Trail Following with Omnidirectional Vision

Christopher Rasmussen Yan Lu Mehmet Kocamaz

***Abstract*— We describe a system which follows "trails" for autonomous outdoor robot navigation. Through a combination of visual cues provided by stereo omnidirectional color cameras and ladar-based structural information, 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 simply modeled as a circular arc of constant width. Using an adaptive measure of color and brightness 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. Stereo visual odometry improves tracker dynamics on uneven terrain and permits local obstacle map maintenance. A motion planner is also described which takes the trail shape estimate and local map to plan smooth trajectories around in-trail and near-trail hazards. Our system's performance is analyzed 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. Whether intentionally or not, all of these features can be navigationally useful to unmanned ground or aerial vehicles in that they both "show the way" and "smooth the way" to a robot following them. Finding and keeping to a path by driving along it or flying above it can greatly 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 and described in detail in Section V) for following hiking and mountain-biking trails through varied field and forest terrain which relies primarily upon vision and secondarily upon ladar to discriminate the drivable region ahead. We assume that the trail is everywhere traversable with a wheeled vehicle (although combining our approach with step planning [1] is an interesting prospect), and also that the trail is non-branching and non-terminating, removing consideration of intersection and 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.

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. 1. View of robot in testing area

The two recent DARPA Grand Challenges (DGC) required vehicles to follow rough roads, but GPS and ladar were sufficient for most successful teams [2], [3]. Vision was not primary for any team, although it was exploited as a means of detecting long-range obstacles for speed control [4] and as a road direction estimator [5]. The navigational tasks in the DARPA Urban Challenge required more road shape estimation ability, and several teams detailed approaches using primarily vision [6] and rich structural information based on a Velodyne ladar [7].

General off-road navigation using vision and ladar was investigated in the DARPA PerceptOR program [8], which preceded the DGC. In the recent 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 [4], a method to learn long-range obstacle appearance from short-range stereo labels was given in [9]. Among LAGR-derived work, [10] and [11] stand out for explicitly looking for path-like corridors of homogeneous color or texture along the ground. The

Fig. 2. Sample trail sections (left camera view). The first image is from the *field* dataset, the next three from *mixed*, and the last from *forest* (see Section VI)

European ELROB competitions have also required path-following skills; one robot effectively followed paths by finding "passages" among scattered trees in ladar data [12].

We reported on an efficient method of segmenting trails in IROS last year [13]. In that paper we took a top-down approach by hypothesizing deformations of a canonical trail *shape* and then scoring them for generalized color contrast. The trail was assumed to be locally straight and only the parameters of its image projection were tracked, meaning that each hypothesized shape was a *triangle*. This worked well on a wide variety of trail images taken from different perspectives and with different cameras, and was sufficient for reactive steering on our robot. However, it was not suited to metric motion planning or the use of vehicle dynamics in the tracker.

In this paper we describe how we have moved the tracking formulation to vehicle coordinates, added curvature to the trail state, and changed to an omnidirectional camera while retaining the core image processing algorithm. There are multiple benefits offered by omnidirectional imagery (representative images from our testing area are shown in Figure 2). First, a wide lateral field of view is beneficial where there is sharp trail curvature or a split, so that all regions of interest are in view simultaneously. Second, by seeing behind the robot, there is more data to confirm trail segmentation, shape estimation, and motion estimation results. Third, self-calibration of sensor placements is simpler with the robot chassis in view, as is dynamic color calibration using a visible color chart.

With this new state formulation we are able to employ a more sophisticated motion planner that resembles the on-road planners used in the Urban Challenge [14], [15]. Furthermore, we describe how stereo imagery from a second omnidirectional camera is used to estimate planar motion via visual odometry in the manner of [16]. Wheel-based odometry is relatively error-prone on sandy trails strewn with rocks and roots, and having visual motion estimates is useful for providing state predictions to the trail tracker as well as maintaining a local map which can be referenced by the motion planner for obstacle avoidance.

In the following sections we will first review the trail detection and tracking foundations of the system which were introduced in [13], then describe each of the new components introduced above, present experimental results and limitations, and finally discuss current and future work.

II. Trail State Estimation

We approximate the trail region $\mathcal{R}$ in front of the robot as a constant-width circular arc with fixed length $d_{\max}$. The intrinsic parameters of the trail are thus its width w and curvature κ. The position of the robot with respect to the trail is characterized 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 intrinsic and extrinsic variables, the current *trail state* $\mathbf{X}$ is a 4-parameter vector $(w, \kappa, \Delta x, \theta)$. A sample trail region is diagrammed in Figure 3(a) in vehicle coordinates and projected to the omnidirectional camera image in Figure 3(b).

Assuming that a trail is present in an image, our method for segmenting it is a top-down, maximum likelihood approach: a number of candidate regions are hypothesized and scored, and the highest-scoring region is the winner. Because trail-following requires us to track the trail region over time, we use particle filtering [17] 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. Dynamics are estimated using stereo visual odometry, described below in Section III. 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 nearly-on the trail.

A. Appearance Likelihood

For a single image we have no *a priori* model of the trail's color or texture. In this case the primary basis of a high appearance likelihood is strong *contrast* between $\mathcal{R}$ and its surround. To be more precise, we define left and right neighboring regions of the trail as $\mathcal{R}_L$ and $\mathcal{R}_R$, respectively, as shown in Figure 3(a) and (b).

A number of different measures have been proposed to measure appearance contrast between image regions. Following our earlier work [13], we adapt a technique from [10]

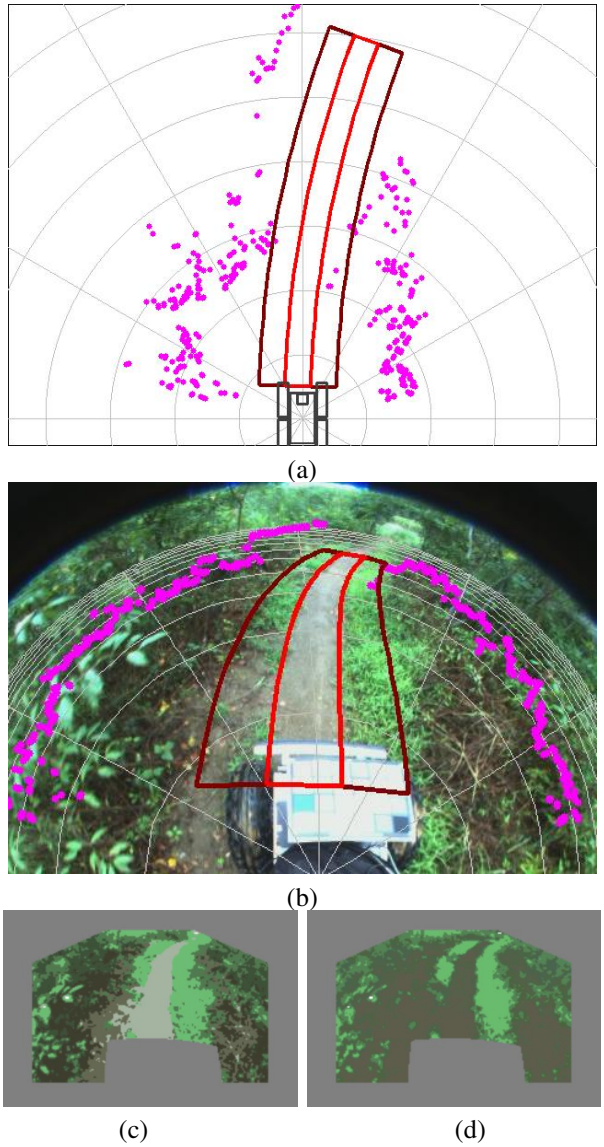

Fig. 3. (a) Robot with candidate trail region and neighboring regions, ladar hits (grid circles are at 1 m intervals); (b) Candidate trail region, ladar projected to camera; (c) **LAB** space k-means labels (colored with mean RGB of each cluster; gray frame is mask); (d) **AB** space k-means labels

based on histograms of k-means cluster labels in CIE-Lab color space. Using this method, a set of color *textons* is first created from the input image by computing an n-dimensional feature vector at each pixel (x, y).

CIE-Lab space is a transformation of RGB space in which the L coordinate encodes lightness or intensity and the a, b coordinates represent chromaticity. CIE-Lab has the advantage of greater perceptual uniformity–in it, Euclidean distance is a somewhat reasonable metric for color similarity. We consider two possible color feature vectors: **LAB**, which uses all three channels, and **AB**, which uses only the chromaticity coordinates and is thus nominally illumination-insensitive (**L**, which uses only brightness, was also tested extensively and found not to be helpful except in desaturated conditions like snow scenes).

Before clustering, textons containing saturated pixels are set aside. k-means is performed on the valid remaining textons to identify a small number of common colors in the image; these are combined with the under- and over-saturated groups to yield $k + 2$ final texton labels $l_1, \ldots, l_{k+2}$. The value used in this paper is $k = 6$.

A trail region $\mathcal{R}$'s color distribution is modeled by a histogram $h = (f_1, \ldots, f_{k+2})$ of the frequencies of the k-means and saturation cluster labels inside it. This allows multi-modal color distributions within trails, which is useful for heterogeneous materials such as leaves or rocks. The appearance dissimilarity between two regions $\mathcal{R}_i$, $\mathcal{R}_j$ is captured by a histogram distance function; we use the common chi-squared metric $\chi^2(h_i, h_j)$.

Rewarding other characteristics such as trail region interior homogeneity is also desirable, since trail regions are often more homogeneous than heterogeneous in color. [18] uses homogeneity but not contrast, while [19], [20] use a ratio of contrast to heterogeneity. Our experiments have indicated that including a homogeneity term in the form of the entropy of the label histogram $H(h)$ is helpful.

We combine the contrast of a hypothetical trail region $\mathcal{R}$ with its left and right neighboring regions $\mathcal{R}_L$ and $\mathcal{R}_R$ and the entropy of the central region as a weighted sum:

$$L_{appear}(\mathcal{R}) = \alpha \frac{\chi^2(h, h_L) + \chi^2(h, h_R)}{2} + \beta(1 - H(h))$$

where the entropy is normalized to $[0, 1]$ based on the number of clusters k. After some experimentation we have found that $\alpha = \beta = 0.5$ gives good results.

B. Cue selection

An issue of interest is whether the choice of features **LAB** or **AB** to cluster with k-means can affect the algorithm. Because of the speed of the procedure, it is easy to simply perform the clustering using each of several alternative sets of cues. Empirically, we have found that on different images and in different lighting situations, which cue is chosen can have a strong effect on the appearance likelihood objective function that we are trying to maximize. Compare Figures 3(c) (**LAB**) and (d) (**AB**), for example. For the same scene, the **LAB** clustering clearly achieves a better contrast between the trail region and the area to the left. The gray frames in the k-means images show the mask which is used to exclude the robot chassis and dark corners of the CCD from image processing, both for accuracy and for efficiency.

Our approach is to do k-means clustering two times, once for each cue alternative, on every image. This results in two different histograms $h_{\mathbf{LAB}}$ and $h_{\mathbf{AB}}$ for every region and thus two different appearance likelihoods for each region hypothesis. To smoothly sample both in image space and cue space, we add a discrete variable to the state in the particle filter denoting whether a particle is to be scored using **LAB** or **AB**. Particles keep their color space labels with a probability p and change to the other label with probability $1 - p$ (we use $p = 0.9$). This allows the population of

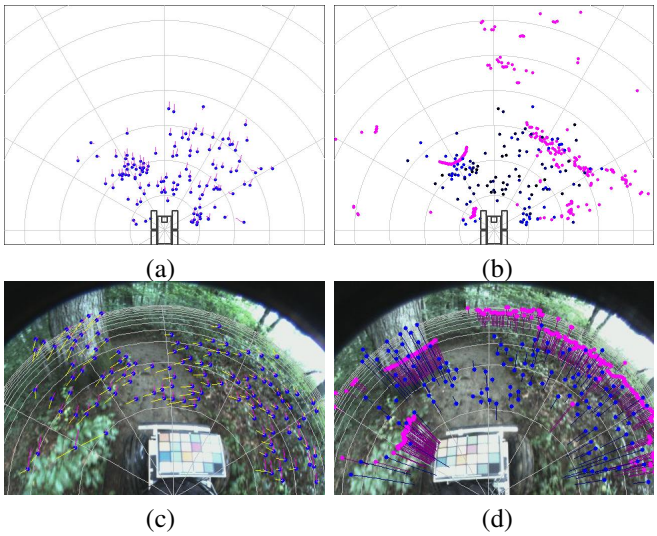

Fig. 4. (a) Triangulated stereo matches which are motion inliers, with tracks; (b) Triangulated inliers with saturation indicating height + ladar hits for reference; (c) Motion inliers in left image and their matches (yellow) and tracks (purple); (d) Motion inliers reprojected into right image + ladar hits, both with vertical height indicators

particles of one kind or another to flourish as illumination and environmental conditions permit, or to maintain parity if neither is favored.

Following this procedure is important for robust functioning of the tracker. Although **LAB** is superior for the majority of images, there are a number for which **AB** is necessary to find the best segmentation.

III. Motion Estimation

We use stereo visual odometry to estimate the frame-to-frame motion of the robot following the basic approaches outlined in [16], [21], [22]. Although the robot undergoes some small pitching and rolling on our testing terrain, our initial experiments have thus far been limited to recovering a planar motion consisting of a rotation and forward translation $(\Delta\theta, \Delta Z)$. Nevertheless, this has been sufficient for the dynamics used in the trail tracker particle filter and for obstacle map maintenance over a $10 - 20$ m scale.

The OCamCalib Omnidirectional Camera and Calibration Toolbox for Matlab [23] was used to obtain intrinsics for the two cameras. Extrinsics were initially estimated with manual measurements and then refined with bundle adjustment using *levmar* [24].

Each camera maintains an independent set of KLT feature tracks [25]. These are tracked from frame t to $t+1$ using only forward matching based on the OpenCV pyramidal optical flow function. The frame-to-frame fundamental matrix is robustly estimated and feature track outliers are initially filtered. Tracks which disappear because of low quality matches or leaving the image frame are replaced randomly from a pool of available KLT features which obey minimum separation from existing tracks.

Mutual matching is performed between the members of the left and right feature track sets at each time step, and matched pairs which do not fit the precomputed left-right fundamental matrix are rejected (although those feature tracks are allowed to continue). The remaining good matches are triangulated, first linearly [26] and then using nonlinear minimization [24].

All N triangulated points at time t which track to points at $t+1$ that are also successfully matched and triangulated constitute a set of putative XZ planar motion vectors. 2-point RANSAC is used to to robustly recover a 2-D rigid transform between frames t and $t+1$, and then nonlinear minimization is performed on the N_I inliers to arrive at a final estimate of $(\Delta\theta, \Delta X, \Delta Z)$. This is projected to the nearest kinematically feasible solution (i.e., ΔX is set to 0). Individual estimates are somewhat noisy, with confidence proportional to the inlier fraction N_I/N, so we perform temporal smoothing. We have gotten good results using a cubic smoothing spline (e.g., the Matlab function `csaps` with $p = 0.01$) where each motion estimate $(\Delta\theta(t), \Delta Z(t))$ is weighted by $N_I(t)/N(t)$ (set to 0 if $N(t) = 0$).

Some sample tracked features and their triangulations are shown in blue in Figure 4. Figures 4(a) and (b) show the triangulated motion inliers in vehicle coordinates. In Figure 4(b) the feature saturation is proportional to height, up to the level of the SICK ladar points, which are shown in purple. Figure 4(c) shows the tracked features in the left image from which the triangulated points were derived, with arrows indicating their matches in the right image (yellow) and tracks from previous images (purple). Figure 4(d) shows the motion inliers reprojected into the right image, with vertical bars indicating their heights off the ground plane. Ladar points are also shown for reference–note, for example, the large tree trunk on the left.

IV. Trajectory Planning

We use the stereo motion information obtained from the methods of the previous section in the tracker dynamics as mentioned in Section II, but also to stabilize recent ladar observations into a local occupancy grid map [27]. This grid map, which travels with the robot, furnishes it with a picture of the constraints on its forward motion as well as a "memory" in case it needs to back up. Samples are shown in Figure 5(a).

Our motion planner is derived from a Dubins car model [28], which accounts for differential constraints on the robot's motion in the form of a minimum turning radius and rules out reverse motion (as we do not have adequate rear-facing obstacle detectors) except in extraordinary circumstances. Under this model, the only maneuvers permitted are sequences of straight-line and left or right circular arc segments (a proportional controller in the motor command handler smoothes transitions between segments). The basic Dubins planner, which works for all start and end (x, y, θ) configurations in the absence of obstacles, is used as the kernel of a lookup-table-like approach to planning along the trail in the presence of obstacles. Briefly, given the currently estimated trail region a single *ultimate* goal pose and a set of nearer *candidate* goals are generated and planned for. Each of these plans is evaluated and possibly pruned based on their

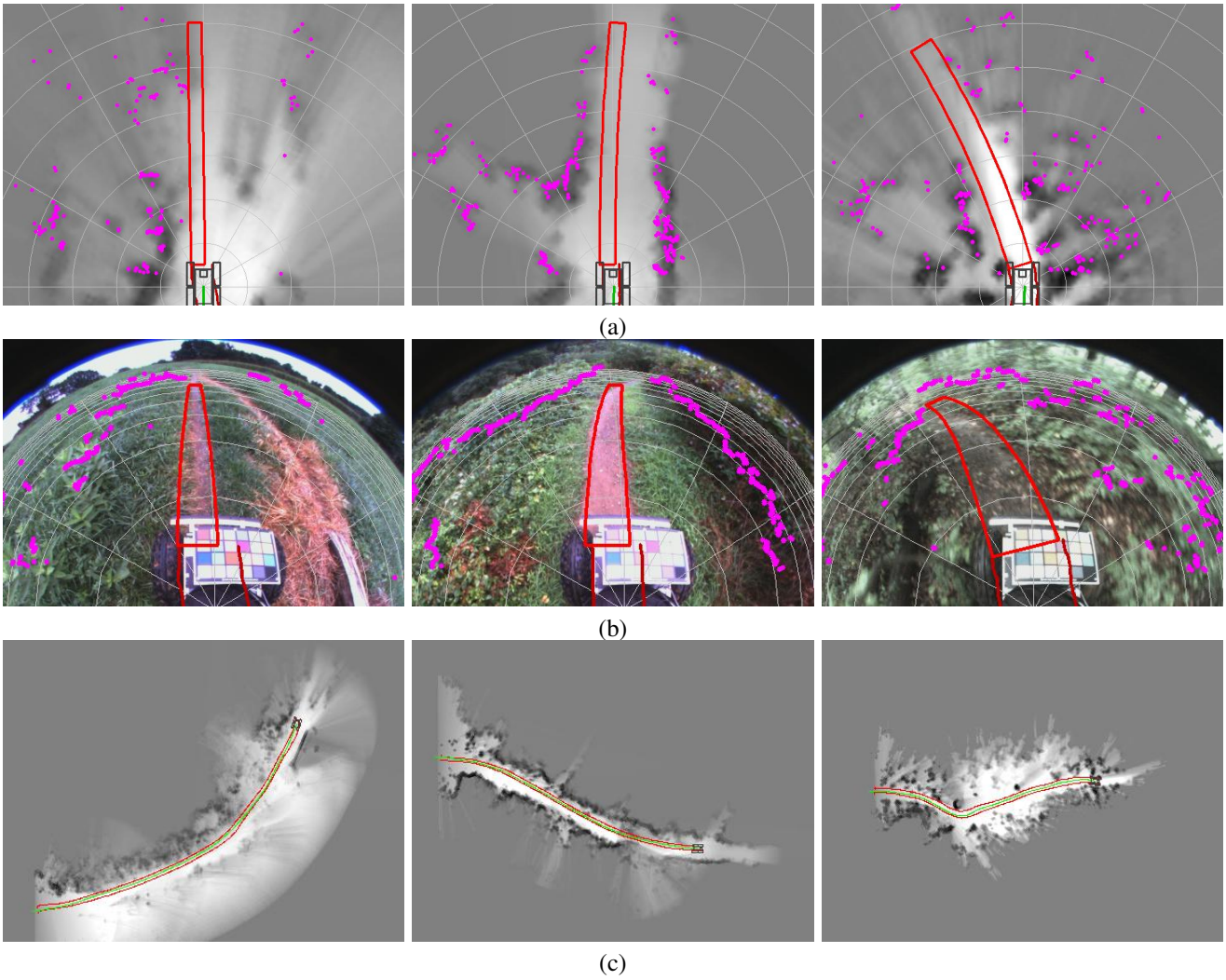

Fig. 5. (a) Local occupancy grids at current robot location created using visual odometry information during runs along *field*, *mixed*, and *forest* (from left to right); (b) Camera view at current location with current tracked trail state overlaid; (c) Partial global maps with trail and position histories overlaid

trajectories colliding with too many obstacles or leaving the trail. From the remaining plans whichever terminates closest to the ultimate goal is selected for execution.

More specifically, given an estimated trail region and an obstacle map, the ultimate goal pose is set to be a point on the trail centerline and tangent to it some constant distance ahead (currently 5 m). If this point is too close to an obstacle, it is moved to the nearest point in free space within the trail. Since the goal is defined relative to the robot's current location, it constantly recedes as the robot advances, and the robot constantly replans its motion.

Candidate goal poses are generated in a regular array spanning the trail region laterally in a series of *mini-lanes* and distally from just in front of the robot out to the ultimate goal, all with θ tangent to the trail. A Dubins plan is constructed from the current robot position to each candidate goal pose, and then *extended* along its mini-lane out to the ultimate goal distance. Selecting candidate goals along the same mini-lane but closer to the robot induces more aggressive lane changes in the manner of "swerves" vs. "nudges" from [2] or "sharp" vs. "smooth" trajectories from [14]. An example set of trajectories are drawn as dark green curves in Figure 6(a). The corresponding camera view, with the candidate motion plans projected, is in Figure 6(e). For clarity, only the sharper trajectories are shown.

If a candidate trajectory leaves the trail because it takes a shortcut across a curved section or requires a 360 degree rotation, it is removed. Under the assumption that the trail is never completely blocked and recognizing that many apparent obstacles are tall grass and "soft" vegetation, the remaining trajectories are ranked by how many obstacles they collide with. Among all trajectories tied for the lowest obstacle density, the one whose endpoint (aka candidate goal pose) is closest to the ultimate goal pose is selected. This provides a centering impulse.

Figure 6(b) shows the robot planning to enter the trail. Trajectories on the left have been removed because they come too close to off-trail obstacles, whereas trajectories

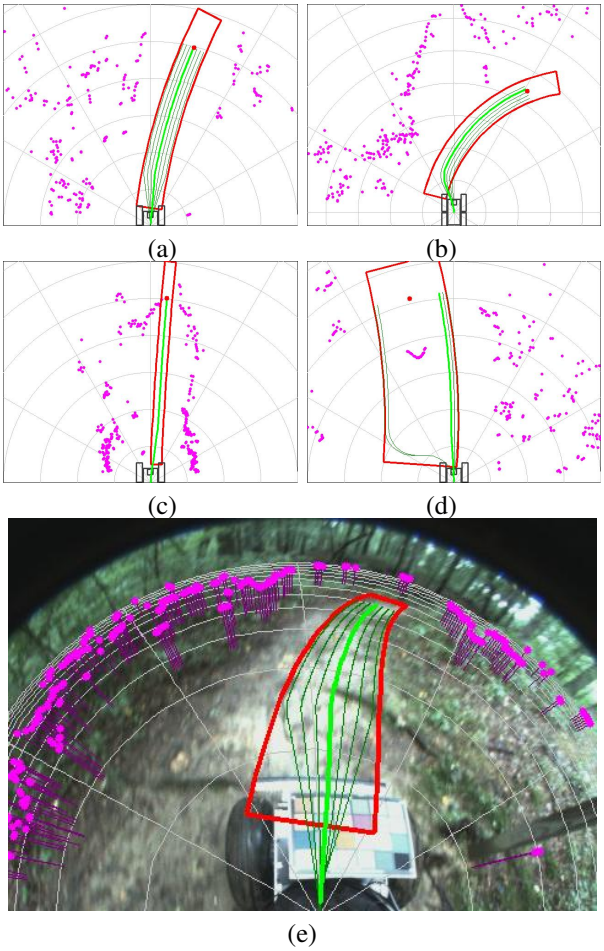

Fig. 6. Motion planning: (a-d) Selected trajectories (green) and candidates (dark green) for sample trail segments. The red dot is the ultimate goal pose. (e) Corresponding camera view for scenario (a) with trail region, trajectories overlaid.

hugging the right side of the trail do not collide. Figure 6(c) shows that even for narrow trail sections a path is always chosen, and Figure 6(d) illustrates how trajectories on either side of an in-trail obstacle may be considered.

V. Equipment

The sensors used for the results in this paper are two Point Grey Flea2 color cameras, a SICK LMS 291 ladar, and a Hokuyo URG-04LX ladar (GPS was not recorded, though the robot is equipped with one). 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 0.5 m off the ground facing forward with a sweep plane parallel to the XZ (i.e., ground) plane. Its FOV is 180° and the maximum range is set to 8 m. The Hokuyo is mounted 1.15 m above the ground facing down in a sagittal orientation (i.e., its sweep plane is the YZ plane). Its FOV is is 240° and its maximum range is 4 m.

The robot used is a Segway RMP 400, with four-wheel differential steering. The robot's primary computer is a Dell Precision M2400 laptop with an Intel Core Duo T9600 2.80 GHz processor and 4 Gb of RAM.

VI. Experiments

Our main testing area for trail tracking is a network of combined hiking/mountain-biking trails in a large regional park in the mid-Atlantic U.S. The trail section from which this paper's data is taken is just over 1 km long and can be logically broken into three contiguous sections of roughly equal length 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 the datasets associated with these segments as *field*, *mixed*, and *forest*, respectively. All data was collected in summer.

The data used here was collected while the robot was being driven manually and all processing was done offline. Successful live integration of earlier generations of the robot's perception and motion planner modules with motor control has been demonstrated in previous published work [29]. Furthermore, earlier versions of the omnidirectional tracker from Section II and motion planner from Section IV were used to win an outdoor robotic path-following competition (`www.igvc.org`) last year against dozens of competitors. For the competition the path was demarcated by lines painted on grass, so the trail likelihood function was edge-based rather than region-based as in II-A. When weather and ground conditions at our testing site permit, in-the-loop experiments will resume.

Our results on the park trail data demonstrate the accuracy and efficiency of the image-only trail finder and tracker of Section II on a diverse set of trail images. For each dataset above, we have manually-generated ground-truth 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. These permit us to quantify the accuracy of our trail region estimates in several ways. First, the notion of region overlap in the image can be characterized with the following polygon area overlap formula suggested by [18]: $Overlap(\mathcal{R}_1, \mathcal{R}_2) = A(\mathcal{R}_1 \cap \mathcal{R}_2)^2 / (A(\mathcal{R}_1) A(\mathcal{R}_2))$.

The median image overlaps with ground truth were 0.361 for the *field* data, 0.774 for the *mixed*, and 0.671 for the *forest*. Qualitatively, the trail was tracked quite well throughout. The last two numbers are quite good, given how difficult many of the images are, but the first number appears relatively low, especially since the field is visually the highest contrast area. We can look more deeply by directly measuring the median absolute error in the trail heading, width, lateral offset, and curvature estimates. These are shown in Table I.

	Overlap	θ (degs.)	Δx (m)	w (m)	κ (1/m)
field	0.361	2.4	0.07	0.20	0.017
mixed	0.774	2.9	0.05	0.05	0.026
forest	0.671	6.6	0.09	0.10	0.065

TABLE I

MEDIAN IMAGE OVERLAP SCORES AND MEDIAN ABSOLUTE ERRORS FOR DIFFERENT TRAIL STATE VARIABLES, RELATIVE TO GROUND TRUTH (436 IMAGES TOTAL)

	ΔZ	$\Delta\theta$
field	0.87	0.73
mixed	0.90	0.82
forest	0.90	0.88

TABLE II

CORRELATION COEFFICIENTS BETWEEN VISUAL ODOMETRY PER-FRAME FORWARD MOTION (ΔZ) AND ANGULAR MOTION (θ) ESTIMATES AND SEGWAY WHEEL ODOMETRY

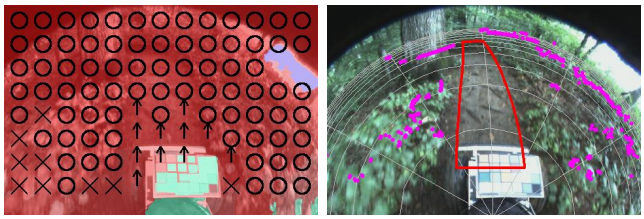

Fig. 7. Output of [30], this paper on selected image from *forest*

The width error for the *field* data was 0.20 m, which was notably high. In this case the median ground truth width was a narrow 0.14 m (a common width for mountain bike paths), which was smaller than a width minimum parameter used in the particle filter. The k-means clustering also often grouped 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. An example of this can be seen in the leftmost image of Figure 5(b). Every other parameter was close to nominal, so it seems that most of the overlap error was due to a consistent overestimate of the trail width.

It is difficult to compare our results directly to other work given the differences in sensors between platforms and objectives of each system. The surface layout classifier of [30] is not a fair comparison, especially with the distortion of the omnidirectional lens, but it is helpful as a baseline. Their system does a a credible job of finding nearby ground in a variety of "normal" images we have previously tested, but roads and trail regions do not seemed to be favored over rougher ground. On our images the output is not useful; one example is shown in Figure 7.

Our visual-odometry-based motion estimates correlate well with conventional odometry derived from the wheel velocities of the Segway RMP 400 base. Plots of per-frame forward motion ΔZ and rotational motion $\Delta\theta$ for the *forest* data are shown in Figure 8, with the visual odometry values shown in red and Segway odometry values shown in green. Pearson's correlation coefficients are shown in Table II ($p < 0.001$ for all values).

VII. 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. Promising preliminary work on incorporating stereo visual odometry into the framework was also discussed, although some work needs to be done to further optimize this module for real-time operation.

In addition to using stereo for motion estimation, we are also investigating stereo-based static obstacle detection as another layer in the local occupancy grid map. The robot's SICK ladar, which is excellent for large obstacle detection, cannot see smaller rocks, logs, or negative hazards such as off-trail drop-offs and ditches. Our feature-based approach to visual odometry is not well-suited to this task, and we are evaluating dense stereo methods that will work with our omnidirectional images.

Thus far we have not used GPS information even where available (maintaining satellite lock can be difficult in areas like *forest*). However, GPS could help smooth motion estimates and in conjunction with trail maps would permit anticipation of forks/intersections. We are also looking at doing full 6-DOF motion estimates from visual odometry for pitch, roll, etc. estimation to make the local map more accurate. Finally, the color checker attached to the robot is not yet being used for color constancy and exposure compensation calculations; these could help considerably.

ACKNOWLEDGMENTS

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

REFERENCES

[1] J. Kolter, Y. Kim, and A. Ng, "Stereo vision and terrain modeling for quadruped robots," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2009.

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

[3] 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.

[4] 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.

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

[6] 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.

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

[8] 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.

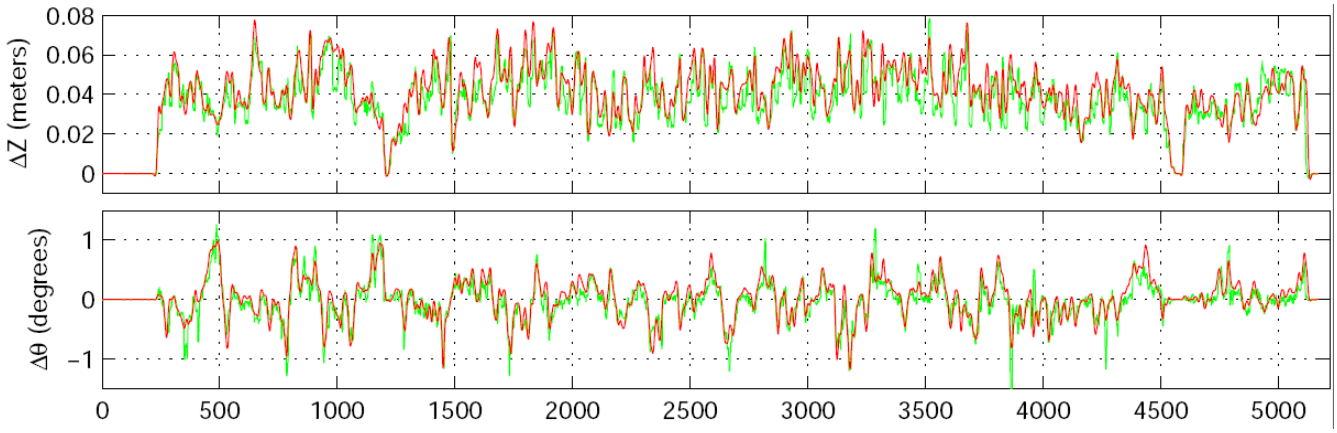

Fig. 8. Top: estimated per-frame forward motion ΔZ using visual odometry (red) vs. Segway odometry (green) for *forest* data; bottom: estimated angular motion $\Delta\theta$.

[9] 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.

[10] 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.

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

[12] 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.

[13] C. Rasmussen, Y. Lu, and M. Kocamaz, "Appearance contrast for fast, robust trail-following," in *Proc. Int. Conf. Intelligent Robots and Systems*, 2009.

[14] D. Ferguson, T. Howard, and M. Likhachev, "Motion planning in urban environments: Part I," in *Proc. Int. Conf. Intelligent Robots and Systems*, 2008.

[15] M. Montemerlo *et al.*, "Junior: The Stanford entry in the urban challenge," *J. Field Robotics*, 2008.

[16] D. Nister, O. Naroditsky, and J. Bergen, "Visual odometry for ground vehicle applications," *J. Field Robotics*, vol. 23, no. 1, 2006.

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

[18] 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.

[19] G. Mori, X. Ren, A. Efros, and J. Malik, "Recovering human body configurations: Combining segmentation and recognition," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2004.

[20] X. Ren and J. Malik, "Learning a classification model for segmentation," in *Proc. Int. Conf. Computer Vision*, 2003.

[21] M. Agrawal and K. Konolige, "Real-time localization in outdoor environments using stereo vision and inexpensive gps," in *Proc. Int. Conf. Pattern Recognition*, 2006.

[22] M. Havlena, T. Pajdla, and K. Cornelis, "Structure from omnidirectional stereo rig motion for city modeling," in *VISAPP*, 2008.

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

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

[25] J. Shi and C. Tomasi, "Good features to track," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 1994.

[26] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[27] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. MIT Press, 2005.

[28] S. LaValle, *Planning Algorithms*. Cambridge University Press, 2006.

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

[30] D. Hoiem, A. Efros, and M. Hebert, "Recovering surface layout from an image," *Int. J. Computer Vision*, vol. 75, no. 1, October 2007.