Using the Touch Screen as a Controller for Portable Computer Music Instruments

Günter Geiger
Music Technology Group
Universitat Pompeu Fabra
Ocata, 1
Barcelona, Spain
ggeiger@iua.upf.es

ABSTRACT

Using mobile devices as instruments in computer music is one of the goals of the "Pure Data anywhere" project [5]. An obstacle we encounter is controllability, because most of the devices do not offer the necessary interface, such as MIDI or USB, in order to be controlled by external controllers. Also, attaching external controllers to the devices would make them less portable.

This paper investigates the possibilities of using the touch screen, an interface that is part of mobile devices like Personal Digital Assistants (PDA's). It describes usage scenarios that have been implemented for the PD anywhere system. As most traditional PDA applications use the touch screen in the same way as a mouse would be used, emphasis is put on the difference between mouse and touch screen interaction for instruments.

We are going to describe interaction models, that were found useful and intuitive and enable the touch screen to become a fairly sophisticated controller for expressive real time music on a PDA.

Keywords

touch screen, PDA, Pure Data, controller, mobile musical instrument, human computer interaction

1. INTRODUCTION

Most mobile devices offer a touch screen which is used together with a pen in order to take the role of the mouse for application control. Apart from some additional buttons, the touch screen is the main interface to the system. Therefore it seems to be natural to use it as a controller for musical instruments that are running on the device. However, just using the touch screen as it is used in standard applications is too limiting. The WIMP (windows, icon,menu,pointer) based paradigm has proved to be difficult to handle for live musical performance, especially if the instrument should be highly interactive and feel natural. It is hard to express musical ideas by moving sliders and pressing buttons, more so if they are very small, as is common on PDA's.

Therefore we have tried to surpass these limitations and go beyond the traditional control paradigms, by using dynamic gestures on the touch screen to control software instruments running on the device itself.

Our goal in defining new interaction principles is to construct an interface that resembles traditional instruments, and especially portable instruments, in several ways.

First, it is one piece, the mobile device, and not a collection of controllers and synthesis engines.

It should stay a portable instrument, which amplifies its applicability to different contexts. Portable instruments play an important role in music culture, they can be moved easily, can be used in different social events and situations and deliver music outside the concert hall or studio. Having too many cables and addons would make the setup of the instrument more compilicated and its usage more error prone.

The instrument should have an interface that maximizes control and gives immediate feedback. It should, actually, feel like an acoustic instrument, but produce sound that acoustic instruments can't produce. It should also offer control on other than note level, such as score level or sound processing level [13].

It should be a learn-able and master-able instrument [7], and the player should be able to have direct feedback on how he advances in mastering the instrument [9].

Under these premises we try to develop a model of interaction with the touch screen.

1.1 Pure Data anywhere

Pure Data anywhere (PDa) is a port of the Pure Data (PD) computer music system [10] to Personal Digital Assistants. PDa is based on the Linux port of Pure Data and runs on any PDA that supports Linux. The graphical user interface is based on tcl/tk. PDa implements sound calculations with fixed point number values, therefore it runs in realtime on processors without floating point unit, such as those used in portable devices.

More information about the implementation can be found in [4].

PDa is currently used and tested on several models of HP iPaq's (namely those who run the familiar linux distribution), on PDA's that natively support Linux such as the Sharp Zaurus, and to a lesser extent (without the graphical frontend) on several of Apple's iPod models using the ipodlinux system. For the iPod there is even a costum graphical frontend called pdpod [8] available.

2. CHARACTERISTICS OF THE TOUCH SCREEN ON MOBILE DEVICES

In order to understand what the touch screen could offer, we are trying to highlight the features of touch screens.

The touch screen on a PDA feels very direct in its interaction, because of the connection of visual and haptic cues as well as the connection of the left hand holding the device and the right hand controlling it. This gives relatively good control over where the device is, and where the stylus is in relation to the device. This is very different from traditional touch screens, because these devices are not mounted on a screen anymore but freely movable, adopting more behaviors of touch tablets.[1]

This is also the reason why moving on the PDA touch screen is fast. Moving from one point to another involves the movement of both hands and arms. The haptic feedback and reference points about the screen dimensions as well as its limited size make it easy to navigate even without seeing the screen.

The touch screen can be operated either with a stylus (which gives more precision, but takes away immediacy) or directly by touching with the fingers. If we use two different fingers with the touch screen we can jump from one area of the screen to another area almost immediately and with relatively high precision. This mode of usage is seldomly used in traditional applications, but it makes perfect sense for musical applications when control has to be discontinuous and fast, or where one wants to achieve special effects.

We already mentioned the portability and the small size of touch screen equipped mobile devices. An important factor for a successful controller [14].

On the touch screen it is easy to move around, especially by small amounts, making the touch screen relatively a higher resolution device than for example the mouse, where it is hard to move just by one pixel. It is easy to remember positions haptically, because one can feel the borders of the screen.

The high precision that can be achieved when using a touch screen is also due to the short ways the stylus or fingers have to travel in order to change parameters, the whole range of the device is about 8cm, whereas the precision is between 320 and 640 units.

Additional haptic cues can be put onto the device, helping the instrumentalist to orient himself without having to look at the screen of the device.

3. THE TOUCH SCREEN AS SOPHISTICATED CONTROLLER

In this section we will work on ways of interaction with the touch screen using the possibilities outlined before.

Wessel [14] defines the principal interaction with digitizing tablets as “scrubbing”, “drag and drop”, “catch and throw”, and “dipping”. Our definitions partly overlap with these, they are not meant as complementing, but as a different point of view on interaction with a device very similar to a digitizing tablet.

We can identify several different forms of interacting with the touch screen, which partly overlap and merge into each other. The fact of being able to use several of the principles at the same time increases the possibilities of interaction and makes the controller more sophisticated. We start with a list of four interaction principles.

3.1 Region based triggering

Region based triggering will fire up distinct events based on where the touch screen was hit. In the WIMP paradigm this corresponds to a button press.

For musical purposes we can enhance the button press. The first enhancement is the power with which the button gets pressed. While interacting in a natural way with the touch screen we found out, that the hit does not take place on one single spot of the screen, but it takes the from of a small line. The speed with which this line gets drawn is an estimate for the power of the hit, and hence can be used as a control parameter for dynamics. Also the length of the line can be used for controlling purposes.

Simultaneity can be approximated by drawing from one region to another region, or by interaction with two fingers at almost the same time.

3.2 Gesture recognition

On a higher and more general level, there is gesture recognition. Gesture recognition allows us to control events from an alphabet of gestures. Not only the outline of a gesture can be used, but also its timing information. This adds another level of freedom to each gesture.

Gestures can be pretty complex, and if we include information about absolute position gesture are a very general way to define interaction on the touch screen.

The meaning of gestures in new musical instruments has been described by other authors [11] [12].

3.3 Border crossing

Another, newly defined interaction scheme we call border crossing. An event gets triggered when the pointer on the touch screen crosses a border. Additional parameters depend on speed of the crossover. This pattern draws from the interaction of plucking a string. Together with the sound feedback and the haptic feedback of the screen border, it is easy to learn the positions of borders by experimenting, and the human cognitive system seems to remember these positions pretty well.

Because of the precision and speed of the stylus on a touch screen, additional trigger information such as “double-triggered” (the border gets crossed two times in a very short time) events can be used to determine parameters on the note level.

Information such as the angle with which the border gets crossed can be employed as a continuous parameter.

3.4 Continuous parameter control

As an extension to the region based triggering, or as a stand-alone technique, the touch screen offers continuous parameter control. This means, that control data gets send according to the position of the pen on the touch screen. This allows us to combine piano-like event triggering with continuous control of the sound. This way the evolution of events can be controlled, depending on the mapping of the x and y values.

Continuous parameter control works best when the control input switches between states. One of the event based paradigms, region based triggering, gesture recognition or border cross triggering can be used to produce events, and for further control continuous parameters are used, until the pen (or finger) gets lifted from the touch screen.

3.5 On Virtuosity

One of the reasons why virtuosity can be reached with traditional instruments is their inflexibility. If one starts to

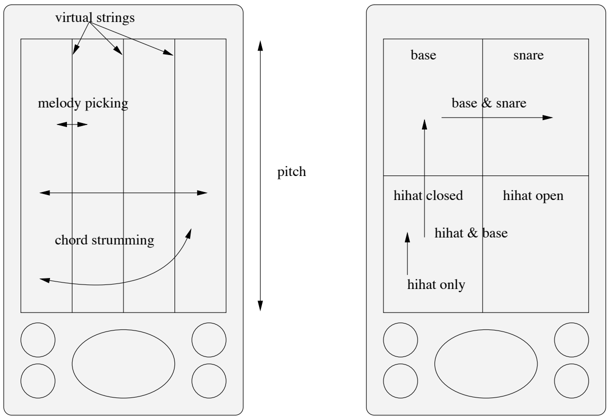

Figure 1: The guitar and drum set screen layout

learn a traditional instrument, she can be pretty sure that the instrument will not change substantially, hence it pays off to put an effort into learning to play it. Paradoxically, the flexibility of most computer music controllers makes it hard to reach higher levels of virtuosity on an instrument.

Unlike work specific to single compositions [2], one of the goals of this article is to come up with a set of interaction principles with the touch screen, that can be regarded as fundamental, studied and practiced as such in order to lead to a higher level of control of the instrument (or, more general, the interface).

We might fail for the first run, but experience and time will show which of the outlined principles preserve and which have to be adapted.

This, of course, assumes that touch screens will stay the main way of interaction with mobile devices.

4. MAPPING

Once refined our ways of interaction, we can work on the mapping of these control inputs to our instrument. At this time we have left the domain of the "controller" and entered the layer of the computer music instrument, generally hidden in the software of the device. Having good mapping strategies allows us to combine the principles of interaction in order to build a pleasing and enjoyable instrument.

The examples here should be seen as a proof of concept and an application of the interaction principles. Our mappings try to be simple in the hope that it will be easy figure out how the instruments react without needing a detailed description of synthesis techniques.

4.1 The virtual guitar

Figure 1 shows an application of the "border crossing" principle, implementing a virtual guitar. Interface resembles the FMOL program [6], but the interaction and sound production is different. The interface behaves similar to a real guitar, it is used to trigger events while crossing (picking or strumming) the virtual strings. Together with the event we can extract two parameters, which are the pitch and the velocity. The pitch is controlled by the vertical position while crossing the string. Simultaneity is achieved by strumming over more than one string, where the curve of the strum influences the type of chord that can be played (e.g. major/minor).

We have chosen on purpose a traditional instrument to simulate, but on a more abstract level of the interface, we can extract a decent amount of parameters from one strumming action. In the 3 string guitar example this would be 3*2, although not completely independent. Our abstract instrument would need something that makes tuned chords (= tuned parameters) useful. Also the velocities of the single strings are tightly coupled. Vibrato effects can be implemented by interpreting movements after "picking", using continous parameter control.

The virtual guitar is an example where additional haptic feedback on the touch screen improves play-ability further. This additional haptic feedback can be achieved by glueing transparent film on the surface, where the borders of the film correspond to the guitar strings and will be felt when strumming over them.

4.2 The virtual drum set

The second screen configuration in Figure 1 shows an example setup of a virtual drum kit. In this setup, events are not triggered by crossing borders, but by hitting certain areas of the screen.

The gesture arrows show how a simple two voice drum pattern would be played with this setup. The smallest rhythmical entity, the closed hi-hat is hit constantly, followed by a "hi hat - bass drum" and "bass drum - snare" combination. More complex drum sets could be built according to the rules for region based triggering outlined above.

According to need, other region layouts can be devised, such as the hi hat in the center, surrounded by other fields. Events that occur together frequently should have neighboring regions.

The virtual drum set mode can also be enhanced by haptically marking the regions.

4.3 The free and unquantified mode

The theremin, being the first controller in the league of new instruments, is one of the few instrument controllers where people have reached the level of virtuosity. It is therefore instructional to take a closer look at it and compare it with the touch screen.

The control of the theremin are two decoupled one dimensional sensors. One of the virtuosity criteria is the ability to hit a specific note and stay in tune. This is done via vibrato and by remembering relative positions which correspond to intervals.

When using a touch screen as a 2 dimensional controller, just using x for volume and y for pitch, one has to fight with similar problems as a theremin player, like finding the right pitch.

Another problem of the touchscreen in this mode is the freedom of movement. The finger can only move in one plane, volume and frequency are tightly coupled, making it harder to hit the notes correctly and perform a decent frequency vibrato.

On the other hand, on the touchscreen frequency and volume vibrato can be combined easily.

The frequency solution is not sufficient, therefore the frequency should be coupled with detection of the speed with which the stylus is moved.

5. THE IMPORTANCE OF FEEDBACK

In digital instrument design, feedback is a very important factor if we want the instrument to be playable. The touch screen, not being designed as an instrument, lacks kinesthetic feedback.

Nevertheless the tactile feedback gives good information about the position of the pen or finger on the touch screen.

The described scenarios have a very tied coupling of event and sound feedback, this is on purpose because the sound is the only dynamic feedback given to the user.

Another well integrated form of feedback is the visual feedback. Unlike with the mouse or other controllers, visual feedback can be displayed where the interaction occurs. This might be helpful for playing the instrument, but the history of instrument playing shows us that only few instruments actually depend on visual feedback. We encountered that having to look at the screen actually takes away a lot of the fun when playing the instrument.

The most important factor that has to be mentioned when playing mobile devices is the feel of the instrument. Having a portable device and being able to control it precisely motivates to go on with it and to learn how to have the total control.

6. FUTURE ENHANCEMENTS

The general way of playing a hand held device is by holding it in one hand and playing the touch screen with the other hand. This interaction works well for the touch screen, unfortunately for the hand that is holding the device it is hard to reach the additional buttons on the device. The only button that is normally reachable is the record button.

Having the ability to input additional information with the hand holding the device would greatly enhance the flexibility of the instrument. A solution for this would be the design of a jacket around the hand held, which serves at the same time as a protection and a input device with several buttons (one for each finger)., similar to the DataEgg[3], an input device developed by NASA for astronauts.

These additional, but more static inputs would make it easier to change states or presets of the instrument or generally to be used as an additional controller.

7. CONCLUSIONS

We have shown that the touch screen of mobile devices can be a versatile and musically meaningful instrument controller. Describing patterns of gesture interaction and explaining instruments implemented with these patterns we have demonstrated their usefulness.

We found playing with an instrument implemented in a portable computer stimulating and interesting, as well in private settings with headphones as in concert setting.

This opens up the road for new applications that convert a PDA into a portable musical instrument and make this kind of computer based music instruments easily accessible and usable, not only in complex concert or studio setups, but also in any other setting, by just pluggin in the headphones and playing along.

8. REFERENCES

[1] Bill Buxton, *Human Input to Computer Systems: Theories, Techniques and Technology*, http://www.billbuxton.com/inputManuscript.html, 2006.

[2] Carlos Cerena, *Gesture control of musical processes - a MAX environment for the Lightning*, Organised Sound **5** (2000), no. 1, 3–7.

[3] Gary Friedman, *Introducing the dataegg*, http://www.e2solutions.com/dataegg/index.html, 1992.

[4] Günter Geiger, *Pda - real time signal processing and sound generation on handheld devices*, Proceedings of the 2003 International Computer Music Conference (San Francisco), International Computer Music Association, 2003.

[5] Günter Geiger, *Pure data anywhere*, http://gige.xdv.org/pda, 2005.

[6] Sergi Jorda, *Faust music on line: An approach to real-time collective composition on the internet.*, Leonardo Music Journal **9** (1999), 5–12.

[7] Sergi Jordà, *Digital lutherie: Crafting musical computers for new musics performance and improvisation*, Ph.D. thesis, Universitat Pompeu Fabra, Barcelona, 2005.

[8] Martin Kaltenbrunner, *Pdpod*, http://ipodlinux.org/Pdpod, 2005.

[9] Sageev Oore, *Learning advanced skills on new instruments.*, New Interfaces for Musical Expression (Vancouver), University of British Columbia, 2005.

[10] Miller Puckette, *Pure data: another integrated computer music environment.*, Second Intercollege Computer Music Concerts (Tachikawa, Japan), 1996, pp. 37–41.

[11] M. Wanderley and M. Battier, *Trends in gestural control of music.*, http://www.cdemusic.org, 2000.

[12] Marcelo M. Wanderley, *Gestural control of music*, International Workshop Human Supervision and Control in Engineering and Music, September 2001.

[13] Marcelo M. Wanderley and Nicola Orio, *Evaluation of input devices for musical expression: Borrowing tools from hci*, Computer Music Journal **26** (2002), no. 3, 62–76.

[14] David Wessel, Matthew Wright, and John Schott, *Intimate Musical Control of Computers with a Variety of Controllers and Gesture Mapping Metaphors*, Conference on New Instruments for Musical Expression, 2002.