Department of Computer Science
George Mason University Technical Reports

4400 University Drive MS#4A5
Fairfax, VA 22030-4444 USA
http://cs.gmu.edu/ 703-993-1530

Semantic Segmentation of Urban Environments into Object and Background Categories

Cesar Cadena and Jana Kosecka
George Mason University
Fairfax, VA
ccadenal,kosecka@gmu.edu

Technical Report GMU-CS-TR-2013-6

Abstract

Advancements in robotic navigation, object search and exploration rest to a large extent on robust, efficient and more advanced semantic understanding of the surrounding environment. Since the choice of most relevant semantic information depends on the task, it is desirable to develop approaches which can be adopted for different tasks at hand and which separate the aspects related to surroundings from object entities. In the proposed work we present an efficient approach for detecting generic objects in urban environments from videos acquired by a moving vehicle by means of semantic segmentation. Compared to traditional approaches for semantic labeling, we strive to detect variety of objects, while avoiding the need for large amounts of training data required for recognizing individual object categories and visual variability within and across the categories. In the proposed approach we exploit the features providing evidence about widely available non-object categories (such as sky, road, buildings) and use informative features which are indicative of the presence of object boundaries to gather the evidence about objects. We formulate the object/non-object semantic segmentation problem in the Conditional Random Field Framework, where the structure of the graph is induced by the minimum spanning tree computed over 3D reconstruction, yielding an efficient algorithm for an exact inference. We carry out extensive experiments on videos of urban environments acquired by a moving vehicle and compare our approach to existing alternatives.

1 Introduction

In recent years the research trends in robotic mapping, navigation and localization focused on developing methods for better understanding of the surrounding environment in order to facilitate more reliable life long navigation as well as endowing the models with additional semantic information. Towards this end numerous semantic mapping approaches have been proposed. They vary in the number and types of semantic classes they consider, sensing modality, features and inference algorithms. The final semantic labels are associated either with regions of an image or partition of 3D point clouds.

While the state of the art of the semantic parsing approaches in outdoors settings achieve relatively high average accuracy of 85-90% on some datasets [20], it is largely due to the fact that majority of 2D or 3D regions belong to non-object semantic categories often referred to as background or "stuff". These categories such as buildings, roads, sky often exhibit lower intra class variability, have strong location priors and ample of training data available. With more detailed scrutiny, the existing approaches consider either very small number of object categories (e.g. cars, trees), exhibit poorer performance on existing object categories (such as cars, pedestrians, bicyclist, traffic signs) [13] or used specialized object detectors to increase the performance [10]. Difficulty of detection of generic objects in street scene is partly due to the large number of possible categories exhibiting larger intra class variability. While in computer vision the semantic labeling is often the final goal, in the robotics setting the choice of the relevant semantic information depends on the task and skills of the robot. Hence in order to develop reusable and scalable systems, it is desirable to develop approaches which can be adopted for different tasks at hand and which separate the aspects related to non-objects and object entities, instead of committing to a fixed set of object labels and associated detectors.

In the proposed work instead of constraining the number of possible object classes, we consider objects as single generic class and propose to segment them regardless of their category. This is a very challenging problem given the high intra-class and inter-class variability between objects. To address this issue we propose to exploit 3D information and depth ordering cues as the evidence about presence and absence of generic objects. The additional goal of our approach is to propose a computational framework which is efficient, works in an on-line setting and can be easily extended to handle additional semantic information. The motivation for our choices is aligned with the need to design increasingly complex robotic systems which can operate over long-periods of time and gather additional information about the environment.

Contribution The main contribution of the proposed work is the development of novel representation, features and associated efficient inference algorithm for the problem of semantic labeling of outdoors urban environments into object and non-object categories. Similarly to the existing approaches we formulate the semantic labeling problem in Conditional Random Field (CRF) framework, where the dependencies between random variables are represented by a graph, induced by different partitions of an image or a 3D map. The distinguishing features of our approach are: a) the use of a tree graph structure in the CRF setting which is induced by the 3D reconstruction; b) the use of simple and efficient features and geometric cues, providing evidence about discontinuities and depth ordering; c) an explicit model of temporal coherency enabling on-line inference; d) an exact and efficient inference amenable for real-time implementation; e) a flexible model structure easily adoptable to a single or multi-frame settings, without a need for extra training.

The semantic output of our method produces detections and associated confidences about the presence of isolated generic objects and semantic labels of non-object categories. The output can be used effectively for priming *specific* object detectors and as a starting point of additional reasoning about various attributes (e.g. static/dynamic, movable, undergoing seasonal change etc).

In the next section, we provide an overview of the related work. In Section 3 we describe the details of our approach. Section 4 describes the experiments on street scene sequences and compares our approach with the state of the art methods. Finally, in Section 5 and 6 we present discussion and conclusions of the presented work and discuss possible future directions.

2 Related Work

The presented work on semantic segmentation of images and 3D point clouds into object and non-object categories is motivated by several previous approaches developed both in Computer Vision and Robotics communities. The approaches developed in the context of robotics applications rely mostly on 3D measurements from laser range finder or dense depth reconstruction and have been also explored in the context of analysis of urban scenes acquired by a moving vehicle. In these methods the graph structure is typically induced by a partitioning of 3D point clouds. Douillard et al. [6] consider 2D semantic mapping over street laser/image data providing computationally intensive solution on a graph induced by Delaunay triangulation. Posner et al. [14] also consider urban scenes and use both laser and image measurements and provide efficient solution considering only two object classes, foliage and vehicles. Dense stereo reconstruction was used on CamVid urban sequences by Zhang et al. [21] further improving the performance, but considering seven specific object classes. In indoors settings several methods have been developed exploiting the RGB-D data. Koppula et al. [9] highlighted the need for efficiency of the final inference and used up to 17 object classes, but were able to exploit stronger appearance and contextual cues due to the scale and different nature of the environment.

In computer vision community the problem of simultaneous segmentation and categorization of image regions was typically considered in a single view setting. The existing approaches differ in the number of semantic labels considered, the datasets used for evaluation, underlying representations and features used to formulate the final inference problem. Tighe and Lazebnik [17], and Eigen and Fergus [7] treated the representations of both object and no-object categories in the same manner and used both the SIFT Flow dataset with 33 semantic labels and Label Me with 253 labels to evaluate the performance of their approaches. The typical average global performance on all classes is about 80%, with 90% of pixels belonging to commonly occurring background categories, such as road, sky, building. The performance on object categories is notably lower and depends on the number of training examples in the dataset. The improvement in the performance on object categories has been shown by Ladický et al. [10], who used video and additional features obtained from confidence maps of specific object detectors. The experiments were performed on urban datasets, with 6 considered object categories.

In addition to single view settings and the use of image appearance cues, several approaches used either 3D information computed from multiple frames using Structure-from-Motion techniques [13, 20] or explicitly modeled temporal relationships between the frames in the inference problem [15, 20]. These strategies further improved the labeling performance [20], while still considering a small number of object categories, with objects being trees, cars, persons and recycle bins. In our work, instead of pursuing the object detection in a fully supervised setting, we would like to exploit both appearance and geometric properties of objects and how they appear in the environment. Alexe et al. [2] propose an approach for generic object detection motivated by a notion of saliency; objects are salient regions surrounded by background and delimited from it by strong contour edges. This approach only exploits the appearance cues, is applicable to a single view setting and more suitable in the context of image based retrieval applications, where images of scenes are well composed, containing little clutter. Closer to our work is the approach of Ayvaci and Soatto [3], which explicitly reasons about evidence of occlusions boundaries extracted from optical flow and relative depth ordering cues.

3 Our proposal

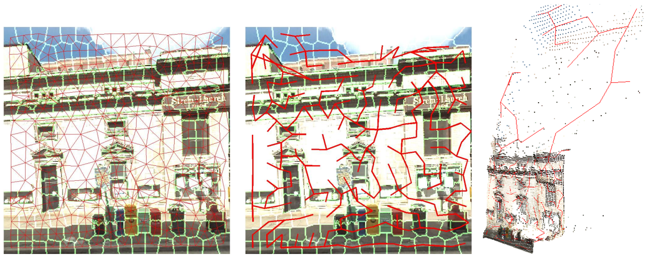
We consider a special case of semantic labeling of urban environments with a single generic object class and few non-object categories. We will consider object to be: "*... a (compact or simply-connected) subset of the domain of an image that back-projects onto a layout of surfaces that is partially surrounded by the medium*" as defined by Ayvaci and Soatto [3]. We start from an image and its associated 3D point cloud and formulate the labeling in the framework of Conditional Random Fields with a tree graph structure encoding the pairwise relationships. This selection is based on desirable properties of CRFs, as explained below, and the coherency and efficiency is provided by the tree graph structure.

3.1 Framework: Conditional Random Fields

Conditional random fields are probabilistic undirected graphical models first developed by Lafferty et al. [11] for labelling sequence data. CRFs are a case of Markov Random Fields, and thus satisfy the Markov properties, where there is no need to model the distribution over the observations [8].

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 flexible when using complex and overlapped attributes. These in our case are different observations extracted from the overlapped regions.

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, the generic object class and the non-object classes: building, ground and sky, i.e. $\mathbf{x}_i = \{object, building, ground, sky\}$.



(a) Dense graph on Image.

(b) MST over 3D.

Figure 1: Graph Structures. 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.

A CRF factorizes the conditional distribution into a product of *potentials*. We consider only the potentials for nodes $\phi(\mathbf{x},\mathbf{z})$ (data-term) and edges $\psi(\mathbf{x},\mathbf{z})$ (pairwise-term). This choice is commonly referred 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 on the graph. The computation of this function can be exponential in the size of $\mathbf{x}$. Hence, exact inference is possible for a limited class of CRF models only, e.g. in tree-structured graphs.

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:

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

where $\mathbf{w}$ is a weight vector, which represents the importance of each term for correctly identifying the hidden states. Weights are learned from labelled training data.

With this formulation we can obtain either the marginal distribution over the class of each variable $\mathbf{x}_i$ by solving Eq. 2, or the most likely classification 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.

3.2 Minimum Spanning Tree over 3D distances

Instead of computing the graph at the pixel level, we over segment the image into superpixels which are more

Default	Observation	Dim.	Comments
	LABcolor	3	
	RGB	3	
	Vert. px loc.	1	
	$\lVert d_i \rVert$	1	Depth
	h_i	1	Height
0	$mean\lVert d_i - d_{j\in N}\rVert$	1	if $d_i < \frac{1}{\lVert N\rVert}\sum_{j\in N}(d_j)$
0	$std\lVert d_i - d_{j\in N}\rVert$	1	if $d_i < \frac{1}{\lVert N\rVert}\sum_{j\in N}(d_j)$
-1	$mean(RepErr)$	1	
-1	$std(RepErr)$	1	
0	$1 - mean(\lVert\vec{n}_i\vec{n}_N\rVert)$	1	Neighbouring Planarity
0	$dist\_to\_plane$	1	Superpixel Planarity
0	$\lVert\vec{n}_i\vec{j}\rVert$	1	Superpixel Orientation

Table 1: Local observations

suitable to capture geometric properties of a region. An usual choice when using superpixels is a dense graph structure, see Fig. 1(a), which connects unrelated classes, e.g. the top superpixel belonging to a newspaper box has at least 3 edges to the building for the dense graph. 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 localities in the scene, see Fig. 1(b). Given that our graph structure is a tree we use the *belief propagation* algorithm [8] to infer the probability class of each node.

3.3 Method

Our approach starts by taking an image and its associated 3D reconstruction. Followed by a superpixel oversegmentation. Each one of the superpixels with at least three 3D points will be a node in the graphical model. The centroid of the 3D point cloud inside of the superpixel is used to compute the minimum spanning tree, defining the edges for the graphical model. The next step is to compute the data and the pairwise features.

3.4 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. We compute the feature for the data-term as:

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

where the local prior $P_i(\mathbf{x}_i|\mathbf{z})$ is the output of a Logitboost classification from a set of observations $\mathbf{z}$. The weak classifiers used in the boosting are weighted regression trees [18]. The observations $\mathbf{z}$ are computed from every superpixel i as following:

- The mean of the Lab-color space.
- The mean of the RGB-color space.
- The vertical pixel coordinate for the superpixel centroid.
- The depth (d_i) and height (h_i) for the superpixel's centroid.[1]
- The mean and standard deviation of the reprojection errors for the 3D point cloud.[2]
- The mean and standard deviation of the absolute difference between the depth d_i and the neighbourhood's depths: $\|d_i - d_{j\in N}\|$. These are only computed if $d_i < \frac{1}{\|N\|}\sum_{j\in N}(d_j)$, 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.
- The neighbourhood planarity computed as one minus the mean of the dot product between the normal to the plane against to the neighbourhood normals.[3]
- The superpixel orientation, taken as the projection of the superpixel's normal on the horizontal plane.[4]

The superpixel neighbourhood N refers to all the superpixels in contact with superpixel i in the image. In Table 1 we also show the default values and the dimensionality of these observations.

The pairwise feature is computed for every edge in the graph as:

$$\mathbf{g}(\mathbf{x}_i, \mathbf{x}_j, \mathbf{z}) = \begin{cases} 1 - \exp\left(-\|c_i - c_j\|_2\right) & \rightarrow \quad l_i = l_j \\ \exp\left(-\|c_i - c_j\|_2\right) & \rightarrow \quad l_i \neq l_j \end{cases} \tag{4}$$

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.

3.5 Multi-view Capability

To enable inference across multiple frames we use the relative transformation between the involved frames obtained using visual odometry. We transform all the 3D point clouds to a common reference and compute the minimum spanning tree over them. The data term features are computed in every single frame in the same way than before. Inference is run on each frame, while the graph connections are induced by the aligned 3D structure.

[1] Similar to the *distance to the vehicle trajectory and height* used in [4].

[2] Similar to the *backprojection residual* used by Brostow et al. [4].

[3] These two features were taken from the proposal of Zhang et al. [21].

[4] Similar to Xiao and Quan [20].

4 Experiments

For our experiments we use a set of images from the Google's Street View. We have available a total of 320 manually labelled non-sequential images as ground truth. We take 224 (70%) of them for Logitboost and parameter learning, and the remaining 96 images for testing and quantitative comparison against the state of art methods in single view semantic segmentation. Another qualitative evaluation over a video sequence is performed on a different part of the dataset, without a ground truth.

We consider the non-object semantic classes (building, ground and sky) and a generic object class, resulting in four class problem. The input images are cropped and rescaled to 320x320 in resolution. Originally, the manual labeling contained the classes: building, vehicle, ground, tree, sky and void. In order to use the labeling for our learning stage we have merged the classes vehicle, tree and void under the same object class. Although the original void class contains a large number of pixels at the boundaries and illumination artifacts, we decide to include it into the object class because also it contains people, poles, trash bins and more important objects.

In this experiment we need to obtain the 3D reconstruction. We first compute the visual odometry using three consecutive omnidirectional images and computing the optical flow [5] we recover a 3D point cloud. The pose of the vehicle is obtained using RANSAC-based epipolar geometry estimation formulated on their 3D rays, i.e. $\mathbf{p'}^T\mathbf{E}\mathbf{p} = 0$, yielding the essential matrix $\mathbf{E}$ [12]. Scale of the translation is estimated by a linear closed form 1-point algorithm on corresponding 3D points triangulated by direct linear algorithm from the previous image pair and the actual one. The estimate in this way offers poses accurate enough even without bundle adjustment unless the baseline is too small as also observed by Tardif et al. [16]. We do not use any additional optimization technique to improve the 3D reconstruction. It contains inaccuracies and errors common to standard methods.

After computing the 3D information for the training set of images, the next step is to train the Logitboost classifier. We implemented a multi-class version of the binary Logitboost classifier released by Tosato et al. [18][5]. The observation are computed for every superpixel obtained by SLIC implementation from the VLFeat library of Vedaldi and Fulkerson [19]. Superpixels with no more than three reconstructed 3D points are discarded from the graph structure. For labeling purposes we assign the class *sky* to them.

The minimum weight spanning tree (MST) is computed from 3D centroids of all the superpixels. Now, using the MST graph, the output of the Logitboost classifier in Eq. 3 and the pairwise potential, Eq. 4, we learn the parameters for the CRF setting. For the learning, inference and decoding with CRFs we use the Matlab code for undirected graphical models (UGM) of Mark Schmidt.[6]

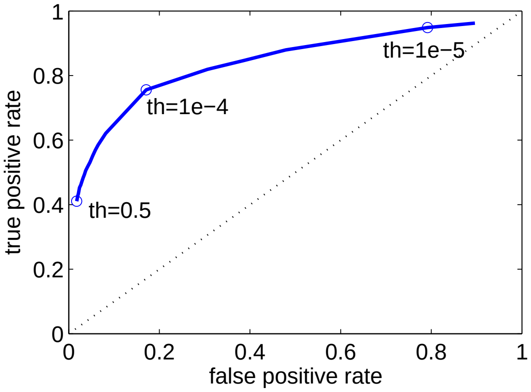


Figure 2: ROC curve for the generic object class.

Unless otherwise specified, the quantities and figures in the next sections are computed only over the testing set of images. The inference results give us the distribution and the assignments over superpixels, we transfer those to every pixel in the superpixel to compute the pixel-wise accuracy of semantic labeling.

4.1 Generic Object Mapping

To determine the probability of objects in a scene, as defined in Section 3, we compute the marginal distribution p in Eq. 2 of each superpixel belonging to one of the four defined classes. We infer the marginals for every node in the graph using the belief propagation method. In Fig. 3 we can see a example of the probability map obtained by our system in a single view setting. We can compute the ROC curve for the generic object detection parametrized by the acceptance threshold th over the probability to be object. The conditions to label a node i as object are either if $p_i(object) > th$ or $p_i(object) = max(p_i)$, we vary th from 0 to 0.5. In Fig. 2 we show the ROC curve.

In Fig. 4 we show the probability map for an experiment on 100 consecutive frames, where our method is able to detect the generic object class despite of its high variability.

4.2 MAP Assignment

To obtain the most likely label assignment for the superpixels we solve the MAP problem. This problem does not require any threshold selection and all the parameters are computed/learned from the data. Table 2 shows the confusion matrices normalized by rows (recall on the diagonal) and by columns (precision on the diagonal). Our approach is able to reach simultaneously high recall and precision for the non-object classes while obtain

[5] Av. from https://sites.google.com/site/diegotosato/academic-activities/code

[6] Available from http://www.di.ens.fr/~mschmidt/Software/UGM.html

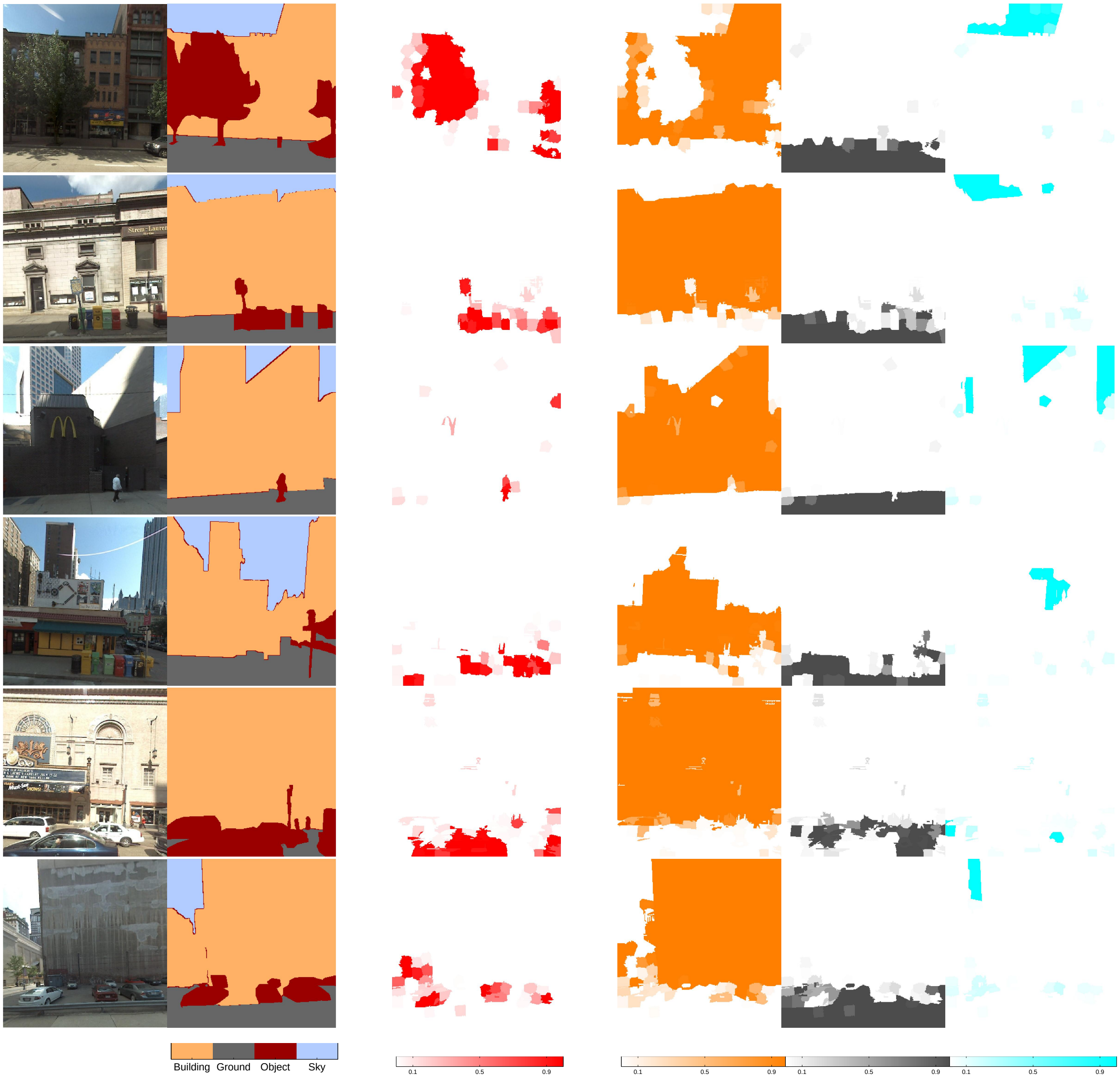


Figure 3: Original image and ground truth labeling on the left. In the middle the probability map for the generic object class, and on the right the probability map for the three remaining classes. Note the common errors in the ground truth, fourth row, where objects are labeled as ground.

a recall of 41% at 67% precision for the generic object class. Which means a $F_{0.5}$ of 0.57, compared for example with a $F_{0.5}$ of 0.52 and 0.61 for foliage and vehicle classes reported by Posner et al. [14] with a labeling system using spatio-temporal context and spending 4s per frame.

Now that we compute the MAP assignment we can make some quantitative comparisons against state of the art methods. The work of Xiao and Quan [20] was the first evaluating a semantic segmentation over the Google Street View dataset. They used a different and bigger set of labelled images (3877 vs 224) for training and (320 vs 96) testing their system. They classified in seven classes: building, ground, sky, person, vehicle, tree and recycle bin. Despite these different experimental settings, we still can compare some numbers. The global pixel accuracy just for the data-term reported by them was 81.2% and ours is 86.46%; the model in single view (without multi-view consistency) was 83.3% and ours is 87.56%. Their system takes 25.7 seconds per frame on average to perform the segmentation.

In Table 3 we show the pixel-wise recall accuracy along with the average and global accuracy for our approach (CRF-MST and Logitboost+SLIC). We also compare our approach with other classifiers used to compute the local prior (data term) in Eq. 3. The classifiers used were: k-nearest neighbours (k-NN) as used by Tighe and Lazebnik [17] and random decision forest (RDF). All the

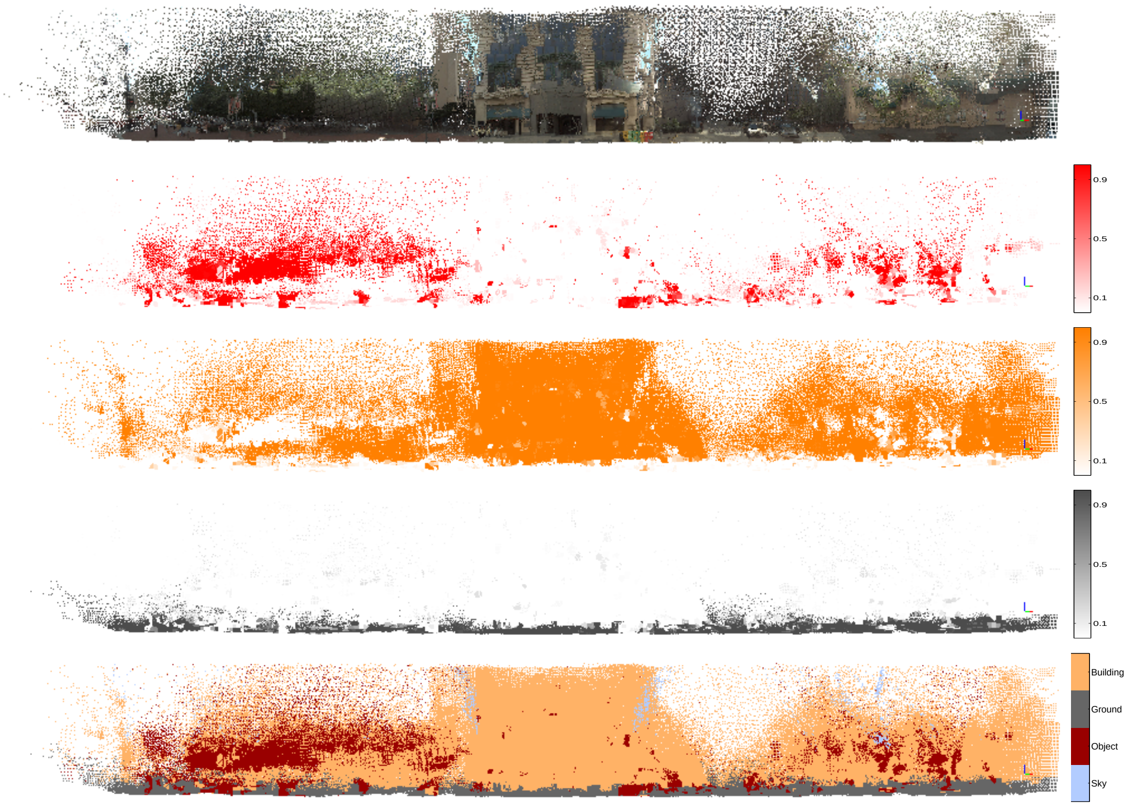


Figure 4: Results over a sequence of 100 frames. The 3D reconstruction (top) and the probability maps for the classes generic objects (red scale), building (orange scale) and ground (gray scale). The MAP assignment is also shown (bottom), where our system found cars, newspaper and trash bins, pedestrians, and trees as part of the object class, shown in the darkest color (red for color versions).

Recall	Building	Ground	Objects	Sky
Building	**91.62**	1.04	1.22	6.12
Ground	6.32	**88.67**	4.63	0.38
Objects	37.75	15.98	**40.76**	5.52
Sky	4.02	0.00	0.17	**95.81**

Precision	Building	Ground	Objects	Sky
Building	**93.82**	4.03	17.94	5.86
Ground	1.44	**87.47**	15.13	0.00
Objects	4.30	7.90	**66.66**	6.40
Sky	0.45	0.00	0.27	**87.74**

Table 2: Confusion matrix for the pixel-wise accuracy, in percentage. Recall and precision values appear on the correspondent diagonals.

		Building	Ground	Objects	Sky	Average	Global
CRF-MST	LB+SLIC	91.62	88.67	40.76	**95.81**	**79.21**	87.56
	K-NN+SLIC	90.38	**92.52**	34.98	95.00	78.22	86.79
	RDF+SLIC	**92.06**	89.84	36.99	95.40	78.57	**87.73**
LB+SLIC (Eq. 3)		91.64	84.67	35.04	94.92	76.57	86.46
Micusik. [13]		75.70	90.20	**42.48**	93.37	75.44	76.72

Table 3: Semantic segmentation for single view in pixel-wise percentage recall accuracy.

classifiers used the same set of observations. In general, Logitboost classifier shows the best general performance followed by RDF and k-NN.We also compare against the full method proposed by Micusik and Košecká [13]. That method uses appearance and geometric cues over watershed superpixel segmentation and takes into account the co-occurrence of superpixels. We train and test their method over the street view images of the current work. Last row of Table 3 shows the results, we can observe that our approach is competitive or better for all the classes, with better average and global accuracy. Their system takes 2.9 seconds per frame on average to perform the segmentation.

A qualitative result of the MAP assignment with our proposal on a trajectory of 100 frames is shown in Fig. 4 bottom.

	Building	Ground	Objects	Sky	Average	Global
SingleView	91.62	88.67	40.76	95.81	79.21	87.56
2 frames	91.49	89.05	41.22	95.52	79.32	87.54
3 frames	91.47	88.83	41.69	95.42	79.35	87.53
4 frames	91.50	89.00	41.73	95.73	79.49	87.60

Table 4: Semantic segmentation for multi-view in pixelwise percentage recall accuracy for our proposal, CRF-MST with Logitboost and SLIC superpixels.

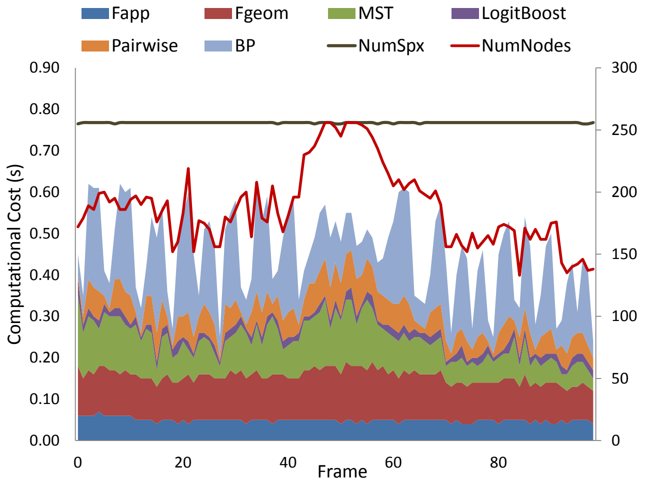


Figure 5: Computational timing performance for the sequence of 100 frames, see Fig. 4.

Finally, as described in Section 3.5, we can obtain the semantic mapping using multiple consecutive frames. We explore the effect of number of consecutive frames over the recall accuracy, see Table 4. The number of frames $k+1$ means taking the scenes from k frames until the current frame. Please note we do not have the ground truth labeling for all the frames in each sequence. The parameter learning for CRFs is done in a single view setting and evaluation is done only on the nodes (superpixels) with ground truth. We obtain a marginal improvement in the average and global accuracy increasing the number of frames considered in the graph, the major improvement was in the generic object class.

4.3 Timing

We compute the timing on the sequence of Fig. 4; our proposal is implemented in Matlab. The computational cost is detailed in Fig. 5, excluding the superpixel oversegmentation and 3D reconstruction. The on-line system runs at 1 fps in a single-thread of a 3.4 GHz IntelCore i7-2600 CPU M350 and 7.8GB of RAM. For the whole system, the average and the maximum times are 488ms and 660ms, respectively. The average cost to obtain the SLIC superpixels is 820ms, although a C++ implementation would take half of that time as reported by [1]. Solving the MAP problem has the same computational cost than obtaining the marginals with the BP algorithm.

5 Discussion

Our method is agnostic to the 3D sensor or reconstruction method used and better accuracy of 3D reconstruction and superpixel segmentation can further improve semantic labeling results. The presented method shows to be robust to errors in the ground truth labeling, where objects are frequently labeled as one of the background classes, see e.g. Fig. 3 fourth row.

The quantitative comparisons of the MAP problem solution with the state of the art of semantic segmentation on urban environments, show that our method improves the performance while still attains the real-time execution. 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 cost is constant with respect to the length of the trajectory. 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 the multi-view setting the size of the graph grows with the number of views used but not with the vehicle trajectory. Given that we are interested in the generic object class we can keep m small.

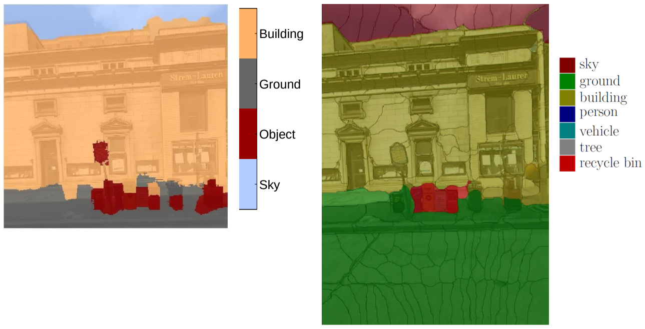
In Fig. 6 we show the result from our method and from Xiao and Quan [20] who use four specific object classes. We can see the disadvantage of multiple specific objects approaches, where the model was not trained for the newspaper box class or trash bin class the solution confuses them with the recycle bin, or assigns an unrelated class (building/ground). We are able to provide a high probability for these object occurrences (e.g. see Fig. 3 second row), and obtain for most of them the correct MAP labels.

We see our proposal as the first stage of a scalable semantic understanding system for mobile robotic. The subsequent stages can use our outcome to find objects or areas of interest to specific tasks of the robot.

6 Conclusion

We have presented a computationally efficient approach for semantic labeling of urban street view sequences into object and non-object categories. The proposed approach effectively uses 3D cues to generate evidence about generic objects. Despite the fact that we do not require object category specific training data, we can achieve better or comparable average accuracy of semantic labeling compared to the state of the art.

We have shown a basic implementation with real time capabilities. We demonstrated that our method can



(a) Our proposal.

(b) Xiao and Quan [20]

Figure 6: The semantic segmentation when solving the MAP problem with our method. The segmentation result from [20] for the same frame (right), who used full uncropped images. Given the set of finite specific objects used in [20] the new/strange objects are misclassified either in the closest related known object (newspaper box as recycle bin) or in a complete different classes (trash bin as building or ground).

work in real scenarios with far from perfect 3D information, illumination artifacts and high variability for all the segmented classes.

Future work is focused on using the semantic segmentation for isolating the stationary part from the detachable, and possibly changing part of the environment. This will allow us to improve other tasks such long-term place recognition or dynamic objects detection/estimation. The presented model can be further extended in a hierarchical manner to incorporate additional information about specific objects of interest if those become available.

7 Acknowledgment

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

References

[1] 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*, 34(11):2274 –2282, nov. 2012. ISSN 0162-8828. doi: 10.1109/TPAMI.2012.120.

[2] B. Alexe, T. Deselaers, and V. Ferrari. What is an object? In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 73 –80, june 2010. doi: 10.1109/CVPR.2010.5540226.

[3] A. Ayvaci and S. Soatto. Detachable object detection: Segmentation and depth ordering from short-baseline video. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(10):1942 –1951, oct. 2012. ISSN 0162-8828. doi: 10.1109/TPAMI.2011.271.

[4] G.J. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla. Segmentation and recognition using structure from motion point clouds. In *Computer Vision - ECCV 2008*, 2008. ISBN 978-3-540-88681-5. doi: 10.1007/978-3-540-88682-2_5.

[5] T. Brox and J. Malik. Large displacement optical flow: Descriptor matching in variational motion estimation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(3):500 –513, march 2011. ISSN 0162-8828. doi: 10.1109/TPAMI.2010.143.

[6] B. Douillard, D. Fox, F. Ramos, and H. Durrant-Whyte. Classification and semantic mapping of urban environments. *Int. J. Rob. Res.*, 30:5–32, January 2011. ISSN 0278-3649. doi: 10.1177/0278364910373409.

[7] D. Eigen and R. Fergus. Nonparametric image parsing using adaptive neighbor sets. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2799 –2806, june 2012. doi: 10.1109/CVPR.2012.6248004.

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

[9] H.S. 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.

[10] L. Ladický, P. Sturgess, K. Alahari, C. Russell, and P.H.S. Torr. What, Where and How Many? Combining Object Detectors and CRFs. In *Computer Vision - ECCV 2010*. Springer Berlin Heidelberg, 2010. ISBN 978-3-642-15560-4. doi: 10.1007/978-3-642-15561-1_31.

[11] J. Lafferty, A. McCallum, and F. Pereira. Conditional Random Fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. 18th International Conf. on Machine Learning*, pages 282–289. Morgan Kaufmann, San Francisco, CA, 2001.

[12] Y. Ma, S. Soatto, J. Kosecka, and S.S. Sastry. *An Invitation to 3-D Vision: From Images to Geometric Models.* Springer Publishing Company, Incorporated, 2010. ISBN 1441918469, 9781441918468.

[13] B. Micusik and J. Košecká. Semantic segmentation of street scenes by superpixel co-occurrence and 3d geometry. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pages 625 –632, Oct. 2009. doi: 10.1109/ICCVW.2009.5457645.

[14] I. Posner, M. Cummins, and P. Newman. A generative framework for fast urban labeling using spatial and temporal context. *Autonomous Robots*, 26:153–170, 2009. ISSN 0929-5593. doi: 10.1007/s10514-009-9110-6.

[15] S. Sengupta, P. Sturgess, L. Ladicky, and P.H.S. Torr. Automatic dense visual semantic mapping from street-level imagery. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 857 –862, oct. 2012. doi: 10.1109/IROS.2012.6385958.

[16] J.-P. Tardif, Y. Pavlidis, and K. Daniilidis. Monocular visual odometry in urban environments using an omnidirectional camera. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 2531 –2538, sept. 2008. doi: 10.1109/IROS.2008.4651205.

[17] J. Tighe and S. Lazebnik. Superparsing: Scalable nonparametric image parsing with superpixels. In *Computer Vision - ECCV 2010*. Springer Berlin Heidelberg, 2010. ISBN 978-3-642-15554-3. doi: 10.1007/978-3-642-15555-0_26.

[18] D. Tosato, M. Farenzena, M. Spera, V. Murino, and M. Cristani. Multi-class classification on riemannian manifolds for video surveillance. In *Computer Vision - ECCV 2010*. Springer Berlin Heidelberg, 2010. ISBN 978-3-642-15551-2. doi: 10.1007/978-3-642-15552-9_28.

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

[20] J. Xiao and L. Quan. Multiple view semantic segmentation for street view images. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 686 –693, oct. 2009. doi: 10.1109/ICCV.2009.5459249.

[21] 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. ISBN 978-3-642-15560-4. doi: 10.1007/978-3-642-15561-1_51.