Appear in Proceedings of the International Conference on Computer
Animation and Social Agents (CASA'05), 2005

Planning Character Motions for Shadow Play Animations

Shu-Wei Hsu
National Chengchi University
g9204@cs.nccu.edu.tw
http://www.cs.nccu.edu.tw/~g9204

Tsai-Yen Li
National Chengchi University
li@nccu.edu.tw
http://www3.nccu.edu.tw/~li

Abstract

In this paper, we propose a shadow play animation system that utilizes motion planning algorithms to generate Chinese shadow play animation automatically according to user's high-level inputs. 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 generate a path with our preferred motion pattern as much as possible. For the character's lower body, we modify its motion path by a simple pendulum model and some collision-avoidance correction mechanism in a post-processing step. With these methods, we are able to generate realistic compliant motions for a character in a shadow play animation.

Keywords: Motion Planning, Chinese Shadow Play, Secondary Motion

1 Introduction

Shadow play, as depicted in Figure 1, is a kind of drama in which silhouettes made of hard paper and hide are projected onto a white screen. 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. In addition to moving characters, a typical scene in a shadow play also consists of environmental objects such as tables, chairs, and bridges that exist in different layers and may intersect with the projection of the character. 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.

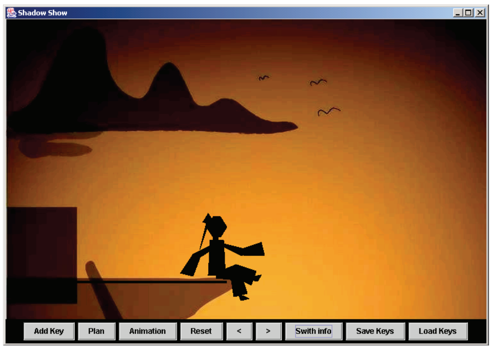

Figure 1: computer generated shadow play image

We have not seen many shadow plays being presented in a digital form. Although some plays have been created with computer animation software [7][8], we have not seen any automatic tools that can generate the motions from an animator's high-level inputs. Nevertheless, 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 [3]. 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.

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 usually is performed to reflect the character's high-level intents. According to the difficulty of a given task, 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 it across the space. However, if the animator wants the character to perform special types of motions (such as martial art), more key frames may be needed in order to create the desired motions. On the other hand, secondary motions are passive or compliant motions. For example, arms swing alternately when the actor walks. 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.

Most motion planners that have been proposed in the literature today do not account for the primary and secondary motions at the same time. Although secondary motions can be obtained by asking a user to specify enough intermediate key frames, the task is too tedious and time-consuming. We have adopted two mechanisms to facilitate the automatic generation of the desired secondary motions in a shadow play. First, for the upper body of a character, we have modified an existing motion planner to generate motions with some preferred patterns, such as cyclic motions, for some degrees of freedom whenever no collisions exist. For example, this mechanism can be applied to make the arms swing naturally while the character is walking. 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. In the next section, we will first review some researches pertaining to our work.

2 Related work

The methods for generating character animations in computer graphics can be roughly classified into three categories. One type of approaches uses procedural methods to simulate and control the motions according to the characteristic of desired motion patterns. For example, [1] uses a dynamic model to simulate a human character's motions in running, biking, and jumping. This type of approaches has better control over the character's motions but the realism of the resulting motions highly depends on the accuracy of the model as well as the motion controller. The second category of approaches uses captured motion data. For example, one can retarget a captured motion into another character with a different size and warp the motion to satisfy environmental constraints. This approach can generate realistic motions but the reusability is low for different types of motions. 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 even for an experienced animator. In [7], the authors attempt to create a shadow play animation with such a key-framed based approach by a commercial animation software package.

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 given initial and goal configurations for robots. In recent years, these techniques have been successfully applied to other domains such as computer animation to facilitate animation design. For example, in [9], a special motion planning algorithm 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 [6], the RRT algorithm is proposed to generate motions for the kinodynamics problem by accounting for both the kinematic 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 [5], a motion planner based on the Probabilistic Roadmap Method (PRM) [4] is 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 animation

In our shadow play animation system, two types of input data are required. First, the user needs to provide a scene description including geometric description of the objects, called *obstacles*, in the environment and the kinematic 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

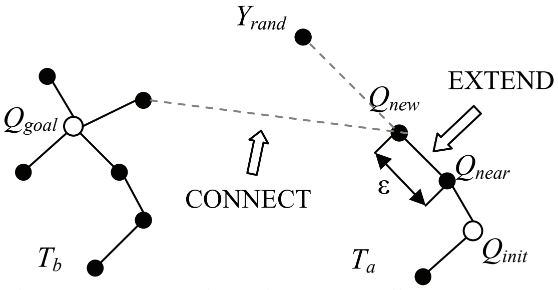

Figure 2: Illustration of the RRT-Connect algorithm

data to generate the motions of the character for each phase between two key frames. A motion pattern module is incorporated into the planning process such that the generated motion will adhere to the desired motion pattern whenever possible. The found motion is then be sent to a post-processing module to generate desired secondary motions such as free swinging and compliant motions.

3.2. *RRT-Connect algorithm*

The planning algorithm that we have used to generate the primary motion is the RRT-Connect algorithm [2], which is a form of probabilistic roadmap with the Rapidly-exploring Random Tree (RRT) structure. RRT 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 2. 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 have been reached.

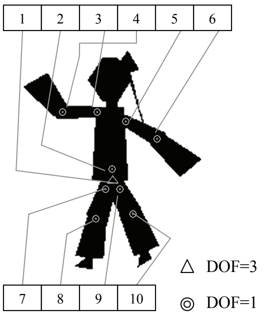

Figure 3: Kinematics and joint ID of the character in a shadow play

3.3. *The motion planning problem for shadow play animation*

The character in a shadow play is the robot of our motion planning problem. In this paper we use a human character consisting of ten joints with a total of 12 DOF, as depicted in Figure 3. The primary motion in this case refers to the motion of the body (pelvis) while the secondary motion refers to the other joints at 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 Planning Upper-Body Motions

4.1. *Generating primary motions 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 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. However, if the number of key frames gets smaller, the designer will lose his/her control of animation details, especially for secondary motions. In the following subsection, we will describe how we modify the original RRT-Connect algorithm to let some degrees of free-

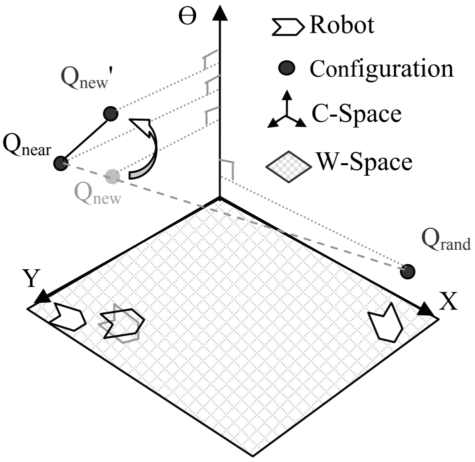

Figure 4: The illustration of the biased EXTEND operation in RRT-Connect

dom favor a motion pattern in the planning process whenever possible.

4.2. Generating cyclic secondary motions

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 for the current walk motion cycle. If so, the state is switched to the other one. The second rule is exactly the opposite.

We can then incorporate the defined 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 4 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 along the θ dimension. Instead of extending Q_{near} linearly toward the randomly sampled configuration Q_{rand} for all dimensions, we will modify θ to follow the specified motion pattern. For example, in Figure 4, θ of Q_{new} is supposed to decrease with the original EXTEND operation. However, according to the state and the rules at that time, θ 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 parameter to respect the motion pattern, the found path can conform to the pattern whenever possible.

In our current implementation of the system, we have applied the periodic motion pattern to joints 3, 4, 5, and 6. Joints 4 and 6 control the upper arm while joints 3 and 5 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. Note that the designer has much space in designing the states and rules for more complex secondary motions.

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, move freely relative to the main body and comply with the objects that it touches. 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.

Assume that the dot p in Figure 5 corresponds to the seventh joint of the character in Figure 3 and the long bar is the thigh. We first compute the linear acceleration of joint 7 and then use it to compute the joint angle. We assume that the position changes of p can be updated according to the following equation.

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

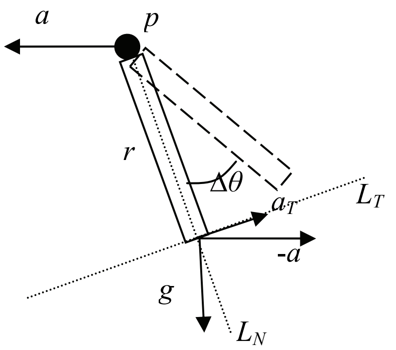

Figure 5: Using a pendulum model to compute the lower-body trajectory of the character

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.

$$a_T = r\alpha, \qquad (2)$$

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 the values of joints 7, 8, 9, and 10 when the body of the character moves.

5.2. Adjusting 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 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 the above pendulum model, when a collision occurs, the nearest collision-free configuration should not be too far from the current one. Therefore, we propose to search the neighboring configurations in a two-dimensional subspace (θ_1, θ_2) for each leg in a breadth-first fashion. 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.

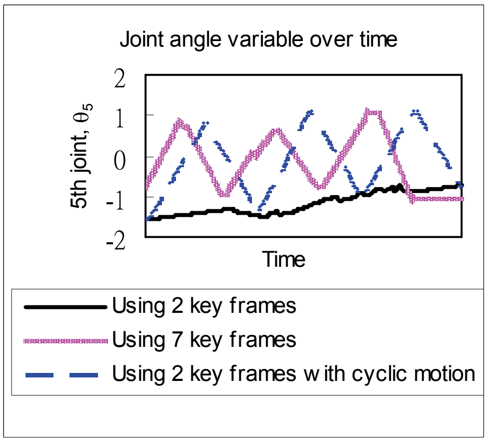

Figure 6: Comparison of three different ways to create the desired cyclic motion

6 Experiments

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 7(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 6. We can see that the motions produced in the second and the third experiments all have the desired effect of periodic motions while the motion for the first experiment does not 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.

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 7(a), the scene consists of several objects but only the ground and the bridge are in the same

(a) (b) (c)

Figure 7: Examples of secondary motions generated by the system

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 7(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 freely after the character exits the boiler. In Figure 7(b) and 7(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

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. 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.

7 Conclusions

The beauty and unique features of 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. We have modified the RRT-Connect algorithm to incorporate motion patterns into the search process for the upper body. We have also adopted a post-processing procedure to generate secondary motions for the lower body that can comply with the 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.

References

[1] J.K. Hodgins, W.L. Wooten, D.C. Brogan, and J.F. O'Brien. Animating Human Athletics. Proc. of SIGGRAPH, pages 71-78, August 1995

[2] J. J. Kuffner and S. M. LaValle. RRT-connect: An efficient approach to single-query path planning. Proc. of IEEE Intl. Conf. on Robotics and Automation, pages 995-1001, 2000

[3] J. Latombe. Robot Motion Planning, Kluwer, Boston, MA, 1991

[4] L. Kavraki, P. Svestka, J. Latombe, and M. Overmars. Probabilistic Roadmaps for Fast Path Planning in High-Dimensional Configuration Spaces. IEEE Tran. on Robotics and Automation, 12:566-580, 1996

[5] M. Kallmann, A. Aubel, T. Abaci, and D. Thalmann. Planning Collision-Free Reaching Motions for Interactive Object Manipulation and Grasping. Proc. of Eurographics, pages 313-322, 2003

[6] S.M. LaValle and J.J. Kuffner. Randomized Kinodynamics Planning. Proc. of IEEE Intl. Conf. on Robotics and Automation, pages 378-400, 2001

[7] Ugur Güdükbay, Fatih Erol and Nezih Erdogan, Beyond Tradition and Modernity: Digital Shadow Theater, Leonardo, 33:4, pages 264-265, 2000.

[8] Y.-B. Zhu, C.-J. Li, I. F. Shen, K.-L. Ma, A. Stompel. A new form of traditional art: visual simulation of Chinese shadow play. Proc. of the SIGGRAPH 2003 conference on Sketches & applications, 2003

[9] Y. Koga, K. Kondo, J.J. Kuffner, and J.C. Latombe. Planning motions with intentions. Proc. of SIGGRAPH '94, pages 395-408, 1994.