Semantic Parsing for Priming Object Detection in RGB-D Scenes

César Cadena and Jana Košecka

Abstract— **The advancements in robot autonomy and capabilities for carrying out more complex tasks in unstructured indoors environments can be greatly enhanced by endowing existing environment models with semantic information. In this paper we describe an approach for semantic parsing of indoors environments into semantic categories of *Ground, Structure, Furniture* and *Props*. Instead of striving to categorize all object classes and instances encountered in the environment, this choice of semantic labels separates clearly objects and non-object categories. We use RGB-D images of indoors environments and formulate the problem of semantic segmentation in the Conditional Random Fields Framework. The appearance and depth information enables us induce the graph structure of the random field, which can be effectively approximated by a tree, and to design robust geometric features, which are informative for separation and characterization of different categories. These two choices notably improve the efficiency and performance of the semantic parsing tasks. We carry out the experiments on a NYU V2 dataset and achieve superior or comparable performance and the fraction of computational cost.**

I. Introduction

The problem of semantic understanding of environments from images or 3D cloud points involves the problem of simultaneous segmentation and categorization of image regions or 3D point clouds. With the advent of RGB-D sensing and availability of video, several approaches have been proposed which exploit the simultaneous availability of 3D structure and appearance information. The existing methods differ in the choice of semantic categories they are trying to infer, the choice features, elementary primitives for labeling and associated inference algorithms. Majority of the existing methods for semantic parsing formulate the final inference in the Markov Random Field (MRF) or Conditional Random Field (CRF) Framework, where the chosen elementary primitives are the random variables and the neighbourhood relationships determine the dependencies between them. While in several instances the final inference problem is formulated directly on 3D points or image pixels [1], large number of pixels and their dependencies make the final inference problem computationally expensive. Popular choice for many approaches is to over-segment the image into superpixels [2], followed by the computation of features over these regions. The top performing methods of [3], [4] often use high-quality segmentation techniques and high dimensional features to instantiate the final learning and inference problem. In the proposed approach we revisit these choices and propose a novel efficient over segmentation of the RGB-D data along with simple and efficient method to compute features for semantic parsing of RGB-D scenes.

a) Proposed Approach: We formulate the semantic labeling problem in the CRF framework, where the dependencies between random variables are represented by a graph, induced by both image superpixels and 3D scene structure. The distinguishing features of our approach are: a) the use of a tree graph structure in the CRF setting which effectively approximates the dependencies and enables exact and efficient inference amenable for real-time implementation and b) the use of simple and efficient appearance and geometric cues, providing evidence about depth discontinuities. We carry out the semantic parsing experiments on a NYU V2 dataset [3], which contains 464 diverse indoor scenes and 1449 annotated frames, achieving superior or comparable performance at the fraction of computational cost.

In the next section, we provide an overview of the related work. In Section III we describe the details of our approach. Section IV describes the experiments on NYU V2 dataset and compares our approach with the state of the art methods. Finally, in Section V we present discussions and conclusions of the presented work and discuss possible future directions.

II. Related Work

Several approaches developed in the context of robotics applications focused on problem of semantic parsing of urban scenes acquired by a moving vehicle and relied mostly on 3D measurements from laser range finder or dense depth reconstruction. In these methods the graph structure was typically induced by partitioning of 3D point clouds. Authors in [5] consider 2D semantic mapping of street scenes using laser and image data providing computationally intensive solution on a graph induced by Delaunay triangulation; [6] use both laser and image measurements and provide efficient solution considering only two object classes, foliage and vehicles.

Computer vision approaches obtain the 3D information using either stereo or 3D reconstruction and use more elaborate appearance features. The state of the art of the semantic parsing approaches in outdoors settings achieve relatively high average accuracy of 85-90% on scenes datasets [1], [7]. In these settings categories such as buildings, roads, sky often exhibit lower intra class variability, have strong location priors and ample of training data available.

César Cadena and Jana Košecka are with the Computer Science Department, Volgenau School of Engineering at George Mason University, Fairfax, VA 20030, US. ccadenal@gmu.edu, kosecka@.cs.gmu.edu.

This research has been funded by the US Army Research Office Grant W911NF-1110476.

In indoors environments several methods have been developed exploiting the RGB-D data. In [8] authors highlighted the need for efficiency of the final inference and used up to 17 object classes. They were able to exploit stronger appearance and contextual cues due to the scale and different nature of the environment. More recently several researchers carried out more comprehensive experiments on larger NYU RGB-D dataset introduced in [2]. Authors in [4] focus of local patch based kernel features and achieved very good average performance while considering 13 structural and furniture classes and grouping all the smaller objects in 'other' category. The proposed features are computed over high quality superpixels obtained with computationally expensive boundary detector. In addition to inference of semantic labels in the work of [3] the authors simultaneously considered the problem of inference of the support relations between different semantic categories. The approach relied on elaborate pre-processing stage involving hierarchical segmentation stage, reasoning about occlusion boundaries and piece-wise planar segmentation. All these stages required a solution to a separate inference problem, using additional features and stage specific energy functions. In the final inference problem the feature vectors computed over superpixels were over 1000 dimensions. In the work of [9] authors bypass the complex feature computation stage and and use convolutional networks in combination with over-segmentation for the indoors scenes labeling task.

III. Our Approach

A. Method

We formulate the labeling process in the framework of Conditional Random Fields (CRFs) with a tree graph structure encoding the pairwise relationships. From a RGB-D image, our approach starts by over-segmentation on the RGB images using the efficient *simple linear iterative clustering* (SLIC) algorithm [10]. The 3D centroid of each superpixel is used to compute the minimum spanning tree, defining the edges for the graphical model. The data and pairwise terms are determined using simple yet discriminative appearance and geometric cues for the classes that we are interested in. The learning and the final inference process is carried out over the graph in the CRFs context. In the remainder of this section we detail the components of our approach and explain the intuition behind them.

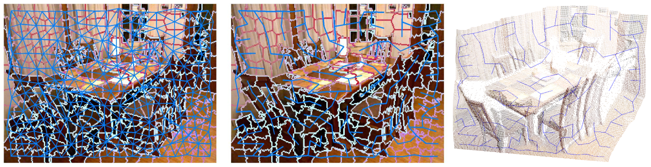
B. Framework: Conditional Random Fields

Instead of relying on Bayes' rule to estimate the distribution over hidden states $\mathbf{x}$ from observations $\mathbf{z}$, CRFs directly model $p(\mathbf{x}|\mathbf{z})$, the *conditional* distribution over the hidden variables given observations. Due to this structure, CRFs can handle arbitrary dependencies between the observations. This makes them substantially more flexible when using complex attributes without the independence assumption. These in our case are different observations extracted from the overlapping regions capturing both local and global context.

The nodes in a CRF are denoted $\mathbf{x} = \langle \mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n \rangle$, and the observations are denoted $\mathbf{z}$. In our framework the hidden states correspond to the m possible classes: $\mathbf{x}_i = \{Ground, Structure, Furniture, Props\}$, as defined by [3]. CRF factorizes the conditional distribution into a product of *potentials*. We consider only the potentials for the nodes $\phi(\mathbf{x}_i, \mathbf{z})$ (data-term) and edges $\psi(\mathbf{x}_i, \mathbf{x}_j, \mathbf{z})$ (pairwise-term). This choice is commonly referred to as pairwise CRFs. The potentials are functions that map variable configurations to non-negative numbers capturing the agreement among the involved variables: the larger a potential value, the more likely the configuration. Using the data and pairwise potentials, the conditional distribution over hidden states is written as:

$$p(\mathbf{x}|\mathbf{z}) = \frac{1}{Z(\mathbf{z})} \prod_{i \in \mathcal{N}} \phi(\mathbf{x}_i, \mathbf{z},) \prod_{i,j \in \mathcal{E}} \psi(\mathbf{x}_i, \mathbf{x}_j, \mathbf{z}) \qquad (1)$$

where $Z(\mathbf{z})$ is the normalizing partition function, and $\langle \mathcal{N}, \mathcal{E} \rangle$ are the set of nodes and edges of the graph.



(a) Dense graph on Image. (b) MST over 3D.

Fig. 1. Graph Structures (blue lines). On the left the most common graph structure used in the computer vision community. On the right the graph structure selected by us, a minimum spanning tree over 3D.

The potentials are described by log-linear combinations of *feature functions*, $\mathbf{f}$ and $\mathbf{g}$, i.e., the conditional distribution in Eq. 1 can be rewritten as:

$$p(\mathbf{x}|\mathbf{z}) = \frac{1}{Z(\mathbf{z})} \exp \left(\mathbf{w}_1 \sum_{i \in \mathcal{N}} \mathbf{f}(\mathbf{x}_i, \mathbf{z}) + \mathbf{w}_2 \sum_{i,j \in \mathcal{E}} \mathbf{g}(\mathbf{x}_i, \mathbf{x}_j, \mathbf{z}) \right) \qquad (2)$$

where $[\mathbf{w}_1, \mathbf{w}_2]$ are weights representing the importance of each term. CRFs learn these weights discriminatively by maximizing the conditional likelihood of labeled training data.

With this formulation we can obtain either the marginal distribution over the class variables $\mathbf{x}_i$ by solving Eq. 2, or the most likely assignment of all the hidden variables $\mathbf{x}$. The latter can be formulated as the *maximum a posteriori* (MAP) problem, seeking the assignment of $\mathbf{x}$ for which $p(\mathbf{x}|\mathbf{z})$ is maximal.

C. Minimum Spanning Tree over 3D distances

Instead of using the pixels as nodes of the graph, we over segment the image into regular superpixels using the efficient SLIC algorithm [10], grouping the pixels with a similar appearance. Typical choice when using superpixels is to model the dependencies between neighbouring superpixels in the image. This often yields dense graph structure, see Fig. 1(a), which is prone to over connect unrelated classes. In our case we define the graph structure for the CRF as a minimum spanning tree over the Euclidean distances between 3D superpixel's centroids in a scene. By definition, the minimum spanning tree connects points that are close in the measurement space, highlighting intrinsic locality in the

Default	Observation	Dim.	Comments
Image Features			
	LAB color	3	
	y_l	1	Vertical pixel location
	H_s	1	Entropy of $hg_s(y)$
3D Features			
	(h_s, d_s)	2	Height and Depth
0	$(\mu_{\Delta d_s}, \sigma_{\Delta d_s})$	2	if $d_s < \frac{1}{\|N\|}\sum_{j\in N}(d_j)$ $\Delta d_s = \|d_s - d_{j\in N}\|$
0	$1 - mean(\|\vec{n}_s\vec{n}_N\|)$	1	Neighbouring Planarity
0	$dist\_to\_plane$	1	Superpixel Planarity
0	$\|\vec{n}_s\vec{j}\|$	1	Superpixel Orientation

TABLE I

LOCAL OBSERVATIONS

scene, see Fig. 1(b). This graph structure was already used in the context of object recognition [11] and place recognition [12]. Given that our graph structure is a tree we can use the *belief propagation* (sum-product) algorithm to exactly infer the marginal distribution of each node, and the max-product algorithm to find the MAP assignment [13].

D. Feature description

With the graph structure defined for our CRF model, we have to define feature functions $\mathbf{f}(\mathbf{x}, \mathbf{z})$ and $\mathbf{g}(\mathbf{x}, \mathbf{z})$ in Eq. 2. The features for the data-term are computed as:

$$\mathbf{f}(\mathbf{x}_s, \mathbf{z}) = -\log P_s(\mathbf{x}_s|\mathbf{z}) \tag{3}$$

where the local prior $P_s(\mathbf{x}_s|\mathbf{z})$ is the output of a k-nearest neighbours (k-NN) classifier from a set of observations $\mathbf{z}$. We compute $P_s(\mathbf{x}_s|\mathbf{z})$ as proposed by [14] in Eq. 4, k is fixed to 10.

$$P_s(\mathbf{x}_s = l_j|\mathbf{z}) = \frac{1}{\sum_{j=1}^{m}\left(\frac{f(l_j)}{\overline{f}(l_j)}\frac{\overline{F}(l_j)}{F(l_j)}\right)}\frac{f(l_j)}{\overline{f}(l_j)}\frac{\overline{F}(l_j)}{F(l_j)} \tag{4}$$

where $f(l_j)$ (resp. $\overline{f}(l_j)$) is the number of neighbours to s with label l_j (resp. not l_j) in the kd-tree. And $F(l_j)$ (resp. $\overline{F}(l_j)$) is the counting of all the observations in the training data with label l_j (resp. not l_j). The observations $\mathbf{z}$ computed for every superpixel s capturing the appearance cues obtained from RGB image (*Image Features*) and the depth cues (*3D Features*) are summarized in Table I and described next.

1) Image Features:

- The mean of each channel in the LAB-color space for the superpixel.
- The vertical pixel coordinate for the superpixel's centroid.
- The entropy of the probability distribution for the superpixel boundaries belonging to the dominant vanishing points in the scene, see Fig. 2 rightmost.

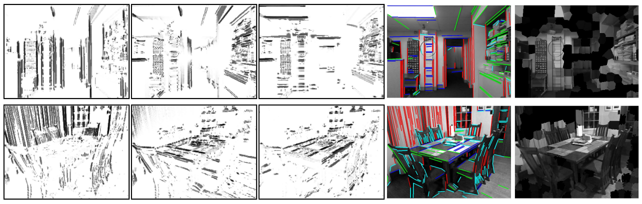


Fig. 2. Gradient mixture model [15]. From left: three gradient probability images of being aligned to each of three vanishing points, a color-coded membership image with lines to one of three vanishing points (red, green, blue), not to be consistent with any (cyan). Rightmost, the image with bright proportional to the entropy of each superpixel.

The entropy feature expresses geometric consistency of a superpixel boundary with a particular vanishing direction. This feature is motivated by the observation that in indoors environments superpixel boundaries are often aligned with the vanishing directions. This observation has been also widely utilized in single-view reconstruction techniques, e.g. [16]. We employ 5-component gradient mixture model for the Manhattan world described in [15]. For each image pixel, the model provides the probability of the pixel lying on an edge, probability to pointing to each of the three vanishing points, see Fig. 2, and the probability of being noise. We take into account only those pixels having the probability of being on an edge higher than being noise. For each of those points a maximum over last 4 probabilities is chosen as a membership of the point to either being consistent with one of the 3 vanishing points or not to be consistent with any, see fourth column in Fig. 2. For a particular superpixel s, we compute a normalized histogram $hg_s(y)$ with four bins $y = \{1, 2, 3, 4\}$ from memberships of all pixels lying along the superpixel boundary. In order to differentiate between clutter or small objects and structural classes we use the entropy of this normalized histogram, Eq. 5.

$$H_s = -\sum_{j=1}^{4} hg_s(y = j)\log\left(hg_s(y = j)\right) \tag{5}$$

2) 3D Features: For the 3D point cloud computed with the depth information we use cues from the 3D position and planarity, for the superpixel itself and for the superpixel with respect to its neighbourhood. The cues are:

- The depth (d_s) and height (h_s) for the superpixel's centroid.
- The mean and standard deviation of the absolute difference between the depth d_s and the neighbourhood's depths: $\|d_s - d_{j\in N}\|$. These are only computed if $d_s < \frac{1}{\|N\|}\sum_{j\in N}(d_j)$, with this condition we encode the *in-front-of* property.
- The superpixel planarity computed as the mean of the distance of all 3D points to a fitted plane by RANSAC [17].
- The neighbourhood planarity computed as one minus the mean of the dot product between the normal to the plane against to the neighbourhood normals [17].
- The superpixel orientation, taken as the projection of the superpixel's normal on the horizontal plane [7].

The superpixel neighbourhood N refers to all the superpixels in contact with superpixel s in the image. In Table I we also show the default values and the dimensionality of these observations. As a result we compute for each superpixel 12 dimensional feature vector with 5 elements from *Image features* and 7 from *3D features*.

Pairwise term: The pairwise feature $\mathbf{g}(\mathbf{x}_i, \mathbf{x}_j, \mathbf{z})$ is computed for every edge in the graph as:

	Ground	Furniture	Props	Structure	Average	Global
Ours	87.9	64.1	31.0	77.8	**65.2**	**66.9**
only Image Features	63.2	47.5	24.5	73.6	52.2	56.1
only 3D Features	**89.5**	**70.0**	16.9	79.4	62.7	65.8
data-term (k-NN, Eq. 3)	87.3	60.6	33.7	74.8	64.1	64.9
Silberman *et al.* [18]	68	**70**	**42**	59	59.6	58.6
Multiscale+depth convnet [19]	87.3	45.3	35.5	**86.1**	63.5	64.5

TABLE II

PIXEL-WISE PERCENTAGE RECALL ACCURACY.

Recall	Predictions			
	Ground	Furniture	Props	Structure
Ground	**87.9**	9.4	1.5	1.2
Furniture	5.9	**64.1**	12.9	17.1
Props	8.5	**32.6**	31.0	27.9
Structure	1.0	14.1	7.1	**77.8**

TABLE III

CONFUSION MATRIX FOR THE PIXEL-WISE ACCURACY IN %.

$$\mathbf{g}(\mathbf{x}_i, \mathbf{x}_j, \mathbf{z}) = \begin{cases} 1 - \exp\left(-\|c_i - c_j\|_2\right) & \rightarrow \quad l_i = l_j \\ \exp\left(-\|c_i - c_j\|_2\right) & \rightarrow \quad l_i \neq l_j \end{cases} \qquad (6)$$

where $\|c_i - c_j\|_2$ is the L2-Norm of the difference between the mean colors of two superpixels in the LAB-color space and l is the class label. Given the graph structure and the features outlined above, we proceed with the description of the learning and inference stage and the performance evaluation of the final semantic segmentation approach.

IV. EXPERIMENTS

In our experiments we use the NYU V2 RGB-D dataset [18], which contains 1449 labeled frames. The labeling spans over 894 different classes produced using Amazon Mechanical Turk. The authors of the dataset also provide a train and test splits and a mapping from the 894 categories to 4 classes: *Ground*, *Structure*, *Furniture* and *Props*, as was used in [18]. We take 795 training frames for building the kd-tree and for CRF parameter learning, and the remaining 654 frames for testing and quantitative comparison against state of art methods in RGB-D semantic segmentation.[1]

We obtain superpixel segmentation using SLIC implementation from the VLFeat library of [20], followed by the computation of the features described in Table I. With the computed features in the training set we build a kd-tree using the implementation of [21] with the default parameters. Then for every superpixel in the training set we obtain the k-NN classification for the training data using Eq. 4 with the $k = 10$ nearest neighbours.

The minimum weight spanning tree (MST) is computed from 3D centroids of all the superpixels. Now, using the MST graph, the output of the k-NN classifier in Eq. 3 and the pairwise potentials, Eq. 6, we learn the parameters in the CRF setting. For the learning, inference and decoding with CRFs we use the Matlab code for undirected graphical models (UGM).[2]

At the testing time, to obtain the most likely label assignment for the superpixels we solve the MAP problem over the CRFs. This problem does not require any threshold selection and all the parameters are computed/learned from the data. The inference results give us the labeling assignments over superpixels, we transfer those to every pixel in the superpixel to compute the pixel-wise accuracy of semantic labeling. In Fig. 3 we show several examples of the output of our approach.

Table III shows the confusion matrix normalized by rows (recall on the diagonal). The *Props* class is the most frequently confused. We can observe in the results, Fig. 3, several planar objects (posters, carpets, placemats) are labeled as *Structure* or *Furniture*. Another source of confusion is due to the quality of the ground truth, where we found several ambiguities in the *Structure* and *Furniture* categories, where kitchen tables, stoves, dishwashers and cabinets were labeled as the two interchangeably.

In Table II we show the pixel-wise recall accuracy along with the average and global accuracy for our approach: CRF-MST and k-NN+SLIC with Image and 3D features. To study the importance of image features vs 3D features, we remove one set at a time keeping the rest of the system intact. The rows only *Image Features* and only *3D Features* show the corresponding performance when using only one type of features. Using only 3D features we can see better results for three out four classes at the cost of very poor accuracy in Props and smaller average and global accuracy. Our full system, with both sets of features obtained the best trade off between all performance measures.

The row *data-term* in Table II shows the result of the k-NN classification using the image and 3D information. It is clear that the MST and the CRF framework improve the general performance.

We also compare against the full method proposed by [19]. That method uses the four channels (RGB and depth) in a multi-scale convolutional network to learn the features, and a 2-layer multi-perceptron as classifier, followed by the aggregation of the final assignments over superpixels computed by [22]. We can observe that our approach is competitive or better for all the classes, with better average and global accuracy. Their system takes 2 days to train but is very efficient in the testing stage, spending 0.7 seconds per frame to perform the segmentation using a parallel implementation for the convolution stage [23].

We also compare against the original work of authors who released this dataset [18]. As shown in Table II, their system still obtains the best accuracy in the *Furniture* and *Props* classes. They use a feature vector of 1128 elements to compute the data term with a logistic regression clas-

[1] The input images are cropped, the external blank boundary is removed, and then rescaled to the half yielding images of 313x234 in resolution.

[2] Code made available by Mark Schmidt at `http://www.di.ens.fr/~mschmidt/Software/UGM.html`

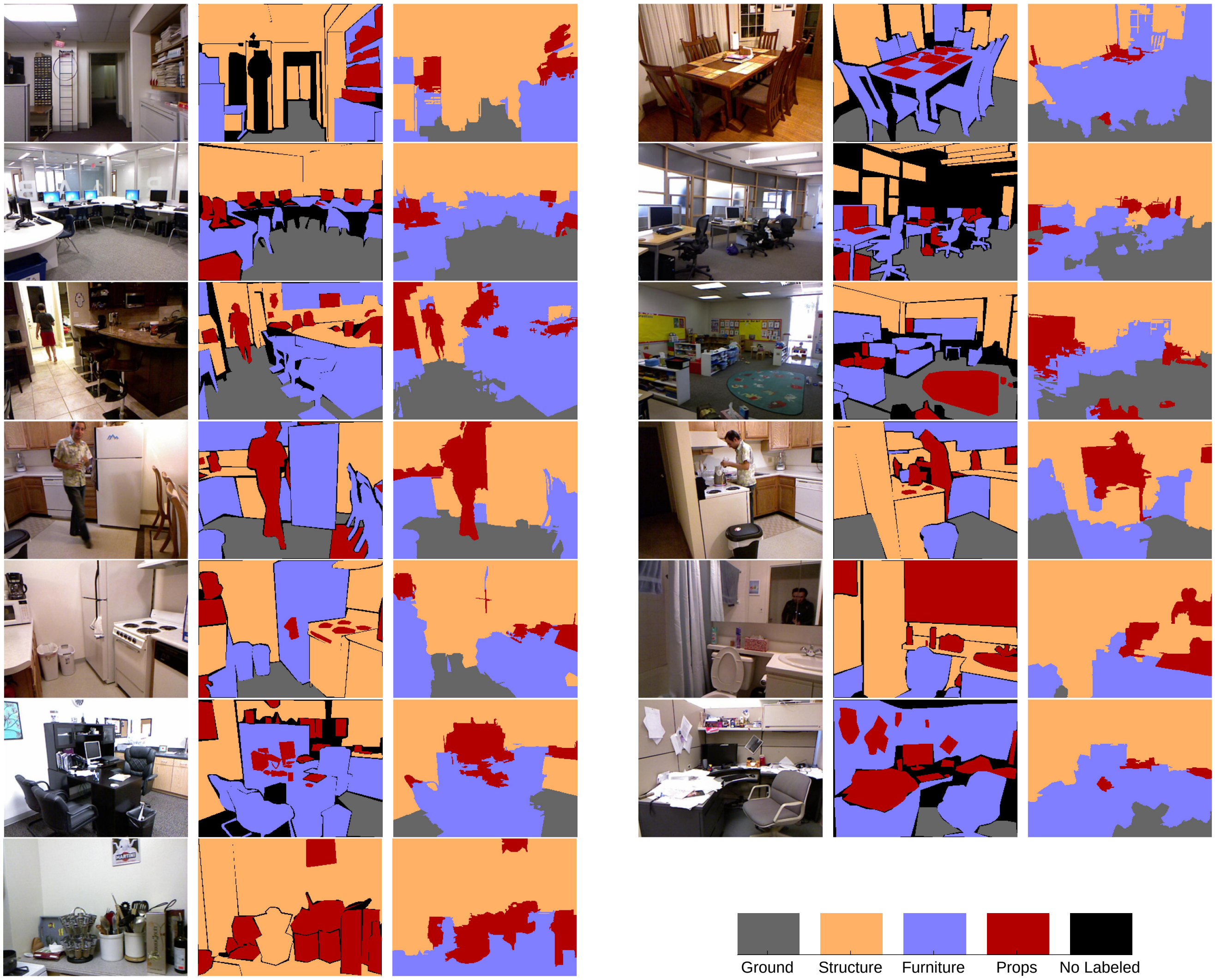


Fig. 3. Original images, ground truth labeling and MAP result from our approach.

sifier. The cues come from color (36), shape (1086) and scene (6) information (supplementary material in [18]). Our representation and the features are notably simpler enabling more efficient feature computation and inference while still achieving comparable performance.

A. Timing

The experiments were carried out with a research implementation of our approach in Matlab. The computational cost is detailed in Fig. 4, excluding the superpixel oversegmentation. The system runs in average at 1 fps in a single-thread of a 3.4 GHz IntelCore i7-2600 CPU M350 and 7.8GB of RAM. Including the SLIC over-segmentation is still able to run at 0.5fps. For the whole system, the mean and the maximum computational times are 1.02s and 1.48s, respectively. The average cost to obtain the SLIC superpixels is 692ms, although a C++ implementation would take half of that time as reported by [10]. In the feature computation the main bottleneck is the computation of the gradient mixture model [15] for vanishing directions. The training stage takes, less than 2 seconds to build the kd-tree, and 180 seconds for the CRF parameter learning. In total including the feature computation, the computational cost for all the training data is less than 20 minutes.

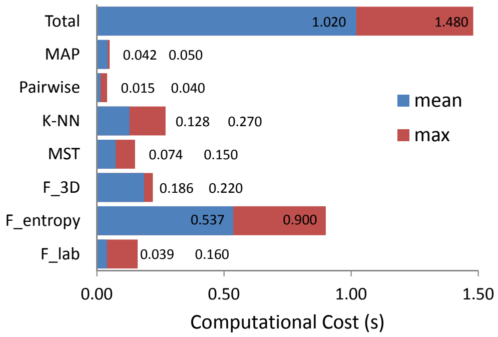


Fig. 4. Detailed mean and maximum computational timing.

V. Discussion

We have shown a basic implementation with real time capabilities that effectively uses appearance and 3D cues to generate evidence about the structure of the scene, while achieving better average and global accuracy of semantic labeling compared to the state of the art. Note that the accu-

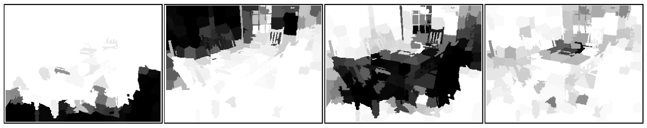


Fig. 5. Exact marginals obtained with the sum-product algorithm. From left: probability to be *Ground*, *Structure*, *Furniture* and *Props*.

racy reported by [19] was computed after training the system for 849 different classes and then, the results were clustered into the four categories, indicating that their approach and features are less robust to the high intra class variability in the four class problem.

We have shown that our graph structure induced by the MST over 3D does not sacrifice the labeling accuracy, and keeps the intra-class components coherently connected. Furthermore, by this selection we gain an exact and efficient inference. The computational complexity for the inference is $\mathcal{O}(nm^2)$, where n is the number of nodes in the graph, and m the number of classes. In that sense our approach is suitable for segmentation problems with small number of classes. While the computational cost could violate a real-time constraint for problems with large number of classes, we believe that a reliable semantic segmentation system should follow a coarse to fine strategy, where the labels for specific objects should be sought if necessary, and not classify large number of different objects classes at once.

An interesting discussion is related to the choice of basic representation and the features. Authors in [18] use over 1000 dimensional feature vectors, concatenating many engineered features, most of them developed previously for diverse but related tasks. On the other hand, authors in [23] propose to avoid the feature engineering and learn the features from the data using convolutional networks. They are still using a 768 dimensional feature vector for RGB channels, and the feature vectors have over 1000 dimensions when including the depth channel [19]. We can see that in both approaches, feature engineering vs feature learning, high dimensional feature space is constructed in which the different classes are more easily separable. In this work we propose and demonstrate, that a fewer (12 dimensional) but meaningful features are sufficient to obtain better semantic segmentation in RGB-D scenes. In addition to the choice of features there are at least two more reasons for this improvement. First, selecting a well-performing local classifier to handle high intra class variability: k-NN in our case vs logistic regression [18], vs 2-layer multi-perceptron [19]. Second, defining a more natural connections between neighbours given the type of data: minimum weighted spanning tree over 3D distances in our system, vs connecting with all the neighbours in the image [18], vs no connections at all between superpixels [19], [23] show a similar comparison.

By solving the marginals, see Fig. 5, instead of the MAP in our system (same computational complexity), the outcome could be used to find specific entities (object recognition) in a second stage or to infer the support relations as proposed in [18].

In our future work we will exploit the sequential information from video RGB-D in our approach. The presented model can be further extended in a hierarchical manner to incorporate additional information about specific objects of interest if those become available. We will explore obtaining the observations in a multi-scale way to improve the performance, this is inspired on the boosting achieved by [23] when compare the multi-scale vs one-scale convolutional networks.

References

[1] L. Ladický, P. Sturgess, K. Alahari, C. Russell, and P. Torr, "What, where and how many? combining object detectors and crfs," in *Computer Vision - ECCV 2010*. Springer Berlin Heidelberg, 2010.

[2] N. Silberman and R. Fergus, "Indoor scene segmentation using a structured light sensor," in *ICCV*, 2011.

[3] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from rgbd images," in *ECCV*, 2012.

[4] X. Ren, L. Bo, and D. Fox, "Rgb-(d) scene labeling: Features and algorithms," in *CVPR*, 2012.

[5] B. Douillard, D. Fox, F. Ramos, and H. Durrant-Whyte, "Classification and semantic mapping of urban environments," *Int. J. Rob. Res.*, vol. 30, pp. 5–32, January 2011.

[6] I. Posner, M. Cummins, and P. Newman, "A generative framework for fast urban labeling using spatial and temporal context," *Autonomous Robots*, vol. 27, no. 6, pp. 647–665, 2008.

[7] J. Xiao and L. Quan, "Multiple view semantic segmentation for street view images," in *Computer Vision, 2009 IEEE 12th International Conference on*, oct. 2009, pp. 686 –693.

[8] H. Koppula, A. Anand, T. Joachims, and A. Saxena, "Semantic labeling of 3d point clouds for indoor scenes," in *In 25th annual conference on neural information processing systems*, 2011.

[9] C. Couprie, C. Farabet, L. Najman, and Y. LeCun, "Proceedings of the international conference on learning representations," in *ArXiv preprint, January 2013,*, 2013.

[10] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 11, pp. 2274 –2282, nov. 2012.

[11] A. Quattoni, S. Wang, L.-P. Morency, M. Collins, and T. Darrell, "Hidden conditional random fields," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 29, no. 10, pp. 1848–1852, Oct. 2007.

[12] C. Cadena, D. Gálvez-López, J. Tardós, and J. Neira, "Robust place recognition with stereo sequences," *IEEE Transaction on RObotics*, vol. 28, no. 4, pp. 871 –885, 2012.

[13] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.

[14] J. Tighe and S. Lazebnik, "Superparsing: Scalable nonparametric image parsing with superpixels," in *Computer Vision - ECCV 2010*. Springer Berlin Heidelberg, 2010.

[15] J. Coughlan and A. Yuille, "Manhattan world: Orientation and outlier detection by bayesian inference," *Neural Computation*, vol. 15, no. 5, pp. 1063–1088, 2003.

[16] D. Hoiem, A. Efros, and M. Hebert, "Recovering surface layout from an image," *International Journal of Computer Vision*, vol. 75, pp. 151–172, 2007.

[17] C. Zhang, L. Wang, and R. Yang, "Semantic segmentation of urban scenes using dense depth maps," in *Computer Vision - ECCV 2010*. Springer Berlin Heidelberg, 2010.

[18] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from rgbd images," in *ECCV*, 2012.

[19] C. Couprie, C. Farabet, L. Najman, and Y. LeCun, "Indoor semantic segmentation using depth information," *CoRR*, vol. abs/1301.3572, 2013.

[20] A. Vedaldi and B. Fulkerson, "VLFeat: An open and portable library of computer vision algorithms," http://www.vlfeat.org/, 2008.

[21] A. M. Buchanan and A. W. Fitzgibbon, "Interactive feature tracking using K-D trees and dynamic programming," in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, 2006, pp. 626–633.

[22] P. Felzenszwalb and D. Huttenlocher, "Efficient graph-based image segmentation," *International Journal of Computer Vision*, vol. 59, pp. 167–181, 2004.

[23] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, "Learning hierarchical features for scene labeling," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. PP, no. 99, pp. 1–1, 2013.