Motion Generation for Glove Puppet Show with Procedural Animation

Chih-Chung Lin[1], Gee-Chin Hou[2], and Tsai-Yen Li[1]

[1] Department of Computer Science, National Chengchi University, Taiwan
{s9421,li}@cs.nccu.edu.tw
[2] Department of TV and Radio, National Chengchi University, Taiwan
phou@nccu.edu.tw

Abstract. Traditional Taiwanese glove puppet show is a unique form of performing art. In this work, we aim to use the aid of computer animation technologies to preserve this cultural heritage and create innovative ways of performance. By observing the demonstration of a puppet master, we analyze the characteristics of how a hand puppet is manipulated and design animation procedures mimicking the motion of a glove puppet. These procedures are implemented on a real-time animation platform for procedural animation. Through the system, we allow a user to perform glove puppet animation with high-level inputs. We hope that, with the help of this system, not only the art of manipulating a glove puppet can be systematically documented, but the entry barrier for learning it can also be greatly reduced.

Keywords: Character Animation, Digital Content, Glove Puppet Show, Procedural Animation, Traditional Art.

1 Introduction

Taiwanese glove puppet show is one of the most popular folk arts in Taiwan and is also an indispensable entertainment in the daily life of Taiwan society. It combines elements of different traditional drama, music, delicate costumes and props. Being an exquisite art of action, the manipulation of glove puppet requires ability of space perception and hand-eye coordination with high level of precision. Because of the nature of these psychomotor skills, repeated practice is a must in order to achieve a mastery level of performance. Young puppeteers are usually trained by oral instructions. With no written-down materials, a senior puppeteer let apprentices know a simplified storyline and have them observe closely when he/she performs. It usually took apprentices more than three years to be competent as the assistant of principal puppeteer. In addition, this kind of show relies heavily on teamwork. The puppets are usually manipulated by one or two puppeteers in a show. The principal puppeteer usually manipulates all puppets and performs the narratives, dialogs and singing all by himself/herself.

In recent years, the applications of computer animation are becoming prevailing due to the fast development of hardware and software for computer graphics. These applications can be found in entertainment, education, and commerce. In addition to

F. Huang and R.-C. Wang (Eds.): ArtsIT 2009, LNICST 30, pp. 247–254, 2010.

commercial films and TV, computer animation also serves as an effective way to preserve traditional performing arts. For example, Li and Hsu [1] have designed an authoring system that can create the animations of traditional shadow play with high-level user inputs. The goal of designing this type of animation system is not only to capture the art in a digital form but also to lower the entry barrier for learning and producing such a show. Furthermore, we hope that innovative forms of performing art can be invented to take advantage of digital technologies.

Puppet animation is a special type of character animation. Currently, the main-stream methods for generating character animations can be divided into three categories. The first type of method generates character animation by sampled data. For example, as low-level signals, sampled data can be obtained with the motion capture technologies when the motions are performed by a real actor [4]. Animations made with this technology are more plausible. However, without knowing the meaning and structure of the motions, it is also difficult to modify a captured motion to fit the constraints of a new environment. The second way is about using commercial 3D animation software to set the key frames of a character by animation professionals and interpolate between key frames to generate the final animations. However, even for good animators, producing an animation in this way is usually labor intensive and time consuming. The third way is about generating animations by simulation or planning procedures [1][8]. It is also called knowledge-based animation since it usually requires a dynamics model of the objects under simulation or motion-specific knowledge to plan and generate the motions. Due to the complexity of the underlying computational model, this type of animations usually is generated in an off-line manner and displayed in real time. Nevertheless, due to the increasing computing power of computers, it is becoming more feasible to generate this kind of animations in real time with appropriate model simplification.

In this paper, we use the approach of procedural animation to generate the animation of traditional glove puppet show. Procedural animation belongs to the third type of methods for animation generation since it uses the knowledge of glove puppet manipulation to design animation procedures. A motion usually is divided into multiple phases delimited by keyframes with distinct spatial constraints in each phase. Then the motion between keyframes is computed by appropriate interpolation functions capturing temporal or spatial constraints. Animations produced with this approach have the advantages of being flexible and computationally efficient for real-time environments. Nevertheless, it is a great challenge to design a procedure with appropriate parameters that can produce plausible motions.

2 Related Work

Most of previous work on traditional Taiwanese glove puppet show was conducted via the motion capture approach. For example, recently the X-Lab at National Chao-Tung University captured the motion of the hand playing a glove puppet with a data glove [9]. They also have implemented a system with a user interface allowing a user to create a puppet show by putting small pieces of motions together. In addition to displaying the motions with 3D graphics, they have also designed a puppet robot of the same size as the hand puppet that can mimic the captured motion. In our work, the design goal is different from capturing and reproducing the hand motion of a puppet show. Instead, we hope to realize the animation by modeling the way that a hand puppet is manipulated in a procedural manner with parameters. By adjusting the parameters, we can produce a good variety of flexible puppet animations.

Puppet animations usually consist of two types of motions: intended primary motions and passive secondary motions. For example, the motions for the body, head and hands are primary motions manipulated by the animator while the motions of the legs, clothes, and hair belong to the secondary motions that move passively according to the primary motion, the gravity, and the environment. Previous work divided the problem into layers of abstraction and treated the secondary motions at the lower layers. For example, in [2], the main actions of a character were acquired by the motion capture technique and then the associated secondary motions were computed. Perlin also proposed to add noise to the primary motion as part of the secondary motion to make the generated motion look more realistic [8].

Li and Hsu [5] also distinguished these two types of motions for producing 2D traditional shadow play animation semi-automatically. The authoring tool allows a user to specify few key postures of the animated character in a 2D environment and then uses a motion planning algorithm to compute the path of primary motions. Secondary motions are then computed according to the obstacle constraints and the gravity force.

3 Animation System Design

3.1 Animation Platform

In this work, we aim to design a 3D animation system customized for real-time display of glove puppet animation. The system is constructed based on an experimental testbed for procedural animation, call IMHAP (Intelligent Media Lab's Humanoid Animation Platform), that was proposed in our previous work [7]. The architecture of the system is shown in Fig. 1. The system was designed with the MVC (Model, View,

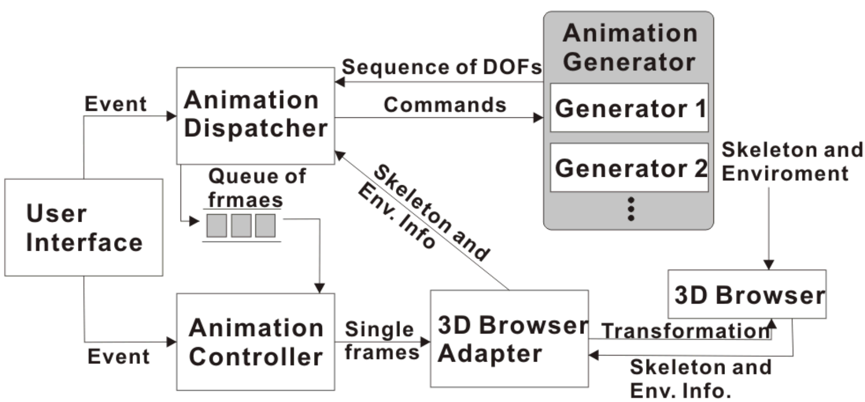

Fig. 1. System architecture of the puppet animation platform

and Controller) design pattern. Model is used for encapsulating data, view is used for interaction with the user, and controller is used for communication between model and view. In Fig. 1, every block represents a module, and the arrows between the modules stands for data flow. In these modules, animation dispatcher and animation generator are the kernel of this animation system defining how the animation generation problem is decomposed into elementary procedures implemented in the animation generators. In MVC, these two components play the role of *model* while the animation controller plays the role of *controller*. The last two components, 3D browser and user interface, play the role of *view* that enables interactive display. For the 3D display, we have adopted the open-source JMonkey package as our 3D browser [3].

3.2 Kinematics Model

The kinematics structure of a puppet is similar to a human figure with minor differences as shown in Fig. 2. For example, the major active joints of a puppet are at the neck, shoulder, thigh, and pelvis. The joints of elbows and knees are usually ignored. The motions of the legs are triggered either by another hand or by the gravity during swinging. The clothes are the passive part that moves with the legs as they swing. As shown in Fig. 2, we attach additional links and joints to the clothes at the front and back of the legs such that we can move the clothes when the legs are lifted.

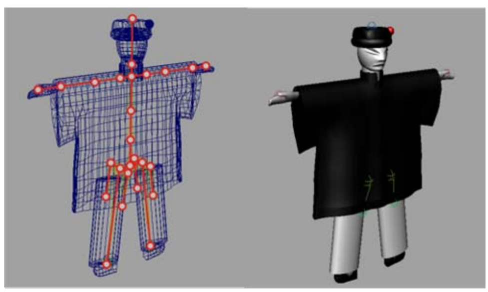

Fig. 2. Kinematics model of the hand puppet including the additional linkages for clothes

3.3 Motion Model

As mentioned in the previous subsection, in procedural animation, different motions require different animation generators. We divide the process of designing an animation procedure into three main steps. In the first step, we observe the motions performed by a master puppeteer and define appropriate motion parameters that can be used to describe the characteristics of the motion. For example, for the walk motion, the adjustable motion parameters include step length, body swing, and so on. In the second step, we decompose the motion into several phases by defining key frames separating these phases according to the motion parameters described above. In the last step, we define the procedure for interpolation between the key frames. Two types of interpolations are needed to produce plausible motions. One is to interpolate along a curve for trajectory following while the other defines how a motion is timed. In the current system, linear interpolation is used for simple motions while Bezier curves are used for more sophisticated motions.

To illustrate how the motions are generated in a procedural manner, we use three different types of motions as examples: *walk*, *run* and *roll-over,* as described below in more details.

3.3.1 Walk

There exist various ways of walking with different meanings under different scenarios. Advised by our puppet master, we have divided a normal walk into several phases. In the walk motion, the adjustable parameters include step length, step height and body swing angle. As shown in Fig. 3, we define four key frames according to the given motion parameters.

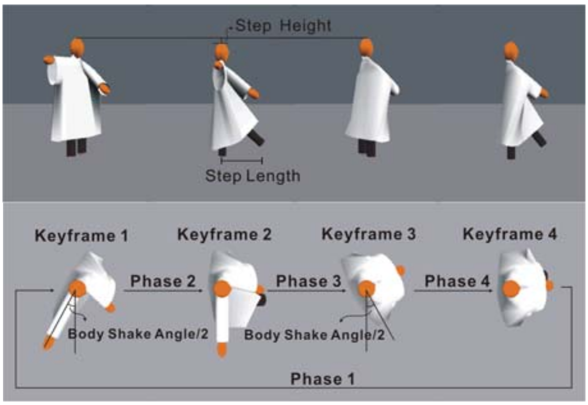

Fig. 3. The keyframes and the procedure parameters for the walk motion

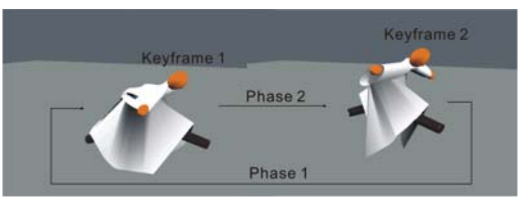

Fig. 4. The key frames of the run motion

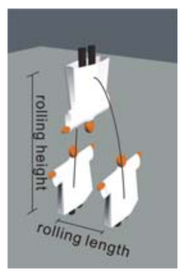

Fig. 5. A snapshot of the roll-over motion

3.3.2 Run

The key of creating this type of motions by a puppeteer is on a sudden swing of one leg upward and then moving the body up and down to swing both legs further to create the running motion as shown in Fig. 4. The swinging leg motions belong to the type of secondary motions since the legs move in accordance with the primary motion of the body. The speed and frequency of a body moving up and down determine the rotational speed of the legs swinging back and forth.

3.3.3 Roll-over

The roll-over motion is specified by two parameters: rolling height and rolling length. Two key frames are needed to define this motion. A Bezier curve is defined with two parameters and used as the trajectory of the puppet. The puppet also must make a complete revolution on the orientation along the trajectory. A snapshot of the generated animation for the roll-over motion is shown in Fig. 5.

4 Experimental Results

We have implemented a java-based animation system that allows a user to design the motions in a glove puppet show with an interactive graphical interface as shown in Fig. 6. At the upper-left corner, the system hosts a 3D browser that can display the animations generated by the user-designed procedures. The user can compile a sequence of motions (such as walk, run, and roll-over) by selecting them and setting their parameters from the graphical user interface on the right. The lowest part of the platform is the interface for play control that allows a user to view the animation continuously or frame-by-frame.

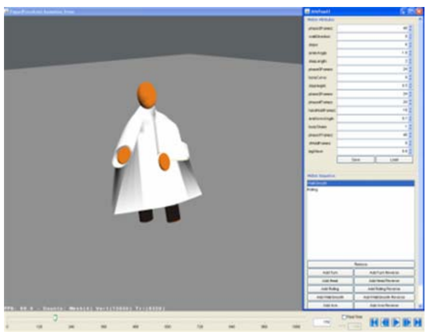

Fig. 6. Graphical user interface for interactive animation generation of puppet show

To illustrate that the motions generated by our animation procedures are rather flexible, we use different sets of parameters to generate animations. In Fig. 7(a), we show the walking motion generated with parameters of (swing_angle, step_length) = (0.3, 1.5) while the parameters are (1, 0.8) for Fig. 7(b). In Fig. 7(c), we show a running motion generated with the parameters of (step_length, step_height)=(0.5, 2) while the parameters are (1.5, 3) for Fig. 7(d). In Fig. 7(e), we show a rolling-over motion generated with the parameters of (rolling_height, rolling_length)=(10, 5) while the parameters are (20, 10) for Fig. 7(f). In Fig. 8, we show a composite example generated by compiling multiple motions into a final show. The puppet ran and then rolled over to make a complete revolution. The motions generated on the platform can be further exported into a MEL script for further rendering in animation packages such as Maya. An example of the motion rendered in Maya is shown in Fig. 9.

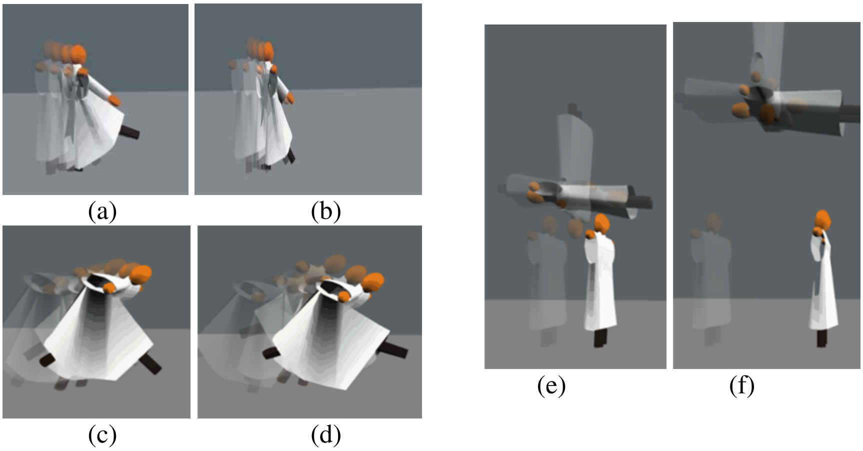

Fig. 7. Using different sets of motion parameters for walking (a)(b), running (c)(d), and rolling-over (e)(f)

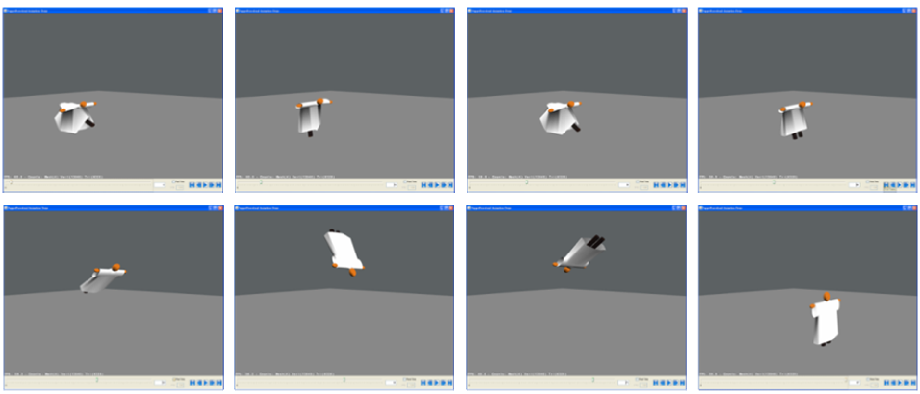

Fig. 8. An example of compiling different motions together. The puppet entered with a running motion and then did a roll-over motion at an end.

Fig. 9. An example of exported animation rendered in Maya

5 Conclusion and Future Work

With the help of computer animation techniques, we hope to preserve and promote the art of glove puppet show, a precious culture heritage in Taiwan. In this paper, we have made a first attempt to model the basic motions of a glove puppet by observing the performance of an expert puppeteer and implementing the motions with procedural animation. These animation procedures will work as reusable basic units for compiling a play in the future. Furthermore, we hope to design an animation scripting language that can be used to document the motions of glove puppets as well as the screenplay. We also hope that the music behind the script can be generated automatically according to the tempo of the show and synchronized with the cues from the puppeteer.

References

[1] Chen, P.F., Li, T.Y.: Generating Humanoid Lower-Body Motions with Real-time Planning. In: Proceedings of 2002 Computer Graphics Workshop, Taiwan (2002)

[2] Dontcheva, M., Yngve, G., Popović, Z.: Layered Acting For Character Animation. ACM Transactions on Graphics (2003)

[3] jMonkey Engine, `http://www.jmonkeyengine.com/`

[4] Kovar, L., Gleicher, M., Pighin, F.: Motion Graphs. In: Proceedings of ACM SIGGRAPH (2002)

[5] Li, T.Y., Hsu, S.W.: An Authoring Tool for Generating Shadow Play Animations with Motion Planning Techniques. International Journal of Innovative Computing, Information, and Control 36(B), 1601–1612 (2007)

[6] Li, T.Y., Liao, M.Y., Tao, P.C.: IMNET: An Experimental Testbed for Extensible Multi-user Virtual Environment Systems. In: Gervasi, O., Gavrilova, M.L., Kumar, V., Laganá, A., Lee, H.P., Mun, Y., Taniar, D., Tan, C.J.K. (eds.) ICCSA 2005. LNCS, vol. 3480, pp. 957–966. Springer, Heidelberg (2005)

[7] Liang, C.H., Tao, P.C., Li, T.Y.: IMHAP – An Experimental Platform for Humanoid Procedural Animation. In: Proceedings of the third International Conference on Intelligent Information Hiding and Multimedia Signal Processing, Tainan, Taiwan (2007)

[8] Perlin, K.: Real Time Responsive Animation with Personality. IEEE Transactions on Visualization and Computer Graphics (1995)

[9] X-lab of National Chao-Tung University, `http://xlab.cn.nctu.edu.tw/modules/x_movie/x_movie_view.php?cid=1&lid=35`