A Generative Model of Terrain for Autonomous Navigation in Vegetation

Carl Wellington
Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213
Email: cwellin@ri.cmu.edu

Aaron Courville
Département d'Informatique et de recherche opérationnelle
Université de Montréal
Montréal, Quebec H3C 3J7
Email: aaron.courville@gmail.com

Anthony (Tony) Stentz
Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213
Email: axs@ri.cmu.edu

***Abstract*— Current approaches to off-road autonomous navigation are often limited by their ability to build a terrain model from sensor data. Available sensors make very indirect measurements of quantities of interest such as the supporting ground height and the location of obstacles, especially in domains where vegetation may hide the ground surface or partially obscure obstacles. We introduce a generative, probabilistic terrain model that exploits natural structure found in off-road environments to constrain the problem and use ambiguous sensor data more effectively. The model includes two Markov random fields that encode the assumptions that ground heights smoothly vary and terrain classes tend to cluster. The model also includes a latent variable that encodes the assumption that vegetation of a single type has a similar height. The model parameters can be trained by simply driving through representative terrain. Results from a number of challenging test scenarios in an agricultural domain reveal that exploiting the 3D structure inherent in outdoor domains significantly improves ground estimates and obstacle detection accuracy, and allows the system to infer the supporting ground surface even when it is hidden under dense vegetation.**

I. Introduction

Autonomous navigation in outdoor environments, such as those encountered in agriculture, mining, and exploration applications, is often performed using a model predictive control framework [1]. In this framework, a terrain model is used in combination with a model of the vehicle to find a dynamic trajectory that avoids obstacles while protecting against roll-over, body collisions, high-centering, and other safety conditions. While faithful models of vehicle dynamics are often available, acquiring an accurate terrain model that includes a description of the supporting ground surface and any obstacles remains a considerable challenge.

Outdoor, off-road environments do not possess structured landmark features such as road markers, straight walls or a flat ground plane. This "unstructured" nature of outdoor environments complicates terrain model estimation and has often been cited as one of the reasons why the development of terrain models of these environments is considered to be challenging [1] [2] [3]. However, such environments do possess a great deal of structure that humans frequently exploit in the performance of tasks we wish to automate. For example, consider a vehicle navigating through a field of vegetation. We can use the knowledge that ground is generally smooth and that vegetation has approximately constant height (with respect to the ground) to permit stable and robust navigation, even through areas where the ground is not directly observed. The challenge lies in expressing this type of structure in a way that can be made useful in autonomous navigation tasks.

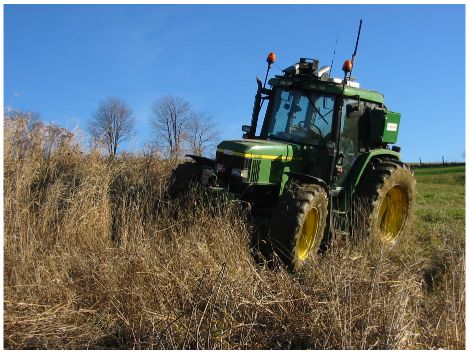


Fig. 1. Tractor test platform

In this paper, we describe a generative, probabilistic approach to modeling terrain. The model exploits the 3D spatial structure inherent in outdoor domains and an array of noisy but abundant sensor data to simultaneously estimate ground height, vegetation height and classify obstacles. Joint inference of ground height, class height and class identity over the whole model results in more accurate estimation of each quantity. For example, inferring the vegetation height permits an improved estimate of the height of the underlying ground. Similarly, knowing the ground height helps disambiguate solid obstacles from the ground surface.

To help constrain the problem of joint obstacle detection and supporting ground surface estimation, our terrain model incorporates a number of spatial assumptions. Two of these are encoded by distinct but interacting Markov random field (MRF) models. Markov random fields specify the relationship between variables through local dependencies. In our terrain model, one MRF models ground height and enforces our assumption that ground height is smoothly varying. The second MRF encodes our assumption that regions associated with a particular class tend to cluster (for example, patches of vegetation of a single type tend to be found together). In addition to the MRF layers, a simple latent variable model is used to express an assumption that vegetation of the same type has a similar height above the ground. These three components interact through a hidden semi-Markov model (HSMM). The hidden semi-Markov model generalizes the hidden Markov model [4] by including an explicit probability distribution over the duration of each state. In our model, the HSMM enforces vertical structural assumptions such as the understanding that vegetation grows on top of ground.

The spatial constraints embodied in the terrain model are combined with information from multiple sensors on the vehicle. The sensor models are automatically learned from training data collected by driving over representative terrain. Training these sensor models to directly detect obstacles is particularly challenging because of the wide variety of objects that might be considered obstacles. People, buildings, equipment, fences and even holes are all obstacles we wish to avoid while navigating in off-road environments. There may be little in common between these things that we wish to classify as obstacles, and it is impractical to train classifiers to individually detect all possible types of obstacles. Instead, our approach is to treat obstacles as having uncertain attributes relative to the more accurate models of the known classes (such as vegetation or ground). As a result, the model considers surprising and otherwise unexplainable measurements as those most likely to be associated with an obstacle. This is a common approach in anomaly detection, and has been used for detecting obstacles in an agricultural domain [5].

Our use of MRFs to encode horizontal spatial correlations precludes an analytical solution to the inference problem. However, by exploiting an organization of our data into vertical HSMM columns, our approach allows us to model 3D structure in a reasonably efficient inference scheme. Gibbs sampling over the MRF structures lets us perform exact inference in the HSMM models using an efficient dynamic programming algorithm. This substantially reduces computation time over a full 3D MRF model, and although our approach is computationally demanding, the system can run in real-time.

II. Related Work

Early work in terrain perception considered domains where the supporting ground surface is directly observable and objects above the ground are generally obstacles [1] [2]. In these domains, the range measurements can be used directly to estimate the ground surface and find positive obstacles, which enables successful navigation. As shown in Figure 2, the presence of vegetation makes the problem much more difficult because the range points from forward looking sensors such as a laser range-finder or stereo cameras do not generally give the supporting ground surface. Classification of vegetation [6] (also common in the remote sensing community – see references in [7]) is not sufficient for this task because a grassy area on a steep slope may be dangerous to drive on whereas the same grass on a flat area could be easily traversable. For successful navigation in many off-road environments, a system must be able to recover the supporting ground height and the true obstacles, even in the presence of vegetation.

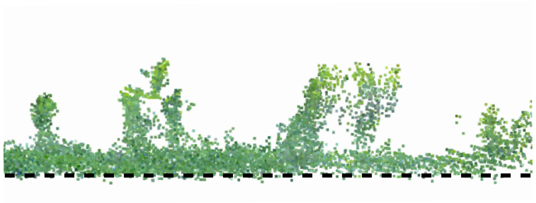


Fig. 2. Side view of penetrable vegetation data, with approximate ground height. There are range points on the ground and the vegetation.

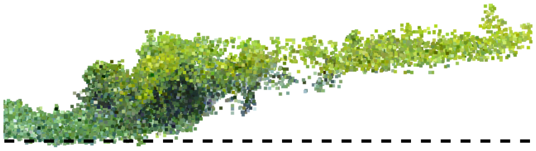


Fig. 3. Side view of data showing transition from low grass on the left to tall dense non-penetrable vegetation on the right, with approximate ground height. In the tall dense vegetation, there are only range points on the top of the vegetation and the ground remains hidden.

A number of researchers have investigated methods that use range data to discriminate sparse vegetation (as shown in Figure 2) from solid substances such as the ground or obstacles. These techniques exploit the fact that range measurements often penetrate sparse vegetation, but do not penetrate solid obstacles. The properties used for discrimination fall into two categories: shape and density.

Shape-based methods begin with a 3D cloud of range points and look at local features of the data points to discriminate between the random spread of points in sparse vegetation and the organized structure of points on solid objects. Researchers have modeled the statistics of laser penetration in grass to find solid objects [8], and they have compared measurements across time and space to filter out areas where the penetration is continually changing [9]. A comparison between techniques that look for the range shadow of solid obstacles and techniques based on local point statistics is given in [7]. The strategy of computing local statistics about the spread of points was expanded in [10] to discriminate between sparse vegetation, solid surfaces, linear structures such as branches, and even concertina wire [11].

Density-based methods attempt to use range measurements to explicitly measure the density of objects in the environment. This has been done by dividing the world into small volumes of space and then maintaining density scores by keeping track of lidar hits and pass-throughs [3], [12] or lidar and radar measurements [13].

The above methods have shown promising results in sparse vegetation, but they do not address the problem of estimating the ground surface in dense vegetation where the ground is completely hidden as in Figure 3. We have previously used online learning methods to automatically learn the mapping from features of the sensor data to ground height predictions [14]. This system was able to predict the location of the ground height under dense vegetation while still finding solid obstacles. However, the system explicitly learns a vegetation height so it is not general across vegetation of the same type with unknown height, and the ground height estimates in dense vegetation are very noisy and have large errors. One reason for this is that this method makes the strong assumption of independence between terrain patches and makes predictions locally without incorporating spatial context. This can make it difficult to disambiguate data from tall vegetation and data from short vegetation, resulting in poor estimates of the hidden ground height.

In this article we relax the independence assumption through the inclusion of spatial correlations. This added structure allows the model to infer the height of vegetation from the local sensor data instead of training on it explicitly, and enables the model to produce a smooth ground height estimate that is well suited to autonomous navigation.

III. Vehicle Platform and Data Representation

Our project team has automated a John Deere 6410 tractor (see figure 1) and equipped it with many sensors for localization and perception [14]. The vehicle has a high-resolution stereo pair of digital cameras, an infrared camera, and two SICK laser range-finders (lidar) mounted on custom actively-controlled scanning mounts. The first scanning lidar is mounted on the roof to get range data over a large area in front of the vehicle, and the second scanning lidar is mounted on the bumper to get high density measurements of nearby terrain and better penetrate upcoming vegetation. The cameras and scanned lidars are precisely calibrated and tightly synchronized with an accurate global vehicle pose estimate, allowing the system to accumulate data into a high quality global map, as shown in Figure 4.

The basic representational structure of our terrain model is the *voxel*: a $15\mathrm{cm}^3$ box-shaped region of 3 dimensional space. We represent the vehicle's local spatial environment as a voxel lattice of size I x J x K, where the ijkth voxel is in the ijth position of a horizontal 2D grid and the kth position above an arbitrary subterranean origin.

Accurate global vehicle pose allows us to assign lidar points corresponding to the same region of space to the same voxel. Exploiting the precise synchronization of the sensors, we project lidar points into the most recent color and infrared images, so that each lidar point results in a vector of appearance measurements for that voxel, including laser remission (reflectance), infrared temperature, and color.[1]

The voxel representation also allows us to maintain a density estimate throughout space by comparing how many lidar rays pass through each voxel (pass-throughs) with the number of lidar rays that hit something in that voxel (hits). Density information is valuable when trying to separate sparse vegetation that contains a mixture of hits and pass-throughs from solid objects that contain a majority of hits and only a few pass-throughs due to sensor noise [3] [14].

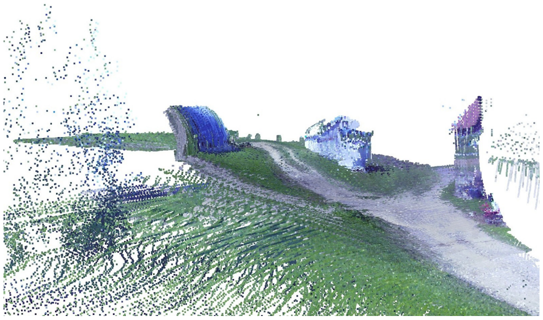


Fig. 4. Colorized lidar points of farm test site buildings as the vehicle drove down a path to the field

Although our data representation is based on the voxel, vehicle navigation is generally performed on a 2D surface, so our ground height estimates and classification results are made in terms of voxel columns. In our model, the ijth voxel column class is described with a multinomial distributed random variable C_{ij} taking on values related to the possible contents of the column, $C_{ij} = c$ with e.g. $c \in \{ground, vegetation, obstacle\}$.

Associated with the kth voxel in the ijth voxel column is the voxel state X_{ij}^k, a multinomial distributed random variable that describes the nature of the material inside the voxel, $X_{ij}^k \in \{ground, c, \textit{free-space}\}$, where c is the class of the ijth voxel column.[2] The ijkth voxel is also associated with an observation vector formed by the concatenation of multiple sensor measurement vectors from an array of sensor modalities, $\mathbf{Y}_{ij}^k = [Y_{den}, Y_{rem}, Y_{ir}, Y_{col}]$. The N lidar measurements are encoded as a binary vector Y_{den} consisting of M ones representing hits and $N-M$ zeros representing pass-throughs. Associated with each of the M hit measurements are corresponding measurements of laser remission, infrared temperatures and color (i.e. $Y_{den} = [Y_{den}^1, \ldots, Y_{den}^N]$, $Y_{rem} = [Y_{rem}^1, \ldots, Y_{rem}^M]$).

IV. Terrain Model

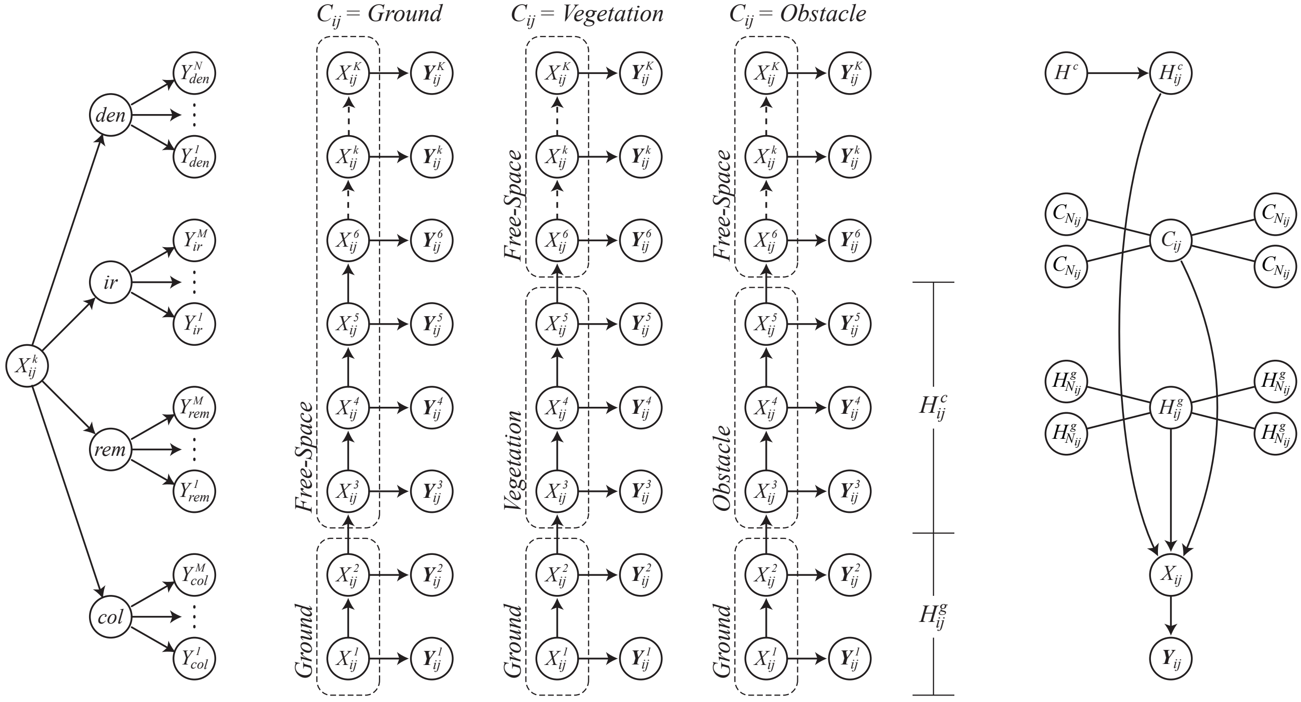
We use a probabilistic generative model to encode the desired spatial correlation assumptions described in the introduction. In this section we describe the model at three different levels:

- Voxel observation models
- Vertical column structure
- Horizontal neighborhood structure

These levels build on each other from the inside out, going from an individual voxel to a voxel column to the entire grid of voxel columns. Figure 5 gives a graphical representation of each of these three levels.

[1] The lidar scans come in at a much higher rate than the image data so multiple scans are projected into the same image. However, the high pixel density of the images means that we collect approximately 100 pixels for every lidar point. This coupled with the continual movement of the scanning lidars makes it unlikely that a single pixel is used more than once, so we treat each color and infrared tagged lidar point as an independent measurement.

[2] In our implementation, the possibility of a voxel column simultaneously containing obstacle and vegetation is excluded, though its inclusion is a trivial extension of the model we present.



(a) Voxel model (b) HSMM models (c) MRF model

Fig. 5. A graphical description of the model showing (a) the voxel, (b) the voxel column, and (c) the connections between voxel columns. For each voxel column ij, the model contains voxel states X^k_{ij}, observations Y^k_{ij}, and a class C_{ij}, class height H^c_{ij}, and ground height H^g_{ij} that interact with neighbors N_{ij} and the common class height H^c

Before describing each level of the model in detail, we offer a perspective on the entire model by exploring the process of generating measurement data. The model contains two Markov random fields. One MRF codifies our assumption that the ground height varies smoothly with distance. Thus we expect the ground height of voxel column ij to be similar to the ground heights of the neighboring voxel columns. From a generative perspective, conditioned on the values of the neighboring ground heights we sample the ground height of voxel column ij. Similarly, we assume that neighboring voxel columns will tend to be members of the same class. Once again, this assumption is represented as an MRF and by conditioning on the class memberships of the neighboring columns we can directly sample a class membership for voxel column ij. To recover the ground heights and class memberships of all the voxel columns this process is repeated for all ij.

In addition to the ground height and class membership, a vegetation class height (relative to ground height) must also be determined for each voxel column with a *vegetation* class membership. We model the distribution of heights of a particular vegetation class as a Gaussian centered about some global mean height. For each vegetation class, a global class mean height is sampled, and then conditioned on this mean height, the individual class heights of the corresponding voxel columns are sampled.

Once the ground height, class membership and vegetation class heights are determined, their values are used to constrain the properties of each voxel column which is described using a hidden semi-Markov model. Each class has an associated HSMM model that determines the state values for each voxel in the vertical chain. For instance, below the sampled ground height, all states assume the value *ground*. For the vegetation classes, the states between the ground height and the vegetation height are *vegetation* and above that the voxels are *free-space*.

Conditioned on the values of the voxel states, the material properties, such as color, laser remission, infrared temperature and density are all sampled independently from the appropriate state specific distribution (e.g. *ground* or *vegetation*). Finally, conditioned on the values of each material property, measurements are sampled independently from the various measurement models.

While it is instructive to understand the model from a generative perspective, in general we are more interested in the opposite problem: how can one use the model to *infer* ground heights, class memberships and vegetation class heights from real sensor data. This will be the subject of section V. We now describe the three levels of the model in more detail.

A. Voxel Observation Models

We assume that voxels form the smallest indistinguishable element of space, occupied completely by one (and only one) voxel state. Each voxel state maintains a distribution over material properties including density, remission, infrared temperature, and color that describe the characteristics of that state, but the material inside a single voxel is assumed to be uniform. For example, the vegetation state may include a range of colors, and therefore different voxels in vegetation may have different colors, but we assume that the color of the vegetation within each voxel is uniform.

There are a variable number of observations of the material properties in each voxel, and care must be taken in how we combine these measurements. Observations are commonly modeled as being independent given the state, so that accumulating measurements leads to an arbitrary certainty in the state value. However, such an approach fails to take account of the inherent uncertainty in the distribution of material properties for a given voxel state value.

This phenomena is best illustrated with an example taken from one of the experimental scenarios we use to validate the model (see section IX-A). This example includes a white shed next to a white road. If the system accumulates color measurements for a particular voxel, then it can become arbitrarily certain of the color of that voxel, but knowing the true color material property of the voxel does not lead to certainty regarding the voxel's state value. Both the ground and obstacle states can exhibit a range of colors, and gathering evidence regarding the color of the voxel does not resolve the uncertainty in material properties possessed by the states.

Statistically, this problem reduces to the fact that in our model, multiple sensor measurements are not independent conditional solely on the value of the state. For example, a voxel may contain a vegetation state that includes both green and yellow material properties. Observing green for that voxel would then increase the probability of subsequent green observations, even if one knows the true state of the voxel. However, the measurements *are* independent given the true value of the measured material property. Continuing with the example, knowing that the true color material property of a voxel is green makes the color measurements of that voxel independent. By explicitly introducing the true unobserved material properties in to the observation model, we can properly account for the non-independence of sensor measurements with respect to the state values.

The graphical model in Figure 5(a) illustrates the conditional independencies between the voxel state X_{ij}^k, the material property random variables den, rem, ir and col, and the measurements. Conditional on X_{ij}^k, the material properties are independent, and conditional on the material properties, the measurements are independent. The voxel material properties are not directly observed, and we are not concerned with their values beyond what they reveal about the state. Thus the material properties constitute nuisance variables that we remove from the observation models through marginalization, as described below.

1) Appearance: The distributions over the voxel appearance properties, including infrared temperature, laser remission, and color are all inherently multi-modal and thus not well described by a simple parametric distribution. For example, remission values in vegetation are either high because of the strong reflectivity of chlorophyll, or very low due to small cross-sectional area. We resort to a Gaussian mixture model (GMM) to describe the distribution of the material properties within a state.

We develop the marginal distribution for the remission values, but the infrared and color data are determined analogously. The true material property rem for state value x is modeled as a GMM with R mixture components. Each individual mixture component i is parameterized with mean μ_i, variance σ_i^2, and mixing coefficient $P(i)$. Note that these parameters are unique to the rem material property. Color and infrared data are parameterized independently.

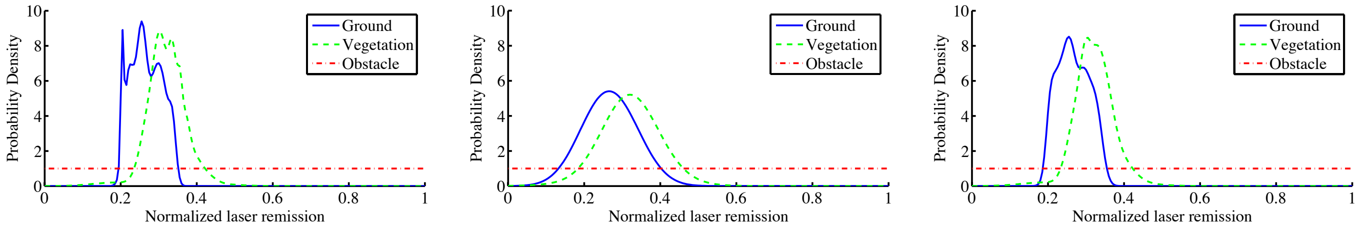
$$p(rem|X_{ij}^k\!=\!x) = \sum_{i=1}^{R} P(i)\frac{1}{\sqrt{2\pi\sigma_i^2}}\exp\left(-\frac{(rem-\mu_i)^2}{2\sigma_i^2}\right) \tag{1}$$

Conditional on the true material property rem, the measurements y_{rem}^m are assumed to be normally distributed, $y_{rem}^m \sim \mathcal{N}(rem, \sigma_y^2)$. We integrate out the nuisance variable rem to get the marginal likelihood for all the remission data $y_{rem} = [y_{rem}^1, \ldots, y_{rem}^M]$, resulting in a mixture of Gaussians that is a function of the data mean $\bar{y}_{rem}$ (see [15]).

$$\begin{aligned}p(y_{rem}\mid X_{ij}^k = x) &= \int p(y_{rem}\mid rem)p(rem\mid X_{ij}^k = x)\,d(rem)\\ &= \sum_{i=1}^{R} P(i)\int\prod_{m=1}^{M} p(y_{rem}^m\mid rem)p(rem\mid \mu_i)\,d(rem)\\ &= \sum_{i=1}^{R} P(i)\frac{1}{\sqrt{2\pi\left(\sigma_i^2+\frac{\sigma_y^2}{M}\right)}}\exp\left(-\frac{(\bar{y}_{rem}-\mu_i)^2}{2\left(\sigma_i^2+\frac{\sigma_y^2}{M}\right)}\right)\end{aligned} \tag{2}$$

As discussed above, although the measurements are independent given the material property: $p(y_{rem}\mid rem) = \prod_{m=1}^{M} p(y_{rem}^m\mid rem)$, equation 2 shows that the measurements are not independent when conditioned on the state: $p(y_{rem}\mid X_{ij}^k = x) \neq \prod_{m=1}^{M} p(y_{rem}^m \mid X_{ij}^k = x)$. Measurements are not independent sources of evidence with respect to the value of the state and only influence state inference indirectly through the latent material property.

Figure 6 further illustrates the effect of including the material property explicitly in the observation model. Figure 6(a) shows the GMM remission material property distribution (as given in equation 1) associated with the ground, vegetation and obstacle states. As the figure shows, the remission values are a moderately informative feature of state, with lower remission values tending to support a hypothesis of ground over vegetation. With only one measurement ($M = 1$), the remission observation model of equation 2 results in the remission probability distribution shown in Figure 6(b). In this case there are two significant sources of uncertainty: the remission material property associated with each state (as shown in Figure 6(a)) and the uncertainty associated with the measurement itself (represented by the quantity σ_y). These two sources of uncertainty combine to form the relatively high variance distribution shown in Figure 6(b). With 40 measurements, Figure 6(c) shows that the measurement uncertainty is significantly reduced yet the uncertainty with respect to



(a) Material property distribution

(b) 1 measurement

(c) 40 measurements

Fig. 6. The remission observation model (b) and (c) converges to the material property distribution (a) as the number of measurements increases

the possible remission values associated with each state (the quantity rem in equations 1 and 2) remains. As more measurements are incorporated, the appearance likelihood $p(y_{rem} \mid x)$ converges to the material property distribution $p(rem \mid x)$. This is also seen by inspection of equation 2. As $M \to \infty$ and as $\bar{y}_{rem} \to rem$ (by the law of large numbers), equation 2 converges to equation 1.

2) Density: Voxel density values range from empty space ($den = 0$) to completely solid ($den = 1$). Analogous to the GMM models used for voxel appearance properties, the distribution of density values for a given state x can be well modeled using a beta distribution $B(a_x, b_x)$ which operates in the range $[0, 1]$ and has parameters a_x and b_x that together specify its mean and variance.

Density is represented as the binary vector Y^n_{den} (lidar hit or pass-through), and we use a binomial distribution to describe the number of hits $M = \sum_{n=1}^{N} Y^n_{den}$ out of N total rays for a given voxel density property den. As above, we integrate over the nuisance parameter den, and we recover the beta-binomial distribution as the marginal likelihood observation model.

$$\begin{aligned} P(M = m \mid X^k_{ij} = x) \\ = \int P(m \mid den) p(den \mid X^k_{ij} = x)\, d(den) \\ = \binom{N}{M} \frac{B(a_x + M, b_x + N - M)}{B(a_x, b_x)} \end{aligned} \tag{3}$$

This model makes the assumption that a voxel with density den generates lidar hits that follow a binomial distribution (the outcome of repeated flips of a biased coin with $P(\text{heads}) = P(\text{hit}) = den$). However, since a given state x has a range of possible densities, which we model with a beta distribution, the distribution over hits M for a given state x becomes a beta-binomial, which has greater variance than a binomial for low amounts of data N, but converges to a binomial as N becomes large.

3) Free-space: The *free-space* state does not possess any meaningful material properties beyond density den. Lidar hits occurring in *free-space* are generally the result of noise so we model the non-density material properties as matching the material properties of the states in contact with *free-space*. For example, the voxel above a *ground* state voxel may contain many pass-throughs with a single hit due to noise that has an appearance that matches the *ground* state. If we modeled the appearance of *free-space* as uniform, then the strong match in appearance data with the *ground* state may overwhelm the density information and prevent the voxel from correctly being classified as *free-space*. By setting the appearance properties of *free-space* to match the state it is in contact with (*ground* in this example), the transition to *free-space* is decided solely on density information.

4) Obstacles: Although we expect obstacles to generally have a fairly high density den, we cannot hope to build an accurate observation model for the appearance of each of the innumerable obstacles one might encounter in outdoor environments, so we simply use a single *obstacle* state with a corresponding uniform distribution over the observable range of material appearance properties. We rely on accurately modeling the features of the trained states to detect obstacles as a default option when none of the other states are consistent.

B. Vertical Column Structure

As we move from lower to higher voxels within a column, we expect to move from *ground* to *vegetation*, or perhaps *ground* to *obstacle* and eventually to *free-space*. We never expect *free-space* to be found below *ground*, nor do we expect *vegetation* to be suspended above *free-space*. This type of structure is naturally imposed by introducing a Markov dependency between voxel states that restricts vertical transitions, thus defining a hidden Markov model within each voxel column. However, the duration of states such as $ground$ and $vegetation$ are not well modeled by repeated self-transitions in a Markov chain because this would induce a geometric distribution on the duration of those states. We resort instead to a hidden semi-Markov model (HSMM) [16] over voxel states, which explicitly represents a state duration (or height distribution) over voxels for each state value.

Figure 5(b) gives the intuition for a hidden semi-Markov model. Unlike an HMM which transitions (possibly to the same state) at every step, an HSMM remains in a single state value for some random duration (e.g. H^c_{ij} in Figure 5(b)), generating observations from that state at each step. At the completion of the duration, the state transitions to a new value according to the transition matrix and the process repeats. The transitions between states remain Markov, but the individual steps are not Markov since the probability of transitioning depends on how long the system has been in that state.

As shown in Figure 5(b), we associate a single HSMM chain structure with each column class C_{ij}, making the resulting column model a mixture of HSMMs. The durations of the ground and class states describe the height of those terrain elements and are given by H^g_{ij} and H^c_{ij}.

The vertical structure helps constrain the location of variables of interest such as the ground height hidden beneath dense vegetation. For example, in Figure 5(b), if we observe measurements that make X_{ij}^5 likely to be *vegetation* and X_{ij}^6 likely to be *free-space*, then the HSMM chain for column class $C_{ij} = vegetation$ would be likely, and we would also expect a *ground* to *vegetation* transition somewhere below X_{ij}^5 because of our vertical transition structure. As described in the next section, we can incorporate other information such as the expected vegetation height and the location of surrounding ground to further constrain the ground height even if it is hidden below dense vegetation and has no direct observations. This neighborhood information is incorporated into the column HSMM models as a prior over the duration H_{ij}^g of the ground state and the duration H_{ij}^c of the class state.

C. Horizontal Neighborhood Structure

In addition to the vertical structure captured by the HSMM column models, there are also significant horizontal dependencies between neighboring columns. The model encodes assumptions of class continuity, the smoothness of ground, and the similarity of vegetation height. This contextual information helps constrain the problem, especially in areas where dense vegetation hides the ground surface or in areas with missing or ambiguous sensor data. We use Markov random fields to encode our assumptions about class continuity and smooth ground.

The Markov random field (MRF) [17] is an extension of one-dimensional Markov chain models to two (or more) dimensions. It is commonly used in spatial domains to model structure such as smoothness, class continuity, or other properties correlating neighboring nodes. It has been successfully applied to many applications including segmentation, noise reduction, surface reconstruction, and texture classification [18].

Our terrain model uses a standard two-dimensional MRF, consisting of an undirected graph of nodes on a lattice structure that connects the voxel columns horizontally and encapsulates the conditional dependencies in the model. For example, the connections in Figure 5(c) show the Markov property for the ground height MRF:

$$P(H_{ij}^g \mid H_S^g) = P(H_{ij}^g \mid H_{N_{ij}}^g) \tag{4}$$

where H_S^g is the set of all ground height nodes and $H_{N_{ij}}^g$ is the set of neighbors of the ijth voxel column ground height H_{ij}^g. The set of neighbors $H_{N_{ij}}^g$ is also referred to as the Markov blanket of the node H_{ij}^g.

A key property of the MRF is that the joint probability over the entire MRF is uniquely specified by the conditional distributions that characterize its local Markov relationships [18]. Therefore, from easily expressible local relationships such as the notion that ground is smooth (with ground height changing little between neighboring voxel columns), we are able to infer global quantities such as the supporting ground surface heights of every voxel column.

As shown in Figure 5(c), we model horizontal structure using two distinct but interacting Markov random fields for class C_{ij} and ground height H_{ij}^g, and a latent variable for the common class height H^c across all columns. These variables interact through the HSMM column models by imposing a prior over the state durations associated with H_{ij}^c and H_{ij}^g and over the column class models C_{ij}.

The neighborhood dependency of C_{ij} reflects the prior assumption that class identities are positively correlated with their neighbors so voxel columns tend to cluster in contiguous groups of the same class. We express this preference using the conditional MRF distribution

$$P(C_{ij} = c \mid C_{N_{ij}}) \propto \exp\Big(-\lambda_C \sum_{\{s,t\} \in N_{ij}} (c \neq c_{st})\Big) \tag{5}$$

where N_{ij} is the set of neighboring indices and $C_{N_{ij}}$ is the set of classes in the neighborhood of the ijth voxel column.

Ground height varies smoothly from one patch of ground to the next, so we expect that H_{ij}^g will be tightly correlated with nearby values. We express this belief using a Gaussian Markov random field

$$P(H_{ij}^g = h \mid H_{N_{ij}}^g) \propto \exp\Big(-\frac{1}{2\sigma_G^2}\Big(h - \frac{1}{|N_{ij}|} \sum_{\{s,t\} \in N_{ij}} h_{st}^g\Big)^2\Big) \tag{6}$$

where $|N_{ij}|$ is the size of the neighborhood (in our experiments we used a 4-connected neighborhood).

We also expect that vegetation of the same class c has a similar height H^c with some variation. This assumption may not be valid for obstacles, so we only apply it to vegetation classes. Given the common height of the vegetation in this area H^c, we model the expected variation with a Gaussian truncated by the interval of possible class heights $I_{[h_{min}^c, h_{max}^c]}$

$$P(H_{ij}^c = h \mid H^c) \propto I_{[h_{min}^c, h_{max}^c]} \exp\Big(-\frac{1}{2\sigma_{H^c}^2}(h - h^c)^2\Big) \tag{7}$$

Because of the variation between different types of obstacles, we treat the class height of obstacle columns as independent.

Other constraints can also be included in the model. For example, we can fix the ground height under the wheels of the vehicle since these ground heights are known. The smooth ground prior in equation 6 then allows this information to propagate through the model and helps constrain the surrounding area.

Our use of Markov random fields to encode spatial correlations among variables is reminiscent of recent work in texture segmentation, where an image is assumed to be composed of a number of contiguous regions of uniform texture. One approach uses one MRF layer to classify textures and a second MRF layer that generates textures using separate parameters for each class [19]. This “double Markov random field” structure is related but distinct from our use of two MRFs in the terrain model described above. Similar to the double MRF, we maintain a single MRF for class segmentation that interacts with another MRF representing the ground surface, but rather than the parameters of one MRF depending on the states of the other, we interpret columns of data based on both the class MRF and the ground MRF.

Markov random fields have also been used in medical imaging applications to segment various tissues from 3D volumetric data [20]. Here material properties were represented in a voxel-based representation similar to our terrain model. However, in [20] the MRF was defined over the full 3D neighborhood of the voxel, whereas we are able to exploit inherent structure in our problem domain to keep the computationally demanding MRF structures to 2D and handling our 3D data in terms of voxel columns.

V. Inference

The interacting Markov random fields of this model capture important structure, but these dependencies prevent analytic determination of the posterior distribution $P(C, H^g, H^c \mid Y)$. The set of HSMMs that describe the data in each column of voxels can efficiently produce distributions over the state durations, which makes it easy to sample from the conditional distribution

$$P(C_{ij}, H^g_{ij}, H^c_{ij} \mid Y_{ij}, C_{N_{ij}}, H^g_{N_{ij}}, H^c) \tag{8}$$

so we use Gibbs sampling [17] to approximate inference.

Algorithm 1 describes the application of Gibbs sampling to our model. The HSMM column models require a distribution over class heights which comes from the common class height latent variable H^c, as shown in Figure 5(c). Samples of the common class height are produced from its conditional distribution given the current column class height samples h^c_{ij}

$$P(H^c{=}\,h \,|\, H^c_{ij\in IJ}) \propto \exp\Big(\frac{-1}{2\sigma^2_{H^c}/D^c}\Big(h - \frac{1}{D^c}\sum_{ij\in IJ, c_{ij}=c} h^c_{ij}\Big)^2\Big) \tag{9}$$

where D^c is the number of columns with class c.

Once the common class heights H^c have been sampled, each voxel column is sampled. The first step of the sampling procedure is to find the priors over class C_{ij}, class height H^c_{ij} and ground height H^g_{ij} from the neighbors, as given in equations 5 and 6, and the common class heights H^c as given in equation 7. The priors on H^c_{ij} and H^g_{ij} are then incorporated into the HSMM model as priors over state durations and are shown in the subsequent equations as $P(H^c_{ij} = h \mid H^c)$ for the class state $x = c$ or $P(H^g_{ij} = h \mid H^g_{N_{ij}})$ for the ground state $x = g$.

Once the prior distributions are found, the class HSMM structures are used to find the probability of the data and the state duration probabilities for each class. HSMMs use a variant of the standard forward-backward dynamic programming solution used for inference in regular HMMs [16]. As shown in figure 5(b), an HSMM maintains durations (corresponding to height in our case) so that a single state is active over a number of spatial steps up the chain. This formalism is very natural for finding ground height or class height because the neighborhood information can be included as a prior on the corresponding state duration.

Algorithm 1 Gibbs sampling from the model

```
Sample common class heights $h^c$ from $P(H^c \mid H^c_{ij\in IJ})$ using all the column class height samples of the same class
**for all** MRF voxel columns $ij$ **do**
  Find ground and class priors from neighbors:
    $P(H^g_{ij} \mid H^g_{N_{ij}})$
    $P(C_{ij} \mid C_{N_{ij}})$
  **for all** Classes $c$ **do**
    Find class height prior from common class height of same class:
      $P(H^c_{ij} \mid H^c)$
    Use class HSMM to find probability of the data and distributions over the ground and class height:
      $P(Y_{ij} \mid C_{ij} = c, H^g_{N_{ij}}, H^c)$
      $P(H^g_{ij} \mid C_{ij} = c, Y_{ij}, H^g_{N_{ij}}, H^c)$
      $P(H^c_{ij} \mid C_{ij} = c, Y_{ij}, H^g_{N_{ij}}, H^c)$
  **end for**
  Compute class distribution:
    $P(C_{ij} \mid Y_{ij}, C_{N_{ij}} H^g_{N_{ij}} H^c)$
      $\propto P(Y_{ij} \mid C_{ij}, H^g_{N_{ij}} H^c) P(C_{ij} \mid C_{N_{ij}})$
  Sample $c_{ij}$ from $P(C_{ij} \mid Y_{ij}, C_{N_{ij}}, H^g_{N_{ij}}, H^c)$
  Sample $h^g_{ij}$ from $P(H^g_{ij} \mid C_{ij} = c_{ij}, Y_{ij}, H^g_{N_{ij}}, H^c)$
  Sample $h^c_{ij}$ from $P(H^c_{ij} \mid C_{ij} = c_{ij}, Y_{ij}, H^g_{N_{ij}}, H^c)$
**end for**
```

The forward-backward computations are still performed over the individual spatial steps X^k_{ij} as in an HMM, but with an HSMM one must solve for the duration of each state, so in addition to summing over possible state transitions x', we also sum over possible state durations h. Equations 10 and 11 give the HSMM forward and backward probabilities $\alpha^k_{ij,c}$ and $\beta^k_{ij,c}$ for spatial step k of the class c chain in MRF voxel column ij. We use the observation independencies and the deterministic transitions of our chain structures to reduce the computational complexity. We use the notation x^- and x^+ to refer to the previous and next states in the chain of the current class.

$$\begin{aligned}\alpha^k_{ij,c}(x) &= P(\text{state } x \text{ ends at } k, Y^{1:k}_{ij} \mid C_{ij} = c, H^g_{N_{ij}}, H^c)\\ &= \sum_{x'}\sum_{h} P(X^k_{ij}{=}\,x, X^{k-h}_{ij}{=}\,x', H^x_{ij}{=}\,h, Y^{1:k}_{ij} \,|\, C_{ij}, H^g_{N_{ij}}, H^c)\\ &= \sum_{h}\prod_{k'=k-h+1}^{k} P(Y^{k'}_{ij} \mid x) P(H^x_{ij} = h \mid H^g_{N_{ij}}, H^c)\alpha^{k-h}_{ij,c}(x^-)\end{aligned} \tag{10}$$

$$\begin{aligned}\beta^k_{ij,c}(x) &= P(Y^{k+1:K}_{ij} \mid \text{state } x \text{ ends at } k, C_{ij} = c, H^g_{N_{ij}}, H^c)\\ &= \sum_{x'}\sum_{h} P(Y^{k+1:K}_{ij} | X^k_{ij}{=}\,x, X^{k+h}_{ij}{=}\,x', H^{x^+}_{ij}{=}\,h, C_{ij}, H^g_{N_{ij}}, H^c)\\ &= \sum_{h}\prod_{k'=k+1}^{k+h} P(Y^{k'}_{ij} \mid x^+) P(H^{x^+}_{ij} = h \mid H^g_{N_{ij}}, H^c)\beta^{k+h}_{ij,c}(x^+)\end{aligned} \tag{11}$$

Since we know by assumption that the chain must end in the final state $x =$ *free-space*, the probability of the data for class c is the final value of α in that state.

$$P(Y_{ij} \mid C_{ij} = c, H^g_{N_{ij}}, H^c) = \alpha^K_{ij,c}(x = \textit{free-space}) \quad (12)$$

As described in Algorithm 1, this is combined with the class prior $P(C_{ij} \mid C_{N_{ij}})$ to find the distribution over classes, which is used to sample a new class.

Finding the distribution over state durations involves combining α and β. As above, equation 13 takes advantage of the deterministic transitions of the chain structures to reduce computation.

$$\begin{aligned}\zeta^x_{ij,c}(h) &= P(\text{state } x \text{ has duration } h \mid Y_{ij}, C_{ij} = c, H^g_{N_{ij}}, H^c) \\ &= \sum_k P(X^k_{ij} = x, X^{k-h}_{ij} = x^- \mid Y_{ij}, C_{ij}, H^g_{N_{ij}}, H^c) \\ &= \sum_k \prod_{k'=k-h+1}^{k} P(Y^{k'}_{ij}|x) P(H^x_{ij}{=}h|H^g_{N_{ij}}, H^c) \alpha^{k-h}_{ij,c}(x^-) \beta^k_{ij,c}(x)\end{aligned} \quad (13)$$

We know that in each chain, every state transition must occur after some duration, so we can normalize by $\sum_h \zeta^x_{ij,c}(h)$ to get the posterior on ground and class height conditional on the neighbors. Samples are then drawn from these distributions.

$$\begin{aligned}P(H^g_{ij} = h \mid C_{ij} = c, Y_{ij}, H^g_{N_{ij}}, H^c) &= \zeta^{x=ground}_{ij,c}(h) \\ P(H^c_{ij} = h \mid C_{ij} = c, Y_{ij}, H^g_{N_{ij}}, H^c) &= \zeta^{x=\text{state } c}_{ij,c}(h)\end{aligned} \quad (14)$$

The time complexity of HSMM calculations is greater than an HMM because of the sum over possible durations, but the observation likelihood products $\prod P(Y^{k'}_{ij}|x)$ can be pre-computed and the state durations to search over can be constrained based on the priors to reduce the complexity to $O(numVoxels * numStates * maxDuration)$ for a single chain.

Although it is typically difficult to show that Gibbs sampling has converged, we have found empirically that the model finds a good estimate quickly, allowing for real-time execution.

VI. Learning

The model described in section IV incorporates prior knowledge about the structure of the environment, but the specific model parameters must be learned from training data. These parameters include the sensor observation models for each state and the neighborhood interactions for class, class height, and ground height. The generative nature of our model allows us to decouple the learning problems, and train each of these observation and neighborhood interaction models individually, thus greatly simplifying the learning task.

A. Observation Models

Collecting labeled training data is often expensive, especially in outdoor environments where there can be high variation in sensor readings so that a large training set is needed. We use an approach based on [14] to collect large quantities of labeled training data to automatically train our observation models. Specifically, we drive through representative terrain of a single class such as *vegetation* and store the sensor measurements from the voxels of columns that we drive over as training examples for that class. This process is then repeated for other classes such as *ground*. Unlike [14] which directly trains on the height of different types of vegetation, we only train on the various material properties of vegetation voxels, allowing us to remain general across vegetation heights.

Each labeled voxel collected by driving through representative terrain is used as a training example for the observation models in equations 1, 2, and 3. For appearance data such as remission, infrared and color, the mean values from each voxel are used to train the GMM observation models (i.e. for remission data μ_i, σ_i^2, $P(i)$ in equation 1) and the variance of measurements within the voxels is used as the GMM measurement model variance (σ_y^2 in equation 2).

Hit and pass-through data from the labeled training voxels are used to find the maximum likelihood parameters of the beta-binomial density model (a_x and b_x in equation 3) for each class state x using a Newton-Raphson method [21]. This handles class states like *ground* and *vegetation*, but the density of *obstacle* and *free-space* states must also be trained. The *free-space* density can be trained using data that includes insects or dust that occasionally returns a lidar point, or it can just be set manually to strongly favor empty space. Similarly, the *obstacle* density can be trained using hit and pass-through data from representative obstacles, or it can be set manually to favor dense objects.

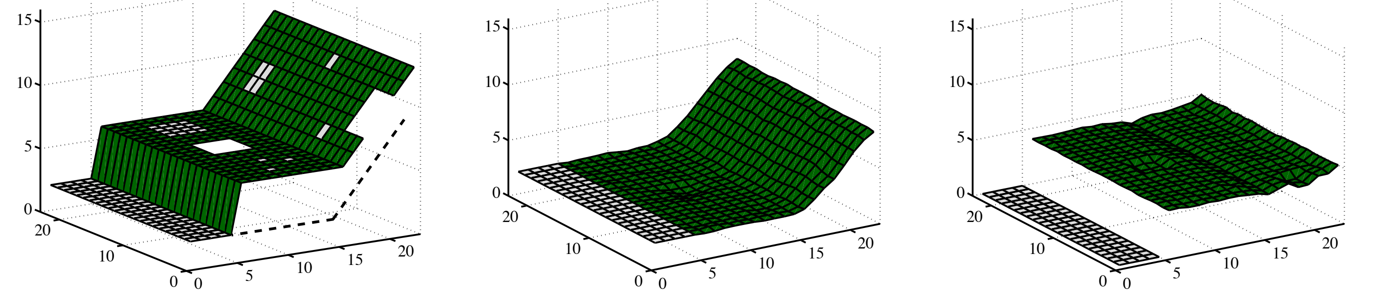
B. Neighborhood Models

The priors given in equations 5 and 6 describe how class and ground height depend on their neighbors, and the prior in equation 7 describes how column class heights are related to the common class height. Each of these priors contains parameters that quantify the tendency for classes to clump together, the smoothness of the ground, and the variability of vegetation class heights. As above, we train these parameters by driving over representative terrain.

As we drive over an area, we record the ground heights measured by the location of our wheels. We use these height sequences to find the standard deviation σ_G of typical ground height variation between voxel columns, which gives us the maximum likelihood estimate of our Gaussian MRF ground neighborhood prior [22].

Similarly, as we drive through vegetation, we get an approximate vegetation height measurement by taking the highest lidar hit and subtracting the known ground height (from the wheel locations). Since we assume that vegetation heights are independent given the common vegetation height in the area, we can find the class prior standard deviation σ_{H^c} directly from this sequence of class heights.

The class interaction prior λ_C gives the probability that a class transitions to a different class. This could be estimated directly using pseudo-likelihood methods [18] with class-labeled data over a large area that includes many class transitions, but unlike the labeled data for the observation models or the ground and class height interactions, this type of training data is difficult to collect. However, changing the



(a) Simulated input data

(b) Classification and ground height

(c) Classification and class height

Fig. 7. Simulation experiment showing transition from ground to tall vegetation with missing and ambiguous data. The dashed line in (a) shows the true ground height. The model is able to infer the hidden ground surface (b), the height of the vegetation (c), and the correct classification.

class interaction prior affects the system output in an intuitive way by controlling how much classes tend to clump together, so this parameter can be set manually.

VII. Run Time

Performing inference in this model is computationally intensive because of the repeated calculations necessary to sample from the model. We maintain a 150x150 grid of 15cm square voxel columns in our terrain map, which covers a 22.5 meter square area. Our system runs a loop that updates the local terrain map at approximately 1Hz. Within this loop, the system computes the observation likelihood products, calculates 20 samples from each column in the map, and updates the mean ground height, class height, and most likely class from the samples in each column.

At a vehicle speed of 1m/s, our procedure results in approximately 200 samples for a given terrain patch before the vehicle reaches it. Although sampling convergence is difficult to prove, the system generally finds the solution quite rapidly in our experiments, allowing us to run the system in real time.

VIII. Simulation Results

We have presented a terrain model that includes spatial correlations to better handle missing and ambiguous data in dense non-penetrable vegetation. The model is designed to recover an accurate estimate of the ground surface despite having only indirect observations and without needing to explicitly train on the vegetation height. This section shows a simple two-class simulation result that verifies these properties.

Figure 7(a) shows simulated data of a transition from ground to tall vegetation. Imagine the vehicle is approaching from the left, so initially the sensors get measurements of the ground, the front of the vegetation, and the top of the vegetation, but since the vegetation is dense there are no range measurements of the ground surface under the vegetation. The dashed line gives the true ground height, showing that the ground under the vegetation is flat and then angles up a hill. There are some columns with missing data, and some voxels in the vegetation match the appearance of ground, as shown by their light gray color. Although there is no data under the ground or vegetation surfaces, the voxels above the ground and vegetation are full of pass-throughs. The ground and vegetation appearance models were set to match the simulated input appearance data so we could study the effects of spatial correlation in the model.

⬜ Ground	🟩 Light yellow vegetation
🟥 Obstacle	🟩 Dark green vegetation

Fig. 8. Classes used in results

Since this example assumes the vehicle is approaching from the left, the system was initialized with the *ground* column class, a ground height of 2, and a voxel state class height of 0 (the voxel state *ground* class has no height). The sampling inference procedure given in algorithm 1 was then run for 100 iterations (each iteration produces samples from every column) which took 0.5 seconds. The final 50 samples from each column were used to find the most common class, the mean ground height, and the mean class height (although we allowed 50 iterations of "burn in" time to let the sampling procedure converge, the system actually converged after approximately 5 iterations).

Figure 7(b) shows the ground height estimates, Figure 7(c) gives the class height estimates, and both show the classification results for each column. These values represent the most likely explanation of the data given the prior knowledge encapsulated in the model. Although the system was never trained on the height of the vegetation, it was able to recover the vegetation height and use it to estimate the ground height including the hill. The ground smoothness and similar vegetation height assumptions combine to constrain the ground height at the transition from visible ground to vegetation and propagate the observed vegetation height at the transition through the rest of the simulated region.

The model structure also allowed the system to handle missing and ambiguous data. The class prior makes it likely that the areas in the vegetation with missing data are also vegetation. The ambiguous data patches in the vegetation have appearance properties similar to ground, but the ground smoothness prior makes it extremely unlikely for the ground to be at that height, so the model infers that these areas are actually vegetation.

The class height estimates in Figure 7(c) are not completely

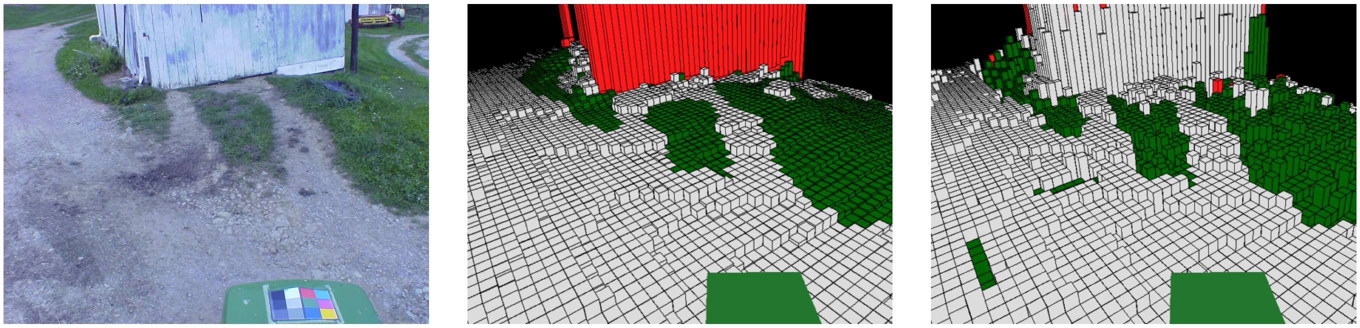

(a) View from the tractor (b) Spatial model output (c) Independent column output

Fig. 9. The white shed (a) has an appearance that is similar to the *ground* class, but the smooth ground prior makes such a height change very unlikely so the spatial model (b) classifies the shed as an *obstacle*. Classifying each column independently (c) results in the shed being classified as *ground*.

uniform. There is a crease where the hill starts because the model ground prior enforces a smooth transition from the flat region to the hill in Figure 7(b) whereas the simulated data in Figure 7(a) has an abrupt angle change. The class heights at the far right become slightly larger because of the conflict between the ground prior enforcing a smooth flat ground estimate and the class height prior enforcing a uniform estimate of the vegetation height.

The class height predictions are slightly lower in the missing data areas because of asymmetric data evidence. In the absence of any data, the class prior would give the missing data areas a symmetric class height distribution around the true class height. However, the large number of pass-throughs above the missing data areas produces a strong constraint that cuts off the top of the class height prior distribution. Therefore the class height samples in areas with missing data are biased low. Since there are no hits in that patch, it is reasonable to expect that the vegetation height is lower in this area.

IX. Results

We have tested this model with real data at a nearby working farm and an undeveloped area with tall weeds. The following examples highlight the benefits of including spatial correlations in areas with dense non-penetrable vegetation. We demonstrate the system's ability to detect obstacles in vegetation and we also show what happens when the smooth ground assumption is violated.

In each case, after training the model on representative terrain, we drive the vehicle through the test area while letting the Gibbs sampler run continuously. Running at 1Hz, the system calculates the observation likelihoods, computes samples from the model, and updates the local terrain map with the most commonly sampled class and the mean of the ground height and class height samples.

As mentioned in section I, our goal is to find the supporting ground surface and the location of obstacles. The result figures in this section show a single class color label (see Figure 8) for each column, giving the most commonly sampled class. Each column also displays a single height based on the mean of one of the transitions in that column, but which transition is displayed is a function of the column's class. For columns labeled as *ground* or *vegetation*, we care about the supporting ground surface, so the results show the inferred ground height (the transition from *ground* to *free-space* or *ground* to *vegetation*). For *obstacle* columns, we care about the height of the obstacle, so the results show the class height on top of the ground height (the transition from *obstacle* to *free-space*).

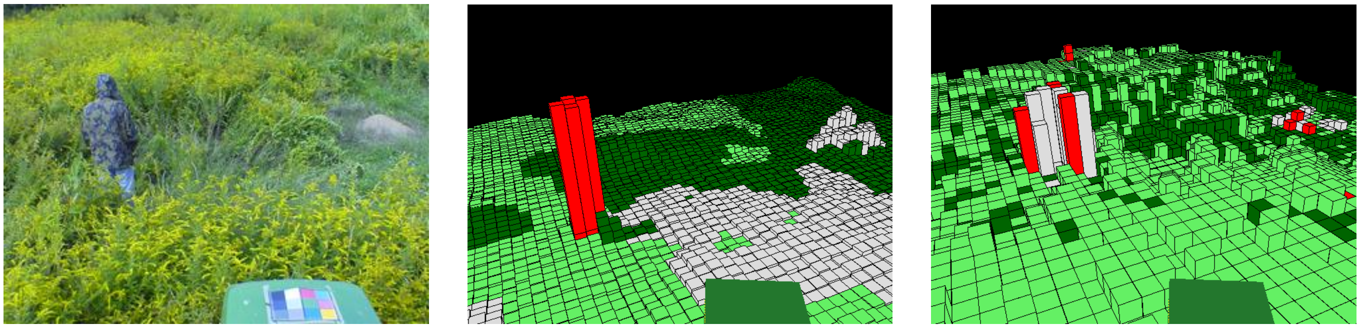
To show the benefit of including spatial correlations, we compare our model result with a system that uses the same trained observation models but makes independent classifications for each column instead of incorporating spatial structure. This comparison system produces the most likely class for each column using only the observations in the voxels in that column. To produce results comparable to our model output, it uses the highest hit in the column as the transition to *free-space*. For *vegetation* class columns, it uses the lowest hit or pass-through in that column as its estimate of ground height.

A. White shed

Figure 9 shows the tractor as it approaches a white shed. This is a large obstacle that could be reliably detected in a variety of ways, but it will serve as a good example of how the various pieces of our model interact to produce the correct result. Figure 9(b) shows the output of the model, including a useful classification of the scene and a smooth ground estimate that would work well for vehicle navigation. It classifies the shed as an obstacle and correctly captures the hill sloping down to the right despite the presence of sparse vegetation.

This example is interesting because the voxels containing the walls of the shed have observations that make the *ground* class much more likely than the broad uniform *obstacle* class. However, the spatial prior on a smooth ground surface makes it extremely unlikely that the ground height would have a tall step discontinuity at the shed wall. Since the density and appearance data are not well described by the vegetation class, the shed is correctly classified as an obstacle.

Figure 9(c) shows the output of the system when the neighborhood interactions are ignored and the columns are assumed to be independent. Without neighborhood information, classification is based solely on the data likelihood for each column HSMM model. Lacking the smooth ground prior, the wall is classified as a collection of tall columns of ground voxels. A vision system that ignores 3D structure and only



(a) View from the tractor

(b) Spatial model output

(c) Independent column output

Fig. 10. (a) A challenging scene with a person in camouflage in tall weeds with low grass and a dirt mound. The person and dirt mound both have a high temperature that matches the *ground* class, but the spatial model (b) disambiguates the two by inferring the hidden ground beneath the vegetation. The independent result (c) shows the ground heights on top of the dense tall weeds and misclassifies parts of the person and dirt mound.

makes a classification based on the observation models we use would produce a similar result. Figure 9(c) also shows that without the spatial ground and class priors, the ground height estimates and classification labels are generally more noisy.

If the white shed actually was an earthen cliff, our model would produce similar predictions and the wall of dirt would be labeled as an obstacle. Our ground smoothness prior encapsulates the assumption that such abrupt changes in the ground surface don't occur. This is not a problem since the output terrain model is used to find a drivable ground surface for robot navigation so a wall of dirt should be classified as an obstacle.

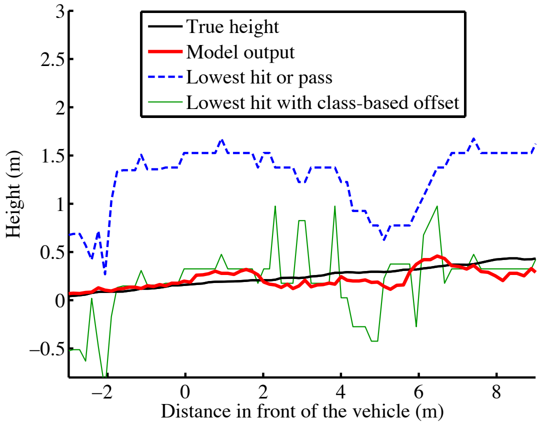


Fig. 11. Ground estimate comparison for the area in Figures 10(b) & 10(c), showing the improved ground height estimates of the spatial model

B. Tall vegetation

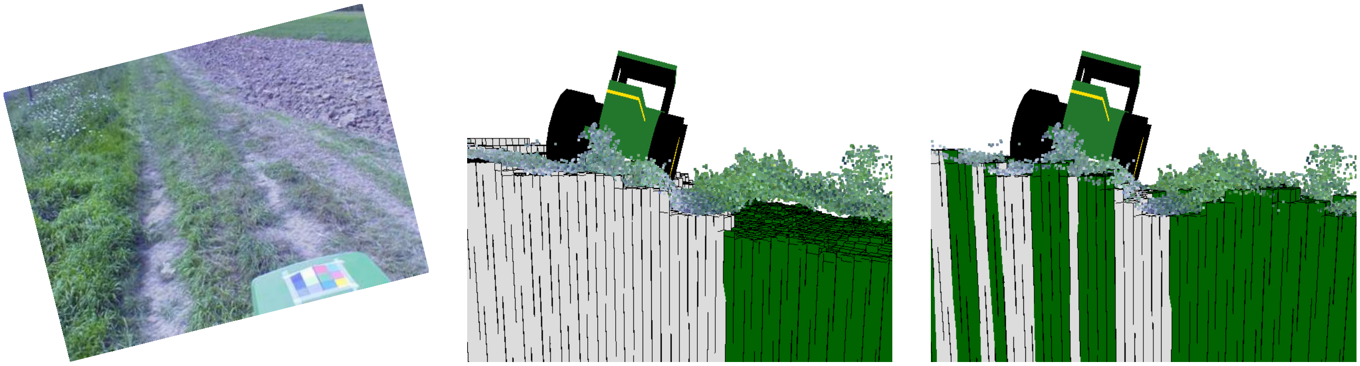
Figure 10 shows a challenging scene containing a camouflaged person in tall weeds with low grass and a small dirt mound to the right. Both the person and the dirt mound have high infrared temperature, so a simple obstacle detection system that performs a threshold on temperature may classify both of these objects as obstacles. We show that the structural assumptions embodied in our model allow the system to disambiguate these two objects.

We trained the model on bare ground, low grass, and tall weeds. Figure 10(b) gives the model ground height and classification results. Inference over the model results in the correct classification of the person and the dirt mound, as well as the two types of vegetation. The area to the right of the person in the shadow of the tall weeds is classified as ground. Although that area is actually low grass, since the system has no data from the area, ground is a reasonable estimate.

Using the model structure and the known ground height under the vehicle allows the system to produce reasonable estimates of the ground height even in areas where the ground is hidden. In addition to providing a smoothing prior, neighborhood interactions allow information to propagate. Fixing the heights under the wheels affects the ground estimates in the surrounding area. Columns with little or no data can still produce useful estimates by relying on their neighborhood. The system can infer the common vegetation height of the tall weeds from areas where the ground is more directly observable, such as under the vehicle or the transition to tall weeds behind the person. The assumption of a common vegetation height, in turn, allows the system to infer the ground height in areas where the ground is not directly observable. Knowing the ground height allows the model to explain the dirt mound as a rise in the ground but the person as an obstacle.

Assuming independence prevents information from propagating. Figure 10(c) shows the lowest point and the class estimates when no neighborhood information is used. The lowest point does not penetrate the dense vegetation so it provides poor estimates of the ground height. Both the dirt mound and the person are classified as a mixture of dirt and obstacle columns. Also without neighborhood information, the vegetation class predictions contain many isolated misclassifications due to ambiguous data.

Figure 11 illustrates the quality of the ground height estimates from Figure 10. After computing estimates of the ground height using our model, we drove through the scene toward the area between the person and the dirt mound, and made measurements of the ground height using our wheel locations. This trajectory is marked as "True height" in Figure 11, and offers a comparison for the estimates produced by the model and those using the lowest hit or pass-through in



(a) View from the tractor

(b) Spatial model output

(c) Independent column output

Fig. 12. The tractor (a) is on a steep side slope with dense vegetation at the bottom of the hill. The spatial model (b) infers the ground height beneath the vegetation which could correctly trigger a roll-over hazard. The independent model (c) incorrectly predicts that the tractor could drive on top of the vegetation.

each column. The model ground estimates are fairly smooth and stay within approximately 20cm of the true value.

Current approaches that filter out vegetation from the ground surface generally rely on the deepest range penetration, but for dense non-penetrable vegetation this performs very poorly since there are no range points that reach the ground. Therefore, in addition to comparing our model to the lowest hit or pass-through, we also compare it to an approach that adjusts the lowest hit in each column based on the independent column classifications shown in Figure 10(c). Instead of using spatial structure to infer the vegetation height from the data as in our model, this approach simply uses the average height of each class from the training data for the offset. Figure 11 shows that this can work well when the classification is correct and the actual vegetation height matches the training data, but it suffers from misclassification and the lack of a smoothing prior.

C. Vegetation on a Slope

The previous section demonstrated how the model's structural assumptions enabled improved ground height estimations. However, in that example the ground was generally flat so a system that simply classified all vegetation as safe could have succeeded without correctly estimating the ground surface (although detecting positive obstacles such as the person would be more difficult without the ground height). This section shows an example where treating vegetation as drivable could lead to dangerous behavior but finding the supporting ground surface enables the vehicle to stay safe.

Figure 12 shows the tractor as it is driving along a path on a steep 14-degree side slope with dense vegetation at the bottom of the hill. The vegetation at the bottom covers and hides the supporting ground surface underneath. Figures 12(b) and 12(c) show a view from in front of the tractor of the range point data, as well as estimates of the ground height and the classifications. As in the other result figures, each column is colored using the class label of that column, and the height of the *vegetation* columns shows the transition from *ground* to *vegetation*. If instead we showed the states of the individual voxels, then the portions of the columns visible in Figures 12(b) and 12(c) would be *ground* state voxels and there would be *vegetation* state voxels above the displayed transition from *ground* to *vegetation*.

The path that the tractor is driving on has a high side slope, and the ground becomes even steeper on the lower part of the hill under the vegetation, which could result in a roll-over hazard. The dense vegetation prevents laser measurements of the ground, so a system that uses the lowest point for the ground height would treat the top of the vegetation as the supporting surface, as shown by the ground height predictions in Figure 12(c). This would make that part of the hill appear to be fairly flat and traversable, when it actually contains a steep side slope and could represent a roll-over hazard.

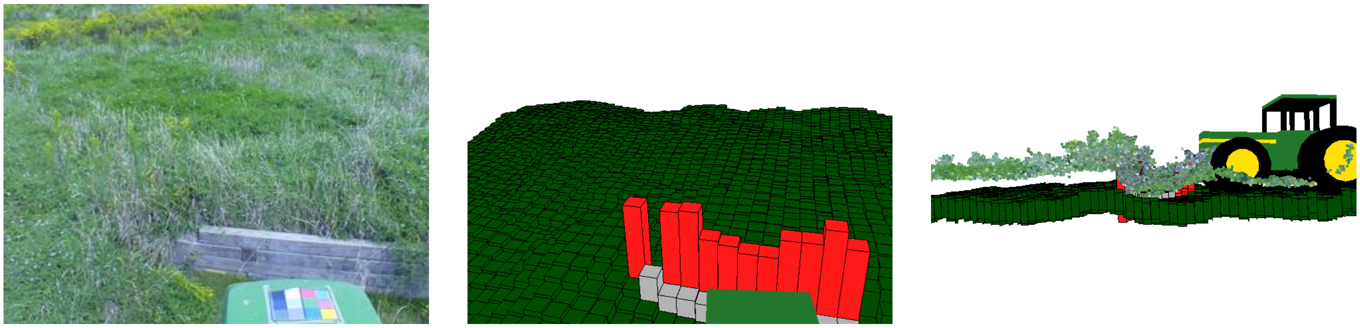
Figure 12(b) shows the spatial model ground height estimates. This approach correctly infers a ground surface below the vegetation, and the resulting ground surface predicts a high slope in that area that could be used with a vehicle model to check for roll-over conditions.

The model assumptions of smooth ground and similar vegetation height enable the system to infer the ground surface below the vegetation, even though the system was never trained on the height of the vegetation. As in the simulation example in section VIII, the transition from ground to vegetation at the edge of the path allows the system to infer a vegetation height which then propagates through the spatial correlations in the model to drop the ground surface below the top of the vegetation. The system also tries to produce smooth ground estimates between the observed ground height on the path near the tractor and the data from the bottom of the slope (not visible in the figures). These various constraints combine to produce an accurate ground estimate in this difficult example.

D. Ledge Step Hazard

The previous sections looked at examples where the model assumptions of smooth ground, class continuity, and similar vegetation height were generally correct. This section explores what happens when model assumptions are incorrect.

Our main test area did not have any areas with non-smooth ground so a wooden ledge was constructed. Figure 13 shows the tractor as it approaches this ledge, and Figure 13(b) gives the model ground height estimates. The ledge has an appearance that matches the ground class, so there is a row



(a) View from the tractor

(b) Spatial model output

(c) Spatial model output (side)

Fig. 13. (a) A ledge creating a discontinuity in the ground surface. The ledge violates the smooth ground assumption so the model (b) makes the reasonable inference that the ledge is an obstacle. (c) The ground height predictions beyond the ledge are lower than the true ground height.

of columns classified as ground, but the ground prior makes the taller portions of the ledge unlikely to be ground and their appearance does not match any other class, so the ledge is reasonably classified as an obstacle.

As shown in Figure 13(b), the ground estimates beyond the ledge are significantly lower than the true ground height. The model has explained the higher data points beyond the ledge with dense medium height vegetation on low ground instead of short vegetation on higher ground. Figure 13(c) provides some insight into why the model produces this estimate. The vehicle is positioned in taller vegetation of similar appearance to the shorter vegetation beyond the ledge. Based on the similarity in appearance, the system infers that the two patches of vegetation belong to the same vegetation class. Consequently, the assumption of common vegetation height propagates the height of the taller vegetation surrounding the vehicle to the area beyond the ledge. This combines with the smooth ground prior to underestimate the ground height beyond the ledge.

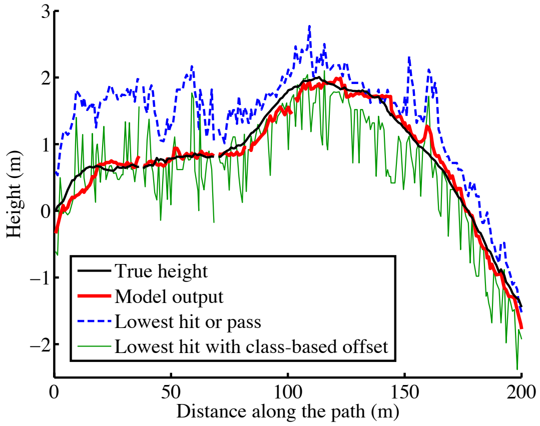


Fig. 14. Ground estimate comparison for longer test path through varied vegetation, showing predictions made 6m in front of the vehicle as the vehicle drove (note the two axes have very different scales)

E. Longer run through varied vegetation

Figure 14 shows ground height estimates for a longer test set through varied vegetation. Unlike Figure 11, which presents a snapshot of the predictions at different distances in front of the vehicle at a given time, Figure 14 shows predictions at a constant 6 meters in front of the vehicle as the vehicle moves forward. The lowest hit line shows that the first 70m of the path contained two sections of tall dense non-penetrable vegetation, and the remainder of the path consisted of low vegetation with various tall sparse vegetation and a few small patches of dense vegetation (e.g. 170m). The model output is generally smooth and closely matches the true height, whereas the lowest hit rarely reaches the ground, and the estimate using lowest hit with class offset is often correct but very noisy because of misclassifications due to its independence assumption.

X. Conclusion

Our probabilistic terrain model provides a natural way of combining different types of sensor data with reasonable assumptions about the world—particularly in agricultural applications—such as ground smoothness, class continuity, and similarity in vegetation height. These assumptions are incorporated as a model prior and help constrain the problem when the sensor data is ambiguous. Except for the class neighborhood prior, which can be easily tuned by hand, all model parameters that control the difficult task of weighting the different sensor data appropriately are automatically learned by the system by simply driving through representative terrain. Our approach can find obstacles without needing to explicitly model them or collect obstacle appearance training data.

We have applied this approach in realistic test conditions and have shown that the use of spatial structure in our model improves ground height estimation and obstacle classification over an equivalent model that ignores spatial structure. Although computationally intensive, the algorithm can run in real time for moderate vehicle speeds of 1-2m/s.

In areas where our model assumptions are generally true, the model structure allows the system to infer the supporting ground surface even when the ground is hidden by dense vegetation. Joint inference of class, ground height, and vegetation height allows the system to produce better estimates of each, since knowing the ground height helps disambiguate obstacles from the ground, and knowing the class and vegetation height helps determine the ground height.

When model assumptions are violated, the model often produces a desirable result, such as treating a ground discontinuity as an obstacle, even though the resulting ground height estimates are inaccurate. This model could also be extended to include additional class models for overhanging obstacles and holes in the hope of broadening the set of environments where our assumptions are valid.

Although we can generally perform inference in the model in real time for moderate vehicle speeds, the algorithm presented is still computationally demanding. Perhaps other approximate inference schemes would be less computationally intensive than Gibbs sampling, while still offering the benefits of including the spatial constraints presented in this model.

Acknowledgement

This work was supported by John Deere under contract 476169

References

[1] A. Kelly and A. Stentz, “Rough terrain autonomous mobility - part 2: An active vision, predictive control approach,” *Autonomous Robots*, vol. 5, no. 2, pp. 163–198, May 1998.

[2] M. Daily, J. Harris, D. Keirsey, D. Olin, D. Payton, K. Reiser, J. Rosenblatt, D. Tseng, and V. Wong, “Autonomous cross-country navigation with the ALV,” in *IEEE Int. Conf. on Robotics and Automation*, vol. 2, April 1988, pp. 718–726.

[3] A. Lacaze, K. Murphy, and M. DelGiorno, “Autonomous mobility for the Demo III experimental unmanned vehicles,” in *Assoc. for Unmanned Vehicle Systems Int. Conf. on Unmanned Vehicles (AUVSI 02)*, July 2002.

[4] L. R. Rabiner, “A tutorial on hidden markov models and selected applications in speech recognition,” *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[5] M. Ollis and A. T. Stentz, “Vision-based perception for an autonomous harvester,” in *IEEE Int. Conf. on Intelligent Robots and Systems*, vol. 3, September 1997, pp. 1838 – 1844.

[6] I. Davis, A. Kelly, A. Stentz, and L. Matthies, “Terrain typing for real robots,” in *Intelligent Vehicles Symposium*, September 1995, pp. 400–405.

[7] M. Hebert, N. Vandapel, S. Keller, and R. R. Donamukkala, “Evaluation and comparison of terrain classification techniques from LADAR data for autonomous navigation,” in *23rd Army Science Conference*, December 2002.

[8] J. Macedo, R. Manduchi, and L. Matthies, “Ladar-based discrimination of grass from obstacles for autonomous navigation,” in *Int. Symp. on Experimental Robotics*, December 2000.

[9] A. Castano and L. Matthies, “Foliage discrimination using a rotating ladar,” in *IEEE Int. Conf. on Robotics and Automation*, vol. 1, September 2003, pp. 1–6.

[10] N. Vandapel, M. Hebert, D. Huber, and A. Akapuria, “Natural terrain classification using 3-D ladar data,” in *IEEE Int. Conf. on Robotics and Automation*, April 2004.

[11] N. Vandapel and M. Hebert, “Finding organized structures in 3-d ladar data,” in *IEEE Int. Conf. on Intelligent Robots and Systems*, 2004.

[12] A. Kelly, “Toward reliable off-road autonomous vehicle operating in challenging environments,” in *International Symposium on Experimental Robotics*, June 2004.

[13] M. Ollis and T. Jochem, “Structural method for obstacle detection and terrain classification,” in *SPIE Unmanned Ground Vehicle Technology*, 2003.

[14] C. Wellington and A. Stentz, “Online adaptive rough-terrain navigation in vegetation,” in *IEEE Int. Conf. on Robotics and Automation*, April 2004.

[15] C. Wellington, “Learning a terrain model for autonomous navigation,” Ph.D. dissertation, Robotics Institute, Carnegie Mellon University, 2005.

[16] M. Ostendorf, V. Digalakis, and O. Kimball, “From HMMs to segment models: a unified view of stochastic modeling for speech recognition,” *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 4, pp. 360–378, 1996.

[17] S. Geman and D. Geman, “Stochastic relaxation, gibbs distributions, and the bayesian restoration of images,” *IEEE Trans. Pattern Anal. Machine Intell.*, 1984.

[18] S. Z. Li, *Markov Random Field Modeling in Image Analysis*, 2nd ed., ser. Computer Science Workbench. Springer-Verlag, 2001.

[19] D. E. Melas and S. P. Wilson, “Double markov random fields and bayesian image segmentation,” *IEEE Trans. Signal Processing*, vol. 50, no. 2, pp. 357–365, February 2002.

[20] Y. Zhang, M. Brady, and S. Smith, “Segmentation of brain MR images through a hidden markov random field model and the expectation-maximization algorithm,” *IEEE Trans. Med. Imag.*, vol. 20, no. 1, pp. 45–57, 2001.

[21] D. M. Smith, “Algorithm AS 189: Maximum likelihood estimation of the parameters of the beta binomial distribution,” *Applied Statistics*, vol. 32, no. 2, pp. 196–204, 1983.

[22] S. S. Saquib, C. A. Bouman, and K. Sauer, “ML parameter estimation for markov random fields with applications to bayesian tomography,” *IEEE Transactions on Image Processing*, vol. 7, no. 7, pp. 1029–1044, 1998.