PuppetAnimator: A Performative Interface for Experiencing Shadow Play

Yue Shi, Yue Suo, Shang Ma, Yuanchun Shi
Key Laboratory of Pervasive Computing
Dept. of Computer Science, Tsinghua University
Beijing, China
shi-y04@mails.tsinghua.edu.cn, suoyue@tsinghua.edu.cn, msjianghu@gmail.com, shiyc@tsinghua.edu.cn

***Abstract*— We present PuppetAnimator, a novel interface that enables non-expert users to experience funny shadow play and create puppet animation by manipulating the character with simple pointing device. Inspired by the real shadow puppet performance, we employ the "sticks from hands to the puppet" metaphor for interaction and design a set of motion modes including Translation, Swing, Drag, Rotation and Two-joint Drive. Based on motions modes' combination, several puppet templates are prebuilt for the user to customize the skin with images and control channels with input devices. Then the user can manipulate the puppet to arrange actions. Features such as recording the user's performance, replaying and saving as video files are also provided. This paper describes the system design rationale and implementation. We demonstrate PuppetAnimator with several puppet actions and pilot study shows that with the system, users can easily make animation of a shadow puppet's action for storytelling.**

Keywords-multi-point; performative interface; shadow puppet

I. Introduction

Chinese shadow play is a characteristic form of art in which shadow puppets are manipulated in front of the illuminated backdrop for storytelling and entertainment. During the performance, each puppet is animated to perform various actions by several sticks held by a puppeteer. Being an ancient drama, shadow play also generates a modern cartoon style which is very popular among people.

However, making and performing a real shadow puppet is a very difficult task which keeps non-expert users from experiencing this form of art. The process to make shadow puppet animation is also time-consuming even with the aid of current animation software. When a user wants to make a shadow puppet style animation to tell a story or import into other system as a material, he has to insert several key frames in a time line, pose the gesture of the characters in each frame with a lot of clicks and drags but not be able to see the action until the last minute. If a tool can enable the user to experience the shadow play and at the same time, quickly create shadow puppet animation in the way similar to puppeteer's, the editing process will be made easier, more straightforward and interesting.

In recognition of the above issues, we have developed PuppetAnimator (Figure 1), a novel interactive system that allows non-expert users to experience shadow play and create shadow puppet style animation by performing the actions of the puppets with point devices. With the system, users can design virtual puppets, arrange their actions and record the ideas as animation clips. The system provides a set of motion modes which are elemental actions in the real shadow play. With the combination of those modes, puppet templates can be designed that simulate the physical behavior of a real puppet in respond to users' inputs. In the performance stage, we make use of a real performance metaphor as the interaction method - holding point input devices like sticks to control the puppet. To make it more accessible to novice users, we choose simple and inexpensive point devices such as mouse and touch pad.

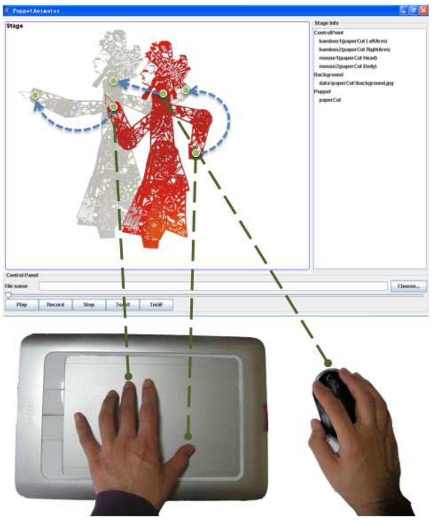


Figure 1. Up: the screenshot of PuppetAnimator. Down: the approach to interact with PuppetAnimator. Input points (green dots) from different devices can be mapped to the points of force application (green dots with white circles) on the puppet to perform an action (blue dashed lines with an arrow).

Our work is a kind of performance animation that provides an approach to enable junior level users to perform their own shadow play. The first contribution of our work is the interface to support this which is easy to learn and straightforward to use. The devices we use to generate input are available to everyone. Second, the set of motion modes we designed constitutes the basic tool box for the puppet manipulation. And the interaction method based on it is demonstrated with several actions such as walking, jumping and dancing. The work is aimed to take effect in informal circumstances such as improvised storytelling and animated GIF production.

II. RELATED WORK

Designing interface to manipulate a puppet and generate skeletal animation is a long standing problem that many techniques and systems are provided.

Some systems have enabled users to control a puppet. In PuppetWall [4], a puppet can follow a stick held in a user's hand. Video puppetry [3] allows users to manipulate paper puppets as control. Although articulated puppets have been taken into consideration, the current interaction is limited: a user can only rotate the segment to change its orientation while shadow puppet manipulation needs more complicated motion modes such as multi-jointed motion and simple pendulum movement. Moreover, there are some interaction techniques looking at interacting with digital objects, especially in tabletop systems. In Andrew's work [1], contact points with other input information detected by the touch screen are combined with physical simulation. As far as we know, none of these systems provides the off-the-shelf interaction techniques that are fine-grained enough to manipulate a shadow puppet.

A lot of software enables the creation of skeletal animation. For example, Flash's bone tool [11] can add virtual bones to the animation character from segments of limbs. To animate the bone structure, a user needs to insert a number of key frames and in each frame, drag the character in a new position. These tools are powerful in creating skeleton animation with rich effects. However, in informal application, this approach is uneconomical in time and not straightforward since the character cannot act until finished.

Performance animation as another method has also been studied. Various input devices such as human body with sensors, joysticks, and stylus are used for the animation creation. Sageev [9] establishes a tangible interface to control the 3D human character's segment orientation. Matthew [7] designs an interface where user can use input sketch to set the motion of a character. Here in the solution of our problem, we take the "sticks from hands to the puppet" metaphor as our method.

Being a kind of the articulated puppet, Chinese shadow puppet is used as an interaction metaphor for building logical narrative [2]. In addition, how to generate Chinese shadow play animation has also been considered. Yi-Bo [10] develops the technique to mode the image of the shadow puppet and renders an animation. Tsai-Yen [8] develops a motion planning technique. In his work, a pendulum model is used for the lower body motion, which is also integrated in PuppetAnimator. While their work focuses on the algorithms to give compliant motion for the shadow puppet, our system considers the problem to provide an easy user participation interface.

III. DESIGN FOR PUPPETANIMATOR

PuppetAnimator makes use of the shadow play performance workflow. To use it, first thing to do is to authorize a virtual shadow puppet character. Next, perform the shadow puppet's action in the "stage panel" of the interface. At last, replay or save the result into animation files. We use the articulated figure model to describe shadow puppet character. Each puppet consists of a set of hierarchical rigid segments such as head, body, arms and legs as well as joints to link them together. To reproduce the feeling of manipulating real shadow puppets, we analyze shadow puppet's common animation actions, decompose them according to the difference of segments and design a set of motion modes to represent them. Based on those motion modes, puppet templates can be designed including the definition of segments linking relationship and the motion mode each segment follows. User can further add images for the skin of the segments and map point device input channels to the points of force application on segments.

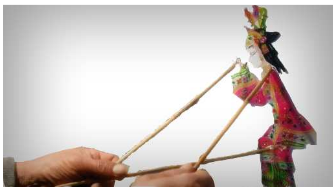


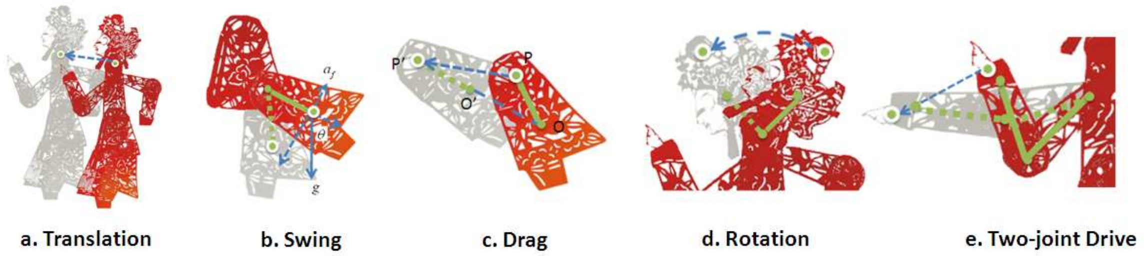
Figure 2. A puppeteer uses sticks to manipulate a puppet [12].

A. Shadow Puppet Motion Modes

In traditional Chinese shadow play, a puppet is animated with several sticks whose ends are held by a puppeteer (Figure 2). The ends of the sticks link to the points of force application on the puppet's head, body, arms or legs. By operating the sticks cooperatively in hand, the puppeteer can perform various puppet actions. Usually three sticks are held, one on the body for the translation of the whole puppet and two on the hands for the movement of arms. Sometimes another stick may be added for the rotation of the head. The segments linking directly to sticks will drive others under the laws of physic. As the puppet is held upright, gravity also affects its movement. For example, the resultant force of gravity and dragging by the body will cause the two legs to swing like a pendulum in use for representing walking action in the drama.

We design a basic set of element motion modes that a simulation system should support to achieve common puppet's actions. Each motion mode can drive one or two segments to move in respond to an external force created by point input or neighbor segments. The motion modes are *Translation*, *Swing*, *Drag*, *Rotation* and *Two-joint Drive* (Figure 3). By configuring those motion modes to certain segments of the puppet and using contemporarily, a user can arrange various actions through simple multipoint input.

1) *Translation:* To change a puppet's position on the stage is a basic operation in shadow play. Usually there is a stick dragging the body which then drives other segments to translate iteratively while keeping their relative position and orientation constant. In the system, we design Translation as a motion mode to enable a segment to move without rotation following its parent segment or input point. (Figure 3a).



a. Translation b. Swing c. Drag d. Rotation e. Two-joint Drive

Figure 3. Motion Modes. External force (blue line) acts on the segment through the point of force application (green dot with a white circle).

2) Swing: Gravity and friction creates single pendulum motion to a segment (Figure 3b). In PuppetAnimator, Swing is designed to produce gravitational effect on the shadow puppet. We use the simple pendulum model to describe its motion which makes the segment looks like suspending on its parent. This mode can be installed on the lower boy of the puppet for the actions such as jumping when the legs are off the ground or walking with two legs swaying back and forth.

3) Drag: If moment is taken into account when a segment is pulled, not only its position but also its orientation will be changed. Drag is designed to achieve this effect (Figure 3c). When driven by it, a segment will move at a minimum while the distance between the center and the point of force application remains constant. We adopt the RNT algorithm [6] to model this mode. It can play a role in calculating the instant position and orientation change when a segment with mass is pulled.

4) Rotation: For a segment, if it is attaching to a fixed point when an external force works on it, it will rotate around the fix point. We design Rotation to enable this effect (Figure 3d). It causes the segment to rotate for the angle that best matches the external force. This can be applied to the head of a shadow puppet for the nod action.

5) Two-joint Drive: The arms are the most flexible segments that a puppeteer will take a lot of effort to control. In the performance, there is a stick linking to the each hand and driving the lower and upper arms to move cooperatively. To support the users of our system to achieve flexible arm actions, we design Two-joint Drive (Figure 3e). When a force pulls from the end, the position and orientation of two linked segments will be recalculated according to the mode. As a problem of Inverse Kinematics, Cyclic Coordinate Descent (CCD) algorithm [5] is chosen to implement it.

Based on the set of the motion modes described above, we can assemble them into a puppet template. First, design a skeletal structure. Second, choose several motion modes and assign them to some segments describing how to respond to an external force. We have provided several puppet templates of carton humans and animals. The skin image of the segments and point input devices are left for the user to customize in XML files.

B. Multipoint Input

We refer to the stick manipulation in real shadow play for the interaction in our system. Point input devices such as mouse and touch pad are used instead of the sticks to control the virtual shadow puppet. Each channel of the input point is mapped to a point of force application located in a segment (Figure 1). When a user moves the mouse or his fingers on the touch pad, messages of the movement will be sent to the PuppetAnimator. Then the message will be mapped as an external force to pull the point of force application over a distance. The motion mode will calculate the new position and orientation for the segments involved. Besides, the puppet can be turned around by clicking right mouse button.

Take the puppet template shown in Figure 1 as an example, we use the Translation mode to act on the body and bind the input from a mouse to it. Meanwhile, two touch points from a touchpad are in charge of creating external forces onto two arms which are driven by Two-joint Drive. Besides, two legs are under the gravity action. Other segments are also linked and driven by Translation or Drag. To perform with this shadow puppet, a user can drag the mouse to make it move with two legs swaying. He can also slide his two fingers on the touchpad controlling two arms to achieve various gestures.

C. User Interface

The main window of our system (Figure 1) has three panels. The upper left one is the stage panel, the performance area where puppets are played. Its background can also be customized. The right one is a list of stage configuration information. The control panel is located at the bottom which includes time line and buttons to control the process of the performance such as play, record, stop, save as AVI and save as GIF.

To make a little animation, first press the record button which starts the recording process, then perform the puppet's actions and press the stop button to end the recording. After that a user can replay what he performed just now or save it as an AVI or GIF format file.

IV. Results and Discussion

The prototype of PuppetAnimator is implemented in Windows using Java. The motion modes are implemented as a separate engine library. The interface is built on Java Swing and displays puppets driven by the motion mode engine. Besides, we use source codes from two open source projects, AVIOuputStream (http://blog.hslu.ch/rawcoder/2008/08/12/writing-avi-videos-using-pure-java) and GifDecoder (http://www. fmsware.com/stuff/gif.html) to encode AVI and GIF format file. We make use of famous cartoon characters as examples and design six puppet templates as well as five actions: falling from the sky, talking gesture, dancing with two arms, climbing and mouse stealing the oil. Each puppet template can be loaded to the system as a plug-in. The system can receive point input messages through UDP in XML format. Two types of senders, mouse and touch pad, are supported.

A pilot study is performed to obtain an initial feedback from the users. 8 graduate students in our lab were invited to join the user study. They were novices at performing shadow puppet. First they had five minutes to get familiar with PuppetAnimator. Then they were asked to make five action animations following the plot of story. Each action involves two or three kinds of motion modes. For each task, they could try as many as 5 times and select the best one they liked to submit. They were also asked to fill out a questionnaire when finished. The questionnaire included questions about the motion modes, interface and manipulation method. The user study was performed on a desktop computer with a mouse and a touchpad to generate multipoint input. The type of the touchpad was Wacom Bamboo Fun CTH-661, which could detect two touch points. To access touch points' position from the touchpad and send message to PuppetAnimator, a stand alone program, derived from the source code of Bamboo-TUIO (http://code.google.com /p/bamboo-tuio/), was written.

We receive positive feedback on the ability enabled by PuppetAnimator to experience shadow puppet manipulation and little animation production. The participants were exciting that they could control the puppet directly. Observations showed that they got started quickly and were able to utilize the motion modes to arrange puppet's actions. The set of motion modes can fulfill these actions' requirement. We also found some user chose to use fingers from two hands, one from each, to manipulate the puppet's two arms. When an action required three input points with two fingers from one hand touching on the touchpad and the other hand holding the mouse, users would spend more time getting familiar with the operation and suggest that using another touch device instead of mouse would make it easier. Questionnaire showed that users agree on applying only a few fingers to control and employing common point device makes the interface more accessible. They also made some suggestions to the features of the system. At the beginning, some confused the mapping of input points and points of force application, so they suggest that adding visualization feedback for points could give them stronger feeling of physical manipulation, which we modified in the updated version.

V. CONCLUTION

PuppetAnimator is an ongoing effort in building tools for non-expert users to experience a Chinese shadow play. We experimented with a prototype system to achieve several puppets' actions and the positive feedback gave us the confidence for PuppetAnimator's application. The real shadow puppet performance requires high level for the professional puppeteer who needs to take effort to practice their skills. Though our work has satisfied users for their basic requirement to manipulate a shadow puppet in their computer, the issue to enable a novice user to reach the expert level comprehensively and easily with the help of interaction tools still remains open. Future work may include extending the motion modes set to support more complicated actions, which aims to reach more natural, efficient interaction effects. A formal study including multi-user cooperation is also considered to draw a stronger conclusion about the effectiveness of our approach.

ACKNOWLEDGMENT

This study is supported by National High-Tech Research and Development Plan of China under Grant No.2009AA01Z336.

REFERENCES

[1] A. D. Wilson, S. Izadi, O. Hilliges, A. Garcia-Mendoza, and D. Kirk, "Bringing physics to the surface," Proc. of the 21st annual ACM symposium on User interface software and technology (UIST'08), pp. 67-76.

[2] A. Paiva, M. Fernandes, and A. Brisson, "Children as affective designers-i-shadows development process," Humaine WP9 Workshop on Innovative Approaches for Evaluating Affective Systems(2006)

[3] C. Barnes, D. E. Jacobs, J. Sanders, D. B. Goldman, S. Rusinkiewicz, A. Finkelstein, and M. Agrawala, "Video puppetry: a performative interface for cutout animation", ACM SIGGRAPH Asia 2008, No. 124.

[4] L. A. Liikkanen, G. Jacucci, E. Huvio, T. Laitinen, and E. Andre, "Exploring emotions and multimodality in digitally augmented puppeteering", Proc. of the working conference on Advanced visual interfaces (AVI'08), pp. 339-342.

[5] L. T. Wang and C. C. Chen, "A Combined Optimization Method for Solving the Inverse Kinematics Problem of Mechanical Manipulators," IEEE Transactions of Robotics and Automation 7. 4 Aug. 1991, pp. 489-499.

[6] M. Hancock, F. Vernier, D. Wigdor, S. Carpendale, and C. Shen, "Rotation and Translation Mechanisms for Tabletop Interaction," First IEEE International Workshop on Horizontal Interactive Human-Computer Systems (TABLETOP '06), pp. 79-88.

[7] M. Thorne, D. Burke, and M. Panne, "Motion doodles: an interface for sketching character motion", ACM SIGGRAPH 2004, pp. 424-431.

[8] T. Li and S. Hsu, "An authoring tool for generating shadow play animations with motion planning techniques," International Journal of Innovative Computing, Information and Control, Vol.3, No.6(B), Dec.2007, pp. 1603-1614.

[9] S. Oore, D. Terzopoulos, and G. Hinton, "A desktop input device and interface for interactive 3D character animation," Proc. Graphics Interface, pp. 133-140.

[10] Y. Zhu, C. Li, I. F. Shen, K. Ma, and A. Stompel, "A new form of traditional art: visual simulation of Chinese shadow play," International Conference on Computer Graphics and Interactive Techniques (ACM SIGGRAPH 2003 Sketches & Applications), pp. 1-1.

[11] Flash Bone Tool, http://www.adobe.com/devnet/flash/articles/concept_bone_tool.html. Last access date December 14, 2010.

[12] wangronghua888.51.com