Footstep Planning for the Honda ASIMO Humanoid

Joel Chestnutt, Manfred Lau, German Cheung, James Kuffner, Jessica Hodgins, and Takeo Kanade

The Robotics Institute
Carnegie Mellon University
5000 Forbes Ave., Pittsburgh, PA, 15213, USA
chestnutt,mlau,gcheung,kuffner,jkh,kanade@cs.cmu.edu

***Abstract*— Despite the recent achievements in stable dynamic walking for many humanoid robots, relatively little navigation autonomy has been achieved. In particular, the ability to autonomously select foot placement positions to avoid obstacles while walking is an important step towards improved navigation autonomy for humanoids. We present a footstep planner for the Honda ASIMO humanoid robot that plans a sequence of footstep positions to navigate toward a goal location while avoiding obstacles. The possible future foot placement positions are dependent on the current state of the robot. Using a finite set of state-dependent actions, we use an A* search to compute optimal sequences of footstep locations up to a time-limited planning horizon. We present experimental results demonstrating the robot navigating through both static and dynamic known environments that include obstacles moving on predictable trajectories.**

***Index Terms*— Humanoid robots, footstep planning, biped locomotion, obstacle avoidance**

I. Introduction

The design of algorithms to compute robust goal-directed navigation strategies for biped humanoid robots is an important area of research. For complex indoor environments designed for humans, this includes dealing with furniture, walls, stairs, doors, and previously unknown obstacles on the floor. For outdoor environments, this includes the ability to navigate on rough terrain and uneven surfaces. Because legged robots have the ability to step over and onto obstacles in their path, they are uniquely suited to overcoming these difficulties. However, existing navigation planning methods fail to consider these additional capabilities, because they were primarily designed for wheeled mobile robots.

We approach the problem of computing global navigation strategies for biped humanoids as one involving an iterated discrete search over a set of valid foot placements. The result of the computation is a sequence of footstep placements that reach a goal region while minimizing encoded heuristics for effort, risk, or the number and complexity of the steps taken. As with other large search domains, computing true optimal solutions for biped navigation is computationally intractable. The challenge then becomes to exploit problem structure and design efficient cost metrics and heuristics that improve search performance. For systems with sensing uncertainty and modeling errors in addition to obstacles in dynamic environments, footstep planning algorithms must be fast enough for real-time replanning. Our approach uses a limited-horizon search that computes the best partial footstep path it can find in the time allowed. The current implementation allows us to compute footstep plans of up to 20-30 steps in one second on typical PC hardware.

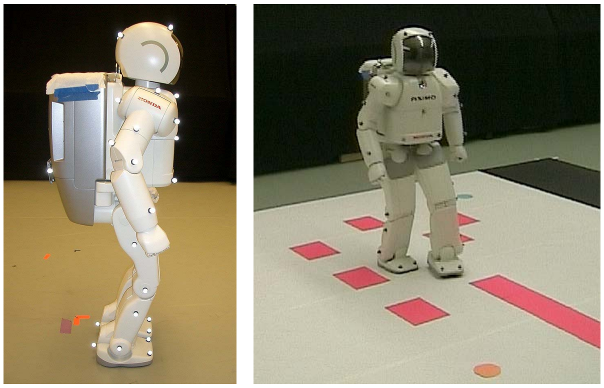

Fig. 1. The Honda ASIMO Humanoid robot at Carnegie Mellon. Side view of the 120 cm tall robot *(left)*; ASIMO navigating using footstep planning to avoid obstacles on the floor *(right)*.

The rest of the paper is organized as follows: Section II gives an overview of related research, Section III describes our biped stepping model and the planning algorithm, Section IV analyzes results from an online implementation using the Honda ASIMO humanoid robot (Figure 1) traversing known environments, and Section V concludes with a summary and directions for future research.

II. Background

Global path planning and obstacle avoidance strategies for mobile robots and manipulators have a large and extensive history in the robotics literature (e.g. see [1], [2] for an overview of early work). Global navigation strategies for mobile robots can usually be obtained by searching for a collision-free path in a 2D environment. Because of the low-dimensionality of the search space, very efficient and complete (or resolution-complete) algorithms can be employed [3]. These techniques can be partially applied to biped humanoid robots. Conservative global navigation strategies can be obtained by choosing an appropriate bounding volume (e.g. a cylinder), and designing locomotion gaits for following navigation trajectories computed by a 2D path planner [4], [5]. However, this always forces the robot to circumvent obstacles. In contrast, legged robots such as biped humanoids have the unique ability to traverse obstacles by stepping over or onto them.

Since reliable, walking biped robots have been developed only recently, much less research attention has been focused on developing complete global navigation strategies for biped robots. Most research has focused on pre-generating stable walking trajectories (e.g. [6]–[8]), or on dynamic balance and control (e.g. [9], [10]). Recently, techniques have been developed to generate stable walking trajectories online [11], [12], though these results do not account for obstacles. For quadruped robots, adaptive gait generation and control on irregular terrain and among obstacles has been previously studied [13]. This method has not yet been applied to biped robots. Sensor-based obstacle-avoidance techniques have been developed for bipeds navigating in unknown environments [14], [15]. However, such reactive methods can become trapped in local loops or dead-ends, because they do not consider global information. In biomechanics, researchers have studied the problem of how humans perform local planning over irregular terrain based on visual feedback [16], [17]. Other related techniques in computer animation that use footprint placement for motion specification have been developed for bipeds [18], [19], and quadrupeds [20], [21]. Large datasets of captured human motion are now available, and techniques have been developed to synthesize animations based on artist specified goals and automatically computed motion transitions [22].

Early work on footstep planning for bipeds involved generating statically stable stepping sequences for a simulated humanoid in cluttered environments [23]. This approach was extended to handle complex height maps, uneven terrain, and allow stepping motions that are not statically stable [24]. Another planner that also uses heightmaps, but generates stepping motions for a simulated world is described in [25]. For our planner, the most complex examples still run only in simulation, but we have created a simplified online version of the planner that is fast enough for continuous replanning on the H7 humanoid robot platform [24], [26]. For the experiments in this paper, we used Honda's ASIMO humanoid robot [27], and developed a state-dependent footstep action mapping to compute footstep plans in time-varying environments with predictable moving obstacles.

In the path planning literature, related approaches using classical AI search or dynamic programming (see [28], [29]) have been applied to finding collision-free motions. Some examples include car-like robots [30], kinodynamic planning [31], calculating sequences of pushing motions [32], planning for constrained systems [33], optimal control of manipulators [34], [35], and a variety of planning problems in a game-theoretic framework [36]. Ultimately, planning problems in robotics can be cast as discrete heuristic search problems involving cost metrics, sets of candidate actions, and goal regions.

III. Footstep Planning

For a biped robot, given a start location, a goal location, and an obstacle-filled environment in which footsteps must be chosen carefully, we wish to find an optimal sequence of actions that causes the robot to walk to the goal location, while not colliding with obstacles, or stepping into any unsafe or unstable locations. Legged robots have the unique ability to step onto or over obstacles or unsafe footholds, which can allow them to traverse terrains that would be impassable to a wheeled mobile robot.

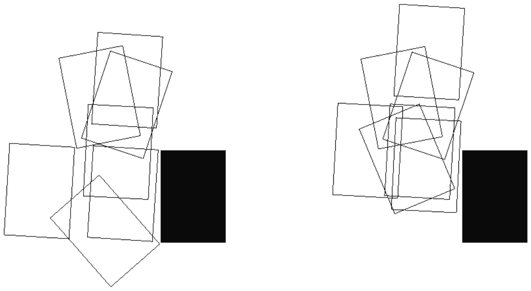

Fig. 2. State dependence of the effects of the actions. The black blocks represent the location of the right foot, and the rectangles are the resulting locations of the left foot from each commanded action. The commands given to the robot are not shown. *Left:* Effects of commanded actions from standing still. *Right:* Effects of commanded actions from full speed walking.

To solve this problem, we utilize a high-level planner that ignores as much of the underlying details of leg movement and trajectory generation as possible, and works instead from a description of the robot's capabilities. It plans at the level of footsteps, and returns a sequence of footholds that the robot can reach carrying it from the initial to the goal location.

A. Biped Model

In our previous work [23], [24], [26], we could describe the capabilities of the robot purely by the locations the swing foot could reach relative to the stance foot. We could provide a list of footstep locations to the controller running on the robot H7, and it would compute a walking trajectory to step on those footholds. With the level of control we have been provided for ASIMO, we cannot specify foot location directly. Instead, we specify a desired displacement of the body. As part of ASIMO's balance control, the actual footsteps the robot takes for a given command varies based on the state of the robot. For example, if told to stop while walking quickly forward, it will not stop immediately, but take one more small step forward in slowing down. Because our commanded actions may be modified by the state of the robot, our planner must take the state of the robot into account while planning a sequence of actions. Figure 2 shows an example of how the effects of actions are modified by the robot's current state.

Let $\mathcal{S}$ be the set of all possible states of the robot, including its joint positions, velocities, and world position and velocity. Without this state dependence, the only information the footstep planner needs is a projection of the state, $\mathcal{L}$, the set of stance foot locations. Let $\mathcal{A}$ be the set of possible actions that can be commanded to the robot. To determine the result of state-dependent actions, the full state of the robot, $s \in \mathcal{S}$, is needed, as well as the environment, $e \in \mathcal{E}$, where $\mathcal{E}$ is the set of environments. In addition, the planner needs the mapping $T : \mathcal{S} \times \mathcal{A} \times \mathcal{E} \rightarrow \mathcal{S}$. With this mapping, the planner can string sequences of actions together, and know what the resulting state will be after each action.

For ASIMO, we do not have access to the full state information. However, we know the history of commands we have given to the robot. By measuring the robot's response to sequences of commands, we found that the state of the robot at step i can be sufficiently described for this mapping by the current stance foot location, $l_i \in \mathcal{L}$, and the last two commanded actions, $a_{i-2}, a_{i-1} \in \mathcal{A}$. Therefore, we can construct an alternate version of T to take us from step i to step $i+1$, using the information available to us (commanded actions) to find the information the planner needs (footstep locations): $T_{ASIMO} : \mathcal{A} \times \mathcal{A} \times \mathcal{L} \times \mathcal{E} \times \mathcal{A} \rightarrow \mathcal{L}$.

$$T_{ASIMO}(a_{i-2}, a_{i-1}, l_i, e_{i+1}, a_i) = l_{i+1}$$

We created this mapping by having ASIMO perform sequences of commands, and recording the resulting motion. The motion was captured using the Vicon[1] optical system. Twelve cameras were used, and each one captures data with a frequency of 120 Hz and a resolution of 1000 x 1000. Six markers were placed on the robot's feet. From the positions of these markers, we determined the positions and orientations of both the left and right feet. We can then compute the relative displacements of the feet from this data. We chose a set of seven actions for each foot, and with the displacements dependent on the previous two actions, we captured 343 sequences of commands to cover all possibilities. Parsing this data gave us T_{ASIMO} for the seven selected actions. This information can then be used as a lookup table during planning.

An additional dependency on the state and the environment can be the set of available actions, $\mathcal{A}$. Certain actions, such as making a step with a specific foot or operating a light switch, may only be applicable in specific states of the robot and specific conditions in the environment. To resolve this dependency, the planner needs the mapping $F : \mathcal{S} \times \mathcal{E} \rightarrow \mathcal{A}$ to compute the available action set.

B. Environment Model

The environment is represented by a grid of cells. Each cell c is represented by $(x, y, i) \in \Re^2 \times \{0, 1\}$, where (x, y) is its location in the grid, and i an information value about whether the terrain is valid or not. Together, the cells create a bitmap representing the free spaces and obstacles of the terrain the planner must overcome. This map of the terrain can be generated from sensor data or built from models of the environment. For this work we did not include height data in the terrain representation. The level of control we have for commanding ASIMO does not include modifying the swing leg trajectory, so we are limited to flat obstacles, or negative obstacles (holes in the floor).

Algorithm 1: PLANPATH($s_{init}, s_{goal}, F, T, E$)

```
    //Initialize search (state, cost, expected, parent);
Q.Insert($s_{init}$, 0, 0, NULL);
while $running\_time < t_{max}$ do
    $s_{best} \leftarrow$ Q.ExtractMin();
    if $GoalReached(s_{best}, s_{goal})$ then
        return $s_{best}$;
    end
    $e \leftarrow E(s_{best}.time)$;
    $\mathcal{A} \leftarrow F(s_{best}, e)$;
    foreach $a \in \mathcal{A}$ do
        $s_{next} \leftarrow T(s_{best}, a, e)$;
        $c_l \leftarrow$ LocationCost($e, s_{next}$);
        $c_s \leftarrow$ StepCost($e, a$);
        $c_e \leftarrow$ ExpectedCost($e, s_{next}$);
        Q.Insert($s_{next}$, $s_{best}.cost + c_l + c_s, c_e$, $s_{best}$);
    end
end
```

For dynamic environments, the information value, i, may vary with time. A moving obstacle will differ in the cells it makes impassable at different times.

C. Planning Algorithm

1) Overview: The planner takes as input an environment, an initial and goal state $s_{init}$and s_{goal}, a mapping F to find possible actions that may be taken, and a mapping T to calculate action effects. If a path is found the planner returns the solution as an ordered list of the actions that should be commanded to reach the goal.

For ASIMO, F returns a set of leg-specific actions based on which leg was the stance leg. T provides the transform from the stance foot location to the next stance location, based on the motion capture data.

2) A Search:* The search through sequences of actions is performed by an A* search, shown in Algorithm 1. The planner continues searching the space until it finds a path to the goal or exceeds a specified time. Three cost functions are used to compute the cost of each node to be inserted into the queue. The first cost is the location cost, which evaluates the footstep location with regard to the environment to determine if that location is a safe place to step. The second cost is the step cost, which calculates the cost to the robot to make the desired step. This weighting is used to prefer certain actions over others. The final cost is the estimated remaining cost-to-go. This cost is calculated by planning backwards from the goal with a standard mobile robot planner as a precomputation step. As the planner explores outward from the goal, it records the cost to reach each location. This cost provides an informed estimate of the actual cost, and is useful for avoiding local minimums. However, it is not an admissible heuristic, as it can vastly overestimate the cost in some situations [37]. This choice means that we sacrifice optimality guarantees for execution speed when using this heuristic.

[1] Vicon is a trademark of Vicon Motion Systems, Ltd.

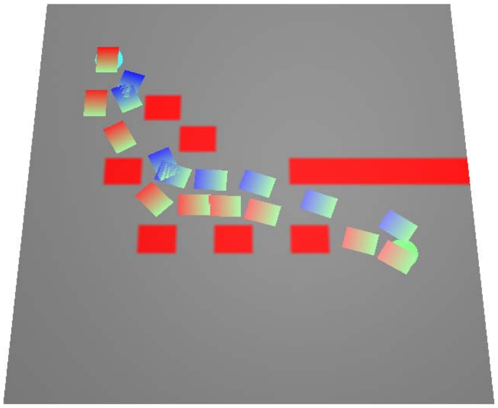

Fig. 3. Plan generated for a static environment. The start location is in the upper left. The solid red regions are obstacles. Blue footprints correspond to footprints for the left foot, while reddish footprints correspond to those of the right foot.

3) Predictable Dynamic Environments: If the environment will change in predictable ways, these changes can be planned for with surprisingly few modifications to the original algorithm. Time must be included in the state of the robot, $s \in \mathcal{S}$, and updated with the time required to execute various actions. This requirement means that the mapping T must correctly handle the time component of the state when mapping to a new state. To extend T_{ASIMO}, we were able to extract the time required for each action for each state from the same data that gave us the original mapping. Finally, instead of taking an environment as input, the planner takes the function $E : \mathcal{T} \rightarrow \mathcal{E}$ to acquire the environment specific to the desired time (shown in Line 6 of Algorithm 1), where $\mathcal{T}$ is the set of possible times. In our implementation, we constructed a set of cell grids, each with an associated time. During execution, the planner would use the map with the closest time to the desired time. Also, for these environments, a straight Euclidean distance metric was used to estimate the remaining cost, instead of the mobile robot planner.

IV. Experimental Results

A. Experimental Setup

Each of the following experiments were performed on the Honda ASIMO robot [27]. The tests were performed on a hard flat surface in an area three meters on each side. The terrain representation used cells that were 0.025 meters per side. The environments and start and goal locations were provided to the planner, and the robot executed the commands dictated by the planner's result.

B. Static Obstacles

An environment with static obstacles were built on the hard flat surface. Each static obstacle is represented by a colored rectangle. The circles in the environment represent the start and goal locations for the robot. We tested different scenarios where we varied the robot's start and goal positions. In each case, the planner successfully returns a set of commands that allow the robot to move from the start to the goal while avoiding the obstacles. One set of start and goal locations is shown in Figure 3, along with the path returned by the planner. Figure 4 shows an example of the robot executing this path.

Notice that the plans allow the robot to step over the obstacles, utilizing its biped capabilities.

C. Dynamic Obstacles

Figure 5 shows an environment with predictable dynamic obstacles. There are two rows of colored rectangles representing the moving obstacles. Each row of obstacles are attached to strings that are then attached to an electric motor. The strings are also represented in the environment as places upon which the robot cannot step. There are two motors, each of which pulls the strings to one side, thereby moving the obstacles. The two rows of obstacles have regular spacings between the blocks, although the spacing is different for each row. The first row moves at 0.1 meters per second, and the second row moves at 0.18 meters per second.

In this environment, the standard mobile robot planner fails, as the start and goal are in disconnected areas of the environment. The ability of the robot to step over the strings connecting the blocks enables it to reach the goal location.

In our example, the robot successfully steps through the dynamic obstacles. It is interesting to note that when passing through the second row of obstacles, the robot sidesteps to wait for an obstacle to move past before stepping through them.

V. Conclusion

We have presented a footstep planner for the Honda ASIMO humanoid robot that computes metric-optimal sequences of footstep positions to navigate toward a goal location while avoiding both static obstacles and obstacles moving along predictable trajectories. The set of possible future foot placement positions are dependent on the current state of the robot. Using a finite set of state-dependent actions, we use an A* search to compute optimal sequences of footstep locations up to a time-limited planning horizon. We present experimental results demonstrating the robot navigating through both static and dynamic environments that include obstacles moving on predictable trajectories.

Future work includes incorporating visual feedback on the ASIMO to enable real-time replanning in dynamic environments.

Acknowledgments

We thank Honda Motor Corporation for generously providing the ASIMO humanoid robot hardware to Carnegie Mellon University for these experiments. This research was

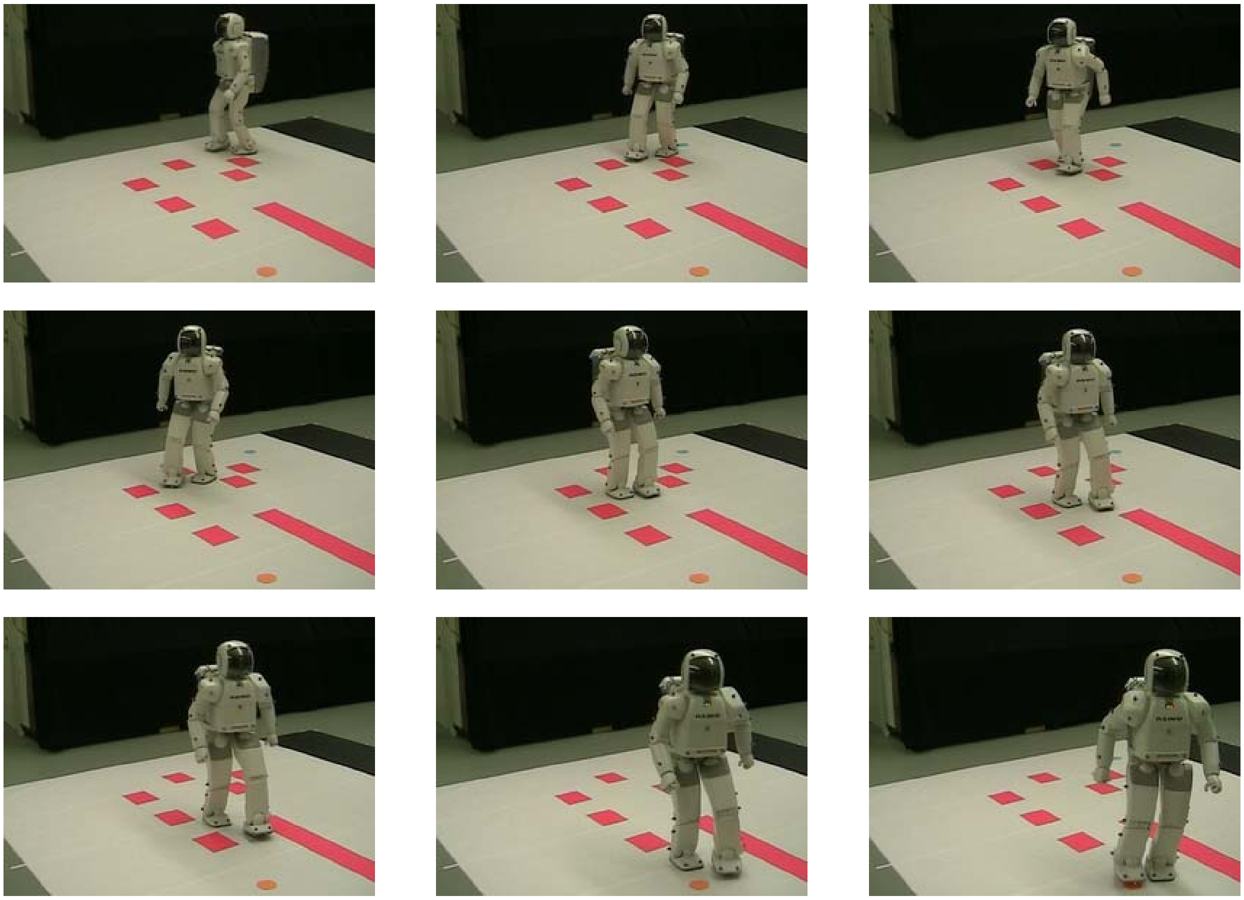

Fig. 4. ASIMO navigating in an environment with static obstacles.

partially supported by NSF grants ECS-0325383, ECS-0326095, and ANI-0224419. Partial support for the 3rd author was provided by NSF CNS-0203912.

References

[1] J. C. Latombe, *Robot Motion Planning*. Boston, MA: Kluwer Academic Publishers, 1991.

[2] Y. K. Hwang and N. Ahuja, "A potential field approach to path planning," *IEEE Trans. Robot. & Autom.*, vol. 8, no. 1, pp. 23–32, Feb. 1992.

[3] A. Stentz, "Optimal and efficient path planning for partially-known environments," in *Proc. IEEE Int'l Conf. on Robotics and Automation (ICRA'94)*, 1994, pp. 3310–3317.

[4] J. Kuffner, "Goal-directed navigation for animated characters using real-time path planning and control," in *Proc. CAPTECH '98 : Workshop on Modelling and Motion Capture Techniques for Virtual Environments*, 1998, pp. 171–186.

[5] J. Pettre, J.-P. Laumond, and T. Simeon, "A 2-stages locomotion planner for digital actors," in *Proc. SIGGRAPH Symp. on Computer Animation*, 2003.

[6] K. Hirai, M. Hirose, Y. Haikawa, and T. Takenaka, "The development of honda humanoid robot," in *Proc. IEEE Int'l Conf. on Robotics and Automation (ICRA'98)*, May 1998, pp. 1321–1326.

[7] J. Yamaguchi, S. Inoue, D. Nishino, and A. Takanishi, "Development of a bipedal humanoid robot having antagonistic driven joints and three dof trunk," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS'98)*, 1998, pp. 96–101.

[8] K. Nagasaka, M. Inaba, and H. Inoue, "Walking pattern generation for a humanoid robot based on optimal gradient method," in *Proc. IEEE Int. Conf. on Systems, Man, and Cybernetics*, 1999.

[9] M. Vukobratovic, B. Borovac, D. Surla, and D. Stokic, *Biped Locomotion: Dynamics, Stability, Control, and Applications*. Berlin: Springer-Verlag, 1990.

[10] J. Pratt and G. Pratt, "Exploiting natural dynamics in the control of a 3d bipedal walking simulation," in *In Proc. of Int. Conf. on Climbing and Walking Robots (CLAWAR99)*, Sept. 1999.

[11] K. Nishiwaki, T. Sugihara, S. KAGAMI, M. Inaba, and H. Inoue, "Online mixture and connection of basic motions for humanoid walking control by footprint specification," in *Proc. IEEE Int'l Conf. on Robotics and Automation (ICRA'01)*, Seoul, Korea, May 2001.

[12] K. Nishiwaki, S. Kagami, Y. Kuniyoshi, M. Inaba, and H. Inoue, "Online generation of humanoid walking motion based on a fast generation method of motion pattern that follows desired zmp," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS'02)*, 2002, pp. 96–101.

[13] S. Hirose, "A study of design and control of a quadruped walking vehicle," *Int. J. Robotics Research.*, vol. 3, no. 2, pp. 113–133, Summer 1984.

[14] M. Yagi and V. Lumelsky, "Biped robot locomotion in scenes with unknown obstacles," in *Proc. IEEE Int'l Conf. on Robotics and Automation (ICRA'99)*, Detroit, MI, May 1999, pp. 375–380.

[15] O. Lorch, J. Denk, J. F. Seara, M. Buss, F. Freyberger, and G. Schmidt, "ViGWaM - an emulation environment for a vision guided virtual walking machine," in *Proc. IEEE Int. Conf. on Humanoid Robotics (Humanoids 2000)*, 2000.

[16] A. Patla, A. Adkin, C. Martin, R. Holden, and S. Prentice, "Characteristics of voluntary visual sampling of the environment for safe locomotion over different terrains," *Exp. Brain Res.*, vol. 112, pp. 513–522, 1996.

[17] A. Patla, E. Niechwiej, and L. Santos, "Local path planning during human locomotion over irregular terrain," in *Proc. AMAM2000*, 2000.

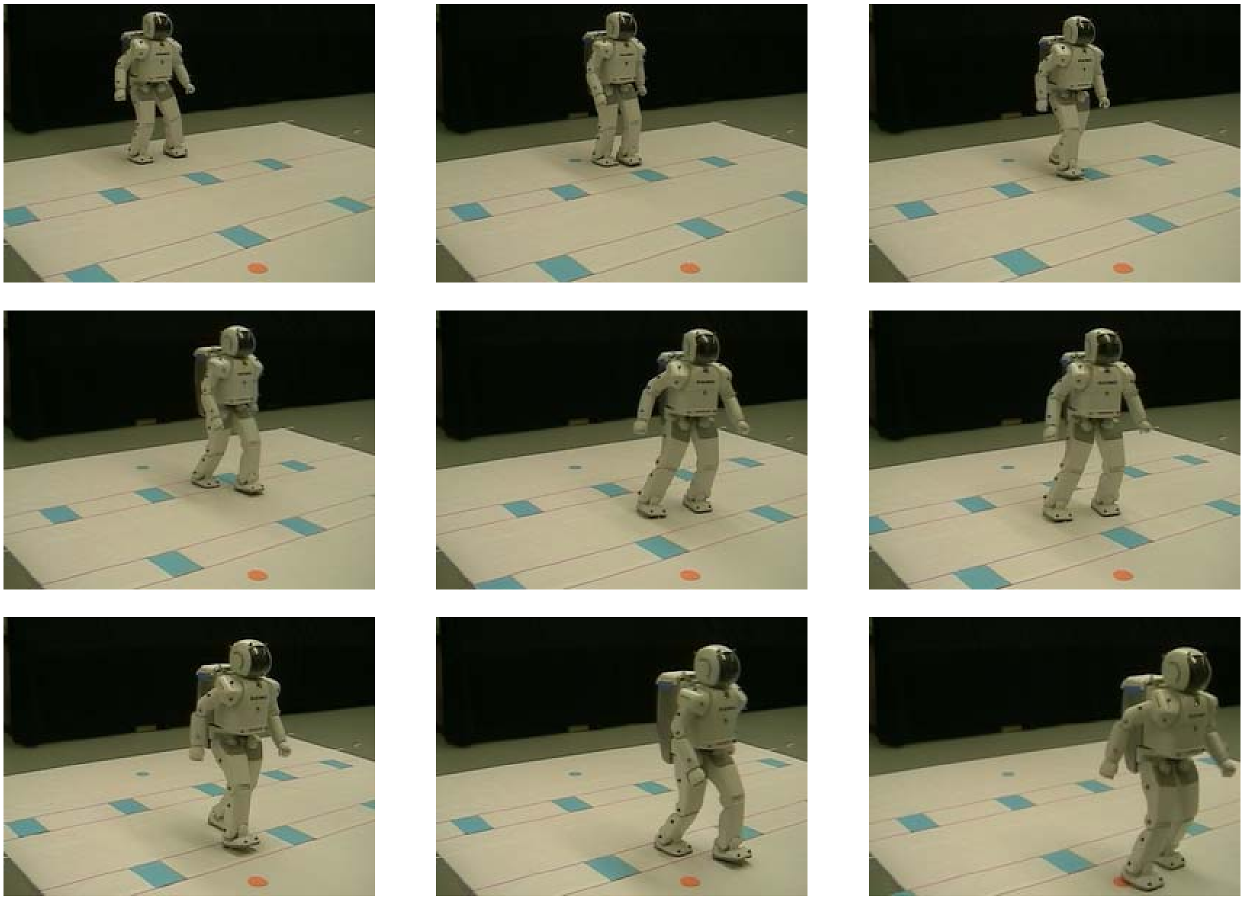

Fig. 5. Example with dynamic obstacles. The obstacles on the floor are moving to the camera's right at differing speeds, forcing the robot to carefully time its steps through the environment.

[18] M. Girard, "Interactive design of computer-animated legged animal motion," *IEEE Computer Graphics & Applications*, vol. 7, no. 6, pp. 39–51, June 1987.

[19] M. van de Panne, "From footprints to animation," in *Proc. Computer Graphics Forum*, vol. 16, no. 4, Oct. 1997, pp. 211–223.

[20] E. Kokkevis, D. Metaxas, and N. I. Badler, "Autonomous animation and control of four-legged animals," in *Proc. Graphics Interface*, May 1995, pp. 10–17, iSBN 0-9695338-4-5.

[21] N. Torkos and M. van de Panne, "Footprint-based quadruped motion synthesis," in *Proc. Graphics Interface*, 1998, pp. 151–160.

[22] J. Lee, J. Chai, P. S. A. Reitsma, J. K. Hodgins, and N. S. Pollard, "Interactive control of avatars animated with human motion data," *ACM Trans. on Graphics*, vol. 21, no. 3, pp. 491–500, 2002.

[23] J. Kuffner, K. Nishiwaki, S. Kagami, M. Inaba, and H. Inoue, "Footstep planning among obstacles for biped robots," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS'01)*, 2001, pp. 500–505.

[24] J. Chestnutt, J. Kuffner, K. Nishiwaki, and S. Kagami, "Planning biped navigation strategies in complex environments," in *Proc. IEEE-RAS/RSJ Int. Conf. on Humanoid Robots (Humanoids'03)*, Munich, Germany, Oct. 2003.

[25] T.-Y. Li, P.-F. Chen, and P.-Z. Huang, "Motion planning for humanoid walking in a layered environment," in *Proc. IEEE Int. Conf. on Robotics and Automationd Systems (ICRA'03)*, 2003.

[26] J. Kuffner, K. Nishiwaki, S. Kagami, Y. Kuniyoshi, M. Inaba, and H. Inoue, "Online footstep planning for humanoid robots," in *Proc. IEEE Int'l Conf. on Robotics and Automation (ICRA'2003)*, 2003.

[27] Y. Sakagami, R. Watanabe, C. Aoyama, S. Matsunaga, N. Higaki, and K. Fujimura, "The intelligent asimo: System overview and integration," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS'02)*, 2002, pp. 2478–2483.

[28] R. E. Bellman, *Dynamic Programming*. Princeton, New Jersey: Princeton University Press, 1957.

[29] R. Korf, "Artificial intelligence search algorithms," in *CRC Handbook of Algorithms and Theory of Computation*, M. Atallah, Ed. Boca Raton, FL: CRC Press, 1998, pp. 36.1–20.

[30] J. Barraquand and J. Latombe, "Nonholonomic multibody mobile robots: Controllability and motion planning in the presence of obstacles," *Algorithmica*, vol. 10, pp. 121–155, 1993.

[31] B. Donald, P. Xavier, J. Canny, and J. Reif, "Kinodynamic motion planning," *Journal of the ACM*, vol. 40, no. 5, pp. 1048–1066, Nov. 1993.

[32] K. M. Lynch and M. T. Mason, "Stable pushing: Mechanics, controllability, and planning," *Int. J. Robotics Research.*, vol. 15, no. 6, pp. 533–556, Dec. 1996.

[33] J. Barraquand and P. Ferbach, "Path planning through variational dynamic programming," Digital - PRL Research Report, Tech. Rep. 33, sep 1993.

[34] J. Bobrow, S. Dubowsky, and J. Gibson, "Time-optimal control of robotic manipulators," *Int. Journal of Robotics Research*, vol. 4, no. 3, 1985.

[35] Z. Shiller and S. Dubowsky, "On computing time-optimal motions of robotic manipulators in the presence of obstacles," *IEEE Trans. on Robotics and Automation*, vol. 7, no. 7, Dec. 1991.

[36] S. M. LaValle, "A game-theoretic framework for robot motion planning," Ph.D. dissertation, University of Illinois, Urbana, IL, July 1995.

[37] J. Chestnutt and J. Kuffner, "A tiered planning strategy for biped navigation," in *Proc. IEEE-RAS/RSJ Int. Conf. on Humanoid Robots (Humanoids'04)*, Santa Monica, California, Nov. 2004.