DESIGN AND APPLICATIONS OF A MULTI-TOUCH MUSICAL KEYBOARD

Andrew McPherson
Drexel University
apm@drexel.edu

Youngmoo Kim
Drexel University
ykim@drexel.edu

ABSTRACT

This paper presents a hardware and software system for adding multiple touch sensitivity to the piano-style keyboard. The traditional keyboard is a discrete interface, defining notes by onset and release. By contrast, our system allows continuous gestural control over multiple dimensions of each note by sensing the position and contact area of up to three touches per key. Each key consists of a circuit board with a capacitive sensing controller, laminated with thin plastic sheets to provide a traditional feel to the performer. The sensors, which are less than 3mm thick, mount atop existing acoustic or electronic piano keyboards. The hardware connects via USB, and software on a host computer generates OSC messages reflecting a broad array of low- and high-level gestures, including motion of single points, two- and three-finger pinch and slide gestures, and continuous glissandos tracking across multiple keys. This paper describes the system design and presents selected musical applications.

1. INTRODUCTION

Over the past decades, a great many electronic music controllers have been developed, but few approach the ubiquity of the piano-style keyboard. The keyboard's versatility and its large number of skilled performers ensure that it will maintain a prominent place in digital music performance for the foreseeable future.

The keyboard is by nature a discrete interface: on the acoustic piano as well as on most MIDI keyboards, notes are defined solely by onset and release, giving the performer limited control over their shape. Though certain MIDI keyboards are equipped with aftertouch (key pressure) sensitivity, this arrangement tends to lack nuance and flexibility. A key must be completely pressed before aftertouch can be used, so aftertouch cannot control articulation. Aftertouch is also difficult to use in rapid passagework, and control is limited to a single dimension.

Separately, interest has been growing in multi-touch music interfaces, particularly touch-screen devices such as the Apple iOS family. Touch-based devices can be used for continuous or discrete control, and relationships between gesture and sound can be dynamically adjusted. On the other hand, touch-screen interfaces lack the tactile feedback that is foundational to many musical instruments, and they require the consistent visual attention of the performer.

In this paper, we explore a synthesis between keyboard and multi-touch interfaces by integrating multiple touch sensitivity directly into each key. We present a new capacitive sensor system which records the spatial location and contact area of up to three touches per key. The sensors can be installed on any acoustic or electronic keyboard. The sensor hardware communicates via USB to a host computer, which uses the Open Sound Control (OSC) protocol [1] to transmit both raw touch data and higher-level gestural features to any sound synthesis program.

Integrating touch sensitivity into the keyboard creates a wide range of new expressive possibilities. In this paper, we will describe the hardware and software design of the touch system and present selected musical applications.

2. RELATED WORK

The idea of integrating touch sensitivity into the piano keyboard was first explored by Moog and Rhea [2], who constructed "multiply-touch-sensitive" keyboards recording the lateral and front-to-back position of the player's finger on the key surface as well as the continuous vertical position of the key itself. The sensors were installed in piano and organ-style keyboards, and the data was made available through a custom microcontroller interface. In this way, each key could be used to control up to three independent musical parameters.

Other authors have explored replacing the MIDI keyboard's discrete triggering with a continuous position measurement. Freed and Avizienis [3] demonstrate a keyboard featuring continuous position measurement and high-speed network communication. Our own previous work [4] uses optical sensing to measure continuous key position on the acoustic piano at 600Hz sampling rate. The sample rate is sufficient to capture anywhere from 10 to 100 points during the brief interval a key is in motion, recording not just the velocity but the *shape* of each key press. In addition to velocity, we demonstrate techniques for extracting up to four additional dimensions (percussiveness, weight into the keybed, depth, and finger rigidity) from each press [5].

The Haken Continuum [6] erases the mechanical boundaries between keys entirely, providing an interface capable of recording up to 10 touches in three dimensions. The continuous design facilitates glissando and vibrato gestures as well as continuous note shaping, though maintaining correct intonation requires precision on the player's part.

Capacitive touch sensing (and, more broadly, electric field sensing) has also seen musical applications beyond the keyboard. Paradiso and Gershenfeld [7] discuss applications ranging from the classic Theremin to baton and bow tracking. Guaus et al. [8] use capacitive touch sensing to measure a guitarist's fingering on the fretboard. The Snyderphonics Manta [1] provides a hexagonal array of capacitive touch sensors whose mapping can be dynamically assigned in software. The Buchla Thunder [2], among other Buchla controllers, is also based on capacitive sensing.

2.1 Revisiting the Touch-Sensing Keyboard

Though previous touch-sensitive keyboard designs exist, we believe that this a concept ripe for future exploration. Open Sound Control [1] offers faster, more flexible communication options than MIDI, enabling the practical use of high-bandwidth performance interfaces. Driven by widespread adoption in consumer electronics, capacitive sensing technology has become both cheaper and more robust, with several integrated digital controller systems available.

Simultaneously, increasing availability of complex, multidimensional performance interfaces has stimulated an interest in the *mapping* problem: given a large input space of performer gestures, how can a performer's actions be translated into sound in the most flexible, intuitive manner? Highly multidimensional interfaces can be difficult to control, but recent work has identified strategies which go far beyond the traditional one-to-one linear mappings between input sensors and synthesizer parameters [9].

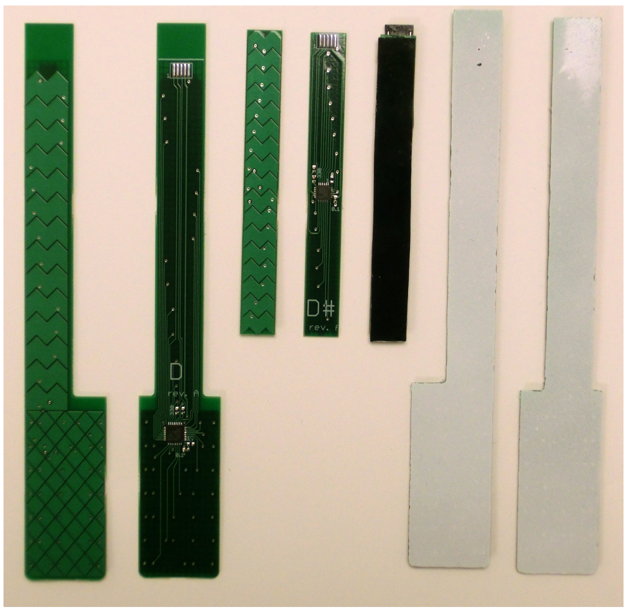

Figure 1. Multi-touch-sensitive piano keys. Boards shown from top, bottom, and with surface laminates.

[1] http://www.snyderphonics.com
[2] http://www.buchla.com/historical/thunder/

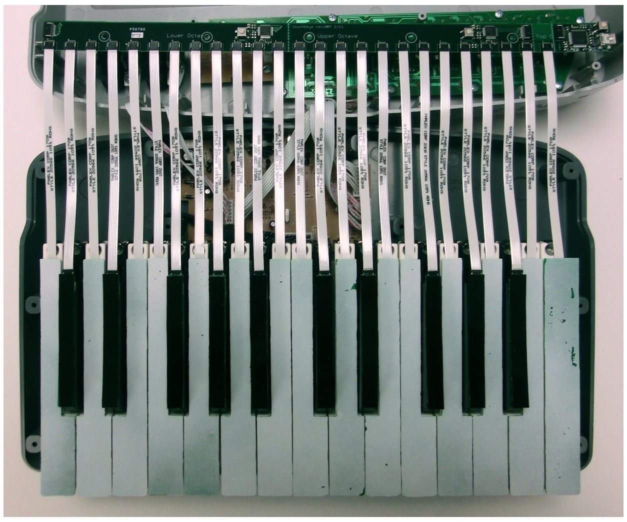

Figure 2. Two octaves of keys with controller board (top), mounted on a MIDI keyboard.

3. DESIGN OVERVIEW

In this context, we introduce a new system for measuring the position and size of up to three touches on each piano key. Black key position is sensed along a single front-to-back axis; white key touches are sensed in two axes on the wide front of the key and a single axis along the narrow back portion. Each key consists of a circuit board with an integrated circuit controller; the front surface of the board is laminated with thin plastic to provide a similar feel to the traditional key surface (Figure 1). The back is laminated with a thicker plastic sheet cut out around the components to provide a flat mounting surface. The entire 3mm thick assembly can replace conventional key tops (on acoustic pianos) or be fastened atop an existing keyboard (for molded plastic keyboards).

Figure 2 shows two octaves of keys attached to the controller board, which communicates with a computer via USB. Each key is scanned 125 times per second; the host computer processes the raw data to extract higher-level features including the addition and removal of touches, motion and resizing of existing touches, and multi-finger gestures including pinches and slides. These features are further described in Section 5.

4. SENSOR HARDWARE

Each key uses a Cypress Semiconductor *CapSense* controller [10] on a circuit board with either 17 or 25 sensor pads (black and white keys, respectively). Each pad forms a capacitor with respect to free space and a ground plane internal to the circuit board; a finger on or near the sensor surface increases its capacitance, which the controller reads as an 10-bit value. [3] On startup, the controller reads baseline values for each sensor with no finger present, subtracting these baselines from subsequent readings. No electrical contact is required between the per-

[3] Detailed principles of capacitive sensing can be found in Paradiso [7] and the Cypress datasheet [10].

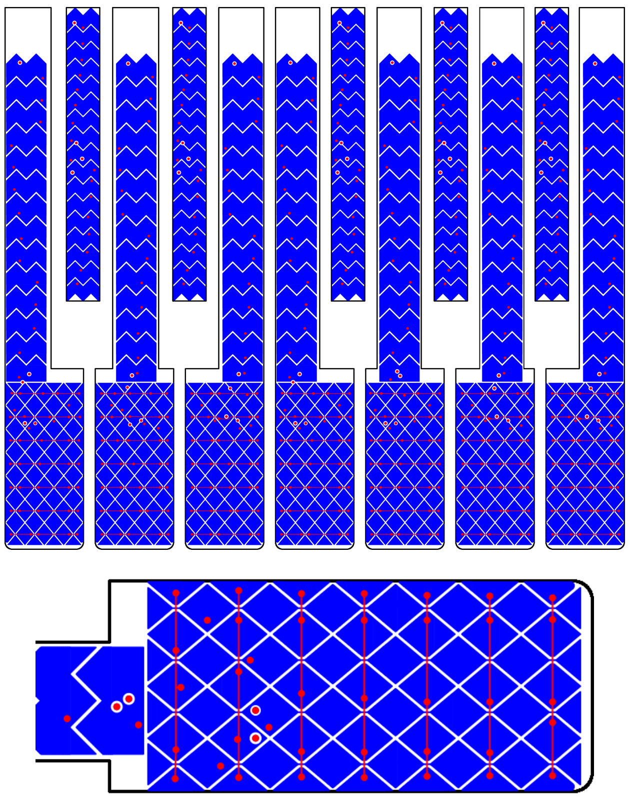
Figure 3. Top: Sensor pad layouts for one octave of keys. Blue pads are on the top layer of the board. Bottom: Close-up of XY sensor layout on white keys. Red wires on an inner layer connect pads into rows, blue wires on the top layer connect into columns.

former's finger and the sensors, and unlike resistive touch sensors, no pressure is required.

Figure 3 shows the sensor pad layout, which is designed so that any touch will activate several adjacent sensors. On the black keys and the narrow part of the white keys, the pads form a linear slider capable of measuring touch position in one dimension. On the wider front of the white keys, small diamond-shaped pads are collected into an interlocking grid of 7 rows and 4 columns using two circuit board layers, allowing horizontal and vertical position to be sensed. Figure 4 demonstrates the calculation of touch position and size. Position is calculated as the centroid of a group of adjacent active sensors:

$$Centroid = (\sum_{k=I_s}^{I_f} kC_k)/(\sum_{i=I_s}^{I_f} C_k) \qquad (1)$$

where $I_s \leq k \leq I_f$ defines a range of sensor indices and the C_k represent capacitance values. Touch size (contact area) is proportional to the sum of all sensors in a group:

$$Size = \sum_{k=I_s}^{I_f} C_k \qquad (2)$$

Raw centroid and size values are scaled to 0-1 range for later processing. Multiple independent touches can be sensed as long as their spacing exceeds the distance between sensor pads (4.75mm for black keys, 6.6mm for white). Centroid calculations are performed on each CapSense controller; a limit of 3 touches per key was chosen to provide a reasonable cap on calculation time. A complete sensor scan and calculation of centroids takes 4ms. Calculated vertical spatial resolution on the black keys is .08mm. On the white keys, resolution is .11mm in the vertical dimension and .09mm in the horizontal dimension.

4.1 Digital Communication

Figure 4 shows a diagram of the communication path. Each key transmits its calculated centroid and size data on a 400kbps I2C bus. I2C bandwidth and bus capacitance limitations preclude all keys from sharing a single bus. Instead, each octave of keys uses a separate I2C bus controlled by an Atmel AVR microcontroller. These "octave controllers" gather the data from each key and transmit it across a shared SPI connection running at 4Mbps. Flat ribbon cables connect the keys to the octave controllers to allow unimpeded key motion. System operation is ultimately controlled by an Atmel AVR with native USB capability (the "host controller"), which gathers the data from the SPI bus and transmits to the computer. The host controller is also responsible for regulating the timing of the sensor scans, initialization, and managing scan parameters.

Up to 8 octaves of keys can be managed by a single host controller. Each controller board (Figure 2) contains two octaves of keys, with a connection for a 13th key on the upper octave (since most keyboards end with a high C).

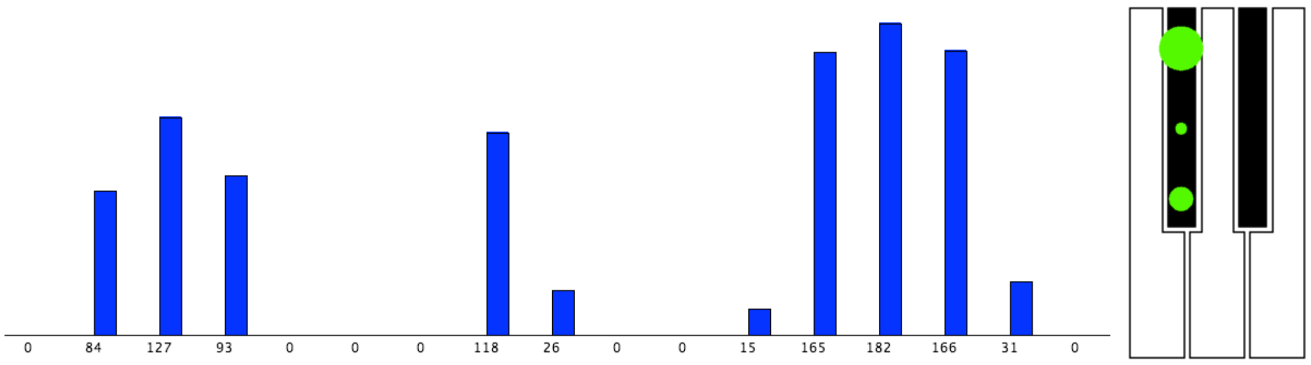

Figure 4. Discrete sensor pad readings are converted to touch position and size by calculating the centroid of multiple adjacent sensor values. Bars show sensor readings for C# key; green circles represent touch location and size.

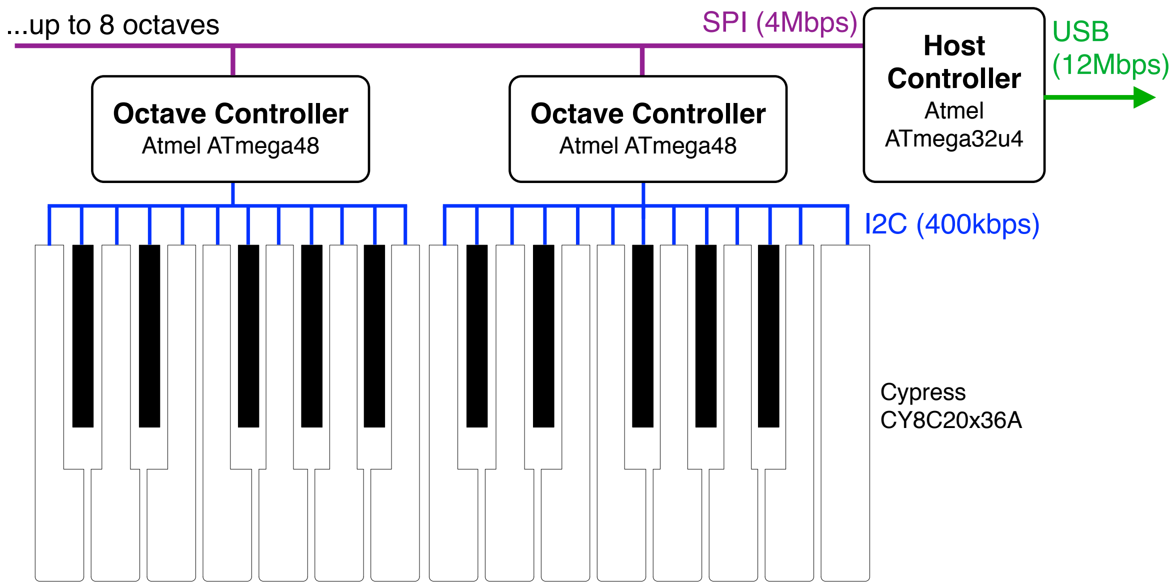

Figure 5. Communication architecture of the touch sensing system. Each octave of keys shares an I2C bus, with all octaves sharing a faster SPI connection. Communication with the computer takes place over USB.

To use more than two octaves, multiple boards are daisy-chained through flat ribbon cable connectors on either end. As only one host controller is needed, this part of the board can be broken off on the attached boards. On startup, the host controller dynamically determines the number of attached octaves.

5. DATA PARSING AND FEATURE EXTRACTION

The host controller appears to the computer as a USB communication class (CDC) device, which is natively supported by all major operating systems. Parsing software reads the incoming frames of touch data, producing OSC messages reflecting both raw touch values and higher-level gestural features. OSC messages can be sent to any port on the local machine or on the network, allowing connection to a broad array of synthesis software. Several programs exist which can further convert these OSC messages to MIDI data.

5.1 Raw Data Frames

For each key, transmitted OSC frames contain the octave and pitch class, the position and size of 3 touches (range 0-1, or -1 when not active), and for the white keys, a single horizontal position (-1 if the touch is not on the wide front of the key). Though it is possible to measure the vertical location of up to 3 touches, the sensor design only allows one unique horizontal position.

5.2 Multi-Key Gestures

Horizontal position sensing on the white keys allows the keyboard to emulate a ribbon controller: a touch can be tracked as it slides laterally across multiple keys. When it reaches the upper edge of one key, a new touch will register at the lower edge of the next. Our software stitches these touches together into a dedicated OSC "sweep" message (Table 1) containing a continuous location on the keyboard as well as information on the keys currently sensing the sweep. This mode of interaction is particularly well-suited for musical interactions based on glissandos or heavy pitch vibrato, though slide messages could also be mapped to any continuous control application.

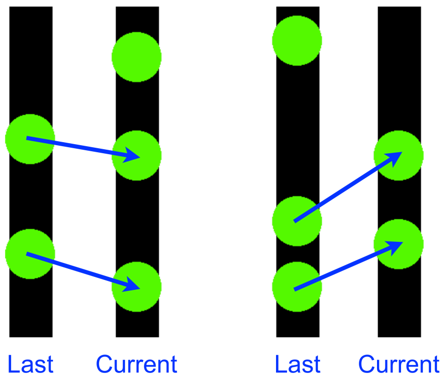

Figure 6. When the number of touches on a key changes, decide which touch was added or removed by minimizing the total motion of the other touches.

5.3 Gestural Feature Extraction

Data arrives from the keys as a series of discrete frames, but to provide expressive control over musical processes, the frames should be stitched together into a continuous picture of the performer's gestural interaction with the keyboard. As Figure 6 demonstrates, this requires some calculation when multiple touches are considered. We assign each new touch a unique ID that an OSC client can track from one frame to the next. When the number of touches changes from one scan to the next, we decide which touch was added or removed by minimizing the overall distance traveled by the continuing touches.

Table 1 summarizes the higher-level features we extract, which include not only single-finger gestures but multi-finger pinch and slide gestures often found on multi-touch mobile and tablet devices. Each mode of interaction can be mapped to distinct sound production behavior.

6. APPLICATIONS

The primary goal of adding touch sensitivity to the keyboard is to create an interface capable of *continuous, expressive* control. At the same time, the performer should not be burdened with an overly complicated interface, nor should he be expected to touch each key with pinpoint accuracy in order to produce acceptable sounds. The following set of examples demonstrate musical mappings that increase the range of subtlety available to the performer while still remaining straightforward to play.

6.1 Expressive Plucked String Synthesis

On the guitar, harp, and other plucked string instruments, the performer interacts directly with the string. Not only is a wide dynamic range possible; the player can also change the timbre of the note by plucking at different locations on the string, or by using the fingernail versus the softer fingertip. By contrast, traditional keyboards allow only dynamic control.

We use Csound[4] to create a virtual instrument based on the `pluck` opcode, which simulates a plucked string using the Karplus-Strong algorithm. The timbre of the synthesized pluck depends heavily on the initial conditions of the virtual string. We use touch location and touch size (measured at the time of note onset) to control the location and sharpness of the pluck. Specifically, the string's initial position is given by two cubic segments (Figure 7); the location of the peak along the string corresponds to touch location on the key, and a smaller touch size produces a sharper curvature. Both of these mappings simulate the conditions of actual physical string plucks: for example, a touch with the fingertip near the end of the key will produce a bright, thin sound, where a touch with the ball of the finger in the middle of the key will produce a rounder timbre with reduced high-frequency content.

[4] http://www.csounds.com/

6.2 Physically-Modeled Piano with Dynamic Action

The pianist interacts with the instrument's strings through a mechanical abstraction: the key controls a complex series of levers terminating in a felt-covered hammer. Strike point, hammer hardness, and the parameters of the bridge and soundboard are all fixed by mechanical design. On the other hand, physically-modeled piano synthesis has made great strides over the past decade, and it presents no such mechanical restrictions.

This mapping uses the Modartt PianoTeq synthesis software[5], which allows all major mechanical parameters to be dynamically assigned by MIDI Control Change messages. The Osculator program[6] is used to map OSC messages from our system to MIDI Control Change values. Key velocity retains its usual meaning. Vertical touch position at onset is mapped to strike point within a constrained range around its default point, giving the pianist more control over the timbre of each note while ensuring sensible musical results. Touch size maps to hammer hardness (smaller touches produce a harder hammer). Like the preceding example, these mappings give the keyboard player an intuitive sense of interacting directly with the piano strings.

[5] http://www.pianoteq.com/
[6] http://www.osculator.net/

OSC Path	Types	Data Contents	Description
/touchkeys/raw	iiffffff	octave (0-7), note (0-12), location/size pairs: [0, 1, 2] (range 0-1, or -1 if not active), horizontal location (-1 for black keys or upper portion of white keys)	Raw touch data
/touchkeys/on	ii	octave, note	Key became active
/touchkeys/off	ii	octave, note	All touches ended
/touchkeys/add	iiiifff	octave, note, touch ID, total # touches (1-3), new vertical location (0-1), new size (0-1), new horizontal location	New touch added
/touchkeys/remove	iiii	octave, note, ID, # remaining touches (1-2)	Existing touch removed
/touchkeys/move	iiiff	octave, note, ID, vertical location, horizontal location	Existing touch moved
/touchkeys/resize	iiif	octave, note, ID, size	Existing touch changed size
/touchkeys/twofinger/pinch	iiiif	octave, note, ID 0, ID 1, distance between touches	Two fingers pinched together or pulled apart
/touchkeys/twofinger/slide	iiiif	octave, note, ID 0, ID 1, (unweighted) centroid between touches	Two fingers moved up or down together
/touchkeys/threefinger/pinch	iiiiif	octave, note, ID 0, ID 1, ID 2, distance between outer touches	Pinch with three fingers on key
/touchkeys/threefinger/slide	iiiif	octave, note, ID 0, ID 1, ID 2, (unweighted) centroid of all three touches	Slide with three fingers on key
/touchkeys/multikey/sweep	iifiiifiiif	sweep ID, sweep octave position, sweep note position, key 0: [octave, note, touch ID, horizontal position], key 1: [octave, note, touch ID, horizontal position]	Continuous sweep across multiple white keys
/touchkeys/multikey/sweep-off	i	sweep ID	Multi-key sweep ended

Table 1. List of OSC messages sent by the touch-sensitive keys, reflecting low-level data and higher-level gestural features. Types `i` and `f` specify integer and floating point data, respectively.

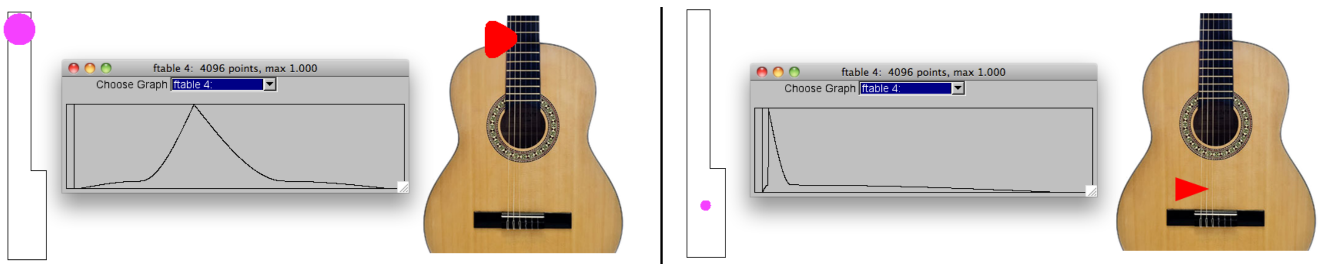
ftable 4: 4096 points, max 1.000
Choose Graph
ftable 4:
ftable 4: 4096 points, max 1.000
Choose Graph
ftable 4:

Figure 7. Expressive plucked string synthesis by mapping key touch location and size to pluck conditions. From left to right in each example: key touch, pluck initial conditions, physical analogy.

We explored several possibilities involving multiple fingers, including a mapping where a note played with two fingers increased the unison width of the piano strings in proportion to the distance between the touches, creating a "honky-tonk piano" effect when the two fingers were widely spaced. We also explored using the width between two fingers to modulate the impedance of the bridge, which affects the note decay. Widely spaced fingers create an unusually long sustain, and closely spaced fingers create notes with a clipped, muted quality. There is no obvious physical analogy for multi-finger touches, so the best mapping will depend on the specific musical situation.

In practice, this application was limited to monophonic playing, since changing PianoTeq settings affected all notes. Also, while some PianoTeq settings, including strike point and hammer hardness, took effect immediately, instrument-wide parameters such as bridge impedance exhibited a lag of up to one second, limiting their utility in practical performance situations. With suitable software adjustments, polyphonic dynamic piano modeling should be possible.

6.3 Continuous Timbre-Shaping

The preceding examples simulate (and extend) acoustic instruments whose note onsets are essentially discrete. We next show how touch sensing can be used to continuously modulate the sound of a note. We created a Csound instrument in which a harmonically-rich pulse waveform is passed through a resonant low-pass filter, similar to many classic analog synth topologies (Figure 8).

Using one finger, vertical position on the keys controls the filter cutoff frequency, and on the white keys, horizontal position can be used to bend the pitch up and down. The volume of the note can be changed with a two-finger "pinch" gesture, with a wider distance between fingers corresponding to higher amplitude. When three fingers are used, total finger spacing ("pinch") moves the note's fundamental frequency up the harmonic series of that key, with wider distance selecting a higher harmonic. Average position of the three fingers ("slide") controls filter cutoff.

6.4 Compound Instruments

A simple mapping produces novel and useful results: on each key press, the number of fingers on the key selects between one of three instrumental voices. Incoming MIDI Note On messages are routed to one of three channels depending on how many fingers are currently on the corresponding key (Figure 9). Note Off messages are sent to all three channels to avoid stuck notes.

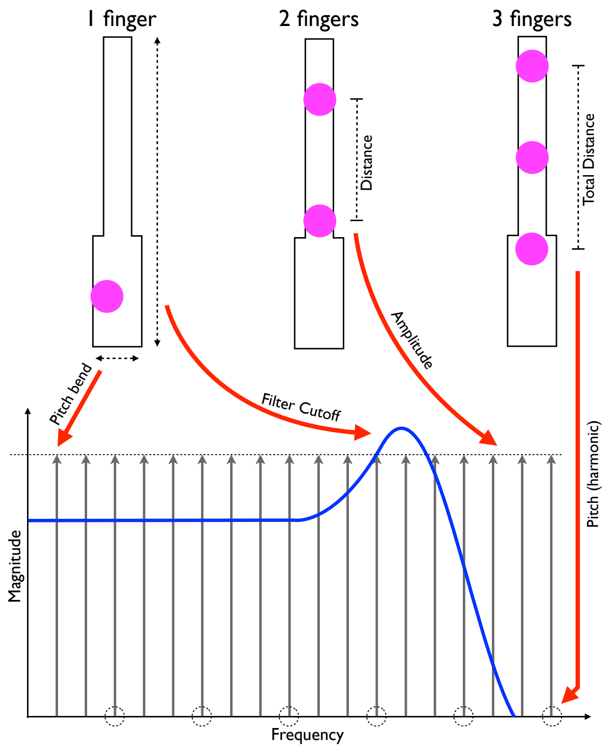

Figure 8. Continuous note-shaping using one, two, and three finger gestures.

Channel programs are configured so that one finger produces a piano sound, two fingers produce a bass sound, and three fingers produce percussion sounds. This arrangement allows a performer to play multiple instruments simultaneously from a single keyboard, with instrument selection performed on a note-by-note basis.

7. DISCUSSION

7.1 Guidelines for Effective Mappings

In developing mappings, two principles should be considered. First, the best musical results are often obtained from a subset of the possible controls. Assigning a separate meaning to every dimension can ultimately degrade the quality of performance by making the system needlessly complex. Second, the inherent asymmetry between the white keys, which sense horizontal position, and the black

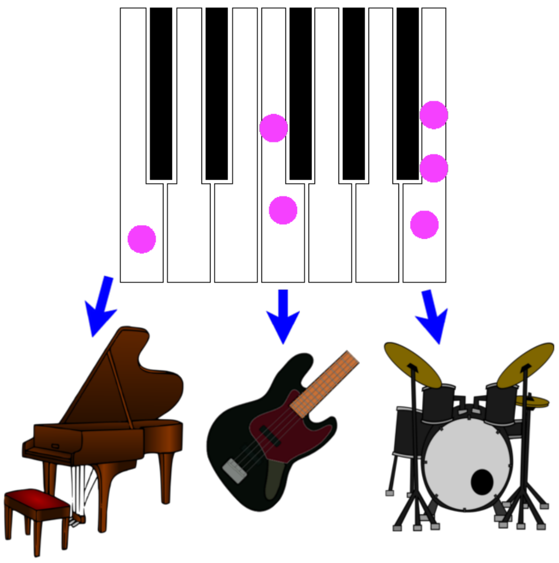

Figure 9. Compound instrument which selects one of three voices depending on the number of fingers on the key.

keys, which do not, must be addressed. This asymmetry reflects the keyboard itself, where the black keys offer a smaller contact area. In most cases, the most important musical parameters should be controllable from every key and not only the white keys.

7.2 Future Work

We recently used continuous position sensing to quantify key press gestures in multiple dimensions [5]. In a similar vein, future work will employ our touch sensors to examine the mechanics of traditional piano performance, with the goal of identifying relationships between expressive intent and finger motion. Dahl et al. [11] provide an overview of current approaches to expressive gesture sensing in keyboard performance, including measurements of pianists' hands, arms, torso and head. Real-time finger position on each key will be a valuable addition to these techniques, particularly since pianists devote a great deal of attention to the details of key "touch" (finger-key interaction).

A thorough understanding of finger motion in traditional performance also opens the door to new mapping strategies designed specifically around existing keyboard practice. We anticipate particular application to our work developing an electronically-augmented acoustic piano which uses electromagnets to shape string vibrations in real time [4], creating an intuitive performance interface extending the capabilities of the traditional piano.

7.3 Conclusion

We have presented a new interface which augments the keyboard with multiple touch sensitivity. The sensors install on existing keyboards, and associated software provides OSC messages for both raw touch information and higher-level gestures. We show several musical mappings, but the possibilities go well beyond these examples. Touch sensitivity can substantially enhance the expressivity of the keyboard by providing continuous control over several aspects of an instrument, and in judiciously designed applications, these additional dimensions need not substantially increase the complexity of performance.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant # 0937060 to the Computing Research Association for the CIFellows Project.

8. REFERENCES

[1] A. Freed and A. Schmeder, "Features and future of Open Sound Control version 1.1 for NIME," in *Proceedings of the Conference on New Interfaces for Musical Expression (NIME)*, Pittsburgh, PA, USA, 2009.

[2] R. A. Moog and T. L. Rhea, "Evolution of the keyboard interface: The Bösendorfer 290 SE recording piano and the Moog multiply-touch-sensitive keyboards," *Computer Music Journal*, vol. 14, no. 2, pp. 52–60, Summer 1990.

[3] A. Freed and R. Avizienis, "A new music keyboard featuring continuous key-position sensing and high-speed communication options," in *Proceedings of the International Computer Music Conference*, Berlin, Germany, 2000.

[4] A. McPherson and Y. Kim, "Augmenting the acoustic piano with electromagnetic string actuation and continuous key position sensing," in *Proceedings of NIME*, Sydney, Australia, 2010.

[5] ——, "Multidimensional gesture sensing at the piano keyboard," in *Proceedings of the 29th ACM Conference on Human Factors in Computing Systems (CHI)*, Vancouver, Canada, 2011.

[6] L. Haken, E. Tellman, and P. Wolfe, "An indiscrete music keyboard," *Computer Music Journal*, vol. 22, no. 1, pp. 30–48, 1998.

[7] J. A. Paradiso and N. Gershenfeld, "Musical applications of electric field sensing," *Computer Music Journal*, vol. 21, no. 2, pp. 69–89, 1997.

[8] E. Guaus, T. Ozaslan, E. Palacios, and J. L. Arcos, "A left hand gesture caption system for guitar based on capacitive sensors," in *Proceedings of NIME*, Sydney, Australia, 2010.

[9] M. Wanderley and P. Depalle, "Gestural control of sound synthesis," *Proceedings of the IEEE*, vol. 92, no. 4, pp. 632–644, 2004.

[10] Cypress Semiconductor, "PSoC CY8C20xx6A Family Technical Reference Manual," http://www.cypress.com/?docID=25608.

[11] S. Dahl, F. Bevilacqua, R. Bresin, M. Clayton, L. Leante, I. Poggi, and N. Rasamimanana, "Gestures in performance," in *Musical Gestures: Sound, Movement and Meaning*, R. Godøy and M. Leman, Eds., New York, NY, 2010, pp. 36–68.