

COORDINATED MOTION PLANNING FOR 3D ANIMATION – USING CHINESE LION DANCE AS AN EXAMPLE

Fu-Sheng Yu and Tsai-Yen Li

Computer Science Department
National Chengchi University
64, Sec. 2, Zhi-Nan Rd, Taipei, Taiwan 116
E-mail: {g9331,li}@cs.nccu.edu.tw

ABSTRACT

Creating complex animations of human characters with simple high-level commands has been a design goal of animation software for a long time. Procedural animation is an effective approach in this direction aiming to creating animation with algorithmic description of how a motion is generated. Nevertheless, creating coordinated motion between animated characters has always been a great challenge because of the computational complexity of the problem. In this paper, we will use Chinese lion dance as an example to develop animation procedures that can generate coordinated motions for the two performing dancers moving on tops of piles arranged in various ways. The procedures use motion planning techniques to determine footstep sequences for achieving a goal configuration and adopt procedural animation methods to generate the articulated motion for realizing the footstep movement. Simulation examples are shown in this paper to demonstrate the effectiveness of the implemented animation system in generating realistic lion dances for a given pile arrangement.

1. INTRODUCTION

The advent of powerful desktop computers has enabled the rapid development of animation software and fast adoption of computer animation in various application domains such as films and games. Generally speaking, the ways of creating a computer animation can be classified into three categories: *motion capture*, *key-framing*, and *procedural*. The methods in these three categories are somewhat complementary and each of them may be more suitable for certain kinds of animations. Most computer animations today were produced with the methods from one or more of these approaches according to the budget of the production, characteristics of a motion, and other requirements of an animation.

The production cost of a high-quality computer animation typically is very high because the tools of current animation software still require a user to perform tedious and time-consuming motion specification in order to produce a complex animation. For example, due to the complexity of the motions involved in performing a Chinese lion dance, it is difficult to produce a realistic Chinese lion dance with the current animation software. Designing animation tools to facilitate the creation of this type of animations with



Figure 1. A snapshot of Chinese lion dancing on quincuncial piles

synchronized, coordinated motions has always been a challenging problem.

The objective of this research is to design a motion planner that can facilitate the generation of animation for the performance of a Chinese lion dancing on tops of piles of various heights, as shown in Figure 1. It takes long-time practices for the lion dancers to perform seamless motion coordination in real life. Generating this kind of performance in animation presents an even more challenging problem if without the helps of a motion planner. Given a geometric description of the environment (piles) and the initial and goal configurations of the lion, the motion planner should be able to generate a feasible path and its associated body motions that can take the lion dancers to reach the goal. The coordinated motions are constrained by the costume connecting the two dancers. The planner needs to consider how to place their footsteps on the piles and how to move their bodies without violating the distance constraint between the two dancers.

Our system is based on the experimental platform, called RhyCAP (Rhythmic Character Animation Playacting), proposed by Chen and Li [2] for Chinese lion dance animation. The main advantage of the system is that it allows the users to use high-level commands to generate lion dance animations on a flat floor procedurally. However, in this work, the footsteps of the lion dancers are determined by a motion planner instead of controlled interactively by the user. In addition, we have improved the motion controllers to allow the animated characters to move on an

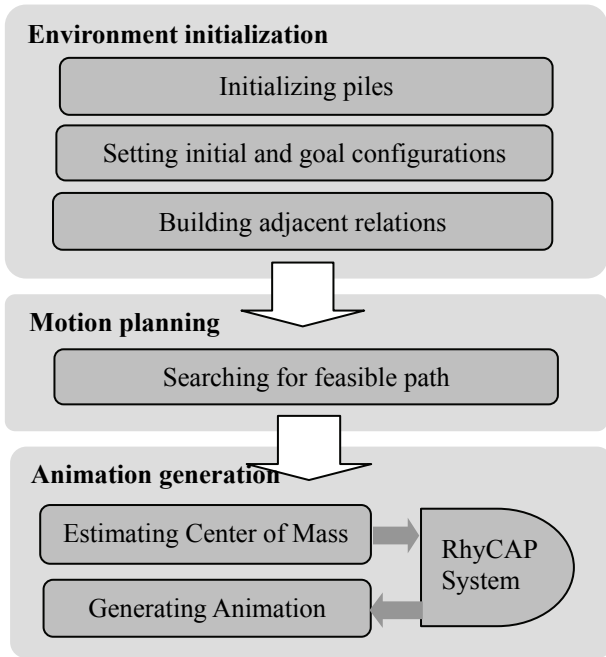


Figure 2. System architecture

uneven terrain such as tops of piles.

In the rest of the paper, we will organize the presentation as follows. In the next section, we will review the work pertaining to our research. In Section 3, we will overview our system architecture and describe how to formulate our problem into a motion planning problem. We will then present the algorithm of our motion planner in Section 4. In Section 5, we will describe our implementation and present some of our experimental results generated by the planner.

2. RELATED WORK

The problem of motion planning has been studied for more than three decades. A good general review of various approaches can be found in Latombe’s book [9]. The application of motion planning techniques in computer animation is more recent. Much work has been shown to be effective in facilitating the automatic generation of complex animations for human characters [6].

Compared to the basic motion planning problem, less work has focused on solving the problem of coordinated motion planning for multiple robots. For example, Desai [3] has proposed a planning system to generate the motion for two mobile robots manipulating and transporting a workload cooperatively in a dynamic environment. A computation model has been used to solve the dynamic motion planning problem for a system of cooperating robots in the presence of geometric and kinematic constraints. Estevens et al. [4] proposed a decoupled approach to solve the geometric and dynamic aspects of the motion planning problem sequentially for two animated virtual humans manipulating an object cooperatively. In these two works, the authors have focused on planning the coordinated manipulation for the upper body only.

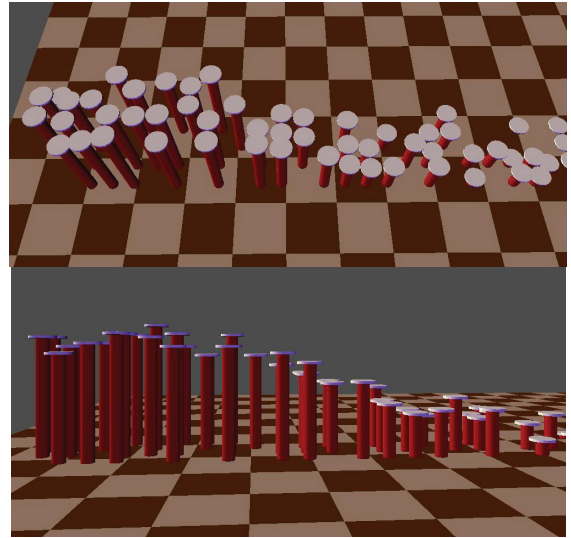


Figure 3. Side view and top view of an example scene with piles of various heights

Motion planning and simulation for the locomotion of a virtual human was mostly studied in the domain of computer animation. Bruderlin [1] has developed a goal-oriented system to simulate and control the footstep motions of a humanoid virtual character. Girard used a mix of kinematic and dynamic methods to simulate human locomotion. The motion of the lower body was defined in a procedural manner while the motion of the whole body was computed with simplified dynamics [5]. Van de Panne [11] created a whole-body animation by formulating it as a global optimization problem on the center-of-mass trajectory based on the placement and timing of footprints. Kuffner proposed a real-time planning system to find collision-free motion for a humanoid character in an interactive virtual environment [7]. In [8], the authors have also introduced the footstep motion planning system for biped robots that makes use of heuristics to facilitate the search for footsteps that can satisfy obstacle and dynamic balance constraints.

In [2], Chen and Li have introduced a hierarchical animation control system, called RhyCAP, that attempts to provide high-level control to the users through rhythmic parameters. The performance of Chinese lion dance has been used as an example to illustrate the idea of composing animations from fundamental motion elements organized in an action graph. The research reported in this paper is based on the platform and animation procedures implemented by this work. However, the problem of coordinating two dancers’ footsteps on tops of piles is a new challenge to be tackled in this paper.

3. PROBLEM DESCRIPTION

3.1 System Architecture

The architecture of our animation system can be composed of three modules: environment initialization, motion planning, and animation generation as shown in Figure 2. We assume that we are given a geometric description of danc-

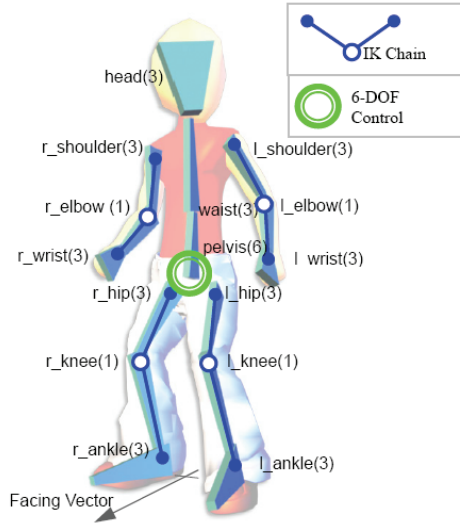


Figure 4. Humanoid kinematics model (Each joint’s DOF is shown in the parenthesis.)

ers and the environment including the locations and heights of an array of piles, denoted by p_n , generated by the system in random. The position of a pile p_i is denoted by (x_i, y_i) . An example scene used in our experiments is shown in Figure 3. We will also be given the initial and goal configurations of the lion, including footsteps of the two dances. In this preprocessing step, we will use the positions of the piles to build their adjacency relationship to order to facilitate future processing. Given the input and constraint specifications, the motion planning module uses the best-first search algorithm to find a sequence of legal footsteps on the piles. This footstep arrangement is then sent to the next module of animation generation to determine the center of mass for each character and generate the corresponding animations for moving on the piles. These animations are generated in a procedural manner with the use of the RhyCAP system [2].

3.2 Configuration and C-space

The Chinese lion dance is performed by two dancers. The head dancer leads the direction of the movement as well as manipulating the lion head. The tail dancer follows the head dancer by holding the waist of the head dancer. The distance between the two dancers is not fixed but kept in a comfortable range. In the problem considered in this paper, we need to account for the positions of the four legs of the two dancers and how they move to reach the goal. Therefore, we can define a configuration of the lion as $q=(s_1, s_2, s_3, s_4)$, where s_j is one of the piles’ indices, p_i . If there are n piles in total, then the size of the configuration space, defined as the set of all possible configurations, will be n^4 . The objective of the planner is to find a legal path connecting the initial configuration, q_i and goal configuration q_g . Although the size of the overall search space is large, only a portion of the space is legal. The legality of a configuration will be defined in more detail in the next section.

3.3 Humanoid Model

The model of virtual character for the lion dancers used in

Algorithm: Lion_Dance_BFP

Input: start & goal configuration q_i , and q_g

Output: path P

begin

```

1 PathFound = false;
2 append  $q_i$  to  $L$ ;
3 mark  $q_i$  as visited;
4  $C_{min} = \text{GetCost}(q_i)$ ;
5 while !PathFound or !Empty( $L$ )
6    $q = \text{removeFirst}(L)$ ;
7   do for each configuration  $q'$  adjacent to  $q$ 
8     if  $q'$  is a legal and unvisited configuration
9     then
10      insert  $q'$  to  $L$  with a pointer toward  $q$ ;
11      mark  $q'$  as visited;
12      if  $\text{GetCost}(q') < C_{min}$  then  $C_{min} = \text{GetCost}(q')$ ;
13      if  $x = q_g$  then PathFound = true;
14 if PathIsFound then
15   return  $P$  by tracing the pointers from  $q_g$  back to  $q_i$ 
end

```

Figure 5. Algorithm of the motion planner for Chinese Lion Dance

this work is similar to the model used in the RhyCAP system. The model consists of hierarchical articulated chains of 15 nodes with a total 40 degrees of freedom as depicted in Figure 4. When computing a pose of a humanoid character, we first determine the location of the pelvis, where the center of mass is assumed. Then the orientation of the body trunk is determined according to the pose in a martial-art form. Finally, the end effectors of the legs are specified according to the footsteps generated by the motion planner. The remaining joints are then determined by solving the kinematics chain by an inverse kinematics (IK) solver.

3.4 Posture Generation

Our system uses the approach of procedural animation as in the RhyCAP system to generate the motions for the virtual characters. An *action* is defined as a motion segment that can position the character into a distinct form of martial art. An action is usually comprised of several motion clips separated by key postures and controlled by elementary motion controllers. A key posture for the lower body is determined by specifying the pelvis and the footstep locations. The pelvis location is computed according to the distance between two footsteps. The wider the legs are separated, the lower the pelvis will be. For example, the key postures for a step-forward action may consist of three controllers: raising the free leg, moving the pelvis, and lowering the free leg. The sequence of controllers for all the possible actions can be concisely represented by the stance action graph proposed in [2]. By traversing through the graph at run time, we can generate the sequence of postures for a given action. The animation between two postures can then be generated by appropriate interpolation procedure implemented by the motion controller.

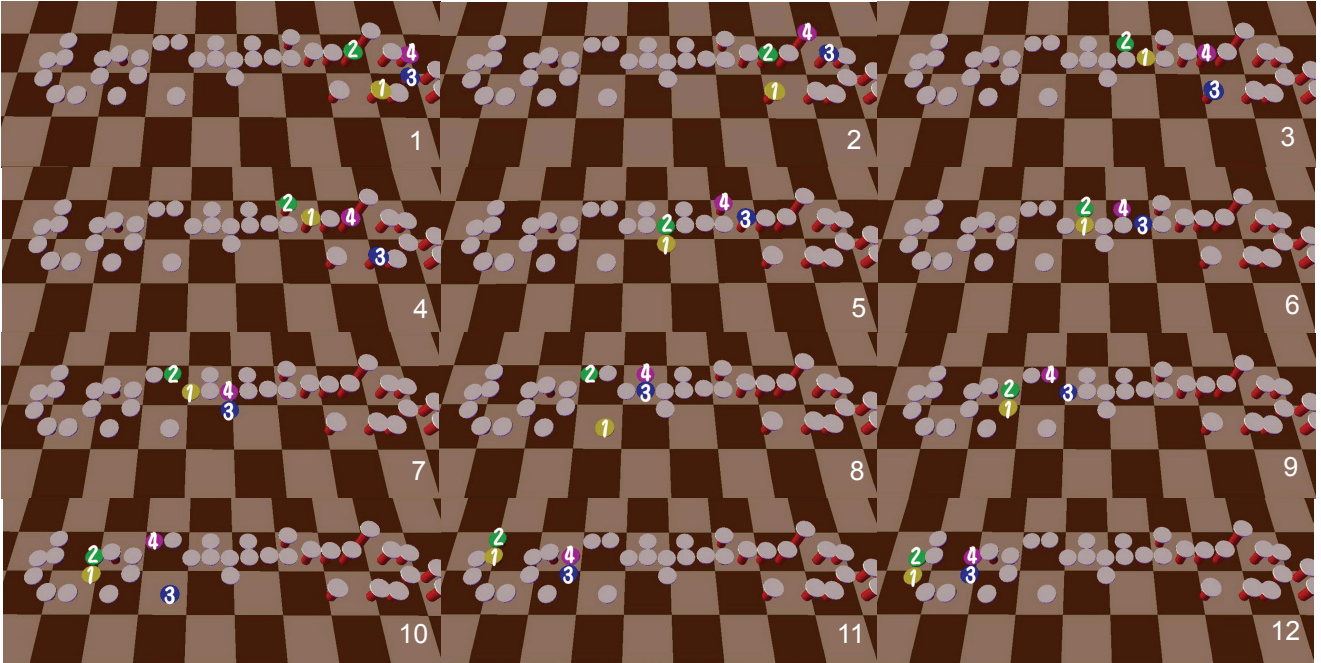


Figure 6. An example of footstep sequence for the four legs of the lion dancers generated by the path planner.

4. MOTION PLANNING AND ANIMATION GENERATION

4.1 Planning Algorithm

Although the dimension of the problem space is four, we may still be able to search the space in a systematic manner since the legal portion of the space could be much smaller than the overall space. Therefore, we have adopted the Best-Search Planning (BFP) algorithm (as shown in Figure 5) that is commonly used in the motion planners for lower dimensional problems. The search starts from the initial configuration q_i , and inserts it into a list L , the candidates for further exploration. In a search loop, the planner gets the lowest-cost configuration q in L according to some cost function to be described in the next subsection. The neighbors of q are explored further and inserted into L if it is legal and unvisited. The definition of legality and neighborhood will be given in the next subsection as well. If one of the neighbors is the goal configuration, then a feasible path can be constructed and returned by tracing from q_g back to q_i . The planner returns failure if L becomes empty, which means that all possible configurations connected to q_i have been visited.

4.2 Legality, Neighborhood, and Search Criteria

In the planner described in the previous subsection, a configuration is evaluated according to the following conditions: (1) Different legs of the dancers cannot fall on the same pile; (2) The legs of the tail dancers cannot get beyond the legs of the head dancers' (3) The difference of two consecutive footsteps cannot exceed certain limit imposed by the kinematical constraint of the human model; (4) The legs of a dancer cannot be spread too wide such that the kinematical constraint of the lower body will be violated; (5) The footstep arrangement cannot separate the two

dancers to an extent that the connection between them may be broken.

All the five conditions above contribute to the definition of legality of a configuration. The first two conditions are hard constraints that must be satisfied. The third condition is used to define the neighborhood relation such that the branching factor in the search tree is not too large. The last two conditions use kinematics constraints to set limits on feasible configurations. In addition, they are also soft constraints that are used to measure the “goodness” of a configuration, in addition to the distance to the goal configuration, when searching for a legal path in the configuration space.

5. EXPERIMENTAL RESULT

5.1 Implementation

We have implemented the aforementioned planner for generating lion dance motion on tops of randomly arranged piles. The development environment is on Microsoft .NET 2003 platform with the C++ language. The user interface, scene management, and animation rendering are based on the Open Inventor API [10], and the geometry of the virtual characters was modeled with the Alias Maya package. The animation of the locomotion for the lion dancers are based on the work of RhyCAP system. We have also implemented a pile arrangement module that can generate a meaningful and challenging array of piles randomly.

5.2 Examples of Generated Path and Animation

In Figure 6, we show an example of footstep sequence generated by the path planner for the lion dancers. Each snapshot shows a configuration consisting of four footsteps, two for the head dancer (yellow(1) and green(2)) and two

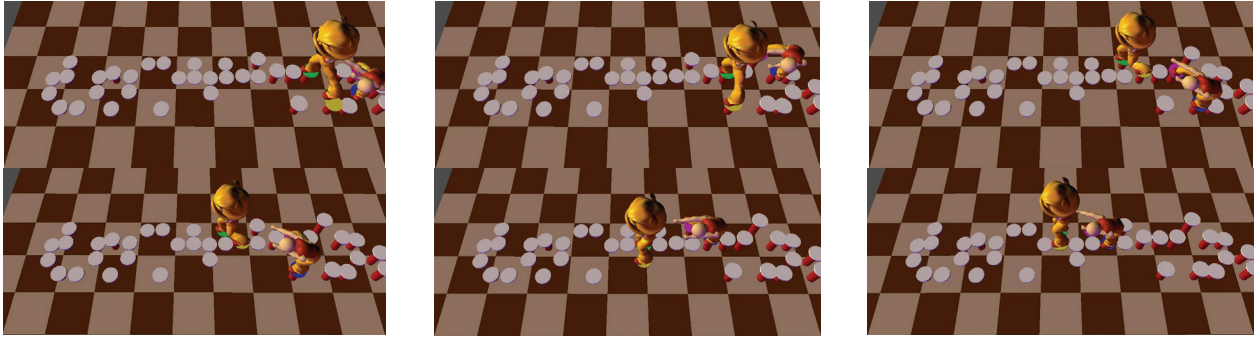


Figure 7. Snapshots of animation generated for the footsteps of the example in Figure 6.

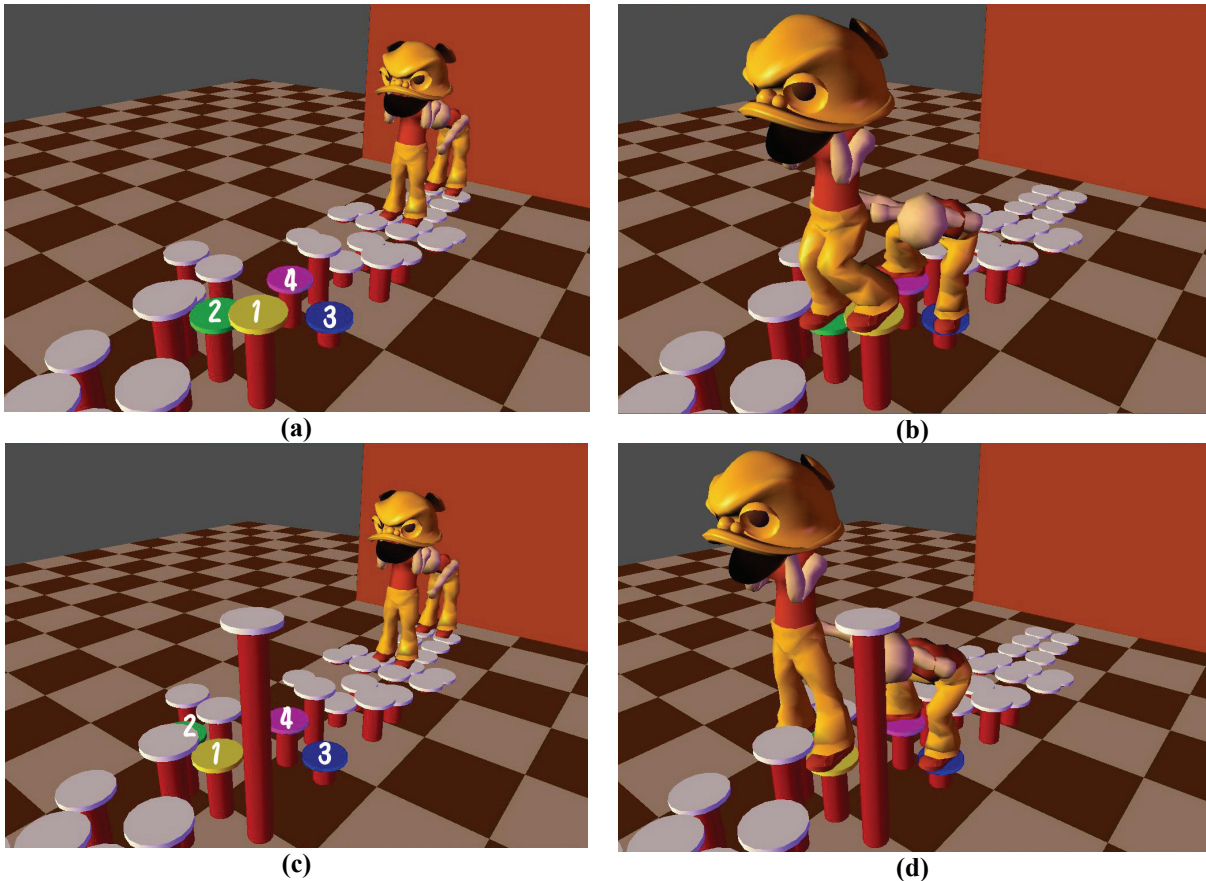


Figure 8. Comparison of search results with and without a obstructive tall pile. Compared to (a) and (b), in (c) and (d), we show an alternative configuration generated by the planner to get around a tall pile.

for the tail dancer (blue(3) and purple(4)). Note that the footstep arrangements in the snapshots are all legal configurations that do not violate the kinematics constraints of the two dancers. In addition, the search criteria will prefer the configurations with easier postures that do not separate the legs too much whenever possible. Nevertheless, this criterion is only a soft constraint that can be sacrificed if necessary. For example, the postures of the head and tail dancers in snapshots 8 and 10, respectively, cause the legs of the dancers to be spread widely.

In Figure 7, we show several snapshots of the animation for the footstep sequence planned by the path planner for the example in Figure 6. In Figure 8, we show a comparison example to illustrate the correctness of the planner. In the path and animation generated in Figures 8(a) and 8(b),

the dancers pass through the array of piles to reach their goals. In Figures 8(c) and 8(d), we raise the height of the pile used to be stepped upon by leg 1 on purpose to see if the planner can avoid this pile and generate a footstep sequence to get around this tall pile. The results show that indeed an alternative path without violating the kinematics constraint was generated to avoid collision with the obstacle.

We have run experiments on several randomly generated scenes to verify the correctness and effectiveness of planner. The experiments were run on a personal computer with a 1.6GHz Intel Pentium M CPU. Although the search space is a four dimensional space, the actual legal space where the feasible path resides is much smaller. The planning time for a typical scene consisting of around 60 piles is

around 4ms. Therefore, the performance of the planner is adequate for being used in an interactive manner.

6. CONCLUSION AND FUTURE WORK

Providing high-level controls for sophisticated computer animation has been the design goal for animation software. In this paper, we have proposed a path planning and procedural animation system that can generate challenging motions for the performance of Chinese lion dance on tops of piles of various heights. The problem is formulated as a path planning problem in a four dimensional search space to find a feasible sequence of footsteps for the lion dancers. The animation for the performance of the dancers is then realized by a procedural animation system based on the RhyCAP system that can generate lower-body motions according to the given footstep locations of the lion dancers. The effectiveness of the animation system is demonstrated with several examples.

Although the path planner is capable of generating lion dances on the piles, the lion can only perform stepping actions that can bring the lion forward. In order for the lion to perform more versatile actions such as jumping to make turns, we are designing more animation procedures and motion controllers that can enrich the actions of the dancers. Once the motion skills are enhanced, the path planning problem will then become more complicated and require special care on choosing these actions. Although we focus on the animations of Chinese lion dance in this paper, the principles of using motion planners to facilitate the generation of global paths and using animation procedures to generate animations in a dynamic environment could be extended to other types of animations as well.

7. ACKNOWLEDGEMENT

This research was funded in part by the National Science Council (NSC) of Taiwan under contract no. NSC 94-2213-E-004-006.

REFERENCES

- [1] A. Bruderlin and T. W. Calvert, "Goal-Directed, Dynamic Animation of Human Walking," *Proc. of ACM SIGGRAPH*, 1989.
- [2] J.R. Chen and T.Y. Li, "Rhythmic Character Animation: Interactive Chinese Lion Dance," *Proc. of Intl. Conf. on Computer Animation and Social Agents*, 2005.
- [3] J.P. Desai, *Motion Planning and Control of Cooperative Robotic Systems*, PhD Thesis, University of Pennsylvania, 1998. Available at URL: <http://www.ircs.upenn.edu/download/techreports/1998/98-27.pdf>
- [4] C. Esteves, G. Arechavaleta, J. Pettre and J. P. Laumond, "Animation Planning for Virtual Mannequins Cooperation," in *Proc. of Eurographics/ACM SIGGRAPH Symp. on Computer Animation*, 2004. Available at URL: <http://www.laas.fr/RIA/RIA-research-motion-character.html>
- [5] M. Girard and A.A. Maciejewski. "Computational Modeling for the Computer Animation of Legged Figures," *Computer Graphics (SIGGRAPH '85 Proceedings)*, pp.263-270, 1985.
- [6] Y. Koga, K. Kondo, J. Kuffner, and J.-C. Latombe. "Planning motions with intentions," *Proc. of SIGGRAPH'94*, pp.395-408, FL, 1994.
- [7] J. Kuffner, "Goal-Directed Navigation for Animated Characters Using Real-time Path Planning and Control" *Proc. of CAPTECH'98 Workshop on Modeling and Motion capture Techniques for Virtual Environments*, Springer-Verlag, 1998.
- [8] J. Kuffner, et al., "Motion Planning for Humanoid Robots under Obstacle and Dynamic Balance Constraints," *Proc. of IEEE Intl. Conf. on Robotics and Automation*, May 2001.
- [9] J. Latombe, *Robot Motion Planning*, Kluwer, Boston, MA, 1991.
- [10] Open Inventor, an object-oriented toolkit for interactive 3D graphics. URL: <http://oss.sgi.com/projects/inventor/>
- [11] M. van de Panne, "From Footprints to Animation," *Computer Graphics Forum*, 16(4):211-224, 1997.