Voronoi Random Fields: Extracting Topological Structure via Place Classification

Technical Report UW-TR

Stephen Friedman
University of Washington
Seattle, WA

Hanna Pasula
University of Washington
Seattle, WA

Dieter Fox
University of Washington
Seattle, WA

Abstract

The ability to build maps of indoor environments is extremely important for autonomous mobile robots. In this paper we introduce Voronoi Random Fields (VRFs), a novel technique for mapping the topological structure of indoor environments. Our maps describe environments in terms of their spatial layout along with information about the different places and their connectivity. To build these maps, we extract a Voronoi graph from an occupancy grid map generated with a laser range-finder, and then represent each point on the Voronoi graph as a node of a conditional random field, which is a discriminatively trained graphical model. The resulting VRF estimates the label of each node, integrating features from both the map and the Voronoi topology. The labels provide a segmentation of an environment, with the different segments corresponding to rooms, hallways, or doorways. Experiments using different maps show that our technique is able to label unknown environments based on parameters learned from other environments.

1 Introduction

Building semantically meaningful representations of indoor environments is a fundamental goal of robot mapping. Over the last few years, the SLAM (simultaneous localization and mapping) community has made tremendous progress in the development of efficient and highly accurate map-building techniques. Most of these techniques focus on accurately capturing the metric layout and / or topological structure of an environment. While metric maps have the advantage of being well suited for robot navigation tasks, they are typically unstructured and contain no information about the different types of places or objects in an environment. Topological approaches, on the other hand, are more compact and expressive in that they describe locally distinctive places, but these distinctive places are often defined based on a robot's perceptual space and are thus not necessarily related to indoor areas such as rooms or hallways.

Other researchers, for instance [Beeson *et al.*, 2005], focused on detecting places such as rooms, hallways, and doorways. However, most of these techniques detect only very specific structures and rely on hand-tuned rules to do so. Recently, [Stachniss *et al.*, 2005] introduced a technique that uses AdaBoost to learn place classifiers based on various features extracted from laser range-scans and cameras. While they achieve good results in classifying individual points in an environment, their approach does not take the connectivity of these points into account, and so does not generate a topological, spatially consistent representation of an environment.

The goal of our research is to generate semantically meaningful, topological-metric descriptions of indoor environments. We aim to label every location in a metric map of an environment with the type of place it belongs to. These labels provide an intrinsic segmentation of the map, one that represents the topological structure of an environment. Such a structure will enable robots to better communicate with humans about places, goals, and trajectories.

To achieve this goal, we introduce Voronoi random fields (VRFs), a novel method for generating semantically meaningful descriptions of indoor environments. Our approach uses a state-of-the-art SLAM technique to generate a metric occupancy grid map of an environment (citation omitted). It then extracts the Voronoi graph of this map [Latombe, 1991], which provides a compact representation of the free space and the connectivity structure of the environment. For each point on the Voronoi graph, VRFs then estimate the type of place it belongs to. This is done by generating a conditional random field with hidden nodes corresponding to the points on the Voronoi graph. Conditional random fields are discriminatively trained graphical models that have been shown to outperform generative approaches in areas such as natural language processing [Lafferty *et al.*, 2001] and computer vision [Kumar and Hebert, 2003]. The place labels estimated by our Voronoi random field provide a segmentation of an environment, with the different segments corresponding to rooms, hallways, junctions, or doorways. Since a robot's place labels should be consistent with how humans perceive the structure of environments, we use human-labeled training data to learn the parameters of our VRFs.

The paper is organized as follows. After describing related work in Section 2, we introduce Voronoi random fields, our application of conditional random fields to labeling Voronoi graphs, and also show how to perform inference and learning in this model. Experiments are described in Section 4, followed by conclusions and a discussion of future work.

2 Related Work

Voronoi graphs are an extremely useful topological representation of indoor environments [Latombe, 1991]. [Thrun, 1998] constructs topological representations of environments by segmenting and pruning a Voronoi graph extracted from an occupancy grid map. This approach does not distinguish between different types of places and is rather brittle since it makes deterministic modeling decisions based on manually tuned, local features. In order to learn a topological map of an environment, [Kuipers and Beeson, 2002] generate distinctive views by unsupervised clustering of raw laser range-data and associate these views with distinctive states. [Tomatis *et al.*, 2003] represent places in a hybrid topological-metric map as corners and openings detected with a laser range-finder. Their focus is on place recognition for SLAM, and the topological nodes are not classified into types of places.

[Beeson *et al.*, 2005] detect topological places based on gateways and pathways in a reduced extended Voronoi graph, which they extract from occupancy grid maps. Their technique relies on various hand-tuned rules to detect places and the authors do not show how it generalizes to complex scenarios in unseen environments.

Our work focuses on *place classification*. We do not intend to detect individual, distinctive places, but rather aim at labeling *any* location in an environment. Furthermore, we do not try to solve the place recognition, or data association, problem, which aims at detecting when a robot has returned to a previously visited place [Tomatis *et al.*, 2003]. We build on the fact that this problem is solved sufficiently well by state-of-the-art SLAM techniques, which generate accurate laser range-maps of indoor environments [Thrun *et al.*, 2005].

In a similar vein as our research, [Stachniss *et al.*, 2005] use AdaBoost to learn place classifiers based on features extracted from laser range-scans and camera images. They show how to connect the classifier outputs to a hidden Markov model, which then performs HMM filtering on the trajectory followed by a robot. While this technique performs very well, it still has several limitations compared to our approach. The HMM only operates on the given trajectory, so the classification of places in an environment depends on the actual path followed by a robot. Furthermore, the approach is not able to make use of the connectivity of an environment when labeling places. In contrast to HMM filtering, we show how to perform collective labeling on a Voronoi graph of an environment. Our model thus takes both local features and connectivity information into account. Additionally, it is able to generate a topological, semantically labeled graph description of an environment.

3 Voronoi Random Fields

3.1 Conditional Random Fields

Conditional random fields (CRFs) are undirected graphical models that were developed for labeling sequence data [Lafferty *et al.*, 2001]. Unlike HMMs and Markov random fields, which assume that observations are independent given the hidden state, CRFs make no assumptions about the dependency structure between different parts of the data. CRFs are thus especially suitable for classification tasks that rely on *complex* and *overlapping* data and features.

The nodes in a CRF represent hidden states, denoted $\mathbf{y} = \langle \mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n \rangle$, and data, denoted $\mathbf{x}$. Note that while individual observations $\mathbf{x}_i$ are often connected to the corresponding hidden states $\mathbf{y}_i$, our data $\mathbf{x}$ consists of a complete map of an environment, and we connect all hidden states to this single map. The nodes $\mathbf{y}_i$, along with the connectivity structure represented by the undirected edges between them, define the conditional distribution $p(\mathbf{y}|\mathbf{x})$ over the hidden states $\mathbf{y}$. Let $\mathcal{C}$ be the set of cliques in the graph of a CRF. Then, a CRF factorizes the conditional distribution into a product of *clique potentials* $\phi_c(\mathbf{x}, \mathbf{y}_c)$, where every $c \in \mathcal{C}$ is a clique in the graph and $\mathbf{x}$ and $\mathbf{y}_c$ are the observed data and the hidden nodes in the clique c, respectively. Clique potentials are functions that map variable configurations to non-negative numbers. Intuitively, a potential captures the "compatibility" among the variables in the clique: the larger the potential value, the more likely the configuration. Using clique potentials, the conditional distribution over the hidden states is written as

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

where $Z(\mathbf{x}) = \sum_{\mathbf{y}} \prod_{c \in \mathcal{C}} \phi_c(\mathbf{x}, \mathbf{y}_c)$ is the normalizing partition function. The computation of this partition function is exponential in the size of $\mathbf{y}$ since it requires summation over all possible configurations of hidden states $\mathbf{y}$. Hence, exact inference is possible for a limited class of CRF models only.

Potentials $\phi_c(\mathbf{x}, \mathbf{y}_c)$ are described by log-linear combinations of *feature functions* $\mathbf{f}_c()$, *i.e.*,

$$\phi_c(\mathbf{x}, \mathbf{y}_c) = \exp\left(\mathbf{w}_c^T \cdot \mathbf{f}_c(\mathbf{x}, \mathbf{y}_c)\right), \tag{2}$$

where $\mathbf{w}_c^T$ is the transpose of a weight vector $\mathbf{w}_c$, and $\mathbf{f}_c(\mathbf{x}, \mathbf{y}_c)$ is a function that extracts a vector of features from the variable values. Using feature functions, we can rewrite the conditional distribution (1) as

$$p(\mathbf{y} \mid \mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp\left\{ \sum_{c \in \mathcal{C}} \mathbf{w}_c^T \cdot \mathbf{f}_c(\mathbf{x}, \mathbf{y}_c) \right\} \tag{3}$$

Now, before we describe how to perform efficient inference and learning in CRFs, we will now show how CRFs can be used for labeling places in an environment.

3.2 Voronoi Random Fields for Place Labeling

To label the different places in an environment, we estimate the place types at the points of a discrete *Voronoi graph* of the environment. Such a graph is defined via the points that are equidistant to the closest two or more obstacles in an environment [Latombe, 1991]. The left panel in Figure 1 shows an occupancy grid map generated by our laser mapping approach along with the corresponding Voronoi graph (after pruning). As can be seen, the Voronoi graph nicely represents the connectivity structure of the environment. Hallway intersections and their entries into rooms are typically represented by points that have three or more neighbors. Note that any point in the environment can be associated with a point on the Voronoi graph. Thus, by estimating place labels on the

Voronoi graph, we can determine the place type of any point in the environment.

We label the points on a Voronoi graph by converting the graph into a conditional random field, which we call a Voronoi random field (VRF). A VRF is constructed from a Voronoi graph by representing the points $\mathbf{v}_i$ on the Voronoi graph as the hidden nodes $\mathbf{y}_i$ of the VRF. The state of the hidden nodes ranges over the different place types: `room`, `doorway`, `hallway`, `junction`, and `other`. The neighboring points in the Voronoi graph are also connected in the VRF, thereby allowing the VRF to take the labels of neighboring points into account and thus probabilistically model connectivity constraints. The observed data $\mathbf{x}$ is extracted from the Voronoi graph and the local map area visible from each Voronoi point. We use two types of features extracted from the data:

Spatial features describe the layout of the local area around a Voronoi point. Most of of our spatial features are inspired by those used by [Stachniss *et al.*, 2005] for laser-based place detection. These features are extracted from the occupancy grid map; they include distance to the closest obstacle and shape information about the area visible from the Voronoi point.

Connectivity features make use of the connectivity information encoded by the Voronoi graph. These features include place types of the neighboring nodes, number of neighbors in the Voronoi graph, and shape information about the graph, such as curvature. Additionally, we use a feature that describes the size of the minimal loop going through a node in the Voronoi graph. This feature turns out to be extremely useful for tasks such as distinguishing doorways from narrow passages in rooms. While doorways are typically not part of a small loop, narrow passages inside rooms are often part of small loops caused by furniture such as tables (see Figure 1).

Our VRFs contain three types of cliques. First, each node in the VRF is a single node clique, the potential of which is a weighted function of the spatial and connectivity features extracted at that node. The second type of clique contains pairs of VRF nodes that are connected in the Voronoi graph. The potential of these cliques measures the spatial compatibility between place types. A third type of clique is generated for each node that has three neighbors. While such "junction" nodes often correspond to intersections in hallways, they also occur frequently in rooms. This final clique type contains four nodes and measures the compatibility between the labels of all nodes connected at such a junction in the Voronoi graph (more than three neighbors could be handled if necessary).

3.3 Inference

Inference in CRFs can estimate either the marginal distribution of each hidden variable $\mathbf{y}_i$ or the most likely configuration of all hidden variables $\mathbf{y}$ (*i.e.*, MAP estimation), as defined in (3). Both tasks can be solved using *belief propagation* (BP), which works by sending local messages through the graph structure of the model. Each node sends messages to its neighbors based on messages it receives and the clique potentials, which are defined via the observations and the neighborhood relation in the CRF. For instance, the "observation" potential of a node is computed by first extracting the features described in Section 3.2 at the map location corresponding to the node, and then feeding them through the log-linear function (2) (we actually use derived features, see Section 3.4).

BP generates provably correct results in graphs with no loops, such as trees or polytrees. However, since virtually all VRFs contain loops, we apply loopy belief propagation, an approximate inference algorithm that is not guaranteed to converge to the correct probability distribution [Murphy *et al.*, 1999]. In our experiments, we compute the MAP labeling of an environment using max-product loopy BP. Fortunately, even when the algorithm failed to converge, our experiments showed reasonable results.

3.4 Learning

CRF Parameter Learning

The goal of CRF parameter learning is to determine the weights of the feature functions used in the conditional likelihood (3). CRFs learn these weights discriminatively by maximizing the conditional likelihood of labeled training data. While there is no closed-form solution for optimizing (3), it can be shown that (3) is convex relative to the weights $\mathbf{w}$. Thus, the global optimum of (3) can be found using a numerical gradient algorithm. Unfortunately, this optimization runs an inference procedure at each iteration, which can be intractably inefficient when using loopy BP in large VRFs.

We therefore resort to maximizing the *pseudo-likelihood* of the training data, which is given by the sum of local likelihoods $p(\mathbf{y}_i \mid \mathrm{MB}(\mathbf{y}_i))$, where $\mathrm{MB}(\mathbf{y}_i)$ is the Markov blanket of variable $\mathbf{y}_i$: the set of the immediate neighbors of $\mathbf{y}_i$ in the CRF graph [Besag, 1975]. Optimization of this pseudo-likelihood is done by minimizing the negative of its log, resulting in the following objective function:

$$L(\mathbf{w}) \;=\; -\sum_{i=1}^{n} \log p(\mathbf{y}_i \mid \mathrm{MB}(\mathbf{y}_i), \mathbf{w}) + \frac{(\mathbf{w}-\widetilde{\mathbf{w}})^T(\mathbf{w}-\widetilde{\mathbf{w}})}{2\sigma^2} \quad (4)$$

Here, the terms in the summation correspond to the negative pseudo log-likelihood and the right term represents a Gaussian shrinkage prior with mean $\widetilde{\mathbf{w}}$ and variance σ^2. Without additional information, the prior mean is typically set to zero. In our approach, we use unconstrained L-BFGS [Liu and Nocedal, 1989] to optimize (4). The key advantage of maximizing pseudo-likelihood rather than the likelihood (3) is that the gradient of (4) can be computed extremely efficiently, without running an inference algorithm. Learning by maximizing pseudo-likelihood has been shown to perform very well in different domains; see [Richardson and Domingos, 2006] or [Kumar and Hebert, 2003] for an example.

AdaBoost for Feature Induction

While the features described in Section 3.2 could be used directly as continuous observations in CRFs, we found that this approach did not yield very good results. This is due to the fact that the log-linear representation underlying CRFs corresponds to a unimodal Gaussian likelihood for each continuous feature, which is not flexible enough to capture the complex relationships between place types and feature values.

We overcome this limitation by extracting suitable features from our continuous feature values using AdaBoost [Freund and Schapire, 1997], which has been applied successfully by [Stachniss *et al.*, 2005] in the context of place recognition. Specifically, we learn a binary AdaBoost classifier for each place type by using maps of labeled indoor environments to generate training data for every class c in the form of $D^c = \{\langle x_1, y_1\rangle, \langle x_2, y_2\rangle, \ldots, \langle x_n, y_n\rangle\}$, where each x_i contains the features extracted for a specific Voronoi point, and $y_i \in \{-1, +1\}$ specifies whether or not this point belongs to the class c. As weak classifiers we use binary decision stumps applied to the features. In this case, AdaBoost sequentially learns a weighted set of T decision stumps, each defined by a feature f_t^c and a threshold θ_t^c. Note that the same feature can be used multiple times with different thresholds. The weighted combination of these decision stumps gives the classification of whether or not a feature vector x belongs to place type c,

$$H^c(x) \quad = \quad \text{sign}\left(\sum_{t=1}^{T} \alpha_t^c \, h_t^c(x)\right) \tag{5}$$

Here $h_t^c(x) = \text{sign}(x_{f_t^c} - \theta_t^c)$ is the output of the t-th decision stump, and α_t^c is the learned weight associated with it. Essentially, AdaBoost automatically determines which of the continuous features (and thresholds applied to them) are most useful in supporting good classification results.

We investigated two ways of combining AdaBoost and CRFs. The first, which we call VRF_D, takes all the AdaBoost decision stumps $h_t^c(x)$ learned for all the classes c and uses them as binary features in a CRF. The CRF then learns the weights for those stumps and for the neighborhood potentials using pseudo-likelihood maximization as described above. The second approach, called VRF_M, uses the weighted sum, or margin [Niculescu-Mizil and Caruana, 2005], of each AdaBoost classifier as a continuous feature (the sum inside the sign in (5)). To learn the weights of these individual place classifiers, VRF_M uses pseudo-likelihood maximization, taking into consideration the neighborhood relations in the network.

For C place types, VRF_D uses CT binary features, T for each of the C place types, and VRF_M uses the C continuous features that are generated by weighted summation of the decision stumps.

3.5 Summary

Our approach learns the parameters of VRFs from labeled maps of training environments. To do so, it first learns a binary AdaBoost classifier for each place type from manually labeled maps. For each map, we generate the corresponding Voronoi random field using the features learned by AdaBoost. The parameters of these VRFs are then learned by maximizing the pseudo-likelihood (4) of the training data, using parameter sharing. The resulting parameters are the weights of the different types of cliques occurring in a VRF. After learning, a novel environment is labeled as follows:

1. *Generate occupancy grid map* from laser range-data collected by a mobile robot (this is done using our SLAM technique, citation omitted for anonymity).
2. *Extract Voronoi graph* from the map.
3. *Generate Voronoi random field* with nodes and edges corresponding to the Voronoi graph.
4. *Extract spatial and connectivity features* from the map, generate the learned AdaBoost decision stumps, and use them as observations in the VRF.
5. *Run max-product loopy BP inference* to estimate the MAP labeling of the nodes in the VRF.

4 Experimental Results

We evaluated our approach on maps of four indoor environments, acquired as laser range-data from the Radish robot data repository [Howard and Roy, 2003]. For each data set, we generated an occupancy grid map and extracted a Voronoi graph from that map. We then manually labeled each point on each graph as either `room`, `doorway`, `hallway`, or `junction`, where a `junction` can be an intersection between hallways or an entry area into a room. The training set contained approximately 1,250 Voronoi points. We extracted 53 different features at each point and used AdaBoost to generate 25 decision stumps for each of the four classes, resulting in a total of 100 binary features used in our VRFs. AdaBoost learning took about a minute; learning the parameters of a VRF took an additional minute on a standard desktop PC. Loopy-BP MAP inference generally took less than a minute per test map when it converged. However, we allowed for a generous time-out based on the graph diameter, such that those that did not converge would run for up to 5 minutes.

To assess the contributions of the different aspects of our VRF framework, we compared the following four approaches using leave one out cross-validation on the maps. The first one, denoted AB_S, uses AdaBoost along with the spatial features to classify each point on the Voronoi graphs. The second one, AB_{SC}, additionally uses connectivity features extracted from the graphs. These features include information such as number of neighbors in the graph, size of the smallest loop containing a node, and the distance to the closest obstacle. The third approach, VRF_D, is a VRF that uses the AdaBoost decision stumps as binary features. The final approach, VRF_M, uses the weighted sum, or margin, of each AdaBoost classifier as features. For learning, we set the prior mean of all weights $\widetilde{w}$ to zero (compare (4)).

	ABld.	Allen	Frbg.	Intel	**Avg.**
AB_S	87.0	90.8	82.0	81.1	85.2 ± 4.4
AB_{SC}	88.6	92.1	86.6	83.7	87.8 ± 3.5
VRF_D	93.6	93.3	91.5	86.4	91.2 ± 3.3
VRF_M	94.2	93.1	91.3	88.2	91.7 ± 2.6

Table 1: Accuracy of leave one out place labeling.

Table 1 summarizes the accuracy of the different techniques in terms of the percentages of correct labels for the four test environments. Not surprisingly, the accuracy of AdaBoost increases consistently when adding connectivity features (for an average increase from 85.2% to 87.8%). The lower two rows display the accuracy of using AdaBoost for feature induction in VRFs. Both VRF approaches perform significantly better than AdaBoost, though the difference between them is not significant. Note that the test maps are

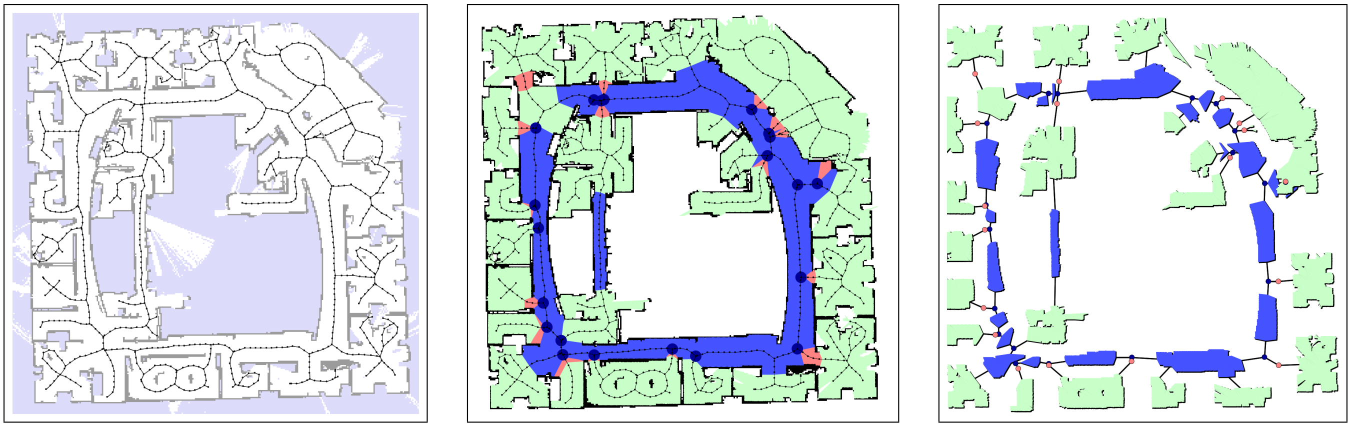

Figure 1: Intel test map: (left) Occupancy grid map built via SLAM along with Voronoi graph. (middle) The labeled Voronoi graph defines a place type for each point on the map. Hallways are colored blue, rooms green, doorways red, and junctions are indicated by dark blue circles. (right) Topological-metric map given by the segmentation of the labeled Voronoi graph. The spatial layout of rooms and hallway sections is provided along with a connectivity structure indicated by lines and points (doorways red, hallways blue).

very diverse, so the cross-validation demonstrates good generalization to new environments for all techniques.

While these numbers indicate an improvement of VRFs over AdaBoost, they do not adequately reflect the true performance gain achieved by our VRF technique. To get an additional assessment of the techniques, we considered the following scenario. A mobile robot is placed at a random location in the map and takes the shortest path to another randomly chosen location. Along its path, the robot records the sequence of rooms, hallway sections, junctions, and doorways it passes through. We used edit distance to compare sequences recorded on maps labeled using our algorithms to those recorded on maps labeled using ground truth. Edit distance determines the minimum number of operations (insertions, deletions) needed to make the inferred path match the ground truth path. Our resulting measure, called topological edit distance (TED), reports the ratio of edit distance to path length; a lower TED ratio means better performance. The final reported statistic is the average of 100 randomly selected paths in the map. The random paths taken on the corresponding maps in different tests were chosen consistently to allow a straightforward comparison. The TED scores of the aforementioned experiments are summarized in Table 4. The VRF_Dmethod offers clear improvements significant at the p<0.06 level over AB_{SC}. The VRF_Mmethod appears better, but the overall improvement is insignificant due to poor performance on the Allen map. One explanation for this is that VRF_Mfailed to correctly label several junctions between hallways, which many paths will pass through.

	ABld.	Allen	Frbg.	Intel	**Avg.**
AB_S	79.4	60.5	76.6	62.6	69.8 ± 9.4
AB_{SC}	35.7	30.9	74.2	59.8	50.1 ± 20.0
VRF_D	18.2	22.1	23.7	25.7	22.4 ± 3.1
VRF_M	14.3	50.6	21.0	22.2	27.0 ± 15.8

Table 2: Topological Edit Distance of leave-one-out place labeling.

Figure 1 shows the performance of our technique on one of our test maps (shown for VRF_D, $VRF_M$is very similar). The coloring of the middle map is given by labeling all points with the label of the nearest point in the VRF. The right panel shows the topological-metric map resulting from grouping contiguous room and hallway regions of the same label into topological nodes. As can be seen, the topological-metric map generated automatically by our VRF nicely represents the environment's connectivity structure (indicated by lines and doorway and hallway nodes) and the spatial layout of individual rooms and hallway sections. Figure 2 provides a visual comparison of $VRF_D$with AB_{SC}(AdaBoost using spatial and connectivity features). In agreement with the results given in Table 2, our approach generates significantly more consistent segmentations of the environments. For instance, AdaBoost generates a large number of false-positive doorways and hallways, especially in the cluttered rooms of the Freiburg map. Our labelings are also more consistent than those reported in [Stachniss *et al.*, 2005].

5 Conclusions

We presented Voronoi random fields, a novel approach to generating semantically meaningful topological-metric descriptions of indoor environments. VRFs apply discriminatively trained conditional random fields to label the points of Voronoi graphs extracted from occupancy grid maps. The hidden states of our VRFs range over different types of places, such as rooms, hallways, doorways, and junctions. By performing inference in the graph structure, our model is able to take the connectivity of environments into account. We use AdaBoost to learn useful binary features from the continuous features extracted from Voronoi graphs and occupancy maps. The parameters of our model are trained efficiently using pseudo-likelihood. Experiments show that our technique enables robots to label unseen environments based on parameters learned in other environments, and that the spatial reasoning supported by VRFs results in substantial improvements over a local AdaBoost technique.

We consider these results extremely encouraging; they provide mobile robots with the ability to reason about their environments in terms more similar to how humans perceive their environments. As a next step, we will add high-level contextual information such as the length of hallways and the shape

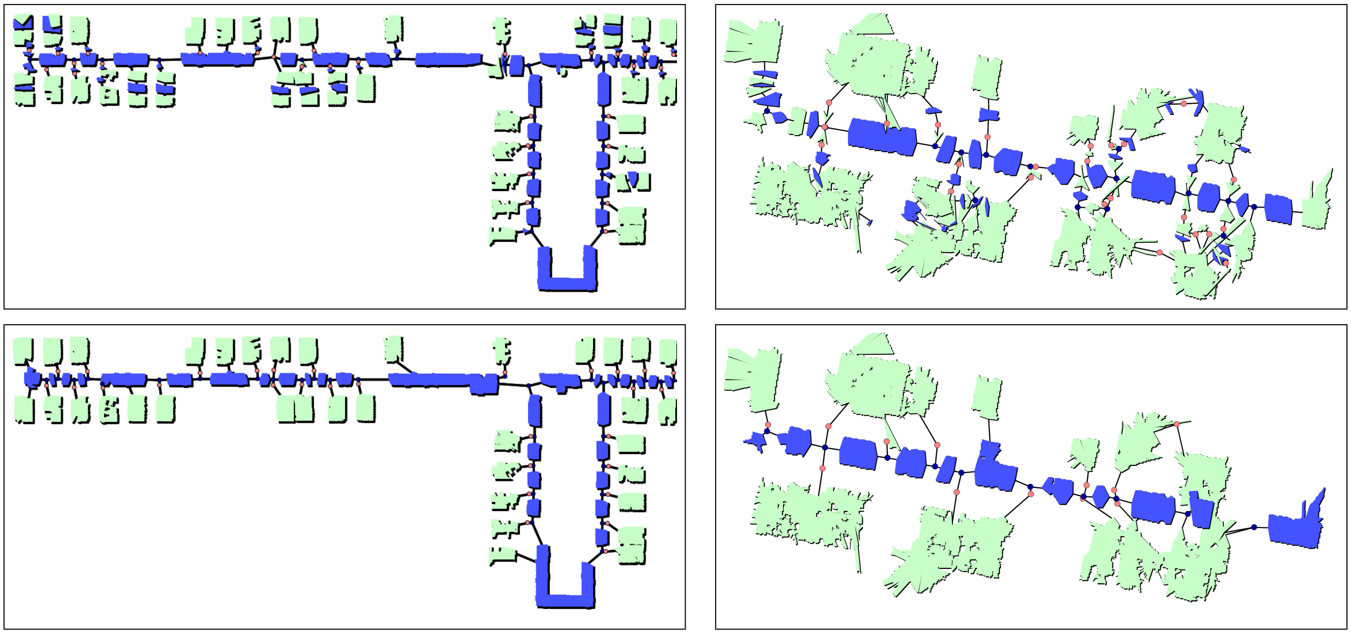

Figure 2: Abuilding (left) and Freiburg (right) test maps labeled by AdaBoost (upper row) and our VRF technique (lower row). The VRF technique is far more accurate in detecting junctions and its segmentations are spatially more consistent than those done by AdaBoost.

of rooms to our model. Each segmentation will then correspond to a two-level CRF, with the upper level representing the features of these places (similarly to [Liao *et al.*, 2006]). To avoid summing over all possible CRF structures, we will replace MAP estimation with a sampling or k-best technique.

An obvious shortcoming of our current approach is the restriction to laser range data. As in the work of [Stachniss *et al.*, 2005; Torralba *et al.*, 2004], we intend to add visual features in order to improve place recognition.

Finally, the longterm goal of this research is to generate maps that represent environments symbolically in terms of places and objects. To achieve this goal, we will combine our VRF technique with approaches to object detection, such as the one proposed by [Limketkai *et al.*, 2005]. In this application, the place labeling performed by the VRF will provide context information for the object labeling performed jointly at the lower levels of a hierarchical CRF model.

References

[Beeson *et al.*, 2005] P. Beeson, N.K. Jong, and B. Kuipers. Towards autonomous topological place detection using the extended Voronoi graph. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 2005.

[Besag, 1975] J. Besag. Statistical analysis of non-lattice data. *The Statistician*, 24, 1975.

[Freund and Schapire, 1997] 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.

[Howard and Roy, 2003] A. Howard and N. Roy. The robotics data set repository (radish), 2003. `radish.sourceforge.net`.

[Kuipers and Beeson, 2002] B. Kuipers and P. Beeson. Bootstrap learning for place recognition. In *Proc. of AAAI)*, 2002.

[Kumar and Hebert, 2003] S. Kumar and M. Hebert. Discriminative random fields: A discriminative framework for contextual interaction in classification. In *Proc. of (ICCV*, 2003.

[Lafferty *et al.*, 2001] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. of ICML*, 2001.

[Latombe, 1991] J.-C. Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, Boston, MA, 1991.

[Liao *et al.*, 2006] L. Liao, D. Fox, and H. Kautz. Hierarchical conditional random fields for GPS-based activity recognition. In *Proc. of the International Symp. of Robotics Research*, 2005.

[Limketkai *et al.*, 2005] B. Limketkai, L. Liao, and D. Fox. Relational object maps for mobile robots. In *Proc. of IJCAI*, 2005.

[Liu and Nocedal, 1989] D. C. Liu and J. Nocedal. On the limited memory BFGS method for large scale optimization. *Math. Programming*, 45(3, (Ser. B)):503–528, 1989.

[Murphy *et al.*, 1999] K. Murphy, Y. Weiss, and M. Jordan. Loopy belief propagation for approximate inference: An empirical study. In *Proc. of UAI*, 1999.

[Niculescu-Mizil and Caruana, 2005] A. Niculescu-Mizil and R. Caruana. Predicting good probabilities with supervised learning. In *Proc. of ICML*, 2005.

[Richardson and Domingos, 2006] M. Richardson and P. Domingos. Markov logic networks. *Machine Learning*, 62, 2006.

[Stachniss *et al.*, 2005] C. Stachniss, O. Martinez-Mozos, A. Rottmann, and W. Burgard. Semantic labeling of places. In *Proc. of the International Symp. of Robotics Research*, 2005.

[Thrun *et al.*, 2005] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT Press, Cambridge, MA, September 2005.

[Thrun, 1998] S. Thrun. Learning metric-topological maps for indoor mobile robot navigation. *Artificial Intelligence*, 99(1), 1998.

[Tomatis *et al.*, 2003] N. Tomatis, I. Nourbakhsh, and R. Siegwart. Hybrid simultaneous localization and map building: a natural integration of topological and metric. *Robotics and Autonomous Systems*, 44(1), 2003.

[Torralba *et al.*, 2004] 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.