Kinects for Low- and No-Sunlight Outdoor Trail-Following

Christopher Rasmussen

***Abstract*—We describe work toward a Kinect-based system for tracking "trails" for autonomous outdoor robot navigation. As the trail is often distinguishable from surrounding terrain by its contrasting height or smoothness, the dense and precise structural data provided by the Kinect is very informative about the shape of the trail ahead, and we discuss height- and normal-based approaches which work well over a variety of situations. Furthermore, we show that depth sensing on the Kinect may work outdoors when there is still enough light for the RGB camera to capture the scene as well, allowing appearance-based analysis of the point cloud to identify color contrast trail edges. Our system's ability to segment a wide range of trails is demonstrated through video sequences collected from a mobile robot platform and analyzed offline.**

I. Introduction

Robustly inferring 3-D scene structure has always been a core task for mobile robot navigation and obstacle avoidance, whether through discrete object detection, occupancy grid mapping, or hybrid techniques. The DARPA Grand Challenge (DGC) robots of 2004 and 2005 used ladars almost exclusively in a variety of geometric configurations to sense positive and negative obstacles without regard for their semantic labels [35, 37]. Recent advances in ladars such as the Velodyne have afforded an even richer view of the world and these have been used to great effect in the DARPA Urban Challenge of 2007 [24, 38, 16] and for the Google Car [14].

Stereo vision has also been popular for many reasons including the amount of information obtainable relative to the expense of the sensor. One successful DGC robot used stereo to a considerable extent [7], and stereo was the primary sensor for the DARPA LAGR program [1]. Color and texture information can very informative about drivable regions [10, 36], and appearance and structural cues used in concert often even more so [25, 22, 12, 15, 5].

The Microsoft Kinect is a stereo camera which offers both color and depth information (aka "RGBD"). The depth maps produced by the Kinect are highly accurate over a wide range of scenes because it generates scene texture for stereo correspondence by active laser projection. The accuracy and low price of the Kinect have made it a very compelling sensor. However, a limitation of the Kinect is that sunlight interferes with the pattern-projecting laser, so it is most suitable for indoor robotics. Nonetheless, we will show that outdoor applications are possible when sunlight is sufficiently diminished: at night, in twilight or early morning conditions, with cloudy weather, or in strong shadow.

In this paper we present an approach to using the Kinect for a mobile robot application we have been studying, trail-following. *Trail* is a semantic label for a portion of the scene that is more specific than simply "drivable" or "obstacle-free". 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 simplifies an autonomous robot's perceptual and motion planning tasks and mitigates hazards which occur in general cross-country navigation. In this sense, trail-following is analogous to the "lane keeping" task from autonomous road following, involving repeated estimation, or tracking, of the gross shape and appearance attributes of a previously-found trail.

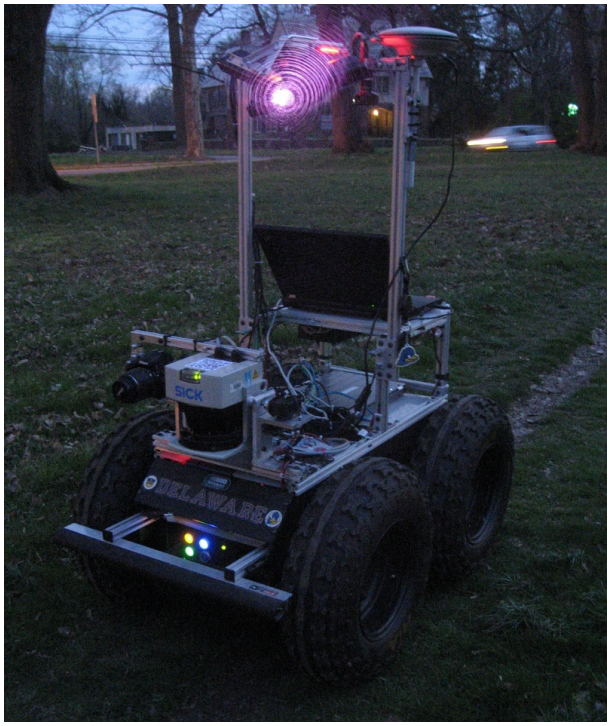

Fig. 1. Robot at night with twin Kinects active

In our early work on trail-following we relied on appearance [26, 27], a very strong cue for differentiating the trail from the background on the basis of color or texture. However, scene structure can also be highly informative, and we have investigated it in more recent work [28, 29] using omnidirectional stereo and a tilting ladar for point cloud acquisition. Vegetation, rocks, trees, walls, ditches, slopes, and other terrain features frequently delimit the trail and thus if protruding

obstacles, whatever their appearance, can be detected, the trail can often be segmented.

For trail segmentation and tracking using only structure information, *height contrast* is an obvious cue. Trails are frequently at a different height than surrounding environmental features such as grass, other vegetation, or other terrain features. However, the trail interior often has height variation itself, and fitting a ground plane does not work well. Another cue is *structural texture*. The distribution of normals on the trail and off the trail may be used as a feature to differentiate them. The simplest possible such feature is normal variance: trails, being engineered, are generally smooth, while the surroundings are bumpier.

A. Related work

Much previous work on the Kinect as a robot sensor can be classified into SLAM-type algorithms which build 3-D metric maps in indoor environments directly from colorized point clouds [17]; and semantic labeling of objects which may include floors, walls, and other navigationally-useful categories [11, 3, 6].

Multi-Kinect rigs have been rare because of the laser interference between adjacent cameras with overlapping fields of view. This interference, which occurs when one Kinect sees the laser pattern of another, is explained and characterized in [32], which also presents a mechanical time division multiple access approach to mitigating it. In [20] three vertically-oriented Kinects with adjacent but not overlapping fields of view were used to create a panoramic imaging system for surveillance and person tracking.

Kinects have not generally been used in outdoor environments because they do not work in direct sunlight. In unpublished work a Kinect was mounted on a drill with a board computer for outdoor image capture around dusk [13]. In [30] a Kinect was mounted on a bicycle and data was collected as the bicyle was ridden on sidewalks and streets "during the day but in illumination conditions that did not affect the performance of the 3D camera", planar patches were fit to the data, and the normals were used to detect undrivable features such as curbs, poles.

There is a considerable body of work on semantic labeling of 3-D point clouds of outdoor scenes acquired from a laser range-finder. A key paper in this area for mobile robots is [19], which classifies points acquired by a terrestrial ladar as flat surfaces (e.g. ground), linear structures such as thin branches or wires, or scattered vegetation based on local characteristics. Similar analysis is often performed on aerial ladar data taken over urban and natural landscapes, where a common task is to perform ground filtering to remove points associated with vegetation and buildings [18, 2]. What is left are "bare earth" points to which a model for the underlying terrain can be fit. The converse problem, of identifying tree points explicitly, was explored in [9]. Such work is relevant to ours because we are also concerned with distinguishing the ground (i.e., the drivable surface) from vegetation on it, and to discern a larger structure (the trail) in the midst of much clutter.

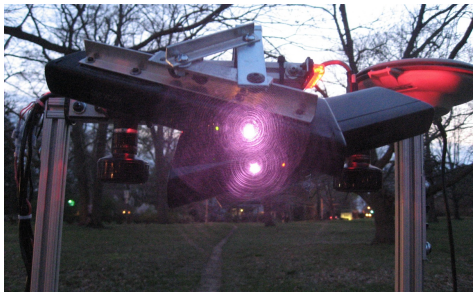

Fig. 2. Close-up of dual Kinect arrangement

II. Methods and Preliminary Results

The robot used for the experiments in this paper is pictured in Fig. 1. It is a Segway RMP 400 with four-wheel differential steering.

A. Depth image capture

Two Kinect cameras were mounted on the robot approximately 1.2 m off the ground, each yawed about 30 degrees out from straight ahead, and tilted down about 45 degrees below horizontal. A close-up is shown in Fig. 2, and the RGB images from each camera for a sample trail scene taken about 25 minutes before sunset are shown in Fig. 3(a). A portion of each Kinect is visible in the other's RGB camera field of view as a thin black triangle, but the depth camera views (e.g., Fig. 3(b)) were not occluded. Black pixels in the depth images represent areas where the depth could not be estimated because of half-occlusion or sunlight interference (this sequence was taken in the early evening close to sunset). With intrinsic and extrinsic calibration parameters of the RGB and depth cameras we can obtain a "colorized" point cloud for the combined rig as shown from an overhead perspective in Fig. 3(c).[1]

The Kinect's field of view is about 57 degrees horizontally by 43 degrees vertically [8], so at least two were deemed necessary to give the robot a sufficient horizontal field of view to not lose sight of the trail at sharp turns. The yaw angles were chosen to make the depth cameras' fields of view adjacent for a panoramic effect while minimizing laser interference between the two cameras as discussed above [32]. This interference manifests itself as noise and missing data in the computed depth images (e.g., the black pixels near the vertical edge between the two images in Fig. 3(b)). Inpainting in the depth image to interpolate missing data is possible [34], but the depth noise is still unsuitable for precise estimates of the normal over small neighborhoods.

[1]The data collected for this work was obtained while the robot was under manual control. It consisted of RGB and depth images from each Kinect, captured on a Lenovo W520 laptop with an Intel Core i7-2720QM CPU and 8 Gb of RAM at about 10 Hz and 640×480 resolution using the `libfreenect` library [23] and downsampled to 320×240 before writing to disk. The depth images were further downsampled to 160×120 for point cloud processing; all further analysis was performed offline.

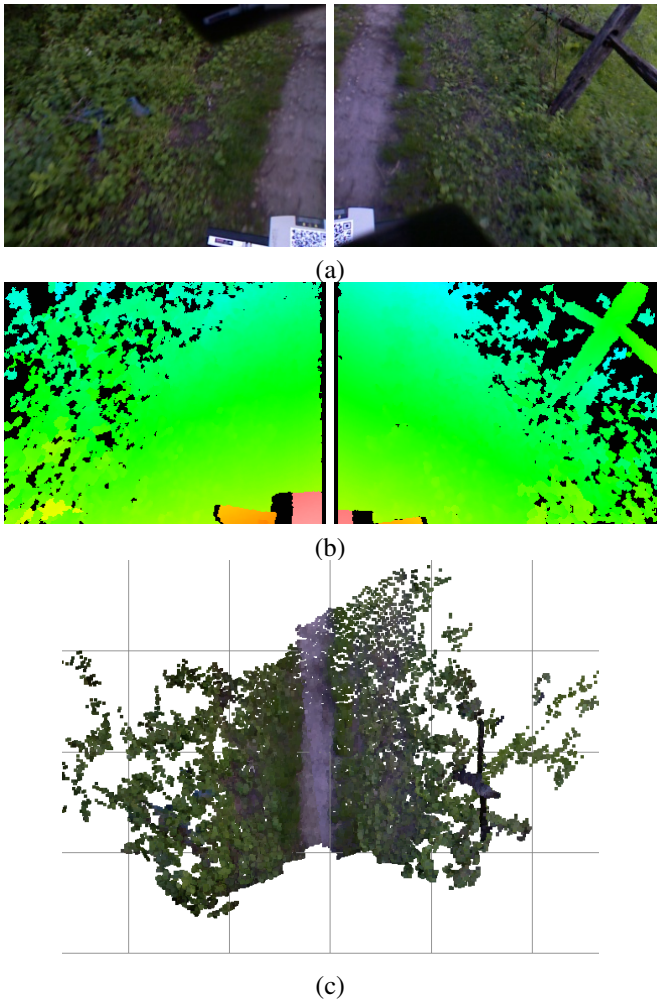

Fig. 3. Sample trail scene, evening. (a) RGB images from left-, right-facing Kinects; (b) Corresponding depth images in false color, with "hotter" colors closer; (c) Overhead perspective view of combined point cloud with RGB values of pixels registered to their 3-D locations

B. Trail state

As described in [27], the trail region $\mathcal{R}$ immediately in front of the robot can be 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 $(w, \kappa, \Delta x, \theta)$. Under the assumption that a unique trail is present in each image, we search for it in a top-down, *maximum likelihood* fashion: multiple candidate regions are hypothesized scored using a *trail likelihood* function L, with the highest-scoring region chosen as the winner.

Because trail-following entails tracking the trail region over an image sequence, we use particle filtering [4] to incorporate a prior $p(\mathbf{X_t}|\mathbf{X_{t-1}})$ on the hypotheses which keeps them

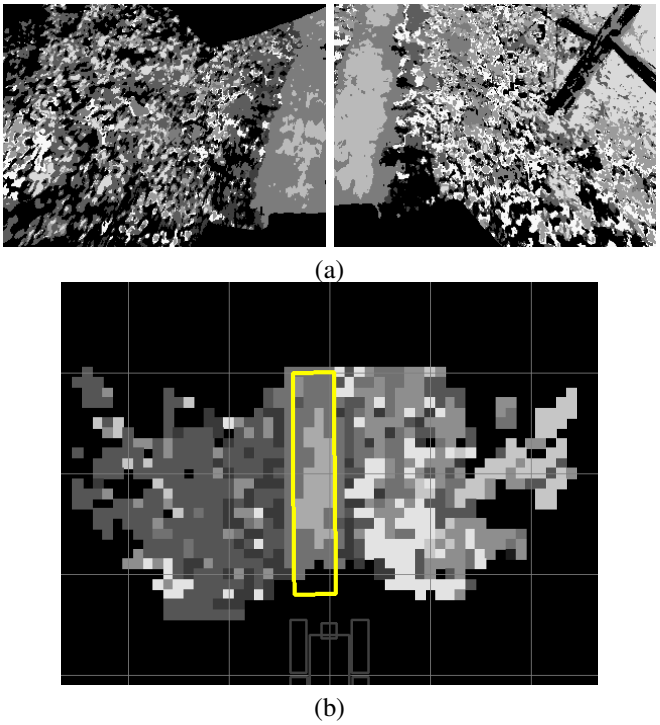

Fig. 4. k-means color labels for sample scene in (a) Original images and (b) Transformed to vehicle-coordinate map with trail estimate overlaid. The mode label for each grid square is shown ($k = 8$).

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 assuming that the robot is on or close to the trail.

C. Color trail detection

In [27, 26] we developed a technique adapted from [5] for computing the color appearance likelihood of a candidate region $L_{appear}(\mathcal{R})$ in a single image based primarily on the assumption that the trail region has a strong *color* and/or *intensity* contrast with left and right neighboring regions $\mathcal{R}_L$ and $\mathcal{R}_R$. To apply this method to the two-camera rig here, we compute a small set of exemplar colors for both Kinect RGB images jointly (after post-processing to try to match their separate exposures) using k-means clustering in CIE-Lab space and assign every pixel one of these k labels. This labeling is illustrated for the sample scene in Fig. 4(a).

The distribution of colors within a given candidate region is characterized by histogramming the labels within it, and thus the *contrast* between that region and its neighbors can be quantified by the χ^2 distance measure or similar. Another distinguishing characteristic of trails is that their color distribution is often more *homogeneous* than the surroundings, and this is quantified with the entropy of the region's exemplar color histogram. The full likelihood of a particular candidate region is then obtained as a weighted combination of the contrast and homogeneity factors.

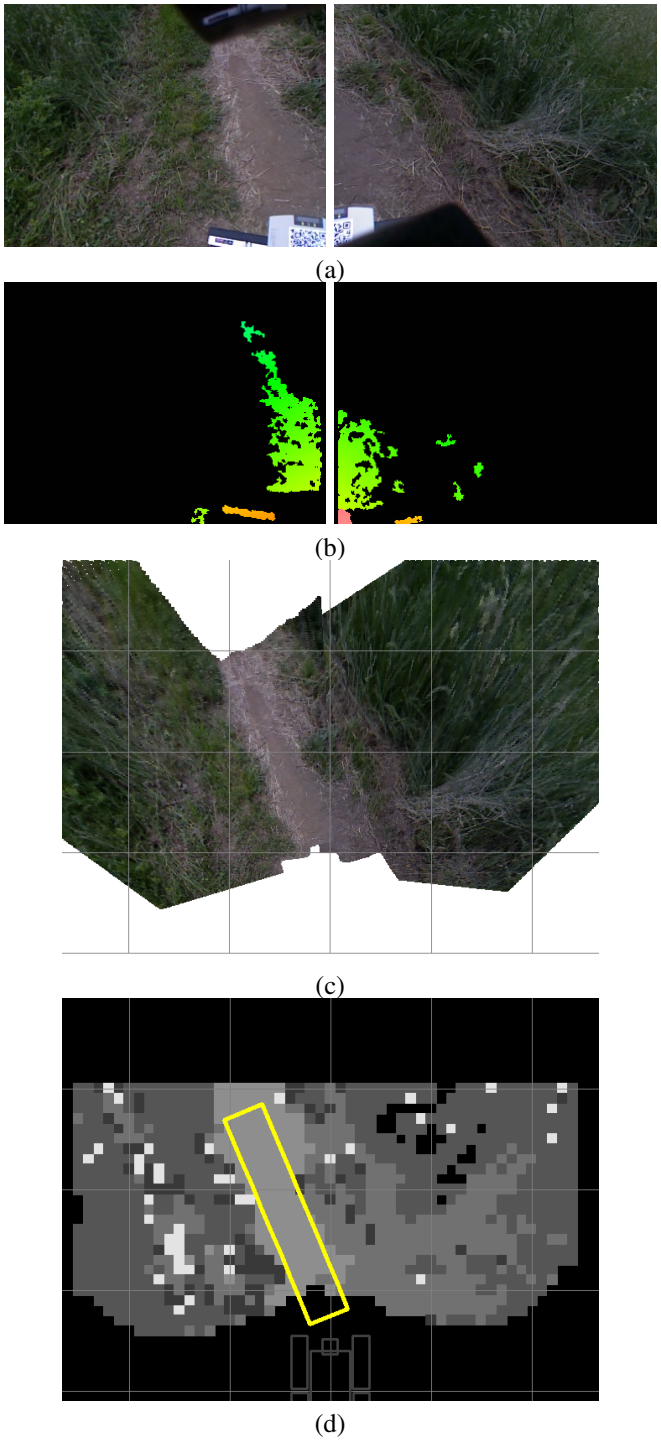

Fig. 5. Trail detection when insufficient depth information is available. (a) Brightly-lit scene; (b) Corresponding depth image, which is missing vast majority of values; (c) Planar projection of RGB values; (d) Corresponding k-means label vehicle-coordinate map and estimated trail region.

In [27] these contrast measures were computed on the omnidirectional image pixels after projecting a candidate region from vehicle coordinates. Here, we are able to accurately "unproject" RGB pixels to vehicle space because we have their true 3-D locations from the Kinect's depth-to-RGB-camera calibration. We rasterize the point cloud by collecting k-means labels in a grid-style *map* in vehicle coordinates where each grid square is 0.1 m on a side. A sample such map is shown in Fig. 4(b), with the maximum likelihood trail region estimate overlaid.

As sunlight conditions brighten, however, the amount of usable depth information returned by the Kinect diminishes. A morning scene which was captured about 3 hours after sunrise on a cloudy day demonstrates this issue in Fig. 5(a). The corresponding depth image is given in Fig. 5(b): depths are available for only a fraction of the image. While the depth information is vital for pure obstacle avoidance, one of the key motivations for trail segmentation is that it represents a form of visual "non-obstacle" detection. Depth values are necessary to accurately project the entire color point cloud into the vehicle-coordinate map, but by assuming that all pixels lie on the ground plane (which is not unreasonable for the trail region) we can simply intersect pixel rays with the ground plane to obtain approximate 3-D locations. Such a ground-plane projection and its associated map are shown in Fig. 5(c) and (d), respectively.

D. Trail detection from height and normal contrast

Here we seek to adapt the color approach above to implement our intuition, outlined in the introduction, that structural height and/or roughness contrast derived from the depth images can also be used to discriminate the trail. The major purpose of a depth-based formulation is to enable trail segmentation when color-based discrimination may fail. One obvious reason is when there is too little illumination as twilight turns to night, or in deep shadow. A second reason is when the appearance characteristics of the trail and the neighboring terrain are very similar, as when fallen leaves in the forest cover everything, or if on snow or dirt where the trail is only distinguished by ruts or footprints.

An attractive idea is to try to find obstacles first and infer the trail region as the remaining drivable area. If the ground is perfectly flat, we can do a robust planar fit and simply threshold on point-to-plane distance to get obstacles. However, this is a problem for several reasons. First, the ground may contain several planes, as with a sidewalk and adjacent street, and it is not guaranteed that the trail plane is the "dominant" one that would be recovered using a RANSAC-like process. An analogous situation occurs in many of our testing areas where the trail is a kind of narrow trough in the midst of plentiful grass or ground cover–in such a case the plane would be fit to the grass and the trail point heights would all be negative outliers. A second reason why fitting a single plane often does not work is that the terrain may have considerable slope changes and undulations which make that single plane a poor description of the ground height for most of the scene.

Our approach is to simplify the method of [21], which fits planes to robot-sized chunks of a stereo-derived point cloud and combines them into a *traversability map* comprising several hazard-related factors. Full repeated plane-fitting is expensive, so we approximate it by computing the median

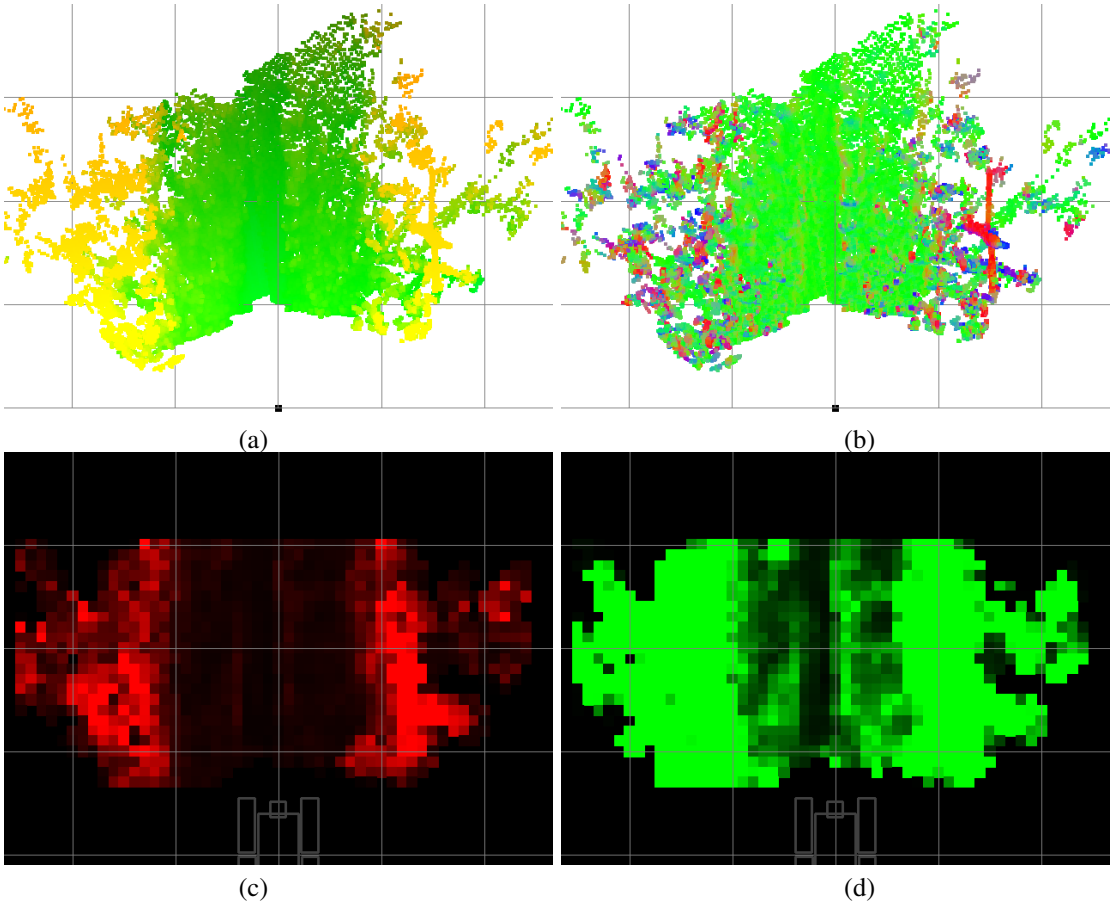

Fig. 6. (a) Another view of combined point cloud for Fig. 3 but with colors now representing height relative to nominal ground plane $Y = 0$: yellow higher, green on ground, blue below ground; (b) Same scene and viewpoint, but with color representing estimated normal direction (red channel = X component, green = Y, blue = Z); (c) Height traversability map and (d) Normal variance traversability map for same scene.

absolute deviation (MAD) of the Kinect depth-derived height map over robot-sized bins. Higher values of the height MAD tend to correlate with bumpy or non-level spots, which are not desirable trail characteristics. If μ_{MAD} is the mean MAD value or "badness" within a hypothesized trail region $\mathcal{R}$, then the height likelihood $L_{height}(\mathcal{R})$ is a weighted sum of the badnesses of the neighboring regions (which we want to be high) minus that of the central region (which we want to be low). A sample "height traversability map" is shown in Fig. 6(c). Note how it picks up on the vegetation growing alongside the trail. Besides step edges, this MAD formulation also is sensitive to substantial slopes.

A major hypothesis of this work is that height contrast is not always sufficient to discriminate the trail. As can be seen in the sample scene in Fig. 3, although there is tall vegetation and a fence near the trail, between those objects and the actual trail are strips of much shorter grass-like ground cover. The height contrast between this area and the trail region is minimal, but the trail's *smoothness* is a distinguishing feature here. We quantify smoothness by measuring local deviation of the surface normal from the expected upright direction.

Specifically, a normal vector was computed for each 3-D point using the Point Cloud Library (PCL) [31] with k-nearest neighborhoods ($k = 25$ unless otherwise noted). This neighborhood definition seemed to produce better results with the spatially-varying density of the points in the point cloud caused by foreshortening over the entire scene than an r-radius neighborhood. The Y (i.e., vertical) component of each point's normal vector was binned in a *normal variance* traversability map, and as above the maximum absolute deviation (MAD) was computed for each bin to obtain a measure of local roughness. For this formulation the normal likelihood $L_{normal}(\mathcal{R})$ was analogously the mean normal Y MAD over the left and right neighbor regions (which we expect to be high–i.e., rougher) minus the MAD of the central hypothesized region (thus penalizing for roughness in the nominal trail region). A sample normal variance map is shown in Fig. 6(d). Note how much tighter it is on the trail.

More results for each method are shown in Fig. 7. Fig. 7(a) is of a wooden bridge trail section with poor color contrast, and Fig. 7(c) was taken about 10 minutes before sunset in the forest, near the limit of the Kinect RGB camera's low-light capabilities. The trail height contrast is not large in this section, but the normal traversability map picks up the smooth section

well. A comparison of the three methods is shown in Fig. 8 for a sharp turn around a tree. While every method tracks the trail successfully here, there is less ambiguity in the normal traversability map of Fig. 8(c). Both the height and color maps show a false "fork": an apparent trail region straight ahead that is both brown and low compared to the surrounding vegetation. The normal traversability map rejects this region more decisively because of its comparative bumpiness.

III. Conclusion

We have presented several promising components of a Kinect-based system for appearance- and structure-based trail-following. It is evident that the depth and color capabilities of the Kinect are roughly complementary outdoors: the darker the scene, the better depth recovery works, but the dimmer the color image. Conversely, brighter scenes generally yield better-exposed color images, but quickly wash out the depth sensor. Accordingly, we continue to work on automatically switching between or weighting the structural and appearance cues in order to track the trail through all kinds of illumination conditions. Thresholds on the number of "good" depth pixels in an image vs. the number of well-exposed (neither under- nor over-saturated) RGB pixels may be used as absolute bounds, but there is still a light range in between where the relative efficacy of these cues must be estimated in a scene-dependent fashion.

Besides trails, we are currently looking at methods for Kinect-based detection of other semantically-meaningful outdoor objects such as trees, posts/poles, and large rocks. These are of interest because the robot may want to study them further for scientific or other applications, but most pressingly because they are obstacle types which are quite dangerous vs., say, tall grass. Distinguishing hard from "soft" obstacles, both of which present as large structures to ladar and stereo sensors, has traditionally been a difficult issue [33].

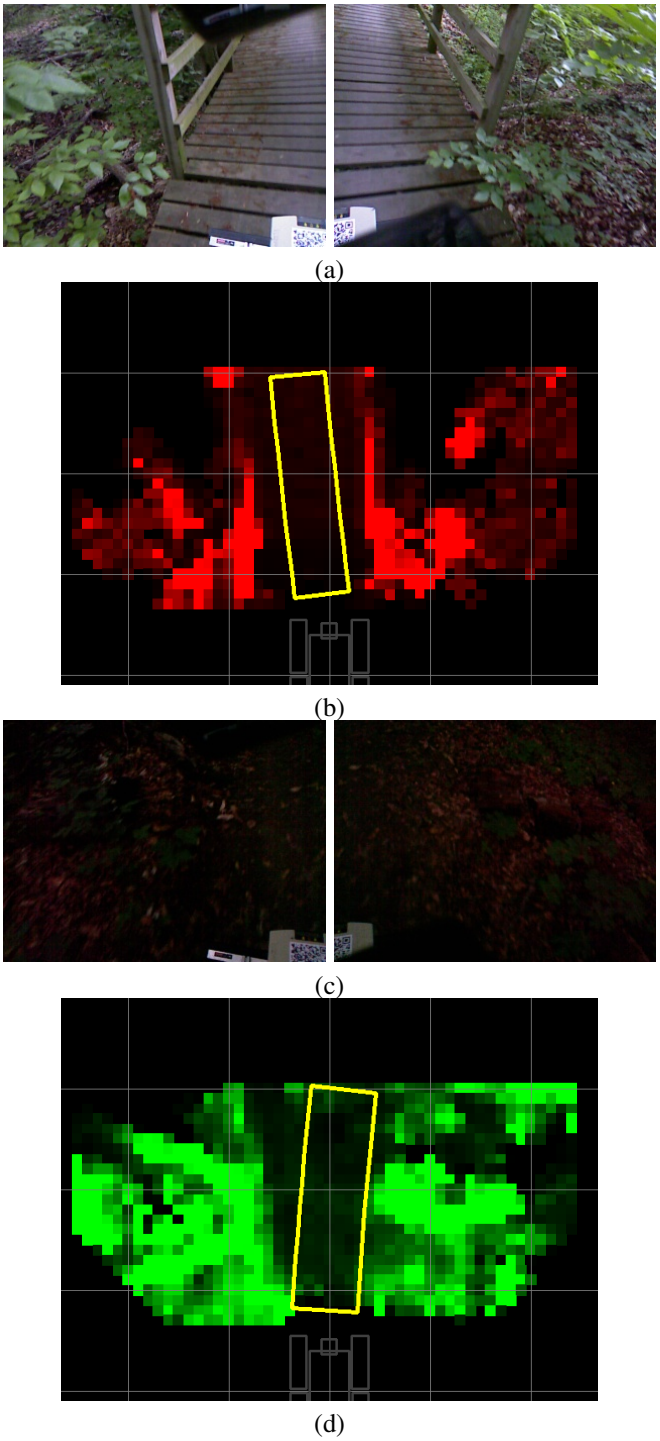

Fig. 7. Sample trail detection results. (a) and (b) Using height traversability map; (c) and (d) Using normal traversability map.

References

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

[2] P. Axelsson. Processing of laser scanner data: algorithms and applications. *ISPRS Journal of Photogrammetry & Remote Sensing*, 54: 138–147, 1999.

[3] J. Biswas and M. Veloso. Depth camera based localization and navigation for indoor mobile robots. In *RGBD Workshop at RSS*, 2011.

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

[5] 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.

[6] P. Bouffard. Quadrotor autonomous flight and obstacle avoidance with kinect sensor. Available at `http://www.youtube.com/ watch?v=eWmVrfjDCyw`. Posted December 5, 2010.

[7] A. Broggi, C. Caraffi, R. Fedriga, and P. Grisleri. Obstacle detection with stereo vision for off-road vehicle navigation. In *IEEE Workshop on Machine Vision for Intelligent Vehicles*, 2005.

[8] N. Burrus. Kinect calibration. Available at `http://nicolas.burrus.name/ index.php/Research/ KinectCalibration`. Accessed March, 2012.

[9] M. Carlberg, P. Gao, G. Chen, and A. Zakhor. Classifying urban landscape in aerial lidar using 3d shape analysis. In *Proc. IEEE Int. Conf. on Image Processing*, 2009.

[10] J. Crisman and C. Thorpe. SCARF: A color vision system that tracks roads and intersections. *IEEE Trans. Robotics and Automation*, 9(1): 49–58, 1993.

[11] J. Cunha, E. Pedrosa, C. Cruz, A. Neves, and N. Lau. Using a depth camera for indoor robot localization and navigation. In *RGBD Workshop at RSS*, 2011.

[12] 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.
[13] B. Graham. The drill of depth. Available at `http://www.rowland.harvard.edu/ cox/ projects/ subprojects/ kinect`. Accessed March, 2012.
[14] E. Guizzo. How Google's self-driving car works. *IEEE Spectrum Magazine*, 2011.
[15] 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.
[16] 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.
[17] H. Koppula, A. Anand, T. Joachims, and A. Saxena. Semantic labeling of 3d point clouds for indoor scenes. In *Advances in Neural Information Processing Systems*, 2011.
[18] K. Kraus and N. Pfeifer. Determination of terrain models in wooded areas with airborne laser scanner data. *ISPRS Journal of Photogrammetry & Remote Sensing*, 53:193–203, 1998.
[19] J. Lalonde, N. Vandapel, D. Huber, and M. Hebert. Natural terrain classification using three-dimensional ladar data for ground robot mobility. *J. Field Robotics*, 23(10):839–861, 2006.
[20] M. Luber, L. Spinello, and K. Arras. People tracking in rgb-d data with on-line boosted target models. In *IROS*, 2011.
[21] M. Maimone, C. Leger, and J. Biesiadecki. Overview of the mars exploration rovers' autonomous mobility and vision capabilities. In *ICRA Space Robotics Workshop*, 2007.
[22] R. Manduchi, A. Castano, A. Talukder, and L. Matthies. Obstacle detection and terrain classification for autonomous off-road navigation. *Autonomous Robots*, 2005.
[23] H. Martin, K. Machulis, J. Blake, and B. White. libfreenect kinect driver and api. Available at `http://openkinect.org`. Accessed March, 2012.
[24] M. Montemerlo et al. Junior: The Stanford entry in the urban challenge. *J. Field Robotics*, 2008.
[25] C. Rasmussen. Combining laser range, color, and texture cues for autonomous road following. In *Proc. IEEE Int. Conf. Robotics and Automation*, 2002.
[26] C. Rasmussen, Y. Lu, and M. Kocamaz. Appearance contrast for fast, robust trail-following. In *Proc. Int. Conf. Intelligent Robots and Systems*, 2009.
[27] C. Rasmussen, Y. Lu, and M. Kocamaz. Trail following with omnidirectional vision. In *Proc. Int. Conf. Intelligent Robots and Systems*, 2010.
[28] C. Rasmussen, Y. Lu, and M. Kocamaz. Integrating stereo structure for omnidirectional trail following. In *Proc. Int. Conf. Intelligent Robots and Systems*, 2011.
[29] C. Rasmussen, Y. Lu, and M. Kocamaz. A trail-following robot which uses appearance and structural cues. In *Proc. Int. Conf. on Field and Service Robotics*, 2012.
[30] A. Robledo, S. Cossell, and J. Guivant. Outdoor ride: Data fusion of a 3d kinect camera installed in a bicycle. In *Australasian Conference on Robotics and Automation*, 2011.
[31] R. Rusu and S. Cousins. 3D is here: Point Cloud Library (PCL). In *ICRA*, 2011.
[32] Y. Schroder, A. Scholz, K. Berger, K. Ruhl, S. Guthe, and M. Magnor. Multiple kinect studies. Technical Report 09-15, ICG, Technical University of Braunschweig, 2011.
[33] 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.
[34] A. Telea. An image inpainting technique based on the fast marching method. *Journal of Graphics Tools*, 9(1):25–36, 2004.
[35] S. Thrun, M. Montemerlo, et al. Stanley, the robot that won the DARPA grand challenge. *J. Field Robotics*, 23(9), 2006.
[36] I. Ulrich and I. Nourbakhsh. Appearance-based obstacle detection with monocular color vision. In *Proceedings of the National Conference on Artificial Intelligence*, 2000.
[37] C. Urmson et al. A robust approach to high-speed navigation for unrehearsed desert terrain. *J. Field Robotics*, 23(8):467–508, 2006.
[38] C. Urmson et al. Autonomous driving in urban environments: Boss and the urban challenge. *J. Field Robotics*, 25(1), 2008.

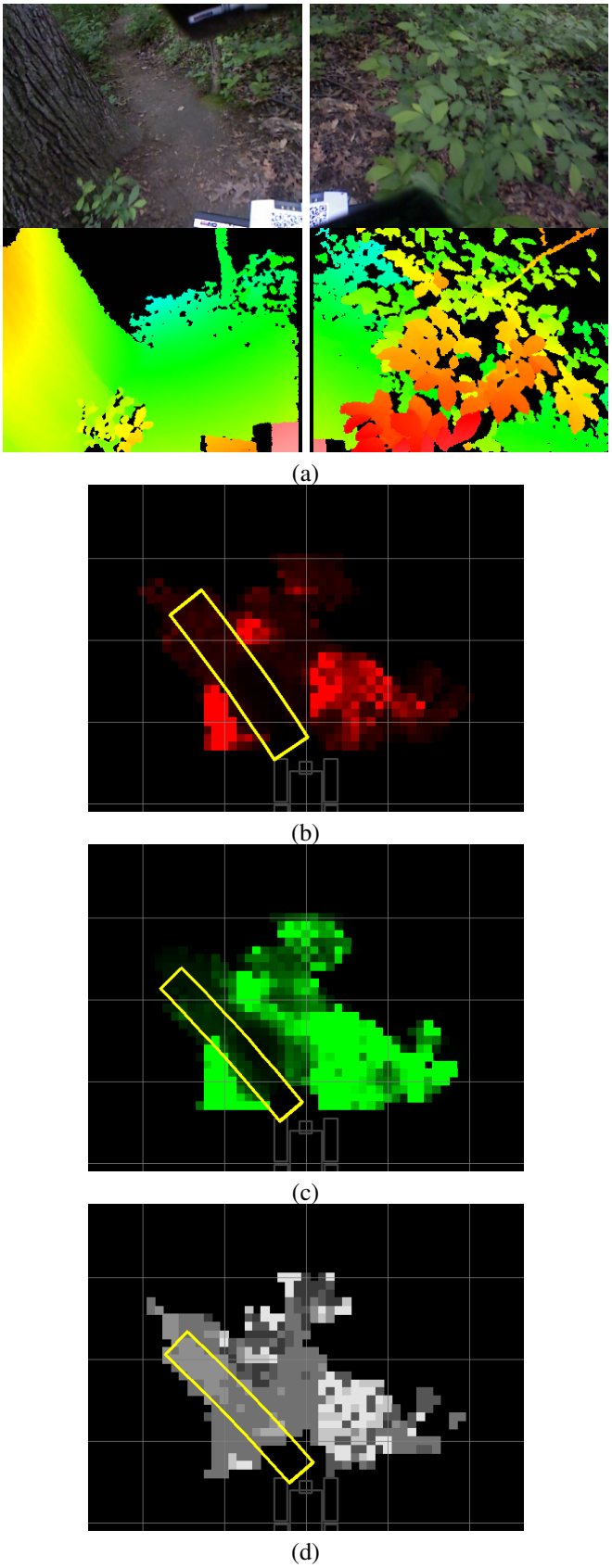

Fig. 8. Comparative trail detection results. (a) RGB images and depth images; (b) Using height traversability map; (c) Using normal traversability map; (d) Using color cluster likelihood