Proceedings of the 2001 IEEE
International Conference on Robotics & Automation
Seoul, Korea • May 21-26, 2001

Planning Motion Patterns of Human Figures Using a Multi-layered Grid and the Dynamics Filter

Zvi Shiller[1], Katsu Yamane[2], Yoshihiko Nakamura[2]
[1] Department of Mechanical and Aerospace Engineering
University of California
Los Angeles, CA 90095

[2] Department of Mechano-Informatics
University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo, 113-8656 Japan

Abstract

This paper presents a practical motion planner for humanoids and animated human figures. Modeling human motions as a sum of rigid body and cyclic motions, we identify body postures that represent the rigid-body part of typical motion patterns. This leads to a model of the configuration space that consists of a multi-layered grid, each layer corresponding to a single posture. A global search through this reduced configuration space yields a feasible path and the corresponding postures along the path. A velocity profile is calculated along the optimal path, subject to the speed and acceleration limits assumed for each posture. Cyclic motions, generated from "primitive" cyclic motion patterns for each posture, are then added to the trajectory produced by the path planner. This "kinematic" motion is then modified by the dynamics filter [16] to result in dynamically consistent behavior. Examples are presented which demonstrate the use of this planner in an office environment.

1 Introduction

Recent research on motion planning for humanoids and animated human figures has been driven by the computer-game and movie industries and by several visible humanoid projects in Japan [2] and in the US. Unlike earlier research, which focused on generating stable motions [14, 8], the focus of recent works has been on synthesizing "natural" looking, or human-like, motions for interactive animation [4, 1, 16, 6] and for real-time control of humanoids [16, 13, 11]. Because of the large number of degrees of freedom involved in synthesizing full motions of humanoids or human figures, current approaches to generating natural-looking motions rely heavily on motion captures [10, 6, 16]. Most motion planners thus first generate a path for the center body, and then fit motion capture data to this path. The focus in the latter procedure is usually on kinematic "correctness," i.e. producing motions that are smooth and continuous in all joints. Such motions, however, are usually dynamically "incorrect," in that they do not satisfy the character's own dynamics and physical constraints due to parametric differences between the original data and the edited motion.

Ensuring dynamic feasibility, or consistency with the equations of motion, is one step towards producing natural-looking animations and physically feasible motion patterns for humanoids. As an effort to this goal, the authors have proposed the dynamics filter [16]. It computes the closest joint motions to the original data that are dynamically feasible and satisfy all physical constraints. The dynamics filter can be used to produce motion patterns reflecting changes in the mass distributions or the geometry of body parts (a fat/thin or tall/short body, heavy boots, a backpack).

Despite the extensive use of character animation in games and movies, only few planners have been developed to account for obstacle avoidance. One such planner was presented in [6] for on-line animation of biped characters. It generates the shortest path, which is tracked by a simple feedback controller. Its efficiency stems from reducing the motion planning problem to a disk moving in a 2D space (the walking surface) by representing the character by a bounding cylinder. The velocity profile along the path is modulated by a PD controller, using the average velocity of the motion capture data as the reference speed. This model requires that the obstacles be spaced sufficiently apart to allow a forward-facing walk, which precludes situations where it is not possible. Applications where greater attention to obstacle avoidance is necessary, include humanoids moving in realistic environments, such as construction sites, power plants, and damaged structures. Other environments, more likely to be used for character animation, include crowded offices, hospitals, class rooms, and movie theaters, where the character is forced to squeeze between closely spaced furniture or people. Addressing environments, where a single motion pattern (or posture) such as forward-facing motion is insufficient, is the focus of this paper.

Sensor-based obstacle avoidance of a humanoid was addressed in [15]. This work focused on planning the foot steps to ensure dynamic stability while avoiding vision-sensed obstacles. In [5], randomized trees were used to plan collision-free grasping of a 7 DOF arm. Both planners address the detailed motions of the legs and hands, and are hence impractical for full-body gross-motion planning.

This paper presents a gross-motion planner that considers several, yet potentially unlimited, number of motion postures by modeling human walk as consisting of rigid body and of cyclic motions. The "rigid-body" motion represents the net motion of the body's main parts. Absent any cyclic components, rigid-body motion consists therefore of the body posture that is characteristic of each motion pattern, such as forward-facing walk or a crawl. The number of steady-state (sustainable over a period of time) postures is assumed small and typical of a given human figure. This allows an efficient representation of the "rigid-body" configuration space by modeling each posture by one layer of a multi-layered graph.

The planner first selects a path and motion postures by searching over the multi-layered graph. It then computes the velocity profile along the path by modeling the human figure as a point mass, propelled by a single force in the direction of motion. The "rigid-body" postures are then substituted with "kinematic" cyclic motions of all body parts. The kinematic cyclic motions are generated from "primitive" cyclic motion patterns, consisting of a single cycle produced from motion capture data. The dynamics filter [16] then filters the kinematic motion to ensure that the resulting motion reflects the actual parameters of the human figure and satisfies its equations of motion. We have implemented the planner for the three most common motion patterns: a forward-facing walk, a walk sideways, and crawling. While crawling is not commonly used in office environments or in movie theaters, it is often the pattern of choice in search and rescue operations. The planner is demonstrated for a human figure moving in a realistic office environment.

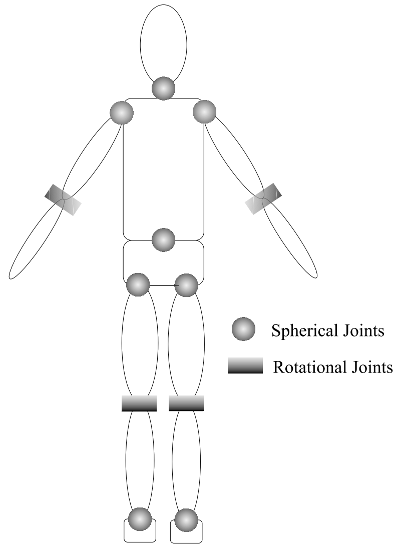

Figure 1: Human figure model

2 Modeling

2.1 Human figures

We consider a full-body human figure, with 13 links and 34 degrees of freedom (including the 6 DOF of the base body), as shown in Figure 1. For gross-motion planning, it is convenient to describe human motions as consisting of rigid body motion and cyclic motion [3]. This allows separating the motion planning problem into first determining the rigid body trajectory, and then attaching to it the cyclic motion. Such decomposition of the problem is necessary to overcome the inherent difficulty of planning in the large configuration space.

Planning for rigid body motions in $\mathcal{R}^3$ can be further simplified by recognizing that typically, the body attains a small number of sustainable postures, some shown in Figure 2. These postures are essentially approximations of the actual motion with all harmonics filtered out. We assume a small number of such postures, although more can be added with no difficulty. Reducing the number of possible configurations to a distinct set greatly simplifies the motion planning problem without significantly loosing motion fidelity.

To simplify collision checking, we represent each motion posture by a bounding polyhedron, as shown in Figure 3. Each polyhedron is then replaced with a cylinder, of the same height, and a radius equal to its

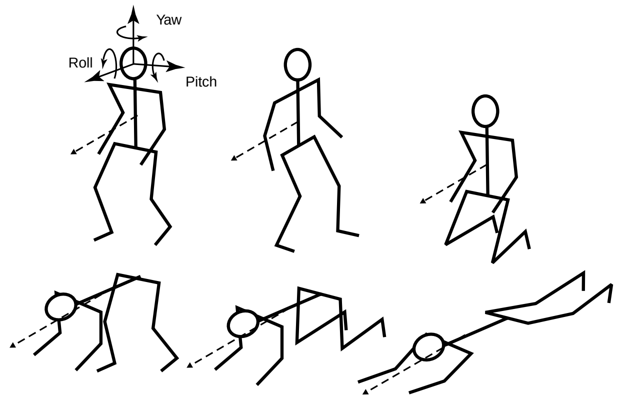

Figure 2: Various motion postures

width (measured perpendicular to the direction of motion), as shown in Figure 4. This allows us to ignore rotation, thus reducing the configuration space of each pattern to 2D. For polyhedra that are wider than they are long, such as for the forward-facing walk, this approximation is conservative, and hence does not represent any difficulty. For polyhedra that are longer than they are wide, such as in sideways walk and crawl, this approximation does not pause a problem for straight-line motions, since the free space along the straight line provides room for the long, and unmodeled, part of the polyhedron. The problem arises at the turning points, where a free space for the cylinder does not ensure sufficient free space for the polyhedron. We circumvent this difficulty conservatively by permitting rotations only at points where sufficient free-space does exist, as discussed later.

Denoting the configuration space of the rigid body motion by G, and the configuration space of the full-body cyclic motion by H, the total configuration space is $Q = G \times H$. By planning for the rigid body motions, we actually plan in G.

Denoting the configuration space spanned by each motion pattern by $C_i \subset \mathcal{R}^2$, we effectively approximate G by $C \subset G$, $C = \bigcup C_i, (i = 1, \ldots, m)$, where m is the number of postures considered. C represents an efficient approximation of the *attainable* configuration space of the rigid body motion. Any inaccuracies introduced with this crude representation will be eliminated at the time the cyclic motion is added to the rigid body motion and filtered by the dynamics filter, as discussed later.

2.2 Obstacles

The obstacles are assumed known, or identified before motion planning begins. The obstacles can be of any shape, although polyhedral shapes are easier to handle.

For each reduced configuration space C_i, we generate the free subspace, $C_{ifree} \subset C_i$, by evaluating collisions at the grid points of a uniform grid, drawn onto the

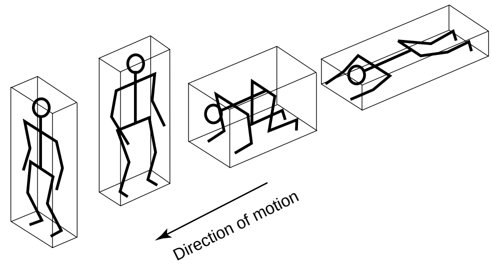

Figure 3: Bounding polyhedra of motion postures.

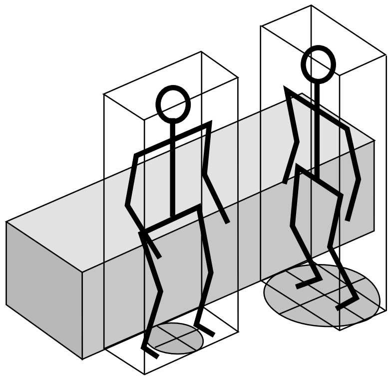

Figure 4: Replacing the polyhedron with a cylinder.

motion surface. At each grid point, the clearance to all obstacles within the height range of the cylinder, corresponding to the particular motion posture, is calculated. Points with clearance smaller than the cylinder's radius are marked "occupied." The union of all "occupied" points represents a conservative approximation of the occupancy map of all obstacles in C_i. The union of all free subspaces produces the total free-space for this problem: $C_{free} = \bigcup C_{ifree}, (i = 1, \ldots, m)$.

3 The Graph

Before constructing the search graph, we first clarify the problem we are actually solving. This problem is different from the standard shortest path problem because in addition to the path length we must account for the various motion postures. Since not all postures are created equal–some are more favorable than others, we assign to each posture a "priority factor" ν, with $\nu_1 = 1$ for the forward-facing walk, $\nu_2 = 1.1$ for sideways walk, and higher values for more taxing postures. We also wish to stay away from obstacles when possible, or equivalently, follow segments of the Voroni diagram. The problem addressed in this paper can therefore be stated as follows:

Denoting the start point $S = (s_x, s_y, i)$ (position x, y and posture i), the goal point $G = (g_x, g_y, j)$, and the free space C_{free}, compute the rigid-body motion, represented by the path $P \subset C_{free}$, from $S \in C_{free}$ to

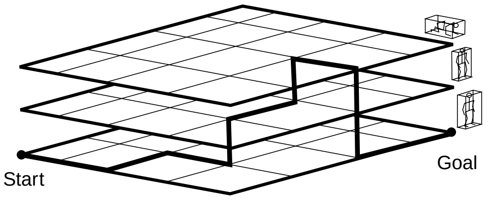

Figure 5: A multi-layered grid.

$G \in C_{free}$ that minimizes the cost function:

$$J = \int_0^{s_f} p(s)\delta(s,i)ds \tag{1}$$

where $p(s)$ is a potential function, reciprocal of the minimum distance to obstacles at each point, $\delta(s,i) = \nu_i$, i is the index of the selected posture, and s is the arc length along the path.

Minimizing (1) selects a short path that stays away from obstacles, using the highest-priority posture feasible. We now construct a weighted graph through which the shortest path solves the stated problem.

The reduced configuration space of each posture, $C_i \subset \mathcal{R}^2$ is represented by a uniform grid, projected onto the motion surface. Each grid point is a node $n_i(x,y,z)$ on the corresponding undirected graph $E_i, (i = 1, \ldots, m)$, with x, y, z being the three attributes of each node, (x, y) representing its location in a fixed coordinate frame attached to the motion surface, and z representing an assigned potential value. Each node in E_i is connected to eight of its nearest neighbors. Nodes in $C_{ifree} \subset C_i$ are assigned a value reciprocal to the distance to the nearest obstacle:

$$n_i(z) = \min_k \frac{d_0}{1 + d_{jmin}} \tag{2}$$

where k is the number of obstacles, d_0 is a constant, and d_{jmin} is the minimum distance of that node to the j-th obstacle. The value z is effectively a potential function, with lows coinciding with the Voroni diagram. This potential is used in the path search to verify sufficient room around turns.

The cost along the edge $e_i((x,y)_j,(x,y)_k) \in E_i$ between nodes $n_i((x,y,z)_j) \in E_i$ and $n_i((x,y,z)_k) \in E_i$ is the Euclidian distance, treating z as the "height" coordinate:

$$c_{j,k} = \sqrt{(x_j - x_k)^2 + (y_j - y_k)^2 + (z_j - z_k)^2}). \tag{3}$$

Minimizing this cost minimizes the path length and maximizes its distance to neighboring obstacles.

The graphs for the individual postures, $E_i, (i = 1, \ldots, m)$ are interconnected to form the total graph E by directed edges at nodes that are in the free space of both graphs. Thus, $e_{i,j,x,y}$ is an edge, connecting E_i to E_j by connecting the node $n_i(x,y), x, y \in C_{ifree}$ to node $n_j(x,y), x, y \in C_{jfree}$. The resulting graph, E, thus consists of interconnected 2D slices of the configuration space, as shown schematically in Figure 5.

Note that the graphs of the individual postures are connected to other postures according to the transition from one posture to the other. The cost for the edges between the graphs is assigned a positive value from a high priority posture to a lower priority posture (such as the sideways walk), and a zero value in the reverse direction. This encourages the search to select the preferred posture, and to return to that posture once such a transition is possible. The cost for each posture is scaled by ν to ensure that the path does not stay in a low priority posture unless absolutely necessary.

4 Path Search

A search for the shortest path through graph E accomplishes the stated objective by virtue of the cost assigned to the edges in each sub-graph E_i, and by the cost assigned to the edges connecting between the individual sub-graphs. The search itself is quite standard and deserves no special discussion, except to address the issue of orientation that was ignored when we generated the free space. This is resolved by penalizing for rotations, unless the minimum clearance to the nearest obstacle at a given node exceeds d_{turn} (see Figure 6):

$$d_{turn} = \frac{1}{2}\sqrt{(\frac{L}{2})^2 + W^2} \tag{4}$$

where W is the width and L is the length of the polyhedron. The minimum clearance is inferred from the potential value calculated in (2). Rotation is identified by a change in direction between the parent of a particular node and the parent of that parent. Obviously, this conservative treatment of rotations may miss possible solutions in tight spaces, but such approximations allow to reduce the configuration space to a small multiple of 2D spaces.

No other special attempt was made to speed up computation since the Dijkstra's algorithm [7] we used was sufficiently fast (a fraction of a second on a slow notebook PC) for a graph that accounts for three postures. The majority of the computation time was spent on processing the nodes and the collision checking. This can benefit from recent randomized tree approaches [5], however, ultimate efficiency was out of the scope of this paper.

The grid path is smoothed by a B-spline to establish curvature information for the computation of the velocity profile, as discussed next.

5 Velocity Profile

A properly selected velocity profile may add "physical reality" to the motion patterns. For this purpose,

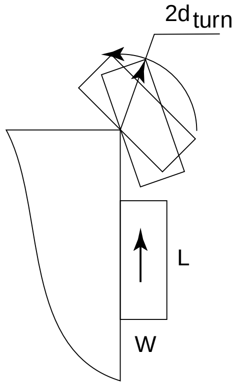

Figure 6: Minimum clearance during a turn.

we model the human figure as a point mass, pushed by a propelling force in the direction of motion. The computation of the velocity profile is formulated as an optimization problem that maximizes speed (or minimizes time), subject to constraints on the propelling force, the limit of friction force between the figure and the ground, and the speed limits for each posture.

This problem is solved efficiently using an existing algorithm for minimizing motion time of dynamic systems along specified paths [12]. The force and speed limits for the individual postures are assigned empirically to reflect typical human behavior. The velocity profile and path are used as inputs to the process of generating kinematic cyclic motions from primitive patterns produced from motion capture data.

6 Substituting Primitive Cyclic Motion Patterns

The reference trajectory generated by the motion planner consists of the projection of the body center on the walking surface, the motion posture, and the velocity vector tangent to the path. At each point, we determine the joint motions by substituting the rigid body postures with kinematic cyclic motion for each posture. The kinematic cyclic motions are motions of the limbs and the body that are scaled in time to reflect the specific motion pattern (posture) and speed along the rigid-body trajectory. They are "kinematic" since they do not necessarily satisfy physical and dynamic constraints. The kinematic motions are periodic cycles of a single cycle, we call "primitive," produced from motion capture data. The transitions between two different postures are generated by linearly interpolating the two motions around the transition time.

This procedure generates motions that may look smooth, but that are not necessarily physically feasible. Modifying the resulting motion patterns to be dynamically correct generally adds realism to the final motion. Ensuring dynamic correctness is done by the dynamics filter.

7 The Dynamics Filter

The reference trajectory, even when generated from a library of captured data, is likely to be physically infeasible due to differences between the dynamics of the human figure and the dynamics of the subject of the motion capture. The dynamics filter [16] is a computational procedure, intended to produce physically feasible motions from a given reference trajectory that may not be physically feasible. It considers the full dynamics of the human figure and is applicable to motions of any type.

The dynamics filter simulates the full-body dynamics of human figures that are driven by a stabilizing controller, using the kinematic motion patterns as the reference input. The stabilizing controller generates joint motions that satisfy system dynamics and are closest to the reference input. The performance of the dynamics filter depends largely on the performance of the stabilizing controller. We have successfully implemented a stabilizing control algorithm, however, the challenge remains to design a controller that is robust to variations in the motion patterns.

The current controller first computes the desired accelerations of the whole body, assuming all generalized coordinates are actuated. The implemented control law accounts locally for the generalized coordinates, and globally for the Cartesian position and orientation of the head. The computed accelerations of the generalized coordinates (joints) that are physically infeasible due to unactuated joints are then modified through an optimization procedure that ensures that the resulting motion is physically feasible. The controller finally produces the joint torques that correspond to the modified accelerations. The dynamic behavior of the whole body, driven by the joint torques, is then simulated.

The dynamics filter was applied to various motions, including a normal walk, jump, and karate kick [16]. We also used the dynamics filter to generate kinematically synthesized motion, such as a long walk and a turn using only one sequence of short-walk data from motion capture. Typical computation time for the controller and the dynamic simulator is approximately 50ms per frame using a common PC, which is the order of 20 times slower than real-time.

The equations of motion used in the optimization and the dynamic simulation are derived using the method developed in [9] for structure-varying kinematic chains. This method was designed to handle kinematic chains whose link connectivity changes dynamically among any possible open and closed configurations. This technique is essential for the problem addressed in this paper since the interaction between the human figure and the environment creates kinematic chains of various topologies.

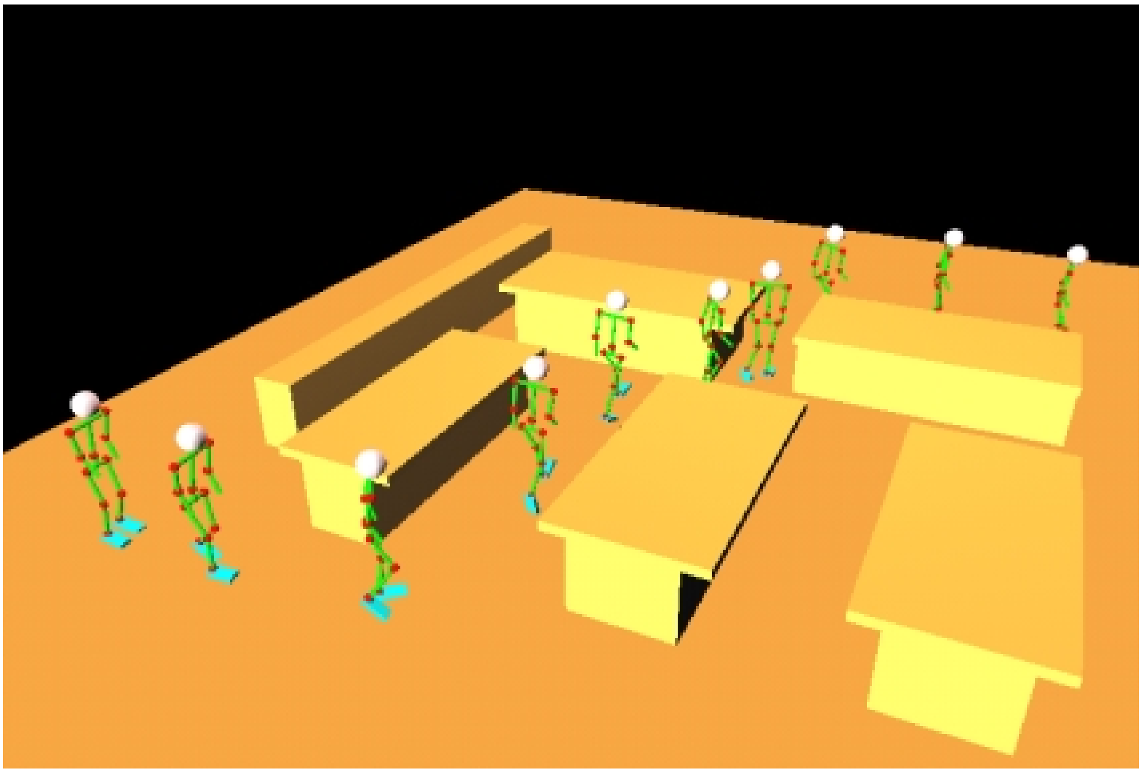

Figure 7: Primitive cyclic motion pattern substituted into the path

8 Examples

The following examples demonstrate the use of the motion planner and the dynamics filter for a humanoid moving in a typical office environment. Figure 7 shows the kinematic cyclic motion fitted to the path generated by the planner from the lower left to the upper right corners of the room. The kinematic motion is not dynamically feasible because of slippage of the feet along the turns. This is partly due to the slower speeds dictated by the velocity profile (not shown) than the speed of the original captured motion data, and partly due to the fact that the original data did not consider turns. This slippage was eliminated by filtering the kinematic motion through the dynamics filter, as shown in Figure 8. Note the slight bending of the back caused by the deceleration before each turn.

Bringing the desks closer together to force crawling and sideways motions resulted in the path and the kinematic cyclic motions shown in Figure 10. The velocity profile along this path is shown in Figure 9. The upper curve is the velocity limit curve, which reflects velocity constraints due to the path curvature and the simplified (point mass) dynamics. The lower curve is the actual velocity profile, which avoids crossing the limit curve and obeys the maximum speed limits assumed for each motion posture. This velocity profile adds "physical reality" to the resulting motion by suggesting typical human behavior.

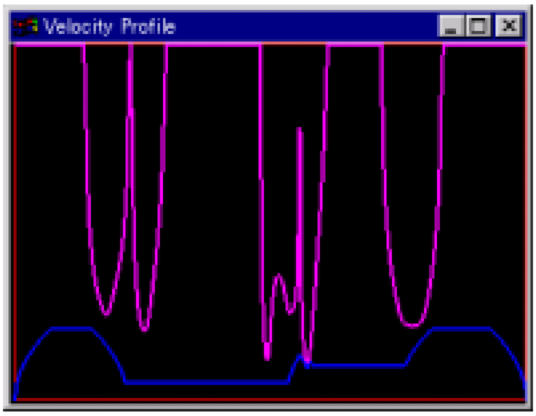

Figure 9: Velocity profile with crawl

9 Conclusions

A practical motion planner for humanoids and human figures has been presented. It considers the most common motion postures, each represented by a 2D grid. The interconnected configuration space is a small multiple of several 2D slices. This space can be efficiently searched for an optimal path that minimizes distance, maximizes clearance to obstacles, and maintains a hierarchical order of motion patterns. The speed along the path produced by the motion planner is maximized to reflect realistic behavior, subject to speed constraints on each motion pattern and simplified robot dynamics. The resulting trajectory is fitted with motion capture data, which is then filtered through the dynamics filter to ensure that the resulting motion satisfies robot dynamics and physical constraints. Numerical experiments were presented, which demonstrate the richness of the motions produced by the motion planner and by the dynamics filter.

Acknowledgments

This work is supported by the CREST project (PI: Y. Nakamura, University of Tokyo), the Japan Science and Technology Corporation.

References

[1] H. Ko and N.I. Badler: " Animating human locomotion with inverse dynamics," *IEEE Transactions on Computer Graphics*, vol.16, no.2, pp.50–59, 1996.

[2] H. Inoue, S. Tachi et al.: " HRP: Humanoid robotics project of MITI," *Proceedings of the First IEEE-RAS International Conference on Humanoid Robots*, Cambridge, MA, September 2000.

[3] J. Ostrowski and J.W. Budrick: " The Mechanics and Control of Undulatory Locomotion," *Int. Journal of Robotics Research*, vol.17, no.7, pp.683–701, July 1998.

[4] J.K. Hodgins, W.L. Wooten, D.C. Brogan, and J.F. O'Brian: "Animating human athletics," *SIGGRAPH*, vol.29, pp.71–78, 1995.

[5] J.J. Kuffner and S.M. LaValle: "RRT-Connect: An efficient apporoach to single-query path planning," *IEEE International Conference on Roboics and Automation*, pp.995–1001, San Francisco, April 2000.

[6] J.J. Kuffner Jr.: "Goal-directed navigation for animated characters using real-time path planning and control," *Proc. CAPTECH '98 : Workshop on Modelling and Motion Capture Techniques for Virtual Environments*, Springer-Verlag, 1998.

[7] E.L. Lawler: *Combinatorial Optimization*, Holt, Rinehart and Winston, New York, 1976.

[8] H. Miura and I. Shimoyama: "Dynamic walk of a biped," *Int'l Journal of Robotics Research*, pp.60–74, 1984.

[9] Y. Nakamura and K. Yamane: "Dynamics Computation of Structure-Varying Kinematic Chains and Its Application to Human Figures," *IEEE Transactions on Robotics and Automation*, vol.16, no.2, pp.124–134, 2000.

[10] Z. Popovic: "Editing Dynamic Properties of Captured Human Motion," *Proceedings of IEEE International Conference on Robotics and Automation*, pp.670–675, San Francisco, CA, April 2000.

[11] Q. Huang, K. Kaneko et al: " Balance control of a biped robot combining off-line pattern with real-time modification," *Proceedings of IEEE International Conference on Robotics and Automation*, pp.3346–3352, San Francisco, CA, April 2000.

[12] Z. Shiller and H.H. Lu: "Computation of path constrained time optimal motions with dynamic singularities," *ASME Journal of Dynamic Systems, Measurements and Control*, vol.114, no.1, pp.34–40, March 1992.

[13] S. Kagami, K. Nishiwaki et al.: " A fast generation method of a dynamically stable humanoid robot trajectory with enhanced ZMP constraint," *Proceedings of the First IEEE-RAS International Conference on Humanoid Robots*, Cambridge, MA, September 2000.

[14] M. Vukobratovic, A.A. Frank, and D. Juricic: "On the stability of biped locmotion," *IEEE Transactions of Biomedical Engineering*, pp.25–36, 1979.

[15] M. Yagi and V. Lumelsky: " Biped robot locomotion in scenes with unknown obstacles," *IEEE International Conference on Roboics and Automation*, pp.995–1001, April 1999.

[16] K. Yamane and Y. Nakamura: " Dynamics Filter — Concept and Implementation of On-Line Motion Generator for Human Figures," *Proceedings of IEEE International Conference on Robotics and Automation*, pp.688–695, San Francisco, CA, April 2000.

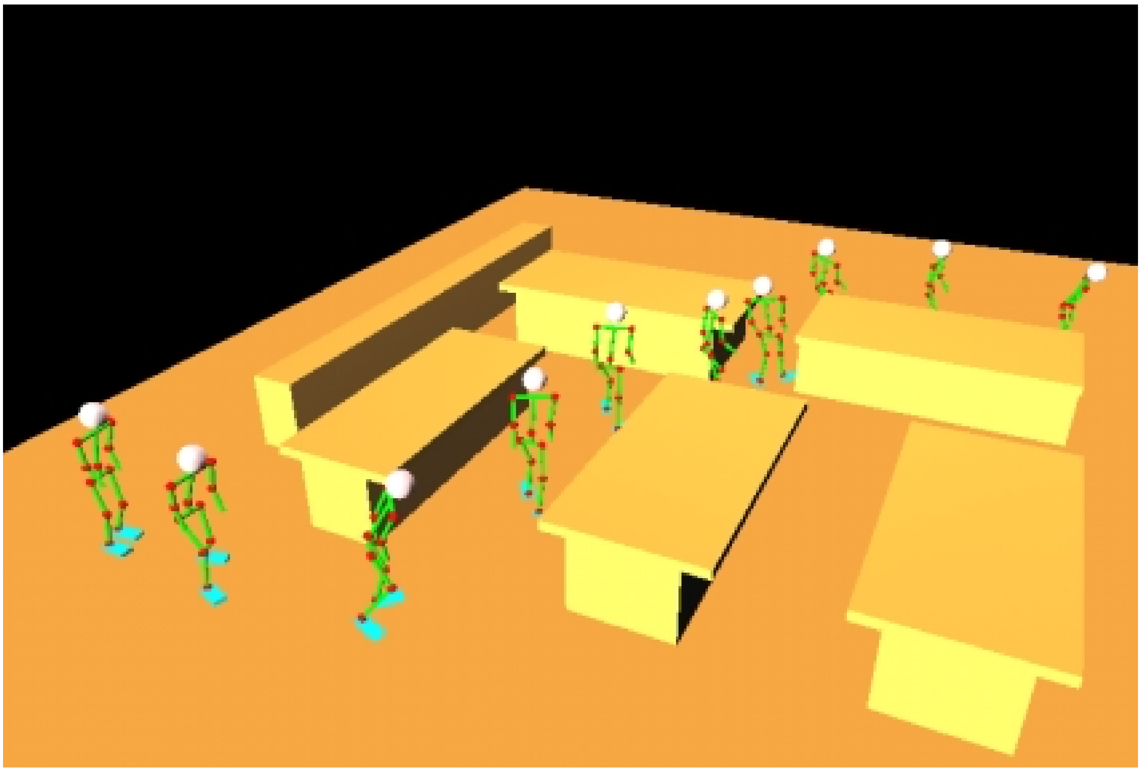

Figure 8: Filtered motion pattern obtained through the dynamics filter

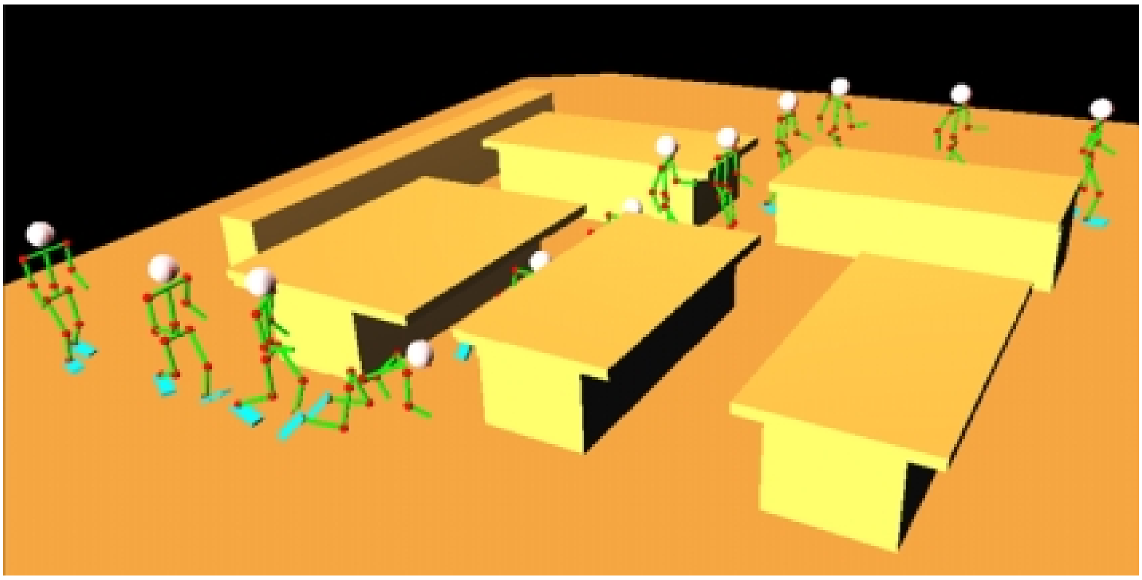

Figure 10: Postures attached to the trajectory with crawl