A Spatio-Temporal Probabilistic Model for Multi-Sensor Multi-Class Object Recognition

Bertrand Douillard* Dieter Fox† Fabio Ramos* Roman Katz * Hugh Durrant-Whyte *

* *ARC Centre of Excellence for Autonomous Systems*
Australian Centre for Field Robotics
University of Sydney
Sydney, NSW, Australia

† *Dept. of Computer Science & Engineering*
University of Washington
Seattle, WA, USA

***Abstract*— This paper presents a general probabilistic framework for multi-sensor multi-class object recognition based on Conditional Random Fields (CRFs). The proposed methodology learns a model for spatial and temporal relationships which is able to integrate arbitrary sensor information by automatically extracting features from data. Spatial and temporal reasoning are unified in the framework as various instances of the more general structured classification problem. We demonstrate the benefits of a spatio-temporal probabilistic model for the problem of detecting seven classes of objects in an urban environment using laser and vision data. We describe how this framework can be used with partially labeled data, thereby significantly reducing the burden of manual data annotation. Finally, we show how this framework can be applied to the generation of large-scale semantic maps.**

I. Introduction

Reliable object recognition is an important step for enabling robots to reason and act in the real world. A high-level perception model able to integrate multiple sensors can significantly increase the capabilities of robots in tasks such as obstacle avoidance, mapping, and tracking.

Although object recognition has been a major research topic in the computer vision community, direct application of the algorithms to robotics problems is not always feasible. There are three main reasons for this. First, robotics applications require real-time object recognition. While real-time algorithms for face detection do exist [40], real-time recognition of general objects is still under development. Second, robots can be equipped with different types of sensors including ranging and visual. The integration of these sensors for object recognition can complement the visual information by providing additional geometric properties of observed objects. Multi-sensor fusion for object recognition is thus a desirable feature to be considered in robotics perception. Third, when navigating, robots observe the same objects from different locations and at different times. This is conceptually different from most object recognition algorithms in computer vision where observations are considered independent. Probabilistic models able to integrate observations at different times and positions are expected to perform more robustly in complex outdoor environments with variable illumination and multi-scale observations.

In order to address these three aspects of the object recognition problem, several groups in the robotics community have developed techniques in which classification is integrated into a mapping solution [3, 9, 20, 23, 29]. Such representations can be extremely valuable since they enable robots to perform high-level reasoning about their environments and the objects therein. For instance, in search and rescue tasks, a mobile robot that can reason about objects such as doors, and places such as rooms, is able to coordinate with first responders in a much more natural way. It can accept commands such as “Search the room behind the third door on the right of this hallway”, and send information such as “There is a wounded person behind the desk in that room” [16]. As another example, consider autonomous vehicles navigating in urban areas. While the recent success of the DARPA Urban Challenge [5] demonstrates that it is possible to develop autonomous vehicles that can navigate safely in constrained settings, successful operation in more realistic, populated urban areas requires the ability to distinguish between objects such as cars, people, buildings, trees, and traffic lights.

As a step towards the long-term goal of equipping a robot with the ability to understand its environment, we propose a classification framework based on Conditional Random Fields (CRFs). CRFs are discriminative models for classification of structured (dependent) data [17]. We show how CRFs provide a flexible framework in which different types of spatial and temporal dependencies can be represented demonstrating that these probabilistic models stand as a general solution to the problem of classification in robotics applications.

The flexibility of CRF-based representations is presented using various models of increasing complexity integrating 2D laser scans and imaging data. We start with a simple chain CRF formed by linking consecutive laser beams in the scans. This configuration models the geometrical structure of a scan and captures the typical shapes of objects. We then incorporate temporal information by adding links between consecutive laser scans based on the correspondences obtained by a scan matching algorithm. This leads to a network in which estimation is equivalent to a filtering algorithm, thus taking into account temporal dependencies in addition to spatial information. This network, and its associated estimation machinery, also have the particularity to allow temporal smoothing as the network grows with the registration of incoming scans. Finally, we show that a CRF can be used to capture the various structures characterizing a geometric map. This involves defining a network on a set of already aligned laser scans and running estimation as a batch process. Via the obtained map sized network, classification is performed jointly across the whole laser map and can, in turn, exploit the larger geometric structures in order to improve local classification.

By building on the recently developed Virtual Evidence Boosting (VEB) algorithm [18], the algorithm used to train the various models is able to automatically select features during the learning phase. The expert knowledge about the problem is encoded as a selection of features capturing particular properties of the data such as geometry, color and texture. Given a labeled training set, VEB computes weights for each of these features according to their importance in discriminating the data. Additionally, an extension of VEB for semi-supervised learning is presented to address partially labeled datasets.

This paper is organized as follows. Related work is discussed first, in Section II. Section III provides a short introduction to CRFs as well as a description of the associated learning and inference techniques. The various models at the core of the proposed framework are presented in Section IV. This is followed by a description of the features used for classification. Experimental results are presented in Section VI. Finally, we conclude in Section VII.

II. Related Work

Object recognition is a long-standing problem in robotics and computer vision. Most of the approaches in computer vision aim at recognizing objects from single images. Classifiers are trained on labeled data and used to either classify images as containing or not an instance of the object, or to segment the object in the image [10, 38, 40]. In robotics, the problem is different. Recognition can be performed in a sequence of images, in many cases combined with other sensor modalities.

Within the robotics community, recent developments have created representations of the environment integrating more than one sensor modality. In [26], a 3D laser scanner and loop closure detection based on photometric information are brought together into the Simultaneous Localization and Mapping (SLAM) framework. This approach does not generate a semantic representation of the environment which can be obtained from the same multi-modal data using the approach proposed here.

In [32], a robust landmark representation is created by probabilistic compression of high-dimensional vectors containing laser and camera information. This representation is used in a SLAM system and updated on-line when a landmark is re-observed. However, it does not reason about landmark classes and therefore does not support the higher-level object detection described in this work.

Object recognition based on laser and video data has been demonstrated in [24]. Using a sum rule, this approach combines the outputs of two classifiers, each of them being assigned to the processing of one type of data. More recently, Posner and colleagues combine 3D laser range data with camera information to classify surface types such as brick, concrete, grass, or pavement in outdoor environments [30, 31]. The authors classify each laser scan return independently which can disregard important neighborhood information. As other researchers have shown, classification results can be improved by jointly classifying laser beams using techniques such as associative Markov networks [39] or conditional random fields [8].

Structured classification is also demonstrated in [29] where objects are classified based on monocular imagery and 3D laser data. This approach does not incorporate temporal information and while it is designed to handle multi-modal data, user-specified inputs are required for each modality. A structured model is also used in [2] where the segmentation of objects from 3D laser scans is based on a Markov Random Field. The model is trained discriminatively using a max-margin objective function. The features used were simple geometric features capturing plane properties of groups of points. The authors considered four classes: ground, building, tree and shrubbery. Friedman and colleagues introduced Voronoi Random Fields, which generate semantic place maps of indoor environments by labeling the points on a Voronoi graph of a laser map using conditional random fields [14].

The particularity of this work is to combine multi-modal data fusion, structured reasoning and temporal estimation into one class of models. This paper builds on previous work by addressing the classification problems tackled in [8, 9] with a single modeling approach. Its key contribution is the presentation of a probabilistic framework based on CRFs which unifies spatial and temporal reasoning as various instances of the more general structured classification problem.

III. Conditional Random Fields

This section provides a brief description of conditional random fields (CRFs) and their associated learning and inference techniques.

A. Model Description

Conditional random fields (CRFs) are undirected graphical models developed for labeling sequence data [17]. CRFs directly model $p(\mathbf{x}|\mathbf{z})$, the conditional distribution over the hidden variables $\mathbf{x}$ given observations $\mathbf{z}$. CRFs factorize $p(\mathbf{x}|\mathbf{z})$ as:

$$p(\mathbf{x} \mid \mathbf{z}) = \frac{1}{Z(\mathbf{z})} \prod_{c \in \mathcal{C}} \phi_c(\mathbf{z}, \mathbf{x}_c), \tag{1}$$

where $Z(\mathbf{z}) = \sum_{\mathbf{x}} \prod_{c \in \mathcal{C}} \phi_c(\mathbf{z}, \mathbf{x}_c)$ is the normalizing partition function. $\mathcal{C}$ is the set of cliques in the CRF graph. The ϕ_c are *clique potentials*, which are functions that map variable configurations to non-negative numbers. Intuitively, these potentials capture the compatibility among the variables in the clique: the larger the potential value, the more likely the configuration. Potentials are constrained to log-linear functions, and learning a CRF requires learning the weights of these functions.

The proposed framework employs pairwise CRFs, a particular type of CRFs which can be formulated as follows:

$$p(\mathbf{x}|\mathbf{z}) = \frac{1}{\mathbf{Z}} \exp\left(w_A \sum_i A(x_i, \mathbf{z}) + w_I \sum_e I(x_{e1}, x_{e2}, \mathbf{z}) \right) \tag{2}$$

Here, the term $1/\mathbf{Z}$ is the normalization factor, i ranges over the set of nodes and e over the set of edges. The functions A and I are the association and interaction potentials, respectively. An association potential A is a classifier which estimates the object type of node x_i using the set of observations $\mathbf{z}$ but does not take into account information contained in the structure of the neighborhood. An interaction potential I is a function associated to each edge e of the CRF graph, where x_{e1} and x_{e2} are the nodes connected by edge e. Intuitively, interaction potentials measure the compatibility between neighboring nodes and act as smoothers by correlating the estimation across the network.

B. Inference

Inference in CRFs can estimate either the marginal distribution of each hidden variable $\mathbf{x}_i$ or the most likely configuration of all hidden variables $\mathbf{x}$ (*i.e.*, MAP estimation), as defined in Eq. 1. Both tasks can be solved using belief propagation (BP) [28], which works by sending local messages through the graph structure of the model. Each node sends messages to its neighbors based on the clique potentials and on the messages it receives.

BP provides exact results in graphs with no loops, such as trees or polytrees. However, since the models used in our approach contain various loops due to temporal relationships, we apply loopy belief propagation (loopy BP), an approximate inference algorithm that is not guaranteed to converge to the correct probability distribution [25]. Fortunately, in our experiments, convergence of loopy BP was observed in few iterations for most cases. An empirical convergence analysis is provided in Section VI-E.

C. Learning via Virtual Evidence Boosting

Learning a CRF model involves determining the quantities A, I, w_A and w_I in Eq. 2. CRFs are trained discriminatively by maximizing the conditional likelihood (Eq. 1) of labeled training data. This optimization is typically performed by gradient-based techniques such as L-BFGS, where gradients are computed using inference in the CRF model [35]. In order to avoid computationally complex inference for gradient computations, several researchers applied pseudo-likelihood training, which does not require running inference [19].

While CRFs can handle high-dimensional continuous and discrete features, the integration of continuous features is not straightforward. This is due to the fact that the incorporation of raw, continuous features in CRFs is similar to uni-modal Gaussian likelihood models in generative approaches such as hidden Markov models. Such simple likelihoods are not well suited to model more complex, multi-modal features and sensor data. Recently, researchers have applied boosting in order to discretize continuous features into binary threshold functions, called decision stumps [14]. The thresholds are learned by minimizing an exponential loss function of the training data [12]. The decision stumps are then used as binary features in a CRF, and the weights for these features are learned using regular CRF training [14].

More recently, Liao and colleagues introduced virtual evidence boosting (VEB), which incorporates feature discretization into CRF training [18]. VEB jointly learns an appropriate discretization of continuous features, the weights of these features, and the weights of neighborhood potentials of the CRF. In essence, this is obtained by performing boosting on both the features and the neighborhood potentials of the CRF. VEB has demonstrated superior performance on both synthetic and real data. Furthermore, the automatic feature discretization makes VEB extremely flexible and allows the incorporation of arbitrary, continuous and discrete features. Since model flexibility is crucial in the context of our object recognition task, we chose to use VEB for learning the parameters of our CRFs.

Through VEB, a CRF model can not only be learnt with fully labeled data but also with partially labeled data. This is achieved by disregarding the unlabeled data when learning the logitboost classifier[1] which plays the role of the association potential A in Eq. 2. However, once learnt, this association potential A can be applied to every single node, whether it is labled or not, in order to generate a local distribution at each node. This local belief is then propagated in the network via BP. In that sense, unlabeled nodes do not contribute to the learning of the interaction potential A but do contribute to the learning of the quantities I, w_A and w_I in Eq. 2.

The slightly modified VEB training is described in Algorithm 1. As specified by the condition at line 3, the local logitboost learning does not use unlabeled data. However, the learned logitboost classifier is applied at all the network nodes (labeled and unlabeled) as the association potential A which generates each node's local estimate. These local estimates are then propagated in the network via BP (line 6 of the algorithm) to provide the joint probability over the set of hidden states $\mathbf{x}$.

Algorithm 1 SemiSupervisedVEB

Input: CRF connectivity structure, M number of rounds of VEB, training data (x_i,$\mathbf{z_i}$), for unlabeled nodes x_i=nan

Output: F

```
1  for $m = 1, \ldots, M$
2      for $i = 1, \ldots, N$
3          if $x_i \neq \mathrm{nan}$
4              Compute boosting weights $w_A(x_i)$;
5              Compute boosting working response $r_i(x_i)$;
6      Run BP using F to obtain virtual evidences $\{\mathbf{ve_i}\}$;
7      Compute $f_m(\{\mathbf{ve}_i, \mathbf{z_i}\})$ by least square regression
7      of $r_i(x_i)$ to $\{\mathbf{ve_i}, \mathbf{z_i}\}$ using weights $w_A(x_i)$;
8      Update $F = F + f_m$;
```

A semi-supervised version of VEB was also proposed in [22]. The main difference with the approach proposed here is in the formulation of the conditional likelihood of the data which is optimized during learning. The algorithm above maximizes the standard conditional likelihood in Eq. 1 while the

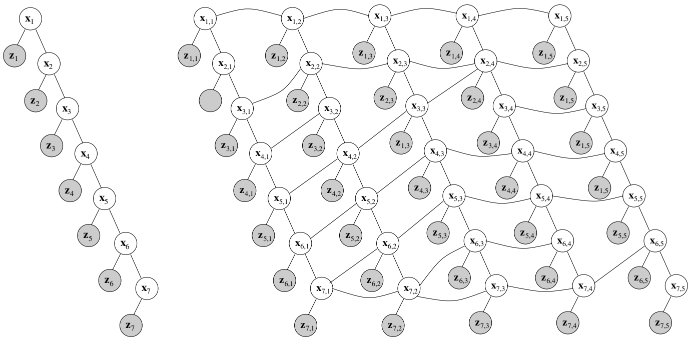
[1] Logitboost is the version of boosting used in the VEB algorithm.

formulation in [22] involves an additional term representing the conditional entropy of the unlabeled data. As mentioned by the authors, one drawback of this latter formulation is that the resulting objective function is no longer convex.

IV. From Laser Scans to Conditional Random Fields

This section shows how the connectivity structure of a CRF can be generated from laser data. Each node of the resulting networks corresponds to a laser return whose hidden state ranges over the objects types: car, trunk, foliage, people, wall, grass and other (the class other representing any other type of objects).

This section is organized according to the increasing complexity of the networks. The representation of spatial relationships is first introduced by modeling single laser scans as chain CRFs. Then, consecutive scans are connected according to their alignment to model temporal relationships and effectively implement operations such as filtering and smoothing. Finally, three types of networks for the generation of semantic maps are described.


$\mathbf{x}_1$
$\mathbf{z}_1$
$\mathbf{x}_2$
$\mathbf{z}_2$
$\mathbf{x}_3$
$\mathbf{z}_3$
$\mathbf{x}_4$
$\mathbf{z}_4$
$\mathbf{x}_5$
$\mathbf{z}_5$
$\mathbf{x}_6$
$\mathbf{z}_6$
$\mathbf{x}_7$
$\mathbf{z}_7$
$\mathbf{x}_{1,1}$ $\mathbf{x}_{1,2}$ $\mathbf{x}_{1,3}$ $\mathbf{x}_{1,4}$ $\mathbf{x}_{1,5}$
$\mathbf{z}_{1,1}$ $\mathbf{z}_{1,2}$ $\mathbf{z}_{1,3}$ $\mathbf{z}_{1,4}$ $\mathbf{z}_{1,5}$
$\mathbf{x}_{2,1}$ $\mathbf{x}_{2,2}$ $\mathbf{x}_{2,3}$ $\mathbf{x}_{2,4}$ $\mathbf{x}_{2,5}$
$\mathbf{z}_{2,2}$ $\mathbf{z}_{2,3}$ $\mathbf{z}_{2,4}$ $\mathbf{z}_{1,5}$
$\mathbf{x}_{3,1}$ $\mathbf{x}_{3,2}$ $\mathbf{x}_{3,3}$ $\mathbf{x}_{3,4}$ $\mathbf{x}_{3,5}$
$\mathbf{z}_{3,1}$ $\mathbf{z}_{3,2}$ $\mathbf{z}_{1,3}$ $\mathbf{z}_{3,4}$ $\mathbf{z}_{1,5}$
$\mathbf{x}_{4,1}$ $\mathbf{x}_{4,2}$ $\mathbf{x}_{4,3}$ $\mathbf{x}_{4,4}$ $\mathbf{x}_{4,5}$
$\mathbf{z}_{4,1}$ $\mathbf{z}_{4,2}$ $\mathbf{z}_{4,3}$ $\mathbf{z}_{4,4}$ $\mathbf{z}_{1,5}$
$\mathbf{x}_{5,1}$ $\mathbf{x}_{5,2}$ $\mathbf{x}_{5,3}$ $\mathbf{x}_{5,4}$ $\mathbf{x}_{5,5}$
$\mathbf{z}_{5,1}$ $\mathbf{z}_{5,2}$ $\mathbf{z}_{5,3}$ $\mathbf{z}_{5,4}$ $\mathbf{z}_{5,5}$
$\mathbf{x}_{6,1}$ $\mathbf{x}_{6,2}$ $\mathbf{x}_{6,3}$ $\mathbf{x}_{6,4}$ $\mathbf{x}_{6,5}$
$\mathbf{z}_{6,1}$ $\mathbf{z}_{6,2}$ $\mathbf{z}_{6,3}$ $\mathbf{z}_{6,4}$ $\mathbf{z}_{6,5}$
$\mathbf{x}_{7,1}$ $\mathbf{x}_{7,2}$ $\mathbf{x}_{7,3}$ $\mathbf{x}_{7,4}$ $\mathbf{x}_{7,5}$
$\mathbf{z}_{7,1}$ $\mathbf{z}_{7,2}$ $\mathbf{z}_{7,3}$ $\mathbf{z}_{7,4}$ $\mathbf{z}_{7,5}$

(a) (b)

Fig. 1. (a) Graphical model of a chain CRF for a single time slice object recognition. Each hidden node $\mathbf{x}_i$ represents one (non out of range) return in a laser scan. The nodes $\mathbf{z}_i$ represent the features extracted from the laser scan and the corresponding camera image (observations). (b) Graphical model of the spatio-temporal model. Nodes $\mathbf{x}_{i,j}$ represent the i-th laser return observed at time j. Temporal links are generated between time slices based on the ICP matching algorithm.

A. Spatial Reasoning

CRFs were selected as the basis for the proposed framework due to their aptitude to encode structure in the classification process. By "structure" we refer here to two different types of dependencies: spatial and temporal. Spatial dependencies come from the natural organisation of the data in subsets of samples with the same label: spatially close samples are likely to have to same label. Temporal dependencies come from overlapping observations performed at successive times: samples generated by the same object and acquired at successive times will be dependent. In the context of a CRF network, these different types of dependencies are represented by various sets of links.

From a classification point of view, the structure of urban environments is characterized by the proximity of laser returns in the same objects. Thus, the first representation aims at capturing such spatial dependencies. This is obtained by instantiating the CRF model as a chain network representing a particular laser scan, as illustrated in Fig 1(a). The links of this chain network encode the spatial dependencies between successive returns.

By performing probabilistic inference, the classes of all the laser returns in the scan are jointly estimated. Local observations are passed onto each node via the association potentials A (Eq. 2) and the resulting local estimates are propagated in the network via the pairwise potentials I.

B. Temporal Reasoning

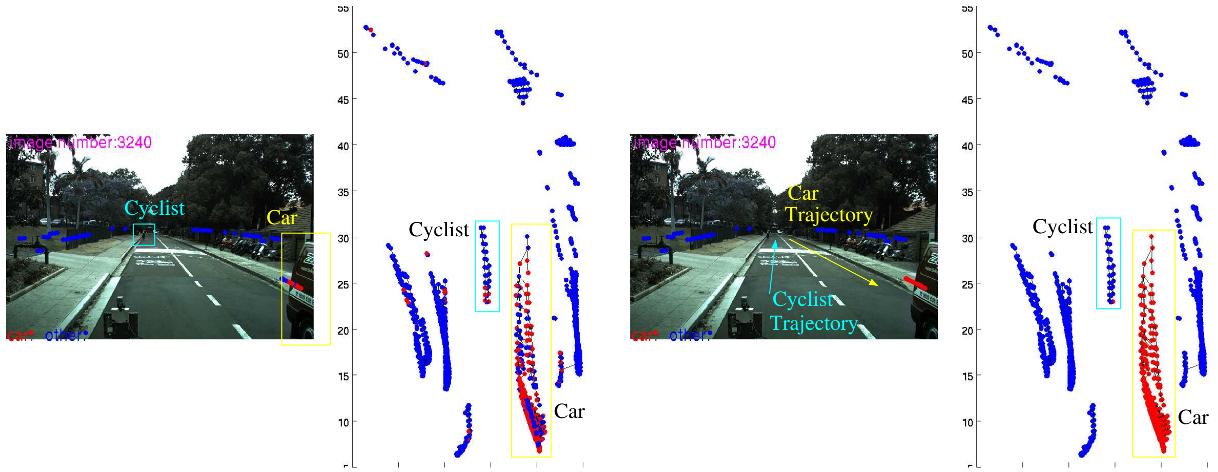
Due to the sequential nature of robotics applications, a substantial amount of information can be gained by taking into account temporal dependencies. Using the same elementary components of CRFs, i.e. nodes and links, we now build a model achieving temporal smoothing in addition to exploiting the geometric structure of laser scans. This model is illustrated in Fig. 1(b).

In this work, the links modeling the temporal dependencies are instantiated such that they represent the associations obtained by the Iterative Closest Point (ICP) matching algorithm [42]. The resulting network connects successive chain networks and is characterized by a cyclic topology. This network models spatial correlations via links connecting the nodes within one scan and temporal correlations via links connecting the successive chain networks.

Corresponding to different variants of temporal state estimation, our spatio-temporal model can be deployed to perform three types of inference:

- Off-line smoothing: All scans in a temporal sequence are connected using ICP. Loopy BP is then run in the whole network to estimate the class of each laser return in the sequence. During loopy BP, each node sends to its neighbors the messages through structural and temporal links (vertical and horizontal links in Fig. 1(b), respectively).
- On-line fixed-lag smoothing: Here, scans are added to the model in an on-line fashion. To label a specific scan, the system waits until a certain number of future scans becomes available. It then runs loopy BP which combines past and future observations to estimate the network's labels.
- On-line filtering: In this case the spatio-temporal model includes scans up to the current time slice resulting in an estimation process which integrates prior estimates.

An example of on-line fixed-lag smoothing is presented in Fig. 2. It can be seen in this figure that the sets of nodes corresponding to the car and the cyclist are correctly classified when a CRF is used to integrate spatial and temporal information. The estimates given by local estimation, *i.e.*, estimation which does not take into account the information provided by the network links, are only partially correct.



(a) Non-Structured Classification

(b) Structured Classification

Fig. 2. Example of classification improvements obtained with the spatio-temporal CRF. Fig. (a) shows the estimates obtained with local classification (*i.e.*, using only the A functions in Eq. 2). Fig. (b) shows the estimates obtained using a CRF as the model displayed in Fig. 1(b). The right part of each figure shows a sequence of laser scans projected in a global frame. The estimates are indicated by the color of each return: red for car and blue for other. The black links represent the temporal edges of the underlying network. The left part of each figure displays the last image of the sequence as well as the projection in the image of the corresponding laser returns. In the sequence used to generate this figure, a car is moving toward our vehicle and a cyclist is moving away from our vehicle. Based on local classification (Fig. a), some of the returns are mis-classified since all the returns associated to the cyclist should be blue and all the returns associated to the car should be red. Based on structured classification (Fig. b), only a very small number of returns are mis-classified.

Since these type of spatio-temporal network contains cycles, inference is based on loopy BP and is as a result only approximate. Alternatives to approximate techniques are discussed in Section IV-C.3.

C. Map Building

We now show how a larger scale CRF network can be built in order to generate semantic maps. The proposed map building approach requires as an input a set of already aligned 2D laser scans. In our implementation, the ICP algorithm was used to perform scan registration. However, in spatially more complex data sets containing loops, consistently aligned scans can be generated using various existing SLAM techniques [37].

In this section, we present three types of CRFs which will be compared to better understand how to model spatial dependencies. We explain how the three different models can be instantiated from aligned laser data and indicate which inference technique is used in each case. Training of these three networks is performed with partially labeled data. As in the previous models, the hidden states represent the object types of the laser returns.

1) Delaunay CRF: In this first type of network, the connections between the nodes are obtain ed using the Delaunay triangulation procedure [7] which efficiently finds a triangulation with non-overlapping edges. The system then removes links which are longer than a pre-defined threshold (50 cm in our application) since distant nodes are not likely to be strongly correlated. The resulting network is displayed as a set of blue edges in Fig. 4.

Since a Delaunay CRF contains cycles, inference is performed with loopy BP.

2) Delaunay CRF with link selection: Structured classification as performed by CRFs is expected to improve on local classification since independence is not assumed, *i.e.*, neighborhood information is modeled through interaction potentials. However, as illustrated by the experimental results, the Delaunay CRF previously described does not improve on local classification. A too coarse modeling of the spatial correlations is responsible for this result. The terms I of Eq. 2 are learnt in this first type of network as a constant matrix instantiated at each of the links. This gives the network a smoothing effect on top of the local classification. Since all the links are represented with the same matrix, only one type of node-to-node relationship is encoded, for example: "neighbor nodes should have the same label". While this type of links may be appropriate for modeling a single scan or in very structured parts of the environment, it may over-smooth the estimates in areas where the density of objects increases.

In order to model more than one type of node-to-node relationships, the network is augmented with an additional node T for every pair of nodes $\{x_i, x_j\}$ as displayed in Fig. 3. The state of this node specifies which type of link is instantiated. For this second type of network, we consider two types of links encoding the following node-to-node relationships: (1) neighbor nodes have the same label, (2) neighbor nodes have a different label. Node T receives an observation S which is the output of a logitboost classifier learned to estimate whether node x_i and x_j are similar based on their respective local observation z_i and z_j. The observation S is a direct observation of the state of node T.

Since this second type of network contains loops, inference is also performed using loopy BP.

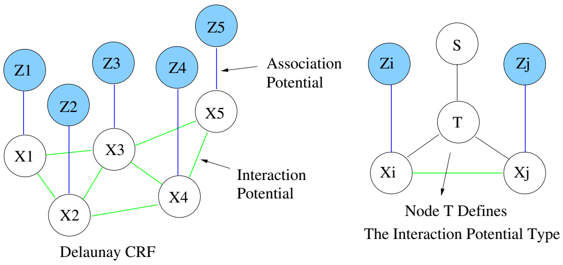


Fig. 3. The Delaunay CRF is presented on the left. On the right, a representation of the additional infrastructure required in a Delaunay CRF to perform link selection.

3) Tree-based CRF: The previous two types of networks contain cycles, which implies the use of an approximate inference algorithm. We now present a third type of networks which is cycle free. To design non-cyclic networks we start from the following observation: laser returns in a scan map are naturally organized into clusters. These clusters can be identified by analysising the connectivity of the Delaunay graph and finding its disconnected sub-components. Disconnected components appear when removing longer links of the original triangulation. In Fig. 4, the extracted clusters are indicated by green rectangles.

Once the clusters are identified, the nodes of a particular cluster are connected by a tree of depth one. A root node is instantiated for each cluster and each node in the cluster becomes a leaf node. The trees associated to the clusters in Fig. 4 are represented by green volumes. A tree-based CRF does not encode node-to-node smoothing but rather performs smoothing based on the identified clusters of laser returns. The root node does not have an explicit state. Its role is to allow the instantiation of a network which does not contains cycles and permit the use of an exact inference technique: with this third type of network, belief propagation is used for inference.

The possibility of using exact inference is a strong advantage since in the case of approximate inference (based on loopy BP for example) the convergence of the algorithm is not guarantied.

As suggested in [25], while convergence of loopy BP in cyclic networks is not proven, it can be experimentally checked. To evaluate the convergence of the inference procedure in the two previous networks, an empirical convergence analysis is presented in Section VI-E.

V. Features For Object Recognition

As formulated in Eq. 2, the computation of the posterior probability requires the set of observations $\mathbf{z}$. In this work, $\mathbf{z}$ consists of high-dimensional feature vectors $\mathbf{f}$ computed for each scan return. $\mathbf{f}$ results from the concatenation of two types of features which are geometric features and visual features:

$$\mathbf{f} = [\mathbf{f}_{\text{geo}}, \mathbf{f}_{\text{visu}}], \tag{3}$$

Geometric features are first described. We then show how visual features can be extracted via the registration of the laser data with respect to the imagery. Finally, we explain how the use of logitboost (as a component of VEB), allows the selection of the most effective features with respect to the classification task.

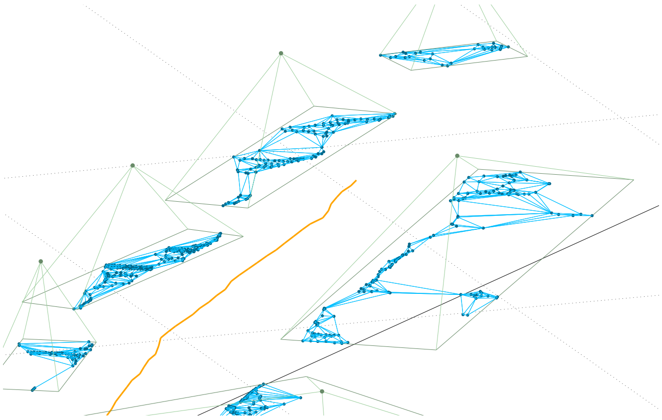


Fig. 4. Representation of a Tree based CRF in one region of a graph generated from data. The trajectory of the vehicle is displayed in orange. Laser returns are instantiated as nodes in the network and connected using the Delaunay triangulation. Nodes and edges are plotted in dark and light blue, respectively. Identified clusters are indicated by the green rectangles while root nodes are plotted in green. Root nodes are connected to all nodes in the cluster but for clarity this is represented by a rectangle enclosing the cluster.

A. Geometric Features

Geometric features capture the shape of objects in a laser scan. The geometric feature vector computed for one laser return has a dimensionality of 231 and results from the concatenation of 38 different multi dimensional features. We present here only the features which are the most useful for classification and explain in Section V-C how such a ranking of the features can be obtained. Some of these 38 features are the following:

$$\mathbf{f}_{\text{geo}} = [\mathbf{f}_{\text{nAngle}}, \mathbf{f}_{\text{minAngle}}, \mathbf{f}_{\text{cSplineFit}}, \mathbf{f}_{\text{cEigVal1}}, \mathbf{f}_{\text{maxFilter}}, \ldots], \tag{4}$$

The features $\mathbf{f}_{\text{nAngle}}$ and $\mathbf{f}_{\text{minAngle}}$ respectively refer to the norm and the minimum of a multi-dimensional angle descriptor $\mathbf{f}_{\text{angle}}$ which has been designed for this application. Its k^{th} dimension is computed as:

$$\mathbf{f}_{\text{angle}}(k) = \|\angle\left(\overline{r_{i-k}r_i}, \overline{r_i r_{i+k}}\right)\|, \tag{5}$$

where r_i refers to the i^{th} return of the scan being processed and k varies from -10 to $+10$. The dimensionality of both $\mathbf{f}_{\text{nAngle}}$ and $\mathbf{f}_{\text{minAngle}}$ features is one. In the various models learnt across our experiments, the features computed from the $\mathbf{f}_{\text{angle}}$ feature were amongst the best for the recognition of the classes tree trunk and pedestrian. In the case of these two classes, these features capture typical curvilinear shapes when for example the scan hits these objects at about one meter above the ground.

The features $\mathbf{f}_{\text{cSplineFit}}$ and $\mathbf{f}_{\text{cEigVal1}}$ characterize the shape of a cluster of returns. Clusters are extracted within one scan based on a simple distance criteria: returns closer than a threshold (we used one meter in our applications) are associated to the same cluster. Based on the identified clusters, various quantities are computed. Feature $\mathbf{f}_{\mathrm{cSplineFit}}$ is obtained as the error of the fit of a 2D spline to the cluster of returns. Feature $\mathbf{f}_{\mathrm{cEigVal1}}$ is the largest eigen value of the covariance matrix describing the cluster. While not being ranked amongst the very first features, cluster based features turned out to be useful in classifying all of the seven classes we have considered in this work. Note that all the returns of one cluster receive the same cluster features.

The feature $\mathbf{f}_{\mathrm{maxFilter}}$ is obtained as the maximum response of a filter run in a window centred on a given return. This filter is essentially a low pass discrete filter processing a scan represented as a sequence of angles. This filter provides a multi-dimensional filter feature whose various dimensions have proven useful in detecting the class car and the class pedestrian.

While our approach for feature design is not related to the work presented in [4], the underlying philosophy is similar. Future work will investigate some of the features proposed in [4] for sub-maps matching in order to use them in a classification system.

B. Visual Features

As will be further detailed in the next section, a CRF learned with a logitboost based algorithm can not only integrate geometric information but also any other type of data and, in particular, visual features extracted from monocular color images. As a consequence, the proposed framework also includes procedures to extracts visual features. A region of interest (ROI) is defined around the projection of each laser return into the corresponding image and a set of features is computed within this ROI. The parameters required to carry out the projection are defined through the camera laser calibration procedure developed in [41]. The size of the ROI is changed depending on the range of the return. This provides a mechanism to deal with changes in scales across images. It was verified that the use of a size varying ROI improves classification accuracy by 4%.

In order to obtain a visual feature vector $\mathbf{f}_{\mathrm{visu}}$ of constant dimensionality despite a size varying ROI, we design vision features which are independent of the patch's size. This is achieved by features which are distributions (e.g. an histogram with a fixed number of bins) and whose dimensionality is constant (e.g. equal to the number of bins in the histogram). A larger ROI leads to a better sampled distribution (e.g. a larger number of samples in the histogram) while the actual feature dimensionality remains invariant.

The overall visual feature vector $\mathbf{f}_{\mathrm{visu}}$ associated to each return has a dimensionality of 1239 and results from the concatenation of 51 multi-dimensional features computed in the ROI. We only describe here the subset of features which turned out to be the most useful:

$$\mathbf{f}_{\mathrm{visu}} = [\mathbf{f}_{\mathrm{pyr}}, \mathbf{f}_{\mathrm{hsv}}, \mathbf{f}_{\mathrm{rgb}}, \mathbf{f}_{\mathrm{hog}}, \mathbf{f}_{\mathrm{haar}}, \mathbf{f}_{\mathrm{lines}}, \mathbf{f}_{\mathrm{sift}}, \ldots] \qquad (6)$$

$\mathbf{f}_{\mathrm{pyr}}$ contains texture information encoded as the steerable pyramid [34] coefficients of the ROI as well as the minimum and the maximum of these coefficients. These extrema are useful in classifying cars which from most point of views have a relatively low texture maxima due to their smooth surface.

$\mathbf{f}_{\mathrm{hsv}}$ and $\mathbf{f}_{\mathrm{rgb}}$ contain a 3D histogram of the RGB and HSV data in the ROI, respectively. HSV and RGB histograms were selected in the representation of each of the seven classes.

$\mathbf{f}_{\mathrm{hog}}$ are histograms of gradients types of features [27]. These features were selected by the learning algorithm for the modeling of the classes car, pedestrian and grass.

$\mathbf{f}_{\mathrm{haar}}$ contains Haar features computed in the return's ROI according to the integral image approach proposed in [40]. Haar features were useful in classifying the classes tree trunk and foliage.

$\mathbf{f}_{\mathrm{lines}}$ contains a set of quantities describing the lines found by a line detector [1] in the ROI. These quantities include the number of extracted lines, the maximum length of these lines and a flag which indicates whether the line of maximum length is vertical. These features have been useful in classifying all of the seven considered classes.

$\mathbf{f}_{\mathrm{sift}}$ contains the Sift descriptor [21] of the ROI's center as well as the number of Sift features found in the ROI. Sift features were selected during the training of various models to represent the classes grass and other.

C. Feature Selection and Dimensionality Reduction

The VEB algorithm which is used in this work to learn the parameters of the CRF models is based on the logitboost procedure. More precisely, VEB is based on a version of logitboost which uses decisions stumps as weak classifiers. With this type of learning algorithm, the dimensions of the feature vector can be ranked according to their ability to discriminate the various classes in the data. Given one dimension of the feature vector, a decision stump defines one threshold and two values in order to best separate the samples according to their labels and returns a number evaluating how well the data is separated. During the training phase, logitboost builds a decision stump for each dimension of the feature vector and uses the quality estimate of each decision stump to select the feature which best improves the classification accuracy. Keeping track of the successive features selected by logitboost provides a way to identify the most useful features pointed out in the two previous sections.

Feature selection as performed by logitboost based on decision stumps can also be seen as a dimensionality reduction procedure. One hundred rounds of logitboost will result in the selection of one hundred dimensions of the original feature vector. This implies that during the testing phase only these one hundred selected features need to be computed allowing the computations times to be maintained acceptable with respect to real-time requirements; see table VI. In addition, since the dimensions of the feature vector are processed one at a time, no overall normalization of the feature vector is required which is an advantage with respect to more standard dimensionality reduction techniques such as [11, 15, 33, 36].

Another interesting aspect of logitboost is linked to its ability to process multi-modal data. Features computed from an additional modality can be concatenated to the overall feature vector as it was done with laser and vision features in section V-A and V-B. The feature vector in that sense plays the role of a proxy between the various modalities and the learning algorithm.

Logitboost has the advantage of finding the best features within a given set but does not compensate for non informative features. This explains why, as suggested by the previous two sections, the features have to be carefully engineered.

VI. Experimental Results

A. Experimental Setup

Experiments were performed using outdoor data collected with a modified car traveling at a speed of 0 to 40 km/h in a university campus and in the surrounding urban areas. The scenes typically observed contained buildings, walls, cars, bushes, trees and lawn fields. We present results using two different datasets in order to demonstrate the generality of the proposed framework. One dataset was acquired in Sydney, Australia while the other was one acquired in Boston, MA, US. Each of the two datasets approximately corresponds to 20 minutes of logging with a monocular color camera and 2D laser scanners. To acquire the two datasets different vehicles and different sensor brands were used.

The evaluations of the various classifiers are performed using n-fold cross validation. This involves breaking down the dataset into n subsets of equal size in order to train a classifier on n-1 subsets and test it on the remaining subset. Training and testing are repeated n times by isolating each time a different subset for testing. All the results presented below are averaged over the n cross validation tests (n being either 5 or 10 depending on the experiments).

B. Spatial and Temporal Reasoning

1) Sydney dataset: In this first set of experiments we consider two classes: car and other. Seven classes results are presented in Section VI-D. Table I summarizes the experimental results in terms of classification accuracy. The accuracies are given in percentages and computed using 10-fold cross validation on a set of 100 manually labeled scans selected in the Sydney dataset. For each cross validation, different models were trained with 200 iterations of VEB. VEB was computed allowing learning of pairwise relationships only after iteration 100. We found that this procedure increases the weights of local features and improves classification results.

Training set	geo only	visu only	geo+visu	geo+visu
Number of time slices in the model	1	1	1	∓ 10
CRF	68.9	81.8	83.3	88.1
logitboost	67.6	81.5	83.2	×

TABLE I

Classification accuracy for a car detection problem (in %)

The first line of table I indicates the types of features used to learn the classifier. Four different configurations were tested: first using geometric features only, second using visual features only, third using both geometric and visual features, and fourth with geometric and visual features integrated over a period of 10 times slices. The second line of table I indicates the number of time slices in the network used to perform classification. "1" means that a network as presented in Fig. 1(a) was used. "$\mp$ 10 " refers to the classifier shown in Fig. 1(b) instantiated with 10 unlabeled scans prior and posterior to the labeled scan.

Two types of classifiers were used: CRFs and logitboost classifiers. While a CRF takes into account the neighborhood information to perform classification, logitboost learns a classifier that only supports so called independent identically distributed classification, *i.e.*, which does not use neighborhood information [13]. This is equivalent to using only the A functions in Eq. 2. Logitboost is used here for comparison purposes in order to investigate the gain in accuracy obtained with a classifier that takes into account the structure of the scan.

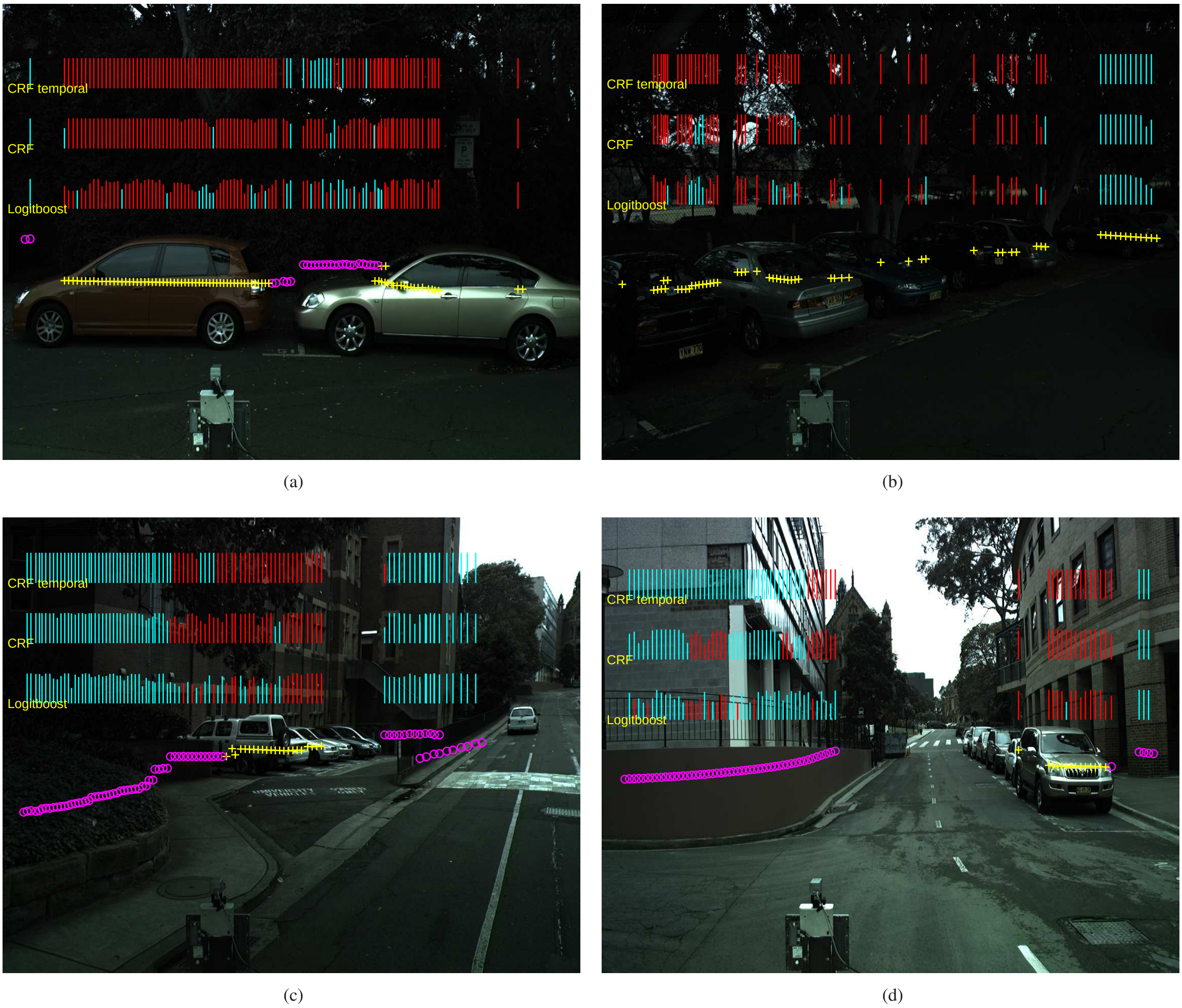
The first three columns of table I show that classification results are improving as richer features are used for learning. It can also be seen that the CRF models consistently lead to slightly more accurate classification.

In addition, as presented in Section IV-B, a CRF model can readily be extended into a spatio-temporal model. The latter leads to an improvement of almost 5% in classification accuracy (right column of table I). This shows that the proposed spatio-temporal model, through the use of past and posterior information, performs better object recognition. The cross in the bottom right of the table refers to the fact that logitboost does not allow the incorporation of temporal information in a straightforward manner.

In order to evaluate the difficulty of the classification task, we also performed logitboost classification using visual Haar features, which results in the well-known approach proposed by Viola-Jones [40]. The accuracy of this approach is 77.09%, which shows that even our single time slice approach (83.26%) outperforms the reference work of Viola & Jones. The improvement in accuracy obtained in our tests comes from use of richer features as well as the aptitude of a CRF to capture neighborhood relationships.

Fig. 5 shows four examples of classification results. It can be seen that the spatio-temporal model gives the best results. While the logitboost classifier tends to alternate correct and incorrect classification across one scan, the ability of the CRF classifiers to capture the true arrangement of the labels (*i.e.*, their structure) is illustrated by the block like distribution of the inferred labels. Figure 5(b) shows the three classifiers failing in a very dark area of the image (right of the image). In the rest of the image which is still quite dark, as well as in images with various lighting conditions (Fig. 5(a), 5(c) and 5(d)) the spatio-temporal model does provide good classification results.

2) Boston dataset: To demonstrate the generality of the proposed framework, the comparisons between the different setups involved in table I were also performed using the Boston dataset. The corresponding results are indicated in table II and were obtained from 5-fold cross validation on a set of



(a)

(b)

(c)

(d)

Fig. 5. Examples of classification results. The label of the returns are displayed with the markers + in yellow and ○ in magenta for the class car and the class other respectively. The height of the bar above each return represents the confidence associated with the inferred label. The color of the bar indicates the inferred label: red means that the inferred label is car and cyan refers to the label other. The classifiers used to generate the different estimates are precised on the left.

400 manually labeled scans. For this second set of tests, the classes of interest were also car and other.

Fig. 6 shows an example of image extracted from the Boston dataset. The laser scanner used to acquire this data is a 3D lidar sensor composed of 64 2D laser scanners positioned on the device with increasing pitch angle. To perform this set of experiments we used the data provided by 6 of these 64 lasers. Unlike in the Sydney dataset, these lasers are downward looking. Examples of scans generated by these 6 lasers are displayed in Fig. 6.

The 6 selected lasers are characterized by a slightly different pitch angle which allows us to build networks from the laser returns such as the one displayed in Fig. 6. While the scan-to-scan links in these networks do not strictly correspond to temporal links (since the 6 lasers fire at the same time) these networks can be thought of as belonging to the category “on-line filtering” described in Section IV-B. Having 6 lasers scanners looking downwards, each of them with a slightly larger pitch angle than the previous one, is approximatively equivalent to having one downward looking scans obtained at 6 consecutive time steps. As a consequence, this setup provides networks of the type “on-line filtering”.

The results in table II show the same trends as table I. As more features are added (moving from the left column to the right column of the table), the classification accuracy increases. Classification accuracy is also increased when using

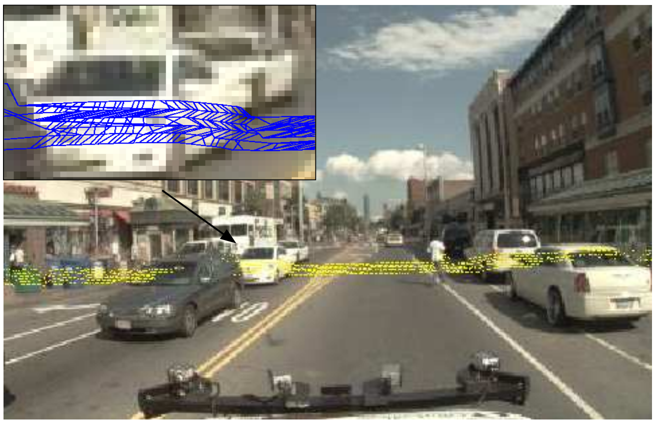


Fig. 6. An example of image from the Boston dataset displayed with the associated projected laser returns (in yellow). A part of the CRF network built from these laser returns in displayed in blue in the inset in the top left corner. The image in this inset corresponds to a magnification of the area indicated by the arrow.

Training set	geo only	visu only	geo+visu	geo+visu
Number of time slices in the model	1	1	1	-5
CRF	81.8	85.0	88.5	90.0
logitboost	81.4	82.6	88.0	×

TABLE II

CLASSIFICATION ACCURACY FOR A CAR DETECTION PROBLEM (IN %)

CRFs, which unlike logitboost, enforces consistency in the sequence of estimates. “-5” in the right column refers to the “on-line filtering” networks which are built by connecting 5 unlabeled scans before each labeled scan. As with the Sydney dataset, temporal information further improves performances.

It is interesting to remark that the classification accuracies achieved on this second dataset for the car detection problem are similar to the ones achieved on the Sydney dataset: the overall accuracy is about 90% in Boston dataset and 88% in the Sydney dataset. The resolution of the imagery as well as the density of the laser returns was quite different between the two datasets: the image size is [240x376] in the Boston dataset and [756x1134] in the Sydney dataset; on average 300 laser returns were available per image in the Boston dataset against 100 in the Sydney dataset. In spite of these differences, the proposed framework provides comparable results which demonstrates its applicability to various datasets.

With respect to the first experiments, the lower resolution of the vision data on one hand, and the larger number of returns available per image on the other hand, lead to a vision classifier with an accuracy (82.6%) only slightly above the one obtained with the laser classifier (81.4%). In the Sydney dataset, a much richer imagery compared to the scan density resulted in 13.9% difference in accuracy between the vision only and the laser only classifiers. As the gap between the information content of the two modalities decreases, the respective classifier displays comparable performances while the proposed framework permits maintaining the overall accuracy by exploiting the best of each modality.

C. Semi-Supervised Learning

Fig. 7 presents car detection results obtained with models learnt on datasets containing a progressively increasing amount of unlabeled data. The Sydney dataset was used for this set of experiments. Fig. 7(a) shows that adding unlabeled data while maintaining the number of labeled returns constant improves classification accuracy. This result experimentally demonstrates the ability of the modified version of VEB presented in algorithm 1 to perform semi-supervised learning. In Fig. 7(b), the total number of scans used is kept constant and the proportion of unlabeled returns is increased. Fig. 7(b) shows that the original accuracy is maintained with only 40% of labeled data. This second result demonstrates that a semi-supervised approach can be expected to critically decrease the required amount of labeled training data thereby reducing the burden associated with manual annotation.

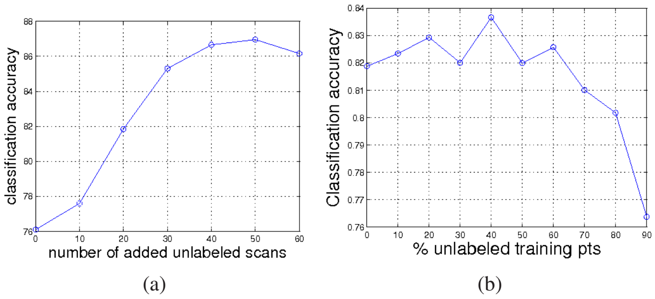


Fig. 7. Behavior of the semi-supervised learner in a car detection problem. Each point in the two plots corresponds to the average performance of 10 one-time-slice models learnt by cross validation. (a) The number of labeled scans is fixed to 30. As more unlabeled scans are added to the training set, labeled returns are spread evenly across the training set while their total number is maintained constant. This plot shows that the classification accuracy is increased by adding unlabeled samples. (b) The training sets contain 90 scans and the testing sets contain 10 scans. The x coordinate means that x% of randomly chosen returns in each of the 90 scans are unlabeled. This plot shows that classification accuracy is maintained with only 40% of the original labeled set.

D. Map Building

This section presents the classification performances obtained with the three models introduced in Section VI-D. For these three networks, the hidden state of each node ranges over the seven object types: car, trunk, foliage, people, wall, grass, and other (other referring to any other object type). Results for local classification are first presented in order to provide a baseline for comparison. All the evualtions were performed using 10-fold cross validation.

The characteristics of the training and testing data averaged over the 10-fold cross validation sets are provided in table III. The Sydney dataset was used for these experiments since it contains horizontal 2D laser scans which can be registered using ICP. The registration of downward looking scans is a more complex problem which explains why these mapping experiments were not reproduced using the Boston dataset.

	Length vehicle trajectory	# scans total labeled	# nodes total labeled
Training set	2.6 km	3843 72	67612 5168
Testing set	290 m	427 8	7511 574

TABLE III

CHARACTERISTICS OF THE TRAINING AND TESTING SETS

1) Local Classification: A seven-class logitboost classifier is learned and instantiated at each node of the network as the association potential A (Eq. 2). Local classification, *i.e.*, classification which does not take neighborhood information into account is performed and leads to the the confusion matrix presented in table IV. This confusion matrix displays a strong diagonal which corresponds to an accuracy of 90.4%. A compact characterization of the confusion matrix is given by precision and recall values. These are presented in table V. Averaged over the seven classes, the classifier achieves a precision of 89.0% and a recall of 98.1%.

Truth \ Inferred	Car	Trunk	Foliage	People	Wall	Grass	Other
Car	1967	1	7	10	3	0	48
Trunk	4	165	18	0	4	0	11
Foliage	25	18	1451	0	24	0	71
People	6	2	2	145	0	0	6
Wall	6	6	21	0	513	1	39
Grass	0	0	1	1	1	146	4
Other	54	5	123	3	24	0	811

TABLE IV

LOCAL CLASSIFICATION: CONFUSION MATRIX

In %	Car	Trunk	Foliage	People	Wall	Grass	Other
Precision	96.6	81.7	91.3	90.1	87.5	95.4	79.5
Recall	97.9	99.3	96.4	99.7	98.5	99.9	95.4

TABLE V

LOCAL CLASSIFICATION: PRECISION AND RECALL

2) Delaunay CRF Classification:

a) CRF without built-in link selection: the accuracy achieved by this first type of network is 90.3% providing no improvements on local classification. As developed in Section IV-C.2, the modeling of the spatial correlation is too coarse since it contains only one type of link which cannot accurately model the relationships between all neighbor nodes. As a consequence, the links end up representing the predominant relationship in the data. In the dataset, the predominant neighborhood relationships are of the type "neighbor nodes have the same label". The resulting learned links enforce this "same-to-same" relationship across the network leading to over-smoothed estimates and explaining why this class of networks fails to improve on local classification. To verify that a better modeling of the CRF links improves the classification performance, we now presents results generated by Delaunay CRFs equipped with additional link selection nodes (as shown in Fig. 3).

b) CRF with built-in link selection: the accuracy achieved by this second type of network is 91.4% which corresponds to 1.0% improvement in accuracy. Since the local accuracy is already high, the improvement brought by the network may be better appreciated when expressed as a reduction of the error rate of 10.4%. This result validates the claim that a set of link types encoding a variety of node-to-node relationships is required to exploit the spatial correlations in the laser map.

3) Tree based CRF classification: The two types of networks evaluated in the previous section contain cycles and require the use of an approximate inference algorithm. The tree based CRFs presented in Section IV-C.3 avoid this issue and allow the use of an exact inference procedure (BP in its non loopy version).

This third type of network achieves an accuracy of 91.1% which is slightly below the accuracy given by a Delaunay CRF with link selection while still improving on local classification. However, the major improvement brought by this third type of network is in terms of computational time. Since the network has the complexity of a tree of depth one, learning and inference, in addition to being exact, can be implemented very efficiently. As displayed in table VI, a tree based CRF is 80% faster at training and 90% faster at testing than a Delaunay CRF. Since both network types use as their association potential the seven-class logitboost classifier, they use the same features which are extracted from a scan and its associated image in 1.2 secs on average. As shown in table III, the test set contains 7511 nodes on average which suggests that the tree based CRF approach is in its current state very close to real time, feature extraction being the main bottleneck.

	Feature Extraction (per scan)	Learning (training set)	Inference (test set)
Delaunay CRF (with link selection)	1.2 secs	6.7 mins	1.5 mins
Tree based CRF	1.2 secs	1.5 mins	10.0 secs

TABLE VI

COMPUTATION TIMES

4) Map of Objects: This section presents a visualization of some of the mapping results. It follows the lay out of Fig. 8 in which the vehicle was travelling from right to left.

At the location of the first inset, the vehicle was going up a straight road with a fence on its left and right, and, from the foreground to the background, another fence, a car, a parking meter and bush. All these objects were correctly classified with the fences and the parking meter identified as other.

In the second inset, the vehicle was coming into a curve facing a parking lot and bush on the side of the road. Four returns mis-classified as other can be seen in the background of the image. The class other regularly generated false positives which is possibly caused by the dominating number of training samples of this class. Various ways of re-weighting the training samples or balancing the training set were tried without significant improvements.

While reaching the third inset, a car driving in the opposite direction came into the field of view of our vehicle's sensors. The trace let by this car in the map appears in the magnified inset as a set of blue dots along side our vehicle's trajectory. Dynamic objects are not explicitly considered in this work. They are assumed to move at a speed which does not prevent ICP from performing accurate registration. In campus types of areas where this data was acquired, this assumption has proven to be valid. In spite of a few mis-classifications in the bush on the left side of the road, the pedestrians on the side walk as well as the wall of the building are correctly identified.

Entering the fourth inset, our vehicle was facing a second car, scene which appears in the map as a blue trace intersecting our vehicle's trajectory. Apart from one mis-classified return on one of the pedestrians, and one mis-classified return on the tree in the right of the image, the inferred labels are accurate. Note that the first right return is correctly classified illustrating the accuracy of the model at the border between objects.

E. Convergence Analysis of the Inference

As mentioned in Section III-B, convergence in graphs with cycles is not guaranteed but can be experimentally checked. In this section, the converge of loopy BP is explored. The Boston dataset was used for this last set of experiments. The behavior of loopy BP in a cyclic network was analyzed using a set 400 manually labeled scans and 5-fold cross validation.

The evaluation is summarized in Fig.9. Inference is performed in each of the networks involved in the cross validation with a varying number of loopy BP iterations. The accuracies provided correspond to the classification of the two classes car and other. The networks used for these tests are the ones described in section VI-B.2.

The left plot of Fig. 9 shows that on average loopy BP convergences after about 5 iterations where the accuracy reaches a plateau and is higher than the accuracy obtained with local classification. The right plot of Fig. 9 shows that, as expected, the inference time increases linearly with the number of loopy BP iterations. Knowing that loopy BP convergences in about 5 iterations permits maintaining the computation times as small as appropriate.

VII. Conclusions

A general probabilistic framework for multi-class multi-sensor object recognition was presented. This framework is based on CRFs which were used as a flexible modeling tool to automatically select the relevant features extracted from the various modalities and represent different types and spatial and temporal correlations.

Based on two datasets acquired with different sensors, eight different sets of results were presented. The benefits of modeling spatial and temporal correlations was first demonstrated on a car detection problem where an increase in accuracy of up to 5% was obtained. The experimental study of the proposed semi-supervised version of VEB suggested that the classifier accuracy can be maintained using only 40% of the original labeled set and can be increased by adding unlabeled samples to the training set. Using partial labeling, our approach can be applied to far larger and hence diverse sets of laser scans and images, which results in better generalization performance. Three different types of networks were introduced to build semantic maps and evaluated on a seven-class classification problem where an accuracy of 91% was achieved. The mapping experiments brought some insights on the smoothing role of CRF links and we showed how over-smoothing can be avoided by creating networks which automatically select the types of links to be used. Computation times were evaluated showing that the larger networks involved in our study are close to being real-time requiring about 11 seconds for inference on a set of 7500 nodes. Finally, an empirical study of the inference algorithm verified its convergence which was observed to be reached in about 5 iterations.

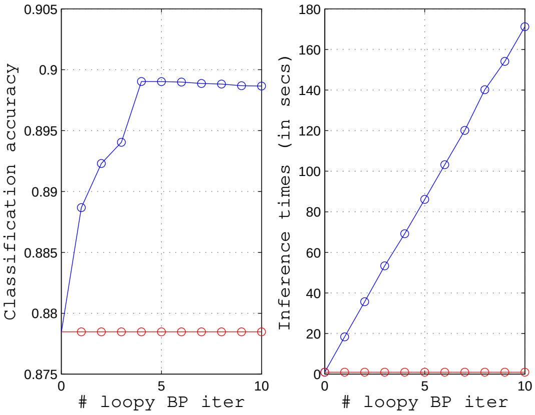


Fig. 9. Empirical analysis of the convergence of loopy BP. On the left, classification accuracies obtained on a car detection problem plotted as function of the number of loopy BP iterations. On the right, the corresponding computation times. The red plots refer to local classification.

These various experimental results have demonstrated that CRFs stand as a general solution to the problem of classification in robotics applications.

While the proposed framework was developed for 2D laser scans, the set of experiments on the Boston dataset (Section VI-B.2 and VI-E) present a first simple extension to 3D laser data and suggest that this CRF framework is not only applicable to 2D laser points.

Current investigations aim at developping CRF models able to deal in real-time with 3D lidar data such as the one available at [6]. 3D lidars can provide up to 1.5 million data points per second which makes a point-wise reasoning too computationally intensive. We are working on grid based approaches where the classifier is designed to estimate the labels of the grid cells rather than the labels of individual returns, using the grid as a way to compress the incoming flow of laser returns.

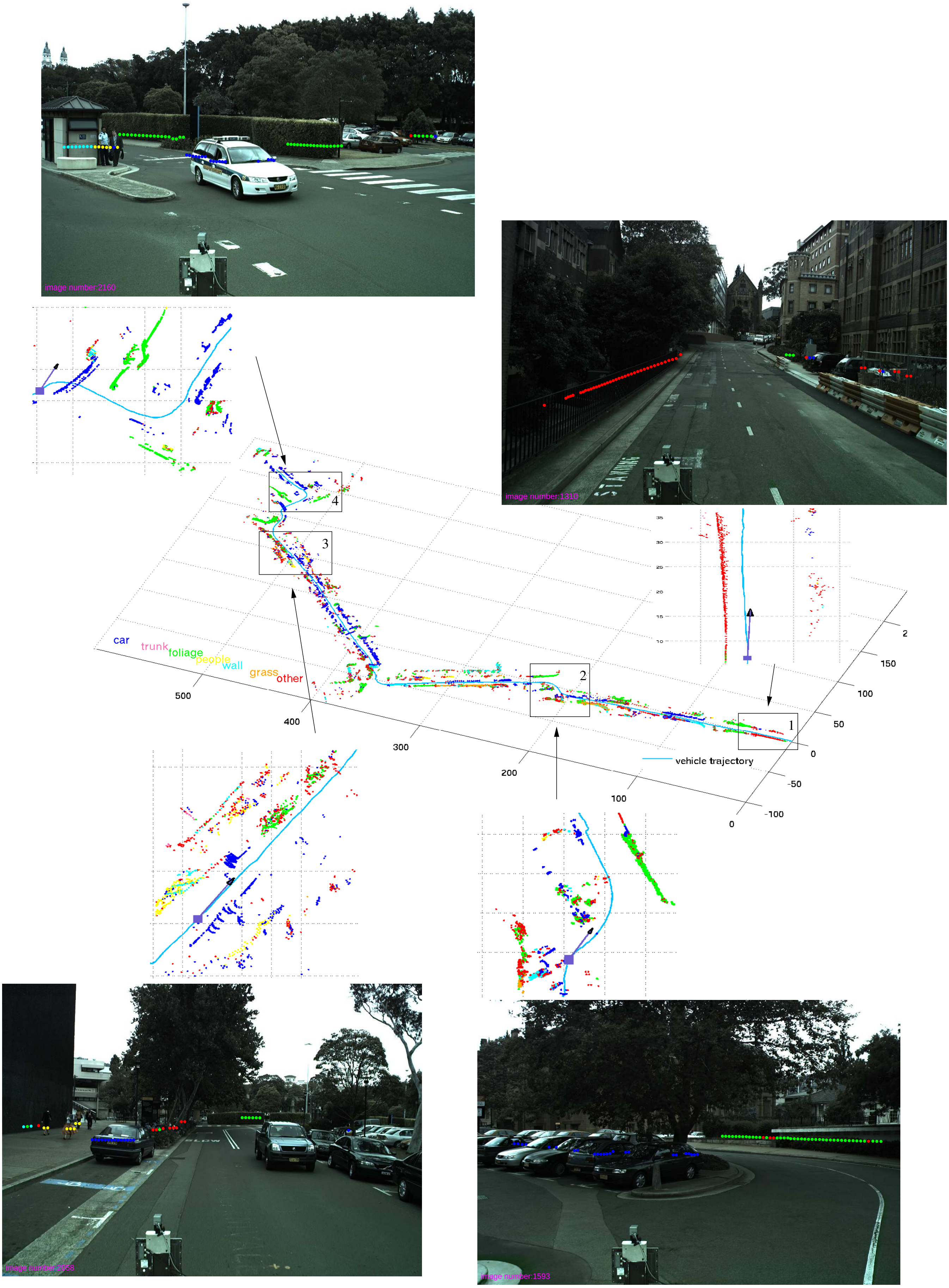


Fig. 8. Visualization of 750 meters long portion of the estimated map of objects with total length of 3km. The map was generated using the tree based CRF model. The legend is indicated in the bottom left part of the 2D plane. The color of the vehicle's trajectory is specified in the bottom right part of the same plane. The coordinate in the plane of the map are in meters. Each inset is magnified and associated to an image displayed with the inferred labels projected back onto the original returns. The location of the vehicle is shown in each magnified patch with a square and its orientation indicated by the arrow attached to it. The laser scanner mounted on the vehicle can be seen in the bottom part of each image.

VIII. ACKNOWLEDGMENTS

The authors would like to thank Albert Huang for sharing the Boston dataset as well as his software and expertise on this data, and Juan Nieto, Jose Guivant, and Oliver Frank for helping with the acquisition of the Sydney dataset. This work is supported by the ARC Center of Excellence programme, the Australian Research Council (ARC), the New South Wales (NSW) State Government, the University of Sydney Visiting Collaborative Research Fellowship Scheme, and DARPA's ASSIST and CALO Programmes (contract numbers: NBCH-C-05-0137, SRI subcontract 27-000968).

REFERENCES

[1] Finding long straight lines code. `http://www.cs.uiuc.edu/homes/dhoiem/`.

[2] D. Anguelov, D. Koller, E. Parker, and S. Thrun. Detecting and modeling doors with mobile robots. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 2004.

[3] D. Anguelov, B. Taskar, V. Chatalbashev, D. Koller, D Gupta, G. Heitz, and A. Ng. Discriminative learning of Markov random fields for segmentation of 3D scan data. In *Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[4] M. Bosse and R. Zlot. Keypoint design and evaluation for place recognition in 2d lidar maps. In *Proc. of RSS-08, Workshop: Inside Data Association*, 2008.

[5] DARPA Urban Challenge. `http://www.darpa.mil/grandchallenge/index.asp`.

[6] MIT DARPA Urban Challenge datasets. `http://grandchallenge.mit.edu/wiki/index.php/PublicData`.

[7] M. De Berg, M. Van Kreveld, M. Overmars, and O. Schwarzkopf. Springer-Verlag, 2000. 2nd rev. ISBN: 3-540-65620-0.

[8] B. Douillard, D. Fox, and F. Ramos. A spatio-temporal probabilistic model for multi-sensor multi-class object recognition. In *Proc. of the International Symposium of Robotics Research (ISRR)*, 2007.

[9] B. Douillard, D. Fox, and F. Ramos. Laser and vision based outdoor object mapping. In *Proc. of Robotics: Science and Systems*, 2008.

[10] L. Fei-Fei and P. Perona. A bayesian hierarchical model for learning natural scene categories. In *Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[11] R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7:179–188, 1936.

[12] Y. Freund and R.E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1), 1997.

[13] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: A statistical view of boosting. *The Annals of Statistics*, 28(2), 2000.

[14] S. Friedman, D. Fox, and H. Pasula. Voronoi random fields: Extracting the topological structure of indoor environments via place labeling. In *Proc. of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2007.

[15] H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24:417–441, 1933.

[16] V. Kumar, D. Rus, and S. Singh. Robot and sensor networks for first responders. *IEEE Pervasive Computing*, 3(4), 2004. Special Issue on Pervasive Computing for First Response.

[17] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. of the International Conference on Machine Learning (ICML)*, 2001.

[18] L. Liao, T. Choudhury, D. Fox, and H. Kautz. Training conditional random fields using virtual evidence boosting. In *Proc. of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2007.

[19] L. Liao, D. Fox, and H. Kautz. Extracting places and activities from GPS traces using hierarchical conditional random fields. *International Journal of Robotics Research (IJRR)*, 26(1), 2007.

[20] B. Limketkai, L. Liao, and D. Fox. Relational object maps for mobile robots. In *Proc. of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2005.

[21] D. Lowe. Discriminative image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2), 2004.

[22] M. Mahdaviani and T. Choudhury. Fast and scalable training of semi-supervised CRFs with application to activity recognition. In *Advances in Neural Information Processing Systems (NIPS)*, 2007.

[23] O. Martinez-Mozos, R. Triebel, P. Jensfelt, A. Rottmann, and W. Burgard. Supervised semantic labeling of places using information extracted from sensor data. *Robotics and Autonomous Systems*, 55(5), 2007.

[24] G. Monteiro, C. Premebida, P. Peixoto, and U. Nunes. Tracking and classification of dynamic obstacles using laser range finder and vision. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2006.

[25] K. Murphy, Y. Weiss, and M. Jordan. Loopy belief propagation for approximate inference: An empirical study. In *Proc. of the Conference on Uncertainty in Artificial Intelligence (UAI)*, 1999.

[26] P. Newman, D. Cole, and K. Ho. Outdoor SLAM using visual appearance and laser ranging. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, Orlando, USA, 2006.

[27] Pyramid Histogram of Oriented Gradients. `http://www.robots.ox.ac.uk/~vgg/research/caltech/phog.html`.

[28] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers, Inc., 1988.

[29] I. Posner, M. Cummins, and P. Newman. Fast probabilistic labeling of city maps. In *Proc. of Robotics: Science and Systems*, 2008.

[30] I. Posner, D. Schroeter, and P. M. Newman. Describing composite urban workspaces. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 2007.

[31] I. Posner, D. Schroeter, and P. M. Newman. Using scene similarity for place labeling. In *Proc. of the International Symposium on Experimental Robotics (ISER)*, 2007.

[32] F. Ramos, J. Nieto, and H.F. Durrant-Whyte. Recognising and modelling landmarks to close loops in outdoor slam. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 2007.

[33] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290:2323–2326, 2000.

[34] E. Simoncelli and W. Freeman. The steerable pyramid: A flexible architecture for multi-scale derivative computation. In *Proc. of the International Conference on Image Processing*, 1995.

[35] C. Sutton and A. McCallum. An introduction to conditional random fields for relational learning. In L. Getoor and B. Taskar, editors, *Introduction to Statistical Relational Learning*. MIT Press, 2006.

[36] J. Tenenbaum, V. DeSilva, and K. R. Muller. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290:2319–2323, 2000.

[37] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT Press, Cambridge, MA, September 2005. ISBN 0-262-20162-3.

[38] A. Torralba, K. Murphy, and W. Freeman. Contextual models for object detection using boosted random fields. In *Advances in Neural Information Processing Systems (NIPS)*, 2004.

[39] R. Triebel, K. Kersting, and W. Burgard. Robust 3D scan point classification using associative Markov networks. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 2006.

[40] P. Viola and M. Jones. Robust real-time ob ject detection. In *International Journal of Computer Vision*, volume 57, page 2, 2004.

[41] Q. Zhang and R. Pless. Extrinsic calibration of a camera and laser range finder (improves camera calibration). In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sendai, Japan, 2004.

[42] Z. Zhang. Iterative point matching for registration of free-form curves and surfaces. *International Journal of Computer Vision*, 13(2):119–152, 1994.