

AN AUTHORIZING TOOL FOR GENERATING SHADOW PLAY ANIMATIONS WITH MOTION PLANNING TECHNIQUES

TSAI-YEN LI AND SHU-WEI HSU

Computer Science Department
National Chengchi University
Taipei, Taiwan 11605
li@nccu.edu.tw; g9204@cs.nccu.edu.tw

Received October 2006; revised March 2007

ABSTRACT. *Shadow play is a Chinese art that has not yet transformed into a digital form. In this work, we attempt to develop an authoring tool to generate shadow play animations according to user's high-level inputs by utilizing motion planning algorithms. The motion of a character in a shadow play show usually consists of intended primary motion and reactive secondary motion. We propose to use the RRT-Connect algorithm to generate primary motions and design two methods to facilitate the generation of secondary motions. For the character's upper body, we try to bias the search of the planner to prefer some motion pattern. For the lower body of the character, we modify its motion path by a simple pendulum model and a collision-avoidance correction mechanism in a post-processing step. Through several illustrative examples, we have shown that we are able to generate realistic compliant motions for a character in a shadow play animation.*
Keywords: Animations, Motion planning, Rapid-exploring random tree (RRT), Chinese shadow play, Secondary motion

1. **Introduction.** Chinese shadow play, as depicted in Figure 1(a), is one of the most ancient genre of drama in China. It is a dramatic form which uses lanterns to project shadows of silhouettes made of hard paper and hide onto a white screen. The artistic effect of the play is produced through light, screen, music, signing and puppet operation. A performer manipulates the characters behind the screen while singing the libretto to tell the story. Motions of a character are realized by transforming various parts of a character via a few sticks on the hands of the animators as shown in Figure 1(b). Usually a good play is the result of tireless practice of an experienced animator. In addition to moving characters, a typical scene in a shadow play also consists of environmental objects such as tables, chairs, and bridges. Since the projection screen is flat, these objects may intersect with the characters from an audience's point of view. For example, a character may move across a table or a tree by assuming that they are in different layers. However, there are also many cases requiring the characters to stay collision-free with environmental objects in order to stand on a ground, sit on a chair, walk on a bridge, etc.

Despite its beauty, this traditional art has been sinking in recent years due to the wear-off of senior performers. One way to preserve and promote this valuable traditional art is by designing easy-to-use authoring tools in computers to attract potential performers and present the art in a digital form for easier dissemination. However, so far we have not seen many shadow plays being presented in a digital form partly due to the enormous time

and expertise involved in creating a play. Although some plays have been created with computer animation software [1], we have not seen any authoring tools that have been designed specifically for shadow plays, not to mention automatic tools that can generate the motions from the high-level inputs of an animator. Nevertheless, we found that the problem of generating a character's motion in a shadow play is very similar to the motion planning problem that has been well studied in Robotics [2]. Therefore, in this work we attempt to design an animation system that can generate the motions of a character in a shadow play automatically. The result is a motion planner tailored to the motion characteristics of the actors in a shadow play. The results can also be applied to solve the motion planning problems for robots or animated characters having unintended or unconscious motions in some degrees of freedom.



FIGURE 1. Chinese shadow play: (a) computer-generated image, (b) picture of puppets manipulating¹

Generally speaking, the motion of an actor in a shadow play can be classified into two categories: *primary motion* and *secondary motion*. The primary motion is usually performed to reflect the character's high-level intents. In accordance with the difficulty of accomplishing the intended task in a given environment, the amount of information that needs to be specified by the animator may vary greatly. For example, specifying the initial and goal configurations of a character may be good enough for moving the character across the space if no further detail motion specification is required. However, if the animator would like the character to perform martial art type of motions when moving toward the target, more appropriate key frames specifying the form of martial art must be given to create the desired motions. On the other hand, secondary motions are passive or compliant motions. For example, arms swing alternately and naturally when the actor walks to a destination. Furthermore, the uncontrolled parts of a character in a shadow play, such as feet, usually swing according to the gravity and comply with the ground silhouette [3].

Most motion planners that have been proposed in the literature today do not account for the primary and secondary motions at the same time. A typical solution may be

¹<http://music.kh.edu.tw/project/puppet/index01.htm>

breaking the motion into many key frames reflecting the secondary motion and then solving the sequence of smaller motion planning problems to create the desired effect. Although this approach is feasible, the task of specifying the key frames becomes too tedious and time-consuming.

In order to reduce the number of key frames that need to be specified by the user, we have adopted two mechanisms to generate the desired secondary motions in a shadow play automatically. First, for the upper body of a character, we have modified an existing motion planner such that the generated motion can follow certain motion patterns, such as cyclic motions, for some degrees of freedom whenever possible. For example, this mechanism can be applied to the arms of the character such that its arms can swing naturally while it is walking. Nevertheless, what such a mechanism imposes is preference instead of requirement. If the motion pattern will cause collisions with the environment, the desire will be ignored until it becomes feasible again. Second, for the lower body of a character, we adopt a post-processing approach to modify the motion generated by a motion planner such that the legs can swing like a pendulum and comply with the silhouette of the environment as the character moves.

We organize the rest of the paper as follows. In the next section, we will review some work related to this research. In Section 3, we will describe the system architecture and then review the RRT-Connect algorithm, which is the base algorithm that we have adopted. In Section 4 and Section 5, we will propose the methods that we have used to generate two types of secondary motions for shadow play animations. In Section 6, we will present some experimental results and discuss the effectiveness of such an approach. Finally, we will conclude the paper in the last section.

2. Related Work. The methods for generating character animations in computer graphics can be classified into three categories. One type of approach uses procedural methods to simulate and control the motions according to the characteristic of some desired motion patterns. For example, in [4], Hodgins et al. use a dynamic model to simulate a human character's motions in running, biking, and jumping. In [5], the authors design a neural controller to control the motion of two links, such as human arms, in performing rhythmic swing motion. The advantage of this type of approaches is that it possesses better control over the motion of the character while the main drawback is that the realism of the resulting motions depends highly on the accuracy of the model as well as the motion controller. The second category of approach uses sampled data as a basis to generate derived animations. For example, one can retarget a captured motion into another character of a different size and warp the motion to satisfy certain environmental constraints. The strength of this approach is on the realism of the generated motions. However, the reusability is low for different types of motions, and the motions for non-human characters are also difficult to capture. The third type of approaches relies on animators to specify key frames and generate the final motion by interpolations between key frames. Key frame specification is a tedious task requiring the expertise of an experienced animator. In [1], the authors attempt to create a shadow play animation in computers but the motions in the animation are specified or generated with a key-framed based approach by their customized animation software package. However, the focus of this work is on how to present realistic light effects of a shadow play.

The objective of this work is to generate the motion of the characters in a shadow play automatically by motion planning techniques. The motion planning problem has been well studied in the past few decades. It was originally brought up to solve the problem of generating collision-free paths between the given initial and goal configurations for a mobile robot or a robotic manipulator in order to simplify the task of robot programming and path specification. In recent years, these techniques have been successfully applied to other domains such as computer animation to facilitate animation design. For example, in [6], a manipulation planning algorithm (for a special motion planning problem dealing with object manipulation) has been developed to generate the motions with intentions for the upper body of a human character. On the contrary, in this paper we attempt to generate unintended secondary motions in addition to the intended ones. In [7], the RRT algorithm is proposed to generate motions for the kinodynamics problem by accounting for both the kinematics and dynamic constraints in the search process. In this paper, we also have adopted a similar approach to bias the search in developing the RRT structure. In [8], a motion planner based on the Probabilistic Roadmap Method (PRM) [9] was adopted to generate motion for a human character with 22 DOF. In the process of constructing the roadmap, the sampling is biased to favor preferred gestures by generating them with higher probability.

3. System Overview.

3.1. Procedural character modeling and animation. The architecture of the shadow play animation system is depicted in Figure 2. Two types of data are required as inputs to the system. First, the user needs to provide a scene description including geometric description of the objects, called obstacles, in the environment and the kinematics parameters of the actors, called robots, in the play. Second, the user needs to specify necessary key frames along the trajectory of the desired motion. The motion planner uses these two types of data to generate the motions of the character in each phase between two key frames. By incorporating the motion pattern module into the planning process, we hope that the generated motion will adhere to the desired motion pattern whenever possible. This found motion will then be sent to the post-processing module to generate desired secondary motions such as free swinging and compliant motions.

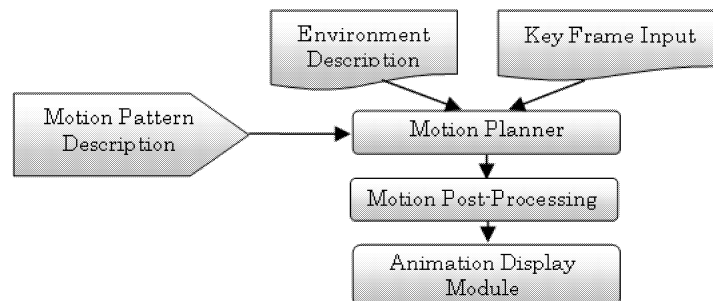


FIGURE 2. Architecture of the animation system

3.2. RRT-connect algorithm. The planning algorithm that we have used to generate the primary motion is the RRT-Connect algorithm [10], which is a form of probabilistic roadmap with the Rapidly-exploring Random Tree (RRT) structure. RRT [11] is a tree structure that has the feature of exploring unvisited freespace region quickly and evenly. The RRT-Connect algorithm is a single-shot planning algorithm that attempts to find a path by connecting two growing RRT's rooted at the initial and goal configurations. The idea of the algorithm is illustrated in Figure 3. T_a and T_b denote two RRT's rooted at the initial configuration, Q_{init} and the goal configuration, Q_{goal} , respectively. The algorithm starts by letting T_a explore outward with the EXTEND operation. This operation first samples a point (Q_{rand}) in the configuration space and then finds the nearest configuration (Q_{near}) to this sampled point. We extend Q_{near} to a new configuration Q_{new} by some distance ε toward Q_{rand} . Then we try to connect a nearest node of T_b to Q_{new} by a CONNECT operation, which is usually a straight-line connection. If the connection succeeds, then we have found a path connecting Q_{init} and Q_{goal} . Otherwise, we will switch the roles of T_a and T_b and let T_b perform the EXTEND operation and T_a perform the CONNECT operation. The process repeats until a path is found or a maximal number of trials are reached.

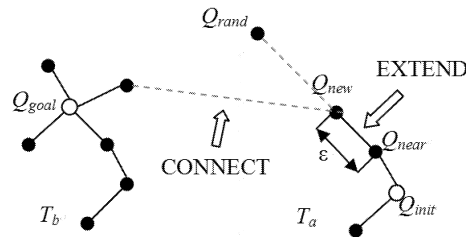


FIGURE 3. Illustration of the RRT-connect algorithm. T_a and T_b are two different RRT's trying to connect to each other with the EXTEND and CONNECT operations.

3.3. The motion planning problem for shadow play. In a shadow play, the character is the moving object that needs to avoid collisions with other objects in a scene. In this paper we use a human character consisting of ten joints with a total of 12 DOF, as depicted in Figure 4. The primary motion in this case refers to the motion of the body (pelvis) while the secondary motion refers to the other joints on the arms and legs. In the current implementation, we assume that only one character is in a scene to simplify the problem. In addition, we use a data file to describe the geometry of the objects in the scene. Each object has a layer attribute indicating the depth of the object. This attribute will be used to determine if the object should be treated as an obstacle in a specific scene.

4. Generating Cyclic Secondary Motion.

4.1. Generating primary motion with RRT-connect. We have adopted the RRT-Connect algorithm to generate the primary motion for the human character in a shadow play. The motion planner is given a sequence of key frames as inputs. For each motion

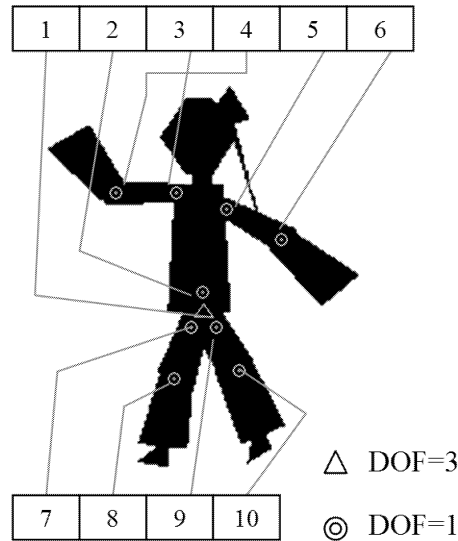


FIGURE 4. Kinematics of the character in a shadow play. The numbers in the squares are the ID's of the available joints.

phase between two key frames, we define a typical motion planning problem for a 12-DOF robot in a 2D workspace. The objects that are in the same layer as the robot are treated as obstacles. From a designer's point of view, the number of key frames that need to be supplied should be kept minimal since specifying these key frames is tedious and time consuming. However, if the number of key frames gets smaller, the designer will lose his/her control of the animation details, especially for secondary motions such as arm swinging during walking. In the following subsection, we will describe how we modify the original RRT-Connect algorithm to let some degrees of freedom favor a secondary motion pattern in the planning process whenever possible.

4.2. Generating cyclic secondary motions. As mentioned in the previous section, we would like to add preferences on motion patterns into the search process. We use a simple state machine to model a motion pattern. Taking the cyclic arm swinging motions as an example, we define two states (swinging forward and backward) and use two simple rules to trigger state transition. The first rule examines if the shoulder joints have reached the upper limits set for the current pace of the walking motion. If so, the state is switched to the other one. The second rule is exactly the opposite.

Once we have the motion pattern defined, we can attempt to incorporate the pattern into the search process of the RRT-Connect algorithm. First, we add a state attribute to each DOF for the configuration in a RRT node. Second, we modify the EXTEND operation to conform to the motion pattern. We use Figure 5 to illustrate the modification by taking a free-flying 3-DOF robot as an example. Assume that the X-Y plane is the workspace for the robot and the configuration space is a three-dimensional space (including θ) as shown in Figure 5. We would like the robot to have a motion pattern of swinging back and forth periodically on the θ dimension. Instead of extending Q_{near} linearly toward the randomly sampled configuration Q_{rand} for all dimensions, we will modify the value of the θ to follow the specified motion pattern. For example, in the example of

Figure 5, θ of Q_{new} is supposed to decrease with the original EXTEND operation. However, according to the state and the rule at that time, the θ value should be increased to respect the motion pattern. If this new configuration Q_{new}' is collision-free, it will replace Q_{new} as the extended node. Otherwise, Q_{new} will be used instead. By altering the value of certain dimensions to respect the motion pattern, the found path will conform to the motion pattern whenever possible. Note that the designer has much space in designing the states and rules for more complex secondary motions.

In our current implementation of the shadow play animation system, we have applied the periodic motion pattern on the joints of 3, 4, 5, and 6. Joints 3 and 5 control the upper arm while joints 4 and 6 control the lower arms. Since the moving directions for the upper arm and lower arm are usually consistent, we only need one state for each arm. Therefore, we have designed two states and two rules as described in the previous section to realize the secondary motion.

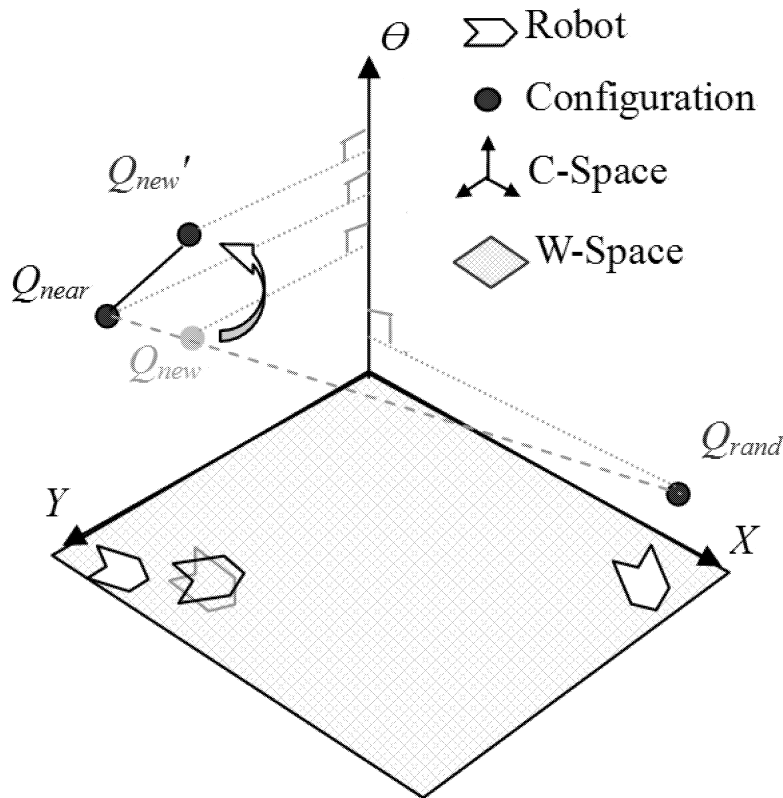


FIGURE 5. The illustration of the biased EXTEND operation in RRT-Connect

5. Generating Compliant Lower-Body Motions. In addition to the periodic secondary motion for the upper body, we also need to modify the path found in the planner to make it look more realistic as in the real shadow play. For example, some parts of the character, such as the legs, are not controlled directly by the animator. Instead, they move relative to the main body and comply with the objects that it touches. Therefore, one can see the legs swing like a pendulum when the main body moves. Although the

animator cannot directly control the legs, he/she can make it comply with the objects in the scene to create actions such as bending knees. In this section, we will describe how we modify the found path in the post-processing step.

5.1. Using a pendulum model to rebuild lower-body motions. Since the motion found in the motion planner for the lower body may not have the desired effect, we discard the motion for the lower body and generate a new compliant motion according to a simple dynamic model described below.

$$\Delta s = v * \Delta t + 1/2 * a(\Delta t)^2 \quad (1)$$

where v and a are the linear velocity and acceleration of p and Δs is its translational change for the given time period Δt . If we are given Δs and v , we can compute the required a that causes the motion. By taking the sum of the linear acceleration of a and the gravity g , one can compute the tangential component a_T of the resulting acceleration. From this component, we can compute the angular acceleration α for the pendulum with the following equation.

$$a_T = r\alpha \quad (2)$$

where r is the length of the pendulum. Then we can use α to update the joint angle θ . We apply the above procedure to compute joints 7, 8, 9, and 10 when the body moves.

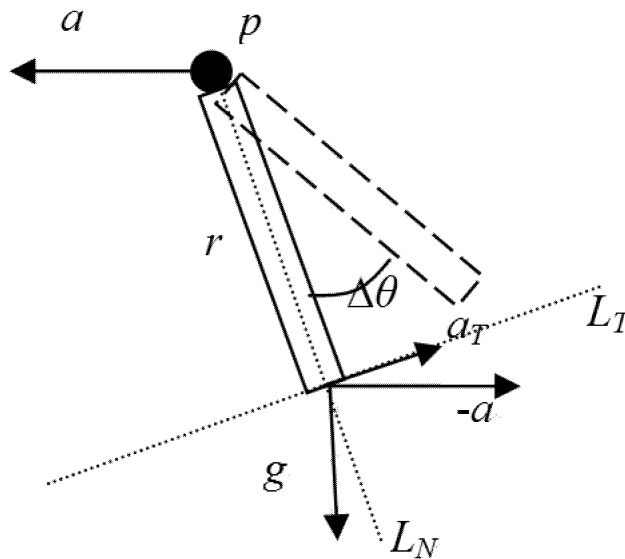


FIGURE 6. Using a pendulum model to compute the lower-body trajectory of the character

5.2. Adjusting the motions to avoid collisions. The lower-body motion generated with the above procedure may not be collision-free. Therefore, we need to adjust the motion to make the lower body remain legal and compliant to the ground and other objects in the scene at all time. In other words, we need to move the new configuration which is inside the C-obstacles to a configuration in the free space. Since we update the lower body incrementally with a pendulum model, when a collision situation occurs, the

nearest collision-free configuration should not be too far from the current one. Therefore, we propose to search the neighboring configurations of the current one as shown in Figure 7. Since the two legs do not intersect with each other, we can treat them independently to find collision-free configurations. Therefore, we only consider a two-dimensional subset (θ_1, θ_2) of the original configuration. Assume that the current (θ_1, θ_2) is inside the C-obstacle. We use a breadth-first search algorithm to look for a free configuration. When there are multiple free configurations for a given depth, the one that is the closest to the previous configuration will be chosen in order to avoid abrupt jump in motion.

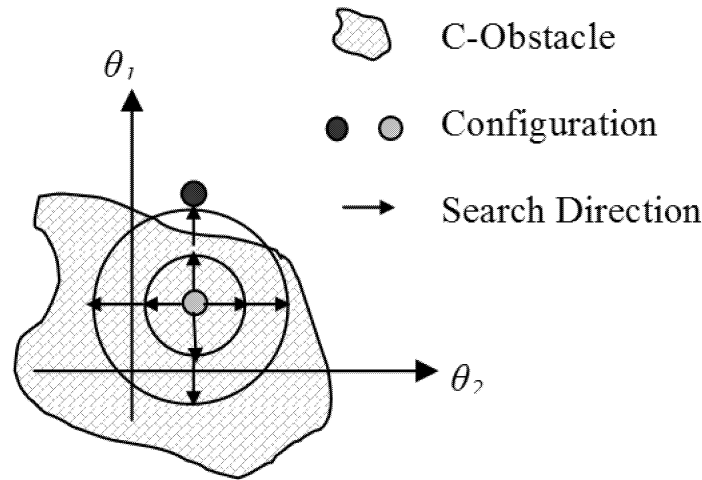


FIGURE 7. Exploring from inside of the C-obstacle to search for a free configuration

6. Experimental Results and Discussions.

6.1. Experimental results. In order to evaluate the effectiveness of the proposed approach for secondary motions, we have done experiments to compare the results of the generated motions with different methods. The example scenario is the one shown in Figure 9(a). In the first experiment, we use the original RRT-Connect algorithm and only specify two key frames at the beginning and the end of the animation. In the second experiment, we use the same planner but specify seven key frames, where the additional five key frames are used to define the secondary motions along the trajectory of the primary motion. In the third experiment, we use the new planner that can incorporate motion pattern and only give it two key frames as in the first case. We record the values of the 5th joint along the path and compare them in Figure 8. We can see that the motions produced in the second and the third experiments all have the desired effect of periodic motions but the motion generated in the first experiment does not possess this property as expected. Since the third experiment only requires the user to supply a minimal amount of key frames, the proposed planner is superior in the objective of generating secondary motions with minimal user intervention.

In addition to showing that the secondary motions can be generated more efficiently, we also have done experiments to demonstrate that the motion pattern is only a preference during the search process and can be sacrificed whenever necessary. For example, in

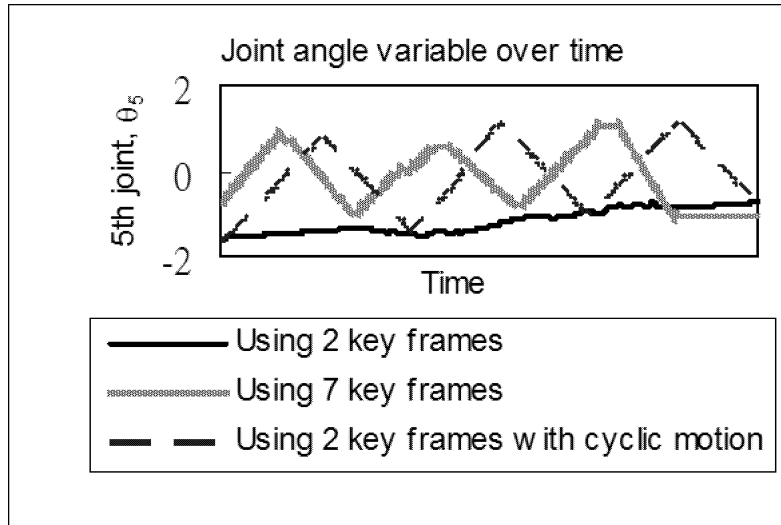


FIGURE 8. Comparison of three different ways to create the desired cyclic motion



FIGURE 9. Examples of secondary motions generated by the system: (a) free arm swinging while the body is moving, (b) arm-swinging is constrained inside the boiler, (c) lower body motion complying with the obstacles.

Figure 9(a), the scene consists of several objects but only the ground and the bridge are in the same layer as the character. Along the path of the primary motion, the arms of the character do not collide with any obstacles and therefore always follow the desired motion pattern. However, in the scenario of Figure 9(b), we assume that the character is initially inside the boiler and would like to escape from it. Due to the geometric constraints of the boiler, we can see that the character's arms can only swing to the degree that they do not cause collisions with the boiler. The arms start to swing widely after the character exits the boiler. In Figure 9(b) and 9(c), we can also see that the lower body of the character moves like pendulums and is able to comply with the obstacles that it encounters.

6.2. Discussions. Using states and rules allows us to incorporate motion patterns on certain degrees of freedom of a character. Although in the current example we only use two states and two rules to define the periodic motion pattern, more states and rules may be required to create more complex motions. Since no time dimension is involved in the current rule design, the state transition can only be triggered by joint values. This is a limitation of the current implementation that needs to be overcome in the future.

In addition, although we have ignored some portion of motions generated by the planner in the post-processing step, we think taking this portion of DOF into account in the planning phase is still necessary. The reason is that ignoring the lower body may cause the planner to find a primary motion that is lower in position and harder for the post-processing step to find a compliant motion for it. Since the post-processing step does not guarantee that it can always modify the path to a feasible one, the completeness of the planner may be put into jeopardy. Therefore, starting the path adjustment from a legal path is a better idea for both planning effectiveness and completeness. Nevertheless, it is possible to use a simpler robot model (due to symmetry on the lower body) in the planner for the primary motion as long as sacrificing the completeness is not likely to happen or is not the main concern of the given application.

During the design of our system, we have consulted several professional books and videos on how to perform shadow play shows through puppet manipulation [3][12]. However, this authoring tool at this current status only focuses on the aspect of motion generation of this performing art and, thus, is by no means complete. Nevertheless, the rhythm and style of a puppet character can be authored by mimicking a real performance with the assistance of our authoring interface. We have also invited a professional performer with hand-on experiences who also masters computers to evaluate the system. The general feedback is rather positive. The evaluator regarded that the animation generated by our tool has captured the motion characteristics of a puppet in a shadow play but the texture and appearance of our puppet can be further improved.

7. Conclusions. The beauty and unique features of a shadow play has motivated us to preserve this traditional Chinese art in a digital form. The difficulty of specifying the desired motions in a shadow play animation further motivate us to use motion planning techniques to generate the motions automatically. According to the characteristics of the character's motion in such an animation, we have modified the RRT-Connect algorithm to incorporate motion patterns into the search process for the upper body of the character. We have also adopted a post-processing procedure to generate secondary motions for the lower body that can comply with gravity and obstacles. We believe that this type of motion planning method can be used to simplify the planning problem for animated characters that have unintended or subconscious motion on certain degrees of freedom.

Acknowledgment. This work is partially supported by National Science Council under contact no. NSC 95-2221-E-004-015.

REFERENCES

- [1] Zhu, Y.-B., C.-J. Li, I. F. Shen, K.-L. Ma and A. Stompel, A new form of traditional art: Visual simulation of Chinese shadow play, *Proc. of the SIGGRAPH 2003*, 2003.
- [2] Latombe, J., *Robot Motion Planning*, Kluwer, Boston, MA, 1991.
- [3] Chin, C. A. and C. T. Hung, *The Chinese Shadow Show*, Wu-Nan, Taipei, Taiwan, 2001 (in Chinese).
- [4] Hodgins, J. K., W. L. Wooten, D. C. Brogan and J. F. O'Brien, Animating human athletics, *Proc. of the SIGGRAPH 1995*, pp.71-78, 1995.
- [5] Han, Q., Z. Qin, X. Yang and B. Wen, Rhythmic swing motions of a two-link robot with a neural controller, *International Journal of Innovative Computing, Information and Control*, vol.3, no.2, pp.335-342, 2007.
- [6] Koga, Y., K. Kondo, J. J. Kuffner and J. C. Latombe, Planning motions with intentions, *Proc. of the SIGGRAPH 1994*, pp.395-408, 1994.

- [7] LaValle, S. M. and J. J. Kuffner, Randomized kinodynamics planning, *Proc. of the IEEE Intl. Conf. on Robotics and Automation*, pp.378-400, 2001.
- [8] Kallmann, M., A. Aubel, T. Abaci and D. Thalmann, Planning collision-free reaching motions for interactive object manipulation and grasping, *Proc. of the Eurographics*, pp.313-322, 2003.
- [9] Kavraki, L., P. Svestka, J. Latombe and M. Overmars, Probabilistic roadmaps for fast path planning in high-dimensional configuration spaces, *IEEE Tran. on Robotics and Automation*, vol.12, pp.566-580, 1996.
- [10] Kuffner, J. J. and S. M. LaValle, RRT-connect: An efficient approach to single-query path planning, *Proc. of the IEEE Intl. Conf. on Robotics and Automation*, pp.995-1001, 2000.
- [11] LaValle, S. M., Rapidly-exploring random trees: A new tool for path planning, *Technical Report 98-11*, Computer Science Dept., Iowa State University, 1998.
- [12] Chiu, Y. F., *The Shadow Puppet Show in Taiwan*, Hilit, Taipei, Taiwan, 1998 (in Chinese).