Integrating Stereo Structure for Omnidirectional Trail Following

Christopher Rasmussen Yan Lu Mehmet Kocamaz

***Abstract*— We describe a system which follows "trails" for autonomous outdoor robot navigation. Through a combination of appearance and structural cues derived from stereo omnidirectional color cameras, the algorithm is able to detect and track rough paths despite widely varying tread material, border vegetation, and illumination conditions. The approaching trail region is modeled as a circular arc segment of constant width. Using likelihood formulations which measure color, brightness, and/or height contrast between a hypothetical region and flanking areas, the tracker performs a robust randomized search for the most likely trail region and robot pose relative to it with no *a priori* appearance model. The addition of the structural information, which is derived from a semi-global dense stereo algorithm with ground-plane fitting, is shown to improve trail segmentation accuracy and provide an additional layer of safety beyond solely ladar-based obstacle avoidance. Our system's ability to follow a variety of trails is demonstrated through live runs as well as analysis of offline runs on several long sequences with diverse appearance and structural characteristics using ground-truth segmentations.**

I. Introduction

Roughly linear terrain features such as roads, hiking trails, rivers, powerlines, and pipelines are common in man-made and natural outdoor environments. Such features can be navigationally useful to unmanned ground or aerial vehicles in that they both "show the way" and "smooth the way". Finding and keeping to a path by driving along it or flying above it can simplify an autonomous robot's perceptual and motion planning tasks and mitigate hazards which occur in general cross-country navigation. The relative narrowness and continuity of such features implies a certain commonality in the framework of detection, tracking, and control, but each path type has unique appearance and structural characteristics worthy of investigation.

In this paper we describe a robotic system (shown in Figure 1(a)) for following hiking and mountain-biking trails through varied field and forest terrain. Our system relies upon color *and* stereo vision to discriminate the drivable region ahead. We assume that the trail is everywhere traversable with a wheeled vehicle, and also that the trail is non-branching and non-terminating, removing the necessity of intersection or dead-end detection. In essence, the task is analogous to "lane keeping" from autonomous road following, involving repeated estimation, or tracking, of the gross shape and appearance attributes of a previously-found trail.

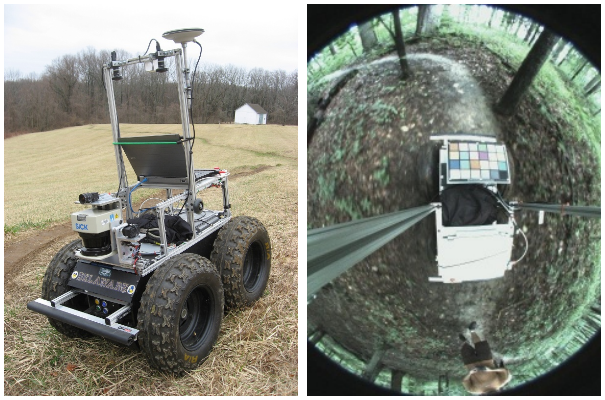


Fig. 1. (a) Robot in park testing area; (b) View from left omnidirectional camera in *forest* trail section (see Section V for explanation of dataset)

The first two DARPA Grand Challenges required vehicles to follow rough roads, but GPS and ladar were sufficient for most successful teams [1], [2]. The DARPA Urban Challenge required more road shape estimation ability, and several teams detailed approaches using primarily vision [5] and rich structural information based on a Velodyne ladar [6].

In the DARPA LAGR program robots had stereo vision instead of ladar and were looking only for open space on their way to a GPS goal, although in constrained areas this was often coincident with path following. Along the lines of [3], a method to learn long-range obstacle appearance from short-range stereo labels was given in [8]. Among LAGR-derived work, [9] and [10] stand out for explicitly looking for path-like corridors of homogeneous color or texture along the ground. The European ELROB competitions have also required path-following skills; one robot effectively followed paths by finding "passages" among scattered trees in ladar data [11]. An approach to non-parametric trail detection using color + intensity saliency maps and agents was recently presented in [12] and extended to tracking in [13].

We reported on an initial version of our omnidirectional trail-following system in [14]. That paper discussed a strictly monocular, *appearance*-based approach to discriminating and tracking the oncoming trail region in an image sequence, coupled with differential motion planning within the parametrized trail region while taking into account ladar-detected obstacles. In this paper we introduce an approach to using stereo-derived scene *structure* estimates as an additional cue at the trail segmentation stage, as well as to

The authors are with the Dept. of Computer & Information Sciences, University of Delaware, Newark, DE, USA. Their e-mail addresses are cer@cis.udel.edu, yanlu@udel.edu, and kocamaz@udel.edu, respectively.

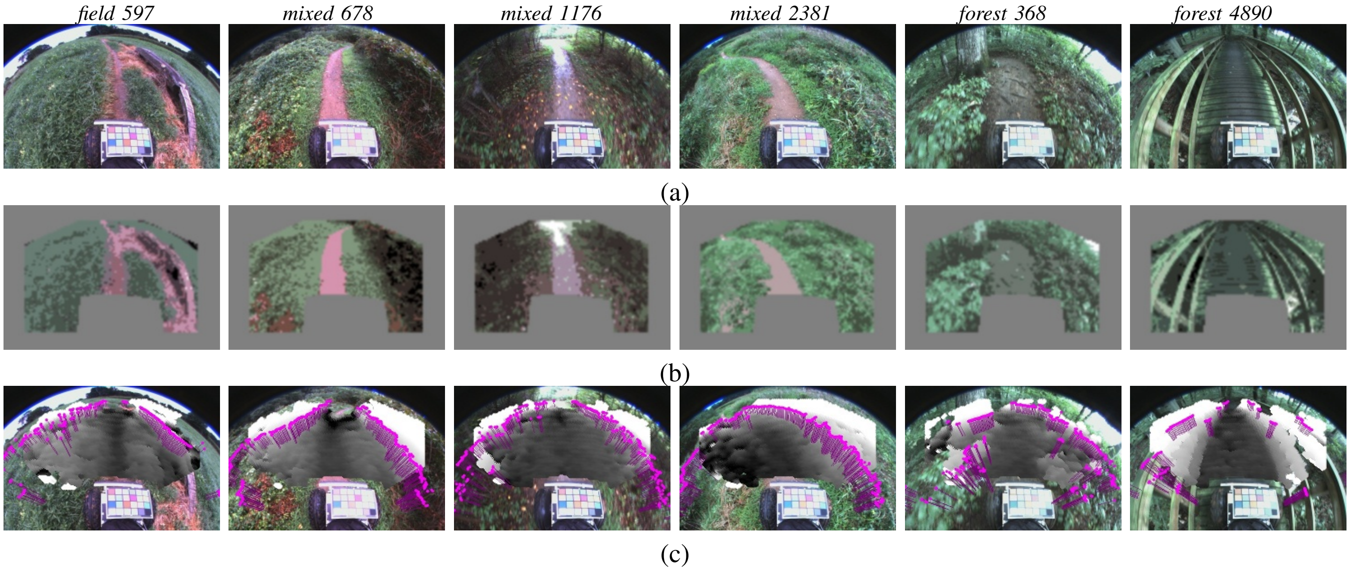


Fig. 2. Selected trail sections in summer (left camera view): (a) Raw images, (b) CIE-Lab color clusters (see Section III), (c) Reprojected dense stereo height maps with ladar hits overlaid (see Section IV-A). Brighter is higher, with saturation at a max of 1 m.

complement ladar-based obstacle detection for motion planning. We have previously reported [15] on a simple technique to augment appearance information about the likelihoods of various trail shape estimates with obstacle density in the image domain (after projecting SICK ladar obstacle detections).

Here we revisit and update that earlier approach using dense stereo, computed over the forward-facing part of our omnidirectional cameras' fields of view (a sample full image is shown in Figure 1(b)). The stereo step yields pixel-level height estimates in vehicle coordinates which are by default registered to the appearance (aka color) information. Some examples of scene images and their associated height maps are shown in Figure 2(a) and (c). Our core hypothesis is that in scenes with low color contrast between on- and off-trail regions, height differences due to nearby bushes, rocks, and trees may help resolve ambiguities. In this paper we offer evidence that by fusing appearance and structural cues we are indeed able to achieve more accurate trail segmentations in many situations and boost overall performance. In addition, stereo height maps offer more information to the motion planning module about potential in- and near-trail hazards which may be missed by the ladar.

In the following sections we briefly review the appearance-based trail detection and tracking components introduced described in [14] before describing in detail how trail scene structure is computed and integrated in the tracking pipeline. We then present results comparing the system with and without stereo-derived structural information, give results from some live runs, and discuss ongoing work and plans for further improvements.

II. Equipment

The sensors used for the results in this paper are two Point Grey Flea2 color cameras and a SICK LMS 291 ladar. Each camera is mounted about 1.15 m off the ground, pointed straight down and rotated so that the longer axis of its CCD is oriented in the direction of vehicle travel. The baseline between them is 0.2 m. The cameras are fitted with omnidirectional Fujinon FE185C046HA-1 lenses which provide a field of view (FOV) of $180°$ along the vehicle Z axis and $145°$ along the X axis. In these experiments the cameras were set for auto-exposure and auto-white balance. All images were captured at 640×480 and downsampled as noted for different vision modules. The SICK ladar is mounted on the robot about 0.5 m off the ground facing forward with a sweep plane parallel (by default) to the XZ (i.e., ground) plane. Its FOV is $180°$ and the maximum range is set to 8 m.

The robot used is a Segway RMP 400, with four-wheel differential steering. The robot's primary computer for image processing, tracking, and motion planning is a Dell Precision M4500 with an Intel Core i5 520M 2.40 GHz processor and 4 Gb of RAM. For live experiments a second computer (a Dell Precision M2400 laptop with an Intel Core Duo T9600 2.80 GHz processor and 4 Gb of RAM) was connected in an onboard LAN to increase performance by handling all low-level device drivers and associated logging.

III. Background

As described in [14], the trail region $\mathcal{R}$ immediately in front of the robot is approximated as a constant-width w arc of a circle with curvature κ over a fixed arc range $[d_{\min}, d_{\max}]$. The position of the robot with respect to the trail is given by its lateral offset Δx from the trail centerline and the difference θ between its heading angle and the tangent to the trail arc. Concatenating the intrinsic width and curvature shape variables with the extrinsic offset and heading error variables, the current *trail state* $\mathbf{X}$ is the 4-parameter vector $(w, \kappa, \Delta x, \theta)$. A sample trail region with

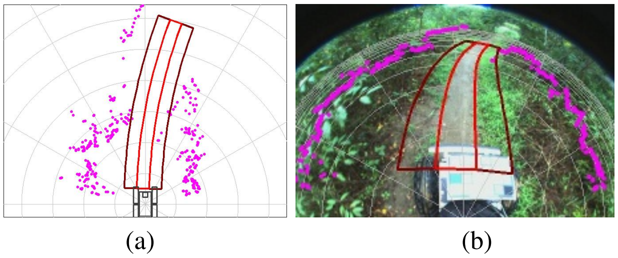


Fig. 3. (a) Robot with candidate trail region and neighboring regions, ladar hits (grid circles are at 1 m intervals); (b) Candidate trail region, ladar projected to camera

$d_{\min} = 0.5$ and $d_{\max} = 6.0$ is diagrammed in Figure 3(a) in vehicle coordinates and projected to the left omnidirectional camera image in Figure 3(b).

Under the assumption that a unique trail is present in each image, it is segmented in a top-down, maximum likelihood fashion: multiple candidate regions are hypothesized and scored using a *trail likelihood* function L, and the highest-scoring region is the winner. For efficiency, we currently only search in the forward half of each omnidirectional image. Because trail-following entails tracking the trail region over an image sequence, we use particle filtering [16] to incorporate a prior $p(\mathbf{X_t}|\mathbf{X_{t-1}})$ on the hypotheses which keeps them near the predicted location of the trail in the current frame as derived from the robot's dynamics. To limit the size of the search space, absolute limits are also set on w and κ based on any knowledge of the trail properties, as well as on Δx and θ under the assumption that the robot is on or close to the trail.

In [14], [15] we presented a technique for computing the color appearance likelihood of a candidate region $L_{appear}(\mathcal{R})$ based on the assumption that the trail region has a strong *color* and/or *intensity* contrast with the left and right neighboring regions $\mathcal{R}_L$ and $\mathcal{R}_R$. This method does not assume that the trail color distribution is known *a priori*, and thus works on a wide range of trail types without training, and during tracking is quite robust to sudden changes in trail material, ambient illumination, or camera exposure. Briefly, following [9] we compute a small set of exemplar colors for each image using k-means clustering in CIE-Lab space and assign every pixel one of these k labels (this labeling is illustrated in Figure 2(b)). A label histogram is computed for each candidate region and its neighbors, and the likelihood is obtained as a weighted combination of *contrast* (as measured by the χ^2 distance between the region and its neighbors) and *homogeneity* (the entropy of the region color distribution).

IV. Integrating Stereo Structure

The color/intensity contrast between the trail region and neighboring regions depends heavily on the trail material and surrounding terrain and vegetation. While it is sufficient in many situations, when the contrast becomes too low trail tracking may become degraded or fail entirely. An additional cue afforded by stereo cameras which may compensate in these situations is that of *height*. Intuitively the trail region itself is expected to be relatively smooth while off-trail regions are rougher (i.e., have higher height variance). Moreover, there is often a measurable contrast between the mean height of the trail and the mean height of regions immediately bordering it, whether due to grass, bushes, or rocks that do not exist in the trail or because a "trough" in the soil has been formed from the passage of previous hikers and bikers. This is clearly visible in Figure 2(c) for *field 597* and *mixed 678*, where the height difference between the trail dirt and neighboring grass is only a few inches.

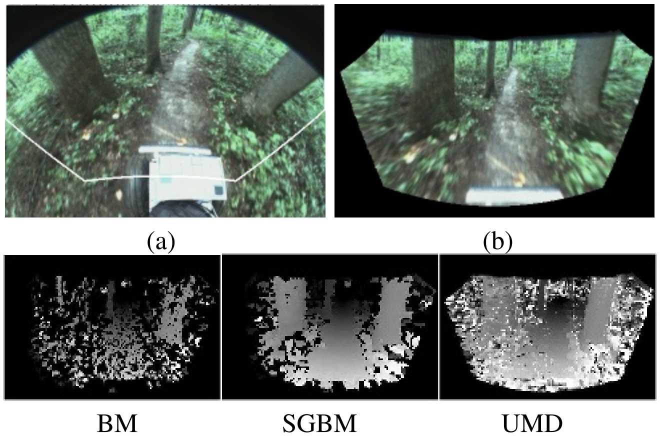
The essential idea of the structure cue is analogous to the appearance likelihood discussed above: subject to shape constraints, look for an image region which has high height contrast with adjacent regions while exhibiting interior smoothness. Below we outline how pixel heights are computed from stereo images and present several variants of a structure likelihood function $L_{structure}$. For the integrated results in this paper we set the likelihood of each trail hypothesis in the particle filter to be a linear combination of its appearance and structure likelihood with appropriate weighting.

A. Height map computation

A quality stereo depth map estimation depends first on accurate calibration. We used the OCamCalib Omnidirectional Camera and Calibration Toolbox for Matlab [17] to obtain intrinsics for the two cameras. Relative extrinsics were initially estimated with manual measurements and then refined with bundle adjustment using *levmar* [18].

Following a common approach to computing correspondences in omnidirectional imagery [19], [20], [21], we rectify the relevant portion of each omnidirectional image into a virtual perspective image such that epipolar lines are image rows; mask out the robot chassis, sensor mast, and peripheral pixels which are not imaged or severely distorted due to the fisheye lens; and then apply a standard pixel correspondence algorithm. The target region of a sample left camera image is outlined in Figure 4(a); this corresponds to a horizontal FOV of approximately 114° and a vertical FOV of 91°. Its rectification after masking is shown in Figure 4(b).

We experimented with three correspondence methods: the default block matching function (BM) in OpenCV [22], a semi-global block matching function (SGBM) [23] which was recently added to OpenCV, and another approach which imposes global consistency constraints (UMD) [24]. A variety of disparity ranges and window sizes were investigated. Sample disparity maps recovered by BM, SGBM, and UMD for the scene in Figure 4 are shown below it. The average running times of the methods on 480×320 rectified images were about 0.03 s for BM, 0.2 s for SGBM, and over 5.2 s for UMD. Judging from many sample image pairs from our dataset, BM produces somewhat sparse and noisy results, SGBM a denser and smoother version of BM, and UMD yields comparable results to SGBM but with more noise. There seems to be sufficient texture in most scenes for reasonable depth recovery, but blurring near the edges of the lens and due to robot motion are sometimes problematic.


(a)

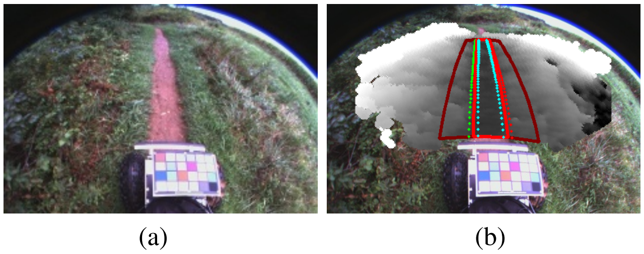
(b)

BM

SGBM

UMD

Fig. 4. (a) Sample left camera image of stereo pair with area to be rectified outlined; (b) Rectified subimage after masking. Bottom row: recovered disparity maps using methods given in the text.


(a)

(b)

Fig. 5. Structure likelihood function. (a) Sample image; (b) Left and right height snakes overlaid on height map

Considering both quality and time, we chose SGBM for disparity recovery in our paper.

B. *Structure likelihood*

Given the disparity map and camera calibration, we can reproject each pixel to vehicle coordinates. Isolating the *height* or Y value of each pixel relative to a nominal ground plane produces a height map which forms the basis for structure calculations. Calculated height maps relative to a ground plane of $Y = 0$ for a set of sample scenes are shown in Figure 2(c).

Computing heights relative to a fixed ground plane such as $Y = 0$ can be problematic when the the trail ahead *slopes*. It is standard to first fit a ground plane to stereo data for obstacle detection [8], [9], but some of our data makes this step complicated. In areas with considerable height variation due to foliage, the ground region often occupies a minority of the image, breaking common techniques like RANSAC. Therefore, we have devised a structure likelihood formulation called *height snakes* which does not directly rely on a ground plane fit while still being insensitive to slope. Rather than comparing region statistics, we directly measure lateral height differences at the trail region-to-neighbor boundaries and penalize for excessive distal height differences inside the trail region.

Specifically, the left and right *edges* of the trail region are discretized into n depth intervals along the arc from $[d_{\min}, d_{\max}]$, indicated by dots in Figure 5(c). A short line segment orthogonal to the edge at the ith depth interval extends from a point just inside the trail region to just outside. Let $\Delta y_{lat}(\mathcal{R}_L)$ and $\Delta y_{lat}(\mathcal{R}_R)$ be the set of absolute inside-outside height differences over all n intervals along the left edge (cyan-green dot pairs in Figure 5(c)) and right edge (cyan-red dot pairs), respectively. Then letting $\Delta y_{dist}(\mathcal{R}_L)$ and $\Delta y_{dist}(\mathcal{R}_R)$ be the set of absolute height differences between the points at depth intervals i and $i + 1$ on the *inside* of each edge, we have:

$$\begin{aligned} L_{\textit{structure}}(\mathcal{R}) = &\textit{Med}(\Delta y_{lat}(\mathcal{R}_L)) - \textit{Med}(\Delta y_{dist}(\mathcal{R}_L)) + \\ &\textit{Med}(\Delta y_{lat}(\mathcal{R}_R)) - \textit{Med}(\Delta y_{dist}(\mathcal{R}_R)) \end{aligned}$$

where $\textit{Med}(\cdot)$ is the median over a set of height differences. Using the median rather than the max ensures that isolated obstacles or errors in stereo matching do not have excessive influence.

C. *Obstacle detection*

Another application of the stereo height maps is in our motion planning module, discussed in detail in [14]. Currently only ladar-detected obstacles are used to modify the default trail centerline-following behavior. However, because the ladar is unable to "see" obstacles above or below its scan plane, it has significant blind spots. For example, in Figure 2(c) *mixed 2381*, there is a steep drop-off on the left side of the trail that the ladar does not see. Similarly, because of its mounting height the ladar scan of the bridge railing in *forest 4890* shows only the vertical supports and not the horizontal planks. The prominent rock from *mixed* in Figure 7 is also too short for the ladar to see. All of these missed hazards show up well in the stereo height map. Besides discrete collision hazards another variable that stereo makes available to the motion planner is *slope*, especially side slope which presents a rollover hazard.

V. Experiments

Our main testing area for trail tracking is a network of combined hiking/mountain-biking trails in a mid-Atlantic U.S. state park which we will term *WCC*. The *WCC* trail from which this paper's data is taken is a ~ 1.7 km long loop and can be logically broken into three contiguous sections comprising (1) open, grassy fields; (2) a mixture of dense bushes and shorter trees, some overhanging; and (3) proper forest with relatively sparse understory foliage. As shorthand, we refer to these segments as *field* (0.6 km), *mixed* (0.4 km), and *forest* (0.7 km), respectively. The entire loop with the sections marked is shown in Figure 6. Depending on the season, variations in plant growth can considerably affect the visual properties of the different trail sections and thus the difficulty of trail segmentation. Several pairs of images captured at the same points on the trail in late summer and in late winter are shown in Figure 7.

A. *Summer, offline*

For these results, data was collected at *WCC* from about 1 km of manual driving in a clockwise direction along the

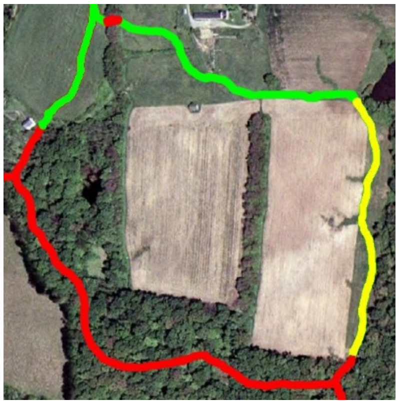


Fig. 6. Aerial image of $\sim$ 1.7 km *WCC* trail loop. *field* segments are shown in green, *mixed* in yellow, and *forest* in red.

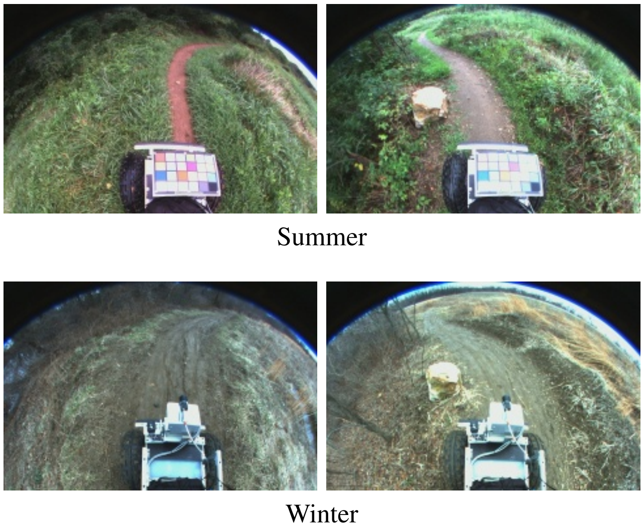


Fig. 7. Sample matched locations along *WCC* trail from summer, winter datasets. First column is from transition from *field* to *mixed*; second column is from middle of *mixed*

trail, starting near the northwest corner, in late summer. All processing was done offline, enabling comparison of the different likelihood methods described in preceding sections without regard for computational cost.

For each trail section of our dataset, we have manually generated ground-truth polygonal segmentations at regularly-spaced intervals. Out of about 17,000 total image frames captured at 10 Hz, we have ground truth for 436, or about 1 in 40. State space trail parameters are extracted from the trail region image polygons by projecting them to vehicle coordinates and performing circle fitting and an independent width fitting procedure. This allows us to monitor tracker performance for each state variable by directly measuring the median absolute error in the trail heading, width, lateral offset, and curvature estimates.

A useful measure of agreement between the tracked trail region and the ground truth trail region in the image can be computed using a polygon overlap formula suggested by [26]: $Overlap(\mathcal{R}_1, \mathcal{R}_2) = A(\mathcal{R}_1 \cap \mathcal{R}_2)^2 / (A(\mathcal{R}_1) A(\mathcal{R}_2))$. To measure overlap with ground truth the raw image polygon is not used, but rather a reprojection to the image of the fitted state space parameters, which tends to result in a smoother polygon and which can be generated for any desired $[d_{\min}, d_{\max}]$ range.

We have run numerous experiments varying the "lookahead" distance $d_{\max}$ from 2 m to 6 m and found little benefit to a larger value for computing Δx and w. This is unsurprising since these are most accurately calculated nearest to the robot. The accuracy of κ goes up slightly with larger $d_{\max}$, but it is θ which sees the most improvement. The overlap score does not change significantly because the more distant portion of the trail is a small fraction of its overall image area.

A summary of results are shown in Table I for the different trail likelihoods discussed above and the three different terrain areas of our dataset. Qualitatively, an appearance-only approach tracks the trail quite well throughout. The width error for the *field* is somewhat high, and mostly accounted for we believe by the k-means clustering often grouping trail dirt pixels with yellow grass growing beside the trail (even for larger k), making the trail look a little wider in the simplified color space searched by the particle filter. This can be seen in the color cluster labels of Figure 2(b) for *field 597*. The structure-only tracker also does quite well with no color information, but it sometimes overestimates the trail width because it sees more free space beside the trail that is not hazardous. This is seen in Figure 2(c) for *mixed 1176*.

Combining the appearance and structure likelihoods leads to an improvement in median overlap scores, and significantly fewer episodes of mistracking due to off-trail distractions. In particular, the tracking problems seen in the *field* section due to similar colors near the trail are significantly mitigated by the structure likelihood locking onto the trail trough in the height map.

		Overlap	θ (degs.)	Δx (m)	w (m)
L_{appear}	*field*	0.443	3.3	0.05	0.12
	mixed	0.839	3.8	0.05	0.04
	forest	0.763	4.5	0.07	0.08
$L_{structure}$	*field*	0.544	3.4	0.05	0.06
	mixed	0.700	4.0	0.06	0.13
	forest	0.520	8.1	0.10	0.19
$L_{appear}+$ $L_{structure}$	*field*	0.579	2.8	0.05	0.08
	mixed	0.817	3.5	0.05	0.06
	forest	0.693	5.0	0.08	0.14
L_{appear} [14] $d_{\max} = 6$ m κ not fixed	*field*	0.361	2.4	0.07	0.20
	mixed	0.774	2.9	0.05	0.05
	forest	0.671	6.6	0.09	0.10

TABLE I

MEDIAN IMAGE OVERLAP SCORES AND ABSOLUTE ERRORS FOR DIFFERENT TRAIL STATE VARIABLES, RELATIVE TO GROUND TRUTH (436 IMAGES TOTAL, $d_{\max} = 2$ M, FIXED STATE $\kappa = 0$)

B. Winter, live

While image overlap and the state parameter errors discussed above provide a useful picture, the ultimate metric is how far the robot can travel along different kinds of trails using the algorithm described. We tested the system by conducting live runs in late winter at *WCC* and two other locations which we will call *Campus* and *Park*. Real-time constraints (our target update rate is 8 Hz) prevented the simultaneous use of appearance and stereo structure information given available computing power, so the robot relied only on color appearance information. The robot speed was fixed at 0.75 m/s.

The *Campus* location is a curving, asphalt-paved pathway about 95 m long and bordered by grass. Significant shadows from trees were cast on about one-third of the path, as seen in Figure 8. The robot was able to traverse this path in both directions several times with no incidents.

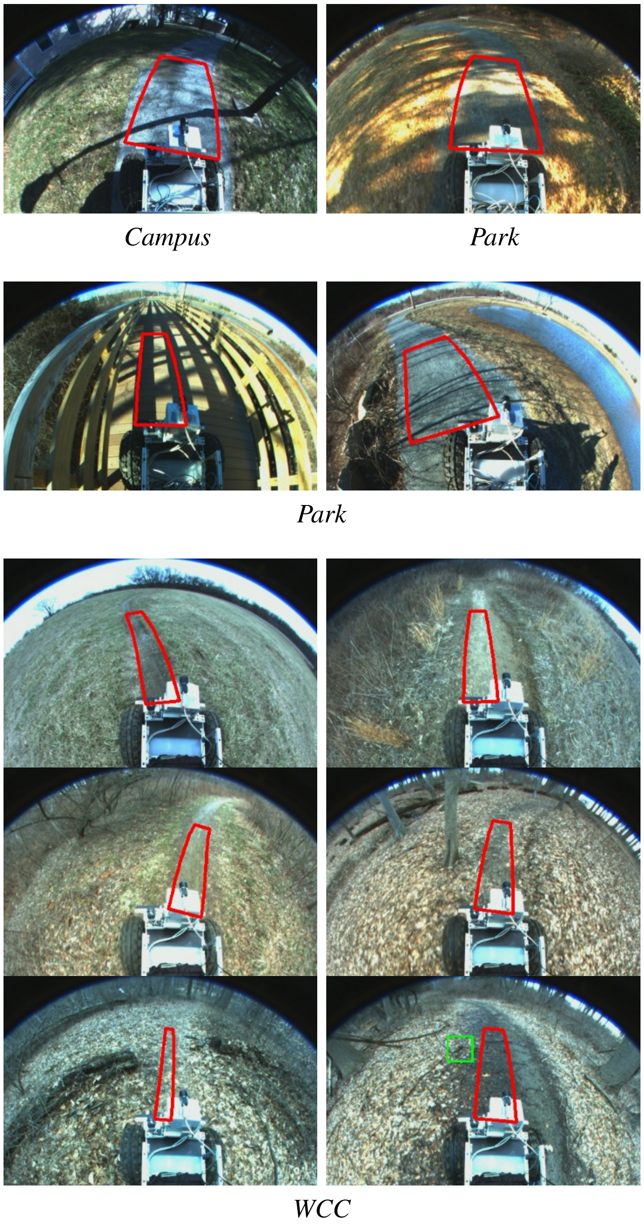
The *Park* location is a gravel path in a county park about 240 m long with a 60 m wooden bridge in the middle and a transition from full sunlight to full shade near the bridge. Testing occurred late in the day, and multiple narrow parallel shadows on the bridge (shown in Figure 8) confused the algorithm, as it expects to see only one high-contrast region–the trail. Nonetheless, it tracked several of these shadows almost all the way across the bridge before being manually stopped because tracking was disrupted by a tree shadow. The robot tracked the sunny section of gravel path on one side of the bridge without incident, and was able to track the shaded section on the other side several times as well, although it once mistracked when the camera exposure changed rapidly as it entered the shady section.

Although the ground was snow-free, at this time of year color contrast is quite low at *WCC* due to dormant vegetation along the *field* and *mixed* sections of trail and significant leaf-fall on the *forest* section. The seasonal difference in trail appearance is clear from the example images in Figure 7.

8 autonomous segments (with restarts as necessary) were run beginning in the middle of the northern *field* section and concluding at the end of the *forest* section. Measuring from GPS logs, a total of approximately 810 m were traveled autonomously, or almost half of the entire loop. Certain large sections were skipped entirely due to off-trail slope hazards that the robot could not see without stereo, but the rest was attempted. The three longest autonomous segments were about 120 m (end of *forest*), 130 m (end of *mixed*), and 310 m (middle of *forest*). Seven of the eight segments were manually terminated because the system eventually mistracked, and one because the robot was about to run into a fallen branch on the trail that was below the ladar scanning height (this branch is highlighted in Figure 8). While tracking performance was definitely less than for the summer data, it was surprisingly good given the visual conditions, and the system performed well physically. The robot maintained control over loose rocks and branches on the trail, through puddles and mud, and climbed and descended nontrivial grades with aplomb.

VI. Conclusion

This paper has presented a system for robotic following of hiking- and mountain-biking-type trails using a combination of visual and ladar cues. The core trail-finder component is fast and robust across a wide range of illumination conditions and types of terrain. The addition of structural information has added a safety margin to the system by reducing the incidence of mistracking that may pose a danger to the robot.



Campus *Park*

Park

WCC

Fig. 8. Sample trail segmentations from winter autonomous runs. The green box in the lower-right image highlights a fallen branch invisible to the ladar but picked up by stereo (see Figure 9)

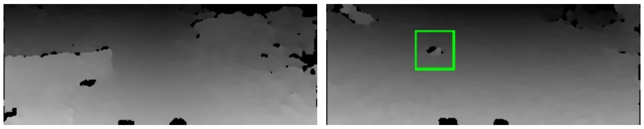


Fig. 9. Detail of stereo disparity maps (computed offline) for the last two images in Figure 8. Clearly visible are the split log on the left and the fallen branch on the right. Stereo will enable longer, safer autonomous runs.

The height contrast cue is a powerful one that is useful in low appearance contrast sections of trail, complementing the system's existing functionality. More optimization work such as multi-threading is necessary to allow the dense stereo to comfortably run in real time alongside the appearance-based tracker, but by reducing the area of the image over which correspondences are computed (as with the windowed versions of the disparity images shown in Figure 9), the system is nearly capable of this already.

Using a fixed linear combination of appearance and structure information is not ideal, as the relative value of each cue varies with terrain. An adaptive approach would allow the system to favor one cue or the other based on which is more "confident" in its solutions. As described in [14], we currently use an adaptive method to govern the relative weight of intensity vs. chromaticity in the appearance likelihood with a discrete variable in the particle filter state denoting which characteristic to histogram over. This has yielded better tracking performance than relying on either cue alone, and we hope to do something similar by mixing structure particles with appearance particles in the future.

Acknowledgments

The authors gratefully acknowledge the support of the National Science Foundation under award 0546410.

References

[1] S. Thrun, M. Montemerlo, *et al.*, "Stanley, the robot that won the DARPA grand challenge," *J. Field Robotics*, vol. 23, no. 9, 2006.

[2] C. Urmson *et al.*, "A robust approach to high-speed navigation for unrehearsed desert terrain," *J. Field Robotics*, vol. 23, no. 8, pp. 467–508, 2006.

[3] 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.

[4] C. Rasmussen, "Roadcompass: Following rural roads with vision + ladar using vanishing point tracking," *Autonomous Robots*, vol. 25, no. 3, October 2008.

[5] A. Huang, D. Moore, M. Antone, E. Olson, and S. Teller, "Multi-sensor lane finding in urban road networks," in *Robotics: Science and Systems*, 2008.

[6] C. Urmson *et al.*, "Autonomous driving in urban environments: Boss and the urban challenge," *J. Field Robotics*, vol. 25, no. 1, 2008.

[7] 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.

[8] R. Hadsell, P. Sermanet, A. Erkan, J. Ben, J. Han, B. Flepp, U. Muller, and Y. LeCun, "On-line learning for offroad robots: Using spatial label propagation to learn long-range traversability," in *Robotics: Science and Systems*, 2007.

[9] M. Blas, M. Agrawal, K. Konolige, and S. Aravind, "Fast color/texture segmentation for outdoor robots," in *Proc. Int. Conf. Intelligent Robots and Systems*, 2008.

[10] G. Grudic and J. Mulligan, "Outdoor path labeling using polynomial mahalanobis distance," in *Robotics: Science and Systems*, 2006.

[11] C. Armbrust, T. Braun, T. Fohst, M. Proetzsch, A. Renner, B. Schafer, and K. Berns, "Ravon — the robust autonomous vehicle for off-road navigation," in *IARP Workshop on Robotics for Risky Interventions & Environmental Surveillance*, 2009.

[12] P. Santana, N. Alves, L. Correia, and J. Barata, "A saliency-based approach to boost trail detection," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2010.

[13] ——, "Swarm-based visual saliency for trail detection," in *Proc. Int. Conf. Intelligent Robots and Systems*, 2010.

[14] C. Rasmussen, Y. Lu, and M. Kocamaz, "Trail following with omni-directional vision," in *Proc. Int. Conf. Intelligent Robots and Systems*, 2010.

[15] ——, "Appearance contrast for fast, robust trail-following," in *Proc. Int. Conf. Intelligent Robots and Systems*, 2009.

[16] A. Blake and M. Isard, *Active Contours*. Springer-Verlag, 1998.

[17] D. Scaramuzza, "Omnidirectional vision: from calibration to robot motion estimation," Ph.D. dissertation, ETH Zurich, Switzerland, 2008.

[18] M. Lourakis, "levmar: Levenberg-Marquardt nonlinear least squares algorithms in C/C++," Available at `http://www.ics.forth.gr/~lourakis/levmar/`. Accessed November, 2009.

[19] N. Winters, J. Gaspar, G. Lacey, and J. Santos-Victor, "Omni-directional vision for robot navigation," in *IEEE Workshop on Omnidirectional Vision*, 2000.

[20] H. Koyasu, J. Miura, and Y. Shirai, "Realtime omnidirectional stereo for obstacle detection and tracking in dynamic environments," in *Proc. Int. Conf. Intelligent Robots and Systems*, 2001.

[21] S. Lin and R. Bajcsy, "High resolution catadioptric omni-directional stereo sensor for robot vision," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2003.

[22] G. Bradski and A. Kaehler, *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly, 2008.

[23] H. Hirschmuller, "Stereo processing by semi-global matching and mutual information," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 25, no. 2, pp. 328–341, 2008.

[24] A. Ogale and Y. Aloimonos, "Shape and the stereo correspondence problem," *Int. J. Computer Vision*, vol. 65, no. 3, pp. 147–162, 2005.

[25] M. M. C. Leger and J. Biesiadecki, "Overview of the mars exploration rovers' autonomous mobility and vision capabilities," in *ICRA Space Robotics Workshop*, 2007.

[26] S. Sclaroff and L. Liu, "Deformable shape detection and description via model-based region grouping," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 23, no. 5, 2001.

[27] C. Rasmussen, "Shape-guided superpixel grouping for trail detection and tracking," in *Proc. Int. Conf. Intelligent Robots and Systems*, 2008.