A VR-based Calligraphy Writing System with Force Reflection

Chang-Wei Liang, Mu-Cheng Hsieh, and Kuu-Young Young*

Department of Electrical Engineering, National Chiao Tung University, Taiwan

(Received 29 June 2011; Accepted 7 October 2011; Published on line 1 December 2011)

*Corresponding author: kyoung@mail.nctu.edu.tw

DOI: [10.5875/ausmt.v1i2.90](https://doi.org/10.5875/ausmt.v1i2.90)

Abstract: Painting software for art production currently uses the mouse, keyboard and graphic tablet as the primary input devices. However, for Chinese-style calligraphy, these devices have difficulty replicating the effect of the traditional calligraphy brush. In this paper, we aim to develop a calligraphy writing system in a virtual environment, using a force-reflection joystick as the input device. A virtual brush model, including the emulation of both geometric- and physical-based deformations, is proposed to enhance the veracity of the visual effect. In experiments, the performance of the proposed system is compared with results using a calligraphy brush and a graphic tablet.

Keywords: calligraphy; virtual environment; force-reflection joystick; virtual brush model

Introduction

The rapid development in computer hardware and painting software has led some artists to use computers as a medium for creating art work. Most graphic systems currently rely on the mouse, keyboard, and graphic tablet as the primary input devices [1-5]. However the mechanics and sensation of using these devices is very different from that of traditional brushes and artists usually need to devote considerable time and effort to becoming proficient in their use.

In this paper, we propose a calligraphy writing system based on virtual reality techniques [1-6], and adopt a force-reflection joystick as the input device [6]. Our focus is on the brush model, a crucial aspect for achieving a realistic effect. To achieve real-time processing, we also optimize the computational efficiency of this brush model. A force-reflection joystick can simulate the haptic sensation similar to that of manipulating a real calligraphy brush. As a demonstration, we apply the proposed system for calligraphy writing and compare its performance to those of a calligraphy brush and a graphic tablet.

Proposed VR-based Calligraphy Writing System

Figure 1 shows the system organization of the proposed VR-based calligraphy writing system. The user manipulates the joystick as if she/he were writing with a calligraphy brush, and the position and orientation of this joystick are sent to the brush model for computing the dynamics (including shape) of the virtual brush when interacting with the virtual paper. Based on the brush dynamics, the force computing system synthesizes the haptic effect in writing and sends it to the user via the force-reflection joystick. Meanwhile, the visual system uses the brushes' geometrical information to provide the user with visual feedback.

1. Brush Modeling

A brush model that can closely emulate the dynamics of a brush is a key element of the proposed system. The hair type, wetness, and painting mechanism of a traditional calligraphy brush may all affect the

Figure 1. System organization of the proposed VR-based calligraphy writing system.

dynamics of the brush which, in turn, influences the shape and stroke of the brush. Below we discuss issues related to brush deformation and the mass-spring-damping model with an implicit integration process.

1.1 Deformation Technique

The techniques used to model deformable objects can be classified into three major categories: geometric-based, physical-based, and hybrid. The geometric-based techniques focus on the appearance of the deformable objects, but not on their physical

properties. The physical-based techniques use a physical model to emulate the object's behavior. The hybrid technique combines the geometric and physical techniques. Given the completeness of the hybrid method, we adopt it here for our proposed brush model.

1.2 Mass-Spring-Damper Model

The mass-spring-damper model uses a number of mass points connected by springs and dampers for elastic-object modeling, and is a popular method for describing physical-based deformations [1]. The behavior of the mass, spring, and damper are described by Newton's equations of motion.

1.3 Implicit Integration

When dealing with Newton's equations of motion in a computer system, a stable integration method is important, and the implicit integration method is suitable for this mass-spring-damper model. The implicit integrations for the i^{th} mass-point are formulated as

$$\mathbf{v}_i^{t+h} = \mathbf{v}_i^t + h \mathbf{F}_i^{t+h}/m_i \quad (1)$$

$$\mathbf{x}_i^{t+h} = \mathbf{x}_i^t + \mathbf{v}_i^{t+h}h \quad (2)$$

where $\mathbf{x}_i^t = [x_{i,x}^t \ x_{i,y}^t \ x_{i,z}^t]^T$, $\mathbf{v}_i^t = [v_{i,x}^t \ v_{i,y}^t \ v_{i,z}^t]^T$, $\mathbf{F}_i^t = [F_{i,x}^t \ F_{i,y}^t \ F_{i,z}^t]^T$, m_i is the mass of the i^{th} mass-point, and h is the integration period. The motion equation for the mass-spring-damper model of the n mass points is described by the linear equation [7]:

$$(\mathbf{M} - h^2\mathbf{J})\Delta\hat{\mathbf{v}}^{t+h} = h(\hat{\mathbf{F}}^t + h\mathbf{J}\hat{\mathbf{v}}^t) \quad (3)$$

where $\Delta\hat{\mathbf{v}}^{t+h} = \hat{\mathbf{v}}^{t+h} - \hat{\mathbf{v}}^t$, $\mathbf{M}$ is a matrix of the mass in

Chang-Wei Liang received his B.S. degree in the Department of Electrical Engineering, Yuan-Ze University, in 2007, and M.S. degree in the Institute of Electrical Control Engineering, National Chiao Tung University, in 2009. Between 2009 and 2010, he served as a communication officer in Taiwan army. His research interests include teleoperation and human-computer interfaces.

Mu-Cheng Hsieh was born in Miaoli, Taiwan, 1969. In 1989, he graduated from the five-year program in the Division of Electrical Engineering, National Taipei Institute of Technology (renamed to National Taipei University of Technology in 1997). He received the M.S. degree in Electrical and Control Engineering from National Chiao Tung University in 2001, where he is currently pursuing the Ph.D. degree. Since 1993, he has been working at National Tsing Hua University, where he is currently a Technical Specialist at the Computer and Communication Center. His research interests include human-machine systems, Internet services, computer haptics, and virtual motion constraint design for simulation systems and robot manipulation.

Kuu-young Young was born in Kaohsiung, Taiwan, 1961. He received his B.S. degree in Electrical Engineering from National Taiwan University, Taiwan, in 1983, and M.S. and Ph.D. degrees in Electrical Engineering from Northwestern University, Evanston, IL, U.S.A., in 1987 and 1990, respectively. Between 1983 and 1985, he served as an electronic officer in Taiwan Navy. Since 1990, he has been with the Department of Electrical Engineering at National Chiao-Tung University, Hsinchu, Taiwan, where he is currently a Professor. He served as the chairman of the department from 2003 to 2006, and the associate dean of Electrical and Computer Engineering College, NCTU, from 2007 to 2010. His research interests include robot compliance control, robot learning control, robot calibration and path planning, robot helper, and Science, Technology, and Society (STS).

the mass-spring-damper model, and $\mathbf{J} = \partial \hat{\mathbf{F}} / \partial \mathbf{x}$. Equation (3) can be solved via the integration process below [8]. As the difference of the speed for each mass-point is given via the independent equation, its time complexity is $O(n)$. This process is formulated as

$$\Delta \hat{\mathbf{v}}_i^{t+h(0)} = h \mathbf{W}_i^{-1} \hat{\mathbf{F}}_i^t \quad (4)$$

$$\begin{aligned} \Delta \hat{\mathbf{v}}_i^{t+h(k)} &= \Delta \hat{\mathbf{v}}_i^{t+h(0)} \\ &+ h^2 \mathbf{W}_i^{-1} \sum_{(i,j)} \mathbf{J}_{ij} \Delta \hat{\mathbf{v}}_j^{t+h(k-1)} \end{aligned} \quad (5)$$

where $\mathbf{W}_i = \mathbf{M}_i - h^2 \mathbf{J}_{ii}$. During the process, we first need to calculate the 0th iterative $\Delta \hat{\mathbf{v}}_i^{t+h(0)}$, and determine the number of iterations k . Speed change for each mass-point $\Delta \hat{\mathbf{v}}_i^{t+h(k)}$ is then obtained iteratively. Finally, we can obtain the position of each mass-point via Equation (2) by executing Equation (5) for k -times.

2. Visual System

The developed visual system deals with both the virtual environment and the ink deposition. The virtual environment provides a virtual calligraphy brush and virtual paper. The shape of the brush is determined by the brush model, so that the user can visualize the spreading and bending of the brush. The virtual environment is built using a 3D graphic engine, and the 2D feedback image can be obtained through the perspective projection. Meanwhile, a simple approach combining texture mapping and orthogonal projection is used to ensure that the ink is only deposited on the area that has contact with the brush surface.

3. Force Computing System

Force feedback is very important in a virtual calligraphy writing system in that it allows the user to feel the pressing depth and the moving speed of the brush, and to sense the brushes' wetness via its stiffness. To maintain continuous force feedback, force computing should be as simple and fast as possible. In this system, force feedback serves two purposes: (1) it constrains the user's workspace and (2) it provides haptic sensations during the manipulation of the brush, such as the gravity of the brush, the normal force when the brush is pressed, and the frictional force when the brush is moved on the paper.

Proposed Brush Model

The brush skeleton and geometrical-based deformation for the brush surface are described using physical-based deformation.

1. Brush Skeleton

Figure 2 shows the geometric structure of the brush model. The model is divided into N layers, with the red line indicating the mass-spring-damper model, the orange line the virtual spine, and the black lateral nodes the brush surface.

Figure 2. The geometric structure of the brush model.

1.1 One-Dimensional Mass-Spring-Damper Model

We adopted a 1-dimensional mass-spring-damper model to emulate the brush skeleton, which is divided into N layers, as shown in Figure 3. Compared with a 3-dimensional structure, the advantage of a 1-dimensional structure lies in its computational efficiency. Fewer mass-points and damper-spring links require a much shorter integration period, thus reducing the integral error and speeding up collision detection.

Figure 3. The 1-dimensional mass-spring-damper model.

1.2 Virtual Spine

Figure 4. The 1-dimensional mass-spring-damper model in contact with an object.

Following contact with an object, the steady state of this 1-dimensional mass-spring-damper model cannot be restored to the original state, as shown in Figure 4. In Figure 4(a), there is no contact and the brush skeleton is initially straight. In Figure 4(b), the brush skeleton contacts with an object and starts to deform. In Figure

4(c), the brush skeleton disengages from the object and shifts to a new steady state. Finally, in Figure 4(d), the new steady state is achieved, different from the original one, with the spring length equal to that of the initial state.

To deal with this phenomenon, we introduce a virtual spine to allow the brush skeleton return to its original state. The virtual spine is composed of N nodes (denoted by $\mathbf{VS}_{0,1,\dots,N-1}$), which correspond to the mass-points ($\mathbf{S}_{0,1,\dots,N-1}$) of the mass-spring-damper model. Each pair of $\mathbf{VS}_i$ and $\mathbf{S}_i$ is connected by a spring-damper. The initial positions of the spine nodes are the same as those of the mass-points, and the distance between two neighboring nodes is constant. Figure 5 shows the behavior of the virtual spine, in which the red line indicates the mass-spring-damper model, the orange line the virtual spine, and the green lines the spring-damper connections between $\mathbf{VS}_i$ and $\mathbf{S}_i$. In Figure 5(a), there is no contact and the brush skeleton is straight. In Figure 5(b), it makes contact with an object and starts to deform. In Figure 5(c), it disengages from the object and begins to shift to a new steady state. Finally, in Figure 5(d), it reverts to the original state with the help of the virtual spine.

Figure 5. The 1-dimensional mass-spring-damper model with the virtual spine in contact with an object.

Figure 6. Virtual spine-deformation with the 1-dimensional mass-spring-damper model.

1.3 Brush Plasticity

With the virtual spine derived above, the steady state of the brush skeleton will not be permanently maintained following the deformation. However, the shape of a real hair brush may acquire a persistent deformation when the user keeps the brush at a consistent angle for a period of time or if the ink dries. To deal with this, the virtual spine is further enhanced by including a deformation algorithm. The basic idea is that the shape of the virtual spine may deform according to the variation of the 1-dimensional mass-spring-damper model.

Virtual-Spine Deformation Algorithm:

Step 1: The deformation of the virtual spine $\mathbf{V}_{\text{goal}}$, as shown in Figure 6, is set as

$$\mathbf{V}_{\text{goal}} = (\mathbf{S}_{N-1}^t - \mathbf{S}_{N-2}^t) / \|\mathbf{S}_{N-1}^t - \mathbf{S}_{N-2}^t\| \quad (6)$$

Step 2: Let $\mathbf{V}_0^{t+h}$ be the direction of the brush shaft, and the vector of the tip of the virtual spine $\mathbf{V}_{N-2}^{t+h}$ is obtained by

$$\mathbf{V}_{N-2}^{t+h} = (1 - D)\mathbf{V}_{N-2}^t + D\mathbf{V}_{\text{goal}} \quad (7)$$

$$\mathbf{V}_{N-2}^{t+h} = \mathbf{V}_{N-2}^{t+h} / \|\mathbf{V}_{N-2}^{t+h}\| \quad (8)$$

where D is a variable to adjust the deformation speed of the virtual spine.

Step 3: Use interpolation to get $\mathbf{V}_i^{t+h}$ for $i \in [1, \dots, N-3]$ and normalize them:

$$\mathbf{V}_i^{t+h} = \left(1 - \frac{i}{N-1}\right)\mathbf{V}_0^{t+h} + \frac{i}{N-1}\mathbf{V}_{N-2}^{t+h} \quad (9)$$

$$\mathbf{V}_i^{t+h} = \mathbf{V}_i^{t+h} / \|\mathbf{V}_i^{t+h}\| \quad (10)$$

Step 4: Let the positions of spine nodes be obtained by

$$\mathbf{VS}_0^{t+h} = \mathbf{S}_0^{t+h} \quad (11)$$

$$\mathbf{VS}_{i,i \in [1,\dots,N-1]}^{t+h} = \mathbf{VS}_{i-1}^{t+h} + L\mathbf{V}_{i-1}^{t+h} \quad (12)$$

where L is the initial length of the spring.

2. Brush Surface

After determining the deformation of the brush skeleton, we move on to locate the positions of the lateral nodes that construct the brush surface. Initially, the cross section of the brush in free space is a circle. But

once the brush is pressed, the cross section becomes an ellipse, as shown in Figure 7. A geometric-based deformation algorithm was developed, in which the lengths and directions of the short- and long- axis of the ellipse are determined according to the wetness of the brush, bending of the skeleton, and contact between the brush and the paper.

Figure 7. Locate the lateral nodes on the brush surface via the ellipse.

Figure 8. Obtain the direction of the short axis by interpolating the cases of (a) Q mass-points and (b) no mass-point contacting the paper.

● Brush-Surface Deformation Algorithm:

Step 1: Determine the direction of the short axis $\mathbf{SA}_i^{t+h}$ for the three cases shown in Figure 8:

Case 1: If the i^{th} mass-point $\mathbf{S}_i$ contacts the paper (see Figure 8(a), when $i \geq N - Q$), then

$$\mathbf{SA}_i^{t+h} = \mathbf{N}_{\text{paper}} \quad (13)$$

Case 2: If Q mass-points contact the paper, but the mass-point $\mathbf{S}_i$ does not (see Figure 8(a), when $i \in [0, \dots, N - Q - 1]$), then

$$\begin{aligned} \mathbf{SA}_i^{t+h} &= \left[1 - \frac{N-Q-i}{N-Q}\right] \mathbf{SA}_{N-Q}^{t+h} \\ &+ \left[\frac{N-Q-i}{N-Q}\right] \mathbf{SA}_0^{t+h} \end{aligned} \quad (14)$$

Case 3: If no mass-point contacts the paper (see Figure 8(b)), then

$$\mathbf{SA}_{N-1}^{t+h} = \mathbf{SA}_{N-1}^t \quad (15)$$

$$\begin{aligned} \mathbf{SA}_{i \in [1, \dots, N-2]}^{t+h} &= \left[1 - \frac{N-i-1}{N-1}\right] \mathbf{SA}_{N-1}^{t+h} \\ &+ \left[\frac{N-i-1}{N-1}\right] \mathbf{SA}_0^{t+h} \end{aligned} \quad (16)$$

Step 2: Decide the direction of the long axis by

$$\mathbf{LA}_{i \in [1, \dots, N-1]}^{t+h} = \mathbf{SA}_i^{t+h} \times (\mathbf{S}_i - \mathbf{S}_{i-1}) \quad (17)$$

Step 3: Normalize the directions of the long and short axes by

$$\mathbf{LA}_{i \in [1, \dots, N-1]}^{t+h} = \mathbf{LA}_i^{t+h} / \|\mathbf{LA}_i^{t+h}\| \quad (18)$$

$$\mathbf{SA}_{i \in [1, \dots, N-1]}^{t+h} = \mathbf{SA}_i^{t+h} / \|\mathbf{SA}_i^{t+h}\| \quad (19)$$

Step 4: Compute the bending degree for each layer (Tilt_i), and the total bending (Tilt_{all}) by

$$\mathbf{V}_{i \in [1, \dots, N-1]} = \mathbf{S}_i - \mathbf{S}_{i-1} \quad (20)$$

$$\text{Tilt}_{i \in [1, \dots, N-2]} = 1 - \mathbf{V}_{i+1} \cdot \mathbf{V}_i \quad (21)$$

$$\text{Tilt}_{\text{all}} = (\sum_{i=1}^{N-2} \text{Tilt}_i) / (N - 2) \quad (22)$$

Step 5: Calculate the expected length for the long axis $\text{RL}_{\text{goal},i}$ for the two cases:

Case 1: If the mass-spring-damper model contacts the paper, then

$$\begin{aligned} \text{RL}_{\text{goal},i} &= \text{RL}_i^t + (\text{RL}_{\text{max},i} - \text{RL}_i^t) \\ &\cdot (1 - W_i) \left(\frac{i}{N-1}\right) \\ &\cdot (\text{TF} \cdot \text{Tilt}_i + (1 - \text{Tilt}_i) \text{Tilt}_{\text{all}}) \end{aligned} \quad (23)$$

where $\text{RL}_{\text{max},i}$ is the maximum length for the i^{th} long axis, W_i is the wetness, and TF is an adjustable weighting factor for the bending degree.

Case 2: If no mass-point contacts the paper, then

$$\text{RL}_{\text{goal},i} = \text{RL}_{\text{initial},i} \quad (24)$$

Step 6: Calculate the lengths of the long and short axes by

$$\text{RL}_i^{t+h} = \text{SD} \cdot \text{RL}_{\text{goal},i} + (1 - \text{SD}) \text{RL}_i^t \quad (25)$$

$$\text{RS}_i^{t+h} = \text{RL}_{\text{initial},i}^2 / \text{RL}_i^{t+h} \quad (26)$$

where SD is an adjustable factor for the surface deformation.

Step 7: Locate the lateral nodes $\mathbf{G}_{i,j}^{t+h}$ on the brush surface by an elliptic equation (illustrated in Figure 7):

$$\begin{aligned} \mathbf{G}_{i,j}^{t+h} &= \mathbf{S}_i + \text{LA}_i^{t+h} \text{RL}_i^{t+h} \cos\left(\frac{2\pi j}{K}\right) \\ &+ \text{SA}_i^{t+h} \text{RS}_i^{t+h} \sin\left(\frac{2\pi j}{K}\right) \end{aligned} \quad (27)$$

System Implementation

The system consists of a personal computer, used to construct the virtual environment and compute the visual and haptic feedback information, and a 6-DOF Sensable Phantom Omni force-reflection joystick [9], used to manipulate virtual objects. The virtual environment, developed using Microsoft C++ 6.0 in Microsoft Windows XP, includes the brush model, the visual system, and the force computing system.

In computing the brush model, the system needs to detect object collisions, and calculate force applied on the mass point, implicit integration, and the deformations of both virtual spine and brush surface. System frequency is at 25 kHz, which is fast enough for implicit integration in dealing with the dynamics of the brush skeleton in real-time. The environment generates a virtual calligraphy brush and paper with a frame rate of about 60 Hz. Ink is only allowed to deposit where contact with the brush surface occurs.

Two kinds of force feedback are used in the system. The first constrains the user's workspace. For instance, to prevent the virtual brush from penetrating the paper, the virtual brush is required to always be above the paper [10]. The second provides the haptic sensations during brush manipulation. Gravity, normal force, and frictional force effects are considered. A constant force normal to the paper is used for the gravity effect. A normal force is also used when the brush contacts with the paper. This normal force $\mathbf{F}_n$, illustrated in Figure 9, is formulated as

$$\mathbf{F}_n = w_n \frac{D_p}{L_p} \mathbf{N}_{paper} \quad (28)$$

where w_n is a weighting factor for the normal force, D_p is the depth of the virtual spine's penetration into the paper, and L_p is the projection length of the virtual spine onto the normal vector of the paper.

For the friction effect between the brush and paper, we combine two kinds of forces: (1) the viscosity force for the effect of ink viscosity and (2) the inertia force for the effect of liquid tension. The viscosity force is formulated as

$$\mathbf{F}_{vis} = -w_f \frac{n}{N} \mathbf{V}_p \quad (29)$$

where n is the number of mass-points in contact with the paper, N is the number of mass-points, $\mathbf{V}_p$ is the projection of the brush speed on the paper surface, and w_f is a weighting factor. As for the inertia force, we assume that the brush pulls a virtual mass point via a virtual spring and virtual damper, with the mass of this point set to be proportional to the contact area as

$$m = m_i + n \cdot m_{mp} \quad (30)$$

where m_i is the initial mass, m_{mp} is the mass in the mass-spring-damper of the brush model, and n is the same as in Equation (29). The inertia force $\mathbf{F}_i$ is then derived as

$$\mathbf{F}_i = -k_s(\mathbf{x}_j - \mathbf{x}_m) + k_d \mathbf{v}_m \quad (31)$$

where k_s is the stiffness of the virtual spring, k_d is the damping of the virtual damper, $\mathbf{x}_j$ is the joystick position, and $\mathbf{x}_m$ and $\mathbf{v}_m$ are respectively the position and velocity of this mass point, which can be derived by the following explicit integration:

$$\mathbf{v}_m^{t+h} = \mathbf{v}_m^t - h(\mathbf{F}_i^t)/m \quad (32)$$

$$\mathbf{x}_m^{t+h} = \mathbf{x}_m^t + \mathbf{v}_m^{t+h} h \quad (33)$$

Figure 9. Illustration of the normal force.

Experiments

Experiments were conducted to demonstrate the proposed system, and its performance was compared with those of a calligraphy brush and a graphic tablet, as shown in Figure 10. Test subjects first answered a questionnaire to collect basic information. They then spent about 10 minutes learning the eight principle strokes (i.e., 'Dot,' 'Horizontal,' 'Erect,' 'Hook,' 'Raise,' 'Bend,' 'Slant,' and 'Pressing forcefully') of the character '永' (Yong) [11]. They then wrote 'Yong' using a calligraphy brush, a graphic tablet, and the proposed system in a random sequence. For comparison, we asked the user to mimic an example of 'Yong' written by Wang Hsi-Chih, a famous Chinese calligrapher.

(a)

(b)

(c)

Figure 10. Writing the character '永' (Yong) with (a) a calligraphy brush, (b) a graphic tablet, and (c) the proposed system.

Ten students, all native Chinese-speakers familiar with Chinese character writing, participated in the experiment. Each participant wrote the character five times with each method, and self-selected the best result for each method. Figure 11 shows the self-selected results, while Figure 12 indicates the degree of the resemblance of the results from both the graphic tablet and proposed system to that obtained with the calligraphy brush, based on the subjective assessment of each participant. Figure 12 does not provide a score for

the brush shape used with the graphic tablet, because the graphic tablet does not provide different brush shapes. In terms of both haptic sensation and manipulation, the proposed system was found to better resemble the experience of using the calligraphy brush than did the graphic tablet. As for ink deposition, the performances of the proposed system and graphic tablet were similar. Figure 13 indicates that the proposed system outperformed the graphic tablet in terms of haptic sensation, ink deposition, and manipulation. Meanwhile, some subjects reported that the manipulation of the proposed system was not intuitive enough for them, because they were not accustomed to using the 3D VR system.

	Calligraphy brush	Graphic tablet	Proposed system
Subject 1			
Subject 2			
Subject 3			
Subject 4			
Subject 5			
Subject 6			
Subject 7			
Subject 8			
Subject 9			
Subject 10			

Figure 11. '永' (Yong) written by ten subjects using a calligraphy brush, a graphic tablet, and the proposed system.

Figure 12. Resemblance of the graphic tablet and the proposed system to the calligraphy brush.

Figure 13. Comparison between the graphic tablet and proposed system.

Figure 14. Difficulty indices of the strokes.

For further analysis, Figure 14 shows the difficulty indices with which the subjects graded the degree of difficulty when writing the eight strokes with the graphic tablet and proposed system. The subjects were asked to select three strokes they found most difficult to write. Most considered 'Dot,' 'Hook,' and, 'Pressing forcefully' more difficult to execute with both systems. Some

reported difficulty controlling the pressing force on the 2D tablet, leading to unsatisfactory execution of the starting and end parts of these three strokes. In addition, the frictional force on the graphic tablet is insufficient to maintain a constant stroke. As for the proposed system, some subjects reported difficulty in properly executing the timing of the contact for these strokes, given the disparity between the 3D input (i.e., the joystick) and the 2D visual feedback. To provide an aesthetic analysis, we invited 10 calligraphy experts to subjectively grade the writing in Figure 11, without knowing which system was used to generate the writing. The final scores were computed as the mean values of their respective scores. Figure 15 indicates that the results produced by traditional calligraphy brush were ranked most highly, followed by the proposed system, with the graphic tablet results placing third.

Figure 15. Aesthetic performance evaluation.

Conclusion

This paper describes a VR-based calligraphy writing system with force reflection. This system provides both visual and haptic feedback, allowing the user to become immersed in the virtual writing environment. In experiments, the performance of the proposed system was compared with a traditional calligraphy brush and a graphic tablet. The results show that, compared to the graphic tablet, the proposed system results in writing which better resembles that done with the calligraphy brush. In future work, we aim to develop more realistic brush and ink depositing models, and more subtle haptic sensations for the user.

Acknowledgement

This work was supported in part by the National Science Council under grant NSC 99-2221-E-009-157.

References

- [1] N. S. H. Chu and C. L. Tai, "Real-time painting with an expressive virtual Chinese brush," *IEEE Computer Graphics and Applications*, vol. 24, pp. 76-85, 2004.
doi: [10.1109/MCG.2004.37](https://doi.org/10.1109/MCG.2004.37)
- [2] J. Lee, "Simulating oriental black-ink painting," *IEEE Computer Graphics and Applications*, vol. 19, pp. 74-81, 1999.
doi: [10.1109/38.761553](https://doi.org/10.1109/38.761553)
- [3] S. Strassmann, "Hairy brushes," in *13th annual conference on Computer graphics and interactive techniques*, New York, NY, USA, 1986, pp. 225-232.
doi: [10.1145/15922.15911](https://doi.org/10.1145/15922.15911)
- [4] H. T. F. Wong and H. H. S. Ip, "Virtual brush: a model-based synthesis of Chinese calligraphy," *Computers & Graphics*, vol. 24, pp. 99-113, 2000.
doi: [10.1016/S0097-8493\(99\)00141-7](https://doi.org/10.1016/S0097-8493(99)00141-7)
- [5] M. S. Yoh, "The reality of virtual reality," in *Seventh International Conference on Virtual Systems and Multimedia*, Berkeley, CA , USA, 2001, pp. 666-674.
doi: [10.1109/VSM.2001.969726](https://doi.org/10.1109/VSM.2001.969726)
- [6] B. Baxter, V. Scheib, M. C. Lin, and D. Manocha, "DAB: interactive haptic painting with 3D virtual brushes," in *28th annual conference on Computer graphics and interactive techniques*, New York, USA, 2001, pp. 461-468.
doi: [10.1145/383259.383313](https://doi.org/10.1145/383259.383313)
- [7] D. Baraff and A. Witkin, "Large steps in cloth simulation," in *25th annual conference on Computer graphics and interactive techniques*, New York, USA, 1998, pp. 43-54.
doi: [10.1145/280814.280821](https://doi.org/10.1145/280814.280821)
- [8] Y. M. Kang and H. G. Cho, "Real-Time Animation of Complex Virtual Cloth with Physical Plausibility and Numerical Stability," *Presence: Teleoperators and Virtual Environments*, vol. 13, pp. 668-680, 2004.
doi: [10.1162/1054746043280637](https://doi.org/10.1162/1054746043280637)
- [9] PHANTOM OMNI, Sensable, [Online]. Available: <http://www.sensable.com/haptic-phantom-omni.htm> [Accessed: June 9, 2011]
- [10] C. B. Zilles and J. K. Salisbury, "A constraint-based god-object method for haptic display," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Pittsburgh, PA , USA, 1995, pp. 146-151.
doi: [10.1109/IROS.1995.525876](https://doi.org/10.1109/IROS.1995.525876)
- [11] "Eight Principles of Yong," [Online]. Available: http://en.wikipedia.org/wiki/Eight_Principles_of_Yong [Accessed: June 20, 2011]