Terrain-Based Sensor Selection for Autonomous Trail Following

Christopher Rasmussen Donald Scott

Dept. Computer & Information Sciences
University of Delaware, USA

cer@cis.udel.edu, donald@udel.edu

Abstract. We introduce the problem of autonomous *trail following* without waypoints and present a vision- and ladar-based system which keeps to continuous hiking and mountain biking trails of relatively low human difficulty. Using a RANSAC-based analysis of ladar scans, trail-bordering terrain is classified as belonging to one of several major types: *flat* terrain, which exhibits low height contrast between on- and off-trail regions; *thickly*-vegetated terrain, which has corridor-like structure; and *forested* terrain, which has sparse obstacles and generally lower visual contrast. An adaptive color segmentation method for flat trail terrain and a height-based corridor-following method for thick terrain are detailed. Results are given for a number of autonomous runs as well as analysis of logged data, and ongoing work on forested terrain is discussed.

1 Introduction

Navigationally-useful linear features along the ground, or *trails*, are ubiquitous in manmade and natural outdoor environments. Spanning carefully engineered highways to rough-cut hiking tracks to above-ground pipelines to rivers and canals, they "show the way" to unmanned ground or aerial vehicles that can recognize them. Built trails also typically "smooth the way," whether by paving, grading steep slopes, or removing obstacles. An ability to recognize and follow trail subtypes amenable to wheeled vehicles such as dirt, gravel, and other marginal roads, as well as true hiking trails, is a vital skill for outdoor mobile robot navigation. The alternative of cross-country route-finding over desert, forested, or mountainous terrain is fraught with difficulties [1–5]. In contrast, a trail is a visibly-marked, precomputed plan that can be followed to lessen danger, minimize energy expenditure, save computation time, and ease localization. The routes given to competitors in the 2004 and 2005 DARPA Grand Challenge Events (GCE) generally followed trails for this very reason. However, as they were densely "marked" with human-chosen GPS waypoints, the burden of sensor-based route-finding was lessened considerably for the robots.

In this paper we present a component of a larger project tackling the problem of enabling UGVs to robustly and quickly follow a wide variety of trails *without*

Fig. 1. Our robot autonomously following flat and thickly vegetated trail segments (see text for definitions)

any waypoints. The subtasks necessary for robust trail-following may be divided into three categories:

Trail keeping Analogous to the sense of "lane keeping" from autonomous road following, keeping to a trail involves determining the shape and boundaries of a non-branching, non-terminating trail that the robot is already on, and controlling the platform to move along it. For *discontinuous* trails marked by blazes, footprints, or other sequences of discrete features, the underlying task is successive guided search rather than segmentation.

Trail negotiation Because trails cannot be assumed to be homogeneously-surfaced free space (as with paved roads lacking other cars), trail negotiation adds vigilance for and responses to in-trail hazards. It includes adjusting direction to skirt solid obstacles such as rocks and fallen logs, as well as slowing, shifting, and other control policy modifications to suite different trail surface materials (e.g., sand, mud, ice) and terrain types (e.g., flat, bumpy, steep slope, etc.).

Trail finding Finding entails identifying trail splits and dead-ends that violate the assumptions of the trail keeper, as well as searching for trails from a general starting position, whether using onboard sensors or *a priori* map and aerial data.

The trail-following component reported on here is a joint image and ladar algorithm for *keeping* to hiking/mountain biking trails which are continuous and relatively free of in-tread hazards. Our platform is a Segway RMP 400 (shown in Figure 1) with a single rigidly-mounted SICK LMS-291 ladar and a color video camera. The robot is controlled via Player client/server software [6].

The types of terrain and vegetation surrounding a hiking trail have a major bearing on the perceptual difficulty of the trail-following task. For simplicity, in this work we assume that trail-bordering terrain types fall into three common categories (examples are shown in Figure 2):

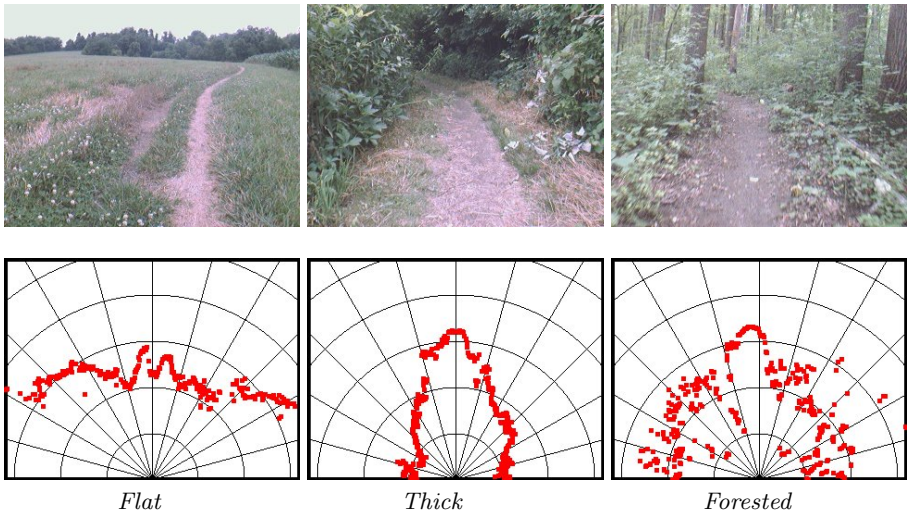


Fig. 2. Examples of main terrain types bordering trail segments considered in this paper. Synchronized ladar scans are shown at right in ladar coordinates. Each arc represents a 1 m range difference; radial lines are 15 degs. apart

Flat Off-trail regions such as grassy fields that have generally poor height contrast with the trail tread, making structural cues from ladar less reliable than appearance cues from the camera. Depending on the season, color contrast between the two regions may be very strong (e.g., in summer) or relatively weak (winter).

Thick Trail segments bordered by dense vegetation such as bushes, trees, or tall grass which form virtual corridors that are highly amenable to ladar cues. Image-based segmentation is often complicated by shadow and glare issues.

Forested Areas under canopy with more widely-spaced obstacles like tree trunks. Here the correlation between obstacle distribution and trail direction is typically weak and visual contrast between on- and off-trail regions is also often poor, as they may both be largely dirt or leaf-covered.

One of our key contributions is a method for recognizing which of the above categories a trail segment belongs to in order to discretely *switch* to a different vision-based or ladar-based algorithm better suited to that terrain.

In the next several sections, we describe our system's methods for terrain type classification; an adaptive, vision-based trail segmentation technique for flat terrain; and a ladar-based approach to corridor following in thick terrain. Our results show the robot successfully operating autonomously using these techniques in an integrated fashion. In the conclusion we discuss the next steps necessary to extend the system to further terrain types and increase performance and safety.

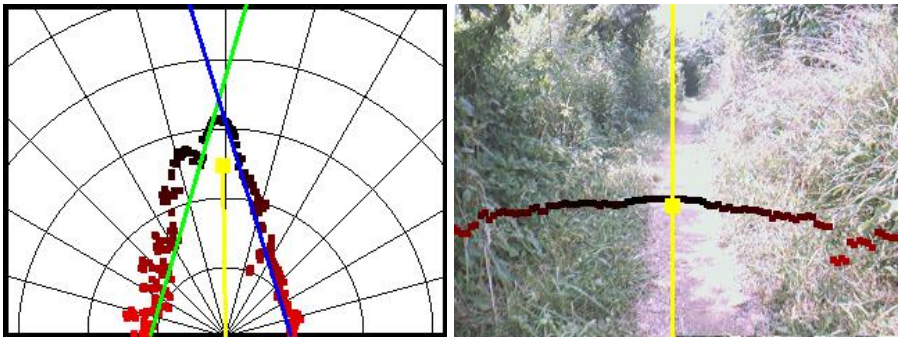


Fig. 3. Ladar-based terrain classification. Left and right edge estimates on thickly-vegetated trail section are shown in green and blue, respectively. The yellow line is the gap direction estimate (explained in Section 4). Saturation of the plotted ladar points is proportional to absolute height in robot coordinates.

2 Terrain Type Classification

Our robot's ladar is mounted about 0.5 meters off the ground and pitched down 10 degrees. Thus, on planar ground the ladar beam sweeps across the ground nearly $s = 3$ m in front of the robot. The scans are set to a maximum range of about 8 m at 0.5 degree resolution over an angular range of 180 degrees. The general strategy behind mounting the ladar in this manner is to obtain data about tall obstacles on the robot's flanks as well as information about the profile of the ground just ahead, including smaller bumps and negative hazards.

Both Stanford and Carnegie-Mellon (CMU), which collectively accounted for the top three robots in the 2005 GCE, relied exclusively on multiple ladars for steering decisions (Stanford used vision as an input to speed control) [7, 8]. However, we assert that ladar alone is not sufficient for successful trail following in each kind of terrain defined above. Stanford's and CMU's approaches worked because the provided GPS waypoints gave complete navigational information, making ladar only necessary for local obstacle avoidance. With no waypoints for guidance, in some areas—particularly flat terrain—ladar is virtually useless to keep the robot on track and appearance cues must be exploited for navigation. As can be seen in the "flat" example of Figure 2, the trail is very difficult to discern in one ladar scan (especially with a large rut to its left) but obvious in the camera image.

Although ladar may be inadequate for navigation in certain kinds of terrain, it can be very informative about *which* terrain type the robot is on. The basic idea is that the gross shape of a single scan is very different depending on which terrain type is present. On level, planar ground, the scan is a horizontal line noisified by grass, ruts, and other features in flat terrain environments. With the range maximum we use for the ladar, nonlinearities due to curvature of the underlying ground are negligible. If the robot is on a sideslope (with the ground

higher on one side and lower on the other), then the line is simply diagonal with a slight jog in the middle corresponding to the level cross-section of trail. If the robot is traveling along the top of a slight ridge or the bottom of a slight valley, the scan shape will appear as a wide $\vee$ or $\wedge$, respectively.

Ladar scans taken in thickly-vegetated segments with their corridor-like structure, on the other hand, are characterized by two more nearly parallel linear clusters (as in the "thick" example in Figure 2). These are the "walls" of the corridor of foliage that the robot should follow. The difference between this configuration and the $\wedge$ shape of some flat terrain is the acuteness of angle between the lines. In other words, for flat and thick terrain situations a reasonable fraction of the ladar data should be well-described by one or two lines. Forested terrain segments are the leftover category, characterized by jumbled ladar scans lacking enough coherent structure to fit a line or lines with any significant support.

Based on these observations, our method for categorizing ladar scan shapes into terrain types is as follows. Let the x axis in robot coordinates be defined by the robot's front wheel axle, with $+x$ to the right and $-x$ to the left of the robot center. On each new scan, two RANSAC robust line-fitting procedures [9] are performed in succession. The first is a nominal *left edge* fit, on which a constraint is enforced that the estimated line must intersect the x axis at a point $x_L < 0$. If no line is found with over some fraction f of inliers (we used 20% here) in a reasonable number of samples, the left edge is considered *not found*. The second RANSAC procedure is a *right edge* fit run on the set of original ladar scan points *minus* the left edge inliers and subject to the constraint that it intersect the x axis at a point $x_R > 0$. If f (with respect to the number of original points) is not exceeded by any line, then the right edge is not found. Note that $|x_L|$ or $|x_R|$ can be very large, allowing a virtually horizontal line fit.

After running these two fitting operations, if neither a left edge nor a right edge is found, the current terrain is considered forested. If only one edge is found, the terrain is considered flat[1]. If both edges are found, the terrain is considered flat if $x_R - x_L > \tau$ and thick otherwise ($\tau = 4.0$ m was used for the runs in this paper).

Somewhat surprisingly, running RANSAC independently on each successive frame yields fairly consistent estimates of the left and right edge lines. We smooth what noise there is with temporal filtering of the terrain classification itself via majority voting over a recent-history buffer.

An example of left and right edge estimates on a thick terrain scene that would be especially difficult for a visual method is shown in Figure 3.

3 Flat Terrain: Image-based Trail Segmentation

On flat terrain the preferred sensory modality is camera imagery, since on- vs. off-trail height contrasts are mostly too small for reliable ladar discrimination.

[1] We currently do not account for the possibility of thick foliage on only one side of the trail (which would permit simple "wall-following"), but this could easily be incorporated into the classification logic

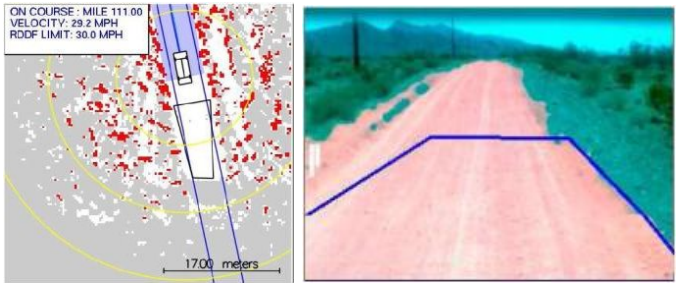
Though the trails we are considering are engineered for walking or bike-riding people only and thus are too small for road-legal cars or trucks, an obvious approach is to consider the task as just a scaled-down version of road following. Vision-based road following has been thoroughly studied on paved and/or painted roads with sharp edges [10–13], but hiking trails rarely have these. However, region-based methods using color or texture measured over local neighborhoods to measure road vs. background probabilities work well when there is a good contrast for the cue chosen, regardless of the raggedness of the road border [14, 13, 15–17].

The most notable recent work in this area is Stanford's 2005 GCE road segmentation method [18]. Specifically, their approach relies on a *driveability map* (depicted in Figure 4(a)) accumulated over many timesteps and integrated from multiple ladars. A quadrilateral is fit to the largest free region in front of the vehicle in the latest map, and this is projected into the image as a training region of positive road pixel instances (Figure 4(b)). The method of modeling road appearance is with a mixture of k Gaussians in RGB space (with temporal adaptation) and does not require negative examples: new pixels are classified as road or non-road solely based on a thresholded distance to the current color model. This follows the framework of [19]'s original image-based obstacle avoidance algorithm, which used a fixed free region and a histogram-based color model.

For several reasons, neither [18] nor [19] is directly applicable to trail following on flat terrain. First, the ladar is not sufficient for selecting an image region belonging to the trail alone because of the lack of height contrast. Secondly, using a fixed positive example region is problematic because the trail is very narrow in the camera field of view compared to a road and can curve relatively sharply. Even in thickly-vegetated terrain, tall foliage does not always grow to the edge of the trail proper, as Figure 2 shows. In both terrain types, without very precise adaptive positioning this approach would lead to off-trail pixels frequently polluting the trail color model .

The very narrowness of the trail suggests a reverse approach: model the off-trail region or *background* with an adaptive sampling procedure and segment the trail as the region most dissimilar to it. In general the background is less visually homogeneous than the trail region and thus more difficult to model, but in our testing area it is not overly so. Under the assumption that the robot is initially on the trail and oriented along it, the background color model is initialized with narrow rectangular *reference areas* on the left and right sides of the image as shown in Figure 2. These areas extend from the bottom of the image to a nominal horizon line which excludes sky pixels and more distant landscape features in order to reduce background appearance variability.

Following the method of [19], a color model of the off-trail region is constructed from the RGB values of the pixels in the reference areas for each new image as a single 3-D histogram with a few bins per channel. The entire image below the horizon is then classified as on- or off-trail by measuring each pixel's color similarity to the background model and thresholding. In particular, if a particular pixel's histogram bin value is below 25% of the maximum histogram



(a) (b)

(c)

Fig. 4. (a,b) Stanford's approach to obtaining road labels (from [18]). (a) Overhead map of obstacle space and free space from multiple ladars integrated over time. Note quadrilateral fit to free space in front of vehicle; (b) Quadrilateral projected to image, with result of road segmentation overlaid. **(c)** Our histogram color classification method for trail segmentation example (frame from one autonomous run). Left and right reference areas are outlined by yellow rectangles; these are the regions from which the non-trail appearance model is built. Pixels classified as on-trail (after density filtering) are highlighted in green. The lateral median of the segmented trail region is indicated by the vertical orange line, which the robot steers to center. The isolated green blobs are gray rocks misclassified as trail.

bin value, it is considered a trail pixel. To reduce noisiness in this initial segmentation, this image is filtered using a 3×3 majority filter. The resulting binary image contains the pixels we consider most likely to belong to the trail (pixels shaded green in Figure 2). Using the image output from the pixel majority filter the trail image center (the orange line in Figure 2) is estimated as the median x value of the filtered trail pixels. This is converted to a trail direction θ with the camera calibration; steering follows a proportional control law.

The reference areas are adapted to exclude the trail region based on a robust estimate of the trail width using the maximum absolute deviation of the lower

section of filtered trail pixels plus a safety margin. If one is squeezed too thin by the trail moving to one side, it is not used. If the trail region is too small or too large, the reference areas are reset to their defaults.

4 Thick Terrain: Ladar-based Corridor Following

When the terrain classifier judges the current segment to be thickly-vegetated, purely ladar-based trail following is a very robust option. An obvious cue for which direction θ to steer the robot comes from the already-computed left and right RANSAC edge lines, as their intersection would seem to be a straightforward indicator of the trail corridor direction. However, because of the "nearsightedness" of our ladar configuration, the edge estimates can vary somewhat due to locally nonlinear structures such as bulges or gaps in foliage lining the trail. While these variations are generally not significant enough to confuse the terrain classifier, they contribute to undesirable instability in intersection-derived trail direction estimates.

A more reliable alternative is to pick the θ within an angular range in front of the robot associated with the *farthest obstacle* at distance r. Similar in spirit to vector field histograms [20], the robot avoids nearby obstacles comprising the walls on either side by seeking the empty *gap* between them. To filter out stray ladar beams which sometimes penetrate deep into trail-adjacent foliage, we calculate obstacle distances for candidate directions using the nearest return within a sliding robot-width window. A simple particle filter [21] is used to temporally smooth θ and r estimates. A proportional control law is then used to generate a steering command.

The farthest-obstacle distance r is used to modulate robot speed. If the robot is traveling over planar ground r should roughly equal the sweep distance s (defined in Section 2 above). Thus, deviations of r from s indicate that the robot is currently pitching up or down or approaching a dip or bump. In any of these cases, the robot assumed to be on bumpy ground and the speed is lowered proportionally to $|r-s|$ (see [22] for a machine learning approach to this problem at much higher speeds).

5 Results

We developed and tested our algorithms on camera and ladar data that were collected while driving the robot around a 1.5 km loop trail used by hikers and mountain bikers and located in a state park. The trail varies from approximately 0.3 m to 2 m wide and is mostly hard-packed dirt with few sizeable rocks or roots within the trail region. Altitude along the trail gradually changes over about a 10 m range, with a few notable humps and dips. Overall, it would be rated as an "Easy" trail under the system of the International Mountain Bicycling Association [23], which is crafted with wheeled motion in mind and is a reasonable scale match for our robot. An aerial view of the entire trail loop is

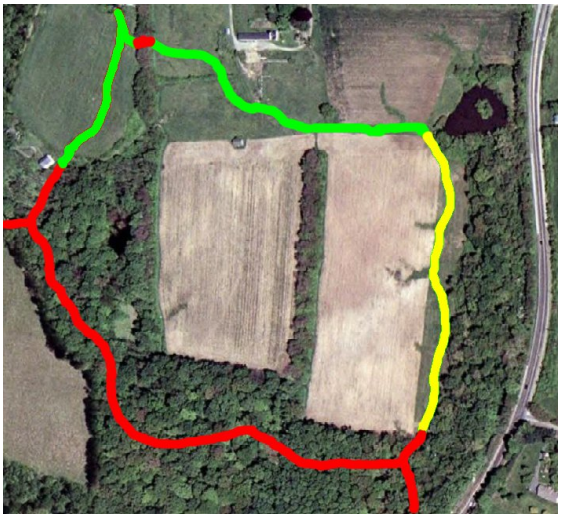


Fig. 5. Aerial view of trail loop where testing was conducted. Colors indicate primary vegetation types bordering trail segments: green marks predominantly flat segments, yellow shows where there was thick, wall-like foliage along the trail, and red shows forested trail segments.

shown in Figure 5, with the flat segments drawn in green, the thick segments in yellow, and the forested segments in red.

Image histogram-based segmentation as described above was very successful for autonomous guidance of the robot along virtually the entire top flat section of the trail shown in Figure 5[2]. The robot covered a distance of several hundred meters on an initial eastward leg until it entered a narrow, thickly-vegetated segment and was manually turned around. The histogram method for background color modeling is somewhat sensitive to illumination variation, and there are several spots in the thick segment where deep shadows would confuse it. The robot was stopped at the end of the reverse westward journey at the location pictured in Figure 4(c) because of large rocks very close to the trail. In this "flat" trail-following mode, the robot had no structural cues for speed control and thus was set to move at a constant rate of 1.0 m / s. The maximum turning rate was set to 0.5 radians / s.

Part of the westward leg is illustrated by images taken at 250-frame intervals in Figure 6. For the most part the segmentation was fairly clean, but there were

[2] For all results, the default width of each reference area was $\frac{1}{6}$ the image width and the horizon line was fixed at $\frac{2}{3}$ the image height. The RGB histogram had 16 bins per channel, and the reference areas were reset if $< 1\%$ or $> 50\%$ of the image was classified as trail. Images were captured at a framerate of 5 fps. The processing resolution after downsampling was 80×60; here we are upsampling the logged results for display

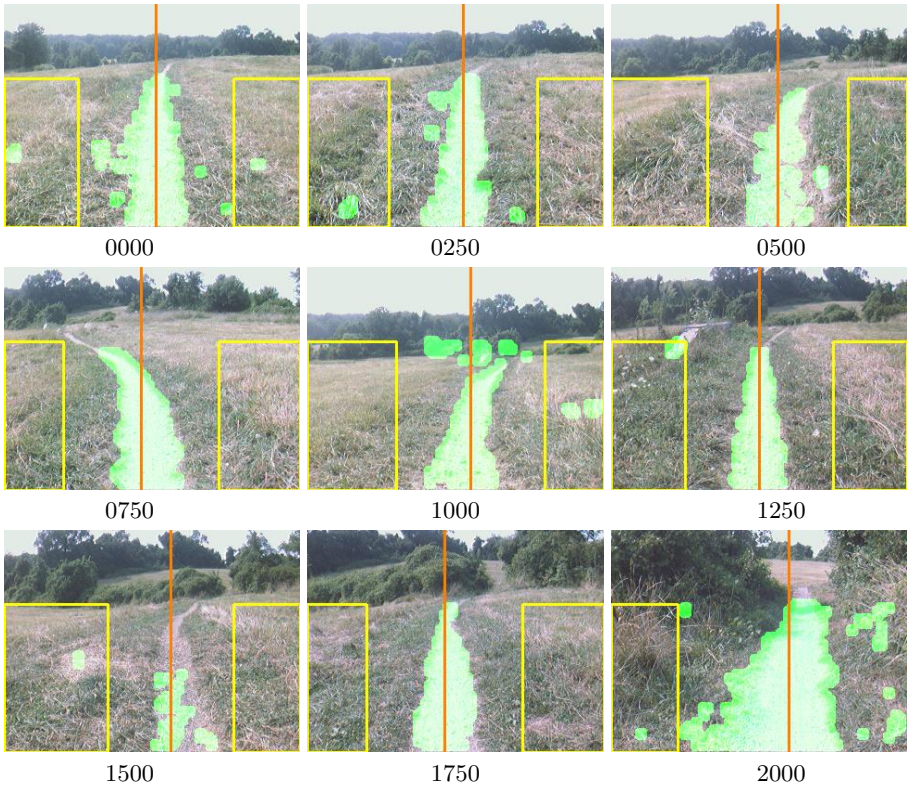


Fig. 6. Part of completely image-based autonomous run on flat trail section. Image labels are camera frame numbers

some sections where the bordering grass contained large brown patches and the color discrimination faltered temporarily, as with frame 1500. When the area of the segmented region dipped below threshold, the robot no longer "saw" a trail and would continue without turning. However, because the trail was not too sharply curved at these points, the adaptive color model recovered in time to refind the trail and correct the overshoot.

The ladar-based terrain mode classifier worked perfectly throughout the runs above, switching from flat to thick correctly toward the end of the eastward leg. This transition is documented in Figure 7 over a sequence of images 100 frames apart. Two strong lines are fit to the ladar data which intersect the robot baseline less than τ apart just before frame 300; this switches the controller from the image segmentation follower to the ladar gap follower. Part of a sequence further into the thickly-vegetated section with the ladar gap follower running is shown in Figure 8.

A nearly 16-minute, manually-controlled run on an earlier day showed similarly successful results. The robot was driven along the trail through thick terrain

(the western half of the flat terrain above before grass over 1 m tall was cut for hay), followed by flat terrain (short grass in the eastern half), then into the thick terrain alley which begins at frame 300 in Figure 7. Both terrain transitions were recognized appropriately and with no other, false switches, and trail tracking was qualitatively accurate. Quantitatively, the trail follower's output steering command predicted the logged human driver's steering command fairly well, with a measured Pearson's correlation coefficient of $\rho = 0.82$ for the histogram-based flat mode and $\rho = 0.75$ for the ladar-based thick mode ($p < 0.001$)[3].

6 Conclusion

We have presented a vision- and ladar-based system for following hiking trails that works well on a variety of flat and thickly-vegetated terrain.

Clearly, the next area needing attention is a robust technique for tackling forested trail segments. As the visual discrimination task is harder, more sophisticated classification methods such as support vector machines and more detailed appearance models including texture [24] may be necessary, though at some cost in speed. Classifying patches or superpixels [25, 2] rather than individual pixels would also most likely yield more geometrically coherent segmentations. Incorporating a whole-trail shape prior [26] into the segmentation process, perhaps with recursive updating, would also help isolate the trail region in an otherwise confusing visual environment.

Some basic *trail negotiation* methods are necessary for robot safety, as the system currently has no mechanism for discovering or avoiding in- or near-trail hazards. Although nearly all of the trail tread in our testing area is obstacle-free, several sections contain roots or are bordered by nearby tree trunks or rocks (as in Figure 4)(c). Existing techniques for visual feature detection and terrain classification [27, 2] should help here, but the problem of discriminating soft obstacles like grass and twigs which can be pushed through from hard obstacles like rocks and logs that cannot is a classically difficult one [4].

7 Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. 0546410.

References

1. B. Nabbe, D. Hoiem, A. Efros, and M. Hebert, "Opportunistic use of vision to push back the path-planning horizon," in *Proc. Int. Conf. Intelligent Robots and Systems*, 2006.

[3] The correlation was highest with the trail follower predicting the manual steering command 1 s later. We are including only data when the robot was traveling more than 0.5 m / s (it was paused several times) and being manually steered left or right at a rate of at least 2 degs. / s (less was a "dead band" treated as straight)

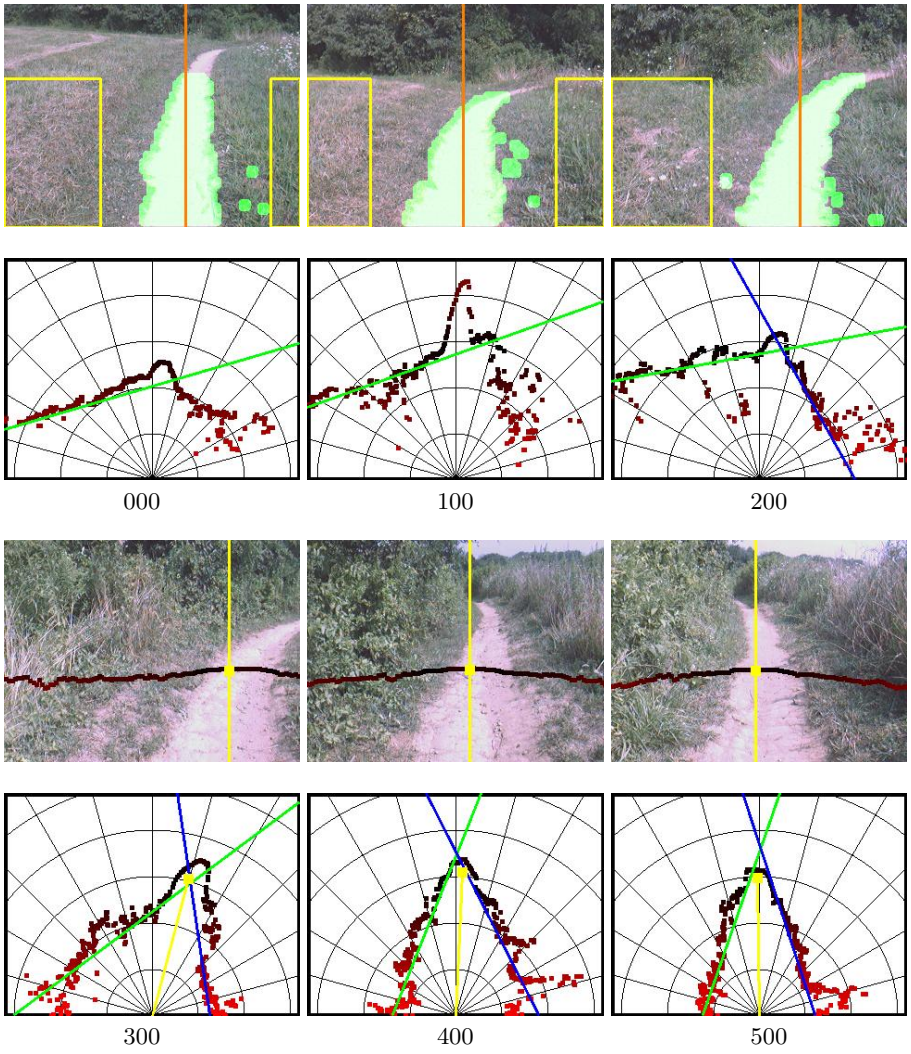


Fig. 7. Autonomous run: transition from flat to thick trail section. Image labels are camera frame numbers. The green and blue lines in the ladar images are estimated left and right edges, respectively. The yellow line in the last three ladar images represents the estimated gap direction and distance after the switch to thick terrain mode.

2. D. Kim, S. Oh, and J. Rehg, "Traversability classification for ugv navigation: A comparison of patch and superpixel representations," in *Proc. Int. Conf. Intelligent Robots and Systems*, 2007.

3. D. Coombs, K. Murphy, A. Lacaze, and S. Legowik, "Driving autonomously offroad up to 35 km/h," in *Proc. IEEE Intelligent Vehicles Symposium*, 2000.

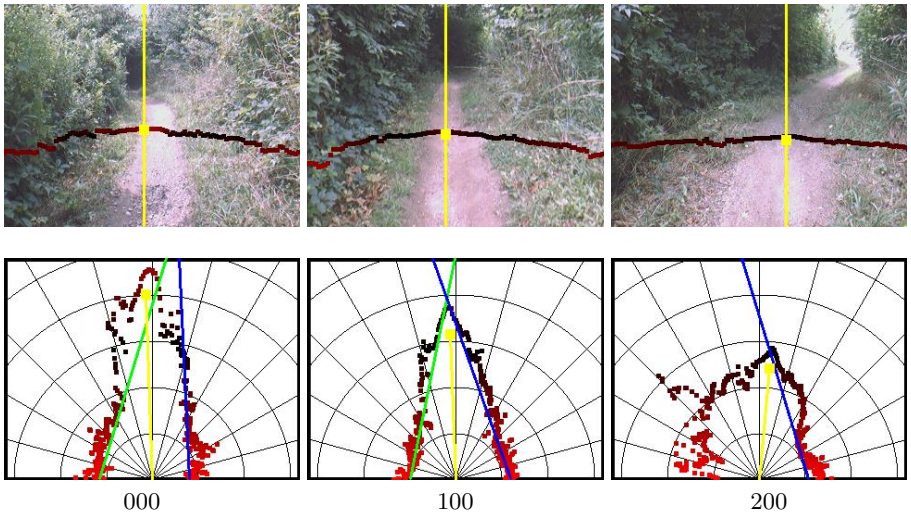


Fig. 8. Ladar-based gap tracking on thickly-vegetated trail section. Image labels are camera frame numbers

4. A. Stentz, A. Kelly, P. Rander, H. Herman, O. Amidi, R. Mandelbaum, G. Salgian, and J. Pedersen, "Real-time, multi-perspective perception for unmanned ground vehicles," in *AUVSI*, 2003.
5. N. Vandapel, J. Kuffner, and O. Amidi, "Planning 3-d path networks in unstructured environments," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2005.
6. B. Gerkey, R. Vaughan, and A. Howard, "The Player/Stage project: Tools for multi-robot and distributed sensor systems," in *International Conference on Advanced Robotics*, 2003.
7. S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, K. Lau, C. Oakley, M. Palatucci, V. Pratt, P. Stang, S. Strohband, C. Dupont, L.-E. Jendrossek, C. Koelen, C. Markey, C. Rummel, J. van Niekerk, E. Jensen, P. Alessandrini, G. Bradski, B. Davies, S. Ettinger, A. Kaehler, A. Nefian, and P. Mahoney, "Stanley, the robot that won the darpa grand challenge," *Journal of Field Robotics*, vol. 23, no. 9, 2006.
8. C. Urmson, J. Anhalt, D. Bartz, M. Clark, T. Galatali, A. Gutierrez, S. Harbaugh, J. Johnston, H. Kato, P. Koon, W. Messner, N. Miller, A. Mosher, K. Peterson, C. Ragusa, D. Ray, B. Smith, J. Snider, S. Spiker, J. Struble, J. Ziglar, and W. Whittaker, "A robust approach to high-speed navigation for unrehearsed desert terrain," *Journal of Field Robotics*, vol. 23, no. 8, pp. 467–508, 2006.
9. M. Fischler and R. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.
10. C. Taylor, J. Malik, and J. Weber, "A real-time approach to stereopsis and lane-finding," in *Proc. IEEE Intelligent Vehicles Symposium*, 1996.
11. G. Stein, O. Mano, and A. Shashua, "A robust method for computing vehicle ego-motion," in *Proc. IEEE Intelligent Vehicles Symposium*, 2000.

12. B. Southall and C. Taylor, "Stochastic road shape estimation," in *Proc. Int. Conf. Computer Vision*, 2001, pp. 205–212.
13. N. Apostoloff and A. Zelinsky, "Robust vision based lane tracking using multiple cues and particle filtering," in *Proc. IEEE Intelligent Vehicles Symposium*, 2003.
14. D. Mateus, G. Avina, and M. Devy, "Robot visual navigation in semi-structured outdoor environments," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2005.
15. J. Crisman and C. Thorpe, "UNSCARF, a color vision system for the detection of unstructured roads," in *Proc. IEEE Int. Conf. Robotics and Automation*, 1991, pp. 2496–2501.
16. C. Rasmussen, "Combining laser range, color, and texture cues for autonomous road following," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2002.
17. J. Zhang and H. Nagel, "Texture-based segmentation of road images," in *Proc. IEEE Intelligent Vehicles Symposium*, 1994.
18. H. Dahlkamp, A. Kaehler, D. Stavens, S. Thrun, and G. Bradski, "Self-supervised monocular road detection in desert terrain," in *Robotics: Science and Systems*, 2006.
19. I. Ulrich and I. Nourbakhsh, "Appearance-based obstacle detection with monocular color vision," in *Proceedings of the National Conference on Artificial Intelligence*, 2000.
20. J. Borenstein and Y. Koren, "The vector field histogram: Fast obstacle avoidance for mobile robots," *IEEE Transactions on Robotics and Automation*, 1991.
21. M. Isard and A. Blake, "Condensation – conditional density propagation for visual tracking," *Int. J. Computer Vision*, vol. 29, pp. 5–28, 1998.
22. D. Stavens and S. Thrun, "A self-supervised terrain roughness estimator for off-road autonomous driving," in *Proceedings of the Conference on Uncertainty in AI*, 2006.
23. International Mountain Bicycling Association, "Trail difficulty rating system," available at `http://www.imba.com/resources/trail_building/itn_17_4_difficulty.html`. Accessed September 12, 2007.
24. M. Varma and A. Zisserman, "A statistical approach to texture classification from single images," *Int. J. Computer Vision*, 2005.
25. P. Felzenszwalb and D. Huttenlocher, "Efficient graph-based image segmentation," in *Int. J. Computer Vision*, 2004.
26. L. Wolf, X. Huang, I. Martin, and D. Metaxas, "Patch-based texture edges and segmentation," in *Proc. European Conf. Computer Vision*, 2006.
27. R. Manduchi, A. Castano, A. Talukder, and L. Matthies, "Obstacle detection and terrain classification for autonomous off-road navigation," *Autonomous Robots*, 2005.