Transforming Ordinary Surfaces into Multi-touch Controllers

Alain Crevoisier [†, ‡]
[†] University of Applied Sciences Western
Switzerland (HES-SO / HEIG-VD)
Rue Galilée 15, CH-1400 Yverdon,

alain.crevoisier@heig-vd.ch

Greg Kellum [‡]
[‡] Music Conservatory of Geneva
Rue de l'Arquebuse 12
CH-1211 Genève

greg.kellum@cmusge.ch

ABSTRACT

In this paper, we describe a set of hardware and software tools for creating musical controllers with any flat surface or simple object, such as tables, walls, metallic plates, wood boards, etc. The system makes possible to transform such physical objects and surfaces into virtual control interfaces, by using computer vision technologies to track the interaction made by the musician, either with the hands, mallets or sticks. These new musical interfaces, freely reconfigurable, can be used to control standard sound modules or effect processors, by defining zones on their surface and assigning them musical commands, such as the triggering of notes or the modulation of parameters.

Keywords

Computer Vision, Multi-touch Interaction, Musical Interfaces.

1. INTRODUCTION

1.1 Multi-touch Everywhere

There is a strong focus on multi-touch interaction in HCI, especially since the work of Jeff Han at New York University, who showed the way for radically new interaction paradigms using a large size multi-touch display [3]. Though using a smaller screen, the Lemur controller [18] has also shown that multi-touch has a great potential for innovative music applications, in particular to create virtual control interfaces that are fully reconfigurable by users. Multi-touch screens have even reached the point of mass production with the release of Apple's iPhone and iPod Touch. However, most of the available technologies and approaches only work in specific conditions and are not suitable for ordinary surfaces. For instance, some sensing systems are embedded into the surface [2, 11, 20], while others are specific to screens, either as an overlay above the screen (Lemur, iPhone, iPod Touch), or as a vision system placed behind a rear projected diffusion screen [3, 4, 19, 25].

Solutions exist to track multiple fingers on a generic surface [6, 7, 15], but they are not suitable for detecting individual contact points, that is, if fingers are touching or not the surface. True detection of touch can be achieved roughly using stereoscopy [8, 14], or more precisely with four cameras placed in the corners of the interactive area [9, 23]. It can also be achieved with a single camera by analyzing the shadow of the fingers [13], or by watching fingers intercepting a plane of infrared light projected above the surface [12]. Virtual Keyboards currently on the market [21, 22] are based on this approach, which has the advantage of requiring less computational power than the other ones. However, those devices do not compute true coordinates of touch and their interactive area is limited to keyboard size. We have adapted this method to be compatible with larger surfaces, and combined it with acoustic onset detection in order to get precise timing information. In addition to fingers, our system can detect oblong objects striking the surface, like sticks and mallets, and it also measures the intensity of taps or impacts, allowing to perform the interface both with percussive and touch gestures.

1.2 Screens vs. Ordinary Surfaces

When considering a large size, reconfigurable music controller, one may easily think to something like the Lemur, with a rear projected screen made multi-touch sensitive thanks to one of the approaches mentioned before. However, setups of this kind are better suited for a fixed installation and are not very practical for transporting to different venues. The same can be said in general for large size displays. Even in case of using simple front projection, placing the projector on a shelf or on the ceiling is usually not straightforward and once the installation is complete, the system and the projection surface cannot be moved easily [13]. On the contrary, flat surfaces, suitable to be used as an interface, like tables and walls, are available everywhere. No need to transport the interface if we can simply carry a compact system that will allow transforming an ordinary surface into an input device.

Another motivation for finding a suitable technology enabling to use ordinary surfaces instead of screens or other dedicated surfaces is the possibility to invent new musical instruments that are not only a control device, but also a sound source. In this case, the idea is to use the surface of a vibrating object, such as a metallic plate or a drum head, both to generate a sound and to control it via real-time sound processing [1].

Therefore, our system is designed from the bottom up to be suitable for various use-case scenarios, either strictly as a virtual control surface, or rather as an augmented percussion instrument, in combination with all sorts of flat vibrating objects and surfaces. In any case, the question of the visual feedback will be of particular concern, since users don't interact with an image, as with other screen based controllers. Solutions have been investigated and will be presented in section 3.

NIME08, June 4-8, 2008, Genova, Italy

2. SYSTEM OVERVIEW

2.1 Setup

Figure 1 gives an overview of the setup. The system is comprised of an infrared camera placed above the upper edge of the surface, and two custom designed illuminators placed in the corners of the surface one wants to make touch sensitive. The illuminators are then connected to a personal computer, where touch positions are mapped to MIDI and OSC control events, using a dedicated software editor. The chosen camera is an OptiTrack Slim:V100 [26], which features embedded blob tracking at 100 fps, allowing for much faster performances and reduced CPU usage than using a normal camera.

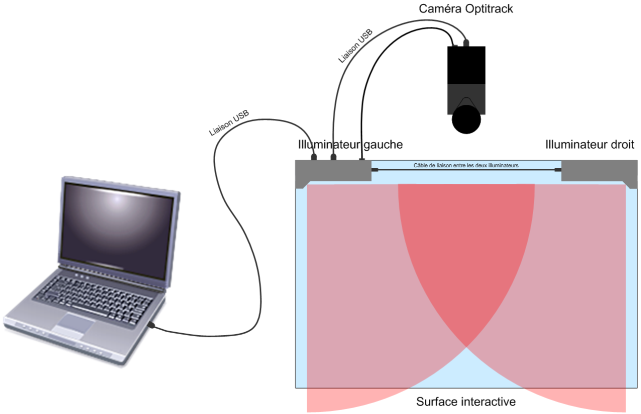


Figure 1. Setup overview, with camera and integrated illuminators.

2.2 Multi-touch Detection

The illuminators are generating a plane of infrared light about 1 cm above the surface. When fingers or other objects are intersecting the plane, reflected light is detected by the infrared camera as brighter spots in the image (Figure 2). Simple blob tracking is performed in the camera using a high-pass filter, and then the positions are sent via USB to the computer, where a calibration procedure is converting them to the physical space using interpolation techniques.

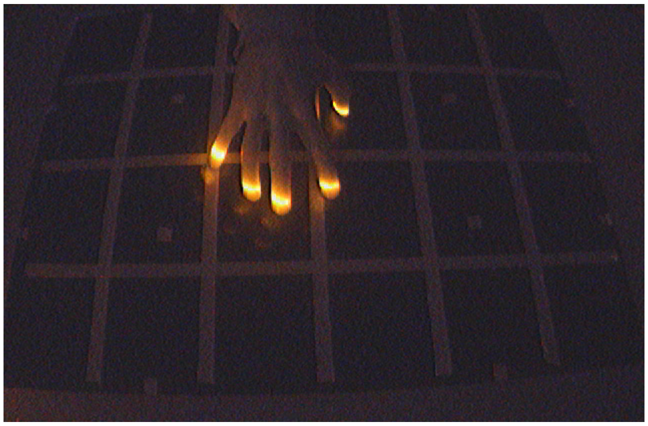


Figure 2. Image seen by the camera. Visible light is filtered out using a 800nm pass filter.

2.3 Integrated Illuminators

Beside their primary function of generating a thin plane of infrared light parallel to the surface, illuminators also integrate acoustic sensors, in order to determine more precisely the timing of impacts on the surface, as well as their intensity and frequency content. They also include several other functions necessary for the proper functioning of the system. A list of all the various functions performed by the integrated illuminators is given below:

- Generation of the light plane, using an infrared laser, mirror and line generator (Figure 3).
- Control of the power of the laser to adjust to ambient light condition (less power is required in low light condition).
- Signal conditioning and amplification for piezo-acoustic sensors.
- Characterization of impacts (onset detection, intensity, and distinction between hard and soft impacts).
- USB hub for connecting the camera.
- Synchronization of the laser with the shutter of the camera.
- Management of security, notably with the use of micro-switches on the bottom of the case (the laser is disabled if the illuminator is not fixed firmly on the surface).

Figure 3 is giving a schematic view of the main illuminator (left one). There are two USB connectors, one to connect to the camera and one to connect to computer. The synchronization signal is provided through a separate cable. A DIN connector allows for connecting the second illuminator, which is more simple (electronic control and signal processing is performed only on the main illuminator).

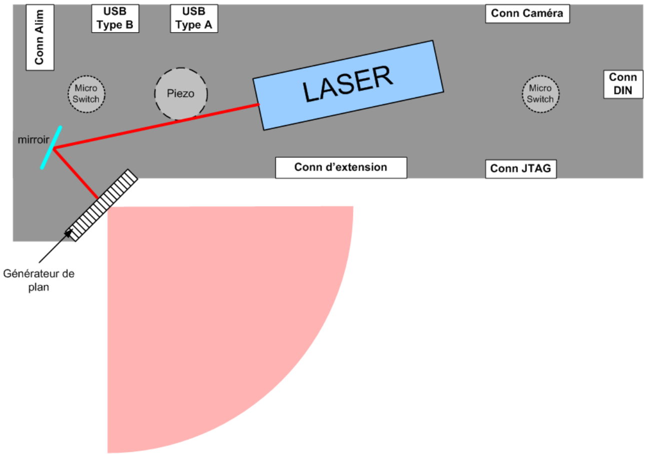


Figure 3. Main illuminator.

2.4 Communication

Contact points and their intensity information are sent to the client application using the OSC protocol. This way, our multi-touch system can be used as input device by a multitude of OSC compatible applications (Reaktor, Max/MSP, SuperCollider, and so on.). The messages are formatted as follows:

/touchEvent id touchState xPos yPos amplitude frequency

The TUIO protocol, developed for communication with table-top tangible user interfaces [5], is also supported. Messages are sent in this protocol using the message type for 2-D cursors, 2Dcur. These TUIO messages do not, however, contain all of the information that is being sent with the previously mentioned OSC message format. TUIO supports sending the identifiers for touch points and their x and y positions explicitly as well as their touch state implicitly. It does not explicitly provide support for sending amplitude or frequency information, but it does provide a single free parameter that can be used to send an int, float, string or blob. We are using this free parameter to send the amplitude of the touch events while discarding their frequencies.

Both our custom OSC messages as well as TUIO can be received by a variety of software clients. We began by working with Max as our preferred client, but we found that mapping a surface in Max to assign functions to various zones on the interface was quite cumbersome. Even though we were using a scripting language inside Max to perform the mapping from contact points to zones, it still took an inordinate amount of time to create the mapping script, Therefore, we have designed a dedicated application for mapping input gestures to MIDI or OSC events, as described in section 4.

3. IN USE

Since no image is projected on the surface, users need to know what they are doing and what the state of their actions is on a different manner. We have explored three different interaction strategies, as presented below.

3.1 Auxiliary Screen

In this configuration, there is no visual feedback at all on the surface and users are watching the computer or laptop screen placed nearby, where fingers positions are represented as colored dots. Control widgets, such as faders or buttons, are then selected by tapping the corresponding finger on the surface (in fact, a quick sequence of Touch Up and Touch Down events with the same contact position). In this configuration, the surface behaves like a giant touch pad with the traditional two steps procedure, positioning of the cursor on the appropriate spot, and selection. The advantage is the simplicity of the approach, but on the other hand, the two steps required to select a widget makes it less appropriate for triggering notes and samples, for instance.

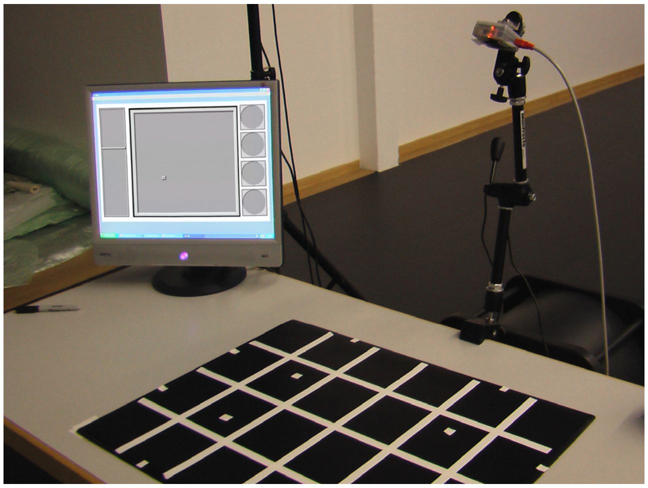


Figure 4. Reference grid on the surface and auxiliary screen.

3.2 Auxiliary Screen & Reference Grid

In this configuration, a visual reference is placed on the surface, in the form of a grid, representing the control area (Figure 4). Control widgets displayed on the screen are aligned according to the same repartition of lines and columns. Figure 5 shows an example of three different mapping layouts that have been designed using Max/MSP. Users can switch from one page to another during performance using the two buttons on the bottom right of each page. The first page features a 4x4 array of pads with a single fader, the second page a 2D continuous controller with the same single fader, and the third page an array of 5 faders. In practice, experiments have shown that the grid on the surface was giving sufficient information to establish a clear correlation between the screen and the surface, allowing to select and activate the desired control widgets in a single step. The advantage is thus a more direct and engaging interaction, compared to the previous approach.

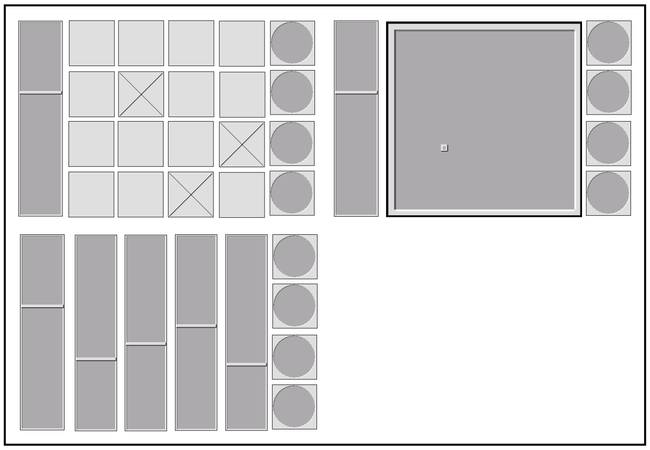


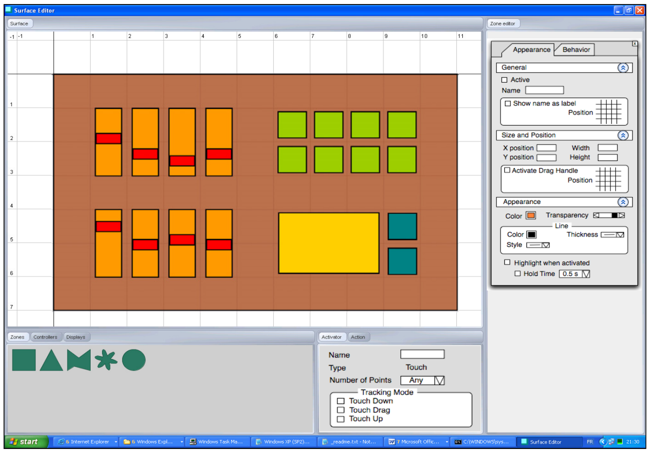
Figure 5. Three different mapping layouts aligned to the same 6x4 grid.

3.3 Reference Grid Only

The last interaction strategy tested is by using only the reference grid. This is certainly the preferred approach for augmented percussions, where the goal is not to control multiple faders or a fine web of control widgets, but rather to trigger samples or use a simple 2D controller mapped to the entire surface. In this case, the absence of screen seemed not to be an inconvenient. On the contrary, it allowed one to be more concentrated on playing the instrument. The only problem found was in being sure of the active page in case of multiple, switchable layouts. In this case, it has been suggested to use a foot controller to change pages (many foot controllers have a LED to indicate the active switch). Another suggestion would be to include a bunch of LED's in a future version of the integrated illuminator, which users could assign freely to page changes or other actions. Lastly, if the surface is horizontal, it is also possible to leave small objects on the surface to get a visual feedback of the value of a continuous controller, or of the state of a switch.

4. SURFACE EDITOR

In order to create control layouts and configure surfaces more easily than using Max, we are currently developing a dedicated software tool. The Surface Editor is organized around a main window, representing the interface, and several configuration and browsing windows that can be either floating or docked on the border of the main screen (Figure 6). The editor has two modes: the editing mode, where all configuration windows are visible, and the full screen mode, where only the interface is visible. Information on the latest version is available on our website [17].



Type Touch

Number of Points Any

☐ Hold Time 0.5 s

Figure 6. The main screen of the Surface Editor.

5. ACKNOWLEDGMENTS

The project presented here is supported by the Swiss National Funding Agency and the University of Applied Sciences. Special thanks to all the people involved in the developments presented here, in particular Pierrick Zoss for the programming of the editor's interface, Aymen Yermani for the initial development of the multi-touch technology, and Mathieu Kaelin for his work on the integrated illuminator.

6. REFERENCES

1. Crevoisier, A. Future-instruments.net: Towards the Creation of Hybrid Electronic-Acoustic Musical Instruments, *Proc. of the CHI workshop on Sonic Interaction Design*, 2008.
2. Dietz, P.H.; Leigh, D.L. DiamondTouch: A Multi-User Touch Technology, *Proc. of the ACM Symposium on User Interface Software and Technology (UIST)*, 2001.
3. Han, J.Y. Low-Cost Multi-Touch Sensing through Frustrated Total Internal Reflection, *Proc. of the ACM Symposium on User Interface Software and Technology (UIST)*, 2005.
4. Jordà, S., Kaltenbrunner, M., Geiger, G. and Bencina, R., The reacTable*, *Proceedings of the International Computer Music Conference* (ICMC2005), Barcelona (Spain).
5. Kaltenbrunner, M., Bovermann, T., Bencina, R. and Costanza, E., "TUIO - A Protocol for Table Based Tangible User Interfaces", *Proceedings of the 6th International Workshop on Gesture in Human-Computer Interaction and Simulation* (GW 2005), Vannes (France).
6. Koike, H., Sato, Y., and Kobayashi, Y. Integrating Paper and Digital Information on EnhancedDesk: a Method for Realtime Finger Tracking on an Augmented Desk System. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 8 (4). 307-322.
7. Letessier, J., and Berard, F. Visual Tracking of Bare Fingers for Interactive Surfaces. *Proc. of the ACM Symposium on User Interface Software and Technology (UIST)*, 2004.
8. Malik, S., and Laszlo, J. Visual Touchpad: A Two-Handed Gestural Input Device. *Proceedings of the International Conference on Multimodal Interfaces*, 2004, 289-296.
9. Martin, D.A., Morrison, G., Sanoy, C., and McCharles, R. Simultaneous Multiple-Input Touch Display, *Proc. of the UbiComp 2002 Workshop*.
10. Polotti, P., Sampietro, M., Sarti A., Crevoisier, A. Acoustic Localization of Tactile Interactions for the Development of Novel Tangible Interfaces, *Proc. of the 8th Int. Conference on Digital Audio Effects (DAFX-05)*, Madrid, Spain, 2005.
11. Rekimoto, J. SmartSkin: An Infrastructure for Freehand Manipulation on Interactive Surfaces. *Proceedings of CHI 2002*. 113-120.
12. Tomasi, C., Rafii, A. and Torunoglu, I. Full-size Projection Keyboard for Handheld Devices. *Communications of the ACM, 46-7* (2003). 70-75.
13. Wilson, A. PlayAnywhere: A Compact Tabletop Computer Vision System, *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST)*, 2005.
14. Wilson, A. TouchLight: An Imaging Touch Screen and Display for Gesture-Based Interaction, *Proceedings of the International Conference on Multimodal Interfaces*, 2004.
15. Wu, M., and R. Balakrishnan, Multi-finger and Whole Hand Gestural Interaction Techniques for Multi-User Tabletop Displays. *Proc. of the ACM Symposium on User Interface Software and Technology*, 2003.
16. http://www.nime.org
17. http://www.future-instruments.net
18. http://www.jazzmutant.com
19. http://www.surface.com
20. http://www.tactex.com
21. http://www.celluon.com
22. http://www.lumio.com
23. http://www.smarttech.com
24. http://www.merl.com/projects/DiamondTouch/
25. http://nuigroup.com/wiki/Diffused_Illumination_Plans/
26. http://www.naturalpoint.com/