Terrain Mapping for Outdoor Robots: Robust Perception for Walking in the Grass

Regis Hoffman and Eric Krotkov

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA. 15213 U.S.A.

Abstract - **The design and implementation of robot perception systems that operate in outdoor environments poses many challenges due to changing weather, lighting, and temperature conditions. We describe the perception system for the Ambler, an autonomous, legged mobile robot that operates in rugged environments, and analyze its performance during a 500 meter autonomous outdoor walking experiment. The perception system aggressively verifies sensor data and uses feedback from terrain contact to increase accuracy. We identify characteristics of a robust perception system, and summarize our experiences in outdoor perception.**

I. INTRODUCTION

The Ambler is an autonomous, walking robot that operates in rugged terrain (Figure 1). The mechanism is configured as two stacks, each with a set of three orthogonal legs. The Ambler body carries on-board computing, power, and control electronics. A laser scanner is mounted on top of the body to acquire 3-D range images of the local terrain [1][7], and is enclosed by a frame that provides weather-proofing for the vehicle when it is idle.

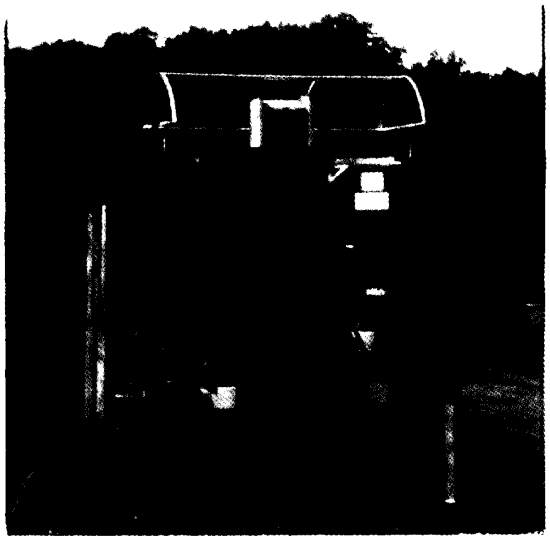

Figure 1. Ambler in the Outdoor Test Course.

The three major components of the Ambler's software system are perception, planning, and control. The perception system builds elevation maps of the local terrain from 3-D range images. The planning system uses elevation maps to determine obstacle-free routes and to select footfall locations, and the control system takes commands from the planner and executes body and leg trajectories.

Robotic systems that address the difficulties of outdoor perception have been demonstrated. The Jet Propulsion Laboratory's Robby used stereo vision to complete a 100 m traverse in outdoor terrain [6]. Work with the Carnegie Mellon NavLab involving both laser rangefinding and camera images to navigate on roads is reported in [8][9]. Road-following for VaMoRs is described in [2].

The Ambler walking system has successfully negotiated an assortment of terrain, including sand, rocks, red clay, grass, and asphalt. During the late spring and summer of 1992, walking experiments were conducted on an outdoor test course (Figure 1) consisting of grass-covered, gently rolling hills (up to 15 degree slope). The temperature ranged between 50° F to 80° F with lighting conditions from bright sunshine to partly cloudy to early dusk.

The perception system has been thoroughly tested in numerous walking experiments, processing tens of thousands of images and tens of millions of terrain elevation points. In the longest walking experiment, the robot autonomously traversed 500 meters over a 21 hour period. During the walk, the perception system acquired 1200 range images and built 4700 terrain elevation maps at 10 cm resolution (containing a total of 2.6 million elevation points). The variety of environmental conditions and the duration of the walking experiment taxed the perception system. Our experience highlights the requirements for a perception system to operate continuously in natural, outdoor environments. We argue that a robust perception system must:

- *Verify* sensor data to detect sensor malfunction,
- *Compensate* for long-term sensor drift by monitoring sensor error,
- *Quickly process* sensor data into its final high-level representation,
- *Respond and recover* from hardware errors detected by the operating system.

This paper reviews the design and implementation of the Ambler perception system. We show how feedback from leg contact with the terrain is used to increase the accuracy of the terrain elevation maps. We also analyze its performance during the 500 meter walk, with particular attention to unexpected problems and how they were solved.

II. PERCEPTION SYSTEM

The perception system builds terrain elevation maps of the local 3-D environment. The elevation maps are computed from range images acquired by a Perceptron laser scanner that provides 256 x 256 pixel range and reflectance images, with 12 bits of data per pixel, at a frame rate of 2 Hz [5]. The sensor uses the phase difference between an emitted amplitude-modulated laser beam and the reflected signal to determine range. The scanner has a 60 degree field of view in the horizontal and vertical directions, and an operating range of 2-40 meters. Figure 2 shows a pair of range and reflectance images. The scene consists of a sand base with tens of 1 meter high boulders (foreground) and a wooden ramp (background).

Figure 2. Range (left) and Reflectance (right) Image.

The perception system takes a sequence of range images I_0, I_1, ..., I_N together with the position of the robot body B_0, B_1, ..., B_N (in a world reference frame). Given this set of range images and the position of the robot when the images were taken, we compute a map of the environment by merging maps created from each single image. Maps can be computed in any reference frame. Common frames include the world frame for global maps, the current body frame for leg recovery, and future body frames for advance planning. If a point in the map cannot be computed (due to invalid range data, outside the field of view, or occlusion), the map point is tagged as *unknown*.

Maps are computed from a sequence of range images, because maps created from a single frame of data do not, in general, contain enough information to accomplish even simple tasks. For example, consider the task of planning the trajectory of a recovering leg. Because the scanner looks forward, the map constructed from a single forward-looking image cannot possibly see obstacles either below or behind the vehicle.

Figure 3 shows a 4 x 5 meter elevation map at 10 cm resolution computed from the range image in Figure 2. The dark areas are labeled unknown regions, either because of terrain occlusion or bad data in the range image.

Three concurrent modules that communicate via message passing, constitute the core of the perception system [3][4]. The Perceptron Interface Module (PIM) acquires range and reflectance image pairs at the completion of a robot body move. The Image Queue Manager (IQM) processes and maintains a list of the most recent images. The Local Terrain Module (LTM) builds maps from the image list on demand from other modules.

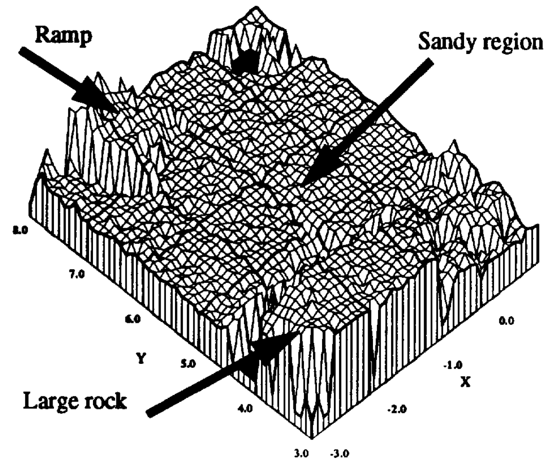

Figure 3. Terrain Elevation Map.

III. DATA VERIFICATION

The scanner geometry and a weak return signal are two primary sources of range measurement error. Geometrically, the scanner aperture introduces invalid range and reflectance pixels in a crescent-shaped area in the lower corners of the images (Figure 4). An internal reflection causes a small region in the lower right portion of the range image to have range values 50 cm too low. Weak return signals arise from both surface material reflectance properties (dark objects reflect less of the laser than light objects), and from surface geometry (surfaces perpendicular to the laser will return a stronger signal than surfaces at a smaller angle of incidence).

To accurately build maps of the terrain, we identify pixels in the range image that are invalid, and do not use them in the computation of the elevation map. A *valid pixel mask* is computed from the range image - the valid pixel mask is 1 (white) if the corresponding range pixel is valid, otherwise it is 0 (black). Scanner geometry defects appear at fixed locations in the image and are detected by region thresholding. Portions of the range image with poor signal return (*i.e.* noisy pixels) are identified by applying a connected region analysis.

Figure 4 illustrates a mosaic of a range and reflectance image (upper half), an edge detection over the range image (lower left), and the valid pixel mask (lower right) computed from the range image. The scene is a relatively flat, grass-covered area. The lower corners of the valid pixel mask show the detection of the aperture effect and the internal reflection (lower right). At the top of the valid pixel mask, noisy range values due to surface geometry are identified. In the example, 76% of the range image pixels are valid - in practice 70-90% are so tagged.

These problems of material sensitivity, internal reflections *etc.* are well understood, and are common to each range image.

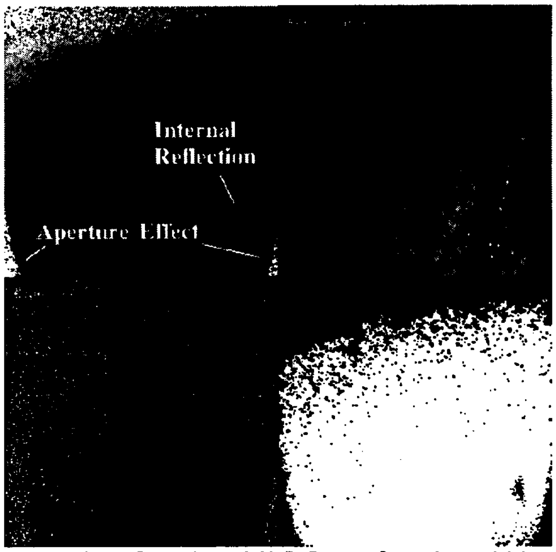

Figure 4. Range Image (upper left). Reflectance Image (upper right). Edge Image (lower left). Valid Pixel Mask (lower right).

During the 500 meter walk, a transient problem was discovered (appearing approximately once every 20 images). The left hand sides of Figure 5 and Figure 6 show range images corrupted by a horizontal band of noise pixels.

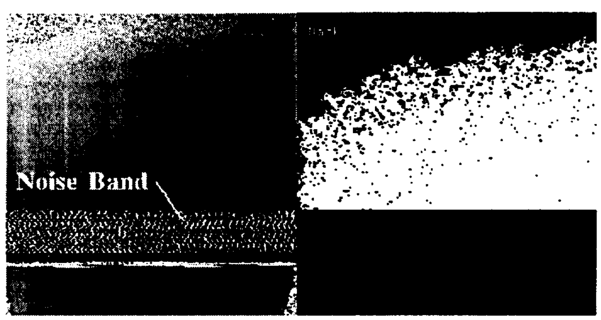

Figure 5. Range Image (left) Corrupted in Lower Half. Valid Pixel Mask (right).

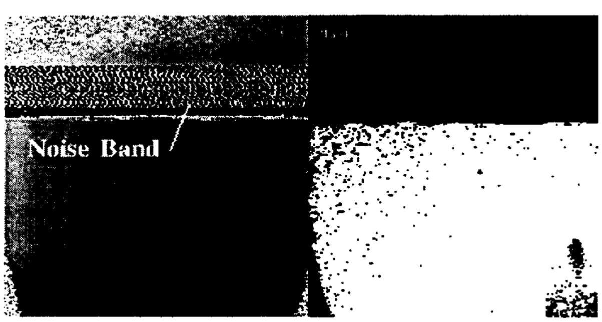

Figure6. RangeImage(left)CorruptedinUpperHalf. Valid Pixel Mask (right).

Though no definitive explanation for these random bands was found, one theory is that vibrations to the sensor electronics caused by the robot's on-board generator loosened a connection or a chip. In this situation, the connected region analysis correctly tags the noise pixels as invalid (right-hand side of Figure 5 and Figure 6). Without the connected region test, this transient problem would not be detected, and the noise bands would be used in the map computation. When they are, they appear as 1 to 2 meter high obstacles in the elevation maps. This exemplifies the requirement to verify the data before it is accepted into the perception system computations.

IV. Error Compensation

Long-duration autonomous operation opens a window of vulnerability to range measurement drift, leading to inaccuracies in the terrain elevation maps. The three main causes of range measurement drift are changes in temperature, terrain material properties, and terrain slope. Range values have been experimentally shown to change 1 cm per degree Fahrenheit ([5] reports a 10 cm per degree drift, but sensor enhancements have reduced this to 1 cm per degree). Different materials (say walking from grass into soil) have different reflectance properties which affect the range measurement. Terrain perpendicular to the incident sensor laser beam will return a stronger (and hence less noisy) signal than terrain at a smaller angle of incidence. During the 500 m walking experiment, we encountered all three types of changes; 20 degree F temperature change, wet to dry grass, and slopes from 0 to 15 degrees.

A. Map Error

To characterize the map error, we use the displacement of the robot legs as a measure of ground truth. We compare the height of terrain maps expressed in the robot body frame (computed by the perception system) to the vertical leg displacement (also in the body frame) and define:

Map error = Vertical leg displacement - Map height

This is illustrated in Figure 7. Each time the robot takes a step, we compare the vertical leg displacement to the height of the perception system's terrain elevation map under the leg. If the map error is greater than zero, the foot is above the sensed terrain, and if the map error is less than zero, the foot is below the sensed terrain.

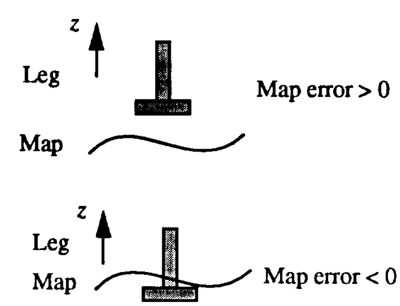

Figure 7. Map Error Computation.

B. Feedback Control of Map Height

The map error term can be used as an error signal in a control loop to reduce the long-term drift problems. We implemented a simple control loop where the height of the perception system maps is the variable being adjusted (Figure 8).

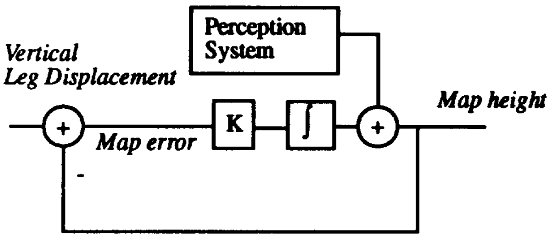

Figure 8. Closed Loop Control of Map Height

The results of open loop operation followed by a closed loop control are shown in Figure 9. In this example, the robot took 371 steps over an 8 hour period, and travelled a distance of 172 meters. During the open loop portion of the experiment, there is a pronounced upward drift in the elevation values. When closed loop operation began (with gain K set to 0.1), we drive the map error to near zero, and minimize the effects of long-term sensor drift. Note that the nominal sensor noise of 10-20 cm remains.

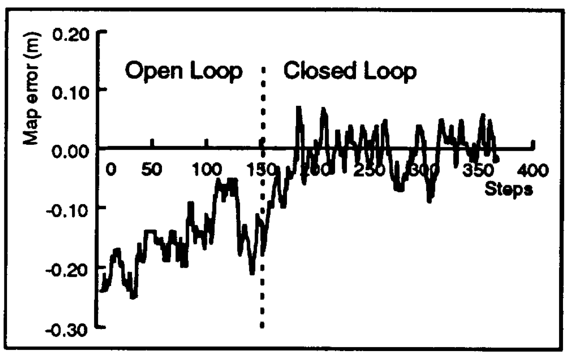

Figure 9. Map Error During Open Loop and Closed Loop Control
Total Distance = 172 meters.

There are two important consequences of the closed loop operation. First, leg recovery (lifting a rear leg, swinging it forward, and planting it on the ground in front of the robot) is more energy efficient because the leg stays very close to the ground (compared to the open loop case where it is further away from the terrain). The second consequence is greater mobility on slopes. Consider the configuration in Figure 10 in which the robot travels along the slope's contour line (towards the observer), and has a leg on the left stack near a lower vertical limit (the leg cannot be raised), and is near an upper vertical limit (the leg cannot move lower) on the right side. If the perception system maps are too high, the planner cannot raise the leg on the left stack to clear the terrain, as this would exceed the leg limit. The opposite problem surfaces on the right stack if perception maps are too low. In this case, leg extensions to contact the terrain are not allowed, because this would exceed the upper leg limit. Decreasing the map error to near zero increases mobility, by permitting a greater range of terrain slopes to be successfully traversed. Both of the advantages of closed loop operation were demonstrated during the 500 m experiment.

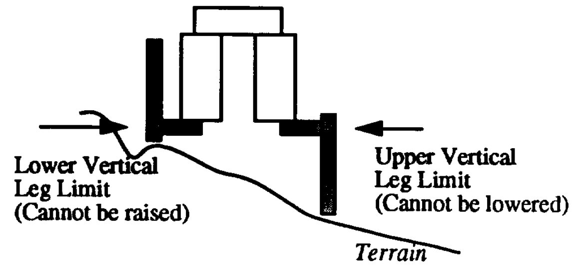

Figure 10. Slope Traversal. Direction of Motion is Forward.

V. Computing

Perception system computer performance is dominated by both the significant memory required to store the list of range images, and the time to compute terrain elevation maps. Without timely map construction, the perception system would be the bottleneck in the walking cycle. For researchers contemplating similar work, we list computer performance metrics from the 500 m walk. All perception computing was done on a Sun Microsystems Sparc II with 64MB of memory.

A. Memory Requirements

For optimum performance, there must be sufficient physical memory to store the list of range images taken as the robot moves through the terrain. In practice, it is impractical to store the complete list due to the finite amount of memory available. However, as the robot moves forward through the terrain, only a subset of the range images taken will be used to build maps because the size of the map is, in general, much smaller than the field of view of a single range image (typical map size from 0.2 to 5 square meters, range image field of view approximately 20 square meters. For small maps completely in front of the body, only one or two range images may be needed; for large maps behind the body, more images from the list may be used. The number of range images needed to fulfill a map request is histogramed in Figure 11. In many cases, only 5 images were used; this number increased to 30 a sizable amount of times. The memory to store 30 images, plus the code to manipulate them, was about 25MB.

B. Map Computation Time

Perception system compute time is dominated by map computation and is determined by the number of range images needed to compute the map. The number of range images is a function of both the geometry (which subset of images have fields of view that overlap the map region), and the quality of images (the more invalid range pixels in a single image, the greater number of images that will be needed from the list).

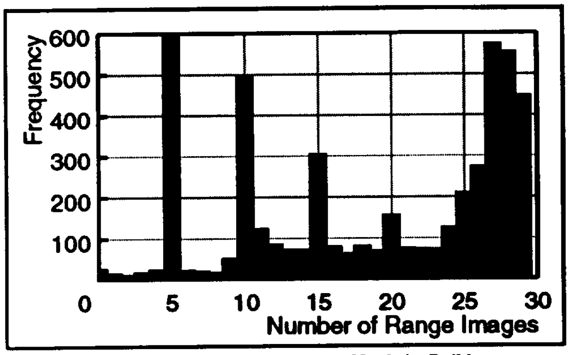

Figure 11. Number of Range Images Needed to Build Terrain Elevation Maps.

Figure 12 histograms the time required to build the 4700 maps computed during the experiment. Most maps were computed in less than 5 seconds, with the maximum time being 13 seconds. Because the maps were of different sizes, the normalized time to compute a single map elevation values is histogramed in Figure 13. This shows it takes about 7 milliseconds to compute a single elevation map point.

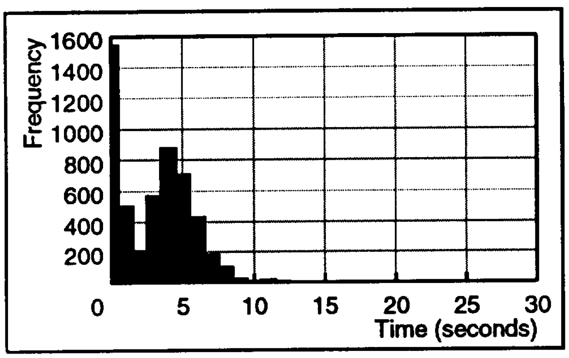

Figure 12. Time to Build Maps.

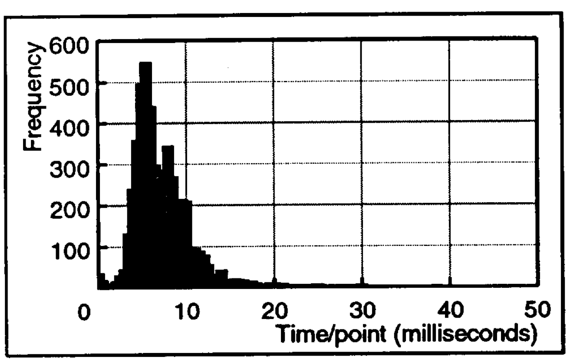

Figure 13. Time/map point.

VI. Response to Hardware Errors

The most severe (and surprising) perception system failure that occurred during the experiment was an intermittent hardware memory parity error. Even though the computer passed all the memory diagnostics, the Image Queue Manager's growing appetite for memory occasionally triggered the parity error.

Because the error was unexpected, no mechanism existed to detect and recover from the error. As perception code at the applications level becomes more robust, the limiting factor will be similar hardware and operating system errors. Next-generation perception systems should detect this class of errors, and restart themselves.

VII. Conclusions

A perception system for an autonomous, mobile robot that operates in outdoor terrain has been described. Based on our walking experiments, we conclude that a robust perception system must verify sensor data to detect sensor malfunctions, compensate for long-term sensor drift, rapidly process sensor data, and respond to hardware errors.

We have demonstrated that feedback compensation from the contact of the leg with the terrain increases the map accuracy, and hence increases the mobility of the robot. We also show that verification of sensor data before its use in map computation allows the robot to operate autonomously for many hours, and not be crippled by sensor malfunction.

References

[1] J. Bares. *Configuration of Autonomous Walkers for Extreme Terrain.* Ph.D. thesis, Department of Civil Engineering, Carnegie Mellon University, 1991.

[2] E. Dickmanns and B. Mysliwetz. Recursive 3-D Road and Relative Ego-State Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence. 14*(2), February, 1992.

[3] R. Hoffman and E. Krotkov. Perception of Rugged Terrain for a Walking Robot: True Confessions and New Directions. *Proceedings of the International Workshop on Intelligent Robots and Systems.* IEEE, November, 1991.

[4] R. Hoffman and E. Krotkov. Terrain Mapping for Long-Duration Autonomous Walking. *Proceedings of the International Conference on Intelligent Robots and Systems.* IEEE, July, 1992.

[5] I.S. Kweon, R. Hoffman and E. Krotkov. *Experimental Characterization of the Perceptron Laser Rangefinder.* Robotics Institute Technical Report CMU-RI-TR-91-1, Carnegie Mellon University, 1991.

[6] L. Matthies. Stereo Vision for Planetary Rovers: Stochastic Modeling to Near Real-Time Implementation. *International Journal of Computer Vision. 8*(1), July, 1992.

[7] R. Simmons, E. Krotkov, W. Whitaker et al. Progress Towards Robotic Exploration of Extreme Terrain. *Journal of Applied Intelligence. 1*(2), 1992.

[8] S. Singh and P. Keller. Obstacle Detection for High Speed Autonomous Navigation. *Proceedings of the IEEE International Conference on Robotics and Automation.* IEEE, Sacramento, California, April, 1991.

[9] Thorpe, C. (editor). *Vision and Navigation: the Carnegie Mellon Navlab.* Kluwer, 1990.