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
shengkai@gmail.com

AND

WEN YEN TANG
Theoretical Design Group
Graduate Institute of Architecture
National Tainan University of the Arts
Tainan, 72045
TAIWAN
sp297922@live.com

Abstract. The development of better User Interface (UI) and 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; Lertsitichai 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’ creative 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 associate diverse 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 step 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 strokes and directions of the calligraphic brush. 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 for future improvement.

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.

3.1.2 The arrangement of brush hairs

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

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

We briefly define three types of strokes to recognize in this project which are thin, middle and thick (Figure 3). A stroke can be interpreted as a thin stroke only when no other hair but the center one is triggered. When one of the hairs marked in lighter gray shown below is triggered, the thin stroke state will be turned into the state of middle stroke. Finally, when one of the hairs in lightest gray is triggered, the thick stroke will be interpreted. Between different strokes, we use the method of interpolation to smooth the transition.
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 discover.

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.](image)

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, the Calligraphic Brush, an 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 finally implement a state machine for operating this system. There are 4 states (A, B, C and D), 4 actions (a, b, c and d) and 5 transition conditions (TC0, TC1, TC2, TC3, and TC4) shown in Figure 6. For example, when no signal is detected (TC4), the system goes to state A with (a) no action. Once whatever signal is detected (TC0), the system goes to state B with action of (b) visualizing signals. The system will further go to state C and D with (c) prediction of stroke and (d) prediction of direction, when stroke and direction are exactly recognized (TC1, TC2).

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 them into stroke types and stroke direction, we
then send these two interpretations 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 7. The preliminary user test.](image)

### 5. Conclusion

This research mostly focuses on the development of the intuitive tangible user interface for enabling interactive algorithmic design. We reference existing research result of translating from 2D calligraphy to 3D shape instead of developing our own method (Yeh, 2006). Furthermore, those algorithms for shape 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