

# A tangible surface for digital sculpting in virtual environments

Edouard Callens<sup>1</sup>, Fabien Danieau<sup>2</sup>, Antoine Costes<sup>2,3</sup>, and Philippe Guillotel<sup>2</sup>

<sup>1</sup> INSA Lyon, France

<sup>2</sup> Technicolor, France

<sup>3</sup> IRISA/Inria, France

**Abstract.** With the growth of virtual reality setups, digital sculpting tools become more and more immersive. It is now possible to create a piece of art within a virtual environment, directly with the controllers. However, these devices do not allow to touch the virtual material as a sculptor would do. To tackle this issue we investigate in this paper the use of a tangible surface that could be used in virtual reality setups. We designed a low-cost prototype composed of two layers of sensors in order to measure a wide range of pressure. We also propose two mapping techniques to map our device to a virtual 3D mesh to be sculpted. Participants of an informal test were asked to reproduce a pattern on three meshes: a plane, a sphere and a teapot. They succeeded in this task, showing the potential of our approach.

**Keywords:** tangible interface, haptic sculpting, virtual environment

## 1 Introduction

With the recent technological developments in Virtual Reality (VR), users can now naturally interact with 3D content: they are fully immersed in virtual environments where they can walk around and manipulate 3D objects. Relying on this technology, first tools are already available to enable content creation. Using controllers, a user can paint or sculpt virtual objects. Although this approach is more intuitive than using a keyboard and a mouse, it is still far away from the direct hand manipulation and it lacks touch feedback [9].

In line with this observation, research has been conducted for decades to improve the creation of 3D models of objects and characters. The manipulation of complex data like those is addressed by the research field of Tangible Interfaces [11], enabling users to naturally manipulate physical objects mapped to virtual ones. Numerous materials have been experimented to manipulate 3D models through physical devices, from curved rubber tape [2] to fabric [7]. Nevertheless there are still challenges to tackle to design a device allowing to sculpt any 3D model in a large scale virtual environment. First the hardware should be light enough to be held and displaced around a 3D model. Second the mapping of the device onto a 3D model must be explicit enough and fit any shape.

In this context, we propose a new device that users can touch and press in order to sculpt a 3D mesh with their bare fingers. The device is a surface composed of foam and two layers of pressure sensitive materials: a custom fabric-based pressure matrix relying on Velostat and a matrix of FSRs (Force Sensitive Resistor). The top layer handles light pressure and localization, while the bottom layer handles medium and strong pressure. This device is connected to a 3D modeling environment, where it is represented by a proxy. Two mapping methods between this proxy and the 3D mesh to be modified were designed to fit the two shapes.

## 2 Related Work

This research work focuses on tangible interfaces for virtual sculpting, where an input device acts represents a 3D object. The concept of virtual clay has been extensively studied by Ishii et al. with their Illuminating Clay and Sand-Scape [5]. The users alter the topography of a clay or sand landscape model with their hands. Meanwhile the changing geometry is captured in real-time by a ceiling-mounted laser scanner or IR (infra-red) light sensing technology. This technology is well adapted for the manipulation of landscape-like shapes but hardly adaptable to a random 3D object. Besides, it is not possible to manipulate an existing object. In a way similar, Tabrizian et al. use a malleable surface to enable the creation of virtual landscape [13]. In addition to the modeling, the user can select a tool to add grass, water or routes. The final model can be directly viewed within a head mounted display. More flexible systems based on a deformable membrane allowing to change the shape of a 3D object were proposed [14] [3]. The mapping of the physical to the virtual surface is direct: bumps are created according to the pressure distribution on the surface.

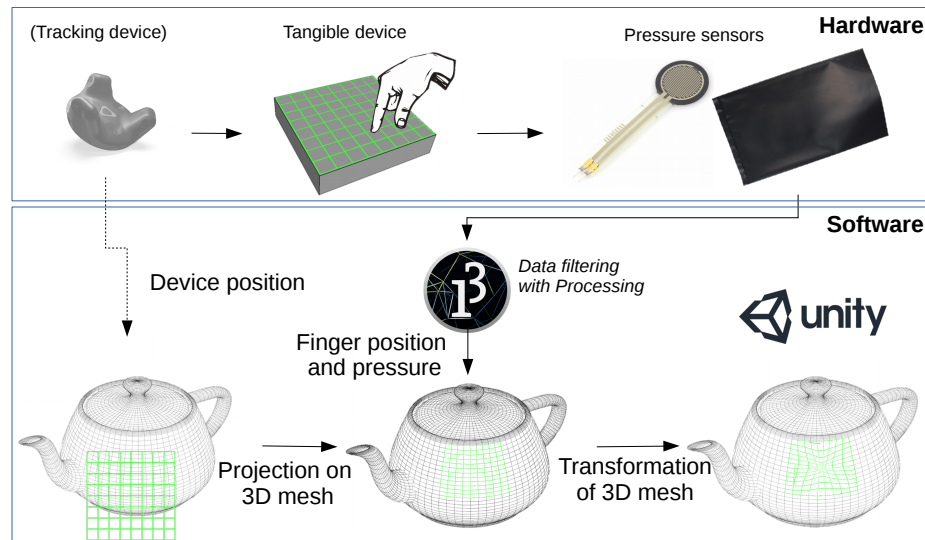
Other materials have been embedded in tangible interfaces. Hook et al. designed a reconfigurable tactile surface based on ferromagnetic technology [4]. This approach is still a proof-of-concept but the authors showed that it might be applicable for virtual sculpting. Foam is also a suitable material for a sculpting device [8]. The user removes part of an actual foam which is tracked and digitally represented. While intuitive, this system does not allow to undo an operation or to work on existing 3D models.

In their review on tangible user interfaces, Shaer and Hornecker pointed out the issue of mapping a physical device to digital information [11]. The direct mapping where the device fully represents the digital information is the more intuitive for the user. But when they cannot be mapped (in the case of abstract digital information for instance), indirect mapping or metaphor has to be used. In the context of virtual sculpting, indirect mapping is necessary when the physical object cannot perfectly match any 3D model. The work of Sheng et al. evokes the idea of a proxy representing the physical device in the virtual world [12]. It helps users to understand how they can interact within this world and how this proxy can be mapped to a virtual object. The authors proposed a global

non-linear mapping and three local relative mappings. Although they detailed the advantages and limitations of each mapping they did not evaluated them.

### 3 Tangible Surface for 3D Sculpting

The overview of our system is depicted in Figure 1. First, a multi-layered and multi-cells device has been designed, based on pressure sensors. The pressure measurements, forwarded to a computer, are pre-processed and filtered by a Processing code providing a pressure and position estimation to the virtual environment (VE) running under Unity3D<sup>4</sup>. In this VE, a proxy represents the physical device. The user can adjust the mapping of the proxy to a target mesh, and press the physical device to deform this mesh.



**Fig. 1.** Workflow overview. The tangible interface is made of foam and pressure sensors (FSRs and Velostat). A proxy maps the physical device to the virtual object surface.

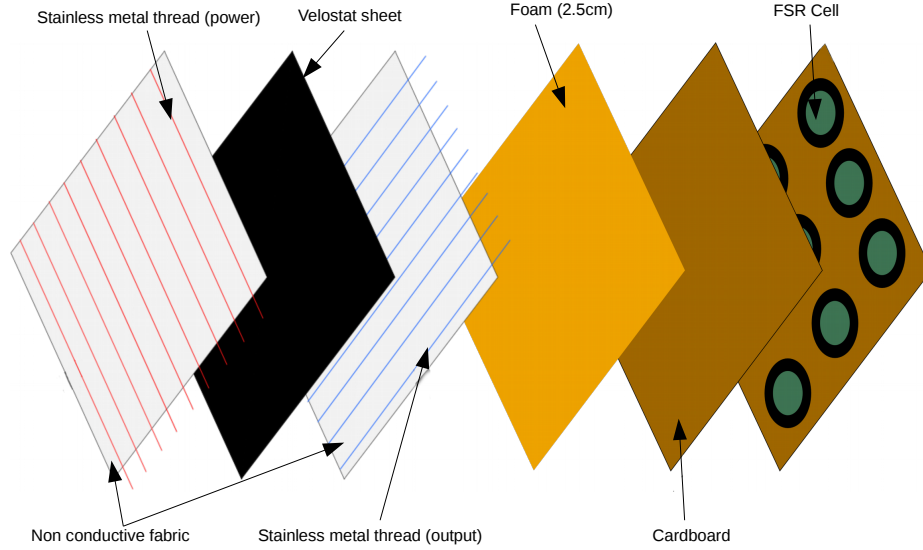
This prototype was designed in such a way that it could be attached to an existing VR controller. Thus, it relies on low cost technology that is light, robust and easily powered. In this work we were interested in the tangible surface itself and therefore we did not consider the tracking part.

#### 3.1 Hardware

Our approach is inspired from the multi-cells architecture [10][15], where several pressure sensors are arranged in a 2D array in order to estimate the pressure lo-

<sup>4</sup> <https://unity3d.com>

cation. To provide a wide range of pressure values, a 2-layers matrix architecture has been used (see Figure 2). The top layer handles light pressures and touch localization, while the bottom layer handles middle and intense pressures with lower localization accuracy.



**Fig. 2.** The multi-layers architecture. The top layer (Velostat) deals with light pressure and touch localization, while the second layer (FSR) handles stronger pressures.

Both layers have been designed as a matrix configuration. To get a high accuracy for the position value, a line per line scanning method has been preferred to independent wiring, arranged in a two-dimensional grid. Each cell can thus be addressed by sharing a common electrode per column and sharing the other one per line with a single voltage divider. Connecting alternatively each column to  $V_{in}$  and measuring the voltage  $V$  on each line, allows to scan the whole matrix.

**Top layer: light pressure sensing and localization** The upper layer is a custom pressure sensor made of Velostat fabric, in-between two grids of stainless steel perpendicular lines. The Velostat sheet is a piezo-resistive material made of carbonized polymer that have its resistance changing depending on the applied pressure. Layered between the two electrodes grids, it acts like a variable resistor.

The resistance value is captured by a micro-controller using a simple voltage divider. The Velostat, variable resistor  $R_{var}$ , is connected between an input voltage  $V_{in}$  and a resistor  $R$  connected to the ground. The voltage  $V$  between the  $R$  poles is measured thanks to an analog digital converter (ADC) integrated into the micro-controller. The ADC gives a digital value proportional to the measured voltage in the range  $[0- V_{in}]$ .  $V$  follows the voltage divider equation



defined as:

$$V = V_{in} \frac{R}{R + R_{var}} \quad (1)$$

As described, the Velostat is stacked between two grids of conductors (see Figure 2). The columns of steel thread on the top grid power each cell, while the lines of conductor on the bottom grid provide the voltage value. The conductive wire has been sewed on a regular cotton fabric in narrow lines. This leads to a resolution suitable for position tracking, as well as it provides a nicer tactile sensation than other materials like foam. The sewing was made to get the conductive Velostat in contact with the thread, while being isolated from the user's fingers. Additional detachable clips were also sewed with the conductive thread on the fabric for an easy connection with the other parts of the system, as shown in Figure 3.

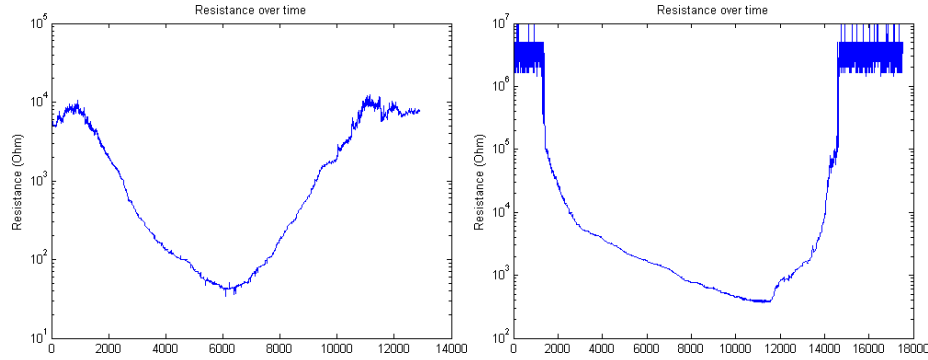


**Fig. 3.** From left to right: overview of the Velostat layer, close up of Velostat sheet and fabric underneath, close up of the detachable clips.

Because of the number of pins (9 columns  $\times$  9 lines=18 pins), a multiplexer is used to simplify the sensing process. It is based on Adafruit's ADS1015 breakouts, more specifically, a 12-bit 4 channels ADC multiplexer communicating in I2C to the micro-controller, including its own ADC converter. Up to four boards with different I2C addresses can be used together, providing 16 analog channels, which is enough for the size of the matrix. On the top grid, a regular 16 channels multiplexer is used to power each line independently, while the others are set to high impedance.

Figure 4 shows a measurement of the Velostat resistance over time. A pressure was linearly (as much as possible) applied with one finger.

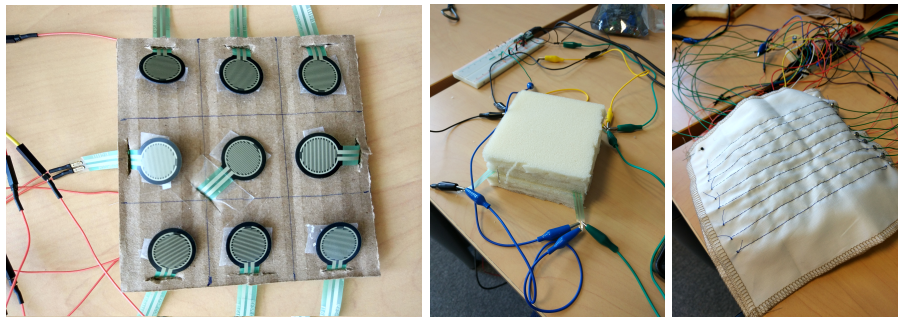
**Bottom layer: intense pressure sensing** The bottom layer embeds a 3x3 *Interlink 402 FSRs* matrix of round pressure sensors (see Figure 5). The 3x3 matrix between a layer of foam and a rigid material surface allows to get a uniform distribution of the pressure. The foam also provides a passive haptic feedback. The same idea, as for the top layer of line scanning, is used to scan the



**Fig. 4.** Resistance over time for the Velostat (left) and for one FSR unit (right).

3x3 matrix using the microcontroller pins to power the columns and reading the 10-bits ADC values. Figure 4 shows the resistance of one FSR sensor over time. The resistance range is two orders of magnitude higher than the Velostat one, but it reaches faster a minimum resistance. Hence, this layer is more suitable for high pressure sensing.

**Micro-controller** The micro-controller board used is a Teensy 3.1, based on the Cortex-M4 chip. The Arduino IDE code is supported through the Tensyduino add-on. This leads to a full device capable of sensing a wide range of pressure on a 2D array (Figure 5). By optimizing ADCs conversion time and I2C communications, the full scan of all the matrices cells takes 90ms, leading to a 11Hz refreshing rate which is enough for finger interactions [6].



**Fig. 5.** Left: the 3x3 FSR matrix on a rigid surface. Middle: the bottom layer and foam. Right: the final device on which the user interacts.

### 3.2 Software

**Driver and Filtering** With I2C boards and matrices cells addresses set up, a dedicated program scans the lines of the two matrices, reads and arranges the data. The ADS1015 ADC output data are converted to 10-bit resolution values providing a 3.3V range. Then the values are packed into OSC messages, transmitted to the PC over USB using the SLIP<sup>5</sup> encapsulated serial communication protocol.

After the SLIP encapsulated serial message is decoded, the raw pressure values are directly filtered in Processing with a 1€ filter [1]. It is an adaptive low-pass filter specifically designed for interactive systems. Easily configurable, two parameters control static jitter and lag. It provides satisfactory results with minor tweaks and is reliable with fast and slow movements on the device (this filter adapts its cut-off frequency depending on the speed of the input). The next step maps the data value from a 10-bit integer to a floating-point number in the range [0-1].

The finger position is computed using the top layer data (with the Velostat). The barycenter of the cells closed to the touch point (selected as the highest pressure value on the matrix) is computed as the final finger position. Finally, the global pressure is computed as an average between the top layer pressure (Velostat) and the bottom layer pressure (FSR matrix). Only cells above a certain threshold are taken into account.

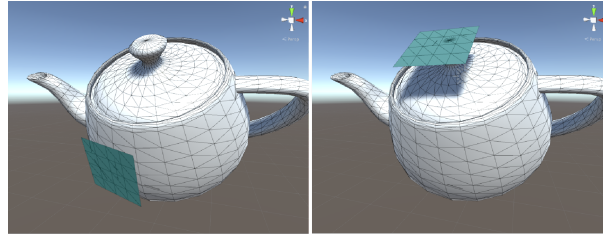
**Virtual environment and Proxy** The virtual environment has been prototyped with Unity3D. The device is manipulated through a proxy that takes the form of a semi-transparent square plane (Figure 6). The key challenge here is to map this planar proxy to any target mesh to be sculpted (parametrization problem). Besides the mapping has to be explicit enough for the users to know what they are manipulating.

The mapping consists in two steps. First, there is an alignment between the proxy and the expected virtual object location. The current prototype is not yet tracked in space, the user has to manually place the proxy on the target mesh. To do so, the user selects a vertex on the target mesh with the mouse, then the center of the proxy is matched to the vertex position. The proxy is also rotated so that the normal of its central vertex is aligned to the normal of the target vertex as shown in Figure 6.

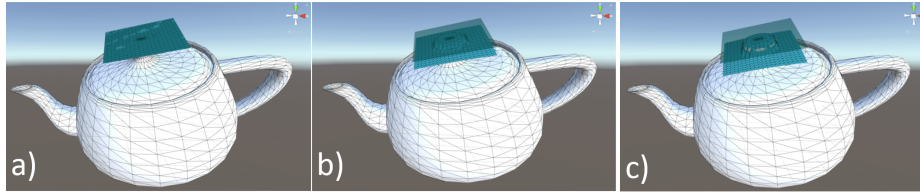
The second step is the geometrical morphing of the proxy to the virtual object mesh. Two solutions have been implemented. The first technique consists in a **projection** of the proxy onto the targeted mesh. Each vertex of the proxy iteratively casts a ray along its normal in direction of the target mesh. If the ray hits a part of the target mesh, the vertex moves along the ray towards the target mesh as seen Figure 7.

The second technique adds an extra **wrapping** step. The goal is to wrap the proxy around the object, for a better accuracy in small parts of a target mesh

<sup>5</sup> Serial Line Interface Protocol, see <https://github.com/CNMAT/OSC>

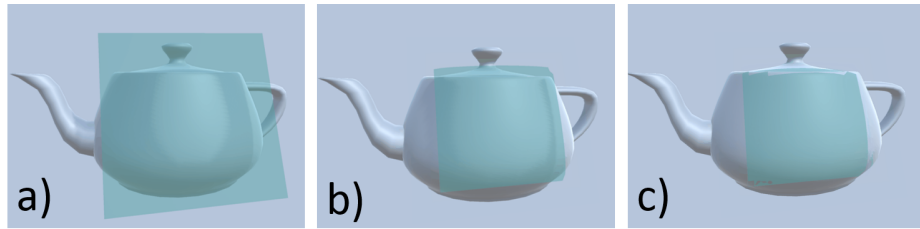


**Fig. 6.** Left: Proxy aligned to a side of a teapot mesh. Right: Proxy aligned to the top of the teapot.



**Fig. 7. Projection** mapping. The proxy is mapped onto the surface of the teapot. Each iteration moves the vertices in the direction of the target mesh normal vectors. a) Initial position, b) Halfway through the process, c) Completely mapped proxy.

(like the teapot handle for instance). Each vertex of the device is scanned from the central vertex to the edges in a circular manner. For each, a ray along its normal is cast, and its normal is aligned to the normal of the hit point. Then, the vertex is moved following previous method, along the normal. All vertices not scanned yet change their normal according to the current one. This way, through the different iterations, the proxy will progressively bend around the targeted mesh as depicted in Figure 8.



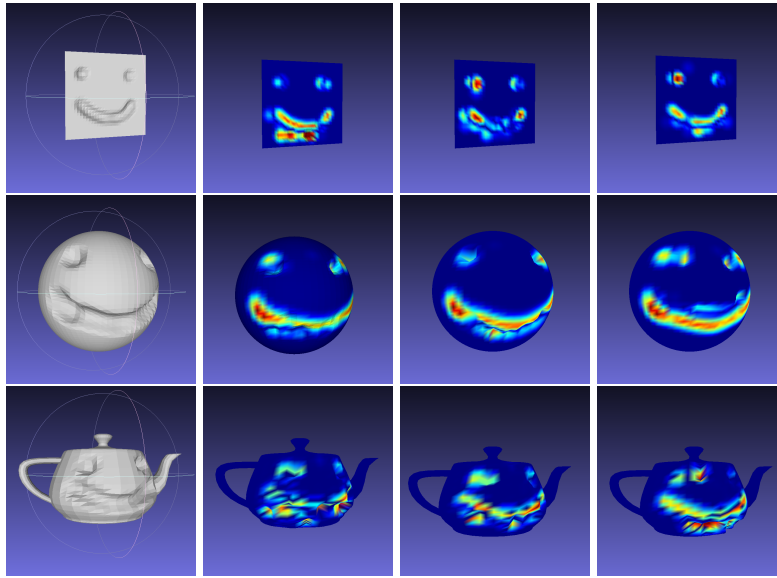
**Fig. 8. Wrapping** mapping. The proxy is mapped onto the surface of the teapot. Each iteration moves the vertices in the direction of the target mesh normal vectors. a) Initial position, b) Halfway through the process, c) Completely mapped proxy.

Once the proxy matched to the target mesh, the user sculpts the virtual object with simple stroke and press gestures on the device, altering the mesh in the area delimited by the mapped proxy. The effect on the mesh is determined by a radius parameter and a deformation speed parameter. Each vertex is moved along the average normal direction computed in the radius around the hit point, with a weighting factor related to the distance to the hit point.

## 4 Results

A pilot test has been conducted to investigate the usability of our prototype, and the influence of the mapping on a sculpting task. In this study, only the top layer with the Velostat was used, as it provides localization and enough pressure information to perform a simple sculpting task. Six participants, including one female, have taken part into the test.

A simple 3D scene has been created in which the user can deform one object (a plane, a sphere or a teapot). The proxy representing the device is also displayed (such as in Figure 6). Three mapping techniques were implemented: the two techniques, "projection" and "wrapping", presented above, and a control technique, "no mapping". This latest technique does not change the shape of the proxy. It remains a plane and a ray is also cast to hit a vertex of the target mesh.



**Fig. 9.** Results from one participant. From left to right: reference mesh, no mapping condition, wrapping technique and projection technique. Colors go from blue (no difference to the reference) to red (maximum difference, normalized with bounding box).

Three deformed objects were created with a professional digital sculpting tool (Sculptris<sup>6</sup>). They served as a reference that the participants had to reproduce (Figure 9 - Left column). The pattern to sculpt was a smiley face on the round or flat part of the objects. This particular shape was chosen to see if the participants could use the whole area of the physical device and trace lines on a curved object.

Once the task completed, we first performed a visual analysis of the meshes deformed by the participants. The Hausdorff distance to the reference was computed with Meshlab and the meshes were colorized accordingly (see Figure 9). All participants succeeded to complete the task: the features of the pattern, eye and smile, are present on the output meshes. Yet, we cannot conclude that one method outperformed another.

We also observed that on the sphere and the teapot, resulting patterns seem to be more centered or compact than on the references. This is probably because the proxy had to be moved or rescaled to perform the task on these objects. By default the proxy was too small to cover the area needed to reproduce accurately the whole pattern. Then, a strategy often adopted by participants was to draw the entire pattern after having moved the proxy.

Finally, we looked at the average time spent by the participants on each object. No significant difference was observed, letting us believe that the mapping techniques seems to not decrease the user's performance.

## 5 Discussion

This first test provided interesting insights that will be taken into consideration to improve our setup. Overall, we observed that the participants succeeded to reproduce the pattern created with a digital sculpting software. At first, the participants directly tried to reproduce the reference shape. Then they erased these attempts, first because of a misplacement of the eyes, then because of a misplacement of the smile. Eventually the participants made something they liked and went to the next object. This observation illustrates well the mapping issue between our flat device and a random object. Interestingly we noticed two strategies to draw the curved line representing the smile. Half of the participants were using a continuous movement pressure on the device, drawing the line in one single gesture. The others were tracing the line by applying multiple touches on the device, forming a series of little holes on the resulted mesh eventually forming a smile.

It seems that working with a mapping technique allows the participant to feel more efficient with the device. Participants reported that the interaction was perceived as more comfortable and better understood. Further studies are however needed to establish the efficiency difference between the two mapping techniques. To differentiate these techniques, a larger set of objects has to be used, including objects with concavity. Obviously more participants are required to properly evaluate the device, but their expertise should be carefully identified.

---

<sup>6</sup> <http://pixologic.com/sculptris>

Professional 3D artists using tablets are familiar with 3D interaction through planar devices.

A visual guide could be displayed both on the physical device and the proxy in order to help the user. Such a guide could be a grid printed on the device that would be deformed on the proxy. Also the top layer, sensible to light touch, could be used to indicate the user’s finger location and the deformation would be only triggered when the bottom layer is activated (stronger pressure).

The use of this two-layer configuration should be also investigated. In this work we made sure that the top layer is enough to perform a sculpting task. The bottom layer would easily extend the pressure amplitude. But other interactions could be designed. Fine details could be edited with the top layer (as a knife would cut clay) while rough details may be handled by the bottom layer (as if a finger presses clay).

In our current implementation the device is a planar surface, but it could easily be shaped differently such as a sphere or a half-sphere with the same sensors and materials. A specific shape could be more adapted to a target model. For instance a sphere would be suitable for face modeling while a plane to landscape modeling. More investigation are to be conducted to determine the limits of the mapping and in which case the shape of the device must be changed.

Finally, the device is currently suitable for interaction with one finger. It can be extended to multiple fingers. For example, blob detection could detect multiple finger inputs on the device. With this feature more interaction techniques could also be supported such as twisting, bending or stretching a 3D mesh.

## 6 Conclusions & Perspectives

In this paper, we introduced a low-cost tactile surface device designed to sculpt 3D meshes. It is composed of two layers of sensors to spatially capture both light touch and strong pressure. We also proposed two methods to map the planar device to an arbitrary 3D mesh. Finally we identified in an informal user test the strengths and weaknesses of the device. In this test participants succeeded in sculpting a pattern on various virtual objects

This work is a first step toward an intuitive mesh manipulation within virtual environments, and highlights many aspects to be studied. Our fabric-based approach allows for a variety of device shapes, which raises several questions: should the device be flat or curved? Could it handles other gestures, like squeeze or pinch? In a next step, the device will be tracked in space. Added to standard VR controllers like the Oculus touch or the Vive controller, it could be usable inside Unity3D in a straightforward way. Such a setup would be relevant for the already existing sculpting or painting tools.

## References

1. Casiez, G., Roussel, N., Vogel, D.: 1€ filter: a simple speed-based low-pass filter for noisy input in interactive systems. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. pp. 2527–2530. ACM (2012)

2. Grossman, T., Balakrishnan, R., Singh, K.: An interface for creating and manipulating curves using a high degree-of-freedom curve input device. In: Proceedings of the SIGCHI conference on Human factors in computing systems. pp. 185–192. ACM (2003)
3. Han, J., Gu, J., Lee, G.: Trampoline: A double-sided elastic touch device for creating reliefs. In: Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology. pp. 383–388. UIST '14, ACM (2014)
4. Hook, J., Taylor, S., Butler, A., Villar, N., Izadi, S.: A reconfigurable ferromagnetic input device. In: Proceedings of the 22nd annual ACM symposium on User interface software and technology. pp. 51–54. ACM (2009)
5. Ishii, H., Ratti, C., Piper, B., Wang, Y., Biderman, A., Ben-Joseph, E.: Bringing clay and sand into digital design: continuous tangible user interfaces. *BT technology journal* 22(4), 287–299 (2004)
6. Jones, L.A.: Kinesthetic sensing. In: in Human and Machine Haptics (2000)
7. Leal, A., Bowman, D., Schaefer, L., Quek, F., Stiles, C.K.: 3d sketching using interactive fabric for tangible and bimanual input. In: Proceedings of Graphics Interface 2011. pp. 49–56. Canadian Human-Computer Communications Society (2011)
8. Marner, M.R., Thomas, B.H.: Augmented foam sculpting for capturing 3d models. In: 3D User Interfaces (3DUI), 2010 IEEE Symposium on. pp. 63–70. IEEE (2010)
9. Massie, T.: A tangible goal for 3d modeling. *IEEE Computer Graphics and Applications* 18(3), 62–65 (1998)
10. Saenz-Cogollo, J.F., Pau, M., Fraboni, B., Bonfiglio, A.: Pressure mapping mat for tele-home care applications. *Sensors* 16(3), 365 (2016)
11. Shaer, O., Hornecker, E.: Tangible user interfaces: past, present, and future directions. *Foundations and Trends in Human-Computer Interaction* 3(1–2), 1–137 (2010)
12. Sheng, J., Balakrishnan, R., Singh, K.: An interface for virtual 3d sculpting via physical proxy. In: GRAPHITE. vol. 6, pp. 213–220 (2006)
13. Tabrizian, P., Petrasova, A., Harmon, B., Petras, V., Mitsova, H., Meentemeyer, R.: Immersive tangible geospatial modeling. In: Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. p. 88. ACM (2016)
14. Watanabe, Y., Cassinelli, A., Komuro, T., Ishikawa, M.: The deformable workspace: A membrane between real and virtual space. In: Horizontal Interactive Human Computer Systems, 2008. TABLETOP 2008. 3rd IEEE International Workshop on. pp. 145–152. IEEE (2008)
15. Zhou, B., Lukowicz, P.: Textile pressure force mapping. In: Smart Textiles, pp. 31–47. Springer (2017)