CALLIGRAPHIC BRUSH

An Intuitive Tangible User Interface for Interactive Algorithmic Design

SHENG KAI TANG
Computational Design Program
Graduate School of Architecture
Carnegie Mellon University
Pittsburgh, PA, 15213
USA
shengkait@gmail.com

AND

WEN YAN TANG
Theoretical Design Group
Graduate Institute of Architecture
Tainan National University of the Arts
Tainan, 72045
TAIWAN
sp297922@giga.net.tw

Abstract. The development of better User Interface (UI) or Tangible User Interface (TUI) for 3D modeling has lasted for decades. With the popularity of free form style achieved by algorithmic methods, the existing solutions of UI/TUI for CAD are gradually insufficient. Neglecting the steep learning curve of algorithmic design requiring solid background of mathematics and programming, the common drawback is the lack of interactivity. All actions rely heavily on mental translations and experimental trial and error. In this research, we try to realize the idea of interactive algorithmic design by developing a tangible calligraphic brush, with this device designer can intuitively adopt algorithmic methodology to achieve highly creative results.

Keywords. Intuitive interface: tangible user interface; algorithmic design.

1. Introduction

The development of better user interface (UI) for 3D modelling has lasted for decades. Most solutions aimed at solving two typical problems of conventional input devices: 1) designers have to translate mental images to complex menu operations; 2) they lack direct tactile feedbacks to manipulate virtual objects.

For the first problem, researchers adopt approach of construction kits which physicalizes modeling primitives connecting to virtual ones on the screen (Aish, 1979; Gorbet and Orth's, 1997; Anderson et al., 2000; Weller et al., 2008). While constructing forms with physical primitives, users receive prompt visual feedbacks from virtual ones. For the second one, enabling "natural" interaction between users and modeling system is commonly seen (Zhai et al., 1999; Zhai, 1998; Murakami, et al., 1994; Rekimoto, 2002; Lertsithichai et al., 2002; Llamas et al., 2003; Lee et al., 2004; Tang and Tang, 2006). This approach focuses on realizing devices being able to continuously sense users' natural performance of hands.

With the popularity of free-form building, those solutions proposed above are gradually insufficient. This is because, on the one hand, primitives such as cube, cylinder and sphere can't fulfill designers' crazy imaginations which totally escape from Cartesian coordinate system. On the other hand, although those sensing devices could still accurately translate users' hand gestures, the complexity of form is out of control of hands. Under this situation, many algorithmic methods are discovered and adopted (Terzidis, 2006; Meredith, 2008).

2. Problem and Objective

Neglecting the steep learning curve of algorithmic design requiring solid background of mathematics and programming, the common drawback is the lack of interactivity. In detail, designers always don't know where to set the seeds, how many loops to run and when to stop. All actions rely heavily on mental translations and experimental trial and error. This suddenly pulls the computational design process back to a decade ago.

In this research, we try to realize the idea of interactive algorithmic design by developing a tangible calligraphic brush which can not only sense user's natural hand gestures when drawing, but further relate different gestures to designate algorithms affecting and generating architectural form on the screen. With this device, designer can intuitively adopt algorithmic methodology to achieve highly creative results.

3. Methodology and Steps

To realize our idea of Calligraphic Brush, we go through a four steps prototyping process which includes hardware design, algorithm design, system design and preliminary test. In hardware design section, we illustrate how to make a digital calligraphic brush consisted of 9 sensible brush hairs. In the algorithm design section, we propose the idea of how to recognize and interpret sensor data into meaningful gestures of drawing and writing. In the system design part, we design a state machine incorporating hardware and algorithm designs to form a complete functional prototype. Finally, we do some preliminary user tests to ensure performance of our prototype and look for possible issues to improve.

3.1 HARDWARE DESIGN

3.1.1 A sensible brush hair

In order to implement the idea of sensible brush hair, we develop our hair sensor based on the idea of IBM Track Point. There are three parts of the sensible hair: brush hair, pointing sensor and base (Figure 1). The brush hair is made of bendable rubber stick attached on the pointing sensor. When the rubber is bended, the force is transferred to the pointing sensor. The pointing sensor operates by sensing applied force by using a pair of resistive strain gauges beneath. The direction and velocity of manipulation depend on the applied force further translated by the TMP754A IC of Phillips mounted on the based.

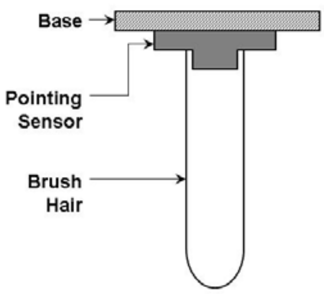


Figure 1. The sensible brush hair.

3.1.2 The arrangement of brush hairs

There are different strokes when writing with conventional calligraphic brush. In detail, depending on different hand gestures resulting in diverse angles and distance from brush hairs toward writing surface, users can create different artistic strokes of calligraphy. In order to realize this characteristic, we arrange 3×3 sensible hairs with different lengths in a square. As we can see in Figure 2, this arrangement can possibly detect the contact sequence of hairs which is further used to interprete the measurement of stroke area.

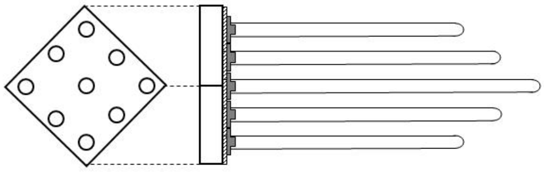


Figure 2. The arrangement of brush hairs.

3.2 ALGORITHM DESIGN

With the hardware developed above, we further design two algorithms to translate 9 sensor signals into meaningful information which are strokes and moving directions of the brush.

3.2.1 Stroke Recognition

There are three types of strokes to recognize which are thin, middle and thick (Figure 3). A stroke can be interpreted as a thin stroke only when no other hairs but the center one is triggered. When one of the hairs marked in dark gray shown below is triggered, the thin stroke will be turned into the middle stroke. Finally, when one of the hairs in light gray is triggered, the thick stroke will be interpreted. Of course, between different strokes, we use the method of interpolation to smooth the transition.

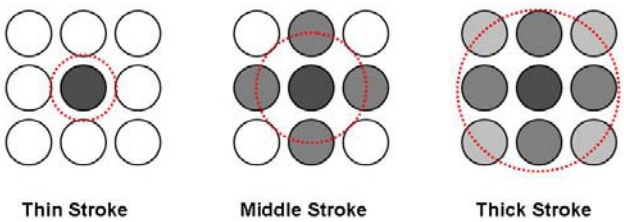


Figure 3. The stroke recognition algorithm.

3.2.2 Direction Recognition

Direction is a very important attribute when writing and drawing. A continuous displacement of direction consists of a segment of line, a path and even a pattern. Hence, how to interpret the discrete sensor signals into an integrated result is the issue to solve.

Our approach is to calculate the integrated vectors of accelerations. As shown in Figure 4, when only one sensor is trigger, we calculate its acceleration by comparing its current vector of velocity and that of previous loop. The direction of acceleration vector is the direction of displacement. If there is more than one sensor triggered, we calculate the integrated acceleration vector after going through the acceleration calculation for each triggered sensor.

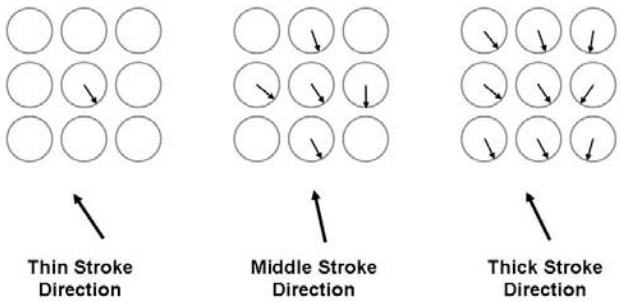


Figure 4. The direction recognition algorithm.

3.3 SYSTEM DESIGN

3.3.1 System Integration

After finishing the previous sections of implementation, we integrate hardware and software components into a working system. There are three parts of this system, Calligraphic Brush, Arduino Matrix and three on-screen applications.

In detail, there are 9 signals, representing velocity of 9 sensors, sent from our Calligraphic Brush to the Arduino Matrix (a homemade microcontroller toolkit based on Arduino). The Arduino Matrix collects these data and binds them into a string. This string is then sent to our on screen applications for further calculation and recognition through serial communication.

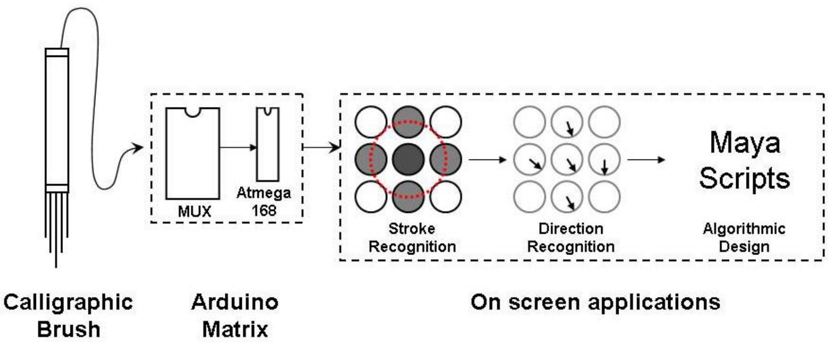
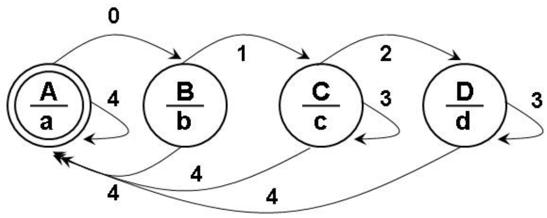


Figure 5. The system design diagram.

3.3.2 State Machine

With the hardware integration idea and exact connections, we implement a state machine for this system. There are 4 states, 4 actions and 5 transition conditions shown in Figure 6. For example, when no signal is detected (T4), the system goes to state A with no actions. Once any signal is detected (T0), the system goes to state B with visualization of signals. The system will further go to state C and D when stroke and direction are recognized.



States	Actions	Transition Conditions
A: not activated	**a:** -	**0:** signal detected
B: activated	**b:** visualize signals	**1:** stroke recognized
C: stroke prediction	**c:** predict stroke	**2:** direction recognized
D: direction prediction	**d:** predict direction	**3:** remain the same
		4: no signal

Figure 6. The state machine diagram.

3.4 PRELIMINARY TEST

With the working Calligraphic Brush which can receive signals from brush hairs and further interpreted it into stroke types and stroke direction, we then read these two information into Maya and connect them with some shape generation script modules. In order to test the performance of our interactive algorithmic design system, we invite three users to play and get some feedbacks. There are two common feedbacks from users: 1) the interface is interesting and does enable user to trigger and stop the Maya scripts intuitively; 2) however, the relationship between drawing strokes and scripts is blur and indeterminate.

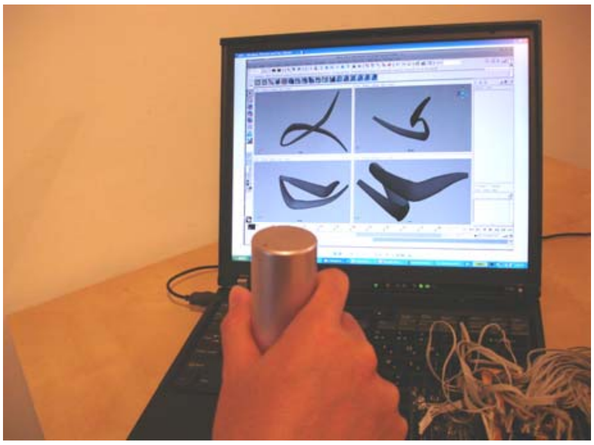


Figure 6. The preliminary user test.

5. Conclusion

This research mostly focuses on the development of the intuitive tangible user interface for enabling interactive algorithmic design. We adopt existing research result of translating from calligraphy to form instead of developing our own one (Yeh, 2006). In detail, those algorithms for form generation are predesigned and sealed. Users only can control gesture, movement, direction and duration to test the interactivity. Enabling users to modify and define their own calligraphic algorithms will be our limitation and future study.

References

Aish, R.: 1979, 3D Input for CAAD Systems. *Computer-Aided Design*, 11(2):66-70.

Anderson, D, Frankel, J., Marks, J., Agarwala, A., Beardsley, P., Hodgins, J., Leigh, D., Ryall, K., Sullivan, E., Ydidia, J.: 2000, Tangible Interaction + Graphical Interpretation: A New Approach to 3D Modeling. *In Proc. of SIGGRAPH 2000*, p.393-402.

Gorbet, M., and Orth, M.: 1997, Triangles: Design of a Physical/Digital Construction Kit, *In Proc. of the Symposium on Designing Interactive Systems 1997*, p.125-128.

Lertsithichai, S. and Seegmiller, M.: 2002, CUBIK: A bi-directional tangible modeling interface, *In Proc. of the Conf. on Human Factors in Computing Systems, CHI 2002*, p. 756-757.

Llamas, Ignacio., Kim, B., Gargus, J., Rossignac, J., Shaw, C.D.: 2003, Twister: a space-warp operator for the two-handed editing of 3D shapes, *ACM Transactions on Graphics (TOG)*, Vol 22 Issue 3, 2003, p.663-668.

Murakami,T. and Nakajima, N.: 1994, Direct and Intuitive Input Device for 3-D Shape Deformation, *Proceedings of the CHI 1994*, p.465-470.

Meredith, M.: 2008, From Control to Design: Parametric/Algorithmic Architecture, *Actar*.

Rekimoto, J.: 2002, SmartSkin: An Infrastructure for Freehand Manipulation on Interactive Surfaces, *Proceedings of the CHI 2002*, ACM Press.

Tang, W.Y. and Tang, S.K.: 2006, A development of tactile modeling interface, *In Proceedings of The Eleventh Conference on Computer Aided Architectural Design Research in Asia 2006*.

Terzidis, K.: 2006, *Algorithmic Architecture*, Architectural Press.

Weller, M. P., Do E. Y. L., and Gross, M. D.: 2008, Posey: Instrumenting a Poseable Hub and Strut Toy, *2nd ACM Conference on Tangible and Embedded Interaction*, Bonn, Germany.

Yeh, Y. N.: 2006: Freedom of Form - *The Oriental Calligraphy and Aesthetics in Digital Fabrication*, A Master Thesis, Architecture, NCTU.

Zhai, S., Kandogan, E., Smith, B., Selker, T.: 1999, In Search of the "Magic Carpet",

Design and Experimentation of a 3D Navigation Interface, *Journal of Visual Languages and Computing*, Vol 10, No.1, 3-17, p.3-17.

Zhai, S.: 1998, User Performance in Relation to 3D Input Device Design, In *Computer Graphics* 32(4), November 1998, p.50-54.