Appearance Contrast for Fast, Robust Trail-Following

Christopher Rasmussen Yan Lu Mehmet Kocamaz

***Abstract*— We describe a framework for finding and tracking "trails" for autonomous outdoor robot navigation. Through a combination of visual cues and ladar-derived structural information, the algorithm is able to follow paths which pass through multiple zones of terrain smoothness, border vegetation, tread material, and illumination conditions. Our shape-based visual trail tracker assumes that the approaching trail region is approximately triangular under perspective. It generates region hypotheses from a learned distribution of expected trail width and curvature variation, and scores them using a robust measure of color and brightness contrast with flanking regions. The structural component analogously rewards hypotheses which correspond to empty or low-density regions in a groundstrike-filtered ladar obstacle map. Our system's performance is analyzed on several long sequences with diverse appearance and structural characteristics. Ground-truth segmentations are used to quantify performance where available, and several alternative algorithms are compared on the same data.**

I. INTRODUCTION

Navigationally-useful linear features along the ground, or *trails*, are ubiquitous in man-made and natural outdoor environments. Spanning engineered highways to rough-cut hiking tracks to above-ground pipelines to rivers and canals, they "show the way" to unmanned ground or aerial vehicles that can recognize them. Built trails also typically "smooth the way," whether by paving, grading steep slopes, or removing obstacles. The perceptual tasks involved in general trail-following may be divided into three categories:

Finding This is the problem of detecting and segmenting a trail with little or no *a priori* information about its specific appearance, location, and shape. Corollary issues include deciding whether the trail is coming to a dead-end or a branch, or more generally counting how many trails are currently in view.

Keeping Analogous to the sense of "lane keeping" from autonomous road following, this involves repeated estimation, or tracking, of the gross shape and appearance attributes of a previously-found trail. For *discontinuous* trails marked by blazes, footprints, or other sequences of discrete features, the underlying task is successive guided search rather than segmentation.

Negotiation When a trail contains hazards such as rocks, roots, logs, and puddles, the gross shape estimate may be insufficient for safe travel. In-trail obstacle detection and more general motion planning are necessary for avoidance maneuvers, as well as control policy adjustments due to bumpiness or tread material changes affecting wheel slip.

In this paper we describe an approach to the tasks of *finding* and *keeping* to a non-branching, non-terminating, continuous trail which relies primarily upon vision and secondarily upon ladar to discriminate the drivable region ahead. Though above we define trails as broad category relevant to UGVs and UAVs, here we focus primarily on hiking and mountain-biking trails through field and forest terrain which are suitable for wheeled travel. Representative images taken along such trails are shown in Figure 1. Our robot platform is a Segway RMP 400 (see Figure 2 on the next page) outfitted with several common sensors, including a monocular color camera, a SICK LMS laser range-finder, and a Hokuyo URG short-range ladar.

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.

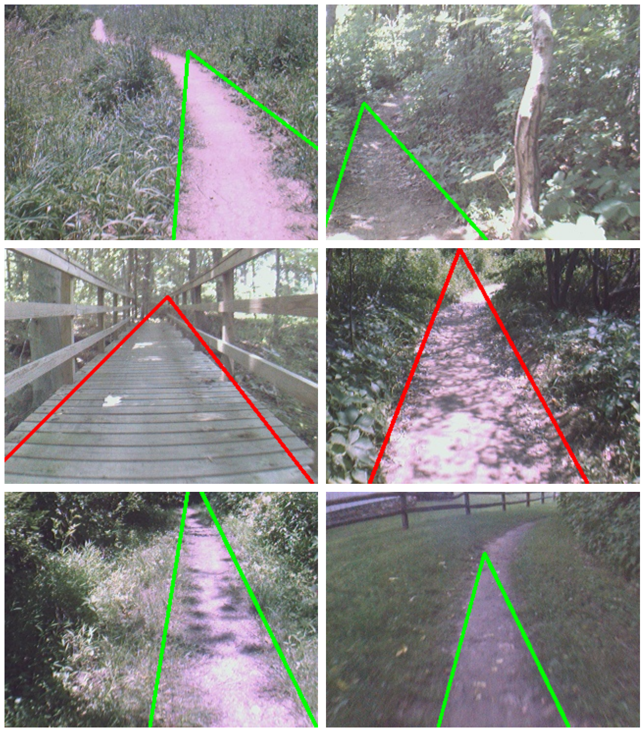


Fig. 1. Representative trail images showing typical summer variation in tread material, adjacent foliage, and lighting conditions. Trail region estimates using the image-only algorithm of Section II are overlaid

Placing trail finding and trail keeping in context, an obvious analogy is to consider these tasks as scaled-down versions of road following. However, while traditional vision-based road following has been thoroughly studied on paved and/or painted roads with sharp edges and shapes which are well-approximated by clothoids or other analytic curves [1], [2], [3], [4], hiking trails lack many of these attributes. as they may have ragged, indistinct borders. This suggests using bottom-up methods based on color or texture to classify image patches as road vs. background [5], [6], [7], [8]. Though more useful, these approaches often assume that the finding task is trivial and may be unable to cope with rapidly changing tread materials and lighting conditions.

The two DARPA Grand Challenges required vehicles to follow rough roads, but GPS and ladar were sufficient for most successful teams [9], [10]. Vision was not primary for any team, although it was exploited as a means of detecting long-range obstacles for speed control [11] and as a road direction estimator [12]. The navigational tasks in the DARPA Urban Challenge largely did not require direct road shape estimation, although a very detailed

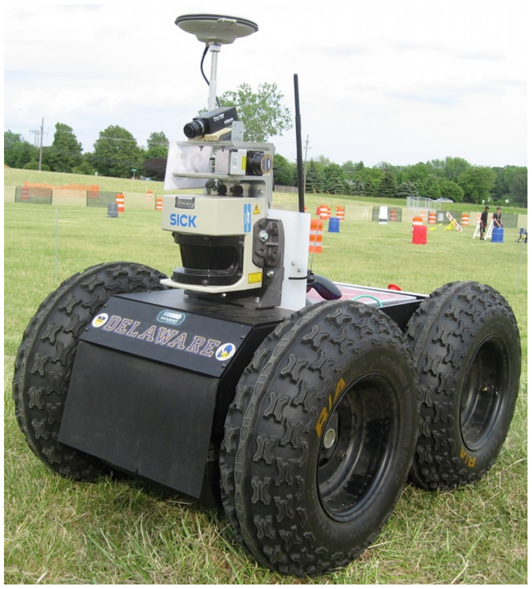


Fig. 2. "*Warthog*": Segway RMP 400 robot platform with color camera, SICK ladar, and sagitally-mounted Hokuyo URG

approach which relied on rich structural information such as from a Velodyne ladar was described in [13]. There are some similarities between the ladar component of this work, that of [12], and the road shape estimator in [13].

General off-road navigation using vision and ladar was investigated in the DARPA PerceptOR program [14], 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 [11], a method to learn long-range obstacle appearance from short-range stereo labels was given in [15]. Among LAGR-derived work, [16] stands out for explicitly looking for path-like corridors of a homogeneous color along the ground. The European ELROB competitions have also called for path-following skills, with one robot effectively following paths by finding "passages" among scattered trees in ladar data [17]. A full ground plane estimate or measures of upcoming longitudinal or transverse slope are necessary along some trail sections to disambiguate obstacles and aid motion planning. A method which uses stereo for this purpose is discussed in [15], and one which uses ladar in the context of urban search and rescue is presented in [18]. Finally, several other researchers have looked into superpixel segmentation as a preprocessing step for dividing a scene into drivable and non-drivable regions, including [19] and [20].

The motivating intuition behind the computer vision aspect of our approach is that the *shape* of trails is subject to less variability than their appearance, and therefore should be a more distinctive cue in finding them. Modulo high-frequency variation along the edges and nonlinearity due to curvature in the distance, we believe that a *triangle* is a reasonable shape template to describe an unoccluded trail viewed under perspective. Therefore, rather than look for, say, a "brown region" which we hope is the trail and then parametrize its shape, we should look for a "triangular region" which simply contrasts with the surroundings and only then parametrize its color.

There has been much recent work on combining bottom-up grouping with top-down shape constraints for segmentation and object detection [21], [22], [23], [24], [25], [26], [27], and one could regard the search for a triangular region in this fashion. A common feature of many such approaches is their use of an oversegmentation technique such as [28], [29] to generate *superpixels* as a preprocessing step. The superpixels are combined in different ways in a bottom-up hypothesis generation step, which is followed by top-down hypothesis-scoring.

In [26]'s formulation, for example, the overall goodness of a region hypothesis $\mathcal{R}$ is assessed through a likelihood function L comprising a linear combination of shape, deformation, and appearance likelihood terms. The shape likelihood measures how closely $\mathcal{R}$'s boundary fits a deformable object class template (e.g., a generic fish or banana silhouette). The more bending or stretching necessary to optimize the template-region fit, the lower the deformation likelihood. Finally, the appearance term measures the agreement of the hypothesized region's pixels with a model of the object's expected color, texture, and so on.

We reported on a trail-following system using such an approach in IROS last year [30]. While mostly successful, there were several drawbacks to the approach. First, the method was too slow for real-time use, taking several seconds per image. Second, it had trouble with multimodal color distributions in the trail region caused by shadows or mixed materials. Partly this was due to an overly simplistic unimodal model of the trail color, but also because superpixel groupings (and therefore trail region hypotheses) were formed by agglomerative clustering of neighbors, which outlier superpixels such as lane lines or rocks could interrupt.

This paper improves upon the previous work in several significant ways which we describe in the following sections. First, the method achieves much greater speed by hypothesizing triangular regions directly instead of superpixel groupings which must then be scored for triangularity. Even with this speed-up, it also works on a wider variety of images due to a more sophisticated and flexible model of trail color. Second, by introducing structural information from a ladar sensor it is able to handle additional trail segments which lack sufficient visual contrast for the image-only method to handle. The combined system is significantly more efficient and reliable.

II. APPEARANCE-BASED TRAIL SEGMENTATION

We approximate the boundary of a trail region $\mathcal{R}$ viewed under perspective from a vehicle on or near the trail as a triangle with its base coincident with the bottom of the image. This *trail triangle* is our shape template in the sense of [26]. For a given triangle T the positions of the top, bottom-left, and bottom-right vertices are $\mathbf{p}^t$, $\mathbf{p}^l$, and $\mathbf{p}^r$, respectively. Since $\mathbf{p}^l$ and $\mathbf{p}^r$ are on the bottom row of the image $y = h - 1$, their x coordinates x^l and x^r suffice to describe them. Thus, a minimal geometric description of the triangle is a 4-D point $\mathbf{t} = (x^t, y^t, x^l, x^r)$, subject to the constraint that $x^r > x^l$. Note that x^l and x^r may be outside of the range $[0, w-1]$, corresponding to the triangle being clipped by the left and/or right image edge. The trail triangle associated with a particular region $\mathcal{R}$ will thus be referred to in a *region* sense as $T(\mathcal{R})$ and in a *point* sense as $\mathbf{t}(\mathcal{R})$. We may drop the reference to $\mathcal{R}$ for notational simplicity when the association is unambiguous.

In [26] and [30], the basic framework for trail finding is to hypothesize possible trail regions $\mathcal{R}$, score each hypothesis with a likelihood function L_{trail}, and return the maximum likelihood hypothesis. As mentioned in the introduction, for complex shapes a typical approach is to first oversegment the image into superpixels, then combine them in different ways using local cues in a bottom-up hypothesis generation step, and finally to rank these hypotheses using a more global, top-down objective function. This is often the only way to proceed with complicated objects, which live in high-dimensional shape spaces that are impractical to sample directly. However, since our template is a simple triangle describable with a 4-D state, it is feasible to directly generate triangle hypotheses and test them.

Here we use a sequential Monte Carlo search method to find good trail hypotheses in a given image. Because we ultimately want to track any found trail regions, it is convenient to implement our single-image search method using a tracking method, particle filtering [31]. In particle filtering, the current state is derived from a weighted sum of all current state hypotheses, or particles. In this paper each trail triangle hypothesis is denoted $\mathbf{t}^i$ and the weight of each particle is proportional to its trail likelihood $L_{trail}(\mathbf{t}^i)$. The best trail region estimate at time t, $\hat{\mathbf{t}}_t$, is taken directly from the particle filter state for all image sequence results. However, for individual images this method results in an estimate which is continually and unnecessarily changing due to noise. In this case, the trail region estimate $\hat{\mathbf{t}}_t$ is simply the particle with the highest trail likelihood seen in t iterations of the filter.

As discussed in the introduction, the trail likelihood function L_{trail} in our earlier work [30] consisted of a linear combination of three terms measuring the appearance, shape, and deformation likelihood of the trail region hypothesis. In this work we use only the appearance likelihood (greatly modified, as described in the next subsection). Because we now only hypothesize triangular trail regions, the shape likelihood can be dropped.

The deformation likelihood measures how closely a trail triangle meets expectations about the trail's apparent width, centeredness, horizon line, curvature, and so on. These of course depend on both the trail's shape properties as well as the camera intrinsics and its pose relative to the trail. Since the particle filter uses a prior on the state $p(\mathbf{t})$ to sample particles, we simply use the deformation distribution as this prior and also drop the deformation likelihood from the trail likelihood. For trail data collected from our robot platform, our deformation distribution is a Gaussian $(\bar{\mathbf{t}}, \mathbf{\Sigma})$ learned from examples drawn from our testing area and manually labeled using the LabelMe tool [32]. A uniform distribution is used in other cases where the camera intrinsics and pose are unknown.

A. *Single-image appearance likelihood*

As with [30], for a single image we have no *a priori* model of the trail's color or texture: the basis of a high appearance likelihood is *contrast* with the surround. In addition to contrast, we also reward hypotheses which exhibit *symmetry*, meaning that neighboring regions to the right and left are similar to one another.

To be more precise, we refer to the hypothesized trail region as T, and to its left and right neighboring regions as T_L and T_R, respectively. These are all triangles. For this work, we set the neighboring regions to have identical bottom widths to the central one. This arrangement is shown in the left image of Figure 3.

A number of different measures have been proposed to measure appearance similarity (the opposite of contrast) between image regions, including brightness in grayscale images [33], Euclidean color distance [34], and color and texture histogram similarity measures such as Bhattacharyya or χ^2 [35], [36] and the Earth Mover's Distance [24]. Here we have chosen to adapt an efficient yet reasonably sophisticated technique from [16] based on histograms of k-means cluster labels in CIE-Lab color space. Using this method, a set of *textons* is first created from the input image by computing an n-dimensional feature vector at each pixel (x, y).

[16] describes both color and texture features, but we found the latter (as defined) rarely informative and did not use them. CIE-Lab

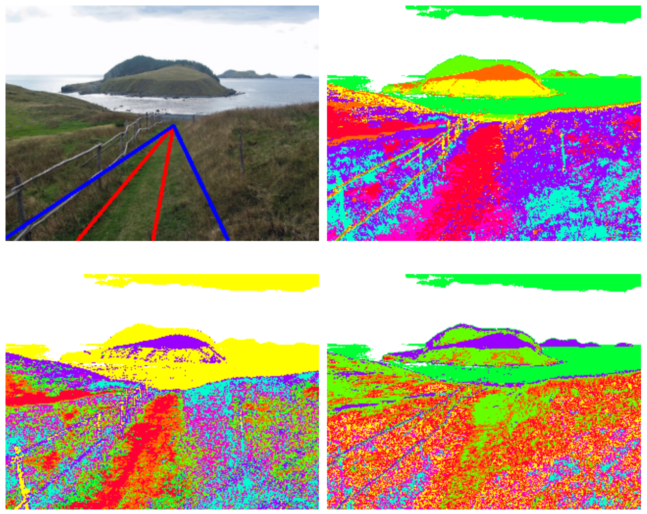


Fig. 3. Hypothetical trail triangle region and sample k-means labels for different cues. Clockwise from upper-left: Central trail region (red) and neighboring regions (blue); cluster labels using $\mathbf{LAB}$ features ($k = 8$); labels using $\mathbf{L}$ features; and labels using $\mathbf{AB}$ features. For this image, a maximum appearance likelihood of 0.693 was found with $\mathbf{AB}$ (the maximum for $\mathbf{LAB}$ is 0.655; for $\mathbf{L}$ it is 0.523).

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 three possible color feature vectors: $\mathbf{LAB}$, which uses all three channels; $\mathbf{AB}$, which uses only the chromaticity coordinates and is thus nominally illumination-insensitive; and $\mathbf{L}$, which uses only brightness.

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 $l_1, \dots, l_{k+2}$. The value used in this paper is $k = 8$. Examples of texton labels using different feature vectors are shown in Figure 3. Note that the over-saturated sky is labeled white in all three.

A triangle region T's color distribution is modeled by a histogram $H = (f_1, \dots, f_{k+2})$ 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 T_i, T_j is captured by a histogram distance function; we use the common chi-squared metric $\chi^2(H_i, H_j)$.

Formally, then, the contrast of a hypothetical trail region T with its left and right neighboring regions T_L and T_R and the symmetry of the flanking regions are combined to obtain:

$$L_{appear}(T) = \frac{\chi^2(H, H_L) + \chi^2(H, H_R) + (1 - \chi^2(H_L, H_R))}{3} \tag{1}$$

In order to avoid biases for triangle hypotheses near the image edges, we set thresholds on the minimum visible area of the neighboring regions for their histograms to be considered statistically significant and therefore included in the appearance likelihood calculation.

Rewarding other characteristics such as trail region interior homogeneity is possible, but from observation these seem to hold for fewer scenes. [26] uses homogeneity but not contrast, while [37], [33] use a ratio of contrast to heterogeneity. Our experiments have indicated that including a homogeneity term (either through raw color variance or the entropy of the label histogram) introduces a bias toward smaller regions, so we do not use it.

B. Feature selection and confidence

An issue of particular interest is whether the choice of a different set of features 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. Recall that the alternatives here are $\mathbf{LAB}$, which combines chromaticity and brightness; $\mathbf{AB}$, which uses only chromaticity; and $\mathbf{L}$, which uses solely brightness. 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.

Our approach is to do k-means clustering three times, once for each cue alternative, on every image. The results for one particular image can be seen in Figure 3. This results in three different histograms $H_{\mathbf{LAB}}$, $H_{\mathbf{AB}}$, and $H_{\mathbf{L}}$ and thus three different appearance likelihoods for each region hypothesis. We do not run three separate particle filters, but rather for each "generation" (a block of, say, 10 iterations) of the particle filter, we make three clones of the particle set and evaluate which feature vector leads to the highest observed appearance likelihood. The particles associated with this cue are the only ones propagated into the next generation. In all of this paper's results figures, the trail tracker estimate is colored according to the "most helpful" cue. For tracking sequences, this is the best cue for the last generation of particle filter iterations; for single images it is the cue associated with the maximum likelihood particle seen thus far.

Following this procedure is important for robust functioning of the tracker. Although $\mathbf{LAB}$ is superior for the majority of images, there are several (such as the one in Figure 3) for which either $\mathbf{AB}$ or $\mathbf{L}$ is necessary to find the best segmentation. A corollary of this technique for cue selection is that the magnitude of the best hypothesis' appearance likelihood is a useful measure of tracker confidence. When the tracker has a strong "lock" on a trail, the score is high (L_{appear} is always in the range $[0, 1]$); when the trail has only marginal contrast or is not present, the trail likelihood invariably drops. There is not space for the evidence, but through testing we have found that a threshold on L_{appear} of about 0.6 is a good dividing line. Images which are classified as not containing a trail are colored red in this paper.

III. INCORPORATING STRUCTURAL INFORMATION

Irrespective of the visual environment, along trail sections bordered by thick foliage, SICK ladar scans are characterized by linear clusters along the direction of the trail. These may be on just one side or both, as seen in the top Figure 5(b), indicating a corridor-like structure. When this structure is present, it is a strong cue for the trail direction. This could lead us to formulate an analog of the trail likelihood function above for trail-following with ladar data. Instead of a triangle in the image, we would be looking for a "trail rectangle" in vehicle coordinates. The deformation distribution would be parametrized in units of meters rather than pixels, and the dynamics of the trail tracker would stem directly from the robot motion. There would be no color for a ladar appearance likelihood, of course. Rather, a high-likelihood trail rectangle hypothesis would be one that is relatively empty of ladar obstacles yet flanked by obstacle-dense neighboring regions.

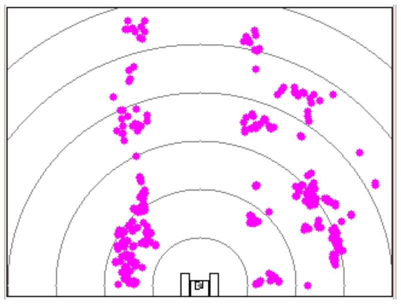
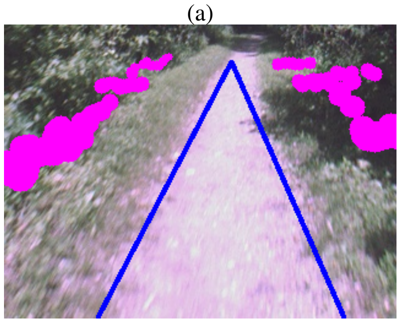

(a)

(b)

Fig. 4. Structural obstacles constrain the trail, but do not necessarily define its edges. (a) Overhead view of SICK ladar scan of trail region (concentric circles are at 1 m intervals); (b) Camera view with ladar obstacles projected.

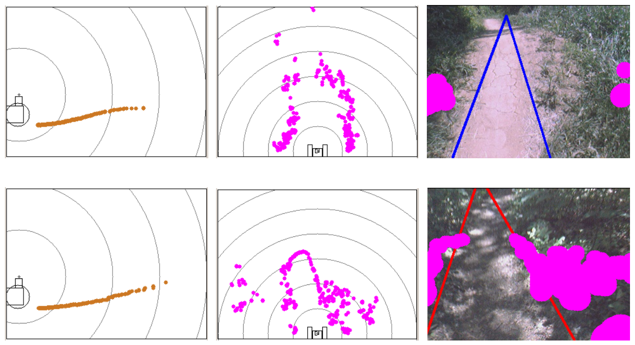

(a) (b) (c)

Fig. 5. Filtering ladar groundstrike, two examples. (a) Sideview of robot platform showing sagittal ground profile from Hokuyo data; (b) Overhead view of simultaneous SICK ladar scan; (c) Camera view of trail with SICK ladar points projected (groundstrikes removed). The groundstrike distance was calculated as 2.3 m in the first example and 3.1 m in the second.

The problem with trying to use ladar alone to find trails is that it only constrains them—it does not always define their borders as visual cues do, as seen in Figure 4. Thus, in order to integrate structural information with the image-based component described above, we project ladar points into the camera image using the relative poses of the ladar and camera and the camera's internal calibration. This allows us to augment the trail likelihood with a term that penalizes hypotheses that stray into obstacle regions. To place a higher priority on avoiding nearby obstacles, projected obstacle points are rendered as circles with radii proportional to their depths, as seen in Figure 4(b).

A. *Groundstrike filtering*

The robot's SICK ladar is mounted to sweep a coronal plane, parallel to the ground. Although it is roughly 0.5 m off the ground, *groundstrikes* or ladar returns which intersect drivable ground features can frequently occur on undulating trails. Groundstrikes may be encountered as the robot approaches the base of an incline or hump along the trail, or as it comes down a slope which is starting to level off. Although groundstrikes are often transient and clear naturally as the robot moves forward, if these situations are not explicitly recognized the robot may erroneously believe that its path is blocked by a "phantom wall."

Such slopes may be detected in several ways, including fitting a ground plane to a stereo depth map [15]. We use a separate Hokuyo ladar mounted to sweep a sagittal plane which yields clean profiles of the ground in front of the robot out to about 4 m, as seen in Figure 5. After removing ladar points which are above the robot (due to overhanging foliage), a RANSAC robust line-fitting algorithm is applied to the ground profile. The line thus obtained is intersected with a line representing the plane of the SICK ladar's sweep, and the result is the estimated *groundstrike distance*. SICK ladar points beyond this distance are removed as suspected groundstrikes; only points nearer are projected into the image.

IV. RESULTS

Our results demonstrate the accuracy and efficiency of the image-only trail finder and tracker of Section II on a diverse set of trail images. These experiments use data collected both from the Web and several trail image sequences taken from our robot platform as it was manually driven. For all of the Web images and at regularly-spaced intervals along these sequences we have manually-generated ground-truth segmentations. These permit us to quantify the accuracy of our trail triangle estimates[1].

We also provide a baseline comparison to several image-based "freespace segmentation" algorithms from other authors. The first comparison algorithm is the surface layout method of [38], which was used for robotic motion planning in [19]. This algorithm oversegments an image into superpixels and then groups them into categories such as ground (i.e., drivable areas), vertical surfaces (i.e., obstacles), and sky using a variety of appearance and geometric features. We use the outdoor, IJCV version of their classifier which is the most advanced of those publicly available and the most appropriate to our data.

The diversity of trail images that the system can process successfully is shown in Figure 6. The data set consists of thirty images containing trails culled from Flickr and Google and cropped and scaled to 320×240 as necessary. Only 15 images are shown here; images of rivers, paved roads, and snow were removed to focus on hiking-type trails. On odd rows, the highest scoring trail hypothesis obtained by our algorithm within 100 iterations of the particle filter on a downsampled 80×60 version of the input is shown for every image (the trail triangle estimate is scaled up and drawn on the full-size image for clearer display). The algorithm does a good job of finding the trail region in each image despite their very different colors, sizes, and image locations. Using the overlap formula above, the median overlap between the found trail triangle and the ground truth polygon over all 30 images was 0.828. On even rows we show the output of the surface layout classifier of [38]. Their system does a a credible job of finding nearby ground, but accuracy decreases with depth, and roads and trail regions do not seemed to be favored over rougher ground.

[1]We use the polygon area overlap formula suggested by [26]: $Overlap(\mathcal{R}_1, \mathcal{R}_2) = A(\mathcal{R}_1 \cap \mathcal{R}_2)^2/(A(\mathcal{R}_1)A(\mathcal{R}_2))$

Source code is not readily available for other recent algorithms which segment images into drivable and non-drivable areas, and the most current batch from DARPA LAGR projects also use stereo depth in an integral fashion. Nonetheless, it is instructive to run our algorithm as well as [38] on some selections from their data as shown in Figure 7. Each column of images (numbered 1 to 6 from left to right) is headed by an input image taken from [16] (1 and 2), [15] (3 and 4), and [20] (5 and 6). The second row shows the output of our algorithm, the third row shows the surface layout of [38], and the fourth shows the output of the source algorithm (note [15]'s output is an overhead view). Our algorithm gives an incorrect or incomplete trail triangle for images 1 and 5. This is most likely because the trails are so wide that a large fraction of them is outside of the image—our appearance likelihood function requires that both the left and right neighboring regions be at least partially visible. Images 4 and 6 are classified as containing no trail—the trail region in 4 is again bordering the image edge, and the fork diminishes its triangularity. The trail triangles in images 2 and 3 are basically correct.

We have also run the single-image trail finder on several long hiking trail sequences represented by the images in Figure 1. On the hiking trail and canyon sequences there is excellent gross accuracy of trail detection and excellent frame-to-frame correlation considering the variability of illumination conditions. Frames from a multi-km-long data set collected on the hiking trail are shown in Figure 8. The median overlap with ground truth over the course of the trail was 0.681. The $\mathbf{LAB}$ cue was used 40.4 percent of the time, $\mathbf{AB}$ 34.6 percent, and $\mathbf{L}$ for the remaining 25 percent of the frames. The ability of the system to negotiate a particularly tricky section of shadows and bright spots is shown in Figure 9. For image sequences, fewer iterations of the particle filter are necessary per frame because of the high frame-to-frame correlation, and thus we get a frame rate of nearly 30 fps.

V. CONCLUSION

We have presented a practical approach to visually finding and following general trails for robot autonomy and showed it working on a number of different kinds of images and image sequences. The method does not require an *a priori* color or texture model for the trail region, working primarily from general cues such as gross shape and self-similar regional appearance vs. contrasting surroundings to localize a variety of trail types without parameter changes. This baseline implementation is fairly accurate on realistic imagery and efficient for its level of sophistication, running at interactive rates suitable for control of a ground robot. We are currently extending the trail tracker state to maintain a color model of the trail region over time. Color hysteresis should help tracker stability in several situations in which an off-trail region may have more contrast than the trail itself. With odometric information from the robot, knowledge of forward motion and turning speed in vehicle coordinates can be transformed into image motion predictions. We are working on incorporating this into the particle filter dynamics.

It is not critical for gross robot motion planning, but for high-speed situations where turn anticipation is important or where in-trail obstacles are a hazard, we are looking at several ways to refine the coarse trail shape estimate returned by our triangle finder to get a more precise region boundary. We have promising preliminary results using a lightweight version of the superpixel grouping method from [30] which runs as a post-process after trail triangle estimation, as well as a pixel-level method which scales

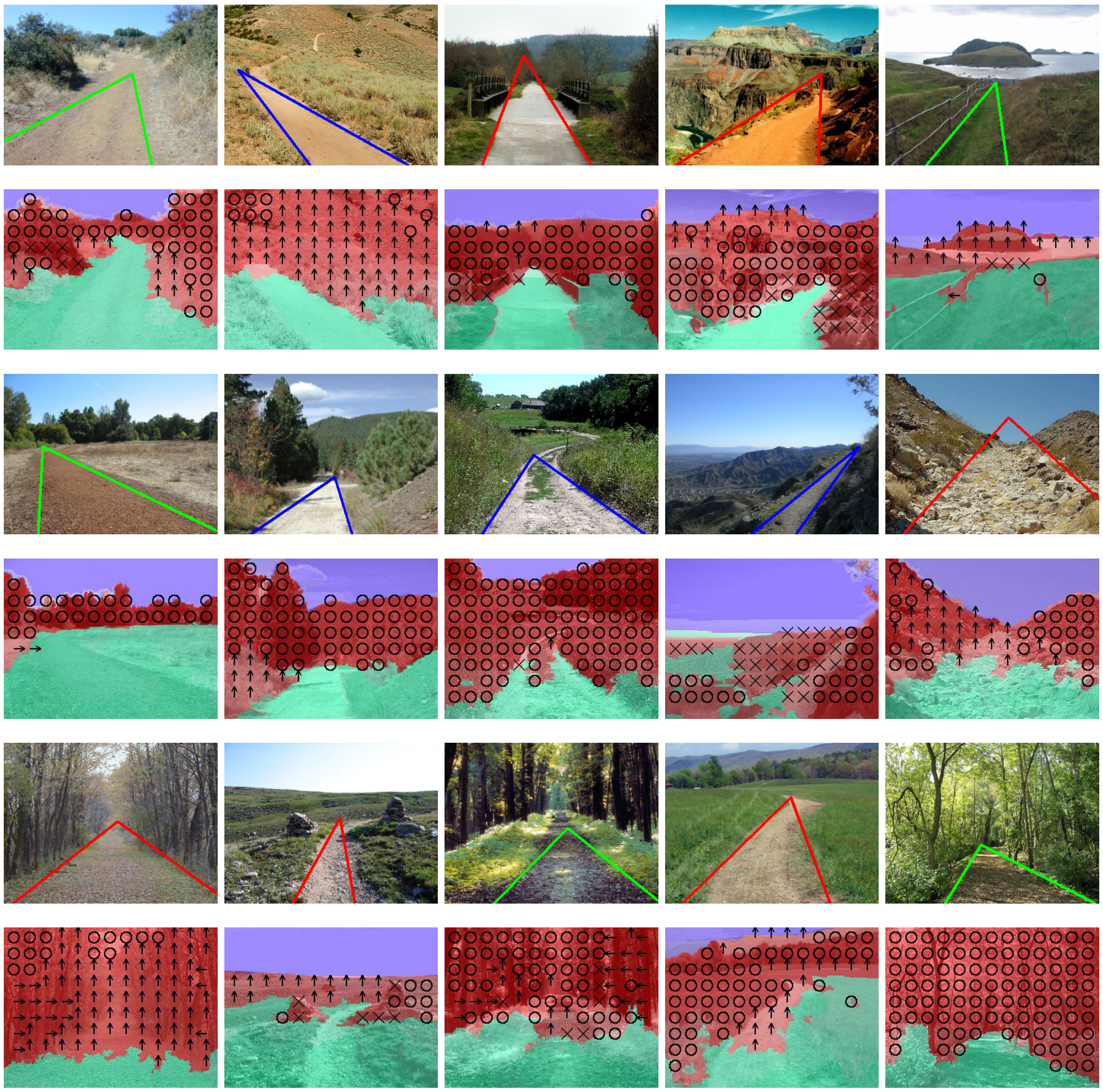


Fig. 6. Results on single-image trail triangle finder on subset of the Web trail data set (see text). Odd rows are output of vision-only component of this paper's algorithm; even rows are output of [38]. Colors indicate which features were automatically chosen as most discriminative: red for $\mathbf{LAB}$, green for $\mathbf{AB}$, and blue for $\mathbf{L}$.

the trail triangle down and up to get high-confidence trail and high-confidence background regions, respectively, in order to run something similar to GrabCut [39] to classify the unknown border region in between.

VI. Acknowledgments

This work was supported by the National Science Foundation.

References

[1] D. Pomerleau, "RALPH: Rapidly adapting lateral position handler," in *Proc. IEEE Intelligent Vehicles Symposium*, 1995, pp. 506–511.

[2] C. Taylor, J. Malik, and J. Weber, "A real-time approach to stereopsis and lane-finding," in *Proc. IEEE Intelligent Vehicles Symposium*, 1996.

[3] B. Southall and C. Taylor, "Stochastic road shape estimation," in *Proc. Int. Conf. Computer Vision*, 2001, pp. 205–212.

[4] 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.

[5] N. Apostoloff and A. Zelinsky, "Robust vision based lane tracking using multiple cues and particle filtering," in *Proc. IEEE Intelligent Vehicles Symposium*, 2003.

[6] J. Crisman and C. Thorpe, "UNSCARF, a color vision system for the detection of unstructured roads," in *Proc. IEEE Int. Conf. Robotics and Automation*, 1991, pp. 2496–2501.

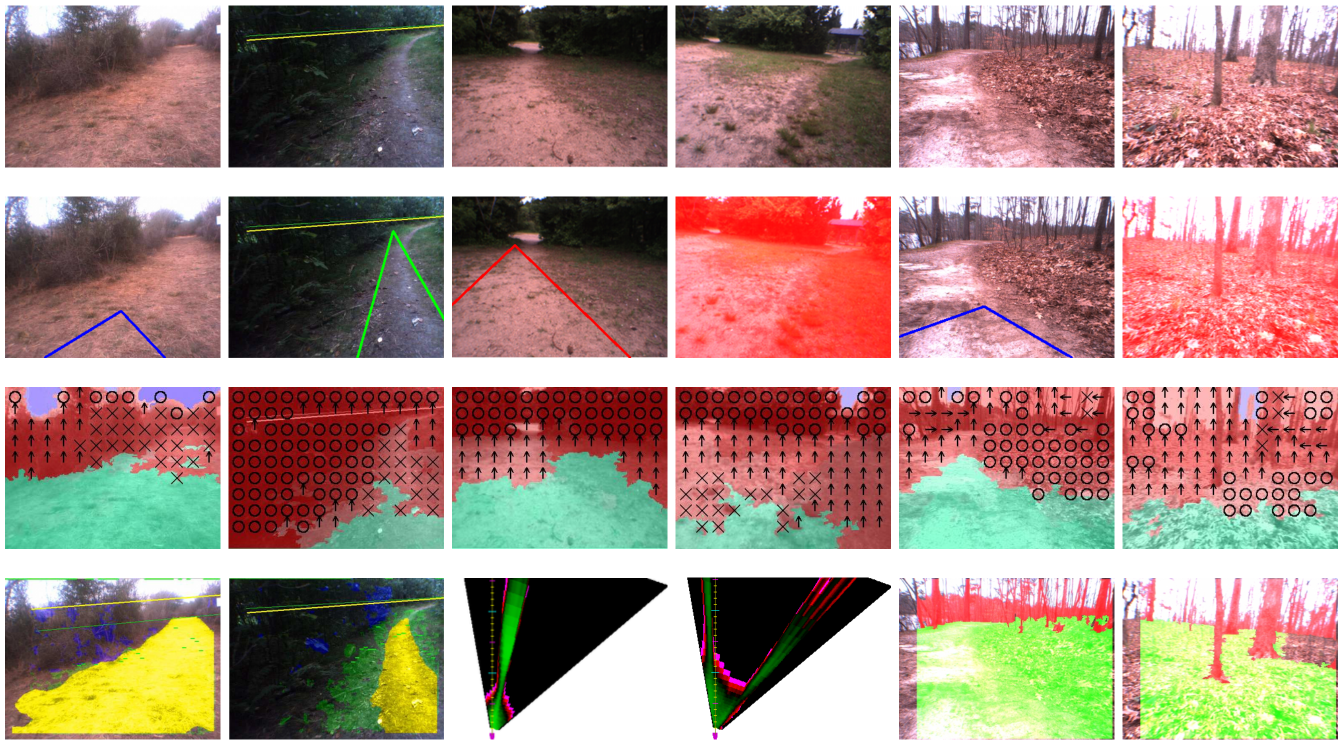


Fig. 7. Comparisons to other algorithms. First row: input image; second row: output from vision-only component of this paper's algorithm; third row: output of [38]; fourth row: output of alternative method. First two columns are from [16], next two are from [15], last two are from [20].

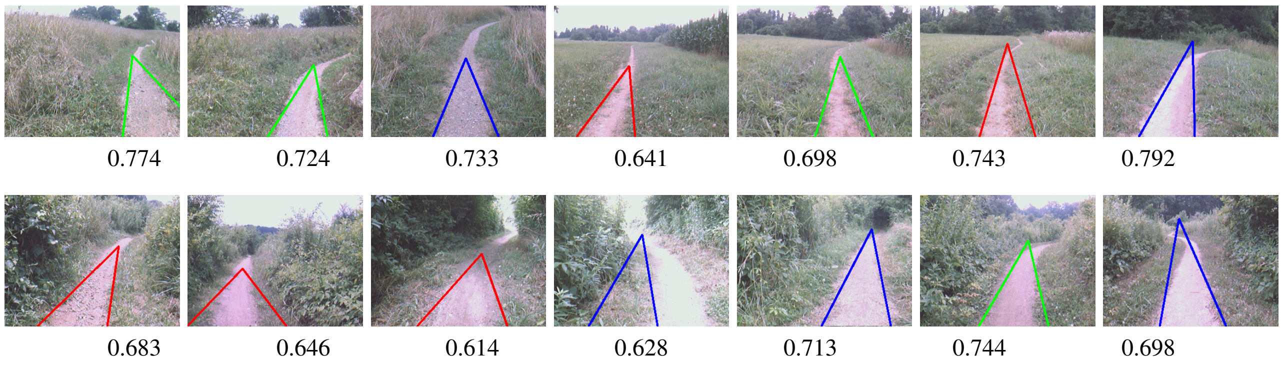


Fig. 8. Results of trail triangle tracker on long hiking trail sequence, shown at 1000-frame intervals. Colors indicate which features were automatically chosen as most discriminative, numbers are overlap with ground truth on that frame. The mis-segmentation in the middle of the bottom row could be eliminated with hysteresis of the trail color.

[7] C. Rasmussen, "Combining laser range, color, and texture cues for autonomous road following," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2002.

[8] J. Zhang and H. Nagel, "Texture-based segmentation of road images," in *Proc. IEEE Intelligent Vehicles Symposium*, 1994.

[9] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, K. Lau, C. Oakley, M. Palatucci, V. Pratt, P. Stang, S. Strohband, C. Dupont, L.-E. Jendrossek, C. Koelen, C. Markey, C. Rummel, J. van Niekerk, E. Jensen, P. Alessandrini, G. Bradski, B. Davies, S. Ettinger, A. Kaehler, A. Nefian, and P. Mahoney, "Stanley, the robot that won the darpa grand challenge," *Journal of Field Robotics*, vol. 23, no. 9, 2006.

[10] C. Urmson, J. Anhalt, D. Bartz, M. Clark, T. Galatali, A. Gutierrez, S. Harbaugh, J. Johnston, H. Kato, P. Koon, W. Messner, N. Miller, A. Mosher, K. Peterson, C. Ragusa, D. Ray, B. Smith, J. Snider, S. Spiker, J. Struble, J. Ziglar, and W. Whittaker, "A robust approach to high-speed navigation for unrehearsed desert terrain," *Journal of Field Robotics*, vol. 23, no. 8, pp. 467–508, 2006.

[11] 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.

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

[13] C. U. et al., "Autonomous driving in urban environments: Boss and the urban challenge," *Journal of Field Robotics*, vol. 25, no. 8, 2008.

[14] 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.

[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

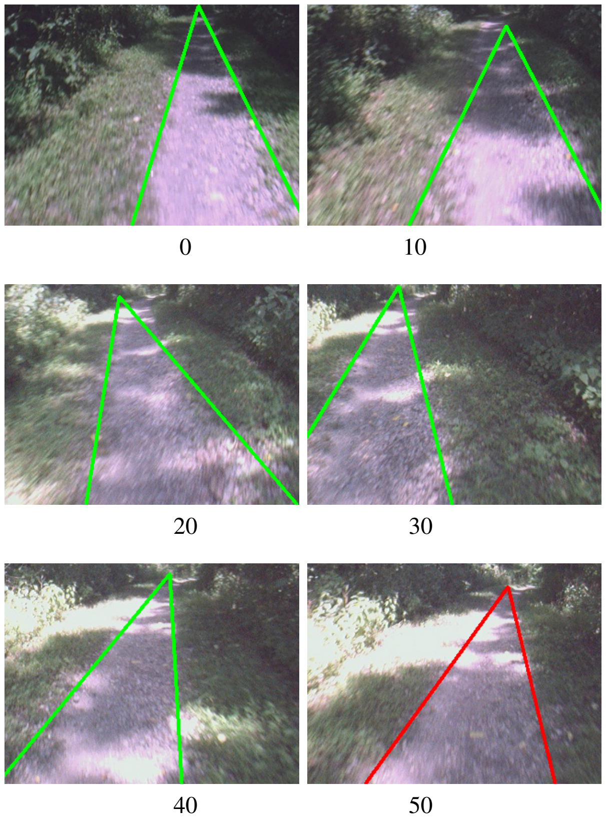


Fig. 9. Negotiation of trail section with strong shadow and exposure changes

propagation to learn long-range traversability," in *Robotics: Science and Systems*, 2007.

[16] 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.

[17] 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.

[18] A. Kleiner and C. Dornhege, "Real-time localization and elevation mapping within urban search and rescue scenarios," *Journal of Field Robotics*, vol. 24, no. 8–9, pp. 723–745, 2007.

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

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

[21] E. Borenstein and J. Malik, "Shape guided object segmentation," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2006.

[22] T. Cour and J. Shi, "Recognizing objects by piecing together the segmentation puzzle," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2007.

[23] A. Levin and Y. Weiss, "Learning to combine bottom-up and top-down segmentation," in *Proc. European Conf. Computer Vision*, 2006.

[24] G. Mori, "Guided model search using segmentation," in *Proc. Int. Conf. Computer Vision*, 2005.

[25] J. Reynolds and K. Murphy, "Figure-ground segmentation using a hierarchical conditional random field," in *Canadian Conf. on Computer and Robot Vision*, 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] J. Wang, E. Gu, and M. Betke, "Mosaicshape: Stochastic region grouping with shape prior," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2005.

[28] P. Felzenszwalb and D. Huttenlocher, "Efficient graph-based image segmentation," *Int. J. Computer Vision*, vol. 59, no. 2, 2004.

[29] J. Shi and J. Malik, "Normalized cuts and image segmentation," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 1997.

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

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

[32] B. Russell, A. Torralba, K. Murphy, and W. Freeman, "Labelme: a database and web-based tool for image annotation," MIT AI Lab, Tech. Rep. AIM-2005-025, 2005.

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

[34] D. Martin, C. Fowlkes, and J. Malik, "Learning to detect natural image boundaries using local brightness, color, and texture cues," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 26, no. 5, 2004.

[35] H. Dunlop, D. Thompson, and D. Wettergreen, "Multi-scale features for detection and segmention of rocks in mars images," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2007.

[36] M. Varma and A. Zisserman, "A statistical approach to texture classification from single images," *Int. J. Computer Vision*, vol. 26, 2005.

[37] 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.

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

[39] C. Rother, V. Kolmogorov, and A. Blake, "Grabcut - interactive foreground extraction using iterated graph cuts," in *SIGGRAPH*, 2004.