Noname manuscript No.
(will be inserted by the editor)

A Generative Framework for Fast Urban Labeling Using Spatial And Temporal Context

Ingmar Posner · Mark Cummins · Paul Newman

Received: date / Accepted: date

Abstract This paper introduces a multi-level classification framework for the semantic annotation of urban maps as provided by a mobile robot. Environmental cues are considered for classification at different scales. The first stage considers local scene properties using a probabilistic bag-of-words classifier. The second stage incorporates contextual information across a given scene (spatial context) and across several consecutive scenes (temporal context) 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 visual and geometric features. By framing the classification exercise probabilistically we take advantage of an information-theoretic bail-out policy when evaluating 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. We demonstrate the virtue of considering such spatial and temporal context during the classification task and analyze the performance of our technique on data gathered over almost 17 km of track through a city.

Keywords Semantic mapping · Machine learning · Markov Random Field · Context-based classification · Image segmentation · 3D laser data

1 INTRODUCTION

Contemporary online mapping and simultaneous localization techniques using lidar now produce compelling 3D geometric representations of a mobile robot's workspace. These maps tend to be geometrically rich but semantically impoverished and, while maps in the form of large unstructured point clouds are meaningful to human observers, they are of limited operational use to a robot. Our work seeks to redress this shortcoming. There is much to be gained by having the robot itself label the map with 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.

The motivation for the work presented here is the necessity for a system capable of producing such semantic labels efficiently enough to be useful in an online robotics setting while taking into account a wide variety of environmental cues. Of particular interest here is the inclusion of contextual information — both within a given scene as well as across consecutive scenes — into the classification procedure.

In this paper we outline a probabilistic method which achieves fast labeling of regions in a scene by performing inference at multiple scales: locally, using scene wide context and, finally, using context provided by evidence across consecutive scenes. Although the application here leverages a combination of 3D range and image data the proposed framework is by no means limited to these modalities. At a local scale, 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

Robotics Research Group, Dept. Engineering Science,
Oxford University, Parks Road, Oxford, OX1 3PJ,
United Kingdom
E-mail: {hip,mjc,pnewman}@robots.ox.ac.uk

”bail-out” policy when evaluating class conditional likelihoods, resulting in large computational savings. Secondly, at the scene-wide scale, we use a Markov Random Field (MRF) to model the expected relationships between patch labels both spatially and temporally, thus capturing some of the strong structural relationships between parts of a typical urban scene. Finally, at the temporal scale, the scene-wide MRFs across consecutive image frames are combined into a single graph over which inference is performed. The operation on entire scene patches yields MRFs of relatively low node-count, just one node for each scene patch, allowing for rapid inference.

While the core classification framework described in this paper was first outlined in [35], the temporal component of the classification framework presented here provides an important extension of this work. Along with a more in-depth discussion and motivation of our original approach we provide an extended evaluation of results and provide evidence of the virtue of considering both spatial and temporal context in the classification task.

The following section provides an overview of related works. Sections 3 and 4 introduce the workspace classes considered and the 3D geometric and appearance-based features used. The generative model providing the local baseline classifications is described in detail in Section 5. In Section 6 we introduce MRFs and describe an intuitive method of extracting the relevant graph structures — over both space and time — directly from the underlying data. Section 7 provides an in-depth discussion of results. Finally, we conclude in Section 8.

2 Related Work

In recent years there has been growing interest within robotics in the problems of environment understanding and scene labeling, particularly as solutions to the SLAM problem become more mature and the limitations of unannotated maps become more apparent. The change has also been driven by the increasing availability of rich sensory data that makes the classification problem more tractable.

A range of machine learning techniques have been brought to bear on the problem. Martinez et al. [24], for example, classify 2D range scans into classes such as *corridor*, *room*, and *door*, applying AdaBoost and Hidden Markov Models. Anguelov et al. [1] described an expectation-maximization (EM) based approach to learn the position of doors in a hallway from 2D line segment maps.

Although 2D laser data is sufficient for very constrained classification tasks, the information content is generally too limited for more general scene understanding. A natural extension is to utilize 3D data. For example, the system described in [2] uses 3D laser data for terrain classification and car detection using a Markov Random Field (MRF) model where inference is performed with Graph Cut. The approach was extended by [41].

Over the past decades, a large body of work in computer vision has also focused on the semantic interpretation of image content, in particular object detection and recognition as well as scene description. The resulting algorithms, whether they apply probabilistic feature-based approaches [31] or use 3D geometric models [32] have matured to a level where impressive performance is achieved. Of particular relevance to the results presented here is the use of image context described in [15]. Though robotics applications have a large overlap with this body of work, they also present unique circumstances, particularly with respect to the availability of multiple sensor modalities and stronger constraints regarding timing performance. However, images are a particularly rich and well researched modality, and the use of visual appearance within robotics is increasing. Hadsell et al. [16], for example, use visual appearance to classify outdoor terrain regarding its traversability by a mobile robot. In [34] image similarity is utilized to perform an unsupervised partitioning of outdoor workspaces and thereby defining descriptive classes such as *park* and *building*. Visual appearance has also been successfully applied in topological mapping and place recognition [8], although with no notion of semantic content.

Much recent robotics work has taken advantage of the multiple sensor modalities available on typical robotic platforms. In [27] a combination of image and 2D laser data were are utilized to classify cars and pedestrians. The classification is carried out separately in each feature space and the results are combined by a Bayesian sum decision rule. Several approaches to the classification of traversability utilize a monocular camera and a fixed 2D laser range finder that faces downwards in front of the vehicle [43, 39]. The assumption is that the 3D pose is known or can be determined with sufficient precision. As a consequence, the laser measurements from different poses can be accumulated to form a 3D point cloud, from which features like planarity or goodness of plane-fit can be computed. Together with visual appearance, these features are used to classify whether or not the terrain in front of the vehicle is traversable. These approaches are related to our work in that they draw their features from image as well as

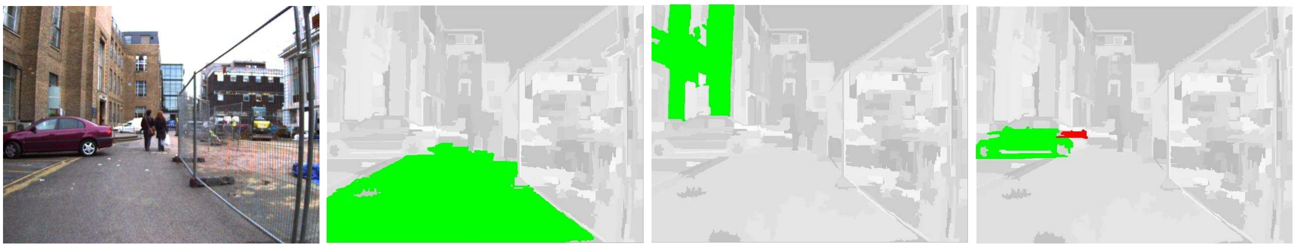


Fig. 1 Classification results for a typical urban scene using spatial and temporal smoothing with a window size of three frames and history depth of one frame (see Section 7 for details): the original image **(left)**; segments classified as 'pavement/tarmac' **(mid-left)**; segments classified as 'textured wall' **(mid-right)**; segments classified as 'vehicle' **(right)**. The colour-coding is wrt. to ground-truth: green indicates a correct label; red indicates a false negative.

3D laser range data. However, multi-class classification is not considered. Similar work by Happold et al. [17] utilise 3D data from stereo vision along with appearance features using a neural network for terrain classification. 3D laser data were combined with visual information in [33], which used support vector machines for classification but did not make use of contextual information. Douillard et al. [9] present a probabilistic framework for object recognition using *Conditional Random Fields* that supports the integration of arbitrarily many sensors, work that was recently extended in [10].

The development of a sound theoretical foundation to inference on graphical models has also spawned a wealth of work where contextual information is taken into account during the labeling process. Contextual information is used explicitly in [6,19] and [10] to classify/detect objects. In particular, the latter relies on Ada-boost and a decision tree of classifiers to model the interplay between objects and scene geometry – for example like that existing between a pedestrian and the ground-plane. Context information, modelled by relational Markov networks, was also used in [22] for classification of segment-based representations of indoor environments. More recently [36] introduced an approach which takes into account spatial relationships between objects and object parts in 3D.

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 - or *superpixels* - rather than over measurements (e.g. laser range data) directly. This yields substantial speed increases over previous approaches. We further define a generative bag-of-words classifier and describe an efficient inference procedure for it.

3 Workspace Classes in Urban Environments

When navigating in an urban context a higher-order knowledge of the environment is indispensable. For example, the detection of cars (moving or stationary) is important for safe operation. Recognition of ubiquitous urban elements such as the colour and texture of surrounding houses (or, more appropriately, of surrounding walls) and the presence or absence of other features such as grass, bushes or trees can provide a useful navigational cue. These considerations give rise to the seven classes defined in Table 1, comprising ground types, building types, and two object categories.

4 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. A cross-calibration between the two sensors allows for the projection of the laser data into the image. The fundamental entities considered for classification are visually homogeneous image segments - or *superpixels* - obtained using an off-the-shelf image segmentation algorithm [12]. For each classified superpixel, we have associated 3D geometric information from the laser as well as colour and texture information from the image.

Table 1 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.

The features used for classification are similar to those described in [33]. For geometry description, the 3D point cloud is first segmented into planar patches using the technique of [42]. The point cloud is divided into cubic cells, and planes are fitted in each cell using MLESAC [40]. Planes in neighbouring cells which have similar surface normal orientation are then merged. Finally, each planar patch is subdivided on the basis of the image segmentation. This yields a set of visually and geometrically homogenous regions in the scene, which are the entities we consider for classification. Each patch is described based on a set of features computed for each laser point contained within the patch boundaries. Colour and texture features are computed from a 15×15 pixel neighbourhood and comprise of a hue- and a saturation histogram of 15 bins each as well as the variance of each of these histograms. The descriptor also includes 3D spatial information in form of the associated plane normal orientation encoded as cosine-distance with the vertical axis. This makes the implicit assumption that the robot pose is always upright. Finally, 2D spatial information is included in the form of the normalized x and y coordinates of the projected laser point within the image. The intuition behind this last set of features is that, for a permanently upright robot, ground classes predominantly occur near the bottom of the image while spanning the entire width. This distribution is different for non-ground classes. The feature set is summarized in Table 2.

The feature set employed here provides relatively weak cues for classification - our purpose in this paper is not to describe the best possible feature set for outdoor robotics, but rather to show what can be achieved with an appropriate inference framework even in the absence of strong features.

Table 2 Features used for classification

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

5 GENERATIVE PROBABILISTIC CLASSIFICATION

The inference framework proposed in this paper is a multi-level approach based on successive combinations of lower-level features. 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 4. Rather than deal with raw data directly, we adopt the *bag-of-words* representation, where the feature vectors are quantized with respect to a "vocabulary" of prototypical features. (see Figure 2). This approach is widely used in the computer vision community [21,37,38]. The vocabulary is constructed by clustering all the standardized feature vectors from a set of training data. Our system uses an incremental leader-follower clustering algorithm [11]. This yields a vocabulary of size $|v|$, defined by the cluster centres. The vocabulary size is determined by a user-specified distance threshold, which implicitly sets the number of clusters. For this work we use a vocabulary of approximately 6,500 words. When the system has been trained, each incoming laser point yields a feature vector which is quantized to the approximate nearest cluster centre using a kd-tree. The laser point's feature vector can then be replaced by a single integer specifiy which cluster centre it quantized to. The laser points in a patch then define a bag-of-words. The bag-of-words for each patch in the scene is the input to the next level of the system, which aims to provide a soft classification of any given patch. The remainder of this section provides a detailed description of the generative probabilistic model we employ for this classification task.

5.1 Generative Model

The core of our classification framework involves learning a generative model of the bag-of-words input. This generative model can be learned from plentiful unlabeled data, across all classes. The learning procedure locates structure in the feature space - for example, two features might commonly occur together because they are generated by different parts of the same object. This structure can be discovered without supervision. Scarce labelled data is employed only after the generative model has been learned. In effect classes are defined with respect to the higher level structure discovered by the generative model learning phase.

Our generative model for patch-level classification is illustrated in Figure 3(c). To build intuition about the structure of our model, we first consider the simpler models in Figure 3(a) and 3(b).

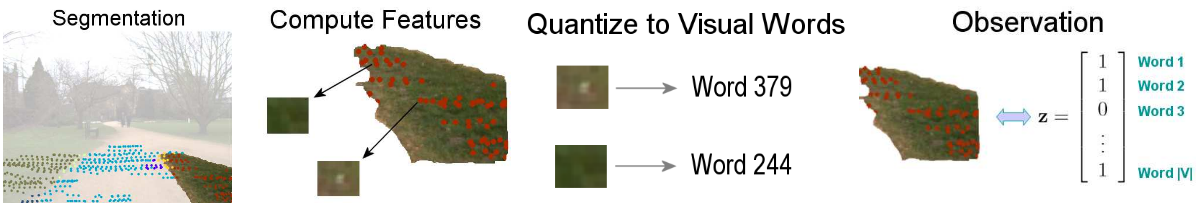
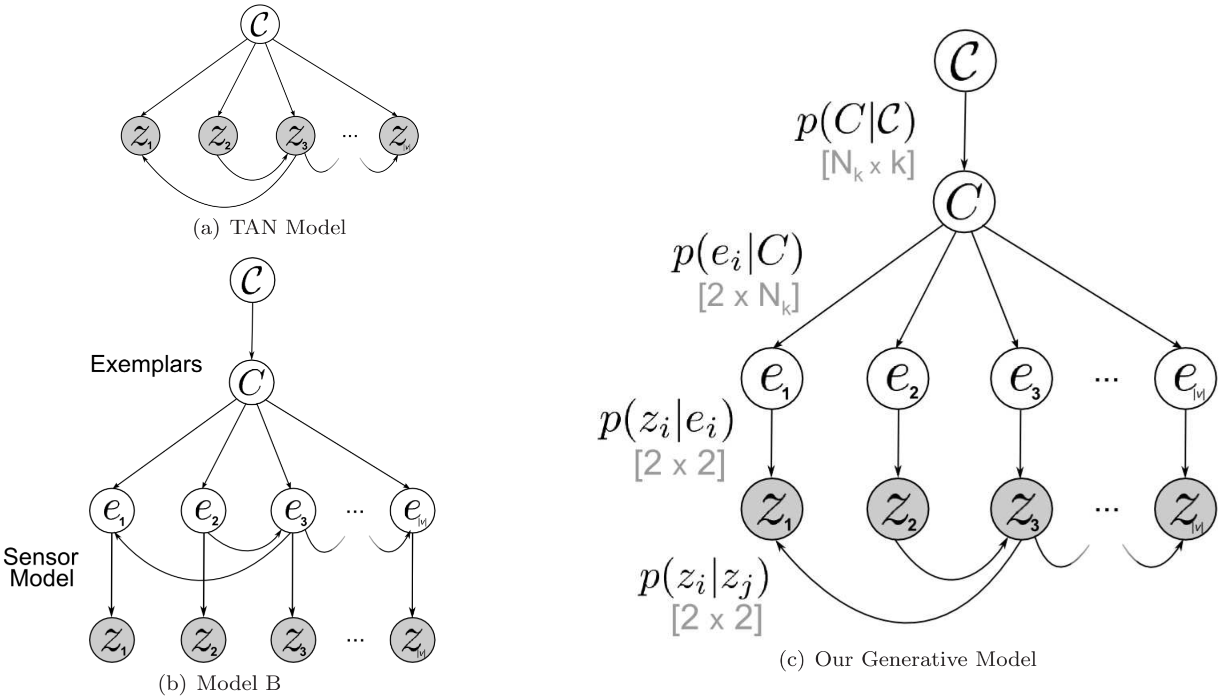


Fig. 2 Feature generation from raw data. Extracted planes are sub-segmented based on the image data. Features are then extracted around each laser point in a patch. Extracted features are quantized to visual words.



(a) TAN Model

(b) Model B

(c) Our Generative Model

Fig. 3 The generative model used in this paper, (c), and several alternative models considered, (a),(b). The relevant conditional probabilities to be estimated along with the size of the associated probability tables are shown in (c).

Figure 3(a) shows a Tree Augmented Naive Bayes or TAN model [13]. Here the class $\mathcal{C}$ directly generates observations z. Word observations are not independent, and this is captured by the edges between the z nodes. These edges form a tree, the structure of which is learned from data using the Chow Liu algorithm (see section 5.4). Related models are discussed in detail in [26]. The main limitation of these models is their limited ability to capture high intra-class variability, due to the restriction that the dependencies between z variables be tree structured. We thus seek a different model to overcome this issue.

The model in Figure 3(b) extends the TAN model by introducing *exemplars* and a *sensor model*. Rather than define a density over observations directly, classes now generate exemplars. Exemplars define a density over the e variables, which in turn produce observations z. Exemplars improve the model's ability to capture high intra-class variability, by essentially decomposing the class density into a set of local densities around exemplars. The effective "distance function" around an exemplar is determined by the sensor model and the correlations in the Chow Liu tree, and so is derived from the data itself. Exemplars also allow for easy online updates to the classifier - a class can be updated simply by adding a new exemplar.

The e variables are introduced to allow for the incorporation of a sensor model. Intuitively, e represents "existence" and z "observation". The two quantities are linked via a sensor model described in Section 5.3 which allows for false positive and false negative detections.

Explicitly separating feature observation z from feature existence e provides a natural framework for dealing with multiple sensors and time-varying sensor accuracy. For example, our expectation of observing a particular visual word in a class might be high, however, if we also know that current lighting conditions are poor, then a failure to observe the word is less surprising. Equally we can allow for the fact that different observations may originate from cameras with different resolution, for example. These effects cannot be incorporated into a model where exemplars C directly define a density over observations z.

Finally, the model in Figure 3(c) is the one we actually use for classification. This is essentially an approximation to model (b). For reasons of tractability, we now impose the tree-structured dependencies between the observed z variables, while the unobserved e variables are now independent. Learning the structure of this model from data is considerably easier, because the dependencies to be determined are between observed variables only.

5.2 Patch-Level Classifier

Our patch-level classifier is inspired by the probabilistic appearance model described in [8] 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 $\mathbf{z} = \left\{z_1, \ldots, z_{|v|}\right\}$ 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.

5.3 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 $\mathcal{C}^k = \left\{C_1^k, \ldots, C_{n_k}^k\right\}$ 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 $\left\{p(e_1|C_i^k), \ldots, p(e_{|v|}|C_i^k)\right\}$. 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, as described in the previous section, it provides a natural framework for incorporating data from multiple sensors, where each sensor has different (and possibly time-varying) error characteristics. Secondly, as we will discuss later, 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.

5.4 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 [8]. The learning scheme we employ is the Chow Liu tree, which locates a tree-structured Bayesian network that approximates the true distribution [5]. 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.

The Chow Liu tree 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 5.3. Returning to Equation 5 and employing the Chow Liu approximation, we have

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

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

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

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

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

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

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

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[1]. The posterior distribution across classes, $p(\mathcal{C}^k|\mathbf{z})$, can now be computed for each patch. It should be noted that this operation is linear in the number of class exemplars in the system.

5.5 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 5.3. 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

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

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 $p(e_q = 1)$.

5.6 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 [23,25,29]. We introduced this technique in [7], and briefly summarize it below.

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

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

Now, define

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

and

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

[1] Clearly this normalization would benefit from a background class, a change we plan to make in future versions of the system.

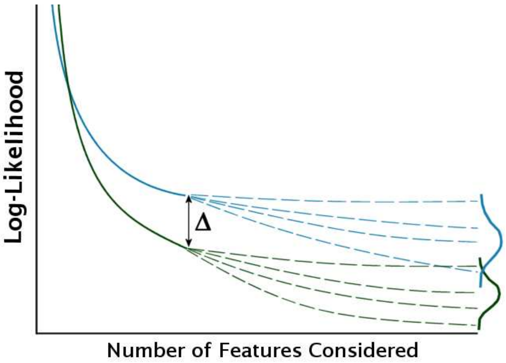


Fig. 4 Conceptual illustration of the bail-out test. After considering the first j words, the difference in log-likelihoods between two exemplars is Δ. Given some statistics about the remaining words, it is possible to compute a bound on the probability that the evaluation of the remaining words will cause one exemplar to overtake the other. If this probability is sufficiently small, the trailing exemplar can be discarded.

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 best hypothesis. *Too far* can be quantified using concentration inequalities [4], which yield a bound on the probability that the discarded exemplar will have a higher final log-likelihood than the current best exemplar, 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 4. 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 exemplar 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 [3,18]. 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. This is because X_j has a multinomial distribution corresponding to the values of $p(e_j|C_i^k)$ for each exemplar in each class. This is a small set of possible values, which allows for rapid computation of the variance.

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 σ^2 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, with only very slight impact on accuracy.

6 Markov Random Fields For Spatio-Temporal 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 maximized by equivalently minimizing an energy function incorporating a *unary* term modeling the data likelihood for each node and a *binary* term specifying the interaction potentials between neighbouring nodes over the set of possible values [14]. Under the assumption of every datum being equally likely (i.e. $p(\mathcal{Z})$ being uniform) a minimization 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 labeling 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.

6.1 Model Structure

Spatial Context. MRFs are a family of graphical models where the set of interdependent variables is modeled as a graph $G(\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ denotes the set of vertices and $\mathcal{E}$ denotes the set of edges, respectively. In the context of our scene labeling problem, each vertex represents an image patch as introduced in Section 4. Neighbourhood relations within each scene are defined between patches sharing a common border, information provided directly by the segmentation algorithm [12]. 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 5.

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 the inference stage from the density of the underlying data. This provides a substantial advantage in terms of speed over related works [9,2] where the complexity of the graphical models is directly proportional to the density of the underlying data.

Temporal Context. In addition to spatial context within an image, we also exploit temporal context between images. The same object should be assigned the same class label in all frames in which it is observed. We enforce this relationship by defining a joint MRF over several consecutive camera frames, as illustrated in Figure 6. We thus have a single inference procedure to handle both spatial and temporal information. Temporal links between patches are determined by projecting laser data from frame i into frame $i - d$, where d denotes the *history depth*. Patches in frame i and $i - d$ that contain more than 20% of common laser points are linked by a temporal edge. Since the image segmentation can vary considerably between images, this is often a one-to-many relationship. The transformation between frames is determined by vehicle odometry information, which is reliable enough over short distances to ensure that patches linked by this procedure usually correspond to the same physical object.

6.2 Model Parameters

The specification of an energy function to be optimized 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 5) 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

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

where we adopt the notation of [20] 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. 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 provided by the patch classifier introduced in Section 4. 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 minimized.

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

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

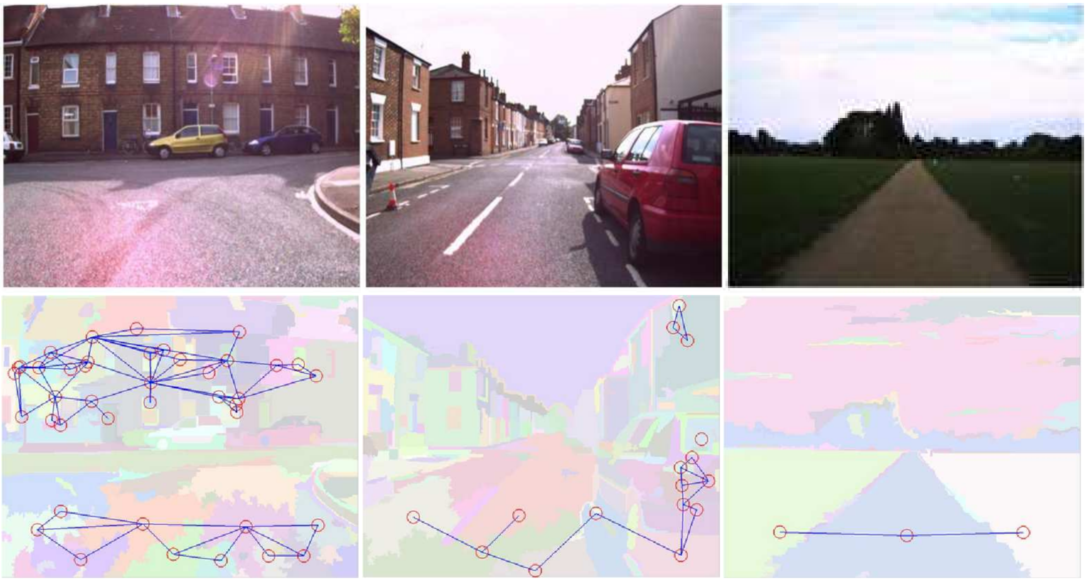


Fig. 5 Typical graphs extracted from urban scenes as recorded by our mobile robot. **Top:** the original scenes. **Bottom:** the corresponding segmented images with the extracted graph overlaid. Circles indicate nodes, lines indicate edges. For images patches which are not marked as nodes no reliable geometry estimates could be extracted from the laser data.

where again the complement is used since a penalty function is specified. We specify two such matrices $\mathbf{\Phi}_t$ and $\mathbf{\Phi}_s$, for the temporal and spatial edges respectively. For spatial edges we specify $\mathbf{\Phi}_s$ such that, for two classes i and j,

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

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. The temporal edges $\mathbf{\Phi}_t$ are specified such that

$$\phi_{\mathbf{i},\mathbf{j}} = 1, \forall i \neq j, \tag{23}$$

$$\phi_{\mathbf{i},\mathbf{i}} = 0, \tag{24}$$

thus enforcing a uniform penalty on all inconsistent temporal labels.

6.3 Inference

The determination of the MAP configuration of states given a set of observations using an MRF is a common task and there exists an abundance of techniques to perform the appropriate energy minimization. A popular choice is max-product belief propagation [30]. This method is based on a message passing scheme and provides exact results when applied to tree-structured graphs. Belief propagation is also a popular choice when the underlying graph structure contains loops. Although in this case convergence of the algorithm is not guaranteed, "loopy belief propagation" has nevertheless been applied with success to a variety of problems [28, 44].

An alternative to loopy belief propagation is sequential tree-reweighted message passing (TRW-S) [20]. Like belief propagation, TRW-S is based on a message passing scheme. However, it is designed to efficiently maximize a lower bound on the energy and *guarantees* that this bound will not decrease in consequent iterations. Because of this performance guarantee we employ TRW-S to perform inference throughout the remainder of this work. It should be noted that based on our potentials, TRW-S scales linearly with the number of edges in the graph and quadratically with the number of classes.

7 Results

The algorithm presented above was tested using two extensive outdoor data sets [2] 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

[2] These datasets were previously introduced in [33].

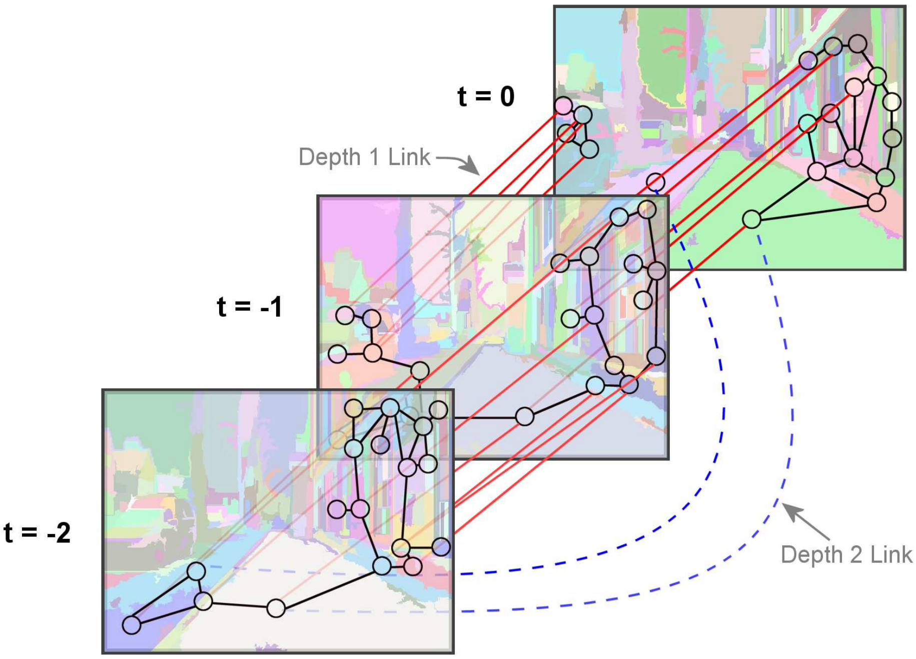


Fig. 6 Conceptual illustration of the temporal MRF for three successive images. Spatial links are shown in black, depth one temporal links in red. Some depth two temporal links are also shown as blue dashed lines. Inference is carried out jointly over this spatio-temporal graph.

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 with a true positive rate $p(z_i = 1|e_i = 1) = 0.35$ and a false positive rate $p(z_i = 0|e_i = 1) = 0$.

We evaluated the performance of the individual stages of the classifier using 4,932 patches from 217 non-consecutive frames of the *Oxford Science Park* data set whose ground truth had been labeled by hand. A typical result is shown in Figure 1. A quantitative analysis of classification performance is presented in Table 3. 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 stated in order to provide a convenient single figure measure of overall classification performance per class.

Table 3 indicates that the patch classifier (pre-MRF) provides a baseline classification of mixed quality. Good results are achieved mainly for common classes (e.g. *pavement/tarmac*, *textured wall* and *smooth wall*, as well as for some less common ones (e.g. *grass*). The effect of both spatial and temporal context is pronounced. For common classes we note a boost to both precision and recall. For rarer classes such as *vehicle* and particularly *grass* the MRF has the effect of boosting precision at the cost of some drop in recall. This tends to happen as weaker partial detections of objects are reassigned based on surrounding labels, typically eliminating many false positives but suppressing weaker true positives.

The quantitative analysis is augmented in Figure 8 in the form of confusion matrices for both the output of our patch-level classifier as well as the output after spatio-temporal smoothing using the MRF. These matrices are normalized, on one hand, such that the values along the diagonals represent per-class *precision* and, on the other hand, such that the values along the diago-

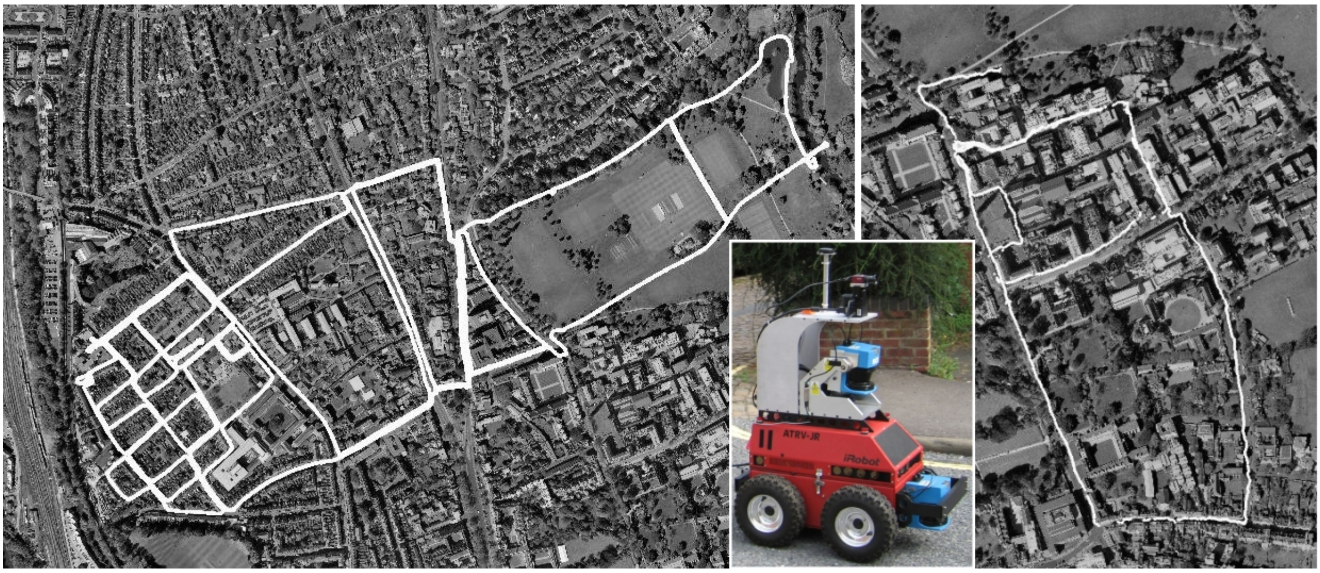


Fig. 7 Aerial map of the *Jericho* data set - 13.2 km, 16000 images (**Left**), and the *Oxford Science Park* data set - 3.3 km, 8536 images (**Right**). Vehicle trajectories are marked in white. **Middle:** *Marge* - our ATRV research platform. Reproduced from citePosnerEtAl:RAS08
.

nals represent per-class *recall* (cf. Table 3). Thus, the former provides information (along the rows) on how reliable the given labels are compared to ground truth — i.e. how much trust can we put in the obtained labels — whereas the latter provides information (along the columns) of how well ground-truth data are retrieved.

Prior to incorporating the MRF, there is notable confusion in precision between the *vehicle, foliage* and wall classes. Results incorporating the MRF show a considerable improvement. While precision increases substantially across all classes, the confusion between the *vehicle, foliage* and wall classes has been reduced. Confusion is also reduced between *tarmac/pavement* and *dirt path.* The remaining confusion is primarily between closely related classes such as the two wall types.

The benefits of MRF smoothing in terms of recall are more varied and particularly striking for the dominant classes *pavement/tarmac, textured wall* and *smooth wall*, where confusion with other classes decreases dramatically. For less common classes, a significant amount of over-smoothing occurs. This is particularly striking in the case of *grass*, which is now commonly misclassified as *pavement/tarmac*. Similar over-smoothing - although significantly less pronounced - occurs for *bush/foliage* (now commonly misclassified as *textured wall*) and *vehicle* (now increasingly misclassified as *smooth wall*). Smoothing effects on recall for *dirt path* are marginal.

The final results demonstrate generally good precision and reasonable recall performance, particularly for the common classes. A comparison of classification performance in terms of $F_{0.5}$ measure with that expected from a classifier making random decisions based on class priors only is provided in Figure 9.

7.1 Spatial And Temporal Smoothing

The MRF provides context-sensitive smoothing of classification results in space (i.e. within any given frame) as well as time (i.e. across several frames in a time window). Figure 10 aims to discern the benefits on the actual classification of smoothing in either dimension. Shown are $F_{0.5}$ variations with window size and history depth (cf. Figure 6) for both individual and combined applications of temporal and spatial smoothing. The left column of the figure demonstrates the effect of temporal smoothing alone. With the exclusion of one class, the classification performance increases significantly with time window size. A further gain is made as history depth increases and more information is included. On the whole, temporal smoothing alone thus provides substantial benefits in classification performance. The adverse effect the smoothing has on *grass* may be explained by a misclassification of the majority of *grass* patches in any given unlabeled frame within a time window due to, for example, adverse lighting conditions or dynamic objects moving through the scene. By virtue of the MRF this significant disinformation is then propagated throughout the window.

The right column of Figure 10 shows the effect of spatial smoothing only and also the benefits of combining both types of context. Note that a window size of one frame corresponds to only considering a single frame overall, indicating that only spatial smoothing is applied. In general, significant improvements of overall classification performance beyond those obtained using temporal smoothing alone can be observed. This is particularly true for *dirt path* and *vehicle*.

Based on the evidence presented in Figure 10 a window size of three frames and a history depth of a single frame were chosen for the results in Table 3.

7.2 Timing

The timing properties of our algorithm are outlined in Table 4. Run times are from a 2GHz Pentium laptop. The mean total processing time was 4.0 seconds.

7.3 Comparative Evaluation

It is instructive to gain an intuition as to how the presented approach compares to the application of state-of-the-art classifiers such as support vector machines (SVMs). To this end this section presents our own results compared to those obtained in [33], where the same datasets as well as nominally the same features were used for superpixel labelling using voted SVM classification. Results are presented side-by-side in Table 5. The voted SVM approach is most directly comparable to the output of the first stage of our system, since both approaches classify entire superpixels without considering context beyond the superpixel boundaries. It is immediately apparent that the SVM classification provides much improved precision and recall values across the classes. In particular, the performance for *grass*, *dirt track*, *bush* and *vehicle* suffers in our probabilistic classifier. Inspection of the confusion matrices in Figures 8(a) and 8(b) reveals that performance is compromised due to confusion between ground and non-ground classes and, to a lesser extent, between classes with distinct colour characteristics. A typical example is *grass*, which suffers from confusion with both *tarmac/pavement* and *dirt path* as well as with *bush*. Similarly, *vehicle* is confused with both wall classes as well as the *tarmac/pavement* class. These confusions are significantly less severe in the corresponding SVM results as depicted in Figure 11. The SVM therefore manages to distinguish more rigorously between classes based on the same features. This suggests that our first stage classifier model does not currently assign the appropriate importance to individual features in the feature set. We believe this effect is caused by combining different feature spaces (e.g. colour, geometry) into a single visual vocabulary. Preliminary experiments suggest that maintaining the separation between feature spaces in the vocabulary noticably alleviates this issue.

Table 4 Timing Information (in milliseconds).

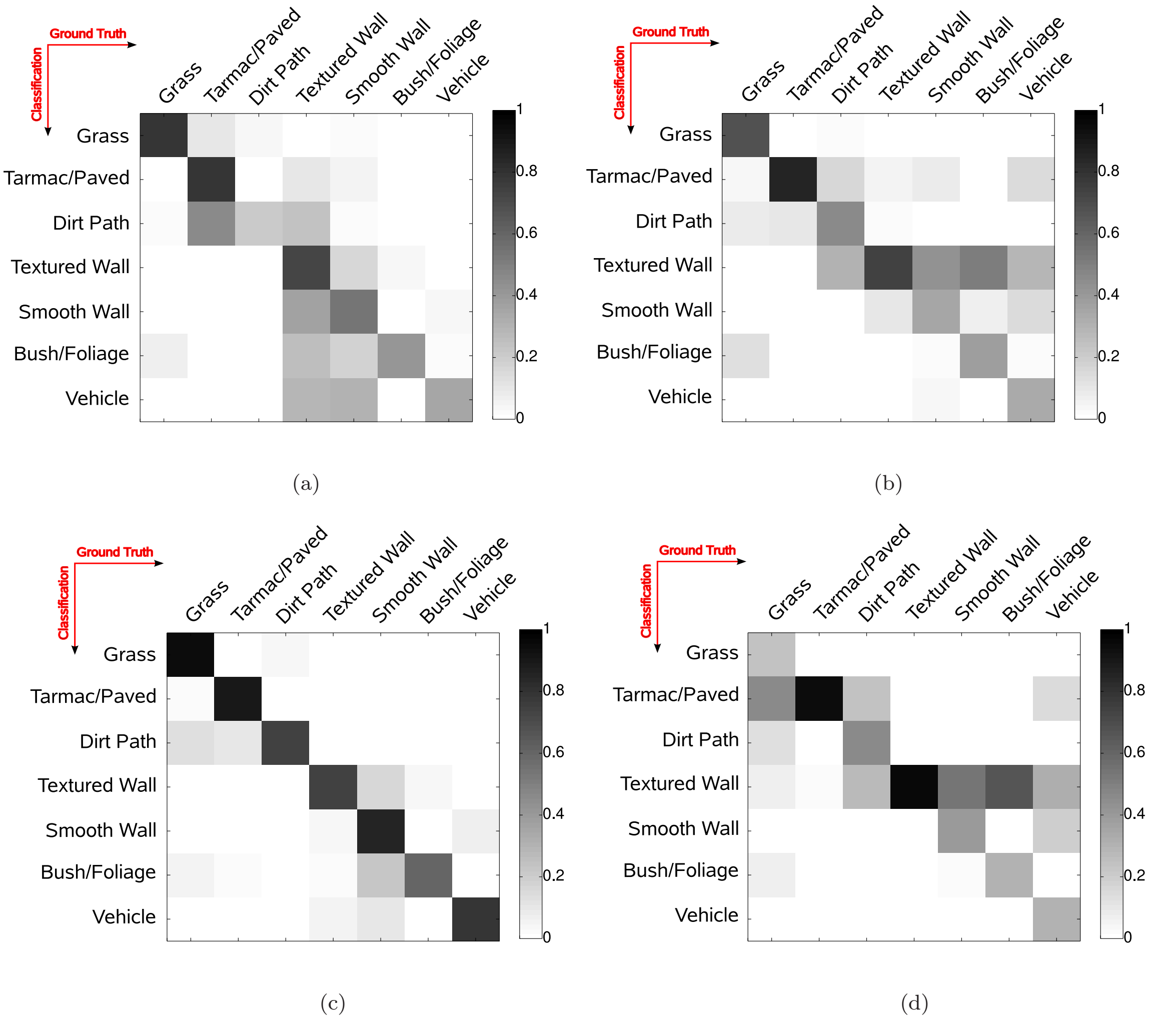
Process	Mean (ms)	Max (ms)
Plane Segmentation	2000	2800
Feature Extraction	89	125
Feature Quantization	4	90
Image Segmentation	960	1130
Patch Classification	850	3480
MRF Construction	63.5	453.9
MRF Inference	6.0	22.0
Overall	4.0 seconds	8.1 seconds

Thus, while the generative probabilistic classifier introduced in Section 5 has several attractive properties from a mobile robotics perspective in terms of flexibility, some important issues remain yet unexplored. It is interesting to note also, that in providing this proof of concept we have neglected more advanced tuning of this part of the system. In particular, the values of the sensor model employed here have been adopted from previous applications such as [7]. Selection of more appropriate values based on training data may open avenues for improvement.

The performance gap between the voted SVM results and the system presented here narrows significantly after MRF smoothing is applied. While this is, of course, a somewhat unfair comparison since no context is applied in the SVM classifications, this point serves to reinforce the positive impact context-based smoothing can have. Furthermore, it should be noted that the use of the probabilistic classifier used in the first stage of our work is by no means mandatory. If no requirement exists to, for example, adapt class models online, any classification framework providing soft class assignments can be substituted.

8 Conclusions

This paper has described and provided a detailed analysis of a two-stage approach to fast region labeling in maps of urban environments. Although the approach described here made specific use of both 3D laser and image data, the algorithms described are not limited to these modalities. The principal contribution of this work is the introduction of a layered classification framework which considers local scene properties in the first stage and then applies spatial as well as temporal context to refine these initial classifications. The results demonstrate the improvements in classification performance obtained by accounting for spatial and temporal context. This is despite the fact that the neighbourhood relations encoded in the MRF are a relatively weak cue; stronger information such as relative location and con-



(a)

(b)

(c)

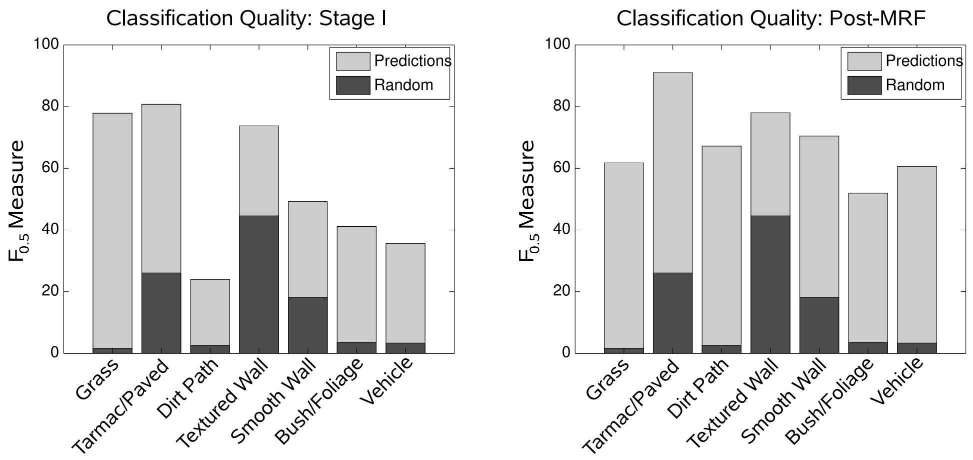
(d)

Fig. 8 The confusion matrices resulting from an application of our classification framework to the (unbalanced) *Oxford Science Park* data set. The top row presents the output of the patch classification stage before MRF smoothing is applied in the form of precision **(a)** and recall **(b)**. The output after MRF smoothing obtained using a window size of three frames with a history depth of one frame is provided in the bottom row, again in the form of precision **(c)** and recall **(d)**. Note that entries on the diagonals of these matrices represent the respective precision and recall values indicated in Table 3.

Table 3 Detailed classification results for the Oxford Science Park data set. Results for the spatio-temporal column were obtained over a three-frame window with a history depth of one frame.

Class Details		**Pre MRF**			**Spatial Context**			**Spatio-Temporal Context**		
Name	# Patches	Precision [%]	Recall [%]	$F_{0.5}$	Precision [%]	Recall [%]	$F_{0.5}$	Precision [%]	Recall [%]	$F_{0.5}$
Gr	82	80.3	69.5	77.9	89.2	40.2	71.7	95.5	25.6	61.8
Ta	1286	79.5	86.1	80.8	89.2	94.9	90.3	89.9	95.7	91.0
Di	127	21.4	47.2	24.0	60.2	46.5	56.8	75.6	46.5	67.2
Te	2199	73.3	75.5	73.8	74.0	93.8	77.3	74.3	97.5	78.0
Sm	898	54.1	36.2	49.2	76.4	39.0	64.1	86.3	40.7	70.5
Bu	175	41.7	38.9	41.1	59.6	35.4	52.5	61.5	32.0	52.0
Ve	165	35.9	34.6	35.6	69.1	33.9	57.3	79.7	30.9	60.6

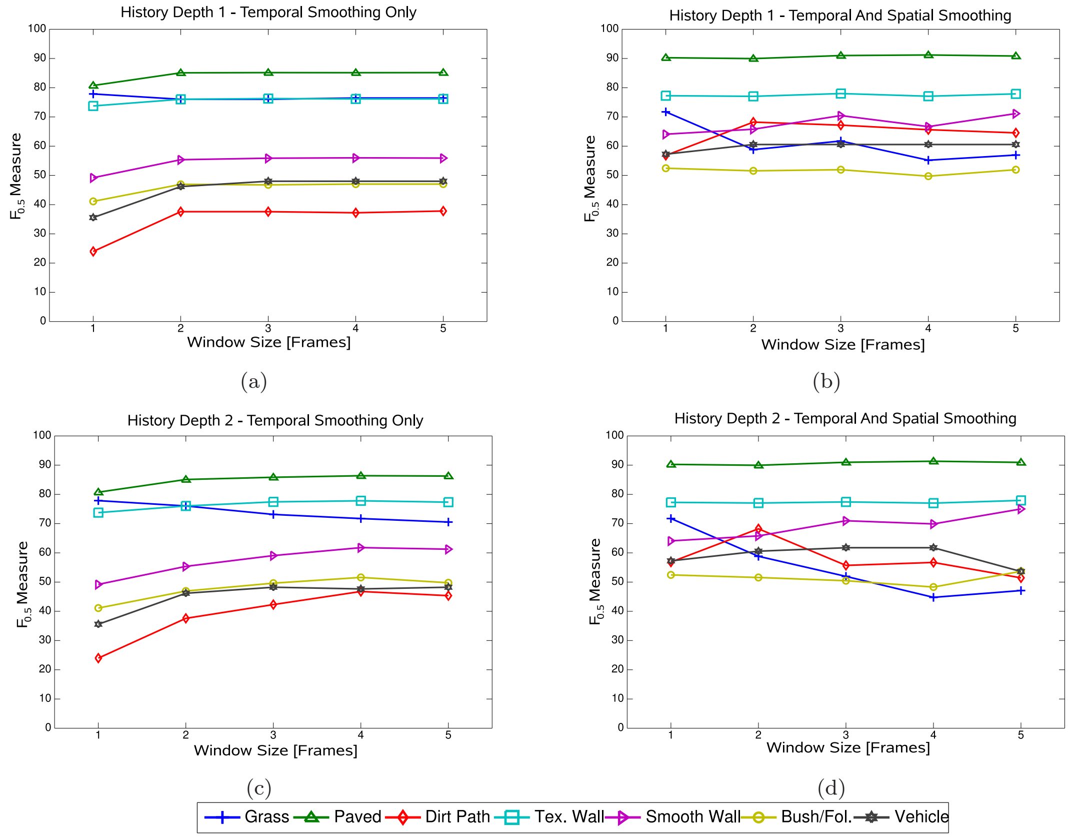
Legend for class shortcuts: **Gr**ass, **Ta**rmac/Paved, **Di**rt Path,
Textured Wall, **Sm**ooth Wall, **Bu**sh/Foliage, **Ve**hicle



(a)

(b)

Fig. 9 A graphical representation of the $F_{0.5}$ values provided in Table 3. The *post-MRF* output was obtained with a window size of three frames with a history depth of one frame. Overlaid are the expected equivalent numbers for a classifier based on class priors only.



(a)

(b)

(c)

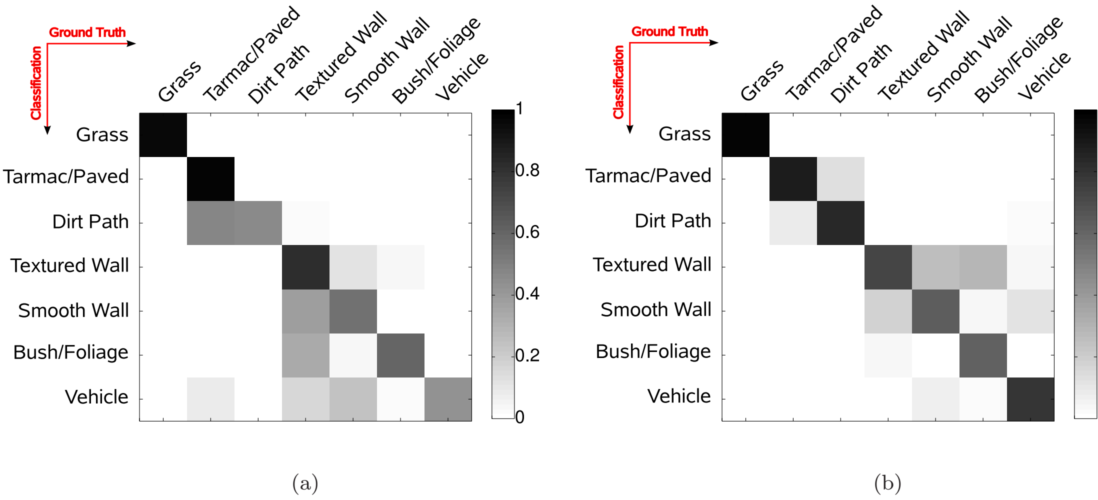
(d)

Fig. 10 An indication of the benefits of temporal and spatial smoothing with increasing window size and history depth. (**a**) and (**c**) present results obtained using temporal smoothing only. Spatial links (i.e. edges linking nodes within the same frame) are ignored. (**b**) and (**d**) present results with both temporal and spatial edge information included.

Table 5 Performance comparison between the stage 1 classifications of our system (Pre MRF) as well as the smoothed results (Spatio-Temporal Context) with voted SVM results on the same dataset with nominally the same features (taken from [33], Table 7).

Class Details	**Voted SVM**			**Pre MRF**			**Spatio-Temporal Context**		
Name	Precision [%]	Recall [%]	$F_{0.5}$	Precision [%]	Recall [%]	$F_{0.5}$	Precision [%]	Recall [%]	$F_{0.5}$
Gr	96.6	98.1	96.9	80.3	69.5	77.9	95.5	25.6	61.8
Ta	97.7	89.0	95.8	79.5	86.1	80.8	89.9	95.7	91.0
Di	46.4	84.8	51.0	21.4	47.2	24.0	75.6	46.5	67.2
Te	82.7	73.5	80.7	73.3	75.5	73.8	74.3	97.5	78.0
Sm	56.9	64.4	58.3	54.1	36.2	49.2	86.3	40.7	70.5
Bu	60.6	62.8	61.0	41.7	38.9	41.1	61.5	32.0	52.0
Ve	43.7	80.1	48.1	35.9	34.6	35.6	79.7	30.9	60.6

Legend for class shortcuts: **Gr**ass, **Ta**rmac/Paved, **Di**rt Path, **Te**xtured Wall, **Sm**ooth Wall, **Bu**sh/Foliage, **Ve**hicle



(a)

(b)

Fig. 11 The confusion matrices resulting from the voted SVM approach described in [33] in the form of precision **(a)** and recall **(b)**. Note that entries on the diagonals of these matrices represent the respective precision and recall values indicated in Table 5. Reproduced from [33], Figure 14.

tainment relations would be expected to improve the results. Furthermore, the inclusion and learning of a relative weighting between unary and binary potentials is expected to improve results since it provides a mechanism to minimize the over-smoothing effected by the MRF in the current system. Conceivably the most direct route to better performance is the addition of more informative features.

Further contributions are the development of efficient and principled methods to accomplish each classification stage. While any classifier capable of providing soft class assignments can conceivably be employed in the first stage of our framework, we opt to describe a probabilistic bag-of-words approach, which employs a principled bail out policy that greatly decreases the computational cost of evaluating likelihood terms. Beyond classification speed, this generative approach has the added advantage of providing a sensor model as a mechanism to incorporate the notion that some of the robot's observations are more trustworthy than others. In addition, the class models can readily be updated online.

Furthermore, the formulation of the MRF model allows the efficient integration of 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 measurement. As a result, the overall per-scene compute time of this method is compelling: at 4.0 seconds it is suitable for online deployment.

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. D. Anguelov, D. Koller, E. Parker, and S. Thrun. Detecting and modeling doors with mobile robots. In *Proc. of the IEEE*

Int. Conference on Robotics and Automation (ICRA), 2004.

2. 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)*, pages 169–176. IEEE Computer Society, 2005.
3. George Bennett. Probability inequalities for the sum of independent random variables. *Journal of the American Statistical Association*, 57:33–45, March 1962.
4. S. Boucheron, G. Lugosi, and O. Bousquet. *Concentration Inequalities*, volume Lecture Notes in Artificial Intelligence 3176, pages 208–240. Springer, Heidelberg, Germany, 2004.
5. C. K. Chow and C. N. Liu. Approximating Discrete Probability Distributions with Dependence Trees. *IEEE Transactions on Information Theory*, IT-14(3), May 1968.
6. N. Cornelis, B. Leibe, K. Cornelis, and L. Van Gool. 3D City Modeling Using Cognitive Loops. In *Proc. of the Third Int. Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT'06)*, 2006.
7. Mark Cummins and Paul Newman. Accelerated appearance-only SLAM. In *Proc. IEEE International Conference on Robotics and Automation (ICRA'08)*, Pasadena,California, April 2008.
8. Mark Cummins and Paul Newman. FAB-MAP: Probabilistic Localization and Mapping in the Space of Appearance. *The International Journal of Robotics Research*, 27(6):647–665, 2008.
9. B. Douillard, D. Fox, and F.T. Ramos. A Spatio-Temporal Probabilistic Model for Multi-Sensor Object Recognition. In *in Proc. of IEEE/RSJ Int. Conference on Intelligent Robots and Systems (IROS)*, 2007.
10. B. Douillard, D. Fox, and F.T. Ramos. Laser and Vision Based Outdoor Object Mapping. In *Proc. of Robotics: Science and Systems*, 2008.
11. R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification (2nd Edition)*. Wiley-Interscience, 2000.
12. P. F. Felzenszwalb and D. P. Huttenlocher. Efficient Graph-Based Image Segmentation. *Int. J. Comput. Vision*, 59(2):167–181, 2004.
13. N. Friedman, D. Geiger, and M. Goldszmidt. Bayesian Network Classifiers. *Machine Learning*, 29(2):131–163, 1997.
14. S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 6(6), November 1984.
15. Stephen Gould, Jim Rodgers, David Cohen, Gal Elidan, and Daphne Koller. Multi-class segmentation with relative location prior. *Int. J. Comput. Vision*, 80(3):300–316, 2008.
16. R. Hadsell, P. Sermanet, J. Ben, A. Erkan, J. Han, U. Muller, and Y. LeCun. Online Learning for Offroad Robots: Spatial Label Propagation to Learn Long-Range Traversability. In *Proc. of Robotics: Science and Systems*, 2007.
17. M. Happold, M. Ollis, and N. Johnson. Enhancing Supervised Terrain Classification with Predictive Unsupervised Learning. In *Proc. of Robotics: Science and Systems*, 2006.
18. W. Hoeffding. Probability Inequalities for Sums of Bounded Random Variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.
19. D. Hoiem, A. A. Efros, and M. Hebert. Putting Objects in Perspective. In *Proc. IEEE Computer Vision and Pattern Recognition (CVPR)*, 2006.
20. V. Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 28(10):1568–1583, 2006.
21. T. Leung and J. Malik. Representing and recognizing the visual appearance of materials using three-dimensional textons. *International Journal of Computer Vision*, 43(1):29–44, 2001.
22. B. Limketkai, L. Liao, and D. Fox. Relational Object Maps for Mobile Robots. In Leslie Pack Kaelbling and Alessandro Saffiotti, editors, *IJCAI*, pages 1471–1476. Professional Book Center, 2005.
23. Oded Maron and Andrew W. Moore. Hoeffding races: Accelerating model selection search for classification and function approximation. In Jack D. Cowan, Gerald Tesauro, and Joshua Alspector, editors, *Advances in Neural Information Processing Systems*, volume 6, pages 59–66. Morgan Kaufmann Publishers, Inc., 1994.
24. O. Ḿartínez-Mozos, C. Stachniss, and W. Burgard. Supervised Learning of Places from Range Data using Adaboost. In *Proc. of the Int. Conference on Robotics and Automation (ICRA)*, pages 1742–1747, 2005.
25. Jirí Matas and Ondrej Chum. Randomized RANSAC with sequential probability ratio test. In Songde Ma and Heung-Yeung Shum, editors, *Proc. IEEE International Conference on Computer Vision (ICCV)*, volume II, pages 1727–1732, New York, USA, October 2005. IEEE Computer Society Press.
26. M. Meilă and M.I. Jordan. Learning with mixtures of trees. *The Journal of Machine Learning Research*, 1:1–48, 2001.
27. G. Monteiro, C. Premebida, P. Peixoto, and U. Nunes. Tracking and Classification of Dynamic Obstacles Using Laser Range Finder and Vision. In *in Workshop on "Safe Navigation in Open and Dynamic Environments - Autonomous Systems versus Driving Assistance Systems" at the IEEE/RSJ Int. Conference on Intelligent Robots and Systems (IROS)*, 2006.
28. K. P. Murphy, Y. Weiss, and M. I. Jordan. Loopy belief propagation for approximate inference: An empirical study. In *In Proc. of Uncertainty in AI*, pages 467–475, 1999.
29. David Nistér. Preemptive RANSAC for live structure and motion estimation. *Machine Vision and Applications*, 16(5):321–329, 2005.
30. J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.
31. J. Ponce, M. Hebert, C. Schmid, and A. Zisserman, editors. *Toward Category-Level Object Recognition*. Lecture Notes in Computer Science , Vol. 4170, 2007.
32. Arthur R. Pope. Model-Based Object Recognition - A Survey of Recent Research. Technical Report TR-94-04, The University of British Columbia, 1994.
33. I. Posner, D. Schroeter, and P. Newman. Online generation of scene descriptions in urban environments. *Robot. Auton. Syst.*, 56(11):901–914, 2008.
34. I. Posner, D. Schröter, and P. Newman. Using Scene Similarity for Place Labelling. In *Proc. of the Int. Symposium on Experimental Robotics (ISER)*, 2006.
35. Ingmar Posner, Mark Cummins, and Paul Newman. Fast probabilistic labeling of city maps. June 2008.
36. A. Ranganathan and F. Dellaert. Semantic Modeling of Places using Objects. In *Proc. of Robotics: Science and Systems*, Atlanta, GA, USA, June 2007.
37. C. Schmid. Constructing models for content-based image retrieval. In *IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION*, volume 2, 2001.
38. 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.
39. S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, K. Lau, C. Oakley, M. Palatucci, V. Pratt, P. Stang, S. Strohband, C. Dupont, L.-E. Jendrossek, C. Koelen, C. Markey, C. Rummel, J. van Niekerk,

E. Jensen, P. Alessandrini, G. Bradski, B. Davies, S. Ettinger, A. Kaehler, A. Nefian, and P. Mahoney. Winning the DARPA Grand Challenge. *Journal of Field Robotics*, 2006.

40. Phil Torr and Andrew Zisserman. MLESAC: A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding*, 78:138–156, 2000.
41. R. Triebel, K. Kersting, and W. Burgard. Robust 3D Scan Point Classification using Associative Markov Networks. In *"Proc. of the Int. Conference on Robotics and Automation(ICRA)"*, 2006.
42. J. Weingarten, G. Gruener, and R. Siegwart. A Fast and Robust 3D Feature Extraction Algorithm for Structured Environment Reconstruction. In *Proc. of the 11th Int. Conference on Advanced Robotics (ICAR)*, 2003.
43. C. Wellington, A. Courville, and A. Stentz. Interacting Markov Random Fields for Simultaneous Terrain Modeling and Obstacle Detection. In *Proc. of Robotics: Science and Systems*, 2005.
44. J. S. Yedidia, W. T. Freeman, and Y. Weiss. Generalized belief propagation. In *In NIPS 13*, pages 689–695. MIT Press, 2001.