NUANCE: ADDING MULTI-TOUCH FORCE DETECTION TO THE IPAD

Romain Michon, Julius O. Smith III, Chris Chafe, Matthew Wright, and Ge Wang
Center for Computer Research in Music and Acoustics (CCRMA), Stanford University
{rmichon,jos,cc,matt,ge}@ccrma.stanford.edu

ABSTRACT

NUANCE is a new device adding multi-touch force detection to the *iPad* touch screen. It communicates with the *iPad* using the *audio jack* input. Force information is sent at an audio rate using analog amplitude modulation (AM). NUANCE provides a high level of sensitivity and responsiveness by only using analog components. It is very cheap to make.

NUANCE has been developed in the context of a larger project on augmenting mobile devices towards the creation of a form of hybrid lutherie where instruments are based on physical and virtual elements.

1. INTRODUCTION

Smartphones and tablets have been widely used as musical instruments and musical controllers during the last decade [1–6] (to only cite a few). In an era dominated by the controller/synthesizer paradigm where these two entities are often physically (or at least conceptually) separated, mobile devices provide a platform allowing the creation of a wide range of standalone musical instruments. The combination of different types of sensors, a microphone, a speaker and a powerful small computer in a battery powered single entity makes this possible. Moreover, the fact that this kind of device is mass produced guaranties a certain level of robustness which is not always the case of DIY (Do It Yourself) interfaces and instruments.

Carried by a huge market, mobile devices evolve very fast, and today's smartphones and tablets are quite different from the ones available a couple of years ago [7]. While only simple instruments and basic interactions could be implemented then, today's devices give access to a huge amount of computational power, low-latency touch screens allowing on the order of ten simultaneous touches, and good quality ADC/DACs.

However, mobile devices were never designed to be used as musical instruments and they are often missing crucial elements to be as expressive and versatile as professional musical instruments [8]. For instance, the lack of force sensitivity on touch screens narrows the range of possible interactions, and makes performance with specific classes of instruments such as percussion and plucked string instruments less intuitive. For example, striking force must be substituted by some other dimension such as the y coordinate in a strike area.

Park et al. addressed this issue a few years ago and proposed a solution using the built-in accelerometer of the device and a foam padding [9]. While this solution is very self-contained as it uses only the built-in sensors of the device, it presents several limitations diminishing the range of practical applications (e.g., no multi-touch support, sensitivity to table/support vibrations, no automatic re-calibration, limited sampling rate, etc.).

Some of the most recent generations of devices such as the *iPhone 6* provide basic multi-touch force detection on the screen ("3D Touch"). [1] This feature has already been exploited by some companies such as ROLI with its *Noise* app [2] to create expressive musical instruments. Unfortunately, this technology is not yet available on larger screen devices (tablets, etc.) that provide a better interface to control certain type of instruments such as percussion [10]. Instead, tablet manufacturers currently favor the use of force sensitive pencils [3] [4] that provide a simpler solution to this problem. Also, the "3D Touch" technology of the *iPhone 6* has some limitations. While it can provide very accurate data in the case of a continuous touch event ("after-touch"), its useability for deriving the *velocity* of a strike gesture on the screen is very limited. This makes it practically unusable to control percussion or plucked string instruments where the attack is very sharp.

In this paper, we introduce NUANCE: a device adding high quality multi-touch low-latency force detection to the *iPad* touch screen, fast and accurate enough to be suitable for deriving striking velocity. NUANCE is based on four force sensitive sensors placed on each corner at the back of the device. It communicates with the *iPad* using its *audio jack* connector through a purely analog system streaming the sensor data as an audio signal. This ensures a fast data rate (up to the audio bandwidth of the *iPad*, nominally 20 kHz) as well as a high sample resolution (bit depth).

After describing the hardware implementation of NUANCE, we demonstrate how it can be used to design musical instruments. We then provide a series of examples, evaluate its performance and discuss future applications and improvements.

[1] http://www.apple.com/iphone-6s/3d-touch/ All the URLs presented in this paper were checked on 2016-07-11.
[2] https://www.roli.com/products/noise.
[3] http://www.apple.com/apple-pencil/.
[4] https://www.microsoft.com/surface/en-us/accessories/pen.

2. HARDWARE

The case of NUANCE is made out of wood (plywood and birch) and black laser cut acrylic (see figure 1). The current version was designed for the *iPad Air 2* [5] but it is also compatible with the *9.7 in iPad Pro*. [6]

An FSR (Force Sensitive Resistors) [7] is placed under each corner of the *iPad* (see Figure 2). The FSRs are covered with a thin (1/4 in) piece of foam whose rigidity was chosen to offer a good compromise between responsiveness and damping [10]. The foam is used to cushion the strikes of the performer and also to give some slack to the *iPad* during continuous push gestures (after-touch).

The signals from the different FSRs is sent to the *iPad* using amplitude modulation (AM). Each force signal controls the gain of its own analog oscillator. The oscillators are very simple based on a *555 timer* (see Figure 3). This kind of circuit doesn't generate a pure sine wave but it is straightforward to efficiently isolate each carrier wave during the demodulation process described in §3 (reducing the dynamic range of the signal is not a problem since it is pretty large, thanks to the audio ADC).

The frequency of each oscillator is different and is controlled by $R1$ and $C1$. Frequencies (2, 6, 10 and 14 kHz) are spread across the bandwidth of the line input of the *iPad* (assuming that the sampling rate of the target app is at least 44.1 kHz). Since we're not carrying an audio signal, and since the sharpest attack we could achieve by tapping the screen was longer than 10ms (corresponding to a bandwidth less than 100Hz), we don't have to worry about sidebands. Moreover, the demodulation technique used on the *iPad* (see §3) significantly reduces the risk of sidebands contaminating neighboring signals.

The FSRs were calibrated to output their maximum value when a weight of approximately 400 grams is applied on the touch screen. This was set empirically to provide the best feeling to our taste but this value (that should remain reasonable to not damage the screen) can be easily adjusted by a small potentiometer mounted on the circuit board. Since the *iPad* itself applies some weight to the FSRs, the minimum value of the range is constantly adjusted on the software side (see §3).

The output of the different oscillators is mixed and sent to the *iPad* using the line input pin of the *audio jack* input.

The system is powered with an external 5 V power supply and connects to the *iPad* using a 3.5 mm (1/8 inch) headset (four-pole) audio jack. The output signal is routed to a 1/4 inch stereo audio jack mounted on the side of NUANCE.

The circuits were chosen to be as simple as possible to reduce the cost of NUANCE to less than $30. Variations of the sine oscillators (stability of the frequency, purity of the sine wave, etc.) are easily compensated on the software side.

[5] `http://www.apple.com/ipad-air-2/`.

[6] `http://www.apple.com/ipad-pro/`.

[7] The FSRs used for the device are *Interlink Electronics FSR 400*: `http://www.interlinkelectronics.com/FSR400.php`.

3. SOFTWARE

The amount of force applied to each FSR of NUANCE is carried by different sine waves to the *iPad* using its single audio analog input. Four band-pass filters isolate the signal of each sine wave (see Figure 4). Their bandwidth is big enough to accommodate the variations of the frequencies of the simple sine oscillators described in §2. The amplitude of the output signal of each filter is extracted and corresponds to the force measured by each FSR of NUANCE.

The DSP (Digital Signal Processing) portion of the different apps compatible with NUANCE that are presented in §4 is implemented using the FAUST [8] programming language [11]. The audio process is wrapped in a `C++` library using MOBILEFAUST [12] making it accessible to the higher level `Objective-C` layers of the app.

The force information extraction system described in the previous paragraph is implemented as a single FAUST function that is executed in the same audio callback function as the synthesizer that it is controlling. In other words, the force data from the FSRs are acquired and are controlling the synthesizer at the audio rate.

Touch events (including the (x, y) position on the screen) retrieved in the `Objective-C` layer are sent to the FAUST DSP object using the API provided by MOBILEFAUST. These data are compared with the data provided by the FSRs to associate a force signal to a specific touch event on the screen (see Figure 4). If the touch event was just initiated, the force is converted into a velocity proportional to the instantaneous force at the beginning of the touch. If the touch event persists, then the force is converted into a series of after-touch events.

As mentioned in §2, the *iPad* itself applies a certain amount of weight to the FSRs. Depending on its position in the case or the level of inclination of the table where NUANCE is installed, this value can vary a little bit. To make sure that the range of the FSRs is always accurate, it is readjusted when there are no fingers touching the screen. If there are several simultaneous active touches (multi-touch) on the screen, a simple triangulation algorithm compares the force level at each FSR with the X/Y position of the touch on the screen to associate a velocity or an after-touch event to it. Obviously, the more simultaneous touches on the screen, the harder it becomes for the system to differentiate independent forces. We find that the system is very accurate in the case of two simultaneous touches, but the force distribution tends to become more uniform if more touches are engaged. However, we find that this is not an issue for many types of percussion and plucked-string instrument control.

4. EXAMPLES

While it would be quite easy to write an app to use NUANCE as a MIDI controller, we like the idea of creating standalone musical instruments taking full advantage of the possibilities offered by this system. For this reason, the different apps that we created and that are compatible with NUANCE target specific instruments. However, we hope to

[8] `http://faust.grame.fr`.

Figure 1. Global view of NUANCE.

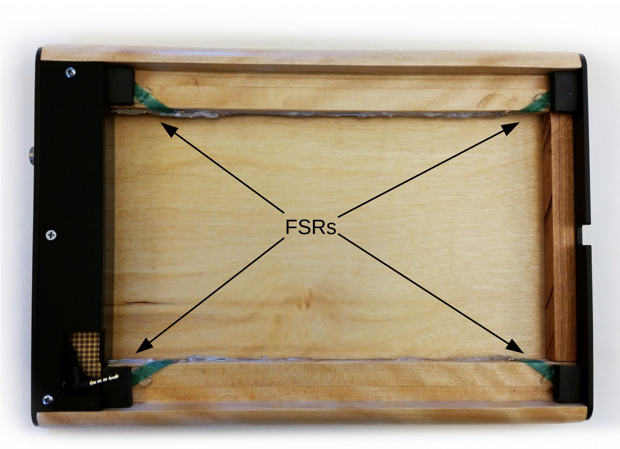

Figure 2. Top view of NUANCE without the *iPad*.

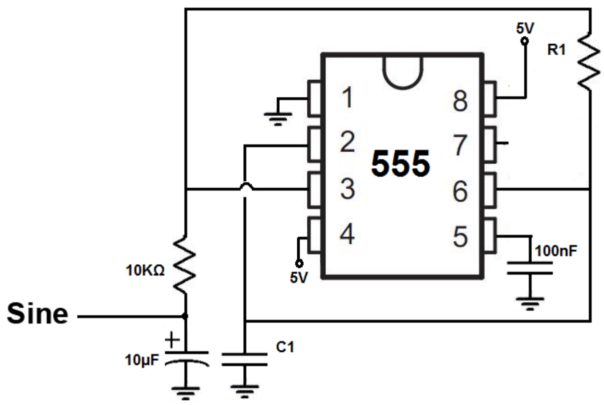

Figure 3. Circuit diagram of one of the simple sine oscillators used in NUANCE.

create a platform (see §6) to facilitate the design of multiple kinds of instruments, based on a mobile device and some hardware augmentation where several instruments would be accessible through a single app.

While NUANCE would work well with a wide range of screen interfaces (piano keyboards, isomorphic keyboards, etc.), we mostly focused percussion instruments so far. The different apps that we created implement one or several drums represented by rectangular regions on the touch screen. The app presented in Figure 5 has three different zones, each controlling a different drum. Each offers a physical representation of the virtual instrument (striking on the edges sounds different than striking at the middle, etc.). The drum synthesizers are all based on modal physical models [13] which strengthens the link between the physical and virtual parts of the instrument, increasing its overall "physical coherence". The physical models were implemented in FAUST using the FAUSTSTK [14].

The multi-touch capabilities of NUANCE allow to simultaneously strike two drums with different velocities. The after-touch information can also be used to interact with the resonance time (T60) of the virtual drums. This results in a highly expressive instrument [9].

5. EVALUATION/DISCUSSION

The "most standard" way to connect an external music controller to the *iPad* is by using MIDI through the lightening connector. While this solution works well for most basic applications (e.g., a MIDI keyboard triggering events in a synthesizer), the limited bandwidth and bit depth, as well as the jitter in the latency of MIDI, can be problematic for applications requiring a high rate of data and precise synchronization [15].

The idea of using the line-input of the *audio jack* plug of mobile devices to send data to it has already been exploited a lot. Various commercial products such as credit-

[9] `https://ccrma.stanford.edu/~rmichon/nuance/` presents a series of demo videos of NUANCE.

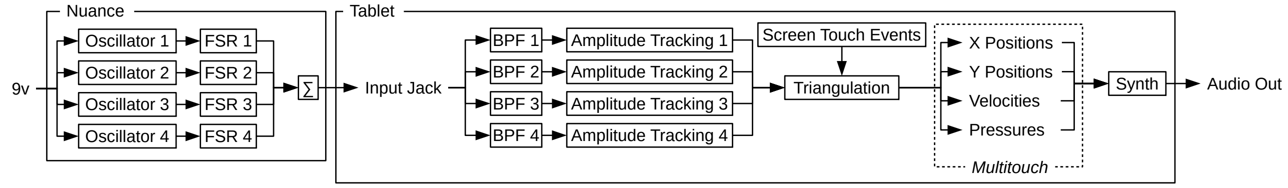

Figure 4. Overview of NUANCE.

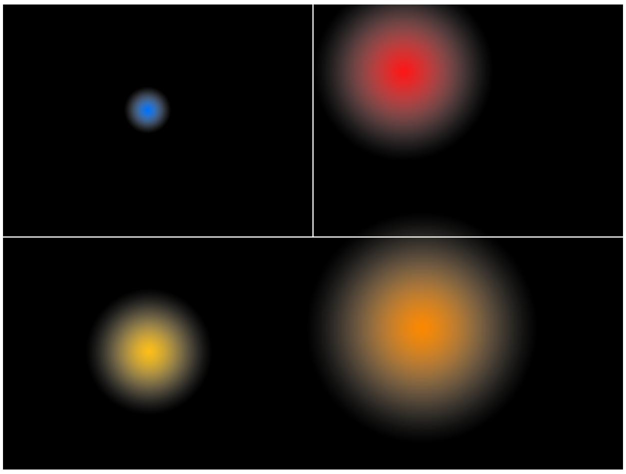

Figure 5. Screenshot of one of the percussion apps compatible with NUANCE. The blurry circles represent the strikes and their velocities (diameter).

card readers, like *Square*, use this technique, but this idea has also been used by the DIY community to send sensor data. For example, [16] is a simple modem that uses the *audio jack* connector of *Android* and *iOS* devices to transmit digital data. Its main limitation is its very small bandwidth (30 bytes/sec). [17] uses a different paradigm where the analog signals are multiplexed and sent one after the other.

Our approach described in the three previous sections is less versatile and more "low-tech" but it allows to stream the signals from four sensors to the *iPad* using the audio bandwidth. The consistency and rate of the information remain constant, greatly simplifying its synchronization with the sound synthesizer. It is also much cheaper and easier to build from scratch.

Its main disadvantage is that the demodulation technique (see §3) that it uses on the mobile device to retrieve the sensor data is rather computationally expensive. However the huge power of modern mobile devices compensates for this, and running the various band-pass filters, amplitude trackers, and the triangulation function presented in Figure 4 only takes 4% of the resources of the CPU on an *iPad Air 2*.

A more general limitation of our system is the fact that force can't be accurately measured on more than two independent simultaneous touches on the screen. We have not found this to be an issue in musical-performance applications to date. Perhaps only a built-in technology, such as 3D Touch, can efficiently resolve this issue. The fact that the most recent versions of *iPhones* address this problem leads us to think that larger devices such as the *iPad* will follow this trend too in future models. As mentioned in the introduction, it seems that tablet manufacturers prefer to settle for a force-sensitive pencil for now, which does not provide multi-touch force. Also, while the technology used by *Apple iPhone 6s* is fully multi-touch, it is not as fast as the method presented in this paper, and it can't be used for example to accurately detect the velocity of striking gestures.

6. FUTURE WORK

While the current version of NUANCE is quite stable and works well in its current state, the ability to power it through the *audio jack* using the technique described in [18], and used in [17], would be a good way to improve it.

Even though the technology used by NUANCE to communicate with the *iPad* is compatible with any other kind of mobile device (both *iOS* and *Android*), it is hard to find a design that adapts well to the size of any device in the same category. We would like to find a solution to this problem so that NUANCE can work with any types of *iPads* and *Android* tablet.

More generally, NUANCE was made in the frame of a larger scale project investigating the idea of "mobile device augmentation" and "hybrid lutherie" where our goal is to create a toolbox/platform to help design this kind of musical instrument.

7. CONCLUSIONS

NUANCE is a new device adding force-touch and velocity detection to the *iPad* touch screen. It communicates with the *iPad* using the *audio jack* input. Force signals are sent at an audio rate using analog amplitude modulation. NUANCE provides a high level of sensitivity and responsiveness by only using analog components. It costs less than $30 to make.

While NUANCE associates a different force value to simultaneous touch events happening on the screen, its ability to precisely distinguish the amount of force applied by different fingers decreases as the number of touches increases. This is probably its main limitation.

We think that adding this extra force dimension to the touch screen of the *iPad* significantly increases its expressive potential for controlling any musical instrument.

Acknowledgments

I would like thank my colleagues at CCRMA (especially

Fernando Lopez-Lezcano) for supporting the large amount of noise pollution that I generate.

8. REFERENCES

[1] A. Tanaka, "Mobile music making," in *Proceedings of the 2004 conference on New interfaces for musical expression (NIME04)*, National University of Singapore, 2004.

[2] G. Schiemer and M. Havryliv, "Pocket gamelan: tuneable trajectories for flying sources in mandala 3 and mandala 4," in *Proceedings of the 6th International Conference on New Interfaces for Musical Expression (NIME06)*, Paris, France, 2006.

[3] G. Essl, G. Wang, and M. Rohs, "Developments and challenges turning mobile phones into generic music performance platforms," in *Proceedings of the Mobile Music Workshop*, Vienna, Austria, May 2008.

[4] G. Essl, M. Rohs, and S. Kratz, "Use the force (or something) - pressure and pressure-like input for mobile music performance," in *Proceedings of the New Interfaces for Musical Expression Conference (NIME-10)*, Sydney, Australia, June 2010.

[5] G. Wang, J. Oh, and T. Lieber, "Designing for the ipad: Magic fiddle," in *Proceedings of the International Conference on New Interfaces for Musical Expression*, Oslo, Norway, May 2011.

[6] G. Wang, "Ocarina: Designing the iphones Magic Flute," *Computer Music Journal*, vol. 38, no. 2, pp. 8–21, Summer 2014.

[7] T. Martin, "The evolution of the smartphone," July 2014, http://pocketnow.com/2014/07/28/the-evolution-of-the-smartphone. Accessed: 2016-04-09.

[8] R. Michon, J. O. Smith, M. Wright, and C. Chafe, "Augmenting the ipad: the bladeaxe," in *Proceedings of the International Conference on New Interfaces for Musical Expression*, Brisbane, Australia, July 2016.

[9] T. H. Park and O. Nieto, "Fortissimo: Force-feedback for mobile devices," in *Proceedings of the International Conference on New Interfaces for Musical Expression*, KAIST, Daejon, Korea, May 2013.

[10] Z. Ren, R. Mehra, J. Coposky, and M. C. Lin, "Tabletop ensemble: Touch-enabled virtual percussion instruments," in *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games (I3D 12)*, Costa Mesa, California, March 2012.

[11] Y. Orlarey, D. Fober, and S. Letz, "An algebra for block diagram languages," in *Proceedings of the International Computer Music Conference (ICMC-02)*, Gothenburg, Sweden, 2002.

[12] R. Michon, J. O. Smith, and Y. Orlarey, "Mobilefaust: a set of tools to make musical mobile applications with the faust programming language," in *Proceedings of the Linux Audio Conference (LAC-15)*, Mainz, Germany, April 2015.

[13] J.-M. Adrien, "The missing link: Modal synthesis," in *Representations of Musical Signals*. Cambridge, USA: MIT Press, 1991, ch. The Missing Link: Modal Synthesis, pp. 269–298.

[14] R. Michon and J. O. Smith, "Faust-stk: a set of linear and nonlinear physical models for the faust programming language," in *Proceedings of the 14th International Conference on Digital Audio Effects (DAFx-11)*, Paris, France, September 2011.

[15] G. Loy, "Musicians make a standard: The midi phenomenon," *Computer Music Journal*, vol. 9, no. 4, pp. 8–26, Winter 1985.

[16] SwitchScience, "Audio jack modem for iphone and android," 2016, https://www.switch-science.com/catalog/364/. Accessed: 2016-04-09.

[17] S. Verma, A. Robinson, and P. Dutta, "Audiodaq: Turning the mobile phones ubiquitous headset port into a universal data acquisition interface," in *Proceedings of the Conference on Embedded Networked Sensor Systems (SenSys)*, Toronto, Ontario, November 2012.

[18] Y.-S. Kuo, T. Schmid, and P. Dutta, "Hijacking power and bandwidth from the mobile phones audio interface," in *Proceedings of the International Symposium on Low Power Electronics and Design (ISLPED)*, Austin, Texas, 2010.