AirPiano: A Multi-Touch Keyboard with Hovering Control

Nicolas d'Alessandro, Joëlle Tilmanne, Ambroise Moreau, Antonin Puleo
numediart – Institute for Creative Technologies
Faculty of Engineering, University of Mons, Belgium
{firstname.lastname}@umons.ac.be

ABSTRACT

In this paper, we describe the prototyping of two musical interfaces that use the LeapMotion camera in conjunction with two different touch surfaces: a Wacom tablet and a transparent PVC sheet. In the Wacom use case, the camera is between the hand and the surface. In the PVC use case, the camera is under the transparent sheet and tracks the hand through it. The aim of this research is to explore hovering motion surrounding the touch interaction on the surface and include properties of such motion in the musical interaction. We present our unifying software, called AirPiano, that discretises the 3D space into 'keys' and proposes several mapping strategies with the available dimensions. These control dimensions are mapped onto a modified HandSketch sound engine that achieves multitimbral pitch-synchronous point cloud granulation.

Author Keywords

Touch Interaction; Motion Capture; Depth Camera; LeapMotion; Musical Instrument Design; Musical Gestures.

ACM Classification

H.5.2. Information Interfaces and Presentation (e.g. HCI): User Interfaces: Input Devices and Strategies

1. INTRODUCTION

Multitouch sensors and depth cameras are two of the major interaction technologies that made their way into consumer products over the last decade. Multitouch screens have reshaped our approach towards contents and our way of designing UIs. However Bret Victor pointed out that this whole ecosystem still uses a *picture under glass* metaphor[1], quite far from anything really tangible or embodied. HCI research is very active in developing the interactive surface form factor much further, for instance with pressure sensing[2], additional objects [5], acoustic sensing [6] or multitouch sensing on piano keys [8]. Depth cameras combined with the realtime estimation of full-body skeletal data [13] have brought a very affordable and hackable technology for motion analysis. Companies like LeapMotion or PMD are now proposing hand skeletal tracking with short-range cameras. Hand tracking from the Kinect point cloud is also an active research area [10]. However this other ecosystem[3] is very focused on touchless mid-air interaction. Pointing these 3D cameras at surfaces for studying contact interaction often results in severe performance reduction.

One could conjecture that a touch gesture is always a combination of mid-air and contact phases. Indeed the hand first reaches towards the surface. Then the touch is actually realised, through a series of contact points (x_i, y_i) on the touch screen and eventually the pressure values p_i. Finally the hand moves away from the surface. These two surrounding mid-air gestures are often called *pre-* and *post-movements*[4]. This perspective appropriately fits into Luciani's theoretical framework about tangible interaction [7] and the overall idea that any gestural action can be understood from several proximity-related viewpoints. Particularly in this case, we can argue that contact interaction in musical performance is a repeated alternance and subtle balance between *ready-to-hand* (nearly touching the surface) and *present-in-hand* (actually touching the surface) scenarios. Following Luciani's approach, this corresponds to a topological discontinuity around the object, between the sense of space and the sense of matter.

The predominance of interaction paradigms suggested by new hardware – swiping fingers on multitouch screens and waving hands in front of depth cameras – and the intrinsic cognitive discontinuity between mid-air and contact gestures might explain why touch and touchless musical interfaces have so often been studied as two different categories. However there is now some evidence that pre- and post-movements are mentally modelled by the performer as part of the actual musical gesture, and not as separate actions [11]. Moreover we think that the necessary gesture tracking technologies are mature enough in order to design affordable musical instruments where the set of available gestures include both mid-air and contact interaction components.

In this paper, we present two configurations combining a touch surface and the LeapMotion 3D tracking of the hand in the hovering space, i.e. the mid-air space over the surface. The first prototype, described in Section 2, uses the LeapMotion *between* the hand and the surface, in this case a Wacom tablet. The second prototype, described in Section 3, takes advantage of the property of polyvinyl chloride (PVC) sheets not to disturb the infrared-based tracking of the LeapMotion. Therefore the depth camera can be placed *under* the active surface and enables the *see-through* tracking of touch gestures. In Section 4, we present the software developed for the calibration of these interfaces and the mapping of finger gestures to sound. Finally we give some conclusions and further work.

---

[1] `http://tinyurl.com/bretvictor`

[2] `http://www.rogerlinndesign.com`

[3] LeapMotion provides a marketplace, called AirSpace.

[4] Also sometimes called *preparation* and *release* movements.

NIME'15, May 31-June 3, 2015, Louisiana State Univ., Baton Rouge, LA.

2. PROTOTYPE 1: WITH WACOM

The first interface prototype aims at extending a regular Wacom tablet with hovering control. Our constraints on the design are dictated by the need for the tablet to remain portable and playable in various postures, including vertical. Indeed much of our work with the Wacom tablet is achieved in the context of the HandSketch project [2] which exhibits unusual playing postures like vertical or sideways tablet holding. There the LeapMotion requires to be attached to the body of the Wacom tablet and keep the overall form factor of the instrument as flat as possible.

The LeapMotion is able to stably track the performer's hand if it sees in from below, i.e. pointing at the palm of the hand with fingers distinctively visible. The other important aspect to take into account is the presence of infrared light reflections within the scene captured by the camera. Indeed pointing the LeapMotion at the Wacom surface creates visible direct reflections of the infrared light source and reflections of the hand in the tablet. These reflections disturb the 3D hand tracking. Recently LeapMotion has enabled the access to the raw images from the camera in their SDK. This feature has allowed us to place the camera in order to bring these reflections out of the captured scene, while still having the full hand tracked during touch interaction. For the Wacom *Intuos Pro L* model, the best results are obtained with a 50-degree angle between the camera socle and the tablet, and a 7-cm distance between the camera center and the edge of the active surface. This setup is depicted in Figure 1 with a view (right) of the prototype.

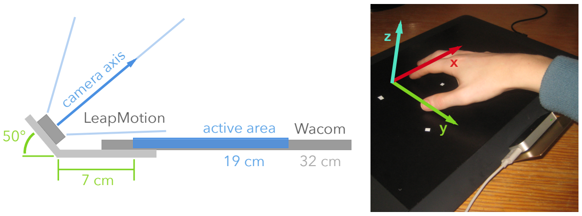

Figure 1: Lateral view (left) and picture (right) of the prototype: the LeapMotion is placed under the hand with a) a 50-degree angle between the camera socle and the tablet, b) a 7-cm distance between the camera center and the edge of the active surface.

It is interesting to highlight that this *oblique lighting* configuration keeps working no matter the orientation of the tablet. It means that the 3D hand tracking can be efficiently used for vertical HandSketch playing postures [2]. This observation extends to the tool tracking feature of the LeapMotion software, meaning that the pen position and orientation can be tracked before touching the surface.

This configuration has also been informally tested with other control devices than a touch surface, like pipes or keyboards, and works similarly. It always consists in finding the camera angle and distance that will bring direct and secondary reflections out of the frame while having fingers coming right at the edge of the frame when the object is touched. These early experiments are very encouraging in the context of extending various musical interaction scenarios with a short-range 3D camera, so as to integrate expressive hovering gestures into the musical instrument.

3. PROTOTYPE 2: WITH PVC

The second interface prototype converts a fixed transparent PVC sheet into the actual interactive surface. This operation is achieved by placing the LeapMotion under the PVC sheet, in a *see-through* configuration. The whole multitouch research community is very familiar with see-through infrared tracking, thanks to projects like the Reactable [5] or the SecondLight surface [4]. Regarding this context, our goal is to propose a low-cost, accessible and customisable musical controller which directly benefits from all the LeapMotion improvements in 3D hand tracking.

Thin polyvinyl chloride (PVC) sheets (here 4 mm) seem not to be interfering with the infrared lighting that is used by the LeapMotion camera. Therefore the PVC sheet is nearly invisible to the LeapMotion, the light reflections are limited and we only loose $0,05$ to $0,1$ in the API *data confidence* measure. This conclusion comes after a series of experiments achieved with various transparent materials.

From this idea that a 4-mm PVC sheet can be our touch surface, we have explored various designs so as to optimise the angle with which the LeapMotion camera points at the hand touching the surface. The LeapMotion SDK exposes a *confidence value* for the tracking, enabling us to perform such optimisation. It turns out that a 50-degree angle between the PVC sheet and the camera axis of the LeapMotion gives the best tracking. With this angle, the LeapMotion correctly tracks on a nearly 30 cm × 30 cm touching area. Pre- and post-movements start to get tracked correctly by the LeapMotion around 40-50 cm above the PVC sheet.

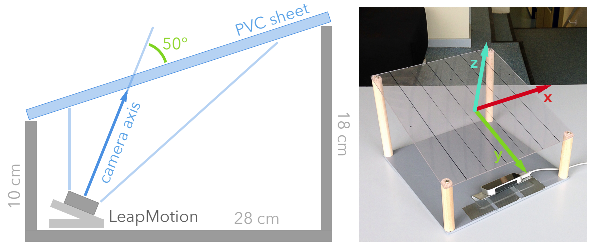

Figure 2: Lateral view (left) and picture (right) of the prototype: the LeapMotion camera axis forms a 50-degree angle with the orientation of the PVC sheet, enabling good see-through hand tracking.

In order to keep the box reasonably small, the 50-degree angle has been shared between the inclination of the PVC sheet (28 degree) and the inclination of the camera (22 degree) relatively to the table. Figure 2 shows the lateral view (left) and an overall picture of the prototype (right).

4. SOFTWARE: AIRPIANO

We have decided to create a generic and modular software so as to handle our two prototypes, called *AirPiano*. Indeed our interaction paradigm is about augmenting touch gestures with their corresponding hovering properties and we think that this mapping strategy should be decoupled from solving the issues related to the camera placement. Therefore our software is able to handle any camera placement relative to the touch surface (however respecting the visibility guidelines presented in Section 2) and then exposes a normalised set of control dimensions for the mapping of finger gestures to sound. In the following subsections, we describe the four components of this software: the LeapMotion OSC streaming, the calibration and coordinate change, the mapping of finger gestures and the sound synthesis.

4.1 OSC streaming

We use the LeapMotion SDK to capture hand parameters. Since version 2, the LeapMotion SDK provides a skeletal model of the hand, i.e. the position and orientation of every bone of the 3D hand model fitted to the user's hand, including the fingers, the palm and the forearm. Using the C++ API, we have created a lightweight OSC streamer for Mac and Windows. The streamer extracts all the joint positions (x_i, y_i, z_i) from each captured frame i and sends them through OSC to the following module.

4.2 Calibration

The coordinate system of the SDK puts the origin at the top of the LeapMotion, x and z are respectively parallel and perpendicular to the long edge of the device and y is the camera axis. However, for our prototype to respond like a touch surface, the x and y axes of the new coordinate system have to be respectively aligned with the horizontal and vertical axes of our touch surface: the Wacom tablet and the PVC sheet, respectively for prototypes 1 and 2. Moreover the z axis has to be an image of the distance between fingertips and that touch surface, i.e. aligned with its normal vector. Right parts of Figures 1 and 2 illustrate this new coordinate system on the top of the prototype.

The coordinate change is achieved by multiplying the incoming 3D data by a 4 × 4 transformation matrix at runtime. The matrix is computed from the equation of plane corresponding to our touch surface in the LeapMotion coordinate system. Three points from that plane are obtained by resting one fingertip at three different locations on that touch surface during the calibration phase.

4.3 Mapping

Many experiments with theremin-like instruments and free movements in a 3D space have shown that there is nothing easy in recalling an arbitrary 3D position and using that as a musical gesture. We think that aiming at various elements on the surface with the fingers is an efficient way of anchoring the hand motion in space and helping the performer with a visual reference. In this project, we have chosen a piano-like strategy. It means that the x axis values after the coordinate change are discretised into several *keys* and those keys are drawn onto the surface. It appears that 6 keys work best for the Wacom, 7 for the PVC sheet. More generally, we think that the optimal number of keys depends on the width of the surface (available range for x values) and the performer's ability to aim at one key from far above the surface. Indeed, while standing right in front of the surface, it was empirically observed to be manageable to aim at the targeted x range starting from 40-50 cm above the surface on which these boundaries are drawn.

The second mapping mechanism that has been implemented consists in detecting which fingers of the whole hand are intentionally pointed down towards the surface in order to trigger sounds. Indeed, while the LeapMotion SDK version 1 enabled fingers to 'disappear' from the scene by folding them on the palm, SDK version 2 always estimates the whole hand configuration. However, in a piano-playing metaphor, only a subset of the fingertips are used to produce sound at a given time. Our approach to solve this problem is to aim at grouping the fingertips according to their z values after coordinate change, i.e. their distance to the surface. We measure distances along z for all fingertip pairs. Using an ajustable *tolerance* on these distances, we can decide to either keep all fingertips in the same group or create a second group with some fingertips being too far away from the first group. If we find 1 to 3 fingertips in a second group lower than the first group, we mark these fingertips as *active*, i.e. they will produce sounds. The higher group contains *inactive* fingertips that are passive in the mapping. This fingertip grouping is illustrated in Figure 3.

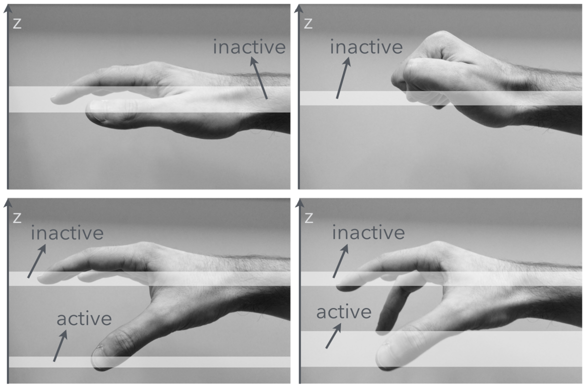

Figure 3: Fingertip grouping principle in order to detect active fingerings, i.e. the ones that will trigger sounds: the lower second group of fingertips is categorised as active and the others as inactive.

When each active fingertip has been identified in a given discretised key, all the other finger position properties can be used in the sound-making process. The LeapMotion SDK gives access to many aspects of finger postures: finger flexion, fingertip direction, finger speed, etc. In these two first musical instruments, we have simply exposed the y and z values of each active fingertip after coordinate change. The z value brings the proximity with the touch surface as an expressive parameter. Considering the keyboard drawn onto the surface, the idea of *getting closer to a given key* is musically very valuable for sound generator or effect intensity. Van Nort *et al.* have shown that sound intensity intuitively correlates with a position in space [9]. Moreover this intensity mapping based on gesture depth is also the strategy used in the HandSketch diagram, where the performer has to reach further on the playing fan to produce more tensed vocal folds values [2]. The y axis proposes to the performer that each key behaves like a fader, moving fingertips up and down, which also has a very visual aspect and is easy to recall. However the y value is harder to aim from far above the surface due to the performer's viewing angle, both with the Wacom tablet and the PVC sheet.

4.4 Synthesis

In this work, we have modified the HandSketch sound engine for the handling of multitimbral synthesis. As described in [1], the HandSketch sound engine is a pitch-synchronous overlap-add (PSOLA) synthesiser working with a point cloud of sound grains in a visual space (2D or 3D). The position of each grain in that space is based on descriptors extracted on the grain. The approach is similar to CataRT [12], but completed with PSOLA-specific sound descriptors such as the inharmonicity of the residual or the ratio between causal and anti-causal formant frequencies [3].

The sound engine comes as a Max patch where the user can load any sound into the system (PSOLA requires sound to exhibit a fundamental frequency). Then the user can access simplified control dimensions and assess the presence of grains along these dimensions with a 2D or 3D visual representation of the point cloud. From HandSketch, this point could synthesis engine was monophonic but we optimised it so as to enable up to 8 tracks to play at the same time. It means that each track can now be a separate point cloud corresponding to a different set of sounds. Different tracks can also tap into the same point cloud, using the multi-probe feature of the browsing tool. The 3D point cloud of one track is illustrated in Figure 4.

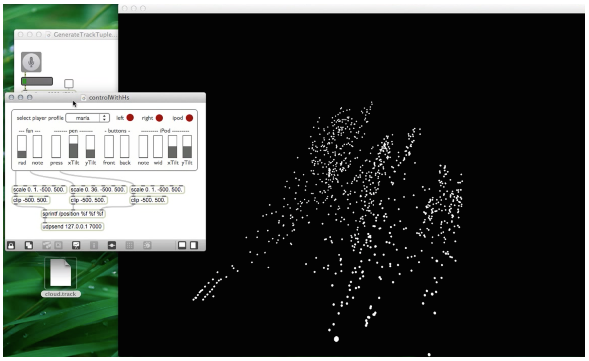

Figure 4: Screenshot of the modified HandSketch sound engine: one track is displayed with its corresponding 3D point cloud of sound grains.

5. EXPERIENCE

These two musical instrument prototypes have already been used in several performances and demonstrated to various audiences: children, non-musicians, trained musicians. In this Section, we describe some of the compositional choices that have been made in order to create a more interesting musical experience for the users.

For the first proof of concept, we decided to actually realise the AirPiano concept by using piano sounds. We used several piano samples at different pitches. Aiming at one key on the instrument selects one note among the others. Piano sounds can be re-synthesised with a very good quality using the HandSketch engine. We used the progression over the initial sound timeline as one dimension to be mapped. However we wanted the sound to be louder as we get closer to the surface. So we mapped this time index in the sound file backwards: the end of the file when z is high, the beginning of the file when z is low. The percussive nature of piano sounds gives a nice increase of intensity when we reach the beginning of the file, i.e. when we get close to the surface. The y values have been mapped to different effects. Among them, increasing the reverb for low y values (hand getting further from the body) seemed to create a quite intuitive metaphor: further hand, further sound and users were intuitively understanding this aspect.

6. CONCLUSIONS AND FUTURE WORK

This paper aimed at demonstrating the use of the LeapMotion in two original setups, tracking the hand in conjonction with a Wacom tablet and through a transparent sheet of PVC. Such prototypes enable the capture of hovering motion around touch gestures, hence studying pre- and post-movements associated with these gestures. This project was our first attempt to bridge a gap between touch and touchless technologies using low-cost equipment. It was shown to be affordable, intuitive and visual on stage; also fun and easy to for with various audiences, like children and non-musicians. In the future, we want to keep exploring form factors that combine touch surfaces and 3D cameras. We also aim at further studying the stylistics of touch gestures, i.e. the variants in how the surface can be touched.

7. ACKNOWLEDGEMENTS

We thank the numediart Creactifs program through which this project could be completed, and more particularly A. Van Laere and D. de Munck. We also thank the UBC Laptop Orchestra (Go Global fund) for their insights in the design. J. Tilmanne is supported by the European Union, 7th Framework Programme (FP7-ICT-2011-9), under grant agreement number 600676 (i-Treasures project).

8. REFERENCES

[1] N. d'Alessandro, O. Babacan, B. Bozkurt, T. Dubuisson, A. Holzapfel, L. Kessous, A. Moinet, and M. Vlieghe. Expressive Voice Analysis for Realtime and Accurate Synthesis of Singing. *Journal on Multimodal User Interfaces*, 2(2), 2008.

[2] N. d´Alessandro and T. Dutoit. HandSketch Bi-Manual Controller: Investigation on Expressive Control Issues of an Augmented Tablet. In *Proc. of International Conference on New Interfaces for Musical Expression*, pages 78–81, 2007.

[3] N. d'Alessandro, A. Moinet, T. Dubuisson, and T. Dutoit. Causal/Anticausal Decomposition for Mixed-Phase Description of Brass and Bowed String Sounds. In *Proc. of the International Computer Music Conference*, volume 2, pages 465–468, 2007.

[4] S. Izadi, S. Hodges, S. Taylor, D. Rosenfeld, N. Villar, A. Butler, and J. Westhues. Going Beyond the Display: A Surface Technology with an Electronically Switchable Diffuser. In *Proc. of the 21st Annual ACM Symposium on User Interface Software and Technology*, pages 269–278, 2008.

[5] S. Jordà, G. Geiger, M. Alonso, and M. Kaltenbrunner. The reacTable: Exploring the Synergy Between Live Music Performance and Tabletop Tangible Interfaces. In *Proc. of the 1st International Conference on Tangible and Embedded Interaction*, pages 139–146, 2007.

[6] P. Lopes, R. Jota, and J. Jorge. Augmenting Touch Interaction Through Acoustic Sensing. In *Proc. of the ACM International Conference on Interactive Tabletops and Surfaces*, pages 53–56, 2011.

[7] A. Luciani. Being There and Being With: The Philosophical and Cognitive Notions of Presence and Embodiment in Virtual Instruments. In *Proc. of the International Computer Music Conference + Sound and Music Computing Joint Conference*, pages 605–612, 2014.

[8] A. McPherson. TouchKeys: Capacitive Multi-Touch Sensing on a Physical Keyboard. In *Proc. of International Conference on New Interfaces for Musical Expression*, 2012.

[9] D. V. Nort, M. M. Wanderley, and P. Depalle. Mapping Control Structures to Sound Synthesis: Functional and Topological Perspectives. *Computer Music Journal*, 38(3):6–22, 2014.

[10] I. Oikonomidis, N. Kyriazis, and A. Argyros. Efficient Model-Based 3D Tracking of Hand Articulations Using Kinect. In *Proc. of the 22nd British Machine Vision Conference*, pages 101.1–101.11, 2011.

[11] C. Palmer, B. Mathias, and M. Anderson. Sensorimotor Mechanisms in Music Performance: Actions That Go Partially Wrong. *Annals of the New York Academy of Sciences*, 1252:185–191, 2012.

[12] D. Schwarz, G. Beller, B. Verbrugghe, and S. Britton. Real-Time Corpus-Based Concatenative Synthesis with CataRT. In *Proc. of the International Conference on Digital Audio Effects*, pages 1–7, 2006.

[13] J. Shotton et al. Real-Time Human Pose Recognition in Parts from Single Depth Images. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1297–1304, 2011.