Rocking the Keys with a Multi-Touch Interface

Thomas Walther*
Center for Computer Research in Music and Acoustics
Stanford University
Stanford, California, USA
thomaswa@ccrma.stanford.edu

Dr. Damir Ismailović
Department of Computer Science
Technische Universität München
Munich, Germany
damir.ismailovic@in.tum.de

Prof. Bernd Brügge
Department of Computer Science
Technische Universität München
Munich, Germany
bruegge@in.tum.de

ABSTRACT

Although multi-touch user interfaces have become a widespread form of human computer interaction in many technical areas, they haven't found their way into live performances of musicians and keyboarders yet. In this paper, we present a novel multi-touch interface method aimed at professional keyboard players. The method, which is inspired by computer trackpads, allows controlling up to ten continuous parameters of a keyboard with one hand, without requiring the user to look at the touch area - a significant improvement over traditional keyboard input controls. We discuss optimizations needed to make our interface reliable, and show in an evaluation with four keyboarders of different skill level that this method is both intuitive and powerful, and allows users to more quickly alter the sound of their keyboard than they could with current input solutions.

Keywords

multi-touch, mobile, keyboard, interface

1. INTRODUCTION

Today's pop music is increasily being influenced by house music, dubstep and other styles of electronic music. These genres heavily rely on a good and unique sound, and even more on altering these sounds during a song. Electronic music artists like Deadmau5[1] or David Guetta[2] write, record and produce their songs in studios, with modern digital audio workstations that allow tweaking and altering every aspect of a sound even after it was recorded. This form of production enables these artists to compose pieces that could not be performed live with current music gear. As electronic music has been becoming more mainstream these days, bands such as Foster the People are taking it on stage. These bands need to perform all the tweaking and modifying of their keyboard sounds live - a big challenge that electronic studio artists did not face.

*Thomas Walther conducted the research for this paper at the Department of Computer Science, Technische Universität München

[1]Deadmau5 is a very popular Canadian electro-house music artist: `http://www.deadmau5.com/`

[2]French house music producer David Guetta has heavily influenced mainstream pop music since 2009: `http://www.davidguetta.com/`

NIME'13, May 27 – 30, 2013, KAIST, Daejeon, Korea.

This tweaking and modifying is done via different kinds of input controls, which can be categorized into controls that toggle a state - e.g. switches, switch pedals and buttons - and controls for altering continuous values such as the volume. The latter group includes classical rotary knobs, endless rotary knobs, faders, expression pedals, distance sensors, modulation wheels, rubber bands and XY single-touch areas. [2, 3, 11, 12]

Figure 1: The Moog Voyager XL has many knobs and buttons, a two-dimensional touch area in the center, a rubber band right between the keys and the tilted control field, and two modulation wheels left of the keys. [2]

With XY single-touch areas that are used by artists today, one can control two parameters at a time by moving the finger horizontally or vertically, where putting the finger in the top right corner sets both values to maximum, and the lower left corner sets both to minimum. All controls for altering continuous values require the keyboarder to look at the control before she or he can use it. This is a real problem for live performances, where musicians may want to focus on their audience instead of aiming for controls. Additionally, moving from one control to another can be a tricky task. Figure 2 shows the alignment of two common sound parameters, reverb and amplifier release time, on the popular synthesizer Access Virus TI2. [5]

Figure 2: On the Access Virus TI2, the reverb knob is on the left, while the amplifier release time knob is on the bottom right.

As a keyboarder has only one hand for controlling the parameters - the other one is playing the keys - controlling both the reverb and the amplifier release time is nearly impossible. Current XY-Pads share the same problem as soon as three parameters have to be controlled.

In this work, a novel user interface is presented that solves both the problem of requiring the keyboarder to look at his controls as well as being limited to one or two parameters only. On the downside, our new method does not give any haptic feedback about the current value, and while visual feedback could be given, we want the user to be free to look at the audience. The user interface we present therefore relies on the user immediately hearing the modifications she or he does. We show in an evaluation that the interface significantly enhances the performance abilities of a keyboarder. After a few minutes of practicing, the users are able to use the new touch interface blindly, and one keyboarder even came up with a creative way of using the touch interface to control four parameters simultaneously.

2. RELATED WORK

Popular commercially available touch-based devices include the Korg Kaoss Pad series, a set of single-touch area effect boxes, and Lemur, a touch based controller that provides a Star Trek like interface for controlling sound parameters. S. Jordà et al. developed the popular reacTable [4], a multi-touch table for designing music. Hyung-Suk Kim et al. built a multi-touch interface for mobile devices that focuses on easy accessability of all parameters of tone generation [6]. Y. Kuhara et al. designed a synthesizer with a multi-touch screen interface based on particles that generate sound as they collide [7]. Lopes et al. developed a multi-touch-based interface for DJ live performances and compared it to traditional DJ controls by using a similar evaluation method as we did in our work [9]. Computer trackpads, especially Apple's Magic Trackpad [1] with its focus on multi-touch input, were a major influence to this work, too.

3. A MULTI-TOUCH USER INTERFACE FOR KEYBOARDS

The user interface we propose does not intend to replace any of the current controls available on a keyboard. Instead, it enhances the keyboard and gives the user the choice to use whichever input method she or he prefers in a certain situation. The presented user interface focuses on continuous parameters such as the volume, and does not aim to provide button-like functionality. We believe that other devices like stomp boxes provide a better way to toggle values, and that touch areas can never be better at toggling a button than real buttons, keys or pedals.

3.1 One Relative Canvas

Our user interface uses the whole touch area as one canvas. The user can touch anywhere as long as she or he hits the screen. The touch area is surrounded by borders to prevent the user from accidentally leaving the touch canvas. The behaviour of the touch area is similar to that of a computer trackpad: it isn't important where the user starts or ends his movement; only the distance her or his fingers travel matters. While moving the fingers, two parameters can be controlled at the same time: one parameter via the vertical, the other via the horizontal movement (c.f. figure 3).

Depending on the number of fingers touching the screen, different parameters can be controlled. The user could, for example, control volume and drive with one finger, and reverb and delay with two fingers. The parameters of the fingers do not add up, that means that in our example, if

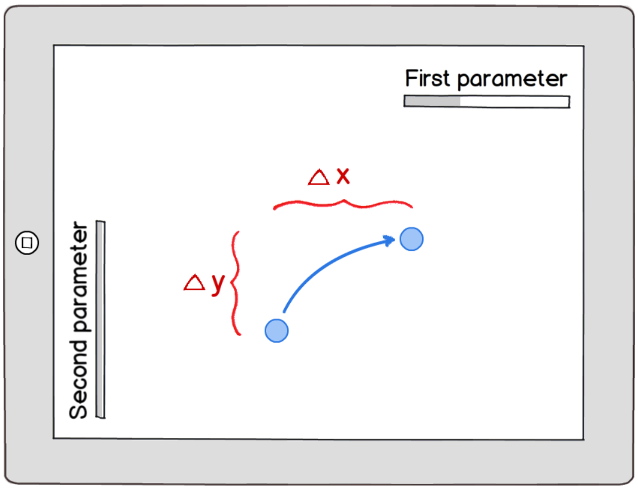

Figure 3: A mockup of the multi-touch canvas.

the user touches the screen with two fingers, only reverb and delay will be controlled, but not the volume or drive. When swiping with multiple fingers, the average swipe distance is used.

3.2 Configuring the Touch Area

The parameters that the user controls are freely assignable by the user. It is also possible to disable a direction of a finger so that one finger controls only one parameter and not two, or to disable both directions so that swiping the screen with a certain number of fingers doesn't do anything (c.f. figure 4). These settings can be saved into presets, and the musician can save and load a different preset for every song she or he performs.

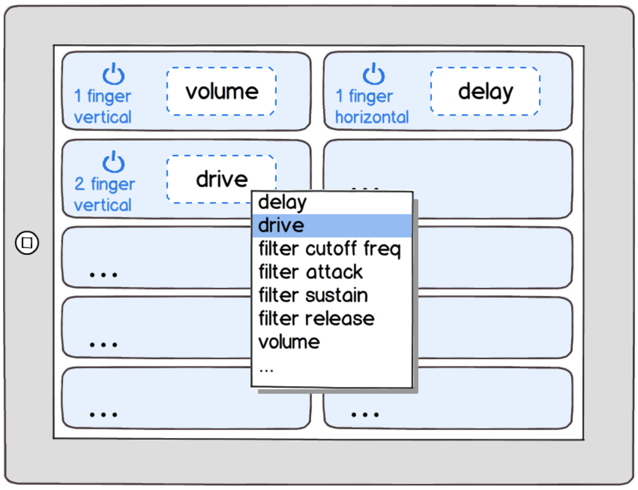

Figure 4: Via the setting screen, the user can select which parameters to control with which fingers and directions, and also disable some finger-direction combinations if he or she doesn't need or want them.

3.3 The Revert Function

Sometimes a musician may want to change her or his sound only temporarily. The setting screen allows the user to specify for each parameter if changes on it should be kept or reverted, after the user released the finger off the screen or changed the number of fingers with which she or he touches the screen. If the user spontaneously decides to keep changes made on a "temporary" parameter, there is a way to do that: let's assume the user has configured the touch area in a way that swiping vertically with both two and three fingers changes the volume, whereas the three-finger movement is only temporary. If the user modifies the sound with three fingers and lifts a finger off the screen, the system will check if the parameter that is to be reverted is still being controlled. If that is the case (as it is in our example), the value will not get reverted.

3.4 Connecting the Touch Area to a Keyboard

The user interface presented in figures 3 and 4 is easily implemented on a modern tablet device. This device is then connected to the keyboard via a USB cable[3] (c.f. figure 5). We use the Musical Instruments Digital Interface (MIDI) connection standard [10] to communicate with the synthesizer because this is the only communication method that commercially available synthesizers support[4].

Figure 5: The setup consisted of a tablet and a synthesizer - the tone-generating part of a keyboard - that were connected via USB. We used an iPad (3rd generation) and a Waldorf Blofeld synthesizer, a commercially available and widely used model.

4. EVALUATION

We performed extensive informal testing where we simply used and optimized the app until we were satisfied. We then tested our results in a case study with four keyboarders of different professional skill level. We explained the interface, provided them a sane set of parameter mappings so that they could start playing and testing without having to set it up from scratch, and watched them while they got used to the interface. Afterwards, we listened to the testers' feedback in an interview. The testers used the system separately to make sure we got individual opinions.

4.1 Results

All keyboarders were familiar with touch devices in general, and were able to comfortably use the touch interface blindly after about 15 minutes of getting used to it. We observed that the less-skilled piano player would stop glancing at the touch area much faster than the very skilled piano player, as he enjoyed the possibilities provided by the touch area that allowed him to modulate the sound quickly. During evaluation, the testers also came up with two novel ways of interacting with our user interface.

Within our informal testing and the case study, we found three major points that need to be considered to make the proposed user interface robust and intuitive.

[3]This assumes, of course, that the keyboard has a USB input, which most modern commercial models do

[4]As our interface changes parameter values relative to their current value, we had to make extensive use of SysEx messages, which are vendor specific extensions to the MIDI standard. Standard MIDI messages do not allow altering values relatively, and MIDI also doesn't provide a standardized way of fetching a current parameter value. Thus, we had to actually hardcode the communication for every synthesizer model.

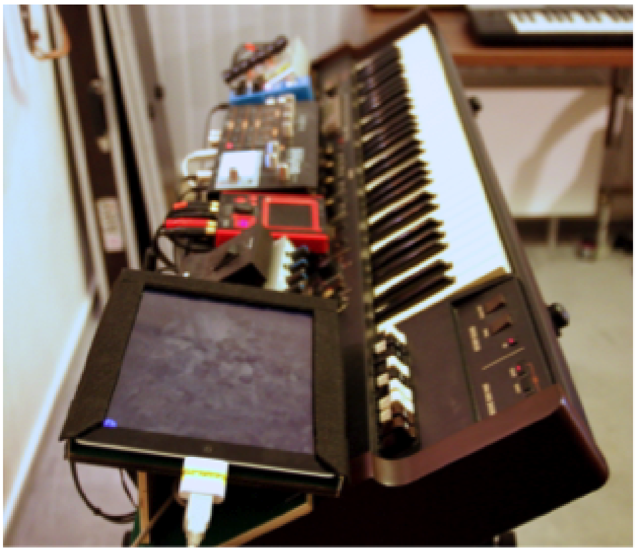

Figure 6: The evaluation setup - a standard, commercially available synthesizer (black device in the center) that is connected to the keys of an organ and a framed tablet (front) that provides the touch interface presented in this work. The other boxes in this picture were not used during the evaluation.

More Fingers, Higher Sensitivity

Swiping with one finger across a touch screen is naturally easier than swiping with four. Additionally, four fingers take up way more space on the screen than one finger, significantly reducing the distance the user could move his or her fingers. To compensate this, the sensitivity needs to linearly increase with the number of fingers on the screen. We found that a 50% higher sensitivity for swiping five fingers in comparison to swiping with one finger was a good value: our testers did not notice any difference in sensitivity between one and five fingers, whereas without this action, they felt that it was much harder to alter a value with five fingers than with one.

A Short Delay in Processing Touch Events

When the user touches the screen, she or he may not hit the screen with all fingers simultaneously. This scenario is illustrated in figure 7.

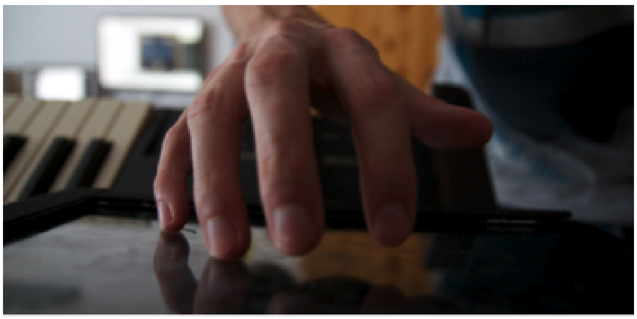

Figure 7: A user touching the screen with four fingers. Two fingers have already hit the screen, while the other two are just about to tap it.

To prevent the touch area from recognizing "intermediate" states, a short delay is introduced into the touch-event handling algorithm. If the number of fingers changes during this short interval, the touch event is discarded. After some informal experiments for a delay length of 20ms, 30ms, 40ms and 50ms, we found a delay length of 30ms to be completely unnoticeable while still providing enough reliability. This result is also in line with the latency research done by N. P. Lago and F. Kon in [8].

Overall Sensitivity of the Touch Area

The current touch-area based solutions cited in our introduction allow the user to set the value of a parameter to any value in its range by moving her or his finger anywhere near the bottom or the top corner. We first intended to provide sensitivity that would allow the same task, i.e. allowing the user to change a value from minimum to maximum (or vice versa) with just a single swipe. However, this turned out to be problematic, since some sound parameters like a vibrato speed only needed fine adjustments, and dramatic changes could lead to a bad sounding result. In this case, our testers wished for a much lower sensitivity. On the other hand, parameters like the filter cutoff frequency often need to be changed drastically. These requirements - subtle and dramatic changes - clearly contradict each other in terms of touch area sensitivity. Further research is needed to find out if the sensitivity has to be configurable on a per-parameter basis or if there is one level of sensitivity that fits all needs.

Further Ways of Interaction

As a result of our case study, we found out an interesting way to adjust the sensitivity: While tapping the screen with five fingers, one moves only one finger to adjust the parameter value. As the calculated distance is the mean average of all five finger movements, parameter sensitivity is only 1/5th this way. The interface also offers a way to adjust not just two but four parameters simultaneously: By swiping diagonally with one finger and continuously tapping with a second finger, one alternatively increases both the parameters that were mapped to one finger and the parameters mapped to two fingers.

4.2 Limitations

As one of the authors is an experienced keyboard player himself, we were able to make sure that every test person had a basic familiarity with playing the keys, as well as basic knowledge of synthesizers and their parameters. Thus, our evaluation only targeted experienced musicians, and not casual users. Also, the time each user spent with our system was, by nature, limited, and thus, we can't include any longterm feedback, which would be very valuable for our discussion about sensitivity in 4.1.

5. CONCLUSION

The touch based input method presented in this work enables new ways of interacting with a musical device. With its relative positioning approach, it solves the problem of requiring a keyboarder to look at his device when grabbing a control. In this respect, this solution is a major improvement over rotary knobs, faders, and touch areas with an absolute positioning approach as found on the Korg Kaoss Pad series and other devices. Furthermore, this method grants access to ten different parameters on the same touch surface - five times as many as single-touch areas provide.

The idea of relative positioning on touch areas isn't new, in fact we were surprised that no one else had yet implemented a similar solution, given the ubiquity of touchpads on notebooks. The similarity to touchpads is at the same time a huge benefit, as it lowers the entry barrier to using the concept. The keyboarders in our case study were able to use the prototype very quickly and perform actions blindly.

Further research needs to be done to improve the level of sensitivity of the touch area. To us, the most interesting question is whether there is one level of sensitivity that fits all use cases or whether the sensitivity needs to be adjustable by the user. As soon as we have added support for more synthesizers, we intend to release the prototype in Apple's App Store. We hope this will give us feedback from a large amount of users.

While this work has been focused on input methods for keyboards, we believe that the multi touch input method presented in this work can also be applied to other fields and sciences where a user needs to be able to blindly change a set of continuous parameters.

6. REFERENCES

[1] Apple. Magic Trackpad. www.apple.com/magictrackpad/, 2010.

[2] J. Buchanan. Minimoog Voyager XL. *FutureMusic*, (237):76–80, 2011.

[3] M. Hirsch. Korg Kaoss Pad Quad. *Keys*, page 76, Apr. 2011.

[4] S. Jordà, M. Kaltenbrunner, G. Geiger, and M. Alonso. The reacTable. *ACM SIGGRAPH 2006 Sketches on SIGGRAPH 06*, page 91, 2006.

[5] Kemper Digital GmbH. *Access Virus TI Catalogue 2009*. Recklinghausen, Germany, 2009.

[6] H.-S. Kim and J. O. Smith. FaustPad : A free open-source mobile app for multi-touch interaction with Faust generated modules. In *Linux Audio Conference*, pages 117–120, 2012.

[7] Y. Kuhara and D. Kobayashi. Kinetic Particles Synthesizer Using Multi-Touch Screen Interface of Mobile Devices. In A. R. Jensenius, A. Tveit, R. I. Godø y, and D. Overholt, editors, *Proceedings of the International Conference on New Interfaces for Musical Expression*, number June, pages 136–137. University of Oslo and Norwegian Academy of Music, 2011.

[8] N. P. Lago and F. Kon. The quest for low latency. *Proceedings of the International Computer Music Conference*, 2004.

[9] P. Lopez, A. Ferreira, and J. A. M. Pereira. Battle of the DJs: an HCI Perspective of Traditional, Virtual, Hybrid and Multitouch DJing. In A. R. Jensenius, A. Tveit, R. I. Godø y, and D. Overholt, editors, *Proceedings of the International Conference on New Interfaces for Musical Expression*, number June, pages 367–372, Oslo, Norway, 2011. University of Oslo and Norwegian Academy of Music.

[10] MIDI Manufacturers Association. *MIDI 1.0 Detailed Specification, Document Version 4.2*. Los Angeles, CA, Sept. 1995.

[11] Novation. Novation SL MkII. global.novationmusic.com/midi-controllers/sl-mkii, 2011.

[12] Roland Music Inc. Synthesizer/Keyboard Catalog 2011, www.roland.dk/support/catalogs/ Synthesizer_Keyboard_Catalog_2011.pdf, 2011