A Generative Model of Terrain for Autonomous Navigation in Vegetation
Carl Wellington, Aaron Courville and Anthony (Tony) Stentz
The International Journal of Robotics Research 2006; 25; 1287
DOI: 10.1177/0278364906072769

The online version of this article can be found at:
http://ijr.sagepub.com/cgi/content/abstract/25/12/1287
A Generative Model of Terrain for Autonomous Navigation in Vegetation

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. A generative, probabilistic terrain model is introduced 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.

KEY WORDS—autonomous navigation, terrain modeling, vegetation, probabilistic inference, learning, Markov random fields, hidden semi-Markov models, Gibbs sampling

1. 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 (Kelly and Stentz 1998). 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 (Daily et al. 1988; Kelly and Stentz 1998; Lacaze et al. 2002). 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.

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 (Rabiner 1989) 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 (Ollis and Stentz 1997).

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 allows one to 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 this approach is computationally demanding, the system can run in real time.

2. 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 (Kelly and Stentz 1998; Daily et al. 1988). 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 (Davis et al. 1995) (also common in the remote sensing community—see references in Hebert et al. 2002) 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.

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 (Macedo et al. 2000), and they have compared measurements across time and space to filter out areas where the penetration is continually changing (Castano and Matthies 2003). A comparison between techniques that look for the range shadow of solid obstacles and techniques based on local point statistics is given in Hebert et al. (2002).

The strategy of computing local statistics about the spread of points was expanded in Vandapel et al. (2004) to discriminate between sparse vegetation, solid surfaces, linear structures such as branches, and even concertina wire (Vandapel and Herbert 2004).

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 (Lacaze et al. 2002; Kelly 2004) or lidar and radar measurements (Ollis and Jochem 2003).

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 (Wellington and Stentz 2004). 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.

3. 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 (Wellington and Stentz 2004). 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 obtain range data over a large area in front of the vehicle, and the second scanning lidar is mounted on the bumper to obtain 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 15cm\(^3\) box-shaped region of 3-dimensional space. We represent the vehicle’s local spatial environment as a voxel lattice of size I × J × K, where the \(ijk\)th voxel is in the \(ij\)th position of a horizontal 2D grid and the \(k\)th 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 appear-

---

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.
sor measurement vectors from an array of sensor modalities, a
ervation vector formed by the concatenation of multiple sen-
Xk able that describes the nature of the material inside the voxel, 
ble contents of the column, 

\[ C_{ij} \in \{c\} \]

The voxel representation also allows us to maintain a den-
ity estimate throughout space by comparing how many lidar 
ays pass through each voxel (pass-throughs) with the number of 
idar rays that hit something in that voxel (hits). Density 
formation is valuable when trying to separate sparse vege-
tation 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 (Lacaze et al. 2002; Wellington 
and Stentz 2004).

Although our data representation is based on the voxel, 
ehicle navigation is generally performed on a 2D surface, 
so our ground height estimates and classification results are 
ade in terms of voxel columns. In our model, the \( i \)th voxel 
column class is described with a multinomial distributed 
random variable \( C_{ij} \) taking on values related to the possible 
ents of the column, \( C_{ij} = c \) with, for example, 
\( c \in \{\text{ground, vegetation, obstacle}\} \).

Associated with the \( k \)th voxel in the \( i \)th voxel column is 
the voxel state \( X^k_{ij} \), a multinomial distributed random vari-
able that describes the nature of the material inside the voxel, 
\( X^k_{ij} \in \{\text{ground, c, free-space}\} \), where \( c \) is the class of the \( i \)th voxel column.\(^2\) The \( ijk \)th voxel is also associated with an ob-
servation vector formed by the concatenation of multiple sen-
or measurement vectors from an array of sensor modalities, 
\( Y^k_{ij} = \{Y_{den}, Y_{rem}, Y_{ir}, Y_{col}\} \). The \( N \) lidar measurements are en-
coded as a binary vector \( Y_{den} \) consisting of \( M \) ones represent-
ing hits and \( N-M \) zeros representing pass-throughs. Associa-
ated with each of the \( M \) hit measurements are corresponding 
measurements of laser remission, infrared temperatures and 
color (i.e. \( Y_{rem} = [Y_{rem}^1, \ldots, Y_{rem}^N], Y_{ir} = [Y_{ir}^1, \ldots, Y_{ir}^M] \)).

4. 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.

Before describing each level of the model in detail, we 
offer a perspective on the entire model by exploring the pro-
cess 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 simi-
lar 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 mem-
berships 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 par-
ticular 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 veg-
etation height are vegetation and above that the voxels are 
free-space.
4.1. 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 9.1). 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.

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 5. We now describe the three levels of the model in more detail.
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 into 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 \( X^i_k \), the material property random variables \( den, rem, ir \) and \( col \), and the measurements. Conditional on \( X^i_k \), 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.

4.1.1. 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 \( \mu_i \), variance \( \sigma^2_i \), and mixing coefficient \( P(i) \). Note that these parameters are unique to the \( rem \) material property. Color and infrared data are parameterized independently.

\[
p(rem | X^i_k = x) = \sum_{i=1}^{R} P(i) \frac{1}{\sqrt{2\pi\sigma^2_i}} \exp \left( -\frac{(rem - \mu_i)^2}{2\sigma^2_i} \right)
\]

(1)

Conditional on the true material property \( rem \), the measurements \( y^m_{rem} \) are assumed to be normally distributed, \( y^m_{rem} \sim N(rem, \sigma^2) \). We integrate out the nuisance variable \( rem \) to get the marginal likelihood for all the remission data \( y_{rem} = [y^1_{rem}, \ldots, y^M_{rem}] \), resulting in a mixture of Gaussians that is a function of the data mean \( \bar{y}_{rem} \) (see Wellington 2005).

\[
p(y_{rem} | X^i_k = x) = \int p(y_{rem} | rem) p(rem | X^i_k = x) d(rem)
\]

\[
= \sum_{i=1}^{R} P(i) \prod_{m=1}^{M} p(y^m_{rem} | rem) p(rem | \mu_i) d(rem)
\]

\[
= \sum_{i=1}^{R} P(i) \frac{1}{\sqrt{2\pi(\sigma^2 + \frac{\sigma^2}{M})}} \exp \left( -\frac{(\bar{y}_{rem} - \mu_i)^2}{2(\sigma^2 + \frac{\sigma^2}{M})} \right)
\]

(2)

As discussed above, although the measurements are independent given the material property: \( p(y_{rem} | rem) = \prod_{m=1}^{M} p(y^m_{rem} | rem) \), equation 2 shows that the measurements are not independent when conditioned on the state: \( p(y_{rem} | X^i_k = x) \neq \prod_{m=1}^{M} p(y^m_{rem} | X^i_k = x) \). 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 \( \sigma \)). 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 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 \( p(y_{rem} | x) \) converges to the material property distribution \( p(rem | x) \). This is also seen by inspection of equation 2. As \( M \rightarrow \infty \) and as \( \bar{y}_{rem} \rightarrow rem \) (by the law of large numbers), equation 2 converges to equation 1.

4.1.2. 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 \( B(\alpha_x, b_x) \), 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 $Y_{den}$ (lidar hit or pass-through), and we use a binomial distribution to describe the number of hits $M = \sum_{n=1}^{N} Y_{den}^n$ 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.

$$P(M = m \mid X_{ij}^x) = \int P(m \mid den) p(den \mid X_{ij}^x) d(den) = \binom{N}{M} B(a_x + M, b_x + N - M)$$

(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 $P$(heads) = $P$(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.

4.1.3. 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.

4.1.4. 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.

4.2. 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) (Ostendorf et al. 1996) 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 for the random duration (e.g. $H_{ij}$ 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 $H_{ij}^{g}$ and $H_{ij}^{c}$.

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 HSM model for column class $C_{ij} = \text{vegetation}$ 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 HSM models as a prior over the duration $H_{ij}^{g}$ of the ground state and the duration $H_{ij}^{c}$ of the class state.

4.3. Horizontal Neighborhood Structure

In addition to the vertical structure captured by the HSM 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) (Geman and Gemen 1984) 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 (Li 2001).

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:

$$P(H_{ij}^{g} \mid H_{s,t}^{g}) = P(H_{ij}^{g} \mid H_{ij}^{g})$$

where $H_{ij}^{g}$ is the set of all ground height nodes and $H_{ij}^{g}$ is the set of neighbors of the $ij$th voxel column ground height $H_{ij}^{g}$. The set of neighbors $H_{ij}^{g}$ 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 (Li 2001). 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 $H_{ij}$ across all columns. These variables interact through the HSM column models by imposing a prior over the state durations associated with $H_{ij}^{g}$ and $H_{ij}^{c}$ 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

$$P(C_{ij} = c \mid X_{N_{ij}}) \propto \exp \left( -\lambda_{c} \sum_{(i,j) \in N_{ij}}(c - c_{ij}) \right)$$

where $N_{ij}$ is the set of neighboring indices and $C_{ij}$ is the set of classes in the neighborhood of the $ij$th voxel column.
Ground height varies smoothly from one patch of ground to the next, so we expect that $H_i^c$ will be tightly correlated with nearby values. We express this belief using a Gaussian Markov random field

$$P(H_i^c = h | H^c_{Nij}) \propto \exp\left(-\frac{1}{2\sigma^2} (h - \frac{1}{|N_{ij}|} \sum h^c_{n})^2\right)$$

(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 $I_{[hc_{min},hc_{max}]}$

$$P(H_i^c = h | H^c) \propto I_{[hc_{min},hc_{max}]} \exp\left(-\frac{1}{2\sigma^2} (h - h^c)^2\right)$$

(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 (Melas and Wilson 2002). This “double Markov random field” structure is related but distinct from our use of 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 (Zhang et al. 2001). Here material properties were represented in a voxel-based representation similar to our terrain model. However, in Zhang et al. (2001) 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.

5. Inference

The interacting Markov random fields of this model capture important structure, but these dependencies prevent analytic determination of the posterior distribution $P(C, H^c, H^v | Y)$. 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

$$P(C_{ij}, H^c_{ij}, H^v_{ij} | Y_{ij}, C_{Nij}, H^c_{Nij}, H^v)$$

so we use Gibbs sampling (Geman and Geman 1984) for 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 $h^c_{ij}$

$$P(H^c = h | H^c_{(i \neq j)}) \propto \exp\left(-\frac{1}{2\sigma^2} \sum (h - h^c)^2\right)$$

(9)

where $D'$ 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

Algorithm 1. Gibbs sampling from the model

Sample common class heights $h^c$ from $P(H^c | H^c_{(i \neq j)})$ 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:

$$P(H^c_{ij} | H^c_{Nij})$$

$$P(C_{ij} | C_{Nij})$$

for all Classes $c$ do

Find class height prior from common class height of same class:

$$P(H^c_{ij} | c, H^c_{Nij}, H^v)$$

Use class HSMM to find probability of the data and distributions over the ground and class height:

$$P(Y_{ij} | C_{ij}, C_{Nij}, H^c_{Nij}, H^v)$$

$$P(H^v_{ij} | C_{ij} = c, Y_{ij}, H^c_{Nij}, H^h)$$

end for

Compute class distribution:

$$P(C_{ij} | Y_{ij}, C_{Nij}, H^c_{Nij}, H^v) \propto P(Y_{ij} | C_{ij}, H^c_{Nij}, H^v)P(C_{ij} | C_{Nij})$$

Sample $c_{ij}$ from $P(C_{ij} | Y_{ij}, C_{Nij}, H^c_{Nij}, H^v)$

Sample $h^c_{ij}$ from $P(H^c_{ij} | C_{ij} = c_{ij}, Y_{ij}, H^c_{Nij}, H^v)$

Sample $h^c_{ij}$ from $P(H^c_{ij} | C_{ij} = c_{ij}, Y_{ij}, H^c_{Nij}, H^v)$

end for
The priors on \( \beta_k \) and \( \alpha_k \) are used to find the probability of the data and the common class heights (corresponding to height in our case) so that a single chain in MRF voxel column \( ij \) is used to sample a new class. We know that in each chain, every state transition must occur after some duration, so we can normalize by \( \sum_\beta \zeta(x) \) to get the posterior on ground and class height conditional on the neighbors. Samples are then drawn from these distributions.

The time complexity of HSMM calculations is greater than an HMM because of the sum over possible durations \( x^+ \), but the observation likelihood products \( \prod k P(Y^k_i | x) \) can be pre-computed and the state durations to search over can be constrained based on the priors to reduce the complexity to \( O(numVoxels \times numStates \times maxDuration) \) 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.

### 6. Learning

The model described in Section 4 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, 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.

#### 6.1. 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 Wellington and Stentz (2004) 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 Wellington and Stentz (2004), 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 $\mu_i$, $\sigma_i^2$, $P(i)$ in equation 1) and the variance of measurements within the voxels is used as the GMM measurement model variance ($\sigma_v^2$ 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 ($a_i$ and $b_i$ in equation 3) for each class state $x$ using a Newton–Raphson method (Smith 1983). 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.

### 6.2. 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 $\sigma_G$ of typical ground height variation between voxel columns, which gives us the maximum likelihood estimate of our Gaussian MRF ground neighborhood prior (Saquib et al. 1998).

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 $\sigma_H$ directly from this sequence of class heights.

The class interaction prior $\lambda_c$ gives the probability that a class transitions to a different class. This could be estimated directly using pseudo-likelihood methods (Li 2001) 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 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.

### 7. Run Time

Performing inference in this model is computationally intensive because of the repeated calculations necessary to sample from the model. We maintain a $150 \times 150$ grid of $15$ cm square voxel columns in our terrain map, which covers a $22.5$ m square area. Our system runs a loop that updates the local terrain map at approximately $1$ Hz. 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 $1$ m/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.

### 8. 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 obtain 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.
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 five 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 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.

9. 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 1 Hz, 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 1, 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.

9.1. 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.

9.2. Person in Tall Vegetation

Figure 10 shows a challenging scene containing a camouflage 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 small dirt mound is initially classified as an obstacle on top of the ground, but then inference in the model finds the more likely answer of the ground rising up in that area (see video in Extension 1). The area of low grass to the right of the person is also classified as ground. This area is in the shadow of the tall weeds in the foreground and, without direct observations, the system is unable to distinguish this low-lying grass from the most likely hypothesis of ground.

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.
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*.

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.

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 each column. The model ground estimates are fairly smooth and stay within approximately 20 cm 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
The previous section demonstrated how the model’s structural assumptions enabled improved ground height estimates. 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 8, 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.

### 9.3. Vegetation on a Slope

The previous section demonstrated how the model’s structural assumptions enabled improved ground height estimates. 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 8, 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.
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.

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.

9.5. 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 m in front of the vehicle as the vehicle moves forward. The lowest hit line shows that the first 70 m 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. 170 m). 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.

10. 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 dif-
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 1–2 m/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.

Appendix: Index to Multimedia Extensions

The multimedia extension page is found at http://www.ijrr.org.

Table of Multimedia Extensions

<table>
<thead>
<tr>
<th>Extension</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Video</td>
<td>Person in tall vegetation example showing sensor data and spatial model output</td>
</tr>
</tbody>
</table>

Acknowledgement

This work was supported by John Deere under contract 476169.

References


Macedo, J., Manduchi, R. and Matthies, L. 2000. Ladar-based discrimination of grass from obstacles for...


