission.