Online Footstep Planning for Humanoid Robots

James Kuffner[1,2]   Satoshi Kagami[2,3]   Koichi Nishiwaki[4]

Masayuki Inaba[4]   Hirochika Inoue[4]

1   Dept. of Computer Science, Robotics Institute, Carnegie Mellon University
     5000 Forbes Ave., Pittsburgh, PA 15214, USA. kuffner@cs.cmu.edu

2   Digital Human Lab., National Institute of Advanced Industrial Science and Technology
     2-41-6, Aomi, Koto-ku, Tokyo, 135-0064, Japan.

3   CREST, JST (Japan Science and Technology Corporation)

4   School of Information Science and Technology, University of Tokyo.
     7–3–1, Hongo, Bunkyo-ku, Tokyo, 113–8656, Japan.

Abstract

We present an online algorithm for planning sequences of footstep locations that encode goal-directed navigation strategies for humanoid robots. Planning footsteps is more general than most existing navigation methods designed for wheeled robots, since the options of stepping over or upon obstacles in a cluttered terrain are available. Given a discrete set of plausible footstep locations, a forward dynamic programming approach is used to compute a footstep sequence to a specified goal location in the environment. Heuristics designed to minimize the number and complexity of the step motions are used to encode cost functions used for searching a footstep transition graph. If successful, the planner returns an optimal sequence of footstep locations according to the cost functions and plausible sets of footstep locations defined. We show results from an experimental implementation of the algorithm running on the H7 humanoid robot. Using a stereo vision system to sense obstacles in the immediate environment and identify a target goal location, the robot updates the current optimal footstep sequence to the goal from its present location.

1   Introduction

As humanoid robotics hardware technology continues to progress, there will be a need for software and algorithms useful to improving the usability and autonomy of these complex machines. One important area of need will be software for safe, autonomous navigation in human environments such as homes and offices.

Research on global path planning and navigation strategies for mobile robots has a large and extensive history in the robotics literature. Since the problem can usually be modeled as searching for a collision-free path in a 2D environment, very efficient and complete algorithms can be employed (for an overview, see [7]). However, most of these techniques apply to wheeled robots which must always circumvent obstacles. In contrast, legged robots (in-
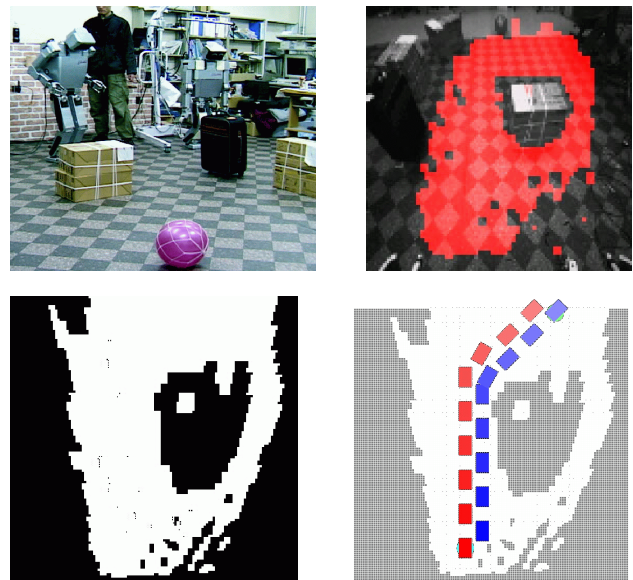

Figure 1: Online footstep planning: (*top row*: H7 humanoid and candidate walking surface area extracted from 3D stereo depth map; *bottom row*: distortion-corrected obstacle map after thresholding, and calculated optimal footstep sequence to target.

cluding biped humanoids) have the unique ability to traverse obstacles by stepping over or upon them. This opens up new avenues and possibilities for strategic navigation planning.

2   Related Research

Patla and colleagues have studied the problem of how humans perform local planning over irregular terrain based on visual feedback[14, 13, 15]. For humanoid robots, most existing research has focused on pre-generating stable walking trajectories (e.g. [2, 21, 9]), or on dynamic
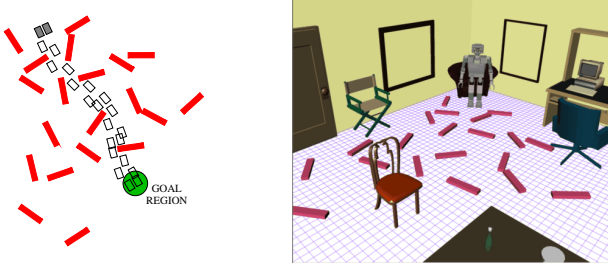
Figure 2: Planned statically-stable footstep sequence for the humanoid H6 navigating in a simulated cluttered office.

Figure 3: Valid footstep region for the right foot of H7 while supported by the left foot (*continuous* and *discrete samples*).

balance and control (e.g. [19, 16]). Recently, techniques have been developed to generate stable walking trajectories online [12, 11], 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 [3]. This method has not yet been applied to biped robots. Sensor-based obstacle-avoidance techniques have been developed for bipeds navigating in unknown environments [20, 8]. However, such reactive methods can become trapped in local loops or dead-ends, since they do not consider global information. Other related techniques in computer animation that use footprint placement for motion specification have been developed for bipeds[1, 18], and quadrupeds[5, 17]. We previously implemented a global planning approach for bipeds that generated statically-stable walking motions for a simulated H6 robot in cluttered environments [6] (see Figure 2). In this paper, we allow stepping motions that are not statically-stable, but also simplify the robot walking model and obstacle collision checking to reduce the computational cost. Our current implementation is fast enough to enable continuous replanning during each step cycle, and has been run successfully on a real robot.

Our approach is to build a search tree from a discrete set of feasible footstep locations corresponding to candidate stable stepping motion trajectories. Using standard dynamic programming techniques, we compute a sequence of footstep placements to reach a goal region in an environment with obstacles. This has the advantage of planning a *global navigation strategy* for bipeds that includes the ability to step over obstacles. Encoded heuristics minimize the number and complexity of the steps taken, as well as guide the search. Given a proper admissable heuristic cost function, the planner can compute *optimal* sequences of footstep locations to reach a goal, according to the cost function defined. This strategy can be computed efficiently on standard PC hardware (usually in a fraction of a second for paths involving less than 20 steps to reach the goal). We have connected a preliminary version of this planner to an online stereo vision system and walking trajectory generator for the humanoid robot H7. The robot continuously updates a *walking area map* computed by the stereo vision system, and replans an optimal footstep sequence to reach the current goal.

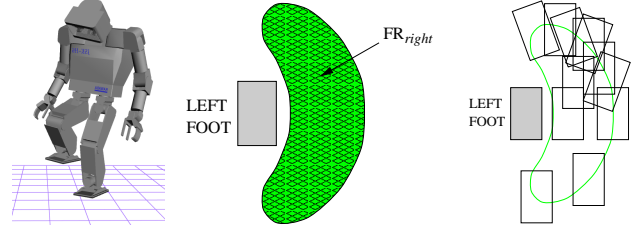The rest of the paper is organized as follows: Section 3

gives an overview of the biped stepping model, Section 4 describes the algorithm, Section 5 shows some results on an experimental implementation using the humanoid H7, and Section 6 concludes with a summary discussion.

3 Biped Navigation Model

Although the search technique adopted is very general, we currently make the following simplifying assumptions:

1. The environment floor is flat and contains non-moving obstacles.

2. A discrete set of feasible footstep placement positions and associated stepping motions are pre-computed.

3. Only the floor surface is currently allowed for foot placement (not obstacle surfaces).

Stable Footstep Locations: The biped model comes with a pre-determined set of feasible footstep locations for each foot. For example, Figure 3 shows the continuous, feasible footstep range FR_{right} for the right foot while balanced on the left foot, and an example discrete set of foot placements. The placements for the left foot simply mirror the right foot placements, which is possible for symmetric bipeds.

In selecting which footstep placements to include in the discrete set used during the search, we chose a distribution of placements along the edge of the reachable region at different relative foot angles, as well as a few interior placements to allow the robot to maneuver in tight areas. This choice is currently only heuristic, and attempts to strike a balance between maximizing the navigation options, while minimizing the total number of discrete placements. In our implementation, we selected a total of 19 placements for each foot (see Section 5).

The number of placements is particularly important, as it determines the branching factor of the search tree. Some relative rotation of the foot sole with respect to the support leg is necessary in order for the robot to alter its global orientation. A few backward foot placements are used to provide additional maneuverability for navigating environments with areas of limited free space. We note that alternatively, one could define multi-step "turn-in-place" motions to explicitly allow for larger changes in orientation at each discrete search step. This has not yet been implemented in our system.
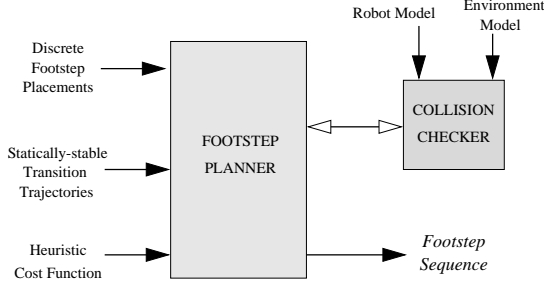
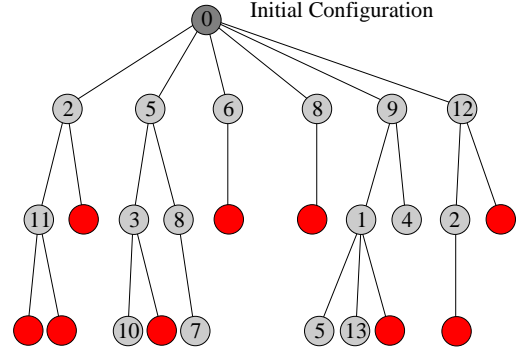Figure 4: Block diagram of the planning algorithm.

Figure 5: Search tree rooted at the initial configuration. Successor states that resulted in collisions (dark leaf nodes) are pruned from the search.

After all footstep locations have been defined, we generate dynamically-stable walking trajectories to transition between different footstep placements using the online algoritm presented in [11].

4 Footstep Planning Algorithm

The planner accepts as input a discrete set of robot footprint locations, a walking surface area environment map, and a heuristic cost function. The 2D footprint contour and environment map is used for collision checking. If the planner successfully finds a solution, it outputs a minimal-cost sequence of encoded footstep placements and transitions. An overview of the planner is shown in Figure 4. Each of the main components are described in the following sections.

Dynamic Programming: Given a set of discrete placements, we adopt a *forward dynamic programming* approach to planning navigation strategies. For an overview of the relevant techniques used here, the reader is referred to any textbook on artificial intelligence or game search trees (e.g. [4] or [10]). Since an exhaustive search is too expensive, we employ A* which uses a greedy (yet admissible) heuristic in order to prune the search tree. If successful, the planner returns an optimal sequence of footstep locations according to the given cost function defined.

Starting from an initial biped configuration Q_{init}, a search tree of possible footstep placements is constructed as in Figure 5. The planner maintains a *priority queue* of search nodes containing a footprint placement configuration and a heuristic *cost value* (see Section 4). The nodes are inserted into the queue based on their cost.

Initially, the queue is initialized with a single node N_{init}, which contains the starting configuration Q_{init}. At each iteration, the planner removes the lowest-cost (higher priority) node N_{min} from the priority queue. Successor nodes of N_{min} are generated that correspond to all potential placement locations for the next footstep. Collision checking is used to eliminate successor nodes which result in obstacle collisions (see Section 4).

When a successor node is generated that falls within the predefined goal region, the search terminates and a solution path back to the root node of the tree is returned. The solution path defines a discrete sequence of footprint place-

ments which when executed will take the robot from Q_{init} to the goal region.

The search can fail in one of two ways:

- No more valid successor nodes can be generated. In this case, no collision-free footstep sequence exists using the given discrete set of relative footprint placements.

- The running time of the search exceeded the maximum allowable or the size of the search tree exceeded a pre-determined limit on the maximum number of nodes. In this case, the planner fails to reach the goal, but the lowest cost path computed so far can be returned.

Estimated Cost Heuristic: We encode a simple greedy heuristic by defining a cost function $\mathcal{L}(Q)$ which is used for maintaining the priority queue. $\mathcal{L}(Q)$ combines the cost of the path taken so far along with an estimate of the cost to reach the goal region.

$$\mathcal{L}(Q) = w_d D(N_Q) + w_\rho \rho(N_Q) + w_g \mathcal{X}(Q, Q_g)$$

The first two terms define the cost of the path to configuration Q from Q_{init}. $D(N_Q)$ is the depth of the node N_Q in the tree. $\rho(N_Q)$ is a function that encodes a small penalty for path sequences that include steps which incur orientation changes or backward steps. These terms have the combined effect of favoring paths with fewer steps, as well as slightly favoring paths with long sequences of forward, straight-line steps.

The final term represents an estimated cost from the current configuration to the goal region. $\mathcal{X}(Q, Q_g)$ approximates the minimum number of steps needed to traverse the straight-line distance between the footprint location at Q, and a footprint in the center of the goal region Q_g.

Each of the terms are weighted relative to each other by the factors w_d, w_ρ, and w_g. These values were determined experimentally, and are listed in Section 5. More research
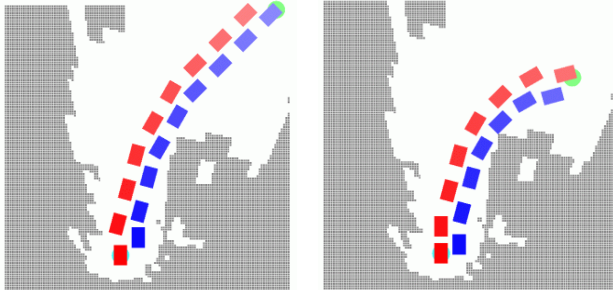
Figure 6: Planned footstep sequences for two different goal locations using the same obstacle map.

Figure 7: Vision Processing steps. *(top row:)* raw camera images and resulting 3D Depthmap; *(bottom row:)* mesh model, planar surface identification, and final walking area map.

is needed to evaluate the effectiveness of different possible values for weighting parameters or alternative heuristic functions (see Section 6).

Obstacle Collision-checking: We require some kind of collision-checking algorithm to test the feasibility of potential footstep placement positions. If a node results in a footstep placement which causes a collision between the robot and an obstacle, that node is pruned from the search tree.

In our implementation, we used a simplified collision-checking strategy based on a simple 2D polygon-polygon intersection test between the outline of the obstacle projection to the walking surface, and the outline of the foot geometry at the attempted footstep placement position. If the polygons intersect, the node is discarded. If not, then an approximate method is used to verify that any candidate free-leg trajectory is free of obstacles. Although not fully utilized in our system, minimum distance information allows us to enforce conservative safety bounds on how close the robot geometry is allowed to approach obstacle surfaces. Stepping motions which result in the robot grazing too close to an obstacle surface can be either pruned from the search tree, or assigned a higher cost.

We should note the special case of obstacles which have a "negative" height (e.g. holes in the floor, or other gaps in the walking surface). Furthermore, since the collision checking is a rather expensive part of the planning process, we adopt a *lazy-evaluation* approach to collision-checking. Instead of testing all successor nodes *prior* to insertion in the queue, we simply insert all successors and perform the minimum distance calculations *after* a node is removed from the priority queue. If there is a collision, we simply discard the node and extract the next node in the queue.

5 Experiments

This section presents some preliminary results using a prototype planner running onine on the humanoid robot H7. Figure 7 shows the output of the 3D vision system.

Figure 8 shows a calculated footstep sequence for a model of the humanoid robot H7 navigating in the laboratory. The sequence is computed to reach a circular goal region. Figure 8 shows snapshots taken while running the
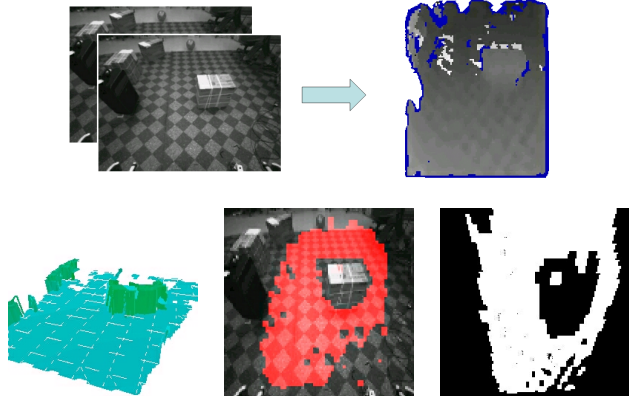
goal following planning pogram. There were a total of 19 discrete foot placements considered for each foot. The search tree contained approximately 1,100 nodes. Considering that the number of nodes required for a brute-force, breadth-first search on a footstep sequence length of 18 steps is more than 10^{21}, this is quite satisfactory. The path was computed in less than 0.5 seconds on an 800 MHz Pentium II running Linux.

The weighting factors used for the cost function were $w_d = 1.0$, $w_\rho = 0.2$, and $w_g = 1.0$. These values were determined experimentally, and offered reasonable results.

6 Discussion

We have presented an online algorithm for planning sequences of footstep locations that encode goal-directed navigation strategies for humanoid robots. Given a discrete set of plausible footstep locations, a forward dynamic programming approach is used to compute footstep sequences to a specified goal location. If successful, the planner returns an optimal sequence of footstep locations according to the cost functions and plausible sets of footstep locations defined. We show results from an experimental implementation of the algorithm running on the H7 humanoid robot. Using a stereo vision system to sense obstacles in the immediate environment and identify a target goal location, the robot updates the current optimal footstep sequence to the goal from its present location.

A number of important improvements can be made to our current system, and these form the basis of our future work: 1) Extending the method to handle environments with uneven terrain; 2) The ability of the robot to step upon the surface of obstacles; 3) Designing and investigating the effects of different heuristics on the efficiency of the search; 4) Including dynamic stepping motions that increase the range of reachable footstep placements (including hopping or jumping between stable states.

Figure 8: Snapshots of the humanoid H7 calculating and executing online the footstep sequence from Figure 1

References

[1] M. Girard. Interactive design of computer-animated legged animal motion. *IEEE Computer Graphics & Applications*, 7(6):39–51, June 1987.

[2] 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)*, pages 1321–1326, May 1998.

[3] S. Hirose. A study of design and control of a quadruped walking vehicle. *Int. J. Robotics Research.*, 3(2):113–133, Summer 1984.

[4] L. Kanal and V. Kumar, editors. *Search in Artificial Intelligence*. Springer-Verlag, New York, 1988.

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

[6] 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.

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

[8] 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.

[9] 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.

[10] N.J. Nilsson. *Principles of Artificial Intelligence*. Tioga, Palo Alto, CA, 1980.

[11] 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.

[12] 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)*, May 2001.

[13] A.E. Patla. How is human gait controlled by vision? *Ecological Psychology*, 10:287–302, 1998.

[14] A.E. 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.*, 112:513–522, 1996.

[15] A.E. Patla, E. Niechwiej, and L. Santos. Local path planning during human locomotion over irregular terrain. 2000.

[16] 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)*, September 1999.

[17] N. Torkos and M. van de Panne. Footprint-based quadruped motion synthesis. In *Proc. Graphics Interface*, pages 151–160, 1998.

[18] M. van de Panne. From footprints to animation. In *Proc. Computer Graphics Forum*, volume 16, pages 211–223, October 1997.

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

[20] 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)*, pages 375–380, Detroit, MI, May 1999.

[21] 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)*, pages 96–101, 1998.