Shape-Guided Superpixel Grouping for Trail Detection and Tracking

Christopher Rasmussen
Dept. Computer & Information Sciences
University of Delaware
cer@cis.udel.edu

Donald Scott
Dept. Computer & Information Sciences
University of Delaware
donald@udel.edu

***Abstract*— We describe a framework for detecting and tracking continuous ”trails” in images and image sequences for autonomous robot navigation. Continuous trails are extended regions along the ground such as roads, hiking paths, rivers, and pipelines which can be navigationally useful for ground-based or aerial robots. Our approach to single-image trail segmentation incorporates both bottom-up and top-down processes. First, good grouping hypotheses are efficiently generated by probabilistic clustering of superpixels based on color similarity. Second, hypotheses are robustly ranked with an objective function comprising shape, appearance, and deformation terms. The shape term measures how well a triangle, the approximate template for a trail viewed under perspective, can be fit to the grouping's boundary. The appearance term reflects the visual contrast between the grouping and its surroundings using a between-class/within-class scatter measure. Finally, the deformation term measures the closeness of the fitted triangle to a learned distribution which captures expected size, location, and other degrees of shape variation.**

Although trail detection is accurate and reasonably fast on a variety of isolated images, we describe how introducing temporal filtering to both the bottom-up and top-down stages increases segmentation accuracy and per-frame speed over image sequences. Results are shown on varied sequences collected from flying and driving platforms, as well as images sampled from the Web.

I. Introduction

Navigationally-useful linear features along the ground, or *trails*, are ubiquitous in manmade 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 computer vision tasks involved in robust trail-following may be divided into three categories:

Finding We use this term for the problem of detecting or segmenting a trail in an image with very little or no *a priori* information about its location, appearance, and specific shape. Corollary issues include deciding whether the trail is coming to a dead-end or a branch, which might be more generally framed as whether a trail is visible at all, or how many trails are in view if there may be more than one.

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.

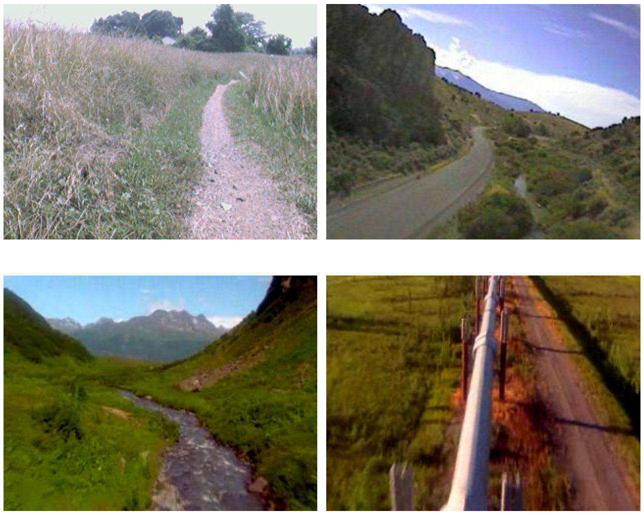


Fig. 1. Example trail types. Clockwise from upper-left: ground view of hiking trail, aerial views of canyon road, pipeline, and river.

Negotiation When a trail contains hazards such as rocks, roots, logs, and puddles, the gross shape estimate may be insufficient for safe travel. Individual detectors for each class of obstacle are necessary for avoidance maneuvers and other control policy adjustments.

In this paper we describe a vision-based approach to part of the overall problem—namely, the tasks of *finding* and *keeping* to a single, non-branching, non-terminating, continuous trail. Our method has two major parts: (1) a randomized search for the most-salient, most-trail-shaped region in a single image with no *a priori* color model; and (2) efficiently tracking a previously-found trail region over an image sequence.

An obvious analogy is to consider trails as a scaled-down version of road following. Vision-based road following has been thoroughly studied on paved and/or painted roads with sharp edges [19], [29], [30], but some of the subtypes of trails we are interested in lack these (e.g., the hiking trail and river in Figure 1). When the road border is ragged, bottom-up methods using color or texture to classify image patches as road vs. background often work well [1], [5], [20], [35], but they tend to function best with manual initialization and a large road region. This last factor is problematic for a low-altitude aerial (i.e., UAV) perspective. From the

This work was supported by an NSF CAREER award (IIS-0546410)

air, trail pixels rarely dominate the image, making mere detection of the trail among many other features a difficult challenge. There has been one cluster of work in this area from essentially a top-down perspective [10], [21], [22], but their algorithms require fairly constrained imagery. [31] has some results on finding straight roads in low-altitude aerial images using a Hough transform on static images only, with no tracking.

Our motivating intuition here is that the *shape* of trails is subject to less variability than their appearance, and therefore should be a very strong cue in finding them. There has been much recent work on combining bottom-up grouping with top-down shape constraints for segmentation and object detection [3], [4], [14], [17], [25], [27], [34] A common feature of many such approaches is their use of an oversegmentation technique such as [8], [28] 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. Many of these researchers, however, are trying to detect complicated, often articulated objects (e.g., horses, people, things from Caltech101, etc.) and must struggle with a large search space for the top-down stage. We are basically looking for a deformation of a *triangle*, which is a reasonable template to describe an unoccluded trail viewed under perspective, modulo high-frequency variation along the edges and some nonlinearity due to curvature near the horizon. The two most similar approaches to ours are [27], [34], which search over low-dimensional transformations of relatively simple shapes like bananas and street signs in fairly uncluttered images. Their methods allow for multiple detections if multiple instances of an object are present, which requires extra work, whereas we for now assume one and only one trail is present in every image. However, we are trying to make our method work in difficult outdoor images.

There has not been much recent work on combining shape-based segmentation and tracking, probably because of the computational cost of some of the superpixel generation algorithms. [24] used a novel technique, constrained Delaunay triangulation, to obtain its superpixels and perform foreground/background classification on grayscale image sequences. [15] used the fast segmentation method of [8] to extract superpixels as a precursor to foreground/background classification. However, to work on sequences their method requires that the first frame be manually labeled. Finally, a few somewhat less global shape-oriented uses of superpixels for segmentation include surface layout classification [12] and [13]'s work on classifying superpixels as obstacle or free space for outdoor robot navigation.

In the following sections we will first describe our method for finding a single trail in a single image, how we modify the method to take advantage of temporal consistency over image sequences while increasing efficiency, and show results as space allows.

II. Single-Image Trail Segmentation

Here we explain the process for detecting a trail in a single $w \times h$ image. We assume that there is exactly one such trail region, roughly in the shape of a triangle with its base aligned with the bottom edge of the image. The basic idea is generate a set of hypothetical trail regions which are reasonable in that they are self-similar and connected, score them using several global shape and appearance criteria, and pick the hypothesis with the best overall likelihood.

A. Generating a trail region hypothesis

A trail region hypothesis is a grouping G of connected superpixels $\{s_1, \dots, s_m\}$. G is constructed through an m-stage agglomerative process such that $G_1 \subset G_2 \subset \dots \subset G_{m-1} \subset G_m = G$. Each stage adds a superpixel to the group: the *seed* hypothesis consists of a single pixel $G_1 = \{s_1\}$, and at stage i a *next member* superpixel s_{next} is chosen from the current grouping's set of neighbors $\mathcal{N}(G_i)$ to form $G_{i+1} = G_i \cup s_{next}$. The seed s_1 always borders the bottom of the image. It is chosen as the superpixel containing a random bottom-row pixel $(x, h-1)$ with x drawn uniformly.

The next member superpixel to add to the current grouping is chosen as follows. Let $\delta(s_j, G_i)$ be the *appearance distance* between a neighboring superpixel $s_j \in \mathcal{N}(G_i)$ and the grouping (details are given in Section II-A.1). A purely greedy approach (e.g., the "best-first" method of [27]) would deterministically select the next member of the hypothetical grouping as $\text{argmin}_j \, \delta(s_j, G_i)$. However, for a fuller exploration of the space of hypothetical groupings (i.e., ergodicity in the Markov sense [32]), we randomize the process by converting each superpixel j's appearance distance $\delta(s_j, G_i)$ to a "next member probability" $p_{next}(s_j|G_i)$ via simple normalization

$$p_{next}(s_j|G_i) = \frac{1/\delta(s_j, G_i)}{\sum_{k=1}^{n} 1/\delta(s_k, G_i)} \tag{1}$$

where n is the number of superpixels neighboring G_i. Now the next member of the grouping s_{next} is just sampled from the neighbors according to $p_{next}(s_j|G_i)$.

The final number of superpixels m in grouping G is determined probabilistically, based on two factors: the grouping's overall size and its appearance variation. For each hypothesis, a maximum number of stages of agglomeration m_{max} is randomly chosen (the upper bound on this was set based on the superpixel segmentation parameters to effectively be an area range). The hypothesis process may terminate early, however, if the selected next member superpixel s_{next} is too dissimilar to G_i. Let $var(G_i)$ be a measure of G_i's internal appearance variation in the same units as the appearance distance function δ (see Section II-A.1). The "acceptance probability" $p_{accept}(s_{next}|G_i)$ is inversely proportional to $\delta(s_{next}, G_i)/var(G_i)$.

1) Appearance measures: A number of different measures have been successfully used to measure appearance similarity in superpixel-based segmentation work, including brightness in grayscale images [23], Euclidean color distance (often in CIELAB space [16]), and color and texture histogram dissimilarity measures such as χ^2 [7], [33] and the Earth Mover's Distance [17].

For efficiency, in this work the appearance distance between a superpixel s and grouping G is based on Euclidean distance in RGB space. Let the mean color of any image region $\mathcal{R}$ be given by $\boldsymbol{\mu}(\mathcal{R}) = (\mu_R(\mathcal{R}), \mu_G(\mathcal{R}), \mu_B(\mathcal{R}))$. Then the appearance distance $\delta(s, G) = |\boldsymbol{\mu}(s) - \boldsymbol{\mu}(G)|$.[1] Letting the vector of color channel standard deviations of a region be analogously defined as $\boldsymbol{\sigma}(\mathcal{R})$, the appearance variation of a grouping is the magnitude $var(G) = |\boldsymbol{\sigma}(G)|$.

B. Scoring groupings

The boundary of a trail under perspective is linearly approximated by a triangle with its base coincident with the bottom of the image. This *trail triangle* is our shape template in the sense of [27]. 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 grouping G will thus be referred to in a *region* sense as T_G and in a *point* sense as $\mathbf{t}_G$.

The overall goodness of a trail region hypothesis G is assessed through a *trail likelihood* function $L_{\mathit{trail}}(G)$ comprising three terms derived from [27]'s formulation. These terms encompass (1) how well G's *shape* is described by a triangle, (2) how internally consistent its *appearance* is while contrasting with its surroundings, and (3) how much distance or *deformation* there is between the best-fitting triangle and the most-likely trail triangle based on a learned distribution. The formula is:

$$L_{\mathit{trail}}(G) = \\ uL_{\mathit{shape}}(G) + vL_{\mathit{appear}}(G) + (1 - u - v)L_{\mathit{deform}}(G) \quad (2)$$

L_{shape}, L_{appear}, and L_{deform} (detailed below) are all constructed to have values in the range $[0, 1]$, and $u = v = \frac{1}{3}$ for all of the results in this paper (although tuning them could help greatly).

1) Shape likelihood: We want the shape likelihood to measure how "triangular" G is, which is equivalent to the shape similarity between G and its best-fit triangle T_G. T_G is initially estimated by a simple fitting procedure: $\mathbf{p}^t$ is set equal to G's highest point, $\mathbf{p}^l$ is set to G's leftmost point within some tolerance of the bottom of the image (since G always has at least one superpixel bordering the image bottom), and $\mathbf{p}^r$ is set to G's rightmost point near the image bottom.

There are no issues here with registering the two shapes first since T_G is derived from G, so the direct area overlap computation suggested by [27] works very well:

$$L_{\mathit{shape}}(G) = \frac{A(G \cap T_G)^2}{A(G)A(T_G)} \quad (3)$$

where $A(R)$ is the area of a region R, and $G \cap T_G$ is the region of intersection between the grouping and its fitted triangle. This quantity is quickly calculated from a polygonal representation of G via clipping against the left and right edges of T_G.

The accuracy of the fitted triangle T_G is most susceptible to noise in the superpixel segmentation or trail curvature near the apex $\mathbf{p}^t$. Significant improvement is often obtained by searching over the x coordinate of $\mathbf{p}^t$ for a local maximum of $L_{\mathit{shape}}(G)$; this is only done after the best grouping hypothesis is chosen for reasons of speed.

2) Appearance likelihood: This is a measure of how different a grouping G's neighboring superpixels are from it in appearance and how much variation there is in the interior of G. Grouping-to-neighbor difference, which we wish to maximize, and within-grouping variation, which we wish to minimize, are somewhat analogous to the concepts of between-class scatter and within-class scatter from linear discriminant analysis [11]. Thus, the form of the appearance likelihood is also a ratio:

$$L_{\mathit{appear}}(G) = \frac{\sum_{k=1}^{n} w(s_k)\delta(s_k, G)}{var(G)} \quad (4)$$

where $w(s_k)$ is the fraction of the total border length of all superpixels neighboring G due to s_k. This is different from [27]'s formulation, which accounts for variation within a grouping but not its contrast with the surroundings. It is quite similar to [18], which cites [23] as an antecedent.

3) Deformation likelihood: This term measures how close the trail triangle T_G is to a distribution which represents expectations about the trail's apparent width, centeredness, horizon line curvature, and so on. These of course depend on both the trail's intrinsic shape properties as well as the camera perspective. In the absence of any specific knowledge of these parameters for a given image or sequence, we use a Gaussian distribution $(\bar{\mathbf{t}}, \boldsymbol{\Sigma})$ learned from user-labeled examples. The deformation likelihood is derived from the Mahalanobis distance to the grouping's fitted triangle $d(\mathbf{t}_G, (\bar{\mathbf{t}}, \boldsymbol{\Sigma})) = (\mathbf{t}_G - \bar{\mathbf{t}})^T \boldsymbol{\Sigma}^{-1} (\mathbf{t}_G - \bar{\mathbf{t}})$:

$$L_{\mathit{deform}}(G) = e^{-\gamma d(\mathbf{t}_G, (\bar{\mathbf{t}}, \boldsymbol{\Sigma}))} \quad (5)$$

The distribution used here was learned from an image sequence consisting of several hundred frames captured by a wheeled robot over a multi-km hiking trail with a variety of turns and width changes (the hiking trail image in Figure 1 was captured on the same trail). The trail region R_i in each image i was manually-segmented using a local copy of the LabelMe tool [26]. A triangle $\mathbf{t}_i$ was fitted to each R_i and $(\bar{\mathbf{t}}, \boldsymbol{\Sigma})$ calculated over the entire set of triangles $\{\mathbf{t}_i\}$.

[1] This value is squared in the computation of δ to penalize more dissimilar superpixels.

III. Tracking Trails in Image Sequences

In this scenario, a robot or camera platform is moving along a trail and capturing a sequence of images. We would like to infer the most likely sequence of trail regions to describe what we are seeing. Of course we could run the single-image procedure described above repeatedly, but this does not enforce the expected frame-to-frame consistency of our interpretations (though we show some results on sequences in the next section). In this section we describe how we use the trail region estimate from the previous frame to guide segmentation and fitting in the current frame.

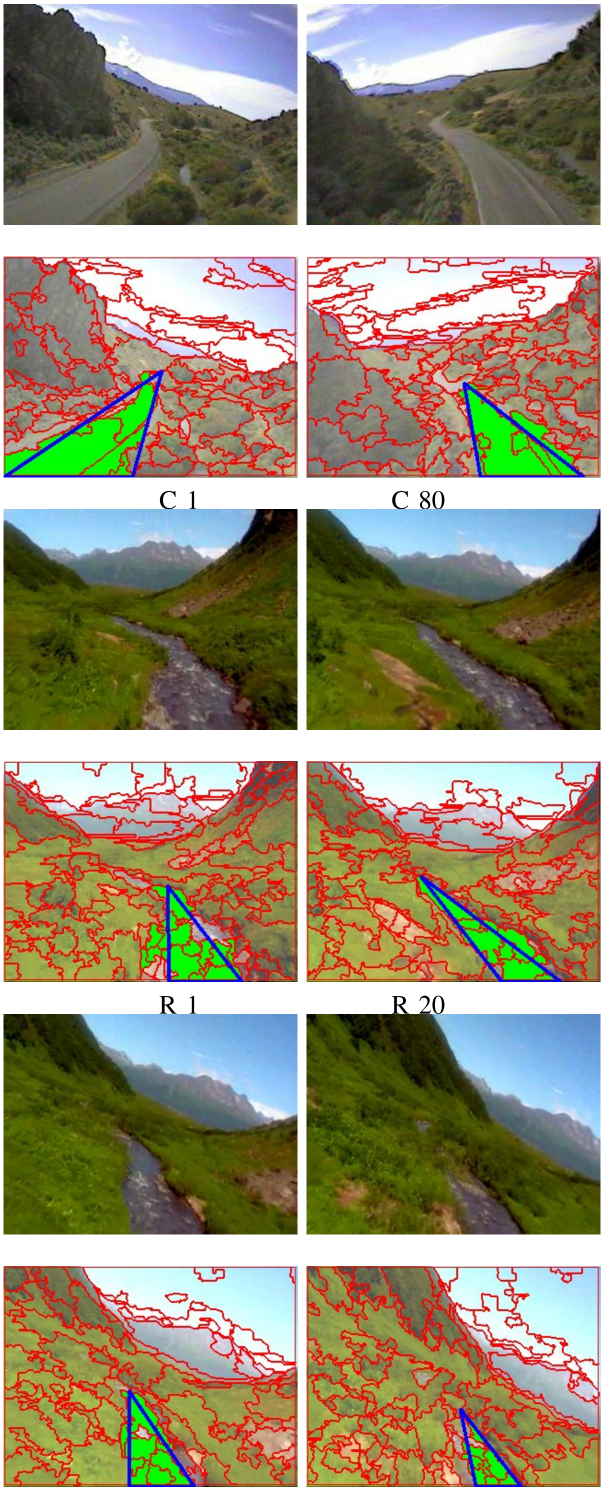
Two things may change between frames: the apparent shape of the trail as we move along it and new sections come into view, and the trail appearance as its material composition and illumination conditions change. Because trail terrain is often highly-nonplanar and trail curves (particularly for hiking trails) may not be well-modeled by analytic curves, it is most convenient to carry out tracking strictly in the image domain.

Our high-level idea is to carry forward the triangle T_G^{t-1} fitted to the best-scoring grouping in the previous frame to the current frame via a suitable dynamical model [2] and search for the "nearest" good grouping hypothesis to T_G^t. To find this, consider the grouping G^* formed by all of the superpixels in the new frame which significantly overlap T_G^t. These constitute a top-down hypothesis about where the trail is now. We assume that the shape and deformation likelihood of this grouping are good given how it was formed, but how is its appearance likelihood? We wish to find what transformation of T_G^t brings it into best alignment with the current trail position and thus maximizes $L_{appear}(G^*)$.

There are numerous possible ways to do this, deterministic and stochastic, but for this paper we simply randomly sample trail triangles around T_G^t, evaluating the appearance likelihoods of the associated groupings G^*, and picking the best one. This is considerably cheaper than carrying out the full grouping process of Section II-A. A more principled analog of this would be to carry out particle filtering [2].

IV. Results

We have run the single-image trail finder on numerous images from the hiking trail, river, and canyon sequences depicted in Figure 1 with Felzenszwalb's superpixel segmentation code [8] as the front-end (for all results in this paper, $\sigma = 0.5, k = 50, \min = 100$). On the hiking trail and canyon sequences there is excellent gross accuracy of trail detection and fairly good frame-to-frame correlation considering the variability of the superpixels and the lack of temporal filtering. The first row of Figure 2 shows two canyon images 80 frames apart and their best-scoring groupings with fitted trail triangles. $n = 100$ grouping hypotheses were generated and scored by the single-image trail finder described above for each 320×240 image, taking less than 1 s per image on a Core Duo T2600 2.16 GHz laptop. Despite dynamic pitching and rolling of the UAV, 95% of the trail detections for the frames between were substantially correct. The second and third rows of the figure show results from the river sequence. This was a more difficult set of images because of the shape and appearance variation of the water, particularly the white of the rapids (see discussion in Conclusion). Nonetheless, over the approximately 60 images spanning by this clip, the


C 1

C 80

R 1

R 20

R 40

R 60

Fig. 2. Aerial sequence excerpts: Single-image detections at equally-spaced intervals (C = canyon sequence frame numbers; R = river sequence frame numbers).

river-trail region was roughly correct for almost 70% of the frames.

Next we present results on the diversity of trail images that the system can process successfully. In Figure 3, 10 images from a larger collection culled from Flickr and Google image searches are shown. The best-scoring grouping—with no further post-processing—and its fitted trail triangle is shown below each input image. The algorithm does a good job of finding the trail region in each image despite their very different colors, sizes, and image locations. This procedure took about 5 seconds per image with $n = 1000$. Running with $n = 100$ hypotheses yields similar results with somewhat rougher grouping edges. For contrast, the superpixel step alone takes about 3 minutes for each of these images using [17] and more than 10 minutes per image using the multiscale normalized cuts of [4].

Finally, we show some results from the tracking procedure running on the hiking trail sequence in Figure 4. The tracker was run on every single image of a multi-thousand frame sequence captured at 5 fps from a robot moving at an average speed of 1 m /s. The figure shows 20-frame intervals from one section. Over the sequence, which has a number of dips and curvy sections, the tracker did not lose the trail, although the fit of the trail triangle was occasionally sloppy, as the sixth frame shows. This should be mostly mitigated by a better dynamical model (currently we are just modeling the trail triangle's motion with nothing more than a random walk). This mode takes about 0.65 s per image after the initial detection; most of this is due to the superpixel segmentation that must be run on each new image.

V. Conclusion

We have presented a practical yet novel approach to visually finding and following 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. The segmentation approach is readily extensible to other cues such as obstacles detected by ladar *à la* [6].

There are number of directions to explore for improvement of the algorithm. First, our essentially unimodal color model for the interior of a grouping fails on highly-variable regions such as scattered leaves or the boulder pile of Figure 5. This will be remedied with a more sophisticated appearance model, possibly using a mixture of Gaussians for multiple color [6] or by explicitly modeling texture [33]. Second, the algorithm has problems when the trail region is broken into several segments separated by areas of visual contrast such as occluding branches, shadows, lane lines, or other features such as the rocks in the stream in Figure 5. These can prevent the hypothesis generator, which works by adding neighboring superpixels with an appearance affinity to the grouping, from reaching the other side of the blockage unless there is another path. What is necessary is a grouping mechanism which can "jump" such obstructions by considering neighbors as nearby superpixels rather than requiring that they abut. For cast shadows in particular, which can occupy relatively large areas, it may be helpful to also apply an explicit shadow removal method [9] or at least employ a more illumination-insensitive color similarity measure.

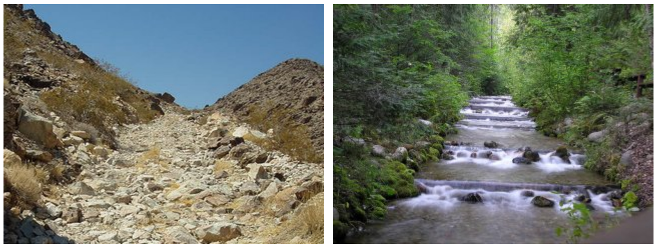


Fig. 5. Difficult images for the trail finder: multi-modal trail color/texture distribution (left) and isolated segments (right)

References

[1] N. Apostoloff and A. Zelinsky. Robust vision based lane tracking using multiple cues and particle filtering. In *Proc. IEEE Intell. Vehicles Symposium*, 2003.

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

[3] E. Borenstein and J. Malik. Shape guided object segmentation. In *Proc. IEEE Conf. Comp. Vision & Patt. Recognition*, 2006.

[4] T. Cour and J. Shi. Recognizing objects by piecing together the segmentation puzzle. In *Proc. IEEE Conf. Comp. Vision & Patt. Recognition*, 2007.

[5] J. Crisman and C. Thorpe. UNSCARF, a color vision system for the detection of unstructured roads. In *Proc. IEEE Int. Conf. Robotics & Automation*, pages 2496–2501, 1991.

[6] H. Dahlkamp, A. Kaehler, D. Stavens, S. Thrun, and G. Bradski. Self-supervised monocular road detection in desert terrain. In *Robotics: Science & Systems*, 2006.

[7] H. Dunlop, D. Thompson, and D. Wettergreen. Multi-scale features for detection and segmention of rocks in mars images. In *Proc. IEEE Conf. Comp. Vision & Patt. Recognition*, 2007.

[8] P. Felzenszwalb and D. Huttenlocher. Efficient graph-based image segmentation. *Int. J. Comp. Vision*, 59(2), 2004.

[9] G. Finlayson, S. Hordley, C. Lu, and M. Drew. On the removal of shadows from images. *IEEE Trans. Patt. Analysis & Mach. Intell.*, 28(1), 2006.

[10] E. Frew, T. McGee, Z. Kim, X. Xiao, S. Jackson, M. Morimoto, S. Rathinam, J. Padial, and R. Sengupta. Vision-based road following using a small autonomous aircraft. In *IEEE Aerospace Conf.*, 2004.

[11] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer-Verlag, 2001.

[12] D. Hoiem, A. Efros, and M. Hebert. Geometric context from a single image. In *Proc. Int. Conf. Comp. Vision*, 2005.

[13] D. Kim, S. Oh, and J. Rehg. Traversability classification for ugv navigation: A comparison of patch and superpixel representations. In *Proc. Int. Conf. Intell. Robots & Systems*, 2007.

[14] A. Levin and Y. Weiss. Learning to combine bottom-up and top-down segmentation. In *Proc. European Conf. Comp. Vision*, 2006.

[15] L. Lu and G. Hager. Dynamic foreground/background extraction from images and videos using random patches. In *Advances in Neural Information Processing Systems*, 2006.

[16] D. Martin, C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Trans. Patt. Analysis & Mach. Intell.*, 26(5), 2004.

[17] G. Mori. Guided model search using segmentation. In *Proc. Int. Conf. Comp. Vision*, 2005.

[18] G. Mori, X. Ren, A. Efros, and J. Malik. Recovering human body configurations: Combining segmentation and recognition. In *Proc. IEEE Conf. Comp. Vision & Patt. Recognition*, 2004.

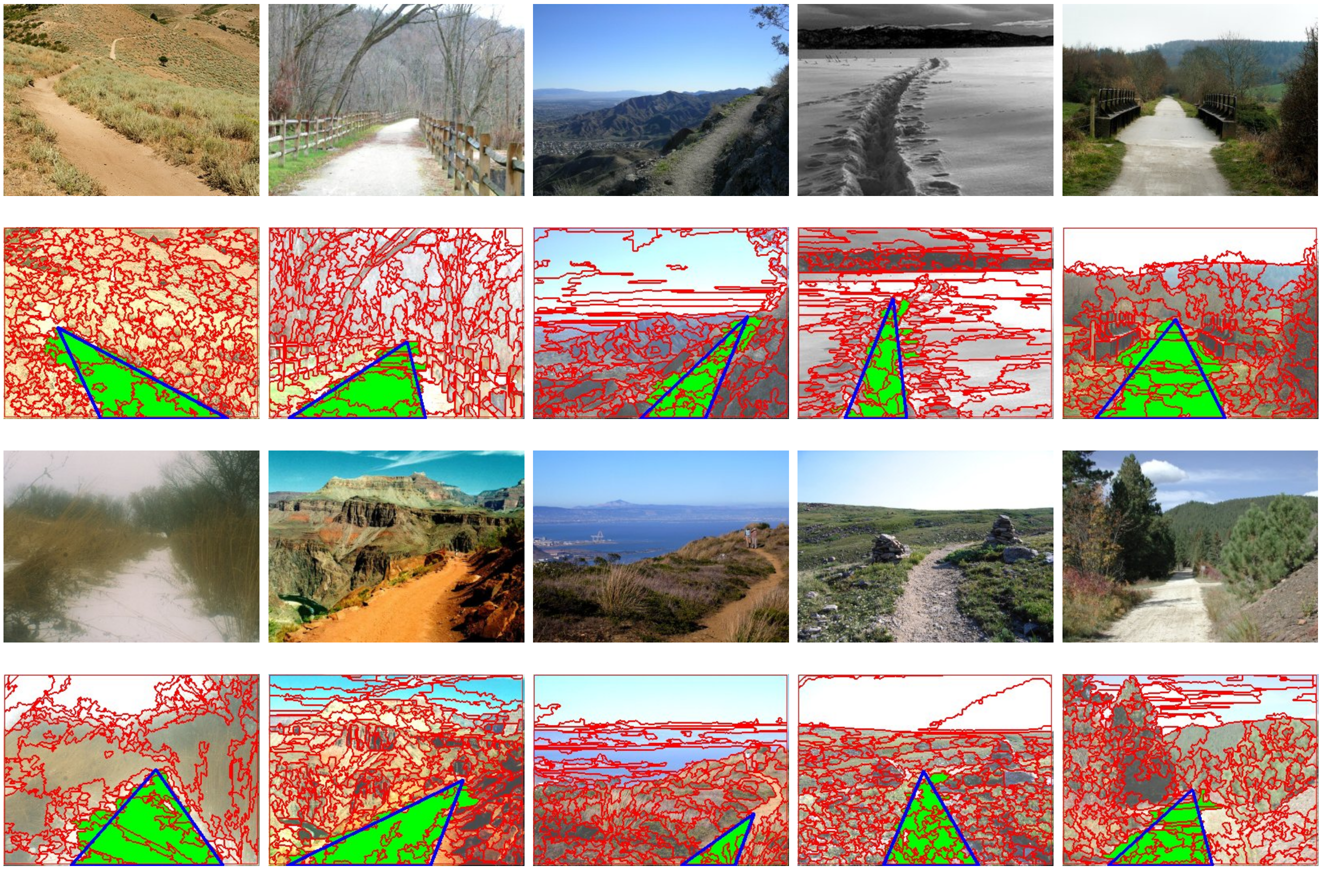


Fig. 3. Sampling of web trail images and output of single-image trail finder. The source images were cropped to a common aspect ratio as necessary (some were originally vertical) and scaled to 320 × 240.

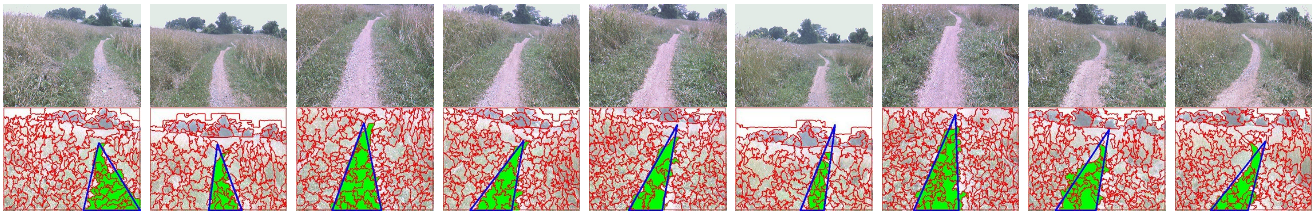


Fig. 4. Tracking trails: Results from hiking trail sequence at 20 frame intervals

[19] D. Pomerleau. RALPH: Rapidly adapting lateral position handler. In *Proc. IEEE Intell. Vehicles Symposium*, pages 506–511, 1995.

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

[21] S. Rathinam, P. Almeida, Z. Kim, S. Jackson, A. Tinka, W. Grossman, and R. Sengupta. Autonomous searching and tracking of a river using an UAV. In *American Control Conf.*, 2007.

[22] S. Rathinam, Z. Kim, A. Soghikian, and R. Sengupta. Vision based following of locally linear structures using an unmanned aerial vehicle. In *IEEE Conf. on Decision & Control*, 2005.

[23] X. Ren and J. Malik. Learning a classification model for segmentation. In *Proc. Int. Conf. Comp. Vision*, 2003.

[24] X. Ren and J. Malik. Tracking as repeated figure/ground segmentation. In *Proc. IEEE Conf. Comp. Vision and Patt. Recognition*, 2007.

[25] J. Reynolds and K. Murphy. Figure-ground segmentation using a hierarchical conditional random field. In *Canadian Conf. on Comp. & Robot Vision*, 2007.

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

[27] S. Sclaroff and L. Liu. Deformable shape detection and description via model-based region grouping. *IEEE Trans. Patt. Analysis & Mach. Intell.*, 23(5), 2001.

[28] J. Shi and J. Malik. Normalized cuts and image segmentation. In *Proc. IEEE Conf. Comp. Vision & Patt. Recognition*, 1997.

[29] B. Southall and C. Taylor. Stochastic road shape estimation. In *Proc. Int. Conf. Comp. Vision*, pages 205–212, 2001.

[30] C. Taylor, J. Malik, and J. Weber. A real-time approach to stereopsis and lane-finding. In *Proc. IEEE Intell. Vehicles Symposium*, 1996.

[31] S. Todorovic and M. Nechyba. A vision system for intelligent mission profiles of micro air vehicles. *IEEE Trans. Vehicular Technology*, 53(6), 2004.

[32] Z. Tu and S. Zhu. Image segmentation by data-driven markov chain monte carlo. *IEEE Trans. Patt. Analysis & Mach. Intell.*, 24(5), 2002.

[33] M. Varma and A. Zisserman. A statistical approach to texture classification from single images. *Int. J. Comp. Vision*, 26, 2005.

[34] J. Wang, E. Gu, and M. Betke. Mosaicshape: Stochastic region grouping with shape prior. In *Proc. IEEE Conf. Comp. Vision & Patt. Recognition*, 2005.

[35] J. Zhang and H. Nagel. Texture-based segmentation of road images. In *Proc. IEEE Intell. Vehicles Symposium*, 1994.