

Vision-Guided Humanoid Footstep Planning for Dynamic Environments

Philipp Michel, Joel Chestnutt, James Kuffner and Takeo Kanade

The Robotics Institute

Carnegie Mellon University

5000 Forbes Ave., Pittsburgh, PA 15213, USA

{pmichel,chestnutt,kuffner,kanade}@cs.cmu.edu

Abstract—Despite the stable walking capabilities of modern biped humanoid robots, their ability to autonomously and safely navigate obstacle-filled, unpredictable environments has so far been limited. We present an approach to autonomous humanoid walking that combines vision-based sensing with a footstep planner, allowing the robot to navigate toward a desired goal position while avoiding obstacles. An environment map including the robot, goal, and obstacle locations is built in real-time from vision. The footstep planner then computes an optimal sequence of footstep locations within a time-limited planning horizon. Footstep plans are reused and only partially recomputed as the environment changes during the walking sequence. In our experiments, combining real-time vision with plan reuse has allowed a Honda ASIMO humanoid robot to autonomously traverse dynamic environments containing unpredictably moving obstacles.

Index Terms—Humanoid Robot, Vision, Footstep Planning, Obstacle Avoidance, Dynamic Replanning.

I. INTRODUCTION

Given the recent advances in stable robotic bipedal walking, navigation autonomy for humanoid robots comprises an increasingly important research area. The ability of legged robots to step over and onto some obstacles makes them ideally suited for environments designed for humans, which invariably contain a wide range of objects and obstacles such as furniture, stairs, doors, uneven ground and, of course, people. Yet, relatively little work has been done to exploit the recent improvements in biped locomotion for the purpose of enabling humanoids to autonomously navigate such environments.

We view the problem of navigating towards a goal in the presence of obstacles as a closed-loop interplay between sensing the environment using vision and planning a goal-directed sequence of footsteps that avoids obstacles. For dynamic surroundings that include unpredictably moving obstacles, it is crucial that any footstep path be quickly replanned if it becomes suddenly blocked by an obstacle. This places time constraints on both the sensing and planning components. Changes in the robot’s surroundings need to be recognized almost instantly, leaving the planner as much time as possible to adjust the current footstep path. We present a vision-based footstep planning system that computes the best partial footstep path within its time-limited search horizon, according to problem-specific cost metrics and heuristics. The computational cost of planning motivates the reuse of as much of

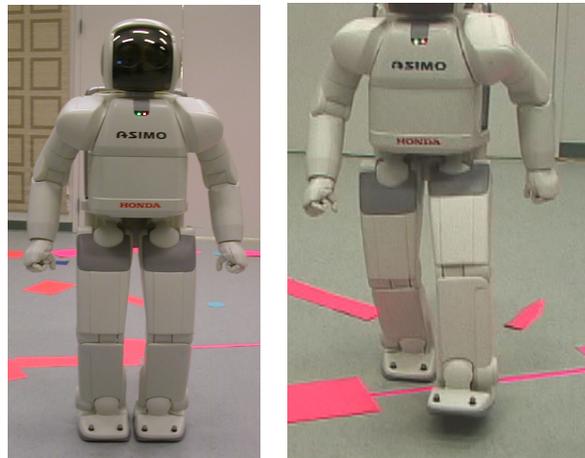


Fig. 1. Carnegie Mellon’s Honda ASIMO Humanoid (left). Stepping over an obstacle during a vision-guided walking sequence (right).

a previous footstep path as possible, instead of starting from scratch each time a step is taken.

The rest of this paper is organized as follows: Section II presents related research. In Section III, we describe how vision is used to build environment representations. The incremental footstep planning process is detailed in Section IV. Section V presents results from our online implementation using the Honda ASIMO humanoid robot [1]. Finally, Section VI concludes with a discussion and pointers to future research.

II. RELATED WORK

Most of the literature in robotic bipedal walking has concentrated on various approaches to reliable, stable gait generation and feedback, while relatively little work has focused on developing global navigation autonomy for biped robots. Emphasis has primarily been on pre-generating walking trajectories [2]–[4], online trajectory generation [5], [6] and dynamic balance [7], without accounting for obstacles.

Obstacle avoidance and local planning based on visual feedback has been studied in humans [8], [9] in order to gain insight into potential robotic implementations. Several reactive perception-based obstacle avoidance techniques for bipeds have been developed [10]–[13], giving robots some

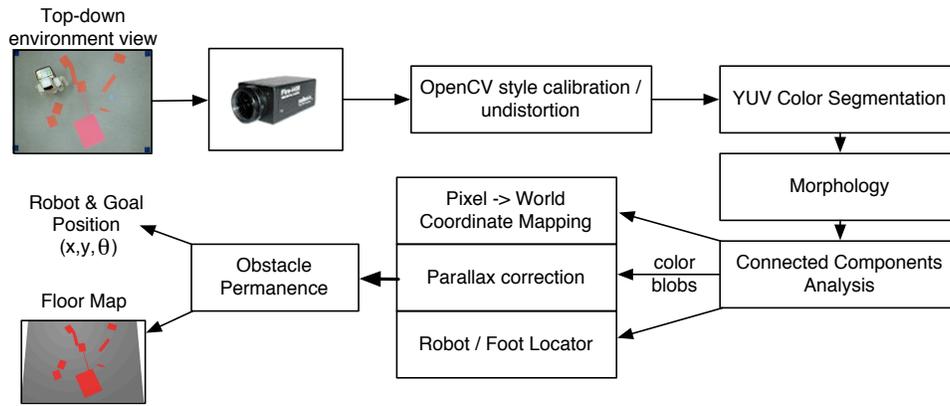


Fig. 2. Overview of sensing. An environment map of obstacles on the floor and the robot/goal location and orientation are computed from ceiling-based vision.

navigation autonomy when faced with an unknown environment by directly modifying the walking trajectory when an obstacle is detected in the robot’s path. However, such reactive approaches do not take into account the global configuration of obstacles between the robot and the desired goal. This makes locomotion in truly cluttered environments challenging and gives rise to the possibility of becoming trapped in local loops and dead ends.

Traditional path planning techniques have been applied to perception-based obstacle avoidance on humanoids [14]. Such methods do not, however, take the ability of a legged robot to step over or onto obstacles into account. Planning at the level of footsteps [15]–[17] allows these capabilities to be directly exploited in order to compute efficient global navigation strategies for humanoid robots.

Efficient and accurate perception plays an important role in any online implementation. Environment mapping and obstacle detection from onboard stereo processing [18], [19] allow three dimensional obstacles to be localized, but stereo implementations often cannot satisfy the delay and accuracy bounds required to have the robot walk near, on, or over obstacles during real-time locomotion. Color-based segmentation [20], [21] is often employed when speed and accuracy are paramount.

III. SENSING THE ENVIRONMENT

The Honda ASIMO humanoid at Carnegie Mellon, shown in Figure 1, operates in a laboratory room containing a number of planar obstacles placed on a hard, flat floor. We employ an overhead camera to compute the position of the robot, the desired goal location, indicated by a marker, and all obstacles. Global sensing is used instead of ASIMO’s onboard cameras due to the limited control we have over head direction during a walking sequence, which makes maintaining a steady gaze at the floor close to robot’s legs difficult. In addition, we found that both onboard vision processing as well as transferring video from the onboard cameras over the robot’s wireless link were infeasible given the time constraints placed on the sensing process by the unpredictably changing environment. All processing described in this section is done in real-time or faster on commodity hardware.

We use an industrial firewire camera mounted 3.5m above the ground, yielding a viewable floor area of about $3.2\text{m} \times 2.4\text{m}$. The camera supplies a 640×480 pixel YUV video stream at 25–30 frames per second. Figure 2 gives an overview of the sensing process. The camera is initially calibrated offline based on the algorithm proposed by Zhang [22], using OpenCV to calculate intrinsic and extrinsic camera parameters as well as computing the radial distortion coefficients. During operation, the incoming camera images can then be rectified on-the-fly.

A. Color Segmentation

We employ colored markers to denote the planar obstacles on the floor (bright pink) and the desired goal location (light blue). Two additional, differently colored markers (yellow and green) attached to the top of ASIMO’s “backpack” are used to identify the robot’s location and orientation. We use four square delimiters (dark blue) to define a rectangular area within which the robot operates.

Color segmentation is performed directly on the YUV stream generated by the camera [20]. This avoids the overhead incurred by a software-based colorspace transformation and allows us to exploit the robustness of the YUV space to color intensity changes arising from variability in environment lighting. In order to segment out obstacles or other markers, color thresholds are first defined by sampling pixel values offline for each marker in a variety of locations on the floor. Simple color thresholding then produces a series of binary masks indicating presence or absence of each marker’s color at every pixel. To eliminate noise, one pass of erosion/dilation using a rectangular structuring element is performed. Finally, a step of connected components labeling is applied to group pixels corresponding to the same marker.

Calculating the moments of each resulting color blob then yields information about its centroid, area, major/minor axes and orientation on the floor.

B. Converting to World Coordinates

Assuming the physical distances between the four delimiters that outline the robot’s walking area are known, scaling can

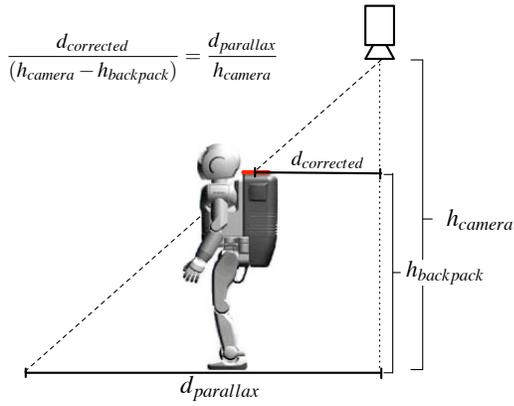


Fig. 3. Parallax correction. A similar triangle relationship links the measured distance $d_{parallax}$ of the robot from the center of the camera image, the backpack height $h_{backpack}$ and the camera height h_{camera} , allowing the floor-projected distance $d_{corrected}$ of the markers to be calculated.

be used to convert between the pixel coordinates of each blob’s centroid and the corresponding real-world distances. Alternatively, knowing the calibrated camera’s distance from the floor also suffices, although we found the delimiter-based approach to be somewhat more accurate. We use a right handed coordinate system centered on the lower left delimiter when viewed from the camera.

The orientation of the robot can be determined from the angle the line connecting the backpack markers forms with the horizontal. To perform footstep planning, the precise location of the robot’s feet is required. This cannot directly be calculated from the raw world coordinates of the the robot markers, as these do not lie on the ground plane and are hence affected by parallax as the robot moves towards the edges of the camera view. Knowing the height of the backpack, the distance of the camera from the ground, and the offset of the markers from the center of the camera image allows us to correct for parallax, effectively by projecting the robot markers onto the ground plane. Figure 3 illustrates. Once parallax has been accounted for, simple offsets from the midpoint between the projected markers can be used to arrive at the location of each foot.

C. Building the Environment Map

A 2D grid of binary valued cells represents the environment. The value in each cell denotes whether the corresponding terrain is valid (obstacle free) or invalid (totally or partially occupied by an obstacle). The environment map is thus a bitmap representation of the free space and obstacles as seen from the ceiling. Figure 4 shows a ceiling view of the environment, the robot and obstacles tracked by the vision system, and the corresponding map generated.

The vision system creates maps with sizes on the order of 300×200 cells for use by the footstep planner. The environment map can be generated by cropping the binary obstacle mask gathered previously from vision at the walking area delimiters and scaling it appropriately (while maintaining

aspect ratio) to fit the map dimensions used by the footstep planner. Alternatively, the map could be constructed by knowing the extents, orientation and centroid (as computed during color segmentation) or some other geometric parameterization of each obstacle. This relaxes the need for matching aspect ratios, but is considerably slower.

The use of non-local, centered sensing gives rise to a problem of obstacle occlusion. Even if the robot is perfectly centered in the camera view (standing directly below the camera), obstacles close to the feet may be occluded by the robot’s own body. The occluded area increases as the robot moves towards the edges of the image. Since obstacles in the occluded area cannot be seen by the camera, they will show up as free space in the environment map used by the footstep planner, causing the robot to potentially step directly onto an obstacle.

We overcome this problem by implementing obstacle permanence in a rectangular area bounding the robot in the image. The cells corresponding to the permanence region in the environment map are not updated from the last value they had before entering the area. This reflects the assumption that whatever obstacles were last seen in the area currently being occluded by the robot are still present now. We have found this rectangular approximation to the shape of the occluded region to work well in our trials.

IV. FOOTSTEP PLANNING

The task of our biped planner is to find as close to an optimal sequence of actions as possible that causes the robot to reach the goal location while avoiding the obstacles in the environment. It should exploit the unique capability of a legged robot to step over obstacles. The planner operates at the level of footsteps, ignoring the lower level details of leg movement and control. The limited level of control we have over the leg trajectories during a footstep limits us to using planar obstacles, as the swing leg will not be able to clear objects higher than a few centimeters.

A. Basic Algorithm

The planner, shown in Algorithm 1, takes as input an environment map E , initial and goal robot states s_{init} , s_{goal} , a mapping F of possible actions that may be taken in each state and an action-effect mapping T . On successfully finding a path to the goal, an ordered sequence of footstep actions to be commanded to the robot is returned.

The planner computes the cost of each candidate footstep location using three cost metrics: (i) a location cost determining whether the candidate location is ‘safe’ in the environment (not part of an obstacle), (ii) a step cost which prefers ‘easy’ stepping actions and (iii) an estimated cost-to-go providing an approximation of the candidate’s proximity to the goal using a standard mobile-robot planner.

An A* search is performed on the possible sequences of actions (walking instructions) that can be commanded to the robot, until an obstacle-avoiding path to the goal is found



Fig. 4. Top-down view of the walking environment (left), vision system output (center), and corresponding environment map with current footstep plan generated (right).

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

```

 $s_{best}$  = VerifyPath( $path, s_{goal}, E$ );
if  $s_{best} \neq NULL$  then return  $s_{best}$ ;
while  $running\_time < t_{max}$  do
   $s_{best} \leftarrow Q.ExtractMin()$ ;
  if GoalReached( $s_{best}, s_{goal}$ ) then return  $s_{best}$ ;
   $e \leftarrow E(s_{best}.time)$ ;
   $\mathcal{A} \leftarrow F(s_{best}, e)$ ;
  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

```

or a specified computation time limit is exceeded. To combine a set of actions into a sequence, a mapping from the commanded action to the resulting state, as well as knowledge of the current environment is required. For the Honda ASIMO, the effect of a commanded action is determined by the state of the robot at the time of execution. Although ASIMO’s complete state information is not directly accessible to us, we found that the current stance foot location and the past two commanded actions are sufficient to accurately describe the current state, thus allowing us to establish the $T(state_i, action_i, environment_i) \rightarrow state_{i+1}$ mapping. The mapping was computed offline by having ASIMO exhaustively perform all possible walking action 3-tuples and recording the resulting robot state using a motion capture system [23].

B. Plan Reuse

At each step, we plan a path towards the goal, but then only have ASIMO execute the first step of this plan before replanning for the next step. To increase our planning horizon within our limited planning time, we would like to reuse some of the computation performed for the previous plan. ASIMO’s discrete set of actions forces us to use a distance threshold instead of a unique goal configuration. Because of this and the state-dependence of ASIMO’s actions, we use a forward search in our planning. This method has prevented us from

Algorithm 2: VERIFYPATH($path, s_{goal}, E$)

```

foreach  $step \in path$  do
  if GoalReached( $step, s_{goal}$ ) then return  $step$ ;
   $e \leftarrow E(step.time)$ ;
   $a \leftarrow step.action$ ;
   $c_l \leftarrow LocationCost(e, step)$ ;
   $c_s \leftarrow StepCost(e, a)$ ;
  if  $c_l = badval$  or  $c_s = badval$  then return NULL;
   $c_e \leftarrow ExpectedCost(e, step)$ ;
   $Q.Insert(step, step.cost + c_l + c_s, c_e, s_{prev})$ ;
   $s_{prev} \leftarrow step$ ;
end
return NULL;

```

using D* [24] to reuse previous computation.

Instead, we use the previous planning result to seed our search queue. While this loses the reasoning behind the path, it allows us to try to build off of the previous result, rather than starting from nothing. Algorithm 2 shows how the previous path is evaluated up until the point where it is no longer valid, or it reaches the goal. If it reaches the goal, the previous plan can be used without modification. If the previous plan does not reach the goal, the footstep planning can then start up from the place where it failed, with all of the steps already in the queue. Note that for this seeding of the queue to provide a benefit, an inadmissible heuristic must be used, to make the planner choose to start from the end of the seeded steps. If the heuristic is admissible, any node in the queue will have a cost equal to or higher than the initial position. We use an inflated heuristic to trade off some optimality for planning speed, allowing this seeding to shift the focus of our planning towards the previous plan.

V. EVALUATION

A. Vision-Planner Integration

The vision-guided footstep planner for the Honda ASIMO, illustrated in Figure 5, is implemented as three components: a vision server, providing environment maps as well as robot and goal positions at around 30Hz, a planning server, which uses the data provided by vision to compute footstep paths, and a client program which coordinates data transfer between

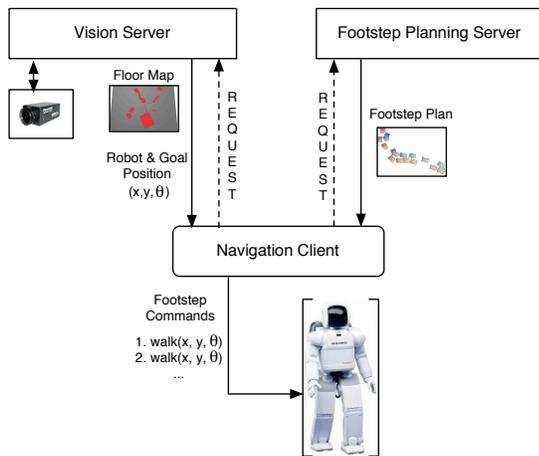


Fig. 5. System components. The navigation client gathers updated sensing data from vision, requests for a new footstep plan to be computed and sends the commands required to execute the path to the robot.

the two servers and wirelessly sends footstep commands to the robot. The components communicate via TCP/IP and run on separate, but commodity hardware.

The client program requests a new footstep plan to be computed from the current environment map and positioning data every time the robot reports that its stance foot has changed during the walking sequence (indicating completion of a footstep). As soon as an updated plan is available, the next command in the computed stepping sequence is sent to the robot. The change in stance foot is thus the discrete event which triggers replanning based on new environment data and transmission of commands to the robot. The time between footsteps, which averages 0.8s, bounds the planner’s time horizon and hence determines the rate at which the perception-action loop runs.

Figure 4 shows a view of ASIMO in a typical, obstacle-filled environment, the robot and obstacles as tracked by the vision system, and the corresponding environment map including current footstep path computed by the planner.

B. Obstacle Avoidance: Unpredictably Moving Obstacles

Figure 6 shows ASIMO walking in the laboratory environment while the surrounding obstacles and the goal location are being changed unpredictably by a human. The task of the experimenter was to intentionally block off the currently computed path to the goal, forcing the planner to recompute the path, reusing from previous plans any footsteps leading up to the occlusion. The goal marker was moved every time the robot was about to reach it, forcing the robot to turn and walk into the new direction of the goal.

We performed on the order of 10–20 trials lasting up to 10 minutes each, during which ASIMO continuously traversed the room without stepping on obstacles. We found the accuracy of the vision-based sensing to be high enough to allow the robot to step within less than 5cm of the obstacles. Most interestingly, in many cases the footstep planner computed

a path that caused the robot to either step over parts of an obstacle (“cutting corners”) or traversing it altogether if the obstacle was small enough to clear in one step (see Figure 1 for an example). This would not have been possible using a standard mobile robot planner.

The vision-guided planner reacts quickly to the dynamic environment, returning a new path within one second of a change in the obstacle configuration or goal position. Only when an obstacle was thrown directly into the floor region occluded by the robot’s body did the robot step on obstacles. In this case, there is no opportunity for the permanence region to ‘absorb’ the obstacle from the surroundings, which hence remains undetected.

VI. DISCUSSION

We have presented an approach to autonomous humanoid walking in the presence of dynamically moving obstacles that combines sensing, planning and execution in a closed loop. By combining fast vision processing with an A* search-based planner operating at the level of biped footsteps and supporting efficient plan reuse, we have enabled a Honda ASIMO humanoid to navigate unpredictable environments along optimal footstep paths for prolonged periods of time.

We are presently working on determining the floor area occluded by the robot at any point in time on a per-pixel basis by projecting a 3D model of the robot in the current walking configuration onto the environment map representing the floor. This will yield a more realistic estimate of the floor directly surrounding the robot’s feet.

Furthermore, we are currently investigating several options for on-body vision that satisfy the real-time constraints of the sensing loop and are more applicable to sensing the floor environment than the robot’s head-mounted cameras. We hope to thus incrementally integrate information from local on-body sensing while the robot is walking to build environment maps of larger areas, giving ASIMO even more freedom to navigate its environment.

ACKNOWLEDGMENT

The authors thank Honda R&D Corporation for generously providing the ASIMO humanoid used in this paper’s experiments to Carnegie Mellon University.

REFERENCES

- [1] Y. Sakagami, R. Watanabe, C. Aoyama, S. Matsunaga, N. Higaki, and K. Fujimura, “The intelligent ASIMO: System overview and integration,” in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS’02)*, 2002, pp. 2478–2483.
- [2] K. Hirai, M. Hirose, Y. Haikawa, and T. Takenaka, “The development of Honda humanoid robot,” in *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA’98)*, May 1998, pp. 1321–1326.
- [3] 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. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS’98)*, 1998, pp. 96–101.
- [4] K. Nagasaka, M. Inaba, and H. Inoue, “Walking pattern generation for a humanoid robot based on optimal gradient method,” in *Proc. of the IEEE Int. Conf. on Systems, Man, and Cybernetics*, 1999.

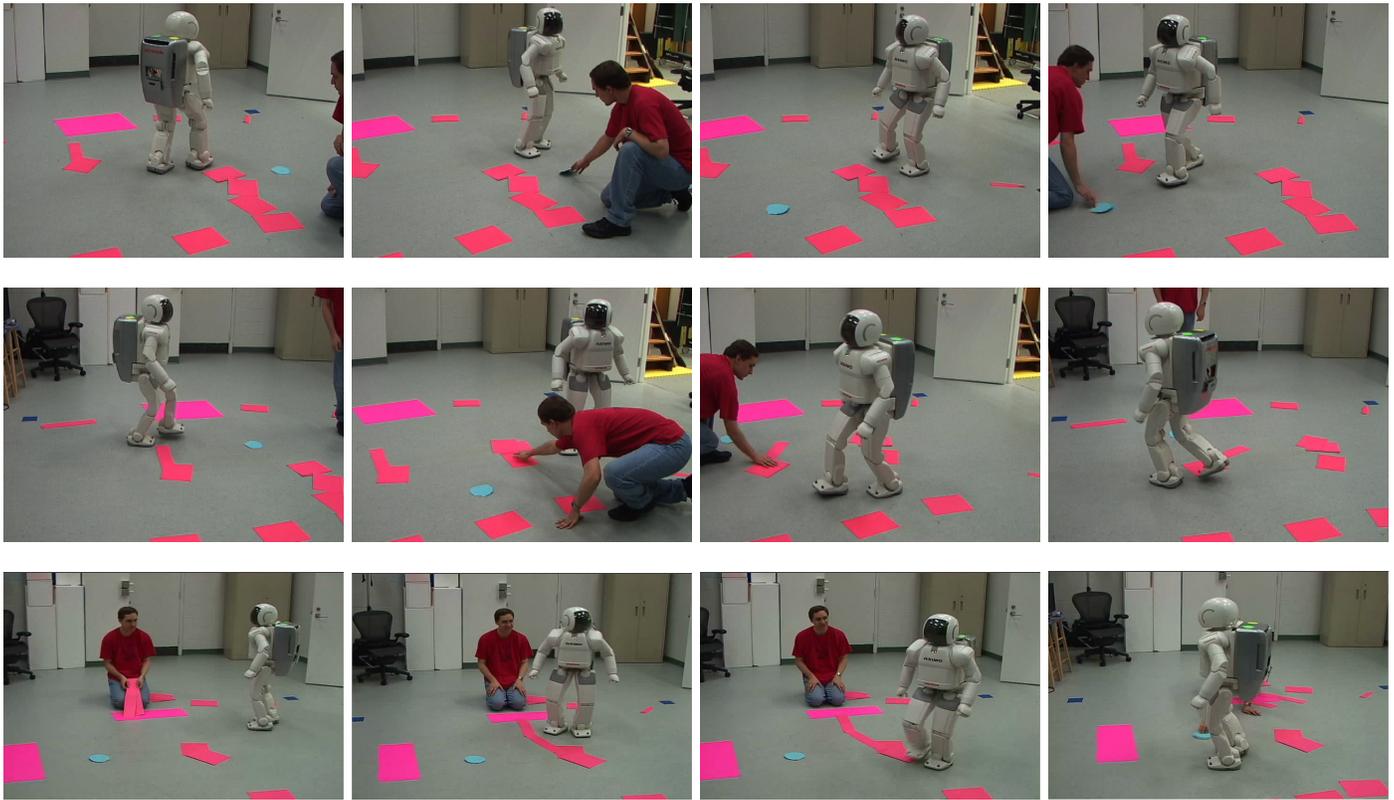


Fig. 6. ASIMO navigating in an environment with unpredictably moving obstacles and a changing goal location.

- [5] 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. of the IEEE Int. Conf. on Robotics and Automation (ICRA'01)*, Seoul, Korea, May 2001.
- [6] 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. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS'02)*, 2002, pp. 96–101.
- [7] J. Pratt and G. Pratt, "Exploiting natural dynamics in the control of a 3D bipedal walking simulation," in *Proc. of the Int. Conf. on Climbing and Walking Robots (CLAWAR99)*, Sept. 1999.
- [8] A. E. Patla, "How is human gait controlled by vision?" *Ecological Psychology*, vol. 10, pp. 287–302, 1998.
- [9] A. Patla, E. Niechwiej, and L. Santos, "Local path planning during human locomotion over irregular terrain," in *Proc. of the Int. Symposium on Adaptive Motion in Animals and Machines AMAM2000*, 2000.
- [10] M. Yagi and V. Lumelsky, "Local on-line planning in biped robot locomotion amongst unknown obstacles," *Robotica*, vol. 18, no. 4, pp. 389–402, 2000.
- [11] M. Yagi and V. J. Lumelsky, "Biped robot locomotion in scenes with unknown obstacles," in *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA'99)*, Detroit, MI, May 1999, pp. 375–380.
- [12] 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. of the IEEE Int. Conf. on Humanoid Robotics (Humanoids 2000)*, 2000.
- [13] R. Cupec, O. Lorch, and G. Schmidt, "Vision-guided humanoid walking - concepts and experiments," in *Proc. of the 12th Int. Workshop on Robotics in Alpe-Adria-Danube Region (RAAD'03)*, Cassino, Italy, May 2003.
- [14] K. Sabe, M. Fukuchi, J.-S. Gutmann, T. Ohashi, K. Kawamoto, and T. Yoshigahara, "Obstacle avoidance and path planning for humanoid robots using stereo vision," in *Proc. of the Int. Conf. on Robotics and Automation (ICRA'04)*, New Orleans, LA, April 2004.
- [15] J. Kuffner, K. Nishiwaki, S. Kagami, M. Inaba, and H. Inoue, "Footstep planning among obstacles for biped robots," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS'01)*, 2001, pp. 500–505.
- [16] J. Kuffner, K. Nishiwaki, S. Kagami, Y. Kuniyoshi, M. Inaba, and H. Inoue, "Online footstep planning for humanoid robots," in *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA'03)*, 2003.
- [17] J. Chestnutt, J. Kuffner, K. Nishiwaki, and S. Kagami, "Planning biped navigation strategies in complex environments," in *Proc. of the IEEE-RAS/RSJ Int. Conf. on Humanoid Robots (Humanoids'03)*, Munich, Germany, Oct. 2003.
- [18] J.-S. Gutmann, M. Fukuchi, and M. Fujita, "A floor and obstacle height map for 3D navigation of a humanoid robot," in *Proc. of the Int. Conf. on Robotics and Automation (ICRA'05)*, Barcelona, Spain, April 2005.
- [19] K. Nishiwaki, S. Kagami, J. Kuffner, K. Okada, Y. Kuniyoshi, M. Inaba, and H. Inoue, "Online humanoid locomotion control using 3D vision information," in *Proc. of the IEEE/RSJ Int. Symposium on Experimental Robotics (ISER'02)*, 99 2002.
- [20] J. Bruce, T. Balch, and M. Veloso, "Fast and inexpensive color image segmentation for interactive robots," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS2000)*, Japan, October 2000.
- [21] Z. Wasik and A. Saffiotti, "Robust color segmentation for the robocup domain," in *Proc. of the Int. Conf. on Pattern Recognition (ICPR'02)*, Quebec City, Quebec, CA, 2002.
- [22] Z. Zhang, "Flexible camera calibration by viewing a plane from unknown orientations," in *Proc. of the Int. Conf. on Computer Vision (ICCV '99)*, Corfu, Greece, September 1999, pp. 666–673.
- [23] J. Chestnutt, M. Lau, G. Cheung, J. Kuffner, J. K. Hodgins, and T. Kanade, "Footstep planning for the Honda ASIMO humanoid," in *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA'05)*, April 2005.
- [24] A. Stentz, "The Focussed D* algorithm for real-time replanning," in *Proc. of the Int. Joint Conf. on Artificial Intelligence (IJCAI'95)*, 1995.