Lessons Learned in Exploring the Leap Motion™ Sensor for Gesture-based Instrument Design

Jihyun Han
Department of Computer Science
University College London
UK
jihyun.han.10@ucl.ac.uk

Nicolas Gold
Department of Computer Science
University College London
UK
n.gold@ucl.ac.uk

ABSTRACT
The Leap Motion™ sensor offers fine-grained gesture-recognition and hand tracking. Since its release, there have been several uses of the device for instrument design, musical interaction and expression control, documented through online video. However, there has been little formal documented investigation of the potential and challenges of the platform in this context. This paper presents lessons learned from work-in-progress on the development of musical instruments and control applications using the Leap Motion™ sensor. Two instruments are presented: *Air-Keys* and *Air-Pads* and the potential for augmentation of a traditional keyboard is explored. The results show that the platform is promising in this context but requires various challenges, both physical and logical, to be overcome.

Keywords
Gestural control, Virtual instruments, Augmented instruments

1. INTRODUCTION
Gestural control for new musical instruments has been under active research for many years (e.g. see [11]). As platforms for gesture recognition advance, new opportunities arise to explore their potential and improve on the size, portability, and other aspects of new and augmented instruments. The Microsoft Kinect™ [10] sensor spawned many new applications in musical control (e.g. Yang and Essl [23]) and the recent Leap Motion™ controller [8] offers possibilities for finer-grained manipulation of musical parameters.

Instruments and control systems based on the Leap Motion™ are emerging in video (examples are described in section 2) but little formal documentation of the challenges and potential of the device in various scenarios exists.

This paper presents practical lessons learned during work to explore the potential of the Leap Motion™ for developing two instruments (Air-Keys and Air-Pad), and in gestural augmentation of a traditional keyboard. These lessons are documented to aid those considering using the platform in this context. The rest of the paper is structured as follows: section 2 reviews background, section 3 presents the two instruments developed in this project and evaluates them, section 4 discusses sensor orientation issues, and section 5 concludes.

2. BACKGROUND
2.1 The Leap Motion™ Sensor
The Leap Motion controller is a small (3" long [8]) USB device that tracks hand and finger motions as input. It is consumer targeted and intended to be used with minimal setup on a physical desktop, potentially offering an accessible means of controlling a virtual musical interface.

It works by projecting infrared light upward from the device and detecting reflections using monochromatic infrared cameras. Its FOV extends 25mm to 600mm with a 150^o spread from the device and it has high framerate (>200fps) and precision (1/100mm per finger) [8].

2.2 Gestural Control of Instruments
Gestural instruments have been developed over many years (see Miranda and Wanderley [11] for a comprehensive treatment of the subject), and new developments (e.g. using cameras [13,14], motion-capture [15], or gloves [12]) are regularly reported. More recently the advent of consumer-level gesture-capture devices like the Kinect™ has led to further developments in the area, e.g. [4, 22, 23].

The recent release of Leap Motion™ has likewise produced many examples of new instruments, largely documented in video alone (e.g. see [7]). One notable exception is the work of Silva et al. who undertook an evaluation of the sensor in the context of digital music instruments (in similar vein to this paper) [17]. In their case, they used visual and tactile feedback to aid the performer in locating their hand position and were able to measure latency using the difference between the timing of sounds recorded on the tactile surface and note triggering by the sensor. Our work evaluates the platform from the perspective of instruments using *touchless action* [18].

These Leap-based instruments can be can be classified using Miranda and Wanderley's framework [11] thus:

2.2.1 Augmented Musical Instruments
These are acoustic/electric instruments augmented by sensors [11]. Soylentcola augmented a guitar with the Leap Motion™ sensor attached to the body to control effects [20]. Expression parameters appear to be controlled by the position of the detected hands in a very practical way. Earnshaw attached the sensor to a drumkit to create a drumstick-controlled theremin-like instrument [1]. Raz enhanced a traditional piano, controlling electronics through hand gesture to combine acoustic and electronic sounds [7]. Bertelli enhanced a snare drum, using the sensor (in combination with other controllers) to manipulate tempo, timbre, and envelope [7]. Thompson's Motion-Controlled Instrument receives chords from a midi controller, and plays them based on timing quantisation, pitch and velocity controlled in the x, y, and z dimensions respectively [21]. These augmented instruments work reasonably well and for timbral control with distinct gestures, the sensor works effectively.

2.2.2 Instrument-like Gestural Controllers
These are gestural controllers modelled after acoustic instruments [11]. Gratoo created an air drum played with sticks [5]. There is noticeable latency between a 'hit' and the sound triggering, suggesting that the LEAP API's built-in key-tap gesture recognition is being used to trigger MIDI messages. If this is the method used, it successfully detects taps, but does not offer velocity sensing.

Heaver constructed an Air Piano [6] using 3 dimensional positioning of the fingers to determine pitch, velocity, and the after-

NIME'14, June 30 – July 03, 2014, Goldsmiths, University of London, UK.

touch of the notes. It appears to suffer from the sensor's difficulty (see later) in tracking fingers that are close together thus sometimes missing notes or falsely activating. Sensomusic (who created a PianoLeap instrument as well as a theremin-like instrument) [16] appears to have adopted a heuristic solution to recreate 'valid' chords, based on the number of detected fingers. The pitch and volume are controlled by the position of the hand. Pianoleap also seems to have a fail-safe design which restricts the available notes to a certain scale, and quantised timing of note activation. The sensor has also been used to create an Air Harp [19], which provides direct interaction with the virtual reconstruction, appearing very responsive and with simple note layout. It also provides selectable scales and expression controls.

2.2.3 Alternate Controllers

These are controllers that do not bear strong resemblance to existing instruments [11] (or in these cases, are often separate gestural interfaces used with other instruments or hands). Earnshaw created an expression controller used with a guitar where the movement of the headstock controls effect parameters [2]. Yehezkel uses the controller to perform dubstep [24] using the hand position to control effects, and Fujimoto uses it for beatboxing [3] using gestures to shape and trigger sounds. Voy and Sqeepo both control various aspects of their synthesised sound through gesture [7], in the latter case creating a particularly intuitive interface in the context of the musical genre. Silva et al. [17] present the Crystal Piano which combines Leap-based gesture recognition with a clear physical surface and visual feedback to guide the user. They observe similar issues to those found in our evaluation here in terms of occasional mis-tracking and lost fingers.

3. INSTRUMENTS

3.1 Air-Keys

This section presents the Air-Keys and the Air-Pads we developed to explore the challenges involved in re-creating a traditional instrument using gestural technology. We examine key issues in turn.

3.1.1 Recognition of key-presses and key-releases

Recognising key presses is fundamental to the operation of the Air-Keys. The Leap Motion™ SDK's way of providing the user with tracking data is done through the 'onFrame' method. This is called on every frame and the manipulation of tracking data is performed inside the method.

Our first approach to detect key-presses used the SDK's built-in support for gesture recognition. The 'key-tap' gesture, where user instantly taps downward and recoils upward was used. While it was simple to make the program trigger the MIDI message on the key-tap, the latency between the user's intention and the trigger was too high for instrumental use as the sensor needs to see the fingers recoil upwards to recognise it as a key-tap. From a MIDI perspective, a single gesture makes appropriately timed creation of note-on and note-off messages more difficult.

The next approach utilised the SDK's ability to track the physical characteristics of fingers. We hypothesised that when the user intends to press a key, the downward velocity of the fingers will be higher than when they move their hand for other reasons. We measure the downward velocity of the fingers in comparison to a threshold, and when it is exceeded, this triggers a note-on message. Likewise, upward velocity and note-off are calculated from upward finger velocities. The problem with this approach is that the methods are called on every frame so MIDI messages were being triggered for every consecutive frame where the fingers moved faster than a certain speed (thus a de-bounce was required).

Using the SDK's ability to refer to the previous frame, the positions of those fingers present in the current frame are compared on the y-axis with those in the previous frame. If a finger was positioned higher than a set threshold in the previous frame, but is currently lower than the threshold, the program triggers the note-on message and vice versa for note-off messages. This creates an imaginary horizontal 'trigger-plane' which makes a sound when fingers pass it.

3.1.2 Recreating the Keyboard Layout

We model the layout of the keyboard through a single class holding information for the positions of the different keys (measured in millimetres). Keys are instantiated from note 41 (F2) to 79 (G5) with note 60 (C5) as the middle. Keys are initially 2 cm wide but key spacing can be dynamically changed.

When the condition for a message trigger is successful, i.e. a finger moves past the vertical threshold, the x-axis value of the position of the finger is used to determine which key has been pressed.

3.1.3 Calculation of Note Velocity

Note velocity is somewhat unintuitive in an air keyboard since there is no physical feedback in relation to the striking force on the 'key'.

To create velocity sensitivity a direct mapping is made between the speed of the finger and the velocity of the note. When the note trigger condition succeeds, as the x position of the finger is identified, the speed of the finger at the moment of trigger is calculated between 0cm/s to 300cm/s and converted to a MIDI velocity value.

This solution lets the player control the loudness of the sound by hitting the notes at higher speed, akin to hitting the keys hard.

3.1.4 Hardware-Related Characteristics

The characteristics of the Leap Motion™ device also affect the performance of the instrument.

Consider a case where a finger has 'pressed' the key at C4. A note-on message for C4 is sent. If, while depressed below the threshold, the finger is moved horizontally and then lifted at a different note, for example, D4, the note initially triggered will still be playing and a meaningless note-off message will be sent to D4. This problem requires the instantiation of a hand model to track finger IDs and map their correspondence to notes. Fingers are added to the map as soon as notes are triggered and are removed as the finger leaves the touch zone. When they are removed, a note-off message is sent for the appropriate note number.

Consider another case where a finger is depressed at C4, but gets 'lost', i.e. disappears in the sensor's field of view after the note-on message was sent. Once the tracking is lost for a finger, the sensor has no way of regaining the ID of the lost finger and gives a new ID to the finger when it reappears in the field. This causes two problems: the note that was pressed by the lost finger is stuck on and has no way of turning off, and the map entry for the lost finger has no way of being removed. Overcoming this requires a frame-by-frame verification of the map to the currently detected fingers, and where fingers in the map are no longer detected, a note-off message is sent to the appropriate note and the map entry removed.

Finally, the sensor's high sensitivity (beneficial in many situations) can allow false detection of near-stationary fingers when they are close to the note-triggering plane. Addressing this simply requires a minimum-speed threshold to be exceeded (currently 1cm/s).

3.2 Air-Pads

We applied the same base mechanism and algorithm to recreate a 4x2 button array drum-pad. We decided to loosen the concept of fine-grain control and make each pad (key) comparably larger than the Air Keys trigger zones. Each pad triggers different note values responsible for kick drum, snare, closed hi-hat, and so on.

The implementation of the Pad Layout is a further modification of the aforementioned Keyboard Layout. Instead of only considering the x axis as the division of keys, we took the z axis into account to separate each pad. The controller class passes both x and z values to the pad layout class on each successful trigger. Each pad is 10cm x 10cm and they are laid out in 2 rows, each containing 4 pads.

3.3 Evaluation and Discussion

The prototype of the Air-Keys and the Air-Pads were then evaluated by the first author in a semi-structured way. For the Air-Keys, The Celebrated Chop Waltz (informally known as the chopstick song) was played to evaluate the practicality of the keys for melodic use. Basic triad chords of Pachelbel's Canon in C were played for the chordal evaluation. Furthermore, a simple drum pattern was played to evaluate the practicality of the drum pads. Each trial was undertaken twice: before practice and after ten minutes of practice. Table 1 shows the observations made.

Table 1: Observations of instrument playability

Criterion	*Observations*
Sense of key positioning (prior to melodic and chordal evaluation)	Once a sense of the position of middle C was established, running from C to G was not very difficult (5/10 trials error-free). When attempting to play stepwise from middle C to the C an octave above, passing the thumb under the hand to play above G (i.e. shifting the thumb to land on F to play F to C fluently with 5 fingers) was not recognized at all (0/10 trials error-free). As the keys are not physical, it was difficult to get the sense of height (threshold). One has to strike down the keys in a wide vertical angle to ensure the threshold is passed.
Melodic performance	In the initial trial, there was a lot of confusion in where the keys were, and how high they were. Key spacing is not easily predictable (e.g. E to C) thus non-stepwise jumps can be hard to play (an average of two to four consecutive mistakes after most non-stepwise jumps). After ten minutes of practice one's sense of the keyboard layout was improved (an average of two mis-locations per phrase containing approximately eight non-stepwise jumps) and after getting used to the threshold height, the velocity of the notes was more easily controllable (increased control after practice corresponds to the reported experience of Bertelli [7]). Flatter hand shapes performed better.
Chordal performance	As with the melodic performance, the initial sense of key layout was poor (finding note locations took three to eight attempts and triad shapes took five to ten attempts). The piece used (Pachelbel's Canon) involved a lot of non-stepwise movement, and was thus far more difficult than the melodic performance. It was harder to get the device to recognize the fingers as they were close together. After ten minutes of practice, in contrast to the melodic performance, hand-finger shape remained stationary while the arm and the wrist moved to press a typical triad chord. Once familiar with the recognized hand shape and the width of keys, it was easier than the initial attempt (two to five attempts for location, with triads taking three to five attempts). However, in a realistic situation, the hands and fingers do not stay stationary nor are they flat, thus it was less practical with triad chords. Expression control of chords using forward/backward movement was quite playable. Once the chord was found, only three fingers were lost from tracking in twenty trials of ten seconds.
Drum pad performance	Adjusting to the buttons was significantly easier than the Air Keys, even at the initial attempt (average of two errors per bar). Trigger threshold confusion was less of a concern, as a typical pad performance involved wider vertical swings at the pad. After ten minutes of practice, a simple drum pattern and variations were easily and fluently performable (maximum two errors per bar). The larger pad area improved the calibration and practicality of the instrument, however, when a finger was struck down to activate a pad, we occasionally observed the nearby fingers mistakenly activating the same pad.

4. ORIENTATION

Our final investigation involved placing the sensor at various angles to the table to investigate its potential in non desk-bound situations.

The Leap Motion™ sensor is designed to be used on a desk facing upward with the hands held over it. Object 3D realisation is undertaken through projection from the device upwards, so vertically-oriented objects are sometimes not correctly recognised. Due to its reliance on reflected infrared, objects such as transparent pens are also not very well seen by the device.

Despite the sensor's intended desk-based usage, there are examples of its use mounted on the body of guitars (e.g. [20]) and the potential for alternative orientations was explored here, with a view to augmenting traditional keyboard instruments (motivated by the work of McPherson et al. [9] but without requiring custom electronics and thus in similar vein to Yang and Essl [23]). Table 2 reports the observations made during informal evaluation.

Table 2: Observations of orientation on sensor performance.

Orientation	*Observations*
Upright (see Fig. 1(a))	This orientation works the most fluently but when a hand and fingers are positioned as for piano performance, the fingers are sometimes not recognised properly as the fingers point downwards and are too close together (an average separation of about 5mm is needed). Fingers remain tracked when stationary or with slow bending.
Rotated 90 degrees about y axis (see Fig. 1(b))	Fingers are not fluently recognised, with detection briefly failing on average three times a second with a stationary hand. Palms are recognised but appear to change direction frequently and disappear about 40cm above the device. 'Pointables', such as a pencil, are recognised until they are completely parallel to the x axis (mirroring the situation with the y-axis when the device is in its normal orientation.
Sensor held upside-down above the hands	Hands and fingers are not recognised unless a plane of solid material is placed above the sensor to block overhead lighting (or room lighting is switched off) in which case the sensor works close to normally but with inverted axes (about two finger mis-tracks every five seconds with a stationary hand).
Sensor placed on its side (see Fig. 1(c) and (d))	Vertical (i.e. in the direction of projection) modelling is weak. As the device is projecting the infrared horizontally towards the user, the fingers are not correctly recognised if they are placed flat on the table. Fingers are barely recognised even if upright (perhaps because of early reflection of the infrared light from the table surface since the sensor performs better when placed on the edge of a table)
Sensor at an angle (see Fig. 1(e)).	Works as normal but pianistic hand shapes can still be hard to detect. Around five mis-tracks of the fingers and two of palms every ten seconds.

We also experimented by placing the sensor at the back of a MIDI keyboard to control various timbral parameters based on sliding the hand back and forth on the keys while holding down a chord. Whilst there was some success in this, the other keys and potentiometers occluded the sensor's view of the hands and thus further work will be needed to determine the best placement and heuristics required.

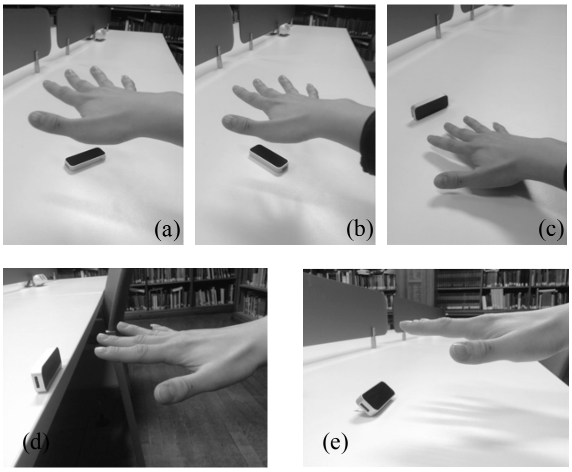

Figure 1: Views of sensor orientation (a) normal, (b) rotated about y axis, (c) on its side, (d) side, edge of table, (e) at an angle.

5. CONCLUSIONS AND FUTURE WORK

This paper has presented lessons learned from work in progress to use recent consumer-level gesture detection technology as means for creating and controlling instruments. Two variants of a virtual instrument were presented: Air-Keys and Air-Pads, both using the Leap Motion™ sensor for control. Observations derived from informal evaluation of these instruments, and using the sensor for augmenting a traditional instrument, were presented.

Future work will include the development of finger-specific adaptive thresholds (rather than a flat plane) based on individual finger velocity to account for more natural pianistic hand shapes, and heuristics for note prediction and triggering in common technical scenarios e.g. thumb passing under the hand in an upward run.

6. ACKNOWLEDGMENTS

This work is partially supported by the EPRSC CREST Platform Grant [grant number EP/G060525/2]. RCUK-funded data may be obtained from Nicolas Gold. Leap Motion™ is a trademark of Leap Motion Inc. Kinect™ is a trademark of Microsoft Inc.

7. REFERENCES

[1] N. Earnshaw. Drums + Leap Motion Theremin experiment. *http://www.youtube.com/watch?v=4ZG9V0MNeh4 (accessed 7th Feb 2014).* 2014.

[2] N. Earnshaw. Building Bridges: Exploring the Musical Interface Gap. *https://labs.leapmotion.com/building-bridges-exploring-the-musical-interface-gap/ (accessed 7th Feb 2014).* 2014.

[3] R. Fujimoto. Humanelectro × Leap Motion. *http://www.youtube.com/watch?v=p9EyDes62sc (accessed 7th Feb 2014). 2013.*

[4] N. Gillian, and J. Paradiso. Digito: A Fine-Grain Gesturally Controlled Virtual Musical Instrument. *In Proceedings of the International Conference on New Interfaces for Musical Expression (NIME)* (Ann Arbor, Michigan, May 21-23, 2012).

[5] gratoo. The Geek Tune (Rocking the Leap Motion). *http://www.youtube.com/watch?v=xp6v96DhzwU (accessed 7th Feb 2014).* 2013.

[6] M. Heavers. Leap Motion Air Piano. *http://vimeo.com/67143314* (accessed 7th Feb 2014). 2014.

[7] Leap Motion Inc. 6 Songs in the Key of Leap. *http://blog.leapmotion.com/post/60865789820/6-songs-in-the-key-of-leap-motion* (accessed 7th Feb 2014). *2013.*

[8] Leap Motion Inc. *The Leap Motion*. *https://www.leapmotion.com/product*. (accessed 7th Feb 2014). 2014.

[9] A. McPherson, and Y. Kim. Design and applications of a multi-touch musical keyboard. *In Proceedings of the Sound and Music Computing Conference* (Padova, Italy. July 6-9, 2011).

[10] Microsoft Inc. *Kinect for Windows*. *http://www.microsoft.com/en-us/kinectforwindows/ (accessed 7th Feb 2014).* 2013.

[11] E.R. Miranda, and M.M. Wanderley. *New Digital Musical Instruments: Control and Interaction Beyond the Keyboard.* A-R Editions, Inc., Middleton, Wisconsin, 2006.

[12] T. Mitchell, S. Madgwick, and I. Heap. Musical Interaction with Hand Posture and Orientation : A Toolbox of Gestural Control Mechanisms. *In Proceedings of the International Conference on New Interfaces for Musical Expression (NIME)* (Ann Arbor, Michigan, May 21-23, 2012).

[13] P. Modler, and T. Myatt. Video Based Recognition of Hand Gestures by Neural Networks for the Control of Sound and Music. *In Proceedings of the International Conference on New Interfaces for Musical Expression (NIME)* (Genova, Italy, June 5-7, 2008).

[14] O. Nieto, and D. Shasha, Hand Gesture Recognition in Mobile Devices: Enhancing The Musical Experience. *In Proceedings of the 10th International Symposium on Computer Music Multidisciplinary Research (CMMR)* (Marseille, France, October 15-18, 2013).

[15] K. Nymoen, S. Skogstad, and A. Jensenius. Soundsaber-a motion capture instrument. *In Proceedings of the International Conference on New Interfaces for Musical Expression (NIME),* (Oslo, Norway, May 30 – June 1, 2011).

[16] sensomusic. sensomusic usine & leap motion air piano and theremin. *http://vimeo.com/77346807 (accessed 7th Feb 2014).* 2013.

[17] E.S. Silva, J.A.O. de Abreu, J.H.P de Almeida, V. Teichreib, and G.L. Ramalho. A Preliminary Evaluation of the Leap Motion Sensor as Controller of New Digital Musical Instruments. *In Proceedings of SBCM - Brazilian Symposium on Computer Music,* (Brasilia, Brazil, October 31 – November 2, 2013).

[18] S. Skogstad, A. Jensenius, and K. Nymoen. Using IR optical marker based motion capture for exploring musical interaction. *In Proceedings of the International Conference on New Interfaces for Musical Expression (NIME),* (Sydney, Australia, June 15-18, 2010).

[19] A. Somers, AirHarp Demo. *http://www.youtube.com/watch?v=U2HFcfOwDtI2012 (accessed 7th Feb 2014).* 2012.

[20] soylentcola. First experiment with Leap Motion and guitar. *http://www.youtube.com/watch?v=1cTI2GWlHY8* (accessed 7th *Feb 2014).* 2013.

[21] T. Thompson. Experimenting with the Leap Motion device as a musical instrument. *http://www.youtube.com/watch?v=KGxo2A7VXuc (accessed 7th Feb 2014)*, 2012.

[22] S. Trail, T.F. Tavares, G. Odowichuk, P. Driessen, W. A. Schloss, and G. Tzanetakis. Non-invasive sensing and gesture control for pitched percussion hyper-instruments using the Kinect. *In Proceedings of the International Conference on New Interfaces for Musical Expression, (NIME)* (Ann Arbor, Michigan, May 21-23, 2012).

[23] Q. Yang, and G. Essl, Augmented Piano Performance using a Depth Camera. *In Proceedings of the International Conference on New Interfaces for Musical Expression (NIME)* (Ann Arbor, Michigan, May 21-23, 2012).

[24] U. Yehezkel, Controlling music with Leapmotion Geco & Ableton. *http://www.youtube.com/watch?v=32iBfZVY_68 (accessed 7th Feb 2014).* 2013.