

Planning Fine Motions for a Digital Factotum

Gustavo Arechavaleta

Claudia Esteves

Jean-Paul Laumond

LAAS-CNRS

7, avenue du Colonel Roche
31077, Toulouse Cedex, France
{garechav, cesteves, jpl}@laas.fr

Abstract—This paper presents a motion planner for a digital mannequin carrying objects in cluttered environments. The goal is to account for 3-dimensional obstacle avoidance, eye-convincing locomotion as well as manipulation constraints within an integrated motion planning approach. This approach combines probabilistic path planning methods, a motion capture based walking controller and inverse kinematics techniques.

I. INTRODUCTION

The increasing use of Computer Animation in films, video games and CAD systems has made it necessary to develop new tools capable of automatically generating animations while allowing a high-level control of the result.

Automatically computing believable human-like motions for digital actors is still a challenging problem. While looking for believability and autonomy, related work has borrowed techniques from two different research areas: mainly Computer Graphics [1], [4], [15], but also Robotics [11].

Previous efforts to automatically plan motions for a virtual human have focused on conferring the actor some basic ability or performing only one task at a time. For instance, in [11] the authors make a virtual character perform manipulation planning. Locomotion has been treated separately from several approaches surveyed in [14].

A two-step path planner for digital actors was proposed in [12]. This approach consists in first planning a collision-free path for a cylinder in a 2D world and then animating the digital actor along the path. This work was extended in [17] to deal with 3D obstacle avoidance. Another approach using automated footprint computation and inverse kinematics is described in [2] to deal with walking on rough terrain.

This paper presents an automated motion planner for a digital mannequin carrying a bulky object in a cluttered environment (i.e. the so-called *digital factotum*). This approach is based on an analysis of the global task according to three types of constraints:

- 3D obstacle avoidance
- believable locomotion
- object manipulation

To address these constraints altogether we combine three types of techniques within the same framework: probabilistic path planning methods to deal with obstacle avoidance; a motion capture based walking controller to provide

believable animations and inverse kinematics techniques to deal with object manipulation.

This work is an extension of [16] where only the first two constraints were considered. Therefore, the main contribution of this paper is to show how to integrate inverse kinematics techniques (borrowed from [20]) while preserving obstacle avoidance.

The problem is thus to plan the motions for a walking virtual factotum that carries a movable object with both hands. For this, three main steps are performed. First a collision-free trajectory is found for the digital factotum and his movable object. Then, manipulation and locomotion are synthesized in the animation step. Finally, residual collisions are treated in the last step.

The remainder of this paper is organized as follows: the next section briefly describes our digital actor's model. Section III explains the model of the elementary manipulation task and details the framework in which our virtual factotum evolves. Section IV describes the three stages of the planner. In Section V results are shown and discussed.

II. DIGITAL ACTOR'S MODEL

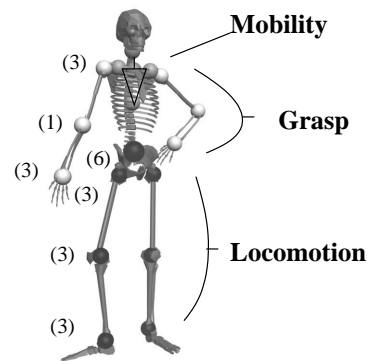


Fig. 1. Our virtual factotum has 53 degrees of freedom decomposed into three groups: locomotion, grasp and mobility.

The kinematic structure of our digital actor is composed of 20 rigid bodies articulated by 18 joints with 53 degrees of freedom (DOF). These bodies and joints form five kinematic chains that converge in the virtual actor's root located on the pelvis. The root degrees of freedom define the actor's position and orientation in the world, while the

remaining DOFs define the orientation of the rest of the joints. Such a kinematic structure is classical (e.g. [1], [7]).

The degrees of freedom are then decomposed into three groups according to the task they are destined to perform (Figure 1), i.e. *grasp*, *locomotion* and *mobility* DOFs.

Locomotion (resp. grasp) DOFs are those which take an active role at walking (resp. manipulating). Each of these two groups involves two kinematic chains, i.e. legs in locomotion and arms in grasp. The mobility DOFs (i.e. spine and clavicles) are those involved in the local deformation of a kinematic chain trajectory when residual collisions are found at some stage.

III. ELEMENTARY TASK-MODELING

To model the task we consider a system composed of our digital actor (53 DOFs) and a movable object (6 DOFs) evolving in a 3-dimensional workspace. The object DOFs are labeled as mobility DOFs.

To ensure that the mannequin will reach the object, a manipulation volume is defined according to the human-arm inverse kinematics (IK) solution [20]. The grasping position of the movable object should consequently always lie inside this volume. Because we consider manipulation task involving both hands, a closed-kinematic chain is formed. To deal with obstacle avoidance, this chain should be treated using a dedicated motion planner. This system is illustrated in Figure 2.



Fig. 2. A volume is defined according to the virtual human arms IK solution. The object has to lie within this volume to ensure the manipulation.

Our 59-DOF system (53 from the digital actor and 6 from the object) can be constrained in order to take into account physical properties as well as application-specific motions. These constraints are given as user's input to the algorithm. For example we might want to keep a tray with Tequila shot glasses horizontal (reducing our system to 57 DOFs) during the animation (Example V-B).

IV. PLANNER ARCHITECTURE

Planning the motions for our system is achieved by applying several consecutive steps. The first step is to plan a collision-free trajectory for the digital actor and the object. Then, the kinematic-chain composed of the hands and the object is closed. Locomotion and manipulation are synthesized. Finally, residual collisions with the spine,

clavicles and arms are treated. The following paragraphs describe each of these steps.

A. Stage I: Trajectory Planning

In this step a collision-free path is planned for a reduced 9-DOF model (Figure 3) of our system. Three of these DOFs are the planar position and orientation of a cylinder bounding the locomotion DOFs of the digital actor. The other six DOFs are the 3-dimensional position and orientation of the object in the environment. The object is considered to be attached to the virtual human's root within the limits allowed by the manipulation volume described in section III.



Fig. 3. The planning step considers a 9-DOF system composed of the bounding cylinder and the object

The path planner is based on probabilistic roadmaps [8]. Such an approach requires:

- a random sampling strategy to compute collision-free configurations, and
- a steering method (or local planner) to compute elementary paths between configuration pairs.

The sampling strategy we use is the visibility-graph approach [19]. Such a strategy performs well in computing the topology of the collision-free space within a compact roadmap. The 6-dimensional sampling domain of the object should be defined in such a way it guarantees that any random object configuration in that domain corresponds to an admissible two-hand grasping position for the actor. This is done by a user-defined 6-dimensional rectangular box approximating the 6-dimensional domain reachable by the grasp DOFs.

The steering method processes the bounding cylinder and the object in different ways. We use a Bezier curve of third degree for the bounding cylinder parameters and a straight line segment for the object parameters. Because the digital actor is assumed to walk only forwards along Bezier curves, the roadmap is directed.

Once the roadmap is computed during the learning phase, a classical query phase is applied to solve a given problem [8].

At this stage, the resulting path ensures a collision-free motion of the lower part of the body and of the object. Then, the path is transformed into a trajectory (i.e.

a time parametrized path) with user-defined velocity and acceleration constraints.

Figure 4 reproduces some configurations of the trajectory resulting from the planning step. The root of the digital actor follows the path computed for the bounding cylinder and the object moves according to his own path. Note that the object remains within the virtual human's reach.

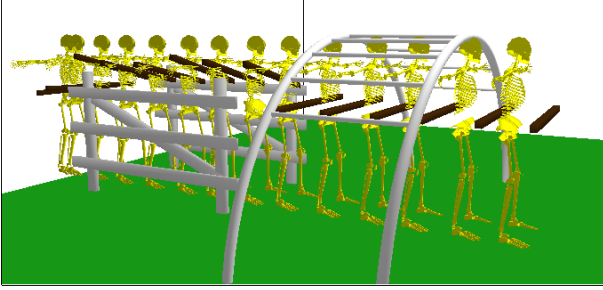


Fig. 4. Some configurations of the 3-dimensional trajectory resulting from Stage I. Here, collision-free trajectories are planned for the bounding cylinder and the object.

B. Stage II: Motion Synthesis

Once a trajectory has been generated for the 9-DOF model, two parallel steps are performed to animate the motions for the whole system (59-DOF). One of these steps is to animate the locomotion DOFs and the other is to animate the grasp DOFs.

To animate the locomotion DOFs, a locomotion controller [17] is applied along the trajectory. This controller is based on a motion capture blending technique and provides as a result a walking sequence along the trajectory. This step animates not only the legs, but also the mobility DOFs (spine and head).

Now, to synthesize the manipulation subtask we consider a closed kinematic chain composed of the virtual actor's arms and the movable object. The object configurations computed in the first stage impose values for the grasp DOFs of the arms. Attaining such values is done by using the inverse kinematics algorithm proposed in [20].

In Figure 5 some frames of the animation resulting from synthesizing posture and task motions are shown. The collision-free motion is preserved for the locomotion DOFs and for the object but not for the spine and arms. Here, we can see a residual collision between the head and the monkey bars which will be avoided on the next step. Note that the digital actor holds the object on all the frames on the shown configurations.

C. Stage III: Residual Collisions

At Stage I, collision-free paths were computed for the object and for the locomotion DOFs of the digital actor. However, the upper part of the virtual human's body (as it is shown on figures 4 and 5) may have residual collisions.

The purpose of the current step is to locally modify the trajectory of the two kinematic chains that compose the upper body (spine-head and arms-object) in order to avoid

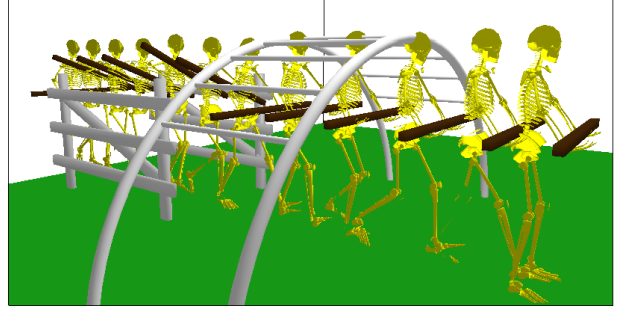


Fig. 5. The 59-DOF system is animated performing two parallel steps, one for animating the virtual actor's posture, the other to animate the task.

collisions. The idea is to find a valid configuration for each of the colliding chains and to apply a warping procedure [21] along the configurations to obtain smooth motions.

If the spine-head is the colliding kinematic chain, the trajectory is randomly modified until a valid (collision-free) configuration is found. This configuration is optimized to obtain a minimal deformation (see [16] for details on the procedure).

More complicated is the case where the closed kinematic chain body-arms-object is in collision. Solving this problem is related to path planning for closed kinematic chains. Among the few methods proposed in the literature [3], [6], [13], we use the RLG algorithm [3]. For this, we consider the movable object as the active part of the closed chain. Once a valid configuration is found (at random) for the object, the configuration of the passive chain (arms) is found using inverse kinematics.

After these 3 stages, if there are no collision-free configurations found, the original path of Stage I is invalidated and a new path is computed.

Figure 6 illustrates the case where the colliding chain is the open chain formed by the spine and the head. In this case, a valid configuration was found when the virtual actor bends his spine to avoid collision. The believability of the motion is preserved by using the warping procedure.

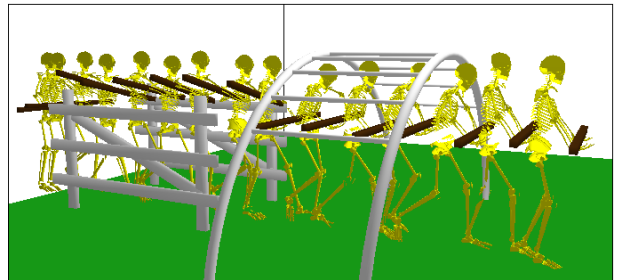


Fig. 6. Without neglecting his task, the digital actor gracefully avoids smashing his head with the monkey bars.

V. RESULTS

This approach has been implemented within the motion planning platform Move3D [18] as an extension of the architecture proposed in [16]. In the following paragraphs

two examples in different scenarios are presented and discussed.

A. Working at the Factory

In this example, our virtual human has become a worker carrying cold steel rails around a typical industrial environment. Figure 7 shows the complexity of the large-scale environment where the digital factotum evolves. Here, there are plenty of obstacles (pipes, drums, beams, ventilation units, etc.) that the actor has to deal with along his way.

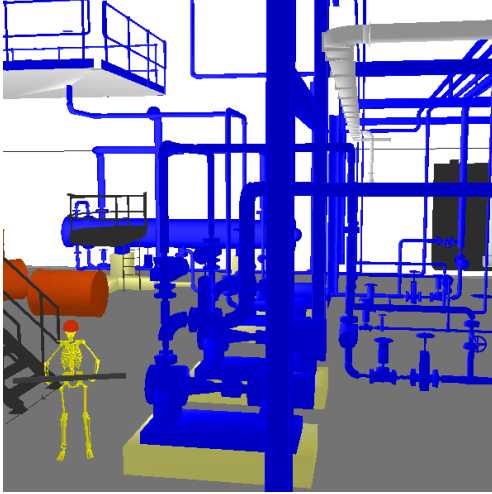


Fig. 7. The large-scaled industrial environment where the digital factotum evolves.

Figure 8 shows key-frames of the animation generated after the initial and final configurations were specified by the user. This sequence of configurations illustrates different posture and task motions accomplished by the worker. At the beginning of the trajectory, two collisions are found, one with the balcony, the other with the metal drums. The collision head-balcony is automatically solved at Stage II by bending the spine and head of the actor. In a similar way, a valid configuration for avoiding the drums is found by rotating the object to maintain the task achievable. Figure 9 is a close-up of the behavior mentioned above. Note that by applying this solution the generated animation remains eye-convincing.

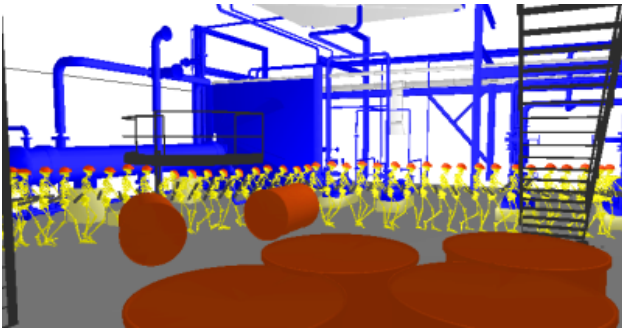


Fig. 8. Our digital actor has to deal with several obstacles while performing his planned trajectory

Once these obstacles are avoided, the digital actor has to smoothly regain his posture (resp. task) to accomplish the next part of the trajectory. Figures 10 (a) and (b) show an example in which the virtual worker adjust his spine's position to attain a smooth change between the end of the narrow passage and the beginning of the open area.

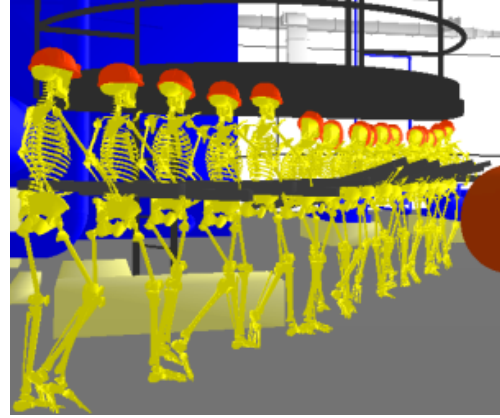


Fig. 9. The actor changes his posture as well as the object position to avoid the balcony and the metal drum.

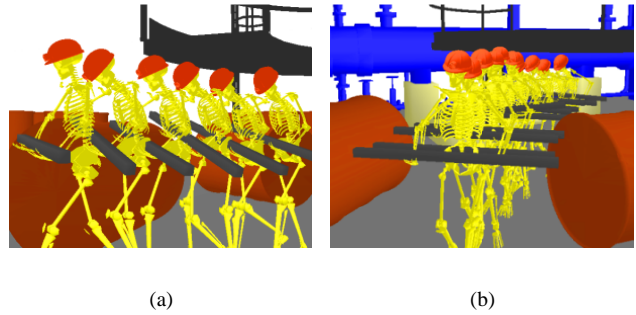


Fig. 10. The virtual human regains his posture to prepare himself for further eventualities along his way. (a) shows how the object position is modified and (b) illustrates the emerging factotum.

Once the balcony is avoided, the virtual worker keeps following his trajectory and avoiding collisions until the final configuration is reached.

B. Eugene "el Mariachi"

In this example our virtual factotum Eugene is asked to take a bottle of Tequila to his guests. The tray with the bottle and the glasses is expected to keep horizontal during the animation. We have thus, in roll-pitch-yaw angles, removed the degrees of freedom that allowed the tray to roll and pitch.

Our living-room environment is small and contains large objects (e.g., a grand piano, a desk, etc.). The walkways are therefore narrow and residual collisions are likely.

Figure 11 shows a trajectory between the grand piano and the wall. The paintings hanging on the wall make the passage narrower at the tray level. Figure 11 (a) and (b) illustrate the initial and final configuration desired for the



(a) Initial configuration

(b) Final configuration



(c) Completed trajectory

Fig. 11. The digital actor has to raise the tray to avoid hitting the grand piano.

animation. In (c) key-frames of the followed trajectory are presented. Note the 3-dimensionality of the trajectory when the object is raised in order to avoid collision with the tail of the piano. In the animation step the arms will follow the object and the locomotion controller is applied to follow the virtual human's trajectory. In this example no residual collisions were found.

In Figure 12 we consider another situation: Eugene should go between the piano and the desk. Here, after performing the planning and animation steps, a residual collision was found between the left arm and the reading lamp.

A valid configuration would be found by randomly modifying the object position (respecting its constraints) and then making the arms follow by IK. However, since we know from the planning step that the object is collision-free a solution that avoids moving the tray is preferred. For this, we profit in this case from the 7-DOF arm IK capability of using the redundant DOF (elbow) to shun the collision. A valid range for this DOF is given by the user in terms of what seems plausible for the application (around 30 degrees in our case). If there is no valid configuration found within this range, the standard procedure is applied.

In the example of Figure 12, a valid configuration was found by slightly swiveling the elbow towards the actor's body.



Fig. 12. The virtual human swivels his elbow to avoid colliding with the reading lamp.

C. Computational time

The planner has been tested on a workstation Sun-Blade-100 with a 500MHz UltraSparc-IIe processor and 512 MB RAM. The required time to compute the examples presented above are given in Tables II and III (averaged over 100 runs).

TABLE I
MODEL COMPLEXITY (NUMBER OF POLYGONS).

	Environment	Factotum
Factory		
- Complete	159,698	17,239
- Col. Test	92,787	14,038
Living-Room		
- Complete	25,007	21,582
- Col. Test	25,007	18,381

In Table I the number of polygons in the different environments as well as the digital mannequin model are presented. Here we have identified the polygons taking part in the collision test. In the living-room environment we consider that all the objects for collision test purposes. For the digital mannequin, the number of polygons considered is limited to the upper part of the body and the object as we know the lower part is collision-free.

Tables II and III show the experimental results of the 3 stages. As it was expected, the time it takes to find a trajectory is significantly higher in the industrial environment because of its size and complexity. Here, the results

TABLE II
COMPUTATIONAL TIME IN FACTORY (IN SECONDS).

	Factory (Fig.8)	
No. Frames	270	537
Stages		
I. Planner		
- Path	4.0	– 4.0
- Trajectory	1.4	– 3.0
II. Animation	0.5	– 0.9
III. Residual Col.	1.5	– 3.6

TABLE III
COMPUTATIONAL TIME IN LIVING-ROOM (IN SECONDS).

Stage	Piano (Fig.11)		Lamp (Fig.12)	
No. Frames	79	157	41	80
Stages				
I. Planner				
- Path	2.4	- 2.4	0.5	- 0.5
- Trajectory	0.3	- 0.7	0.1	- 0.3
II. Animation	0.27	- 0.45	0.1	- 0.1
III. Residual Col.	0.0	- 0.0	0.13	- 0.26

are expressed considering a pre-computed roadmap of the environment in a learning phase. The learning phase takes 7.6s for the Factory and 3.8s for the Living-Room. Tables II and III present only the results of the queries. Two different animations are generated for each trajectory, the second improving the animation quality (i.e. doubling the number of frames).

The fact that the trajectory planning step is the most time-consuming is mainly due to the path optimization. This optimization depends only on the length of the computed path and not on the number of frames in the animation. Unlikely, the sampling step relies strongly on the number of frames desired.

The time it takes to compute the animation step varies proportionally with the number of frames remaining fast in all the examples.

The residual collision step relies heavily on the complexity of the environment but also on the number of frames with collision. Note in the Piano example that the collision computing time is zero because there are no collisions.

VI. CONCLUSIONS

We presented a strategy to plan human-like motions for a virtual character that walks and holds an object at the same time. A complete animation is automatically generated by performing three consecutive steps: planning a trajectory, synthesizing motions and avoiding residual collisions.

The kinematic structure of the digital actor has been decomposed allowing a simple representation of the task. This task description can be adapted to permit cooperative manipulation between several virtual characters. This is the objective we intend to reach in a next step.

Note that performing complex tasks such as the tasks considered in this paper, may be achieved using dynamical or physical approaches developed in robotics [10] as well as in graphics [5], [9]. Nevertheless, the dynamical and physical approaches, as powerful as they are, are based on *local* optimization. Work should be done to integrate them within a *global* planning perspective. We intend to develop such a work in the near future.

ACKNOWLEDGMENT

G. Arechavaleta and C. Esteves benefit from SFERE-CONACyT grants. This work is partially funded by the European Community Projects FP5 IST 2001-39250 Movie and FP6 IST 002020 Cogniron. The authors would like to thank Julien Pettré for his valuable help.

Animations related to this work can be found at <http://www.laas.fr/RIA/RIA-research-motion-character.html>

REFERENCES

- [1] N. Badler, C. Phillips, and B. Webber, *Simulating Humans: Computer Graphics, Animation, and Control*. University of Pennsylvania, Philadelphia: Oxford University Press, 1992.
- [2] M. Choi, J. Lee, and S. Shin, "Planning biped locomotion using motion capture data and probabilistic roadmaps," *ACM transactions on Graphics*, Vol. 22(2), 2003.
- [3] J. Cortés, T. Siméon, and J.-P. Laumond, "A random loop generator for planning the motions of closed kinematic chains using prm methods," in *Proc. IEEE International Conference on Robotics and Automation (ICRA'2002)*, 2002.
- [4] R. Earnshaw, N. Magnenat-Thalmann, D. Terzopoulos, and D. Thalmann, "Computer animation for virtual humans," *IEEE Computer Graphics and Applications*, 1998.
- [5] P. Faloutsos, M. van de Panne, and D. Terzopoulos, "Composable controllers for physics-based character animation," in *Proceedings of ACM SIGGRAPH 2001 Conference*, 2001.
- [6] L. Han and N. Amato, "A kinematics-based probabilistic roadmap method for closed chain systems," in *Proceedings of International Workshop on Algorithmic Foundations of Robotics (WAFR'00)*, 2000.
- [7] J. J. Kuffner, "Goal-directed navigation for animated characters using real-time path planning and control," *Lecture Notes in Computer Science CAPTECH'98*, vol. 1537, pp. 171–186, 1998.
- [8] L. Kavraki, P. Svestka, J.-C. Latombe, and M. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," in *Proc. IEEE Transactions on Robotics and Automation*, 1996.
- [9] O. Khatib, O. Brock, K. Chang, D. Ruspini, L. Sentis, and S. Viji, "Human-centered robotics and interactive haptic simulation," *Robotics Research*, vol. 23, no. 2, February 2004.
- [10] O. Khatib, K. Yokoi, K. Chang, D. Ruspini, R. Holmberg, A. Casal, and A. Baader, "Force strategies for cooperative tasks in multiple mobile manipulation systems," *Robotics Research: The Seventh International Symposium*, pp. 333–342, 1996.
- [11] Y. Koga, K. Kondo, J. Kuffner, and J. Latombe, "Planning motions with intentions," *Computer Graphics*, vol. 28, pp. 395–408, 1994.
- [12] J. Kuffner, "Autonomous agents for real-time animation," Ph.D. dissertation, Stanford University, Stanford, CA, December 1999.
- [13] S. LaValle, J. H. Yakey, and L. E. Kavraki, "A probabilistic roadmap approach for systems with closed kinematic chains," in *Proc. IEEE Transactions on Robotics and Automation*, 1999.
- [14] F. Multon, L. France, M.-P. Cani, and G. Debonne, "Computer animation of human walking: a survey," *The Journal of Visualization and Computer Animation*, vol. 10, pp. 39–54, 1999.
- [15] R. Parent, *Computer Animation: Algorithms and Techniques*. The Ohio State University: Morgan-Kaufmann Publishers, 2001.
- [16] J. Pettré, J.-P. Laumond, and T. Siméon, "A 2-stages locomotion planner for digital actors," in *Proc. ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. San Diego, California: Eurographics Association, 2003, pp. 258–264.
- [17] J. Pettré, T. Siméon, and J.-P. Laumond, "Planning human walk in virtual environments," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'02)*, Lausanne, Switzerland, 2002.
- [18] T. Siméon, J.-P. Laumond, and F. Lamiroux, "Move3d: A generic platform for motion planning," in *Proc. 4th International Symposium on Assembly and Task Planning (ISATP'2001)*, 2001.
- [19] T. Simeón, J.-P. Laumond, and C. Nissoux, "Visibility based probabilistic roadmaps for motion planning," *Advanced Robotics*, vol. 14, no. 6, 2000.
- [20] D. Tolani, A. Goswami, and N. Badler, "Real-time inverse kinematics techniques for antropomorphic limbs," *Graphical models*, no. 5, pp. 353–388, 2000.
- [21] A. Witkin and Z. Popovic, "Motion warping," in *Proc. of SIGGRAPH'95*, 1995.