

# Interacting Markov Random Fields for Simultaneous Terrain Modeling and Obstacle Detection

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

Aaron Courville  
Robotics Institute  
Carnegie Mellon University  
Pittsburgh, Pennsylvania 15213  
Email: aaronc@ri.cmu.edu

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

**Abstract**—Autonomous navigation in outdoor environments with vegetation is difficult because available sensors make very indirect measurements on quantities of interest such as the supporting ground height and the location of obstacles. We introduce a terrain model that includes spatial constraints on these quantities to exploit structure found in outdoor domains and use available sensor data more effectively. The model consists of a latent variable that establishes a prior that favors vegetation of a similar height, plus multiple Markov random fields that incorporate neighborhood interactions and impose a prior on smooth ground and class continuity. These Markov random fields interact through a hidden semi-Markov model that enforces a prior on the vertical structure of elements in the environment. The system runs in real-time and has been trained and tested using real data from an agricultural setting. Results show that exploiting the 3D structure inherent in outdoor domains significantly improves ground height estimates and obstacle detection accuracy.

## I. INTRODUCTION

Outdoor environments such as those encountered in agriculture, mining, and the exploration of hazardous environments are often viewed as being “unstructured”. This absence of structured features, such as road markers, straight walls and a flat ground plane has often been cited as one of the reasons navigating within 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 or crop. We can use the knowledge that the ground is generally smooth and the vegetation has approximately constant height to infer the ground height and allow 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.

Local autonomous navigation in outdoor environments is often performed in a model predictive control framework that searches over dynamically feasible control arcs for a safe trajectory [2]. In this framework, a terrain model with obstacles and the supporting surface 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 [4]. While faithful models of vehicle dynamics are often available, acquiring an



Fig. 1. Tractor test platform

accurate terrain model that includes a description of the load-bearing surface and any obstacles in the local environment is a considerable challenge.

Early work in terrain perception assumed smooth terrain with discrete obstacles [1] [2], and achieved good results in these domains by looking at the height of the sensor readings from individual grid cells. More recent work has included cluttered environments with vegetation that are much more difficult because the range points from sensors do not generally give the load-bearing surface. To handle these challenging domains, researchers have tried to model various parts of the problem. A common approach is to model how a range sensor penetrates vegetation to discriminate between vegetation and solid objects [3] [5] [6]. Another approach begins with a cloud of range points and looks for various features or structure at a local level [7] [8]. We have used online learning methods to automatically learn the ground height in vegetation from features [4]. A common characteristic among these approaches is that they make the strong assumption of independence between patches of terrain for estimating ground height or class. We hope to achieve better results by relaxing this independence assumption through the inclusion of spatial correlations.

Spatial correlations in images have often been expressed using Markov random fields [9] [10] and these techniques have been used for vision problems such as the segmentation of various land types in satellite images [11], but these approaches use only 2D image data instead of the 3D data that is generally available to an off-road robotic system.

In this paper, we describe a generative, probabilistic approach to modeling terrain. We exploit 3D spatial structure inherent in off-road domains and an array of noisy but abundant sensor data to jointly produce better estimates of the ground height and more accurate classification of obstacles and other areas of interest, even in dense non-penetrable vegetation.

Our terrain model consists of two distinct but interacting Markov random field models (MRFs) and a latent variable for common vegetation height. One MRF models ground height and enforces our assumption that ground height is smoothly varying. The second MRF encodes our assumption that class patches of space tend to cluster (for example, patches of vegetation of a single type tend to be found together). The latent variable for common vegetation height enforces our assumption that vegetation of the same type generally has a similar height. These three components interact through a hidden semi-Markov model (HSMM) that enforces vertical structural assumptions such as the understanding that vegetation grows on top of ground.

The structure in the terrain model is combined with information from multiple sensors on the vehicle using sensor models that are automatically learned from training data. Obstacles are treated as having uncertain attributes so obstacle appearance does not need to be explicitly trained.

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 allows for an improved estimate of the height of the underlying ground. Similarly, knowing the ground height helps disambiguate solid obstacles from the ground surface.

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 fully 3D MRF model, and allows our system to run in real-time.

## II. 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 [4]. The vehicle has a high-resolution stereo pair of digital cameras, an infrared camera, and two SICK laser range-finders (ladar) mounted on custom actively-controlled scanning mounts. The first scanning ladar is mounted on the roof to get range data over a large area in front of the vehicle, and the second scanning ladar is mounted on the bumper to get high density measurements of nearby terrain and better penetrate upcoming vegetation. The cameras and scanned ladars are precisely calibrated and tightly synchronized with an accurate global vehicle pose estimate.

The basic representational structure of our terrain model is the *voxel*: a  $15\text{cm}^3$  box-shaped region of 3 dimensional space. We represent the vehicle’s local spatial environment as a voxel lattice of size  $I \times J \times K$ , 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 ladar points corresponding to the same region of space to the same voxel. Exploiting the precise synchronization of the sensors, we project ladar points into the most recent color and infrared images, so that each ladar point results in a vector of appearance measurements for that voxel, including laser remission (reflectance), infrared temperature, and color.<sup>1</sup>

The voxel representation also allows us to maintain a density estimate throughout space by comparing how many ladar rays pass through each voxel (pass-throughs) with the number of ladar 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] [4].

## III. TERRAIN MODEL

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  $ij$ th 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.  $c \in \{\text{ground}, \text{vegetation}, \text{obstacle}\}$ .

Associated with the  $k$ th voxel in the  $ij$ th voxel column is the voxel state  $X_{ij}^k$ , a multinomial distributed random variable that describes the nature of the material inside the voxel,  $X_{ij}^k \in \{\text{ground}, c, \text{free-space}\}$ , where  $c$  is the class of the  $ij$ th voxel column.<sup>2</sup> The  $ijk$ th voxel is also associated with the observation vector  $\mathbf{Y}_{ij}^k = [Y_{den}, Y_{rem}, Y_{ir}, Y_{col}]$ , containing vectors of  $N$  ladar hit and pass-through density measurements of which the  $M$  hits include laser remission values, infrared temperatures and color data (i.e.  $Y_{den} = [Y_{den}^1, \dots, Y_{den}^N]$ ,  $Y_{rem} = [Y_{rem}^1, \dots, Y_{rem}^M]$ ).

### A. 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.

The measurement vector,  $\mathbf{Y}_{ij}^k$  contains a variable number of noisy measurements of the material properties. The graphical

<sup>1</sup>The ladar 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 ladar point. This coupled with the continual movement of the scanning ladars makes it unlikely that a single pixel is used more than once, so we treat each color and infrared tagged ladar point as an independent measurement.

<sup>2</sup>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.

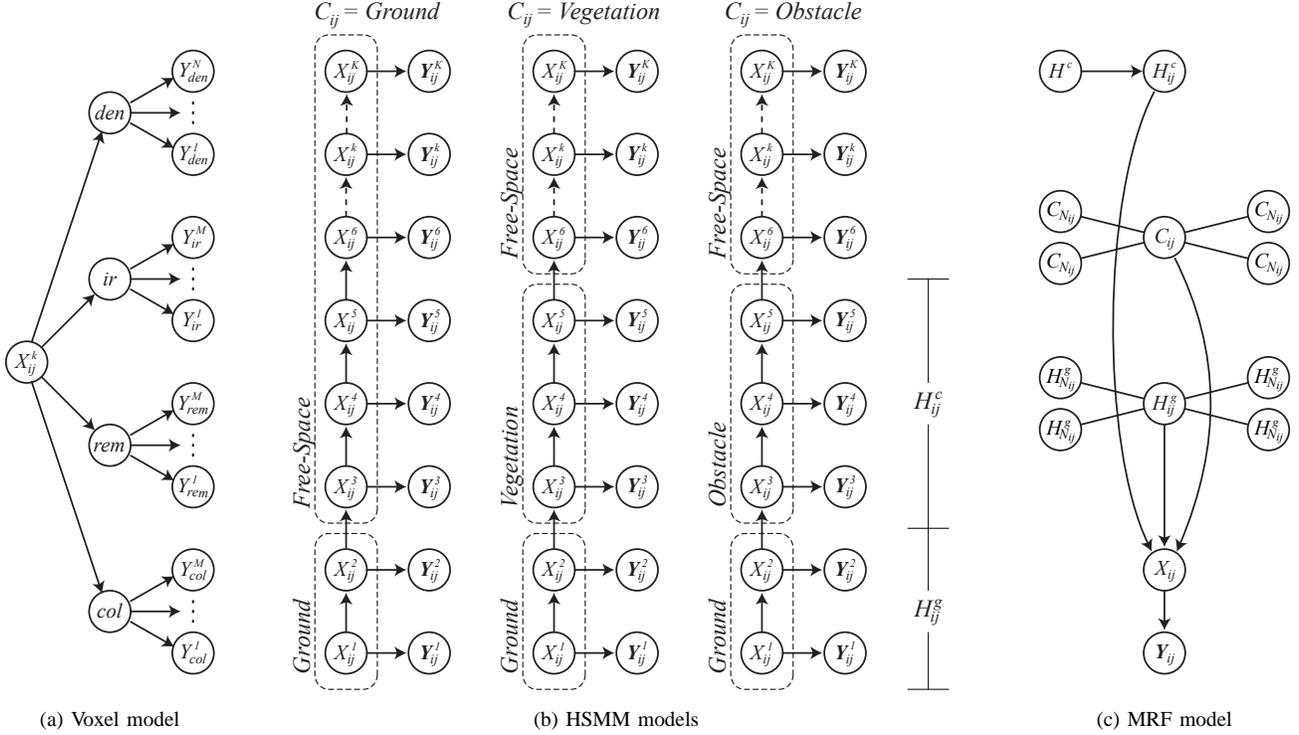


Fig. 2. A graphical description of the model showing (a) the voxel, (b) the voxel column, and (c) the connections between voxel columns. For each voxel column  $ij$ , the model contains voxel states  $X_{ij}^k$ , observations  $Y_{ij}^k$ , and a class  $C_{ij}$ , class height  $H_{ij}^c$ , and ground height  $H_{ij}^g$  that interact with neighbors  $N_{ij}$  and the common class height  $H^c$

model in Figure 2(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 material properties constitute nuisance variables that we remove from the observation models through marginalization.

Density values range from empty space ( $den = 0$ ) to completely solid ( $den = 1$ ), and we use a beta distribution  $B(a_x, b_x)$  to describe the density values of each state  $x$ . The measurements of density  $Y_{den}^n$  are binary (ladar hit or pass-through), so we use a binomial distribution to describe the number of hits  $M = \sum_{n=1}^N Y_{den}^n$  out of  $N$  total rays. When we integrate over the nuisance parameter  $den$ , we recover the beta-binomial distribution as the marginal likelihood observation model.

$$\begin{aligned}
 & P(M = m \mid X_{ij}^k = x) \\
 &= \int P(m \mid den) p(den \mid X_{ij}^k = x) d(den) \\
 &= \binom{N}{M} \frac{B(a_x + M, b_x + N - M)}{B(a_x, b_x)}
 \end{aligned} \quad (1)$$

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 mixture of Gaussians 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  $x$  is modeled as a mixture of Gaussians with individual mixture means  $rem_i$ , variances  $\sigma_i^2$ , and mixing coefficients  $P(i)$ .

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

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

$$\begin{aligned}
 & p(y_{rem} \mid X_{ij}^k = x) = \int p(y_{rem} \mid rem) p(rem \mid X_{ij}^k = x) d(rem) \\
 &= \sum_{i=1}^R P(i) \int \prod_{m=1}^M p(y_{rem}^m \mid rem) p(rem \mid rem_i) d(rem) \\
 &= \sum_{i=1}^R P(i) \frac{1}{\sqrt{2\pi\left(\sigma_i^2 + \frac{\sigma_y^2}{M}\right)}} \exp\left(-\frac{(\bar{y}_{rem} - rem_i)^2}{2\left(\sigma_i^2 + \frac{\sigma_y^2}{M}\right)}\right)
 \end{aligned} \quad (3)$$

Equation 3 shows that the marginal appearance distributions become more broad when there are few data points ( $M$  is small), reflecting the increased uncertainty in the material property and hence the state.

The *free-space* state does not possess any meaningful material properties beyond density  $den$ . Ladar 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*.

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. Mixture of Hidden Semi-Markov Chains of Voxel Columns

When moving 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 as states in a Markov chain which would induce a geometric distribution on the duration of states. We resort instead to a hidden semi-Markov model (HSMM) [12] over voxel states, which explicitly represents a state duration (or height distribution) over voxels for each state value.

As shown in figure 2(b), we associate a single HSMM chain structure with each column class  $C_{ij}$ , which makes 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  $H_{ij}^c$ .

### C. Markov Random Field Model of Interacting Voxel Columns

The HSMM column models capture the vertical structure between the states, but there are also significant horizontal dependencies between neighboring columns. As shown in Figure 2(c), we model these dependencies using two distinct but interacting Markov random fields (MRFs) [10] for class  $C_{ij}$  and ground height  $H_{ij}^g$ , each dependent on the values of their respective neighbors, 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 on the state durations associated with  $H_{ij}^c$  and  $H_{ij}^g$  and imposing a prior over HSMM 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

$$P(C_{ij} = c | C_{N_{ij}}) \propto \exp\left(-\lambda_C \sum_{\{s,t\} \in N_{ij}} (c \neq c_{st})\right) \quad (4)$$

where  $N_{ij}$  is the set of neighboring indices and  $C_{N_{ij}}$  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  $H_{ij}^g$  will be tightly correlated with nearby values. We express this belief using a Gaussian Markov random field

$$P(H_{ij}^g = h | H_{N_{ij}}^g) \propto \exp\left(-\frac{1}{2\sigma_G^2} \left(h - \frac{1}{|N_{ij}|} \sum_{\{s,t\} \in N_{ij}} h_{st}^g\right)^2\right) \quad (5)$$

where  $|N_{ij}|$  is the size of the neighborhood.

We 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

$$P(H_{ij}^c = h | H^c) \propto \exp\left(-\frac{1}{2\sigma_{H^c}^2} (h - H^c)^2\right) \quad (6)$$

## IV. INFERENCE

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

$$P(C_{ij}, H_{ij}^g, H_{ij}^c | Y_{ij}, C_{N_{ij}}, H_{N_{ij}}^g, H^c) \quad (7)$$

so we use Gibbs sampling [9] for approximate inference.

Algorithm 1 gives 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 2(c). Samples of the common class height are produced from its conditional distribution given the current column class height samples  $h_{ij}^c$

$$P(H^c = h | H_{ij \in IJ}^c) \propto \exp\left(\frac{-1}{2\sigma_{H^c}^2/D^c} \left(h - \frac{1}{D^c} \sum_{ij \in IJ, c_{ij}=c} h_{ij}^c\right)^2\right) \quad (8)$$

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  $H_{ij}^c$  and ground height  $H_{ij}^g$  from the neighbors, as given in equations 4 and 5, and the common class heights  $H^c$  as given in equation 6. The priors on  $H_{ij}^c$  and  $H_{ij}^g$  are then incorporated into the HSMM model as priors over state durations and are shown in the subsequent equations as  $P(H_{ij}^c = h | H^c)$  for the class state  $x = c$  or  $P(H_{ij}^g = h | H_{N_{ij}}^g)$  for the ground state  $x = g$ .

---

**Algorithm 1** Gibbs sampling from the model

---

Sample common class heights  $h^c$  from  $P(H^c | H_{ij \in IJ}^c)$  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:

$$P(H_{ij}^g | H_{N_{ij}}^g) \\ P(C_{ij} | C_{N_{ij}})$$

**for all** Classes  $c$  **do**

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

$$P(H_{ij}^c | H^c)$$

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

$$P(Y_{ij} | C_{ij} = c, H_{N_{ij}}^g, H^c) \\ P(H_{ij}^g | C_{ij} = c, Y_{ij}, H_{N_{ij}}^g, H^c) \\ P(H_{ij}^c | C_{ij} = c, Y_{ij}, H_{N_{ij}}^g, H^c)$$

**end for**

Compute class distribution:

$$P(C_{ij} | Y_{ij}, C_{N_{ij}}, H_{N_{ij}}^g, H^c) \\ \propto P(Y_{ij} | C_{ij}, H_{N_{ij}}^g, H^c) P(C_{ij} | C_{N_{ij}})$$

Sample  $c_{ij}$  from  $P(C_{ij} | Y_{ij}, C_{N_{ij}}, H_{N_{ij}}^g, H^c)$

Sample  $h_{ij}^g$  from  $P(H_{ij}^g | C_{ij} = c_{ij}, Y_{ij}, H_{N_{ij}}^g, H^c)$

Sample  $h_{ij}^c$  from  $P(H_{ij}^c | C_{ij} = c_{ij}, Y_{ij}, H_{N_{ij}}^g, H^c)$

**end for**

---

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 [12]. As shown in figure 2(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.

The forward-backward computations are still performed over the individual spatial steps  $X_{ij}^k$  as in an HMM, but an HSMM 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 9 and 10 give the HSMM forward and backward probabilities  $\alpha_{ij,c}^k$  and  $\beta_{ij,c}^k$  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.

$$\alpha_{ij,c}^k(x) = P(\text{state } x \text{ ends at } k, Y_{ij}^{1:k} | C_{ij} = c, H_{N_{ij}}^g, H^c) \\ = \sum_{x'} \sum_h P(X_{ij}^k = x, X_{ij}^{k-h} = x', H_{ij}^x = h, Y_{ij}^{1:k} | C_{ij}, H_{N_{ij}}^g, H^c) \\ = \sum_h \prod_{k'=k-h+1}^k P(Y_{ij}^{k'} | x) P(H_{ij}^x = h | H_{N_{ij}}^g, H^c) \alpha_{ij,c}^{k-h}(x^-) \quad (9)$$

$$\beta_{ij,c}^k(x) = P(Y_{ij}^{k+1:K} | \text{state } x \text{ ends at } k, C_{ij} = c, H_{N_{ij}}^g, H^c) \\ = \sum_{x'} \sum_h P(Y_{ij}^{k+1:K} | X_{ij}^k = x, X_{ij}^{k+h} = x', H_{ij}^{x^+} = h, C_{ij}, H_{N_{ij}}^g, H^c) \\ = \sum_h \prod_{k'=k+1}^{k+h} P(Y_{ij}^{k'} | x^+) P(H_{ij}^{x^+} = h | H_{N_{ij}}^g, H^c) \beta_{ij,c}^{k+h}(x^+) \quad (10)$$

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

$$P(Y_{ij} | C_{ij} = c, H_{N_{ij}}^g, H^c) = \alpha_{ij,c}^K(x = \text{free-space}) \quad (11)$$

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

Finding the distribution over state durations involves combining  $\alpha$  and  $\beta$ .

$$\zeta_{ij,c}^x(h) = P(\text{state } x \text{ has duration } h | Y_{ij}, C_{ij} = c, H_{N_{ij}}^g, H^c) \\ = \sum_k P(X_{ij}^k = x, X_{ij}^{k-h} = x^- | Y_{ij}, C_{ij}, H_{N_{ij}}^g, H^c) \\ = \sum_k \prod_{k'=k-h+1}^k P(Y_{ij}^{k'} | x) P(H_{ij}^x = h | H_{N_{ij}}^g, H^c) \alpha_{ij,c}^{k-h}(x^-) \beta_{ij,c}^k(x) \quad (12)$$

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

$$P(H_{ij}^g = h | C_{ij} = c, Y_{ij}, H_{N_{ij}}^g, H^c) = \zeta_{ij,c}^{x=\text{ground}}(h) \\ P(H_{ij}^c = h | C_{ij} = c, Y_{ij}, H_{N_{ij}}^g, H^c) = \zeta_{ij,c}^{x=\text{state } c}(h) \quad (13)$$

The time complexity of HSMM calculations is greater than an HMM because of the sum over possible durations, but the observation likelihood products can be pre-computed and the state durations to search over can be constrained based on the priors to reduce the complexity to  $O(\text{num Vowels} * \text{num States} * \text{maxDuration})$  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.

## V. LEARNING

The model described in section III 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 [4] 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 [4] 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.  $rem_i$ ,  $\sigma_i^2$ ,  $P(i)$  in equation 2) and the variance of measurements within the voxels is used as the GMM measurement model variance ( $\sigma_y^2$  in equation 3).

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 1) for each class state  $x$  using a Newton-Raphson method[13]. 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 4 and 5 describe how class and ground height depend on their neighbors, and the prior in equation 6 describes how column class heights are related to the common class height. Each of these priors contains a parameter that gives the strength of the prior, and describes how much classes tend to clump together, how smooth the ground is, and how little class heights vary. 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.

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  $\sigma_{Hc}$  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 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.

## VI. RESULTS

We have tested this model in a nearby working farm and an undeveloped area with tall weeds. The following three examples demonstrate the improved performance gained from using the structure we have built into our model.

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 observation likelihood products, computes samples from the model, and updates the local terrain map with the maximum *a posteriori* (MAP) class label, mean ground height, and mean class height from the samples in each column.

### A. White shed

Figure 3 shows the view from 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 4 shows the output of our model including the MAP class labels: obstacle (red), vegetation (green), ground (gray), and the mean ground height for driveable areas. Obstacle columns are shown at their class height. The model produces a reasonable 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 on a voxel basis the ground class is much more likely than the broad uniform obstacle class for the voxels from the shed. However, the MRF and HSMM spatial constraints imposed on the ground surface make 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 5 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. Figure 5 also shows that without the ground and class priors, the ground height estimates and classification labels are generally more noisy.



Fig. 3. View from the tractor of a white shed

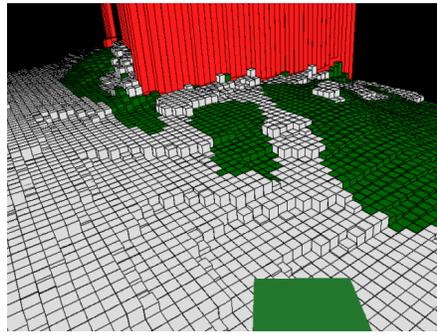


Fig. 4. System output, including ground heights and classification

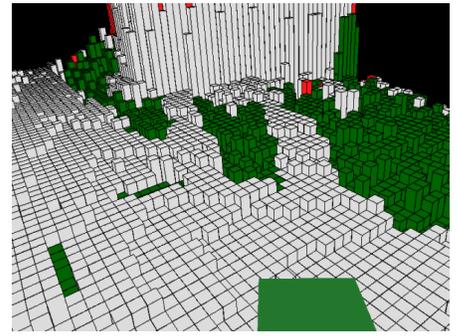


Fig. 5. Independent voxel column output with neighborhood interactions turned off, showing incorrect classification of the shed



Fig. 6. View from tractor of tall weeds, low grass, person, and small dirt mound

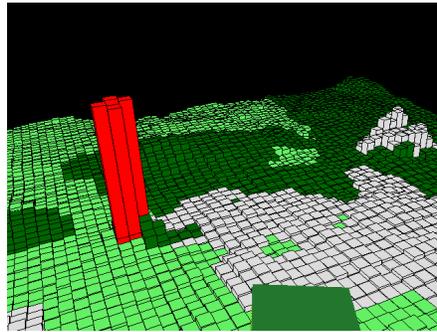


Fig. 7. System output, including ground heights and classification

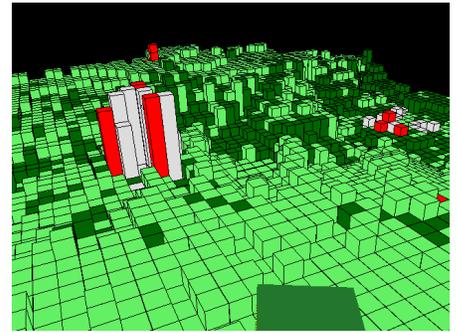


Fig. 8. Lowest hit or pass for ground height in vegetation and independent classification, showing poor ground height estimates and misclassifications

### B. Tall vegetation

Figure 6 shows the view from the tractor in a challenging scene: 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. We trained on the two types of vegetation and bare ground. Figure 7 gives the mean ground heights and MAP 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 using their neighborhood. The system can infer the common vegetation height of the tall weeds from areas where it is observable, such as under the vehicle or the transition to tall weeds behind the person. The assumption of vegetation height similarity then 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.

The range points do not penetrate the tall weeds in this example, as shown in Figure 8, which uses the lowest hit or pass-through in each vegetation column for ground height and the MAP class labels when no neighborhood information is used. Assuming independence prevents information from propagating and the resulting ground height estimates are poor. Also, both the person and the dirt mound contain a mixture of obstacle and ground classes.

Figure 9 shows a plot of the quality of the ground height estimates from Figures 7 and 8. 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 9, 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 20cm of the true value.

As another comparison, we show an approach that adjusts the lowest hit in each column based on that column’s independent classification. 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 9 shows that this can work well when the classification is correct and the actual vegetation height

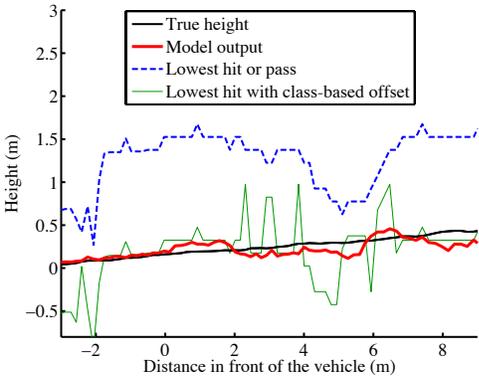


Fig. 9. Comparison between ground estimates of the area in Figures 7 & 8

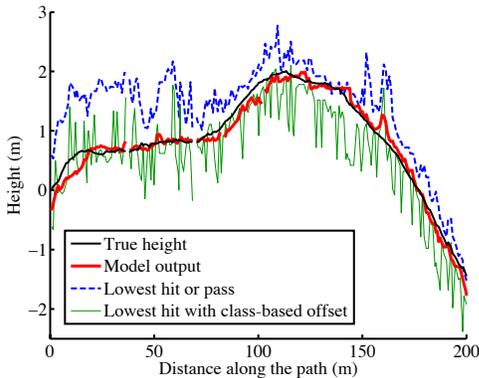


Fig. 10. Comparison between ground estimates for longer test path, showing predictions made 6m in front of the vehicle as the vehicle drove

matches the training data, but it suffers from misclassification and the lack of a smoothing prior.

### C. Longer run through varied vegetation

Figure 10 shows ground height estimates for a longer test set through varied vegetation. Unlike Figure 9, which presents a snapshot of the predictions at different distances in front of the vehicle at a given time, Figure 10 shows predictions 6m in front of the vehicle over time as the vehicle drove. The lowest hit line shows that the first 70m of the path contains two sections of tall dense non-penetrable vegetation, and the remainder of the path has 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 lowest hit with class offset is often correct but very noisy because of misclassifications due to its independence assumption.

## VII. CONCLUSIONS AND FUTURE WORK

We have described a novel model structure that allows multiple Markov random fields to interact through a hidden semi-Markov model for improved ground height estimation and classification for outdoor navigation. This structure enforces spatial constraints within a column and between

neighboring columns. The model provides a natural way of combining different types of sensor data. It can find obstacles without needing to explicitly model them or collect obstacle appearance training data. It can infer vegetation height to produce estimates of the supporting ground surface even when the ground is hidden by dense vegetation. Except for the class neighborhood prior, the sensor and interaction model parameters can be easily trained by simply driving through representative areas. The system runs in real-time on real data and we showed that including the neighborhood structure significantly improved both the ground height estimates and the obstacle classification over an equivalent model without neighborhood interactions.

We are currently working on several improvements to the model. The system is set up essentially as a batch process, even though data is continually coming in and the sampling procedure continues over time. We would like to make it a true online algorithm. We are experimenting with further class models to handle hanging obstacles and holes. Finally, we are looking into belief propagation and other approximate inference schemes that might be less computationally intensive than Gibbs sampling.

## ACKNOWLEDGEMENT

This work was supported by John Deere under contract 476169

## REFERENCES

- [1] M. Daily, J. Harris, D. Keirse, 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.
- [2] 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.
- [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 (AUUVSI 02)*, July 2002.
- [4] C. Wellington and A. Stentz, "Online adaptive rough-terrain navigation in vegetation," in *IEEE Int. Conf. on Robotics and Automation*, April 2004.
- [5] 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.
- [6] 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.
- [7] N. Vandapel, M. Hebert, D. Huber, and A. Akapur, "Natural terrain classification using 3-D ladar data," in *IEEE Int. Conf. on Robotics and Automation*, April 2004.
- [8] N. Vandapel and M. Hebert, "Finding organized structures in 3-d ladar data," in *IEEE Int. Conf. on Intelligent Robots and Systems*, 2004.
- [9] S. Geman and D. Geman, "Stochastic relaxation, gibbs distributions, and the bayesian restoration of images," *IEEE Trans. Pattern Anal. Machine Intell.*, 1984.
- [10] S. Z. Li, *Markov Random Field Modeling in Image Analysis*, 2nd ed., ser. Computer Science Workbench. Springer-Verlag, 2001.
- [11] 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.
- [12] 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.
- [13] 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.