A Statistical Model for Recreational Trails in Aerial Images

Andrew Predoehl
University of Arizona
predoehl@cs.arizona.edu

Scott Morris
TopoFusion
smorris@topofusion.com

Kobus Barnard
University of Arizona
kobus@sista.arizona.edu

Abstract

We present a statistical model of aerial images of recreational trails, and a method to infer trail routes in such images. We learn a set of textons describing the images, and use them to divide the image into super-pixels represented by their texton. We then learn, for each texton, the frequency of generating on-trail and off-trail pixels, and the direction of trail through on-trail pixels. From these, we derive an image likelihood function. We combine that with a prior model of trail length and smoothness, yielding a posterior distribution for trails, given an image. We search for good values of this posterior using a novel stochastic variation of Dijkstra's algorithm. Our experiments, on trail images and groundtruth collected in the western continental USA, show substantial improvement over those of the previous best trail-finding method.

1. Introduction

Recreational trails like the one shown in Figure 1a represent a challenge for computer vision. They lack well defined rigid shape, they occupy relatively few pixels in an aerial image (Fig. 1b), they are surrounded by highly variable clutter, and are frequently obscured by trees. Trails are usually smaller and more winding than vehicular roads, and their shapes demand statistical description. Automatic identification of trails would be useful not only for bikers, hikers, and land managers, but also as a model problem in other domains that depend on identifying locally-linear structures in images, such as blood vessels or neurons grown *in vitro*.

In this paper we present a statistical model describing trails and trail images, based on a segmentation of images by textures. The image model uses a set of *textons*, a characteristic set of texture elements learned from training data [26]. For each texton, we learn its probability of generating on-trail pixels (Fig. 1c), and, when it does so, the direction in which that trail is oriented (Fig. 1d). Using this image model, we derive a likelihood function for a trail's image (§2.2). That, combined with a simple prior for our trail model (§2.1), defines a posterior distribution for trails.

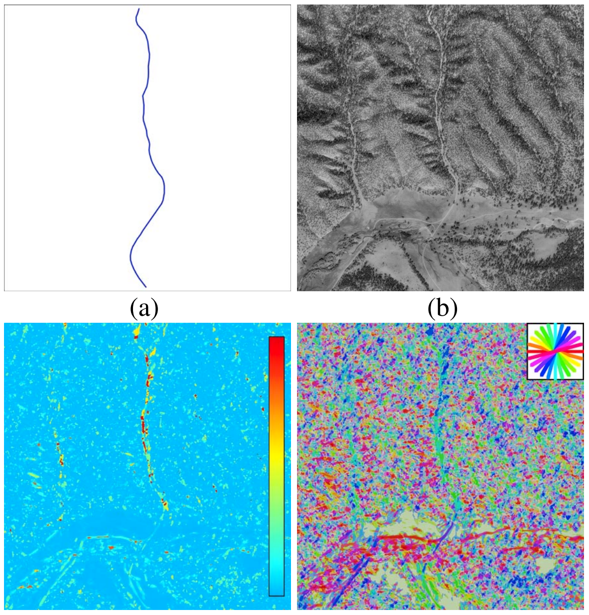


Figure 1. Example trail (a), trail image (b), and learned characteristics of image textons (c, d). In (c), color indicates the ratio of frequencies, for each superpixel's texton label, at which that texton generates on- and off-trail pixels. In (d), hue and saturation show the trail direction of each superpixel's texton, conditioned on it generating a trail pixel. Hue indicates direction, and saturation indicates axial concentration. (Best viewed in color.)

In order to infer the route of a latent trail between two known endpoints, we use a sampling approach to search this posterior for a good value. Like earlier work in road- and trail-finding, e.g. [29], we build a graph from the image evidence, and generate inference proposals as simple paths in this graph. More specifically, by carefully constructing the edge weights (§3.1) we produce a graph in which short paths between nodes tend to have high likelihood in the statistical model. In §3.2 we present a novel heuristic method, related to Dijkstra's algorithm, that uses this inexact dual relationship to propose independent data-driven route proposals. Using our model to judge among the proposals, we retain the most probable route as the best one.

1.1. Previous work

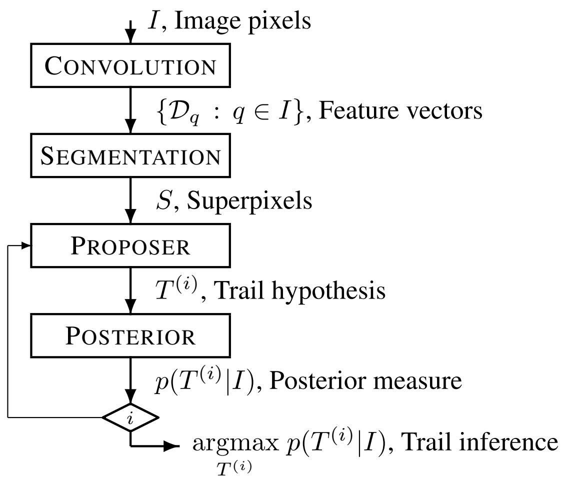
Locally-linear structures are found in many problem domains. For example, biomedical researchers and clinicians need to trace neurons [2, 3] and blood vessels [1]. They have used a variety of approaches, such as active contours [21], minimal paths [9], and many other methods. See [23, 25] for thorough recent reviews. However, there are substantial differences between trail extraction and vessel segmentation. Trails tend to be thinner than typical vessels (our model neglects their width) and they are more often occluded entirely. In addition, the clutter surrounding trails tends to be more variable, which motivates our approach based on textons and superpixels.

Trail-finding shares many characteristics with road-finding, which has a substantial literature. Reviews of road-finding literature can be found in [6] and [28]. Bottom-up approaches using two or three steps are common to many road-finding applications, going back at least to Fischler *et al.* [15]. Statistical approaches are less common, but the work of Geman and Jedynak [17] is an interesting exception. They too developed a statistical model of roads and road images, but assumed more conditional independence in their image data than we did. Their *active testing* method computes an explicit decision tree while tracing a road. In contrast, each invocation of our quasi-Dijkstra procedure produces a random path that is essentially one realization of an implicit decision tree.

To our knowledge, the only vision literature specific to trail-finding is that of Morris and Barnard [29], who trace trails via a two-step process comparable to other road-finders: they assign each image pixel an energy value based on a Naive Bayes classifier, then search for a path through the pixel graph that minimizes an energy function employing both local and global factors. Our approach differs from theirs in several aspects. First, we base our objective function on a more comprehensive statistical characterization of image textures. Second, we incorporate another high-level step—an intermediate super-pixel segmentation (see Fig. 2) based on textons, justified by the naivety of independent pixel characteristics in this domain. Third, our image likelihood function uses a statistical model of trail direction learned for the textons. Fourth, our trail proposal scheme differs substantially from theirs.

Many road-finding methods are based on solving a dynamic programming or shortest path problem in a pixel or similar lattice. Similarly, our trail proposer finds short, but not necessarily shortest, paths in the superpixel graph, as a heuristic method to explore the space of routes between trail termini. This proposer is implemented by modifying Dijkstra's algorithm to introduce an element of randomness into its main loop (§3.2), which we believe is a novel approach.

The challenge of characterizing the resulting distribution of paths or path lengths bears some similarity to the stochastic shortest path problem [5, 10, 31], but the latter research has more to do with finding an optimal traversal policy. Results from queueing theory [7, 18, 19] suggest that many priority queues exhibit a power-law waiting time like eq. (15), but whether this has implications for graph traversal is not clear.



I, Image pixels
CONVOLUTION
$\{\mathcal{D}_q : q \in I\}$, Feature vectors
SEGMENTATION
S, Superpixels
PROPOSER
$T^{(i)}$, Trail hypothesis
POSTERIOR
$p(T^{(i)}|I)$, Posterior measure
i
$\operatorname{argmax}_{T^{(i)}} p(T^{(i)}|I)$, Trail inference

Figure 2. Overview of trail-finding steps. We use a bank of oriented filters to generate pixel features, which we use to segment the image into superpixels of contiguous texture. We generate independent trail proposals $T^{(i)}$ for $i = 1, 2, \ldots$ by searching for nearly-shortest paths in the superpixel graph, and keep the proposal with the highest posterior measure.

There are a few similarities between our approach to trail-finding and robotic path planning algorithms such as Rapidly-exploring Random Trees (RRTs) [24] and related approaches [4, 22]. In each case, a stochastically-built tree provides a route to a goal point. However, there are significant differences. With ordinary RRTs, as with other standard robotic path planning, candidate paths are cleanly partitioned into feasible and infeasible categories. That contrasts with trails, which are frequently occluded by tree cover. Also, in robotics, one usually seeks a feasible path that minimizes a cost function; whereas in the present work, we want a path that maximizes a probability distribution.

1.2. Data and Problem Definition

We used groundtruth derived from GPS tracks collected by Morris and Barnard [29] from the Great Divide Mountain Bike Route (GDMBR), which traverses the western continental United States. This route was partitioned into trail pieces that each fit into a 2 km square bounding box aligned north-south. Grayscale aerial imagery surrounding each trail piece, originating from the US Geological Survey, was downloaded from Microsoft Research Maps [8]. The result is 1526 trail pieces and images (one trail piece per image), at a resolution of 1 m/pixel. We pose the inference problem as follows. Given the image and the endpoints of the corresponding trail piece, how does the trail connect those endpoints? We evaluate our answer by computing how far the highest-posterior route strays from groundtruth.

We represent a trail piece T as a sequence of eight-connected pixel locations in the image. The number of locations is denoted $|T|$. Within T, a subsequence of exactly 50 distinct pixel locations, the *trail vertices* (two endpoints and 48 interior vertices), define the trail. All other pixel locations in T are determined by using Bresenham's line algorithm [11] between successive trail vertices. When necessary, we use dynamic programming [30] to reduce a sequence of route points to the 50-vertex size criterion. In the case of groundtruth, this introduces negligible distortion. The 50-vertex requirement is not essential to the model: the only part that depends upon it is the prior (§2.1). The likelihood will work with an arbitrary list of pixel locations, as long as the path has a well-defined tangent everywhere.

2. Image and trail models

We model the textures of the image using a Gaussian mixture (GMM). Our features are image brightness, plus the oriented energy response from twelve Gaussian filter kernels elongated with length/width ratio of 4, and rotated with a 15° increment. Each kernel has a sigma of 4 pixels in the narrow direction, which is comparable to the width of typical paths in our image data.

The 13-dimensional feature vectors are drawn from a selection of on-trail and off-trail pixels, and used to train a GMM of 100 modes using EM [14]. The intent is to learn a comprehensive set of texture elements—textons—found across all input images. Because the oriented Gaussian kernels roughly match the appearance of visible paths, the textons also tend to cluster trail pixels according to the direction of trail.

Next we label each image pixel with its most probable texton (GMM mode). During training, we then learn for each texton some additional characteristics: the frequencies with which it generates on-trail and off-trail pixels, and the axial direction (i.e., direction wrapped into the interval $[0, \pi)$ radians) associated with on-trail pixels. The former we learn by counting image pixels on-trail and off-trail in the training data. The latter we model as a von Mises distribution [27] (an angular distribution) using a maximum-likelihood (ML) fit of axial directions sampled from groundtruth trail pixels.

The results of this statistical learning are denoted as follows. For texton label k, let $\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k$ denote the mean and covariance of the feature vectors generated by the mode. Let $Q_1(k)$ denote the probability that a randomly-chosen on-trail pixel has label k. (In other words, it is the probability of the label, conditioned upon the pixel's on-trail status.) We compute this as a ratio of pixel-counts. Similarly, $Q_0(k)$ is the analogous probability for off-trail pixels of k. When k does generate an on-trail pixel, its axial direction is modeled by von Mises parameters μ_k, the expected direction, and κ_k, the concentration. We denote the PDF of an axial von Mises distribution evaluated at angle $\theta \in [0, \pi)$ by

$$f_{\text{AM}}(\theta; \mu, \kappa) = 2\mathcal{M}\left(2\theta; 2\mu, \kappa\right), \tag{1}$$

where $\mathcal{M}(\theta; \mu, \kappa)$ represents the PDF of an ordinary (radial) von Mises distribution. The inner factors of two in (1) cause the distribution to have period π radians, and the outer factor of 2 is required for normalization. When the concentration is zero, the distribution is uniform and, $f_{\text{AM}}(\theta; \mu, 0) = \pi^{-1}$. In Fig. 1c, color indicates ratio $Q_1(k)/Q_0(k)$, for each pixel's label k. The directional parameters are shown in Fig. 1d, where the hue and saturation vary with parameters μ_k and κ_k, respectively.

The texton labels induce a segmentation of the image into superpixels. A superpixel s is an 8-connected region of pixels sharing a common label, whose size $|s|$ we limit to at most 16384 pixels. Maximal connected regions of pixels that share a common label but exceed that size are partitioned into multiple superpixels by a 128×128 grid. However, such regions rarely overlap trails. Superpixels tend to be small (in our data, 99% have area under 500 pixels).

Segmentation is important to our model because it lets us assume a measure of conditional independence that balances the competing requirements for a model that is both realistic and computationally tractable. We have found an assumption of pixel-wise independence, conditioned solely on a trail, to be insufficiently realistic to improve upon earlier results. In our dataset, neighboring pixels frequently have highly-correlated feature vectors because the appearance of on- and off-trail terrain often changes slowly compared to the relatively narrow size of the feature kernels we have chosen. Yet the model will be intractable unless we can factor the image likelihood into independent components. Segmenting the image by a learned palette of textons lets us relax our independence assumptions, to respect the correlations in the imagery, yet still partition the superpixels into on- and off-trail classes.

2.1. Trail model

Since the trail representation approximates a polygonal path, it has a well-defined tangent direction at every pixel location strictly between two trail vertices: let trail pixel q be between successive vertices $v_i = [x_i, y_i]^\dagger$ and v_{i+1}. The direction at q is that of unit vector $\frac{v_{i+1}-v_i}{\|v_{i+1}-v_i\|}$. We can also define a direction at the trail vertices, as a composite of the directions of the neighboring path edges: at vertex v_i with predecessor and successor vertices v_{i-1} and v_{i+1}, we define the direction at v_i as that of vector $\frac{v_i-v_{i-1}}{\|v_i-v_{i-1}\|} + \frac{v_{i+1}-v_i}{\|v_{i+1}-v_i\|}$.

Trail curvature tends to change gradually, and we model that prior knowledge by $p_c(T)$, a product of von Mises distributions on the differences of successive vertex angles at interior vertices (Fig. 3). This prior lends preference to

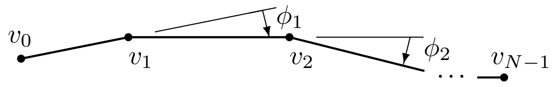


Figure 3. Trail notation used in prior. The successive differences of angles $\phi_1, \phi_2, \ldots, \phi_{N-2}$ discretely approximate the derivative of path curvature.

trails that have constant curvature. For a trail T defined by $N = 50$ vertices, with interior angles $\phi_1, \phi_2, \ldots, \phi_{N-2}$, the prior PDF is

$$p_c(T) = \prod_{i=1}^{N-3} \mathcal{M}\left(\frac{\phi_i - \phi_{i+1}}{2}; 0, \kappa_c\right). \tag{2}$$

Concentration κ_c is learned using a ML estimator on the corresponding angles in the training data.

In addition, we model prior knowledge about a trail's polygonal path length $\ell(T)$. By construction, each trail piece extends across a square bounding box $L_{\text{min}} = 1900$ meters on a side. The excess, $\Delta L = \ell(T) - L_{\text{min}}$, has a distribution that is roughly exponential, so we use prior $p_L(T) = \lambda \exp(-\lambda \Delta L)$, for which we estimate $\lambda = 1/\mathbb{E}[\Delta L]$ from training data. Our overall trail prior is thus

$$p(T) = p_L(T) p_c(T)^{1/(N-3)}, \tag{3}$$

where exponent $\frac{1}{N-3}$ compensates for the number of factors in $p_c(T)$.

2.2. Image likelihood

We develop the likelihood $p(I|T)$, where I is a grayscale image surrounding a trail hypothesis, T. We partition I into superpixels $S = \{s_1, s_2, \ldots, s_{|S|}\}$, and further partition S according to whether a superpixel intersects T. We denote the superpixels containing trail pixels by $S_1 = \{s \in S : s \cap T \neq \emptyset\}$. Assuming conditional independence of the superpixels, the likelihood is

$$p(I|T) = \prod_{s \in S} p(\mathcal{D}_s|T) = \prod_{s \in S \setminus S_1} l_0(s) \cdot \prod_{s \in S_1} l_1(s;T), \tag{4}$$

in which $\mathcal{D}_s$ represents the features of superpixel s, $l_0(s)$ represents the likelihood of $\mathcal{D}_s$ in a off-trail region, and $l_1(s;T)$ represents the likelihood of $\mathcal{D}_s$ when it intersects one or more pixels of T.

Observe that product $\prod_{s \in S} l_0(s)$ is independent of the trail hypothesis T. Hence we can cancel the contribution of the numerous off-trail superpixels:

$$p(I|T) \propto \frac{\prod_{s \in S \setminus S_1} l_0(s) \cdot \prod_{s \in S_1} l_1(s;T)}{\prod_{s \in S} l_0(s)} = \prod_{s \in S_1} \frac{l_1(s;T)}{l_0(s)}. \tag{5}$$

We now derive the likelihood functions in (5). In both l_0 and l_1, for each pixel we will account for three characteristics found there: the pixel's features as conditioned by its texton label, the pixel's label as conditioned by trail overlap, and the pixel's label as conditioned by trail directionality. In order to make the numerator and denominator of (5) share a consistent measure, we must include all three factors at each pixel.

We begin with likelihood $l_0(s)$ of the image data in off-trail superpixel s. Let q be any pixel in s, and let $c(q)$ and $c(s)$ respectively denote the texton label of q and the texton label common to all pixels of s. We assume independence of pixel data, given a common labeling:

$$\begin{aligned} l_0(s) &= p(\mathcal{D}_s \,|\, s \text{ occurs off-trail}) & (6)\\ &= \prod_{q \in s} \mathcal{N}\left(\mathcal{D}_q \,;\, \boldsymbol{\mu}_{c(s)}, \boldsymbol{\Sigma}_{c(s)}\right) \cdot Q_0(c(s)) \cdot \pi^{-1}. & (7)\end{aligned}$$

Here $\mathcal{D}_q$ represents the feature vector at q, and $\mathcal{N}(\mathcal{D}_q\,; \boldsymbol{\mu}_{c(s)}, \boldsymbol{\Sigma}_{c(s)})$ is the multivariate normal PDF. The factor π^{-1} accounts for the noninformative directionality of the texture in s.

When s intersects T at pixel q, we assess the likelihood of the directional appearance of $\mathcal{D}_q$ by assuming uniform prior distributions of texture direction and trail direction, in which case,

$$p(\text{direction of } \mathcal{D}_q \,|\, T) = p(\theta_T(q) \,|\, \text{direction of } c(s)). \tag{8}$$

Then the likelihood of image data within superpixel s is a product of the likelihoods of its on-trail and off-trail pixels:

$$\begin{aligned} l_1(s;T) = &\prod_{q_1 \in s \cap T} \Big(\mathcal{N}\left(\mathcal{D}_{q_1}\,;\, \boldsymbol{\mu}_{c(s)}, \boldsymbol{\Sigma}_{c(s)}\right) \\ &\quad \cdot Q_1(c(s)) \cdot f_{\text{AM}}\left(\theta_T(q_1); \mu_{c(s)}, \kappa_{c(s)}\right) \Big) \\ &\cdot \prod_{q_0 \in s \setminus T} \Big(\mathcal{N}\left(\mathcal{D}_{q_0}\,;\, \boldsymbol{\mu}_{c(s)}, \boldsymbol{\Sigma}_{c(s)}\right) \\ &\quad \cdot Q_0(c(s)) \cdot \pi^{-1} \Big) & (9) \\ = &\left(Q_1(c(s)) \cdot f_{\text{AM}}\left(\theta_T(s); \mu_{c(s)}, \kappa_{c(s)}\right)\right)^{|s \cap T|} \\ &\cdot \left(\pi^{-1} Q_0(c(s))\right)^{|s \setminus T|} \\ &\cdot \prod_{q \in s} \mathcal{N}\left(\mathcal{D}_q\,; \boldsymbol{\mu}_{c(s)}, \boldsymbol{\Sigma}_{c(s)}\right). & (10)\end{aligned}$$

We combine (7) and (10) into (5), canceling the off-trail pixel factors in the numerator and all the normal densities:

$$p(I|T) \propto \prod_{q \in T} \left(\frac{Q_1(c(q))}{Q_0(c(q))} \cdot \frac{f_{\text{AM}}\left(\theta_T(q); \mu_{c(q)}, \kappa_{c(q)}\right)}{\pi^{-1}} \right). \tag{11}$$

Intuitively the two kinds of ratios in this product can be interpreted as a logical conjunction: not only should the image textures along T "look like" trail (i.e., large ratios Q_1/Q_0), but also the directions learned for those textons should align with T (i.e., large ratios $f_{\text{AM}}(\theta_T)/\pi^{-1}$).

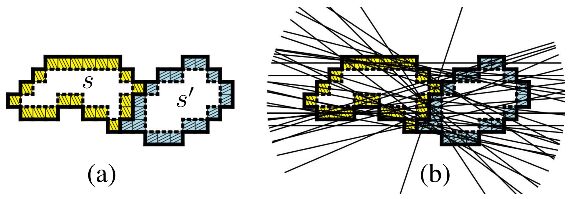


Figure 4. Modeling superpixel geometry. The geometric relationship of superpixels s and s' is modeled by an angular distribution. We model a random trail through both s and s' by a straight line from a border pixel of s (yellow blocks) to a border pixel of s' (blue blocks), shown in (a). We take a random sample of 50 such axes, shown in (b), to which we fit an axial von Mises.

3. Inference

Using the prior (3) and likelihood (11), we define posterior distribution

$$p(T|I) = \frac{1}{Z} p(T) p(I|T)^{1/|T|} \tag{12}$$

in which exponent $1/|T|$ makes the likelihood neutral with respect to trail size, and normalization factor Z is left unknown. We would like to use eq. (12) for inferring an unknown trail T, given image I and the endpoints of T. Exact maximization seems intractable, since in general the solution to the Longest Acyclic Path problem is NP-hard [16]. Instead, we explore the space of trails that bridge between two known endpoints by searching for *short* paths in a carefully weighted graph of superpixels (an approximately dual problem). We use a variation of Dijkstra's algorithm to find the shortest and nearly-shortest paths—trail proposals—in this weighted graph of superpixels. Then we use eq. (12) to select the most probable proposal.

3.1. Edge weights in superpixel graph

Our proposer is inspired by Morris and Barnard [29], who also computed shortest paths for trail inference. We use their same basic idea: an edge that is highly likely to be on a trail should get low weight. However, we face two technical challenges when translating the likelihood ratio of (11) to edge weights: we require an inverse relationship that remains finite, and we lack the directional factor $f_{\text{AM}}(\theta_T)$ (i.e., while creating a trail proposal, we cannot yet know its tangent).

In place of factor $f_{\text{AM}}(\theta_T)$, we compute a statistical approximation based on superpixel geometry. Any graph path comprising the edge from s to s' corresponds to a trail proposal whose geometry overlaps both superpixels, so we model a random trail proposal connecting s and s' (Fig. 4). We compute the ML parameters $(\mu_{s,s'}, \kappa_{s,s'})$ of an axial von Mises distribution of a random straight track traversing both s and s'. Then we compare this geometric model with the directional model learned for texton $c(s')$. If the two distributions are similar, it is more likely that a path entering s would extend into s'. The Bhattacharyya kernel [20] k_B lets us compare the distributions in closed form, and so we treat its numerical value $b(s, s')$ as a proxy for ratio $f_{\text{AM}}(\theta_T)/\pi^{-1}$ appearing in (11):

$$b(s, s') = k_B([\mu_{c(s')}, \kappa_{c(s')}]^{\mathsf{t}}, [\mu_{s,s'}, \kappa_{s,s'}]^{\mathsf{t}}). \tag{13}$$

Edge weight w must have an inverse relationship with local likelihood ratio $\rho = \frac{Q_1}{Q_0} b$ that is well-behaved (i.e., w can never be too large), because good proposals sometimes traverse edges through regions of unlikely-appearing texture, where ρ is very small. Thus $w \propto \rho^{-1}$ would work poorly. To set an upper bound on weights, we instead use a relationship like $w \sim (1+\rho)^{-1}$. Weight w also needs a direct relationship with superpixel size, otherwise the proposer would have a bias in favor of large superpixels. Thus we set $w \propto |s'|$, area of s' in pixels. (It is better to scale w by $|s'|$ than $|s'|^{1/2}$, because superpixels tend to be narrow.)

In order to balance the sizes of the weights in image regions likely and unlikely to be trail, we include parameters α and γ, which are trained by grid search so as to minimize Hausdorff distance between groundtruth and the shortest path in the superpixel graph. Thus our choice for edge weight between superpixels s and s' is

$$w(s, s') = \frac{|s'|}{1 + \gamma \left(\frac{Q_1(c(s'))}{Q_0(c(s'))} \cdot b(s, s') \right)^{\alpha}} . \tag{14}$$

3.2. Sampling short paths

We have developed a variation on Dijkstra's algorithm to sample paths in this weighted graph that are short but not necessarily the shortest. As typically presented (e.g., in [12]), each iteration of Dijkstra's algorithm performs an EXTRACT-MIN operation on a priority queue of vertices, prioritized by distance. Our idea is to alter the priority queue to support an EXTRACT-NEAR-MIN operation that, at each iteration, extracts a vertex (superpixel) selected randomly, with a preference for vertices of smaller distance. For a vertex s with distance $d(s)$ in the queue, its probability of being the next vertex removed from the queue U is given by a power law,

$$\Pr(s \text{ will be drawn next}) = \frac{d(s)^{-\beta}}{\sum_{u \in U} d(u)^{-\beta}} , \tag{15}$$

where $\beta = 1.5$ was chosen empirically. The value of β affects the dispersion of the sampled paths.

We pay no time-complexity penalty for this approach. The stochastic priority queue is implemented with a red-black tree that stores non-normalized probability mass $d(s)^{-\beta}$ with entry s, and maintains at each tree node a sum of all subtree nodes' probability masses. Thus we can perform an EXTRACT-NEAR-MIN operation in time $O(\log |U|)$, where $|U|$ is the number of vertices (superpixels) in the priority queue. Since the superpixel graph is pla-

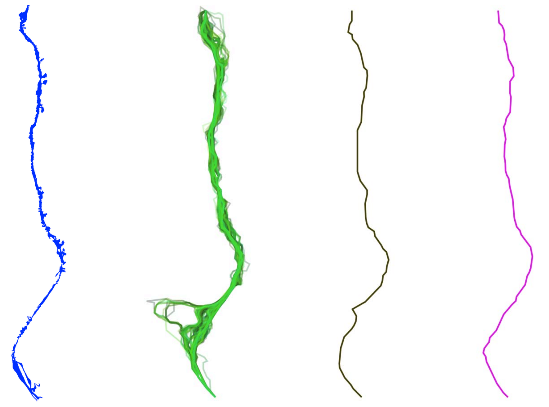


Figure 5. Examples of intermediate and final results of inference process: (a) the groundtruth pixel footprint of superpixels of a typical trail piece, i.e., all pixels of the superpixels touching groundtruth; (b) 200 short paths generated by the quasi-Dijkstra method; (c) shortest path found by Dijkstra's algorithm; (d) short path approximately maximizing eq. (12). (Compare with Fig. 1.)

nar, sampling a short path with this implementation uses time $O(|S| \log |S|)$.

To generate a trail proposal, we use either this quasi-Dijkstra algorithm, or the unmodified Dijkstra's algorithm, to find a short path P in the superpixel graph bridging between endpoints. P is a simple path of superpixels, $P = (s_{i_1}, s_{i_2}, \ldots, s_{i_{|P|}})$, where $i_1, i_2, \ldots, i_{|P|}$ are the indices of the chosen superpixels.

Because our prior model requires a polygonal path with a fixed number of vertices, we take some additional steps to reduce the superpixel path into a polygonal path of pixels. Qualitatively, these steps have little effect on the path. First we compute the path's pixel footprint $F_P = \bigcup_{j=1}^{|P|} s_{i_j}$. Fig. 5a shows an example footprint. Next we reduce F_P to a polygonal path P' by computing an ordinary shortest path in the pixel graph, using 8-way adjacency and Euclidean distance. Alternatively we could have used a medial-axis algorithm, but F_P is almost always thin enough that the difference would be negligible. Finally we reduce the number of vertices to 50, using a straightforward dynamic programming algorithm [30].

4. Results and Conclusions

We use the Hausdorff distance metric between groundtruth and inferred path for evaluation; a perfectly inferred path will have a distance of zero. Given a fixed error threshold, Fig. 6 shows the success rate of three methods of generating and assessing trail proposals. Alternatively, this plot can be viewed as the CDF of the Hausdorff error for a randomly selected trail. The top curve, labeled "QD-MAP," denotes the success rate when we generate 200 trail proposals by the quasi-Dijkstra procedure described above, and keep the proposal with

Table 1. Comparison of results for GDMBR trail inference. Error metric is Hausdorff distance between groundtruth and inference. *NB-Sampler* is the naive-Bayes classifier and sampler of [29]. Other labels are as described in §4.

	Median err. [m]	Mean err. [m]	Success rate, err. < 50 m
NB-Sampler	-	118 ± 8	60%
Shortest path	40.5 ± 3.5	137 ± 13	$54\% \pm 2\%$
QD-ML	28.9 ± 1.8	105 ± 10	$67\% \pm 2\%$
QD-MAP	27.0 ± 1.4	96 ± 12	$68\% \pm 2\%$

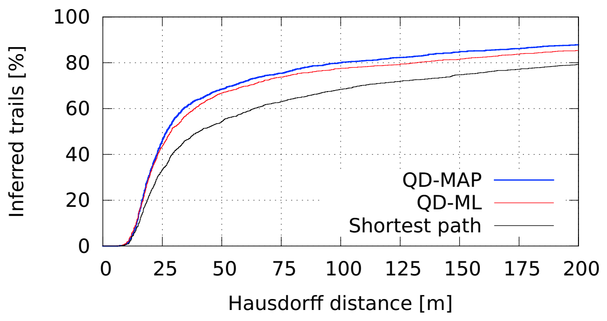


Figure 6. Cumulative distribution of Hausdorff distance between groundtruth and best route inference, i.e., success rate vs. error threshold. QD-MAP shows the success rate of quasi-Dijkstra proposals that maximize eq. (12). QD-ML shows the success rate of quasi-Dijkstra proposals that maximize image likelihood. We also show the success rate of the shortest path in the superpixel graph.

maximum posterior probability as measured by (12). Empirical testing suggests 200 proposals is enough to converge to a good inference.

For comparison, we present two variations on this method. The curve labeled "QD-ML" shows the success rate of the quasi-Dijkstra path proposal that maximizes image likelihood (eq. (11) with $p(T)$ replaced by unity). The lowest curve shows the success rate of the shortest path in the superpixel graph. The latter's poor performance clearly shows that the primal-dual relationship suggested for the superpixel graph is only approximate: the best inferences are often graph paths that are short but not shortest.

The gap between MAP and ML curves shows the benefit of the prior model (§2.1). The value of the prior is also evident when inference fails. If we fix the error threshold at 50 meters, the success rate for QD-MAP is 68%, but among the failure cases, in 385 out of 481 images (80%) the failing inferred trails have a lower posterior measure than that of groundtruth. One could hope that a more sophisticated proposer might recover some of these trails. In contrast, among the QD-ML failure cases, 322 out of 508 images (63%) have an image likelihood given the failed inference that exceeds the image likelihood when given groundtruth. Hence a maximum-likelihood approach to inference will necessarily fail in all those cases, even if the proposer were an oracle. This sort of confusion points to the limitations of

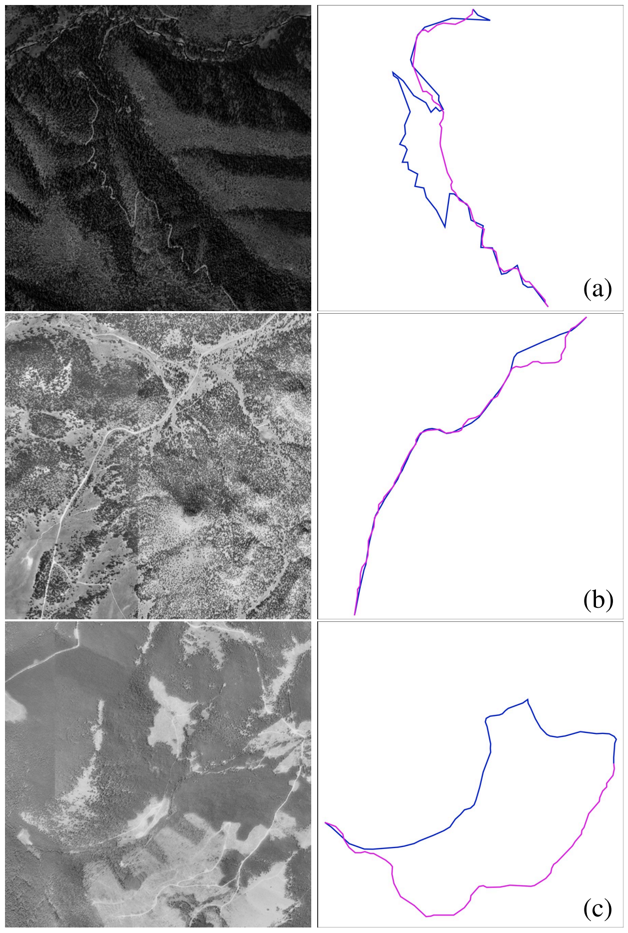


Figure 7. Example failures. Groundtruth is shown in blue, inference is purple. Inference (a) is too steep to be plausible, indicating that elevation data would improve results. Example (b) suggests that inference could be improved by narrowing the search space to retain partially-correct results. Inference (c) shows that when image evidence is scant or contradictory, the problem is ambiguous.

our likelihood model along the lines analyzed by Yuille and Coughlan [13, 32].

We compare our results to those of Morris and Barnard in Table 1. They proposed a 50 meter Hausdorff distance as a standard threshold of success, and presented three related methods of trail inference. Their most successful method was a sampling approach to minimize an energy function, which achieved errors of 50 m or less for 60% of their test trails, out of 500 trail images. We were not able to use the same 500 trail images, but the trail pieces we used were drawn from the same route (the GDMBR) and our images come from the same USGS corpus of imagery. Our method outperforms the sampling approach with a higher rate of successful inference and lower mean error. Intervals on our results represent 95% confidence, generated using 14-way cross validation.

We show some examples of failed inference in Fig. 7. Because our approach does not use elevation data, trail switchbacks are often missed, and our prior model cannot reject proposals such as that of Fig. 7a, a route that is too steep to be a plausible trail. Our proposer generates independent routes, and thus lacks the ability to improve faulty sections of a nearly-correct route, like that shown in Fig. 7b. Finally, the trail-finding problem itself is ambiguous when image evidence suggests more than one route between the endpoints, as in Fig. 7c. Sample successes are shown in Fig. 8. We note in conclusion that one of the advantages of a statistical approach is that the models are amenable to refinement. As we continue to improve our model and inference procedures, we look forward to better results.

Acknowledgment Resources from the iPlant Collaborative contributed to these research results. The iPlant Collaborative is funded by a grant from the National Science Foundation (#DBI-0735191). URL: www.iplantcollaborative.org. This research was also supported by a GAANN fellowship.

References

[1] M. A. Abdul-Karim, B. Roysam, N. M. Dowell-Mesfin, A. Jeromin, M. Yuksel, and S. Kalyanaraman. Automatic selection of parameters for vessel/neurite segmentation algorithms. *IEEE Trans. on Image Proc.*, 14(9):1338–1350, Sep. 2005.

[2] K. A. Al-Kofahi, A. Can, S. Lasek, D. H. Szarowski, N. Dowell-Mesfin, W. Shain, J. N. Turner, and B. Roysam. Median-based robust algorithms for tracing neurons from noisy confocal microscope images. *IEEE Trans. on Info. Tech. in Biomedicine*, 7(4):302–317, Dec. 2003.

[3] K. A. Al-Kofahi, S. Lasek, D. H. Szarowski, C. J. Pace, G. Nagy, J. N. Turner, and B. Roysam. Rapid automated three-dimensional tracing of neurons from confocal image stacks. *IEEE Trans. on Info. Tech. in Biomedicine*, 6(2):171–187, Jun. 2002.

[4] N. M. Amato and Y. Wu. A randomized roadmap method for path and manipulation planning. In *Intl. Conf. on Robotics and Automation*, Apr. 1996.

[5] G. Andreatta and L. Romeo. Stochastic shortest paths with recourse. *Networks*, 18:193–204, 1988.

[6] M.-F. Auclair-Fortier, D. Ziou, C. Armenakis, and S. Wang. Survey of work on road extraction in aerial and satellite images. Technical Report 247, Université de Sherbrooke, 2000.

[7] A.-L. Barabási. The origin of bursts and heavy tails in human dynamics. *Nature*, 435:207–211, May 2005.

[8] T. Barclay, J. Gray, and D. Slutz. Microsoft terraserver: A spatial data warehouse. Technical Report MS-TR-99-29, Microsoft Research, Feb. 2000.

[9] F. Benmansour and L. D. Cohen. Tubular structure segmentation based on minimal path method and aniosotropic enhancement. *Intl. J. of Comp. Vision*, 92(2):192–210, 2011.

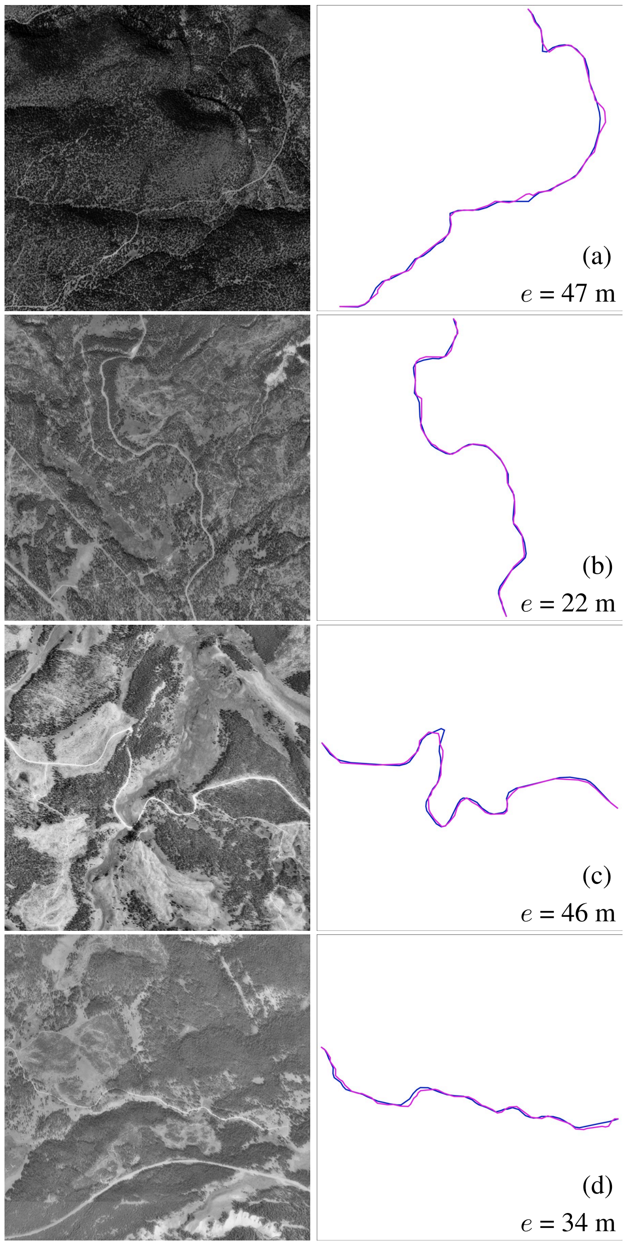


Figure 8. Example successes. Groundtruth is shown in blue, inference is purple. Hausdorff error e, in meters, is shown for each.

[10] D. P. Bertsekas and J. N. Tsitsiklis. An analysis of stochastic shortest path problems. *Mathematics of Operations Research*, 16(3):580–595, Aug. 1991.

[11] J. E. Bresenham. Algorithm for computer control of a digital plotter. *IBM Systems Journal*, 4(1):25–30, 1965.

[12] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. MIT Press, 2nd edition, 2001.

[13] J. M. Coughlan and A. L. Yuille. Bayesian a* tree search with expected o(n) node expansions: Applications to road tracking. *Neural Computation*, 14(8):1929–1958, 2002.

[14] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *J. Royal Stat. Soc.*, 39:1–38, 1977.

[15] M. A. Fischler, J. M. Tenenbaum, and H. C. Wolf. Detection of roads and linear structures in low-resolution aerial imagery using a multisource knowledge integration technique. *Computer Graphics and Image Processing*, 15:201–223, 1981.

[16] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Co., 1979.

[17] D. Geman and B. Jedynak. An active testing model for tracking roads in satellite images. *IEEE Trans. on Pattern Analysis and Mach. Intel.*, 8(1):1–14, Jan. 1996.

[18] G. Grinstein and R. Linsker. Biased diffusion and universality in model queues. *Physical Review Letters*, 97(130201), 2006.

[19] G. Grinstein and R. Linsker. Power-law and exponential tails in a stochastic priority-based model queue. *Physical Review E*, 77(012101), 2008.

[20] T. Jebara, R. Kondor, and A. Howard. Probability product kernels. *J. Mach. Learning Research*, 5:819–844, 2004.

[21] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *Intl. J. of Comp. Vision*, 1(4):321–331, 1988.

[22] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, 12(4):566–580, Aug. 1996.

[23] C. Kirbas and F. Quek. A review of vessel extraction techniques and algorithms. *ACM Computing Surveys*, 36(2):81–121, Jun. 2004.

[24] S. M. LaValle and J. J. Kuffner. Rapidly-exploring random trees: Progress and prospects. In *Proceedings Workshop on the Algorithmic Foundations of Robotics*, 2000.

[25] D. Lesage, E. D. Angelini, I. Bloch, and G. Funka-Lea. A review of 3d vessel lumen segmentation techniques: Models, features and extraction schemes. *Medical Image Analysis*, 13:819–845, 2009.

[26] J. Malik, S. Belongie, T. Leung, and J. Shi. Contour and texture analysis for image segmentation. *Intl. J. of Comp. Vision*, 43(1):7–27, 2001.

[27] K. V. Mardia and P. E. Jupp. *Directional Statistics*. Wiley, 2000.

[28] J. B. Mena. State of the art on automatic road extraction for gis update: a novel classification. *Pattern Recognition Letters*, 24(16):3037–3058, 2003.

[29] S. Morris and K. Barnard. Finding trails. In *Proc. IEEE Conf. Comput. Vision and Pattern Recogn.*, 2008.

[30] J.-C. Perez and E. Vidal. Optimum polygonal approximation of digitized curves. *Pattern Recognition Letters*, 15:743–750, 1994.

[31] G. H. Polychronopolous and J. N. Tsitsiklis. Stochastic shortest path problems with recourse. *Networks*, 27:133–143, 1996.

[32] A. L. Yuille and J. M. Coughlan. Fundamental limits of bayesian inference: Order parameters and phase transitions for road tracking. *IEEE Trans. on Pattern Analysis and Mach. Intel.*, 22(2):160–173, Feb. 2000.