Fast Probabilistic Labeling of City Maps

Ingmar Posner and Mark Cummins and Paul Newman
Mobile Robotics Group,
Dept. Engineering Science
Oxford University
Oxford, UK
Email: {hip, mjc, pnewman}@robots.ox.ac.uk

***Abstract*— This paper introduces a probabilistic, two-stage classification framework for the semantic annotation of urban maps as provided by a mobile robot. During the first stage, local scene properties are considered using a probabilistic bag-of-words classifier. The second stage incorporates contextual information across a given scene via a Markov Random Field (MRF). Our approach is driven by data from an onboard camera and 3D laser scanner and uses a combination of appearance-based and geometric features. By framing the classification exercise probabilistically we are able to execute an information-theoretic bail-out policy when evaluating appearance-based class-conditional likelihoods. This efficiency, combined with low order MRFs resulting from our two-stage approach, allows us to generate scene labels at speeds suitable for online deployment and use. We demonstrate and analyze the performance of our technique on data gathered over almost 17 km of track through a city.**

I. INTRODUCTION

This paper addresses the fast labeling of mobile robot workspaces using a camera and a 3D laser scanner. We motivate this work by noting that, although contemporary online mapping and simultaneous localization techniques using lidar now produce compelling 3D geometric representations (a.k.a maps) of a mobile robot's workspace, these maps tend to be geometrically rich but semantically impoverished. Our work seeks to redress this shortcoming. Maps in the form of large unstructured point clouds are meaningful to human observers, but are of limited operational use to a robot. There is much to be gained by having the robot itself upgrade the map to include richer semantic information and to do so online. In particular, the semantics induced by online segmentation and labeling has an important impact on the action selection problem. For example, the identification of terrain types with estimates of their spatial extent has a clear impact on control. Similarly the identification of buildings and their entrances has a central role to play in mission execution and planning in urban settings.

In this paper we outline a probabilistic method which achieves fast labeling of 3D point clouds by using a combination of appearance and geometric features. In particular we use combined 3D range and image data to perform inference at two distinct levels. Firstly, over local scales, classification is based on the co-occurrence of appearance descriptors, which capture both visual and surface orientation information. We frame this classification problem in probabilistic terms, which allows the implementation of a principled "bail-out" policy to be invoked when evaluating class conditional likelihoods, resulting in very large computational savings. Secondly, at the scene-wide scale, we use a Markov Random Field (MRF) to model the expected relationships between patch labels and to thus incorporate the rich prior information common to many parts of our man-made environment. Our MRFs have a relatively low node-count, just one node for each scene patch, yielding rapid inference.

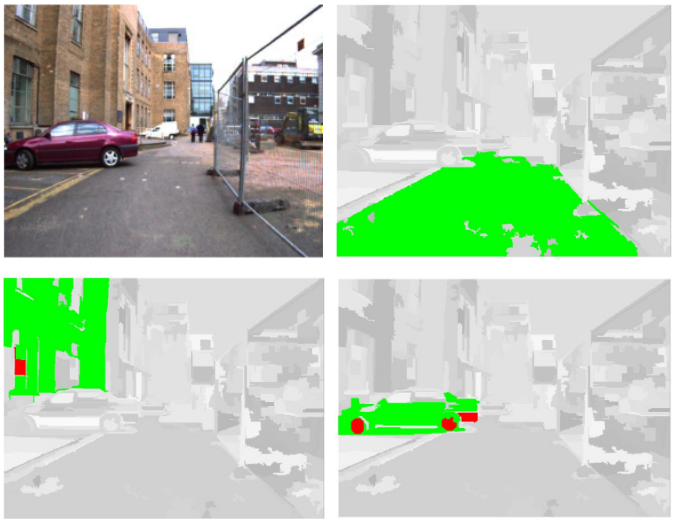


Fig. 1. Classification results for a typical urban scene: the original image **(top left)**; segments classified as 'pavement/tarmac' **(top right)**; segments classified as 'textured wall' **(bottom left)**; segments classified as 'vehicle' **(bottom right)**. The colour-coding is wrt. to ground-truth: green indicates a correct label; red indicates a false negative.

II. RELATED WORK

Recently there has been a surge in the literature regarding environment understanding within robotics, particularly as available sensory data becomes richer and the limitations of unannotated maps become more apparent. A variety of machine learning approaches to the problem have been explored, with more recent approaches utilizing contextual as well as local information to improve classification performance. In [1] the authors classify 2D laser data into types of indoor scenes using boosting. Contextual information was used explicitly in [2] by way of a model based on relational Markov networks to learn classifiers from segment-based representations of indoor workspaces. More recently [3] introduced an approach which takes into account spatial relationships between objects and object parts in 3D. 3D laser data were used in [4], where they were segmented to detect cars and classify terrain using Graph Cut applied to a Markov Random Field (MRF) formulation of the problem, an approach which was extended by [5].

Particularly relevant to the work presented here are papers which consider a combination of vision and laser data in an outdoor setting. [6] considers the task of pedestrian- and vehicle detection, using 2D laser data. In [7] a more sophisticated inference framework based on Conditional Random Fields was brought to bear on the vehicle detection problem, with preliminary results also reported for multi-class labelling. 3D laser data were combined with visual information in [8], which used support vector machines for classification but did not make use of contextual information.

The work presented here also leverages a combination of laser data with vision. Our main contribution lies in the definition of an efficient contextual inference framework, based on a graph over plane patches rather than over measurements (e.g. laser range data) directly. This yields substantial speed increases over previous approaches. As an integral part of this framework we further define a generative bag-of-words classifier and describe an efficient inference procedure for it. Finally, the work presented here further distinguishes itself from related work by combining information from two complimentary sensors – full 3D geometry and appearance. Thereby our approach gains the capacity of providing *more detailed* workspace descriptions such as the surface-type of building(s) encountered or the nature of ground traversed.

III. Classes And Features

The system described in this paper utilizes data from a calibrated combination of 3D laser scanner and monocular camera, both mounted on a mobile robot. Our basic processing pipeline is similar to that described in [8] – the major contribution of this paper is to extend the inference machinery. Briefly, incoming 3D laser data are segmented into local plane patches using a RANSAC procedure (see Figure 2). Plane patches are then sub-segmented into visually homogeneous areas using an off-the-shelf image segmentation algorithm [9]. The product of this feature extraction pipeline is a set of visually similar image patches which have 3D geometry attributes associated with them. Our classification framework proceeds by classifying each patch individually. The final stage then consideres scene-wide interactions between these local patches.

In contrast to much of the existing work in the area, we consider a relatively rich set of seven classes in three categories. Classes are listed in Table I, and comprise ground types, building types and two object categories. Labeling the environment into classes such as these is a useful step towards a number of autonomous tasks such as path following, location recognition and collision avoidance.

TABLE I
Classes

Class	Description
Ground Type	
Pavement/Tarmac	Road, footpath.
Dirt Path	Mud, sand, gravel.
Grass	Grass.
Building Type	
Smooth Wall	Concrete, plaster, glass.
Textured Wall	Brickwork, stone.
Object	
Foliage	Bushes, tree canopy.
Vehicle	Car, van.

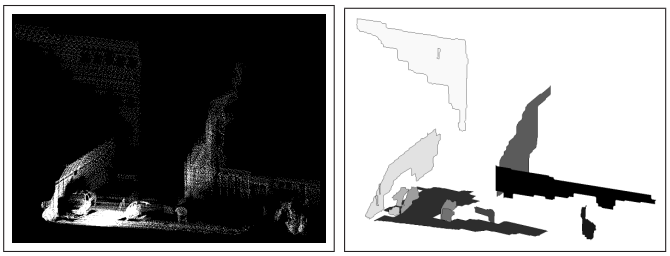


Fig. 2. An original 3D laser scan **(left)** and its approximation by planar patches as generated by the segmentation algorithm **(right)**.

Classification is performed on the basis of the features listed in Table II. These features are computed for all laser points in a patch, proivded that the points are visible in the camera image. Colour and texture features are computed over the 15x15 pixel local neighbourhood of each projected laser point.

IV. Generative Probabilistic Classification

The inference framework proposed in this paper is a multi-level approach based on successive combinations of lower-level features. At the lowest level, individual laser points are mapped to appearance-words based on the set of features described in Section III. The next level of the hierarchy pools information from multiple laser points by grouping them into patches based on boundaries in both the image and the point cloud. Each patch is then assigned a pdf over class membership by a bag of words classifier.

The highest level of the hierarchy takes account of spatial context by using an MRF defined over the set of patches. This improves local decisions by incorporating information from the gross geometric arrangement of classes in the scene.

A. Level 1 - Classification of Individual Laser Points

The lowest level input to our system is the collection of laser points in the scene. Each laser point is described by a feature vector, using the features described in Section III. Rather than deal with raw data directly, we adopt the *bag-of-words* representation [10], where the feature vectors are quantized with respect to a "vocabulary". The vocabulary is constructed by clustering all the feature vectors from a set of training data, using an incremental clustering algorithm. This yields a vocabulary of size $|v|$, the vocabulary size being determined by a user-specified threshold. The cluster centres then define the vocabulary. When the system has been trained,

TABLE II
Features used for classification

Feature Descriptions	Dimensions
3D Geometry Orientation of surface normal of local plane	1
2D Geometry Location in image: mean of normalised x and y	2
Colour HSV: hue & sat. histograms in local neighbourhood	30
Texture HSV: hue & sat. variance in local neighbourhood	2

incoming sensory data is mapped to the approximate nearest cluster centre using a kd-tree. Each patch is then described by a bag-of-words, which is the input to the next level of the system.

B. Level 2 - Patch-level Classifier

Our patch-level classifier is inspired by the probabilistic appearance model introduced in [11] and the theory presented below is an extension of that work into a more general classification framework. Building on the output of the lower-level vector quantization step, an observation of a patch $\mathbf{z} = \{z_1, \ldots, z_{|v|}\}$ is a collection of binary variables where each z_i indicates the presence (or absence) of the i^{th} word of the vocabulary within the patch. We would like to compute $p(\mathcal{C}|\mathbf{z})$, the distribution over the class labels given the observation, which can be computed according to Bayes rule:

$$p(\mathcal{C}^k|\mathbf{z}) = \frac{p(\mathbf{z}|\mathcal{C}^k)p(\mathcal{C}^k)}{p(\mathbf{z})} \tag{1}$$

where $p(\mathbf{z}|\mathcal{C}^k)$ is the class-conditional observation likelihood, $p(\mathcal{C}^k)$ is the class prior and $p(\mathbf{z})$ normalizes the distribution.

C. Representing Classes

Given a vocabulary, individual classes are represented within the classification framework by a set of class-specific examples, which we call exemplars. Concretely, for each class k the model consists of n_k exemplars $\mathcal{C}^k = \{C_1^k, \ldots, C_{n_k}^k\}$ where C_i^k is the i^{th} exemplar of class k. Exemplars themselves are defined in terms of a hidden "existence" variable e, each exemplar C_i^k being described by the set $\{p(e_1|C_i^k), \ldots, p(e_{|v|}|C_i^k)\}$. The term e_j is the event that a patch contains a property or artifact which, given a perfect sensor, would cause an observation of word z_j. However, we do not assume a perfect sensor — observations z are related to existence e via a sensor model which is specified by

$$\mathcal{D}: \quad \begin{cases} p(z_j = 1|e_j = 0), & \text{false positive probability.} \\ p(z_j = 0|e_j = 1), & \text{false negative probability.} \end{cases} \tag{2}$$

with these values being a user-specified input. The reasons for introducing this extra layer of hidden variables, rather than modeling the exemplars as a density over observations directly, are twofold. Firstly, it provides a natural framework to incorporate data from multiple sensors, where each sensor has different (and possibly time-varying) error characteristics. Secondly, as outlined in the following section, it allows the calculation of $p(\mathbf{z}|\mathcal{C}^k)$ to blend local patch-level evidence with a global model of word co-occurrence.

D. Estimating the Observation Likelihood

The key step in computing the pdf over class labels as per Equation 1 is the evaluation of the conditional likelihood $p(\mathbf{z}|\mathcal{C}^k)$. This can be expanded as an integration across all the exemplars that are members of class k:

$$p(\mathbf{z}|\mathcal{C}^k) = \sum_{i=1}^{n_k} p(\mathbf{z}|C_i^k, \mathcal{C}^k)p(C_i^k|\mathcal{C}^k) \tag{3}$$

where $\mathcal{C}^k$ is the class k, and C_i^k is an exemplar of the class. Given $p(\mathcal{C}^k|C_i^k) = 1$ (an assumption that none of the training data is mislabeled) and $p(C_i^k|\mathcal{C}^k) = \frac{1}{n_k}$ (all exemplars within a class are equally likely), this becomes

$$p(\mathbf{z}|\mathcal{C}^k) = \frac{1}{n_k} \sum_{i=1}^{n_k} p(\mathbf{z}|C_i^k) \tag{4}$$

The likelihood with respect to the exemplar can now be expanded as:

$$p(\mathbf{z}|C_i^k) = p(z_1|z_2, ..., z_n, C_i^k)p(z_2|z_3, ..., z_n, C_i^k)...p(z_n|C_i^k) \tag{5}$$

This expression cannot be tractably computed — it is infeasible to learn the high-order conditional dependencies between appearance words. We thus seek to approximate this expression by a simplified form which can be tractably computed and learned for available data. A popular choice in this situation is to make a Naive Bayes assumption — treating all variables z as independent. However, visual words tend to be far from independent, and it has been shown in similar contexts that learning a better approximation to their true distribution substantially improves performance [11]. The learning scheme we employ is the Chow Liu tree, which locates a tree-structured Bayesian network that approximates the true distribution [12]. Chow Liu trees are optimal within the class of tree-structured approximations, in the sense that they minimize the KL divergence between the approximate and true distributions. Because the approximation is tree-structured, its evaluation involves only first-order conditionals, which can be reliably estimated from practical quantities of training data. Additionally, Chow Liu trees have a simple learning algorithm that consists of computing a maximum spanning tree over the graph of pairwise mutual information between variables — this readily scales to very large numbers of variables.

We use the Chow Liu tree to model the fact that certain combinations of visual words tend to co-occur. It can be learnt from unlabeled training data across all classes, and approximates the distribution $p(\mathbf{z})$. To compute $p(\mathbf{z}|\mathcal{C}^k)$, the class-specific density, we find an expression that combines this global occurrence information with the class model outlined in section IV-C. Returning to Equation 5 and employing the Chow Liu approximation, we have

$$\begin{aligned} p(\mathbf{z}|C_i^k) &= p(z_1|z_2, .., z_n, C_i^k)p(z_2|z_3, .., z_n, C_i^k)..p(z_n|C_i^k) \\ &\approx p(z_r|C_i^k) \prod_{q=1}^{|v|} p(z_q|z_{p_q}, C_i^k) \end{aligned} \tag{6}$$

where z_r is the root of the Chow Liu tree and z_{p_q} is the parent of z_q in the tree. Each term in Equation 6 can be further expanded as an integration over the state of the hidden variables in the exemplar appearance model, yielding

$$p(z_q|z_{p_q}, C_i^k) = \sum_{s_{e_q} \in \{0,1\}} p(z_q|e_q = s_{e_q}, z_{p_q}, C_i^k)p(e_q = s_{e_q}|z_{p_q}, C_i^k) \tag{7}$$

which, assuming that sensor errors are independent of class and making the approximation $p(e_j|z_j) = p(e_j) \forall i \neq j$

becomes

$$p(z_q|z_{p_q}, C_i^k) = \sum_{s_{e_q} \in \{0,1\}} p(z_q|e_q = s_{e_q}, z_{p_q}) p(e_q = s_{e_q}|C_i^k) \tag{8}$$

further manipulation yields an expansion of the first term in the summation as

$$p(z_q = s_{z_q}|e_q = s_{e_q}, z_p = s_{z_p}) = \frac{a}{a+b} \tag{9}$$

where $s_{z_q}, s_{e_q}, s_{z_p} \in \{0, 1\}$ and

$$a = p(z_q = \overline{s_{z_q}}) p(z_q = s_{z_q}|e_q = s_{e_q}) p(z_q = s_{z_q}|z_p = s_{z_p})$$

$$b = p(z_q = s_{z_q}) p(z_q = \overline{s_{z_q}}|e_q = s_{e_q}) p(z_q = \overline{s_{z_q}}|z_p = s_{z_p})$$

which is now expressed entirely in terms of the known detector model and marginal and conditional observation probabilities. These can be estimated from training data. Thus we have a procedure for computing $p(\mathbf{z}|\mathcal{C}^k)$.

Returning to Equation 1, the prior $p(\mathcal{C}^k)$ can be learned simply from labeled training data, $p(\mathbf{z}|\mathcal{C}^k)$ we have discussed above, and to normalize the distribution we make the naive assumption that our set of classes fully partitions the world. Clearly this work would benefit from a background class, a change we plan to make in future versions of the system. The posterior distribution across classes, $p(\mathcal{C}^k|\mathbf{z})$, can now be computed for each patch.

E. Learning A Class Model

The final issue to address in relation to the patch-level classifier is the procedure for learning the class models described in section IV-C. Class models consist of a list of exemplars obtained from ground-truth (i.e. labeled) data. The term $p(e_q = 1|C_i^k)$ represents the probability that exemplar i of class k contained word q (this is a probability because our detector has false positives and false negatives). Given an observation labeled as this class, the properties of the exemplar can be estimated via

$$p(e_q = 1|C_i^k, \mathbf{z}) = \frac{p(\mathbf{z}|e_q = 1, C_i^k) p(e_q = 1|C_i^k)}{p(\mathbf{z}|C_i^k)} \tag{10}$$

where $p(\mathbf{z}|C_i^k)$ can be evaluated as described in the previous section and the prior term $p(e_q = 1|C_i^k)$ we initialize to the global marginal $p(e_q = 1)$.

F. Approximation Using Bounds

Computing the posterior over classes, $p(\mathcal{C}^k|\mathbf{z})$, requires an evaluation of the likelihood $p(\mathbf{z}|\mathcal{C}_j^k)$ for each of the exemplars in the training set. As the number of exemplars grows, this rapidly becomes the limiting computational cost of the inference procedure. This section outlines a principled approximation that accelerates this computation by more than an order of magnitude. The key observation is that while the posterior over classes depends on the summation over all exemplars (as per Equation 4), typically the value of the summation is dominated by a small number of exemplars, with the rest providing negligible contribution. By evaluating the exemplar likelihoods in parallel, those with negligible contribution can be identified and excluded before the computation is fully complete. This is a kind of preemption test, similar to procedures which have been outlined in other domains [13].

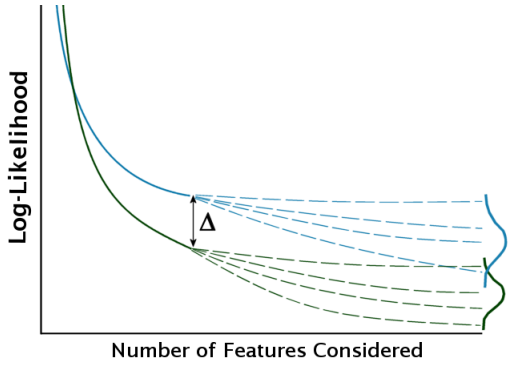


Fig. 3. Conceptual illustration of the bail-out test. After considering the first j words, the difference in log-likelihoods between two exemplars is Δ. Given some statistics about the remaining words, it is possible to compute a bound on the probability that the evaluation of the remaining words will cause one exemplar to overtake the other. If this probability is sufficiently small, the trailing exemplar can be discarded.

Recalling Equation 6, the log-likelihood of the current observation having been generated by exemplar i is given by

$$\ln(p(\mathbf{z}|C_i^k)) \approx \sum_{q=1}^{|v|} \ln(p(z_q|z_{p_q}, C_i^k)) \tag{11}$$

Now, define

$$d_q^i = \ln(p(z_q|z_{p_q}, C_i^k)) \tag{12}$$

and

$$D_j^i = \sum_{q=1}^{j} d_q^i = \sum_{q=1}^{j} \ln(p(z_q|z_{p_q}, C_i^k)) \tag{13}$$

where d_q^i is the log-likelihood of the i^{th} exemplar given word q, and D_j^i is the log-likelihood of the i^{th} exemplar after considering the first j words. At each step of the accelerated computation D_j^i is computed for all i, and incrementally increased j - that is, we are computing the log likelihoods of all exemplars in parallel, considering a greater proportion of the words at each step. After each step, a bail-out test is applied. This identifies and excludes from further computation those exemplars whose likelihood is *too far* behind the current leader. *Too far* can be quantified using concentration inequalities [14], which yield a bound on the probability that the discarded exemplar will catch up with the leader, given their current difference in log-likelihoods and some statistics about the properties of the words which remain to be evaluated.

Concretely, consider two exemplars a and b, whose log likelihood has been computed under the first j words, and whose current difference in log-likelihoods is Δ, as shown in Figure 3. Now, let X_j be the relative change in log likelihoods due to the evaluation of the j^{th} word, and define

$$S_j = \sum_{q=j+1}^{|v|} X_q \tag{14}$$

so that S_j is that total relative change in log likelihoods due to all the words that remain to be evaluated. We are

interested in $p(S_j > \Delta)$ – the probability that the evaluation of the remaining words will cause the trailing exemplar to *catch up*. If the probability is sufficiently small, the trailing hypothesis can be discarded. The key to our bail-out test is that a bound on the probability $p(S_j > \Delta)$ can be computed quickly, using concentration inequalities such as the Hoeffding or Bennett inequality [15]. These concentration inequalities are essentially specialized central limit theorems, bounding the form of the distribution S_j, given the statistics of the components X_j (which we can think of as distributions before their exact value has been computed). For the Hoeffding inequality, it is sufficient to know $\max(X_j)$ for each j, that is, the maximum relative change in log likelihood between any two exemplars due to the j^{th} word. We can compute this statistic quickly - it is simply the difference in log likelihoods between the exemplars with highest and lowest probability of having generated word j, which we can keep track of with some simple book-keeping. Bennett's inequality additionally requires a bound on the variance of X_j, which can also be cheaply computed.

Applying the Bennett inequality, the form of the bound is

$$p(S > \Delta) < \exp\left(\frac{\sigma^2}{M^2}\cosh(f(\Delta)) - 1 - \frac{\Delta M}{\sigma^2} f(\Delta)\right) \tag{15}$$

where

$$f(\Delta) = \sinh^{-1}\left(\frac{\Delta M}{\sigma^2}\right) \tag{16}$$

and M and v are the maximum and variance values of the remaining features, such that

$$p\left(|X_q| < M\right) = 1, \forall q \in [j+1, |v|] \tag{17}$$

$$\sum_{q=j+1}^{|v|} E\left[X_q^2\right] < \sigma^2 \tag{18}$$

Typically we set our bail-out threshold $p(S > \Delta) < 10^{-6}$. The speed increase due to this bail-out test is data dependent — in our experiments it is typically a factor of 60 times faster than performing the full classification without bail-out test.

V. Markov Random Fields For Spatial Context

The estimation of the set of most likely values of a set of interdependent random variables from available data is a standard machine learning problem. Such context-dependent inference can be achieved using a family of graphical models known as Markov Random Fields (MRFs). An MRF models the joint probability distribution, $p(\mathbf{x}, \mathcal{Z})$, over the (hidden) states of the random variables, $\mathbf{x}$ and the available data, $\mathcal{Z}$. For pairwise MRFs, it is well known that this joint probability can be maximised by equivalently minimising an energy function incorporating a *unary* term modelling the data likelihood for each node and a *binary* term specifying the interaction potentials between neighbouring nodes over the set of possible values [16]. Under the assumption of every datum being equally likely (i.e. $p(\mathcal{Z})$ being uniform) a minimisation of this energy function is equivalent to finding the most likely configuration of labels given the observed data - i.e. a maximum a posteriori (MAP) estimate of $p(\mathbf{x}|\mathcal{Z})$. In the following we describe how an MRF can be applied in the context of our scene labelling endeavour. In particular, we outline how the model structure of an MRF is derived for each scene from the available data, how the model parameters are obtained and, finally, how a MAP estimate over $p(\mathbf{x}|\mathcal{Z})$ is achieved.

A. Model Structure

MRFs are a family of graphical models where the set of interdependent variables is modelled as a graph $G(\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ denotes the set of vertices and $\mathcal{E}$ denotes the set of edges connecting neighbouring nodes, respectively. In the context of our scene labelling problem, each vertex represents a patch as introduced in Section IV. Neighbourhood relations within each scene are established using the segmented image obtained in Section III using [9]. Of course, adjacency in an image implies, but does not guarantee, adjacency in the 3D scene. Therefore, in estimating adjacency from 2D information a trade-off is made between the ability of determining neighbourhood relations efficiently and the introduction of incorrect adjacencies due to the loss of depth information. In practice, we found the number of false adjacencies introduced by this approach to be negligible. Typical examples of graph structure extracted from scenes recorded by our mobile platform are shown in Figure 4.

It should be noted that the one-to-one correspondence between vertices and image patches implies that the number of nodes in the MRF for a particular frame is independent of the number of measurements taken of the scene. Thus, the abstraction away from individual measurements (e.g. laser range data) to the patch level decouples the complexity of our inference stage from the density of the underlying data. This provides a substantial advantage in terms of speed over related works [7, 4] where the complexity of the graphical models is directly proportional to the density of the underlying data.

B. Model Parameters

The specification of an energy function to be optimised provides a convenient and intuitive way of incorporating scene properties. Consider the set of labels, $\mathbf{x} \in \mathbb{Z}^{N_n}$, for a particular configuration of a graph with N_n nodes. Each node s has an observation vector, $\mathbf{z}^s$, associated with it (c.f. Section IV) and can be assigned one of N_c labels such that $x_s \in \{1, \ldots, N_c\}$. We specify the energy of any such configuration to be given by

$$E(\mathbf{x}|\theta, \lambda) = \lambda \sum_{s \in \mathcal{V}} \theta_s(x_s) + (1-\lambda) \sum_{(s,t) \in \mathcal{E}} \theta_{st}(x_s, x_t) \tag{19}$$

where we adopt the notation of [17] in that θ defines the parameters of the energy: $\theta_s(\cdot)$ is a unary data penalty function; and $\theta_{st}(\cdot)$ is a pairwise interaction potential. λ represents a trade-off parameter which will be explained shortly. θ_s specifies the cost of assigning a given vertex any of the available labels. Intuitively, for a given node s, θ_s can be specified as a function of the posterior distribution over all classes for that node given the associated data, $p(\mathcal{C}|\mathbf{z}^s)$, as

Fig. 4. Typical graphs extracted from urban scenes as recorded by our mobile robot. **Top:** the original scenes. **Bottom:** the corresponding segmented images with the extracted graph overlaid. Circles indicate nodes, lines indicate edges. For images patches which are not marked as nodes no reliable geometry estimates could be extracted from the laser data.

provided by the patch classifier introduced in Section IV. In particular, the penalty of assigning label k to node s can be expressed as

$$\theta_s(x_{sk}) = 1 - p(\mathcal{C}^k|\mathbf{z}^s) \tag{20}$$

The complement of $p(\mathcal{C}^k|\mathbf{z}^s)$ is used since θ_s refers to a penalty function which is to be minimised.

The pairwise potential θ_{st} encodes prior domain information in the form of penalties incurred by assigning specific labels to adjacent (i.e. connected) nodes. This is an intuitive formulation of the preference that nodes of certain labels are more likely to be connected to nodes of certain other labels. It follows that θ_{st} can be specified in terms of a square-symmetric matrix $\mathbf{\Phi}$ of size $N_c \times N_c$ such that

$$\theta_{st}(x_i, x_j) = 1 - \phi_{i,j} \tag{21}$$

where again the complement is used since a penalty function is specified. In this work we have chosen to specify $\mathbf{\Phi}$ such that, for two classes i and j,

$$\phi_{\mathbf{i,j}} = \frac{L_{i,j}}{L_i + L_j - Li, j} \tag{22}$$

Here $L_{i,j}$ denotes the total number of links connecting nodes of labels i and j, and L_i denotes the total number of links originating from nodes of label i. It follows that $\phi_{i,j} \leq 1 \; \forall(i,j)$. Appropriate values for both $L_{i,j}$ and L_i are obtained from a hand-labelled training set.

Finally, Equation 19 is a function of the trade-off parameter, λ, which provides control over the relative contributions of the unary and the binary terms to the overall energy. It is specified such that $\lambda \in [0, 1]$. In this work λ is obtained by grid-search which selects a value that optimizes a measure of classifier performance on a set of labeled data. MAP estimation is performed using sequential tree-reweighted message passing (TRW-S) [17] because of its desirable convergence properties and speed.

VI. Results

We tested our algorithm using two extensive outdoor data sets spanning nearly 17 km of track gathered with an ATRV mobile platform. The system was equipped with a colour camera mounted on a pan-tilt unit and a custom-made 3D laser scanner consisting of a standard 2D SICK laser range finder (75 Hz, 180 range measurements per scan) mounted in a reciprocating cradle driven by a constant velocity motor. The camera records images to the left, the right and the front of the robot in a pre-defined pan-cycle triggered by vehicle odometry at 1.5 m intervals. The *Jericho* data set was recorded in a built-up area in Oxford over 13.2 km of track (16,000 images in total). The *Oxford Science Park* data set was recorded in the science park area in Oxford over 3.3 km of track (8,536 images in total). The two datasets were collected in different areas of the city, with only a very small overlap between the two regions.

The *Jericho* data set was used for training. The features from this set were used to learn the visual vocabulary and the Chow Liu tree. The class models were built from 1,055 patches which were segmented and labeled by hand. Automatically segmented versions of the same labeled data were used to learn the MRF binary potentials. An appropriate value for the sensor model used by our patch-level classifier was determined empirically as $p(z_i = 1|e_i = 1) = 0.35$ and $p(z_i = 0|e_i = 1) = 0$.

The *Jericho* data set is unsuitable for training the parameter λ since the patch-level classifier will correctly classify all patches in the training set, thus placing complete confidence in the unary potentials and leading to biased results. Therefore, λ was instead determined using an independent training set

obtained by sampling randomly from the *Oxford Science Park* data. The sample comprised a quarter of the entire data set (55 of 220 frames). The parameter value was then determined by grid search over its range. Different values of λ lead to different classification results, thus to select a value we must define a measure of classifier performance which we wish to optimize. We present results for two different such 'tuning policies':

Tuning Policy 1. Define a per-class error function as

$$\mathbf{e} = 1 - \mathbf{p} \bullet \mathbf{r} \tag{23}$$

where $\mathbf{p}$ is the vector of class precision values, $\mathbf{r}$ is the vector of class recall values and $\bullet$ denotes the Hadamard product. Thus, classes with a low precision-recall product will have a large error. Tuning policy 1 selects λ so as to minimize $\|\mathbf{e}\|_2$. The intention here is to maximize the precision-recall product, with a bias toward improving the worst performing classes.

Tuning Policy 2. Maximize the number of true positives across all classes.

We evaluated the performance of the classifier using 3,938 patches from the *Oxford Science Park* data set, which were not involved in training λ and whose ground truth had been labeled by hand. Classification performance is summarized in Figure 5 and in Table III. A typical example is shown in Figure 1.

We present three sets of results, with confusion matrices visualized in Figure 5. 5(a) is based entirely on the output of the patch-level classifier, showing performance before MRF smoothing is applied. 5(b) shows the results incorporating the MRF tuned according to policy 1, and 5(c) the results from MRF policy 2. Prior to incorporating the MRF (5(a)), there is notable confusion between the *vehicle, foliage* and wall classes. Results incorporating the MRF (5(b),5(c)) show a visible improvement of the confusion matrix. Particularly noteworthy is improvement on the *vehicle* and *foliage* classes, where confusion with wall classes has been substantially reduced. The remaining confusion is primarily between closely related classes such as the two wall types.

Numerical measures of performance are presented in Table III. It should be noted that our test data is *unbalanced,* in the sense that there are many more instances of some classes than others, reflecting their relative frequency in the world. A consequence of this is that performance figures such as overall accuracy are not very informative, because they mostly represent classifier performance on the largest class. We chose not to balance the data because such an evaluation would be unrepresentative of classifier performance in the real world. We quote instead the per-class precision and recall. $F_{0.5}$ measures are also provided in order to provide a measure of overall classification performance per class for all policies.

The timing properties of our algorithm are outlined in Table IV. Run times are from a 2Ghz Pentium laptop. The mean total processing time was 3.9 seconds, which compares favourable to similar systems such as [7], where the authors quote 7 seconds to classify a single 2D laser scan.

TABLE IV

TIMING INFORMATION (IN MILLISECONDS).

Process	Mean (ms)	Max (ms)
Plane Segmentation	2000	2800
Feature Extraction	89	125
Feature Quantization	4	90
Image Segmentation	960	1130
Patch Classification	850	3480
MRF	2	9
Overall	3.9 seconds	7.6 seconds

VII. CONCLUSIONS

This paper has described and provided a detailed analysis of a two-stage approach to fast region labeling in 3D point-cloud maps of cities. The contributions of this work are two-fold: the first stage classifier is framed using a probabilistic bag-of-words approach, which provides for a principled bail out policy that greatly decreases the computational cost of evaluating likelihood terms. Further contribution lies in an efficient formulation of the MRF to integrate contextual information. In contrast to related approaches, the size of graph we use is small — indeed with just one node per region rather than one per laser range measurment. As a result, the overall per-scene compute time of this method is compelling: at 3.9 seconds (on average 5.6 times faster than our previous support-vector machine based approach [8]) it is suitable for online deployment.

The approach presented in this paper further provides several attractive features above and beyond our own previous work: the probabilistic nature of this approach enables a principled extraction of confidence estimates for classification results; the sensor model provides a mechanism to incorporate the notion that some of the robot's observations are more trustworthy than others; and finally, the class models can readily be updated online, allowing, in principle, for lifelong learning.

VIII. ACKNOWLEDGEMENTS

The authors would like to thank M. Pawan Kumar for many insightful conversations. The work reported here was funded by the Systems Engineering for Autonomous Systems (SEAS) Defence Technology Centre established by the UK Ministry of Defence.

REFERENCES

[1] O. Ḿartínez-Mozos, R. Triebel, P. Jensfelt, A. Rottmann, and W. Burgard, "Supervised semantic labeling of places using information extracted from sensor data," *Robot. Auton. Syst.*, vol. 55, no. 5, pp. 391–402, 2007.

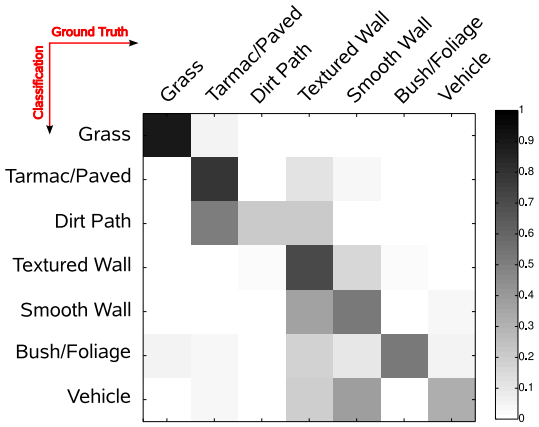
[2] B. Limketkai, L. Liao, and D. Fox, "Relational object maps for mobile robots." in *IJCAI*, L. P. Kaelbling and A. Saffiotti, Eds. Professional Book Center, 2005, pp. 1471–1476.

[3] A. Ranganathan and F. Dellaert, "Semantic modeling of places using objects," in *Proc. of Robotics: Science and Systems*, Atlanta, GA, USA, June 2007.

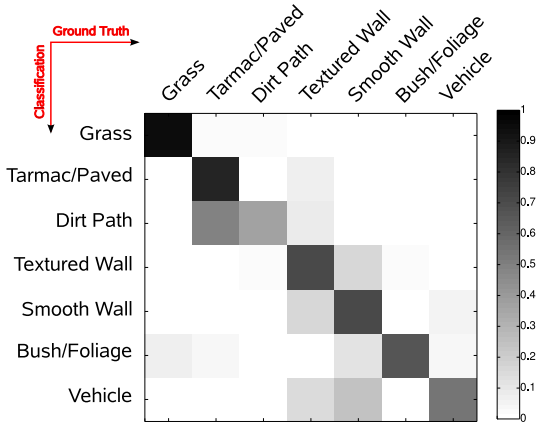
[4] D. Anguelov, B. Taskar, V. Chatalbashev, D. Koller, D. Gupta, G. Heitz, and A. Y. Ng, "Discriminative learning of Markov random fields for segmentation of 3D scan data." in *CVPR (2)*. IEEE Computer Society, 2005, pp. 169–176.

[5] R. Triebel, K. Kersting, and W. Burgard, "Robust 3D scan point classification using associative markov networks," in *"In Proceedings of the International Conference on Robotics and Automation(ICRA)"*, 2006.

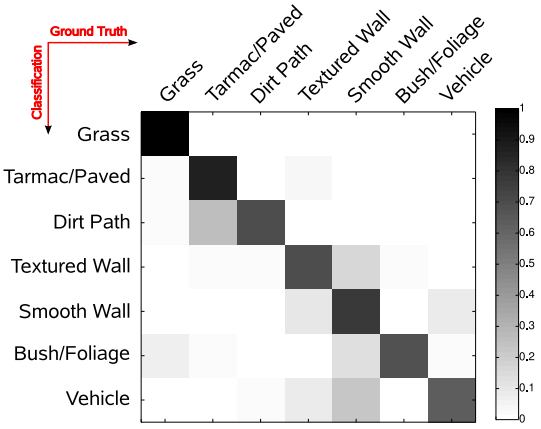
[6] G. Monteiro, C. Premebida, P. Peixoto, and U. Nunes, "Tracking and Classification of Dynamic Obstacles Using Laser Range Finder and Vision," in *Workshop on "Safe Navigation in Open and Dynamic Environments - Autonomous Systems versus Driving Assistance Systems" at the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2006.



(a)



(b)



(c)

Fig. 5. The confusion matrices resulting from an application of our classification framework to the (unbalanced) *Oxford Science Park* data set: **(a)** the output of the patch classification stage before MRF smoothing is applied; **(b)** the output after MRF smoothing obtained using tuning policy 1 and **(c)** the output after MRF smoothing obtained using tuning policy 2. Note that entries on the diagonals represent the *precision* with which the particular class is classified (cf. Table III). See text for more details.

TABLE III

DETAILED CLASSIFICATION RESULTS FOR THE OXFORD SCIENCE PARK DATA SET.

Class Details		**Pre MRF**			**Post MRF (Tuning Policy 1)**			**Post MRF (Tuning Policy 2)**		
Name	# Patches	Precision [%]	Recall [%]	$F_{0.5}$	Precision [%]	Recall [%]	$F_{0.5}$	Precision [%]	Recall [%]	$F_{0.5}$
Grass	74	90.0	73.0	86.0	95.1	52.7	81.9	100.0	25.7	63.3
Pavement/Tarmac	1078	79.0	84.8	80.1	85.4	92.1	86.7	87.4	91.6	88.2
Dirt Path	116	21.8	37.9	23.8	38.2	29.3	36.0	70.7	25.0	51.8
Textured Wall	1678	71.4	75.6	72.2	72.6	89.5	75.4	70.2	94.1	74.0
Smooth Wall	688	53.7	34.5	48.3	72.6	37.8	61.3	77.4	37.8	64.0
Bush/Foliage	161	52.7	49.1	51.9	67.0	44.1	60.7	69.5	45.3	62.8
Vehicle	143	32.2	34.3	32.6	55.4	43.4	52.5	63.8	25.9	49.3

[7] B. Douillard, D. Fox, and F. Ramos, "A Spatio-Temporal Probabilistic Model for Multi-Sensor Multi-Class Object Recognition," in *in Proc. 13th Intl. Symp. of Robotics Research (ISRR)*, 2007.

[8] I. Posner, D. Schröter, and P. M. Newman, "Describing composite urban workspaces," in *In Proc. Intl. Conf. on Robotics and Automation (ICRA)*, 2007.

[9] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *Int. J. Comput. Vision*, vol. 59, no. 2, pp. 167–181, 2004.

[10] J. Sivic and A. Zisserman, "Video Google: A text retrieval approach to object matching in videos," in *Proceedings of the International Conference on Computer Vision*, Nice, France, October 2003.

[11] M. Cummins and P. Newman, "Probabilistic appearance based navigation and loop closing," in *Proc. IEEE International Conference on Robotics and Automation (ICRA'07)*, Rome, April 2007.

[12] C. Chow and C. Liu, "Approximating Discrete Probability Distributions with Dependence Trees," *IEEE Transactions on Information Theory*, vol. IT-14, no. 3, May 1968.

[13] O. Maron and A. W. Moore, "Hoeffding races: Accelerating model selection search for classification and function approximation," in *Advances in Neural Information Processing Systems*, 1994.

[14] S. Boucheron, G. Lugosi, and O. Bousquet, *Concentration Inequalities*. Heidelberg, Germany: Springer, 2004, vol. Lecture Notes in Artificial Intelligence 3176, pp. 208–240.

[15] G. Bennett, "Probability inequalities for the sum of independent random variables," *Journal of the American Statistical Association*, vol. 57, pp. 33–45, March 1962.

[16] S. Geman and D. Geman, "Stochastic relaxation, gibbs distributions, and the bayesian restoration of images," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 6, no. 6, November 1984.

[17] V. Kolmogorov, "Convergent tree-reweighted message passing for energy minimization," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 28, no. 10, pp. 1568–1583, 2006.