A Flexible Finger-mounted Airbrush Model for Immersive Freehand Painting

Ruimin Lyu, Yuefeng Ze, Wei Chen, Fei Chen, Yuan Liu, Lifang Chen, Haojie Hao
Jiangsu Key Laboratory of Media Design and Software Technology
Jiangnan University
Wuxi, China
magicbrush@qq.com

***Abstract*—To provide immersive freehand painting experience, we proposed a flexible airbrush model making use of the hands tracking capability of Leap Motion Controller. The airbrush model uses a common screen as the painting canvas. When the user moves hands over the screen, the brush model continually acquires his/her hands movement data and extracts multiple control signals which describes multiple gestures. The virtual airbrush moves along with the user's hands movement as if it is fixed on his/her finger, and its properties change with gestures' change. When the virtual airbrush intersects with the screen, it continually exerts paints onto the screen. User test shows that the user can easily create multifarious brush stroke effects by directly operating over the screen.**

Keywords—brush modeling; airbrush; paint system; paints simulation; hands tracking

I. Introduction

In the past, physical tools are inalienable in the practice of painting. Is it possible to enable freehand painting with good immersive experience? To achieve this, four issues should be studied, as shown in Fig. 1.

The first issue is the hands tracking capability that enables one to paint without any physical burden. Traditional painters always keep a large bunch of tools. Digital painting systems relieve artists from the burden of physical paints and toolkit but the tablet and stylus are stills needed. Nowadays, the emerging natural hands interaction technologies provided by Leap Motion controller gives a promising solution. But hands tracking capability itself is far from freehand painting.

The second issue is to make the canvas as either the operational plane or the displaying plane. In a common digital painting system, the separation of tablet and screen leads to unsatisfactory immersive experience. By combining a tablet and a screen, an embedded LCD tablet solves the problem, but this solution is expensive.

The third issue is to make flexible interactive features for a brush tool. This issue relates to both the stylus and the virtual brush model. Comparing to an entry level stylus that only provides pressure sensing capacity, an advanced stylus provides better sensing capacity with higher accuracy, framerate and more dimensions of control signals, e.g., the stylus' tilt and rotation. Also the virtual brush model should be made to be properly controlled by the sensing capacity of the stylus, which means the brush properties (e.g., scale, rotation, opacity) can be bind to the multidimensional control signals. As a result, with an advanced stylus with properly matched virtual brushes, the artist can use flexible interactive techniques to create multifarious stroke effects. This issue is important because it relates to providing the experience of free artistic exploration and expression. Is it possible to make a brush to provide more flexible interactive capacities than a typical professional setup?

The forth issue is the customizability of the brush tool. In traditional media painting, an artist always keeps various types of brush. In a typical digital painting system, the brush tool can be configured as different presets of different stroke effects. In both cases, the goal of the toolkit's setup is to provide enough tools for different painting techniques. But too many tools pose a heavy burden for the user to choose and configure. Is it possible to make a highly customizable brush tool which is versatile to be used for different purpose with only slight modifications?

Aiming at the above issues, we propose a flexible and customizable virtual 3D airbrush based on the hands tracking technologies. With a common PC and a Leap Motion controller that are affordable for most people, the brush enables one to directly paint on the screen by hands movement without any extra tools or wearable devices.

II. Related works

Previous researches about digital brush centered on the modeling techniques of brushes, and thus can be categorized based on the brush types: (1) Hard brushes. Some researchers applied 2D simulations, such as the digital wax crayon proposed by Rudolf et al [1]. While other researchers tend to use 3D physical modelling techniques, such as the virtual painting knife and 3D geometrical brushes [3]. (2) Bristle brushes. Most researches among this type employs 3D physical simulation techniques, e.g., the virtual bristle brushes in Photoshop [4], physical-based virtual Chinese brushes [5]. (3) Special effects brushes. This type of brushes are modeled not to simulate classic painting brushes but to create uncommon stroke effects. Some 2D brush used special texture mapping technique to place

978-1-5090-5507-4/17/$31.00 ©2017 IEEE
ICIS 2017, May 24-26, 2017, Wuhan, China

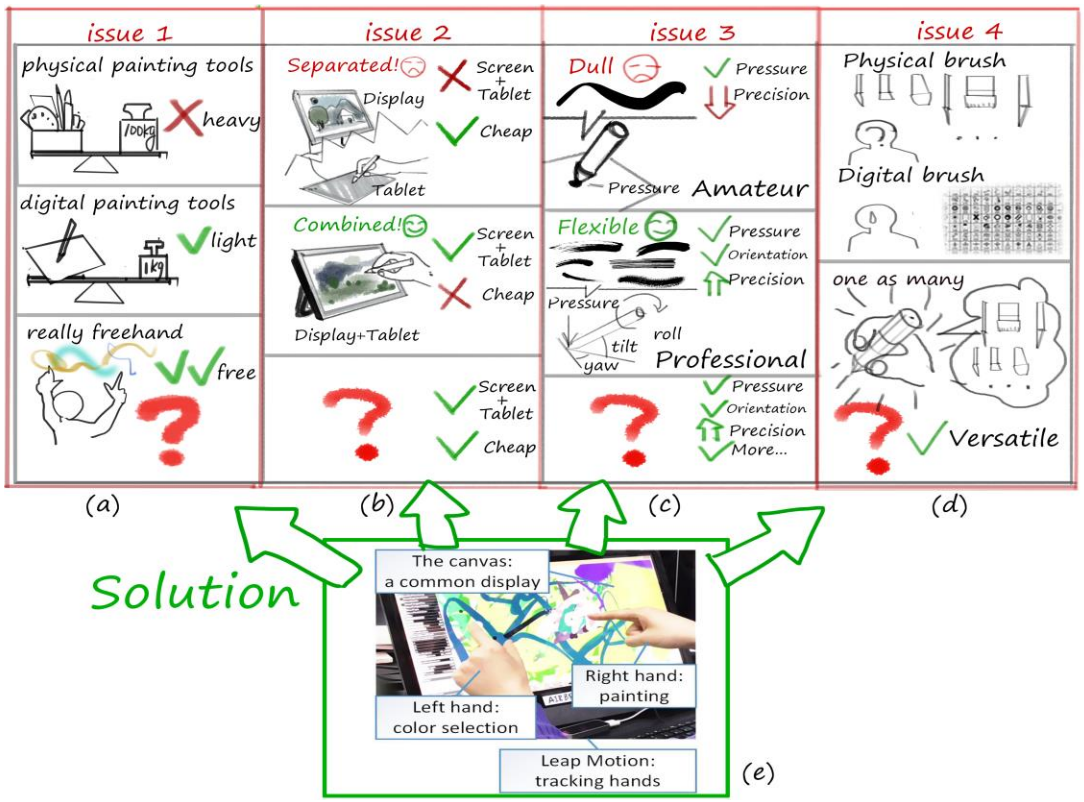

Fig. 1. Motivation and Solution

patterns along stroke paths, e.g., DecoBrush [6]. Some brushes employ uncommon mechanism to create special effects, e.g., Smoke brush [7]. Some other technologies tried to model the mechanism of modern painting tools and media, e.g., the jet painting system digitally recreate Jackson Pollock's drip painting effects [8]. 4) Airbrush modeling and augmentation. Some researcher designed digital airbrush with special input device [9]. Other researches tried to augment physical airbrush with digital computing capabilities [10].

Apart from the above brush modeling techniques, some other researches specially concentrated on improving the interaction experience of digital brushes. Drawing Prism is an early painting system that enables one to directly paint on a monitor[11]. Sun et al added an orientation sensor to a digital stylus to recognize the user's different purposes, which assists the quick switching between different digital brushes [12]. Harada et al proposed the method to control brush properties by non-linguistic vocalization [13]. Peter et al proposed Fluidpaint system based on the simulation of fluid dynamics and physical brushes [14]. However, most of the above researches requires the user to operate or wear some special device and thus are not related to relieving the user from physical constraints.

In recent years, the emerging natural interaction technologies inspired the trend of freehand digital painting. Kazi et al proposed the digital sand-art performance system SandCanvas based on Kinect [15]. Corel Ltd released Painter Freestyle which is the first commercial app that enables one to paint in the air using pure hand gestures [16]. However, this app does not provide the tight combination of the display plane and the painting plane, and thus the immersive experience is still not satisfactory.

In sum, none of the past researches solves all the proposed 4 issues.

III. Brush Model Overview

The conceptual model of the proposed finger-mounted airbrush is shown in Fig. 2. The brush is a continual paints flow constrained in an elliptic cone. It is fixed on the tip of the user's right hand's forefinger, with its direction keeps along the finger, and thus moves in pace with the forefinger's movement. The canvas is a common display. When the cone intersects with the canvas, the paints flow continually transfer to the canvas within the intersection.

Fig. 3 shows our technical framework that realize the above concept model. The model mainly includes two sub-models: the interaction model and the painting model.

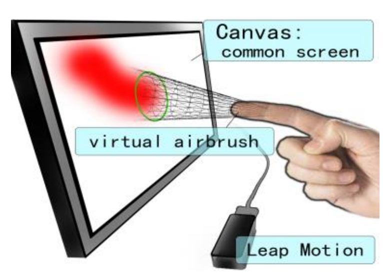

Fig. 2. The conceptual model

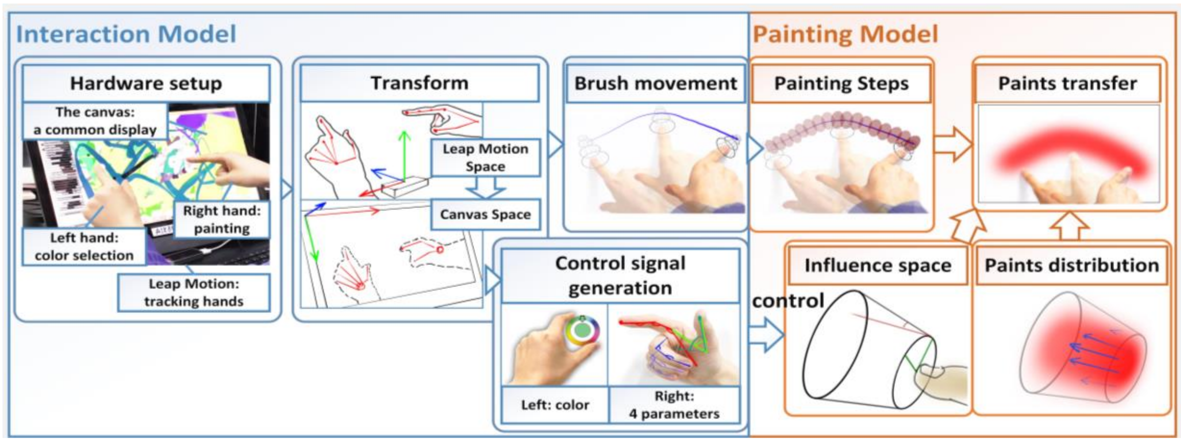

Fig. 3.Brush Model Overview

The interaction model relates to the interaction mechanism between the users' hands and the brush. It includes four parts:

Hardware setup: The method to setup the Leap Motion Controller and the canvas.

Transform: Transform the hands' movement from the Leap Motion coordinates system to the canvas coordinates system. This part is the key to achieve the tight combination of operational plane and displaying plane.

Brush movement: The mechanism of moving the brush in pace with the finger's movement.

Control signal generation: Generating multidimensional control to control brush properties.

The painting model relates to the interaction mechanism between the brush and the canvas. It includes four parts:

Influence space: The geometrical shape of the paints flow.

Paints distribution: The distribution of the paints flow within the influence space.

Painting steps: The mechanism to simulate the continual painting along the brush's moving path.

Paints transfer: How the paints transfer to the canvas.

The above two sub-models are described in section 5 and 6, followed by the brush customization method discussed in section 6.

IV. Interaction Model

A. Hardware Setup

The hardware setup includes a PC, a common screen and a Leap Motion controller. The screen is the canvas, which acts as either the operational plane or displaying plane. The Leap Motion controller can be randomly placed only if it can properly track the user's hands' movement near the screen.

B. Transfrom

To achieve the tight combination of the operational plane and the displaying plane, we need to sense the hands' movement relative to the screen. The key step is transform the hands' position from the Leap Motion space to the canvas space. We can estimate the transform by points matching as below.

The two spaces are shown in Figure 4, and the transform can be represented as matrix multiplication:

$$P_c = TP_L \tag{1}$$

In the above equation, P_L and P_C are respectively the coordinates of a position in the Leap Motion space and the canvas space, T is the transform matrix. Once T is known, any data sensed by the Leap Motion controller can be transform to the position relative to the canvas. The proposed method find out T *by* SVD (Singular Value Decomposition) based on 4 register points which are sampled as 4 pairs of Pc and P_L.

C. Control Signal Generation

To enable flexible brush control using more information other than pressure and orientation, which are usually provided by commercial tablet, the proposed method computes multidimensional control signals, which are denoted as:

$$Ctrl = \{b_1, \varphi, \omega, b_2, h, s, v\} \tag{2}$$

The computation of these signals is based on the hands information tracked by the Leap Motion controller, including the position and orientation of every joint and fingertip, the direction of every finger. Here we omit the detailed description of the sensed data from Leap Motion, which can be found in the official developer's document.

Fig. 4 shows the method to compute these control parameters. All the computation are performed after transform all the sensed data to the Canvas space by eq (1). The meaning and computation of them are:

- Forefinger curvature (b_1): the curvature of the right forefinger;
- Thumb open (φ): the degree that the right thumb stretches away from the palm;
- Thumb roll (ω): the degree of the right thumb rolling around the palm;
- Last 3 fingers curvature (b_2): the general curvature of the last three fingers of the right hand;
- Color (h, s, v): the color of the paints, which are controlled by the left hand and introduced below;

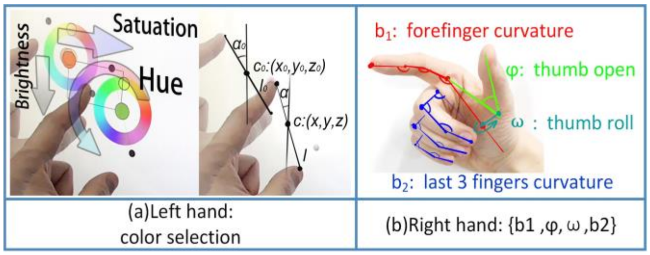

Fig. 4. Multidimensional control signal generation

h,s,v are controlled by the left hands. Three properties of the line segment of the thumb tip and the forefinger tip are continually computed:

- L: the length;
- c:(x,y,z): the center position;
- α : the angle to the canvas's xy plane around z axis;

Two conditions are also judged continually:

$$\begin{cases} z < D_1 \\ L > D_2 \end{cases} \tag{3}$$

where D_1 and D_2 are two certain thresholds. If the two conditions are not both satisfied, *h, s, v* are kept constant. Once they are both satisfied, the three properties are recorded as reference values: L_0, c_0:(x_0,y_0,z_0) and α_0. Afterwards, *h, s, v* are updated by the below computation if the line segment is changed:

$$\begin{cases} h' = h + f_h(\alpha - \alpha_0) \\ s' = s + f_s(x - x_0) \\ v' = v + f_v(y - y_0) \end{cases} \tag{4}$$

These control signals can be arbitrarily bind to brush properties to enable customized brush control. To be conveniently bind to brush parameters of different intervals in later procedure, all the control signals in eq (5) are normalized after the above computations.

V. Painting Model

A. Influence Space

The influence space is the geometrical bound of the paints flow. The paints flow act on the canvas in the intersection of the influence space and the canvas. As shown in Fig. 5, the influence space of the brush is an elliptic cone along the brush's z direction, which is always synchronized with the forefinger's direction. The paints flow are constrained in this cone. Its geometry can be described as 4 parameters:

$$\text{Cone:} \{r_b, \varepsilon_b, \theta_b, \varphi_b\} \tag{5}$$

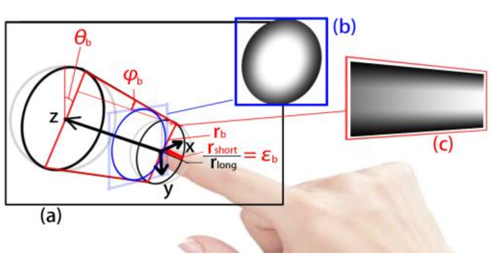

Fig. 5. Influence Space and Paints Distribution: (a) Influence Space; (b) and (c) paints alpha

The meaning of each is:

r_b: The long radius of the base ellipse;
ε_b: The ratio of the long radius versus the short radius;
θ_b: The roll angle of the cone around the *z* axis;
φ_b: The angle represents how much the paints flow diverges or converges;

In application, we add constraints of the length in *z* direction as [z_{min}, z_{max}].

B. Paints Distribution

We simplified the paints flow as a static field rather than a dynamic flow field. This is because that we intended to make enough visual fidelity of paints flow with high computation speed, rather than to make physical fidelity.

The paints is represented by {*h,s,v,alpha*}, each means hue, saturation, value and transparency. The distribution of *h,s,v* values is simplified as even distribution in the influence space, which equal to the *h,s,v* values in control signal. The alpha value distribution is represented as:

$$alpha(\mathrm{r},\mu,\mathrm{z}) = \begin{cases} d(z)a_0 & r_n(\mathrm{r},\mu,\mathrm{z}) < 1-f \\ d(z)a_0(1-\frac{r_n(\mathrm{r},\mu,\mathrm{z})-1+f}{s}) & 1-f < r_n(\mathrm{r},\mu,\mathrm{z}) < 1 \\ 0 & 1 < r_n(\mathrm{r},\mu,\mathrm{z}) \end{cases} \tag{6}$$

Where $alpha(\mathrm{r},\mu,\mathrm{z})$ means the alpha value at cylindrical coordinate$(\mathrm{r},\mu,\mathrm{z})$, a_0 and f are both normalized parameters, $r_n(\mathrm{r},\mu,\mathrm{z})$ is a normalized value which is computed by:

$$r_n(\mathrm{r},\mu,\mathrm{z}) = \frac{r}{\sqrt{(\varepsilon_b \cdot \mathrm{r}_b \cdot (\mathrm{z} \cdot \tan(\varphi_b)+\mathrm{r}_b) \cdot cos(\mu+\theta_b))^2 + (\mathrm{r}_b \cdot (\mathrm{z} \cdot \tan(\varphi_b)+\mathrm{r}_b) \cdot sin(\mu+\theta_b))^2}} \tag{7}$$

$d(z)$ is a function of variable z, and represented as:

$$d(z) = 1/(\frac{z \cdot \tan(\varphi_b)+r_b}{r_b})^2 \tag{8}$$

The above 3 equations means that the alpha distribution on a cross-section of z≠0 is just a simple scaled distribution on the plane *z=0*.

The parameters describing the paints flow can be summarized as:

$$\text{Paints:} \{h_0, s_0, v_0, a_0, f\} \tag{9}$$

C. Painting Steps

In physical painting, the brush continually paints on the canvas. To simulate this process, a digital brush periodically paint on the canvas along the stroke paths. We call a period as a painting step. In application, the painting process is always unsynchronized with the brush moving process, or the painting step is independent of movement step, as shown in Fig. 6. In the proposed brush model, the painting step is chosen as a generalized variable L:

$$L = w_m \int l + w_t \int t + w_e \int [|\alpha| + |\beta| + |\gamma|] \tag{10}$$

Where $\int l$ is the accumulation of distance change, $\int t$ is the accumulation of time change, and $\int [|\alpha| + |\beta| + |\gamma|]$ is the accumulation of Euler angle's change, and w_m w_t w_e are weights for the three parts. We set the painting step as L_{step}. In this process, four parameters describes the behavior of continual painting:

$$\text{Step}: \{w_m, w_t, w_e, L_{step}\} \quad (11)$$

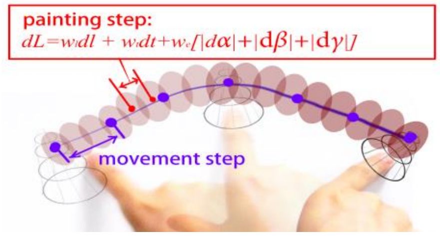

Fig. 6. movement step and painting step

D. Paints Transfer

The brush paints on the canvas at every painting step by 3 steps:

1. Interaction: compute the intersection of the influence space and the canvas, which is always an elliptic area on the canvas;

2. Compute paints: for every pixel on the canvas within the intersection, compute the color from the paints flow as C_{br}:{C_s,*alpha*}={{r_s,g_s,b_s},*alpha*}, in which *r,g,b* is the color value converted from *h,s,v* values from the paints flow, alpha is computed by equation (11);

3. Paints on the canvas: for this pixel, update its current color by:

$$C'_{canvas} = alpha \cdot C_s + (1 - alpha) \cdot C_{canvas} \quad (12)$$

where C_{canvas}, C'_{canvas} is the color of this pixel before and after update, respectively. The above formula is actually the alpha compositing.

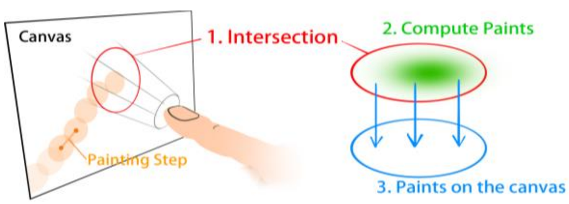

Fig. 7. A convenient strategy of binding control signals to brush parameters

VI. Brush Customization

The interaction between the user and the brush is achieved by binding control signals (equation (5)) to the brush properties. The brush properties can be described by multiple parameters, which is the union of (7) (11) (13):

$$\text{C} = \{\text{Cone}, \text{Paints}, \text{Step}\} = \{\{\text{r}_b, \varepsilon_b, \theta_b, \varphi_b\}, \{h_0, s_0, v_0, a_0, f\}, \{w_m, w_t, w_e, L_{step}\}\} \quad (13)$$

The multidimensional control signals (equation (5)) generated by the interaction model can be bind to any of the above 14 parameters. Here we introduce the default bindings of control signals to brush parameters. This binding strategy is convenient in practice. As shown in Fig. 8, the binding of control signals to brush parameters are:

$$B_{default}: (\text{h}, h_0), (\text{s}, s_0), (\text{v}, v_0), (b_1, f), (\varphi, \text{r}_b), (\omega, \varepsilon_b), (b_2, a_0) \quad (14)$$

Where every bracelet represent a pair of control signal and brush parameter. The meaning of each pair is:

- $(h, h_0), (\text{s}, s_0), (v, v_0)$: the color selected by the left hand is bind to the color of the paints flow.
- (b_1, f): the curvature of the right forefinger is used to control the softness of the paints flow. The more curved the forefinger, the softer the paints flow is, and thus the brush acts like a conventional airbrush.
- (φ, r_b): thumb open is used to control the general size of the paint flow. The more the thumb stretch away from the palm, the larger the brush is.
- (ω, ε_b): thumb roll is used to control the ellipticity of the paints flow.
- (b_2, a_0): the last three fingers' general curvature is used to control the general opacity of the paints flow.

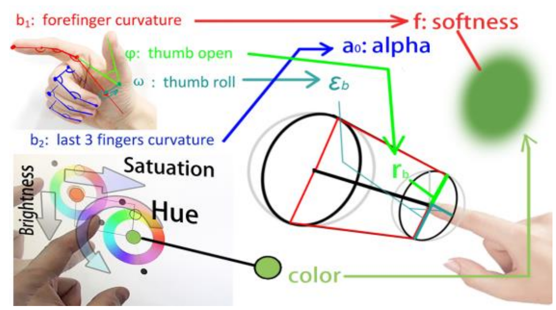

Fig. 8. Paints transfer

VII. Result

A. Freehand painting with a common screen as a tablet

To use the proposed brush, the user should first setup the hardware properly and perform points matching described in Section 4.B. Therefore, any common screen can be converted into a "tablet" screen which senses the user's hands movement. Therefore, the user can directly paint on the screen with two hands. Although this "tablet screen" cannot sense the pressure from the user's finger, it instead senses distance and orientation of the fingers relative to the screen, which is more than a tablet's capability. As shown in Fig. 9, the user creates multifarious stroke effects by flexible hand gestures.

B. Flexible Stroke Effects

This section demonstrates the flexibility of the interaction model. We alienate the brush parameters from the multidimensional control signal (section 4.D), and then tried 3 techniques to create different stroke effects. In this test, because all the control signals are not used, the user can only control the forefinger to manipulate the position and the orientation of the brush. Fig. 10 shows that even with this string constraint, various stroke effect can be created by intuitive techniques.

VIII. Conclusion

The proposed airbrush models both the user-brush and the brush-canvas interactions. The result shows that this technology enables immersive freehand painting on a common screen with a Leap Motion controller. With the brush, the user can create multifarious stroke effects by flexible interaction techniques and

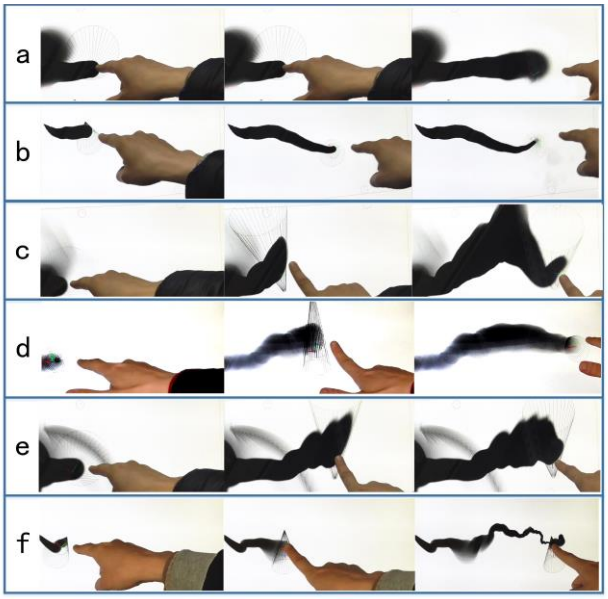

Fig. 9. Stroke Techniques by pure control of the forefinger's

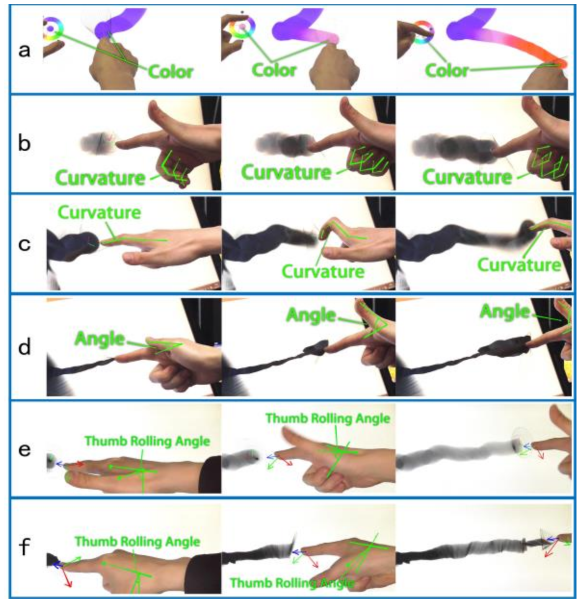

Fig. 10. Flexible brush control techniques

customizations. The technology also implies that other types of brush can be modelled and controlled in some similar ways of the proposed brush. In the future, it's worthy to simulate bristle brushes controlled in similar interaction techniques as the proposed brush. It's also promising to design special brushes that can only be properly control by hands' movement.

ACKNOWLEDGMENT

This research is supported by National Natural Science Foundation (Grant No. 61602213, No. 61103223 and No. 61602214) and National Science and Technology Support Program (Grant No. 2015BAH54F01).

REFERENCES

[1] Rudolf D, Mould D and Neufeld E, "A bidirectional deposition model of wax crayons," Computer Graphics Forum, Blackwell Publishing Ltd., vol. 24, pp. 27-39, 2005.

[2] Okaichi N, Johan H, Imagire T, et al, "A virtual painting knife," The Visual Computer, vol. 24, pp. 753-763, 2008.

[3] Kumagai K and Takahashi T, "PETICA: an interactive painting tool with 3D geometrical brushes," ACM SIGGRAPH ASIA 2010 Sketches, ACM, pp. 35, 2010.

[4] DiVerdi S, Krishnaswamy A and Hadap S, "Industrial-strength painting with a virtual bristle brush," Proceedings of the 17th ACM Symposium on Virtual Reality Software and Technology, ACM, pp. 119-126, 2010.

[5] Chu N S H and Tai C L, "Real-time painting with an expressive virtual Chinese brush," Computer Graphics and Applications, IEEE, vol. 24, pp. 76-85, 2004.

[6] Jingwan Lu, Connelly Barnes, Connie Wan, et al, "DecoBrush: Drawing Structured Decorative Patterns by Example," Proceedings of ACM SIGGRAPH, ACM, 2014.

[7] Sarah Abraham and Donald Fussell, "Smoke Brush," Proceedings of the Workshop on Non-Photorealistic Animation and Rendering, ACM, 2014.

[8] Lee S, Olsen S C and Gooch B, "Interactive 3D fluid jet painting," Proceedings of the 4th international symposium on Non-photorealistic animation and rendering, ACM, pp. 97-104, 2006.

[9] Konieczny J and Meyer G, "Airbrush simulation for artwork and computer modeling," Proceedings of the 7th International Symposium on Non-Photorealistic Animation and Rendering, ACM, pp. 61-69, 2009.

[10] Shilkrot R, Maes P, Paradiso J A, et al, "Augmented Airbrush for Computer Aided Painting (CAP)," ACM Transactions on Graphics (TOG), vol.34, pp. 19, 2015.

[11] Richard Greene, "The drawing prism: a versatile graphic input device," ACM SIGGRAPH Computer Graphics, ACM, 1985.

[12] Sun M, Cao X, Song H, et al, "Enhancing naturalness of pen-and-tablet drawing through context sensing," Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces, ACM, pp. 83-86, 2011.

[13] Harada S, Saponas T S and Landay J A, "VoicePen: augmenting pen input with simultaneous non-linguisitic vocalization," Proceedings of the 9th international conference on Multimodal interfaces, ACM, pp. 178-185, 2007.

[14] Peter Vandoren, Luc Claesen, Tom Van Laerhoven, et al, "FluidPaint: an Interactive Digital Painting System using," Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces, ACM, 2009.

[15] Kazi R H, Chua K C, Zhao S, et al, "SandCanvas: A multi-touch art medium inspired by sand animation," Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ACM, pp. 1283-1292, 2011.

[16] Jeremy Sutton, "Air painting with Corel Painter Freestyle and the leap motion controller: a revolutionary new way to paint!" ACM SIGGRAPH 2013 Studio Talks, ACM, pp. 21, 2013