Directing Intelligent Digital Actor with Real-Time Motion Planning

Tsai-Yen Li
Computer Science Department, National Chengchi University,
64, Sec. 2, Zhi-Nan Rd, Taipei, Taiwan 116
li@nccu.edu.tw

Abstract

Automatic motion generation for digital actors under real-time user control is a challenging problem for virtual environment applications such as on-line games. Despite many advances have been made in the past few years, most systems cannot generate full humanoid motions in real time on an uneven terrain cluttered with obstacles of various heights. In this paper, we propose a real-time motion planning system designed to enable the intelligent digital actor to walk according to the user's control while avoiding collisions with the environment. We adopt a prioritized decoupled approach to plan the full motion of the human figure. The system consists of a locomotion planner for the lower body and a compliant motion planner for the upper body. The locomotion planner generates appropriate footstep locations as well as the collision-free intermediate motions for the lower body. The compliant motion planner can generate motions for the upper body to avoid higher obstacles. Several examples are used to demonstrate the effectiveness of the planning system in creating realistic motions for digital actors. We believe that the realism and controllability of digital actors enabled by this work will inspire more applications of real-time motion planning.

1 Introduction

Although the concept of "digital actor" is not novel, realizing such an intelligent character in a real-time application remains challenging. The ultimate goal of digital actor is to provide an autonomous realistic animated agent which is able to sense, act, and interact in a virtual environment. This goal is very similar to the problems faced by the robotics community [9]. However, the target application of digital actor allows one to make appropriate assumptions to simplify the problem and to reduce the uncertainties that often occur in the real world.

The topic of real-time character animation has been addressed by several research communities, and the results of each community are complementary. Computer Graphics community puts emphasis on realism of the character modeling and motion simulation. Realism can be obtained by motion imitation thanks to motion capture technologies. More recently, in the robotics community, mainly in the area of motion planning, many researchers have attempted to realize autonomous digital actors by enabling them with the capacity of task and locomotion planning [7][10].

In this work, we focus on the motion planning aspect of digital actor and assume that the information about the environment is provided by the system at run time and the global motion of the digital actor is controlled by the user with a mouse in real time. The system is responsible for generating appropriate local motions in real time to accommodate the environment for the given global path. More specifically, given the expected trajectory of the humanoid character in a window of future time frames, the motion planning system is capable of generating a sequence of footsteps and the corresponding intermediate collision-free motions for the lower body. For example, the actor may stride over short obstacles or step onto a stair without explicitly being instructed. In addition, if there exist obstacles at a higher location along the global path, the digital actor should also be able to plan her upper-body compliant motion to avoid potential collisions. For example, an actor may flex her arm in order to avoid hitting an object on a table, or she may bend her waist in order to pass a lower doorway.

The rest of the paper will be organized as follows. We will first review some previous work pertaining to our research in the next section. Then, we will describe the planning problem we consider in this paper in more details in Section 3. We will then present the real-time motion planners for the lower body and the upper body in the next two sections, respectively. In Section 6, we will demonstrate the effectiveness of the planners with several examples. Finally, we will conclude the paper in the last section.

2 Related Work

2.1 Character animation

Among the character animation approaches in the literature, motion capture is a powerful and widely used technique for generating realistic human motion. However, since the motions were captured in a particular environment, the motion clips may not be easily adapted to other scenarios with different constraints. Many techniques have been proposed to overcome this problem and reuse the captured motions for a wider range of scenarios. For example, motion warping [15], blending [14], dynamic filter [16], and signal processing algorithms all attempt to address this problem by offering some additional flexibility through interpolating or extrapolating joint trajectories in the time domain, frequency domain, or both. In practice, they usually can be applied to a limited set of situations involving minor changes to the environment. Significant changes typically lead to unrealistic or invalid motions.

Alternatively, kinematics-based methods are more flexible than the motion capture technologies in handling obstacles in an uneven terrain. Forward and inverse kinematic models have been designed to synthesize walking motions for human figures in a procedural manner. Early work in computer animation focused on generating human walk motion to achieve a high-level goal [1] or generating a dynamically stable motion for a given path on a flat or uneven ground [6][12]. However, motion planning techniques are not adopted for generating humanoid motions automatically until recent years.

2.2 Decoupled planning for humanoid motions

Due to the computational complexity of the motion planning problem, planning the motion of a digital actor with full degrees of freedom (DOF) still cannot be tackled directly [9]. Most researches take a two-level planning approach that decouples the motion of the humanoid base (pelvis) from the rest of the body (limbs) [7][10][14].

In [7], a gross motion planner has been proposed to generate humanoid body motion on a flat ground in real time. All obstacle geometry within the character's height range is projected orthographically onto a grid as obstacle. Then the gross planning problem is reduced to find a collision-free path for a disc in a 2D plane. Captured cyclic motion data along with a PD controller is used to generate the final motion for the character as it tracks the global path. In [8], a humanoid robot with real-time vision and collision detection abilities is presented. The robot can plan its footsteps amongst obstacles but cannot step onto them.

In [14], a digital actor is modeled as active (all degrees of freedom attached to the legs) and reactive (attached to the upper parts of the body) degrees of freedom. A collision-free moving trajectory can be computed for the active part of the digital actor. Then the motion warping technique is applied to the motion capture data along the path when the reactive part of the digital actor collides with obstacles. In [2], the proposed planner evaluates footstep locations for viability using a collection of heuristic metrics of relative safety, required efforts, and overall motion complexity. At each iteration of the search loop, a feasible footstep is selected from the footstep transition sets.

In [3], sequences of valid footprints are searched through augmented probabilistic roadmap with a posture transition graph, and then the hierarchical displacement maps each pair of footprints to their corresponding motion clips. Since the motion clips are captured in advance, the locomotion may not be flexibly adapted to uneven terrain to avoid collisions.

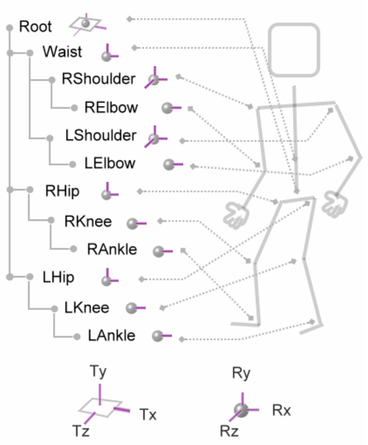


Figure 1: Kinematics model of the digital actor.

The two-level planning approach is also used by the planner proposed in [10][11]. The environmental obstacles can have different heights at various layers. The planner is able to find the global path to reach a goal at a location of different height. A local motion planner is also used to realize the locomotion of the digital actor along the generated path. In addition, [11] enables the digital actor to have the ability of striding over scattered columns connecting disconnected regions. Although the planner is quite efficient and can be used in an on-line manner, it is not designed to be used in a real-time interactive application. Besides, the planner only considers the locomotion of the lower body.

3 System Overview

3.1 Problem description

Assume that we are given a 3D geometric description of the humanoid digital actor (A) and the objects (B) in the environment. The uneven ground is described as a terrain map and treated as a regular object. A full human figure model has more than sixty DOF. Appropriate assumptions have to be made in order to simplify the planning problem. The kinematic model of the actor used in our system is depicted in Figure 1. We assume that the root of the hierarchy has four DOF that allows the actor to translate in three dimensions and rotate on the standing plane. The upper body has freedom on the waist (2DOF) and the two arms (4DOF each). Therefore, the actor is capable of rotating or bending her waist and swinging her arms freely. The lower body has four DOF for each leg (2DOF on the hip). Consequently, the full kinematic hierarchy of the actor considered in this work contains twenty-two DOF.

In our system, we assume that mouse is the device for directing the digital actor. The vertical and horizontal

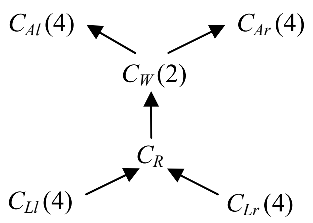


Figure 2: Order of planning by which the configurations of the branches in the hierarchy are determined.

components of a dragged vector are interpreted as the translational and rotational velocities of the object under control as in most 3D graphical user interfaces. The main difference is that the object under control now is the digital actor instead of the viewpoint. Obviously, using a 2D input device to control an actor with twenty-two DOF is not an easy task.

It is the objective of the system to generate realistic actor motions according to user mouse commands and environmental constraints. First, the body of the actor needs to remain collision-free from all objects in the environments. When walking, the actor should have at least one foot touching the ground. In addition, as directed, the actor should be able to step onto objects at different heights if it is kinematically feasible. Most importantly, the required computational cost of the planning system cannot defeat the interactive nature of the underlying application.

3.2 System architecture

Despite motion planning techniques have made great advances in recent years, the curse of dimensionality remains. Due to the real-time requirement of the system, it is doomed to be intractable to use a centralized approach to consider the composite configuration space of the whole body. Instead, we adopt a decoupled approach by decomposing the body of the human figure into hierarchy of five smaller branches that are considered sequentially. These branches include two legs, two arms, and the waist. The root of the hierarchy is at the pelvis and the two arms are connected to the root via the waist. The configuration spaces for the left leg, right leg, waist, and left arm and right arm are denoted by C_{Ll}, C_{Lr}, C_W, C_{Al}, and C_{Ar}, respectively. The order of determining the configuration for each branch in the proposed planning system is depicted in Figure 2. The numbers in the parenthesis indicate the dimensionalities of each individual configuration space.

Given a user mouse input command, the horizontal location and facing direction of the actor for the next step can be computed. If the input is not changed often, the path for the actor in the next *n* steps can be further derived. By taking the intersection of the ground and the sagittal plane of the actor, one can compute the ground profile along the global path. This ground profile is taken as an input to the locomotion planner for determining footsteps. The locomotion planner (to be described in Section 4) determines the configuration of the lower body (C_{Ll} and C_{Lr}) as well as the pelvis (C_R). The computed pelvis configuration is then sent to the compliant motion planner (to be described in Section 5) to determine the configuration of the upper body (C_W, C_{Al}, and C_{Ar}).

In order to achieve real-time planning performance, we have to limit the window of footsteps to be considered in the planners to a small number, say three. In the real life when we walk on the ground, we also only account for the next few steps ahead only. Therefore, short-term planning is not only more feasible but also more intuitive for controlling the digital actor. Nevertheless, the decoupled approach is incomplete in nature. If any of the two planners fail, the motion of the actor for the next step is not generated. In this case, the user may feel stuck at the location and need to take some maneuvers in order to move forward again. The real-time performance requirement is important for the application but the failure situation is not fatal because it could also happen in any user interface with collision detection but no planning.

4 Locomotion Planning on Uneven Terrain

In this section, we will describe a planner that can generate appropriate footsteps and intermediate leg motions for the window of the next *n* steps from the current configuration on an uneven terrain. First, we assume that we are given the ground profile along the global path. In order to reduce the computational cost, we project the lower body of the actor into its sagittal plane and assume that the legs only move on a 2D plane. The dimension of rotating the legs vertically to make turns is temporarily ignored and will be recovered to respect the path in a post-processing step. Next, we will first describe how the lower-body motion for a walk cycle is generated for a given footstep and then the planning algorithm that has been used to generate a sequence of efficient footsteps.

4.1 Generation of walk locomotion

We use a procedural animation approach to generate the walk motion of the digital actor as shown in the system flow chart in Figure 3. We assume that the footstep planner to be described in the next subsection will output a desired foot location for the next step. According to the level of the next step, the motion can be classified into three categories: moving on a leveled ground, uphill, and downhill. Key frames in a motion cycle are defined differently for different motion categories as depicted in Figure 4. For example, for climbing uphill, the front leg is bent first while the rear leg is bent first for stepping downhill. Once the key frames have been computed according to the next footstep location, the motions be-

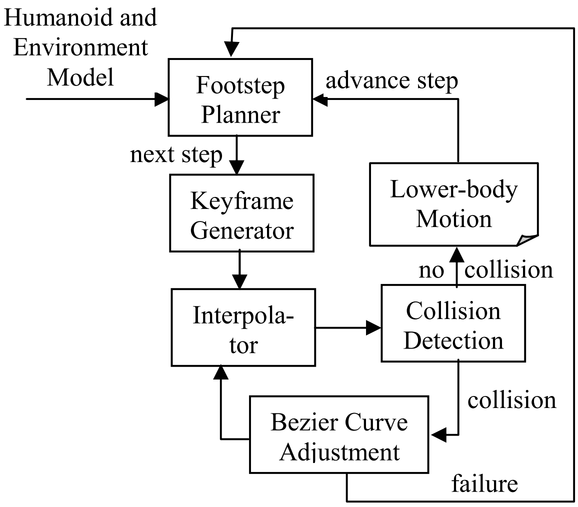


Figure 3: System flow chart of the locomotion planner

tween any two key frames can then be computed by interpolating joint angles or end point locations. For the free leg swinging in the air, a Bezier curve in the Cartesian space is defined and interpolated for the ankle. If the curve causes the leg to collide with the ground profile, the Bezier curve adjustment module is evoked to adjust the curve upward with the two control points until it becomes collision-free [10]. If no feasible curves are possible, the system will return failure to the footstep planner for further testing.

4.2 Footstep planning

Given a ground profile and the kinematic model of the digital actor, the footstep planner uses a sliding window approach to generate feasible footstep locations for the next n steps ahead. The collection of the footstep lengths for the next n steps is defined as *footstep configuration*, $q_f = (s_1, s_2, s_3, \ldots, s_n)$, where $s_{min} \leq s_i \leq s_{max}$, $i = 1, 2, \ldots, n$. s_{min} and s_{max} are the lower and upper bounds of the footstep lengths. If we divide the range of (s_{min}, s_{max}) into m intervals, then there would be a total of $n*(m+1)$ footstep configurations in the search space of the footstep planner.

The planner starts the search at some initial configuration $q_{init} = (s_{init}, s_{init}, \ldots, s_{init})$, where s_{init} is some initial neutral footstep length. Since this initial configuration may not be collision-free, the planner will first attempts to search for a collision-free one by performing a breadth-first search. Once a collision-free configuration is found, the planner proceeds to search for the best configuration according to the user-specified criteria such as optimal energy consumption [10]. Due to the real-time planning requirement, local optimal on the search criteria is assumed to be satisfactory. Therefore, the planner stops and returns the current configuration whenever the neighbors are all worse than the current one.

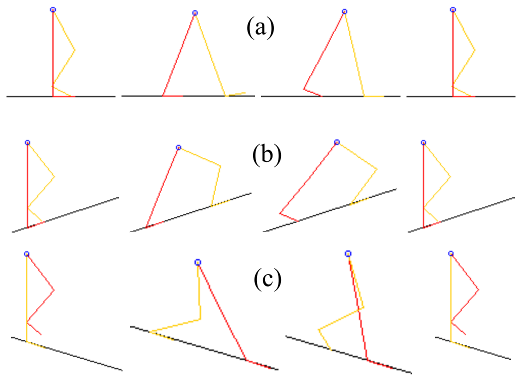


Figure 4: Key frame definitions for walking (a) on a flat ground, (b) uphill, and (c) downhill.

Although n steps are planned, only one step is consumed at a time. For example, assume that a legal footstep configuration $q_0 = (s_1, s_2, \ldots, s_n)$ is found by the planner but only the next footstep s_1 is used. When the planner is called again, the search window slides forward by one step, and the initial configuration is reset to $(s_2, s_3, \ldots, s_n, s_{init})$. Although there is no guarantee that only the last footstep needs to be checked, this configuration usually serves as a good starting point for the next search to converge.

5 Upper Body Motion Planning

The locomotion planner described in the previous section is capable of generating a collision-free motion for the lower body of the digital actor in the next n step. However, before the actor makes the next step forward, the system needs to further ensure that the upper body is also collision-free. According to Figure 2, the root (pelvis) location $q_R(t)$, once determined by the locomotion planner, become a function of time when the plan is executed. The upper body will then move based on the pelvis trajectory. We again take a decoupled approach to decompose the upper body into three parts: waist, left arm, and right arm. The decoupling is prioritized such that the waist is determined first and then the arms.

5.1 Motion pattern as objective function

The upper-body motion of a digital actor is not arbitrary when the actor is walking. For example, the waist is kept erect whenever possible, and the arms swing alternatively in order to keep body balance for the foot motions. These local motions are usually defined according to the underlying action and treated as a preferred motion patterns that one should adhere to whenever possible. We define the motion patterns for the part x of the upper body by an objective function $O_x(t)$, where x belongs to $\{W, Al, Ar\}$. For example, O_W could be a linear function to keep the waist erect while O_{Al} and O_{Ar} could be a sinusoidal function corresponding to the foot motions. These

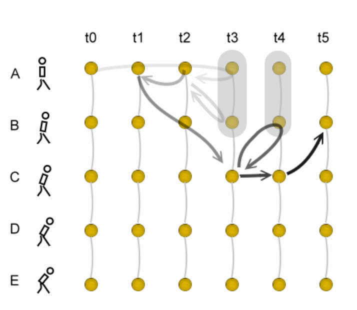


Figure 5: Illustration of the search process in CT-roadmap. A to E represent the nodes in the roadmap while t_i's are the indices of time.

motion patterns are discretized into a sequence of configurations indexed by time according to time resolution of the animation frames generated by the locomotion planner.

5.2 Building CT-Roadmap for the upper body

Due to our decoupled planning approach, the motions of the upper body are computed under the constraints of the generated lower-body motions. As the pelvis moves along a given trajectory of length k, the relative configuration of the upper body with respect to the environment also changes. The concept of Configuration-Time Space (CT-space) has been used in modeling such a time-dependent planning problem [4]. However, the dimensionalities of the subproblem we have for each individual C-space (C_W, C_{Al}, C_{Ar}) discourage us to build an explicit CT-space (up to 5D) for direct search. On the other hand, the concept of probabilistic roadmap has been shown to be effective for high-dimensional motion planning problems [5]. Therefore, we propose to build a roadmap in CT-space, called *CT-roadmap*, to represent the structure of the freespace. In other words, CT-roadmap is a roadmap extended to include the additional time dimension.

The CT-roadmap for each subproblem is initialized with some common motion patterns for that portion of the body, and the roadmap may be enhanced incrementally as the search proceeds. Due to the nature of CT-space, we restrict the search to advance forward in time to the neighboring nodes only. We have adopted a Best-First Planning (BFP) algorithm to search for the goal of a legal configuration in the final time step k. The “best” criterion is defined as the closeness of the current configuration from the desired one computed by the objective functions described in the previous subsection. If all the trials to advance the current node fail, the system will enhance the roadmap by growing a new node with some distance r away from the current node. If the new node does not bring the search advance in time, the system backtracks to a earlier node as illustrated in Figure 5.

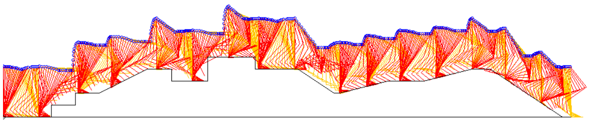


Figure 6: Animation example of the lower body walking on an uneven terrain

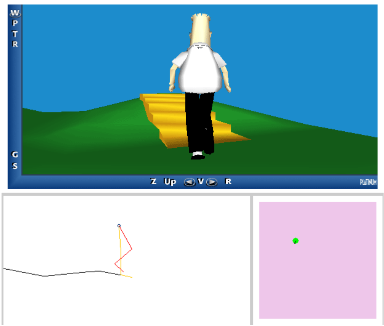


Figure 7: User interface of the planning system allowing interactive control of the digital actor

6 Experimental Results

The planning system proposed in this paper has been implemented in Java and connected to a VRML [17] browser for interactive control and 3D display. All experimental data are measured on a regular PC with a P4-2.8GHz CPU. For the lower body, collisions are checked between the legs and the ground profile with 2D geometry. For the upper body, the V-clip package [12] for 3D collision detection is used.

In Figure 6, we show an example of the lower-body motions of the actor walking on an uneven terrain. The motions are generated in real time by the locomotion planner when the user controls the actor with the user interface provided by the system depicted in Figure 7. In Figure 8, we use an example to show the effectiveness of the footstep planner. In Figure 8(a), the actor steps into a trap that prevents her from moving forward. A footstep planner with a sliding window of three steps (n=3) and footstep resolution of thirteen intervals (m=13) can perceive this problem and stride over the gap as shown in Figure 8(b). The average planning time for each step of the lower-body motion is 30 ms for the example shown in Figure 6.

In Figure 9, we show snapshots of the example that we have used to test the motion planner for the upper body. The actor keeps a normal walk motion by swinging the arms alternatively as shown in Figure 9(a). When the actor continues to move forward, the actor has to rotate her waist to avoid collisions with the obstacles in the environment as shown in Figure 9(b). In Figure 9(c) and 9(d), we show another scenario requiring the actor to bend her waist to pass under the slide. The average frame

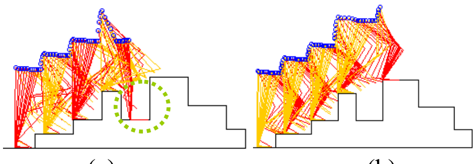
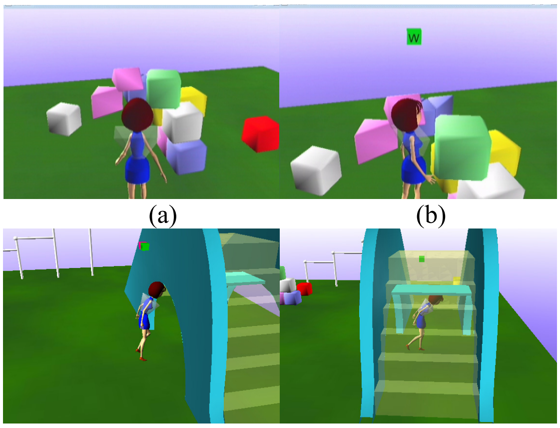

(a) (b)

Figure 8: Comparison of lower-body motions with (b) and without (a) footstep planning

rate for this example is 14.2 fps; thus, the performance of the planning system can indeed satisfy the requirement of controlling the digital actor interactively as specified in the beginning of the paper.

7 Conclusions

Controlling the motion of a digital actor in real time has always been a challenging problem because of the computational complexity involved. In this paper, we have proposed a prioritized decoupled planning approach allowing the motion of a digital actor to be controlled by the user interactively. Different portions of the body are planned sequentially with appropriate assumptions being made to achieve real-time performance. The concept of sliding window has been used in the footstep planner to save the planning results in the previous steps. In addition, the concepts of objective function and CT-roadmap have been used to make the upper body follow some motion patterns and be compliant to the environment. We believe that the work proposed in this paper will open a new direction for realizing intelligent digital actor in an interactive environment.



(a) (b)
(c) (d)

Figure 9: Snapshots of the digital actor avoiding collisions with the environmental obstacles under interactive user control

References

[1] A. Bruderlin and T. W. Calvert, "Goal-Directed, Dynamic Animation of Human Walking," *Proc. of ACM SIGGRAPH*, 1989.

[2] J. Chestnutt, J. Kuffner, K. Nishiwaki, and S. Kagami, "Planning Biped Navigation Strategies in Complex Environments," *Proc. of IEEE Intl. Conf. on Humanoid Robotics*, 2003.

[3] M.G. Choi, J. Lee, S.Y. Shin, "Planning Biped Locomotion using Motion Capture Data and Probabilistic Roadmaps," *ACM Trans. on Graphics*, 5:1–25, October 2002.

[4] M. Erdmann and T. Lozano-Perez, "On Multiple Moving Objects," in *Algorithmica*, 2(4):477-521, 1987.

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

[6] H. Ko and N.I. Badler, "Animating Human Locomotion with Inverse Dynamics," *IEEE Trans. on Computer Graphics*, 16(2):50-59, 1996.

[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, K. Nishiwaki, S. Kagami, M. Inaba, and H. Inoue, "Footstep Planning Among Obstacles for Biped Robots," *Proc. of 2001 IEEE Intl. Conf. on Intelligent Robots and Systems (IROS 2001)*, 2001.

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

[10] T.Y. Li P.F. Chen, and P.Z. Huang, "Motion Planning for Humanoid Walking in a Layered Environment," *Proc. of the 2003 Int. Conf. on Robotics and Automation*, 2003.

[11] T.Y. Li and P.Z. Huang, "Planning Humanoid Motions with Striding Ability in a Virtual Environment," *Proc. of the 2004 Intl. Conf. on Robotics and Automation*, 2004.

[12] B. Mirtich, "V-Clip: fast and robust polyhedral collision detection," *ACM Trans. on Graphics*, 17(3):177-208, 1998.

[13] H. Miura and I. Shimoyama, "Dynanic Walk of a Biped," *Intl. J. of Robotics Research*, pp.60-74, 1984.

[14] J. Pettre, J.-P. Laumond, and T. Simeon, "A 2-Stages Locomotion Planner for Digital Actors," *Proc. of Eurographics/SIGGRAPH Symposium on Computer Animation*, 2003.

[15] A. Witkin and Z. Popovic, "Motion Warping," *Proc. of ACM SIGGRAPH*, 1995.

[16] K. Yamane and Y. Nakamura, "Dynamics Filter – Concept and Implementation of On-Line Motion Generator for Human Figures," *Proc. of IEEE Intl. Conf. on Robotics and Automation*, April 2000.

[17] VRML 97 spec., URL:<http://www.vrml.org>