Recursive Inference for Prediction of Objects in Urban Environments

Cesar Cadena and Jana Košecka

Abstract Future advancements in robotic navigation and mapping rest to a large extent on robust, efficient and more advanced semantic understanding of the surrounding environment. The existing semantic mapping approaches typically consider small number of semantic categories, require complex inference or large number of training examples to achieve desirable performance. In the proposed work we present an efficient approach for predicting locations of generic objects in urban environments by means of semantic segmentation of a video into object and non-object categories. We exploit widely available exemplars of non-object categories (such as road, buildings, vegetation) and use geometric cues which are indicative of the presence of object boundaries to gather the evidence about objects regardless of their category. We formulate the object/non-object semantic segmentation problem in the Conditional Random Field Framework, where the structure of the graph is induced by a minimum spanning tree computed over a 3D point cloud, yielding an efficient algorithm for an exact inference. The chosen 3D representation naturally lends itself for on-line recursive belief updates with a simple soft data association mechanism. We carry out extensive experiments on videos of urban environments acquired by a moving vehicle and show quantitatively and qualitatively the benefits of our proposal.

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 lifelong navigation and mapping. The goal of this work is to endow the models of urban environments with semantic information, which would enable reasoning about presence of different semantic classes (objects) in an on-line setting. We propose to tackle this problem by means of an on-line recursive semantic segmentation of a video stream into object and non-object (road, vegetation, buildings) categories.

The most common techniques for semantic segmentation of urban environments focus on a small number of commonly encountered semantic classes (e.g. road, sky, buildings, trees, cars). While the state of the art of the semantic parsing approaches in outdoors settings achieve relatively high average accuracy of 85-90% on some

Cesar Cadena
George Mason University, USA, e-mail: ccadenal@gmu.edu

Jana Košecka
George Mason University, USA, e-mail: jkosecka@cs.gmu.edu

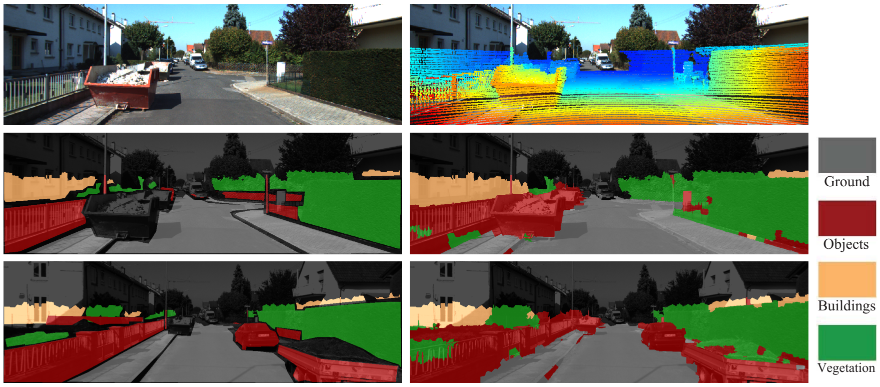


Fig. 1 First row: Original image (left) with the reprojected 3D point cloud (right). Other rows: Ground truth labeling (left) and MAP result from our approach (right).

datasets [29], it is largely due to the fact that majority of 2D or 3D regions belong to non-object semantic categories. These categories such as buildings, roads, vegetation and 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) or exhibit poorer performance when number of object categories grows and the training examples are sparse [19]. The performance could be notably improved by using a specialized sliding window based object detector pipelines [25] or explicit models of higher order dependencies between individual regions captured by higher order potentials in MRF (CRF) framework [14, 7]. These approaches however are not suitable for an on-line setting and typically require a large amount of training examples and/or an expensive training and inference procedure.

In our approach instead of modeling complex spatial and label dependencies or requiring large number of training examples for object categories, we propose an intermediate semantic representation of urban scenes into a single generic *object* category and non-object categories of *road, building* and *vegetation*. In order to effectively gather evidence about generic objects regardless of their category, we use informative 3D features indicative of occlusion boundaries and depth ordering cues, which were found previously useful in research on perceptual grouping.

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 the 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 point cloud. 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 scene structure and allows exact and efficient inference amenable for real-time implementation; 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 an on-line inference; d) 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 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 existing approaches vary in the number and types of semantic classes they consider, the sensing modality, features and inference algorithms.

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. Authors in [5] consider 2D semantic mapping over street laser/image data providing computationally intensive solution on a graph induced by Delaunay triangulation. Both laser and image measurements are used in [21], where efficient solution is provided considering only vehicles as object class. Dense stereo reconstruction was used on CamVid urban sequences by [30] further improving the performance, but considering seven specific object classes.

In computer vision community the problem of simultaneous segmentation and categorization of image regions was typically considered in a single view setting. Non-parametric approaches of [24], and [6] 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.

In addition to a single view setting several approaches explicitly modeled temporal relationships between the frames in the inference problem or exploited 3D structure obtained from visual reconstruction [29]. These strategies further improved the labeling performance while still considering a small number of object categories, with objects being trees, cars, persons and recycle bins. Recently, Sengupta at al. [23] have obtained 3D semantic models in urban scenes where the labeling of every single image was transferred to the 3D reconstruction by voting. However, neither the 3D information nor the sequential nature was used for the inference itself. Floros and Leibe [7] segment urban scenes using images and 3D in a CRFs setting using high order potentials between 3D points with their reprojection in several images in the sequence. Their system is only applicable to static environments and is not amenable for real-time implementation. In our case we directly formulate the inference on a graph induced by 3D structure of the scene, which enables us to use the odometry to guide a soft data association between consecutive frames. Our formulation hence naturally fits the standard probabilistic recursive filtering approach suitable for variety of perceptual tasks.

Additional ideas related to the problem of generic object detection can be found in works which exploit different models of saliency, various perceptual grouping cues and unsupervised object discovery. Alexe et. al. in [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. Our approach is motivated by work of [3], which explicitly reasons about evidence of occlusions boundaries extracted from optical flow and relative depth ordering cues. Also related to our work are several attempts to discover objects in urban scenes. In [26] authors propose a completely unsupervised approach to semantic parsing of outdoors scenes based on the idea of online clustering, demonstrating a capability of discovering categories of car, vegetation, building and ground place in the absence of any labelled data. While this approach is very effective for large objects, generalization to a large number of categories and possibly small objects is be more difficult. Alternative approach for parsing the environments into static parts and moving objects has been recently proposed in [28]. The authors demonstrated successful detection of cars, pedestrians and bicyclists in 3D laser scenes, using the independent motion cue. We view our approach as complementary to the previously proposed techniques. The proposed semantic segmentation of video into object and non-object categories yields a representation that can be used effectively as priors for detection of more broader class of categories (e.g. mailboxes, traffic/road signs, fire hydrants, moving objects).

3 The approach

3.1 Method

We formulate the semantic parsing in the framework of Conditional Random Fields (CRFs) with a tree graph structure encoding the pairwise relationships. We assume that an image and a 3D point cloud of the scene are available. Our approach starts by over-segmenting the image using the efficient *simple linear iterative clustering* (SLIC) algorithm [1]. Every superpixel in the image is interpreted as a cluster in the 3D point cloud for further computations. The 3D centroid of each cluster is used to compute a minimum spanning tree over Euclidean distances, 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 framework. In the remainder of this section we detail the components of our approach and explain the intuition behind them.

3.2 Framework: Conditional Random Fields

Conditional random fields are probabilistic undirected graphical models first developed by [16] for labelling sequence data. CRFs are a case of Markov Random Fields, and thus satisfy the Markov properties. 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 attributes that are too complex to model their probability distribution and the assumption of independence, as in a naive Bayes setting, is too strong [12].

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: $\mathbf{x}_i = \{ground, objects, building, vegetation\}$.

A CRF factorizes the conditional distribution into a product of *potentials*. We consider only the potentials for 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 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:

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

where $Z(\mathbf{z})$ is the normalizing partition function, and $\langle \mathscr{N}, \mathscr{E} \rangle$ 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:

$$p(\mathbf{x}|\mathbf{z}) = \frac{1}{Z(\mathbf{z})} \exp\left(w_1 \sum_{i \in \mathscr{N}} \mathbf{f}(\mathbf{x}_i, \mathbf{z}) + w_2 \sum_{i,j \in \mathscr{E}} \mathbf{g}_c(\mathbf{x}_{i,j}, \mathbf{z}) + w_3 \sum_{i,j \in \mathscr{E}} \mathbf{g}_m(\mathbf{x}_{i,j}, \mathbf{z}) \right) \tag{2}$$

where $\mathbf{w} = [w_1, w_2, w_3]$ is a weight vector, which represents the importance of each term. CRFs learn these weights discriminatively by maximizing the conditional likelihood of labeled training data. We will describe every term of Eq. 2 in detail in 3.4. 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 hid-

Table 1 Local observations

Source	Default	Observation	Dim.	Comments
Image		LAB color	6	Mean and standard deviation
		RGB color	6	Mean and standard deviation
		y_s	1	Vertical pixel location
3D		$\tilde{\mathbf{p}}_s$	3	$[x_s, abs(y_s), z_s]$
	0	$(\mu_{\Delta d_s}, \sigma_{\Delta d_s})$	2	if $d_s < \frac{1}{\lVert N \rVert} \sum_{j \in N}(d_j)$, $\Delta d_s = \lVert d_s - d_{j \in N} \rVert$
	0	$1 - mean(\lVert \mathbf{n}_s \mathbf{n}_N \rVert)$	1	Neighbouring planarity
	0	$\frac{3\sigma_3}{\sigma_1 + \sigma_2 + \sigma_3}$	1	Superpixel planarity
	0	$\lVert \mathbf{n}_s \cdot \hat{\mathbf{k}} \rVert$	1	Superpixel vertical orientation

den 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.

3.3 Minimum Spanning Tree over 3D distances

Instead of computing the graph at the pixel level, we over segment the image into superpixels. The CRF graph structure is typically determined by the neighbourhood relations between superpixels, often connecting unrelated semantic classes (e.g. a person's head with the sky in the background). 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. 2(b). Given that our graph structure is a tree we use the *belief propagation* algorithm [12] to infer the probability class of each node.

3.4 Feature Functions Description

Now we define the feature functions $\mathbf{f}(\mathbf{x}, \mathbf{z})$ and $\mathbf{g}(\mathbf{x}, \mathbf{z})$ in Eq. 2. Starting by the data-term for each superpixel s that is 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 [24] in Eq. 4.

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

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 the image (*Image Features*) and the depth cues (*3D Features*) are summarized in Table 1 and described next.

Image Features: From the appearance of the superpixel we only use color cues and its vertical location, as follows: the mean and standard deviation of each channel in the LAB and RGB color spaces for the superpixel, and the vertical pixel coordinate for the superpixel's centroid.

3D Features: For the 3D point cloud we use cues from the position and planarity, for the superpixel itself and for the superpixel with respect to its neighbourhood. The cues are: the modified 3D coordinate $\tilde{\mathbf{p}}$ for the superpixel's centroid with the absolute value in its lateral coordinate, then we have depth, height and positive lateral distance; the mean and standard deviation of the absolute difference between the depth d_s and the neighbourhood's depths: $\|d_s - d_{j\in N}\|$, but 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 encoded by the curvature of a superpixels' point cloud [11] using the SVD and sort the singular values such that $\sigma_1 > \sigma_2 > \sigma_3$; the neighbourhood planarity computed as one minus the mean of the dot product between the normal to the plane against the neighbourhood normals [30], where the normal corresponds to the singular vector associated to σ_3; and, the superpixel vertical orientation as an absolute value of its normal's vertical component.

The superpixel neighbourhood N refers to all the superpixels in contact with superpixel s in the image. In Table 1 we also show the default values and the dimensionality of these observations. As a result we compute for each superpixel a very simple 21 dimensional feature vector with 13 elements from *Image features* and 8 from *3D features*.

Pairwise potentials

We define two pairwise potentials, one capturing the color proximity $\mathbf{g}_c(\mathbf{x}_i,\mathbf{x}_j,\mathbf{z})$ and the other the metric proximity $\mathbf{g}_m(\mathbf{x}_i,\mathbf{x}_j,\mathbf{z})$ of two superpixels. The potentials are:

$$\mathbf{g}_c(\mathbf{x}_i,\mathbf{x}_j,\mathbf{z}) = \begin{cases} 1-\exp\left(-\|\mathbf{c}_i-\mathbf{c}_j\|_2\right) \rightarrow l_i = l_j \\ \exp\left(-\|\mathbf{c}_i-\mathbf{c}_j\|_2\right) \quad \rightarrow l_i \neq l_j \end{cases} \tag{5}$$

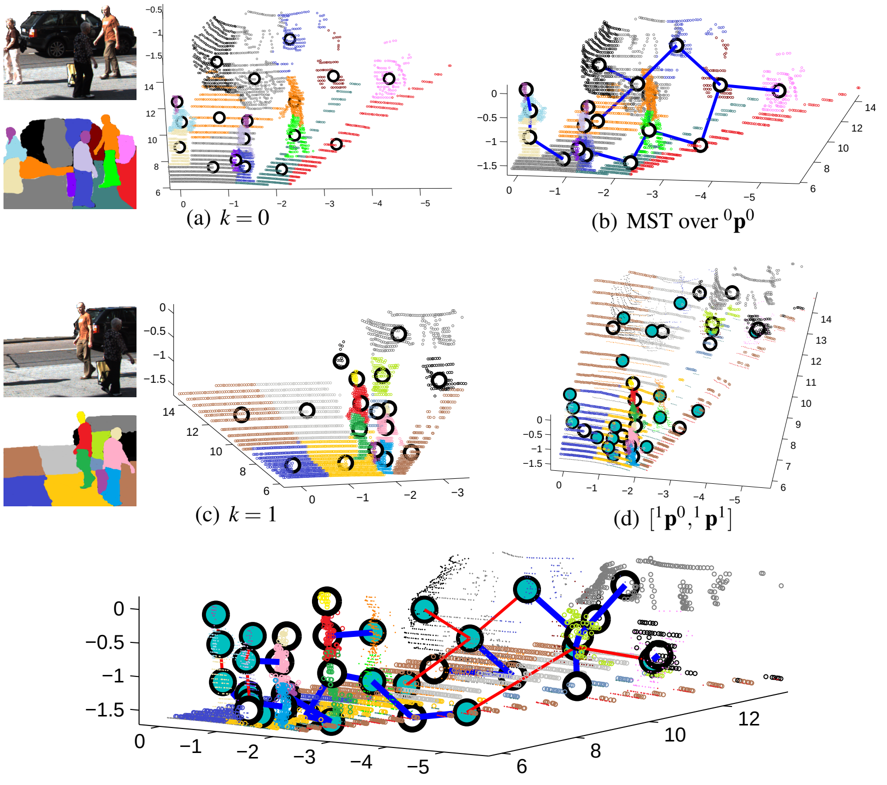
$$\mathbf{g}_m(\mathbf{x}_i,\mathbf{x}_j,\mathbf{z}) = \begin{cases} 1-\exp\left(-\|\mathbf{p}_i-\mathbf{p}_j\|_2\right) \rightarrow l_i = l_j \\ \exp\left(-\|\mathbf{p}_i-\mathbf{p}_j\|_2\right) \quad \rightarrow l_i \neq l_j \end{cases} \tag{6}$$

where $\|\mathbf{c}_i-\mathbf{c}_j\|_2$ and $\|\mathbf{p}_i-\mathbf{p}_j\|_2$ are the L2-Norm of the difference between the mean colors in the LAB-color space, and centroid's 3D positions, respectively, of two superpixels and l is the class label.

3.5 Recursive Inference

So far we have described all the components to carry out the inference in a single frame composed by an image and a 3D point cloud. But in a normal operation of a robot, this information comes streamed. We would like to take an advantage of the sequential nature of the data to carry out the inference without losing the single frame if needed, while keeping the robustness against, for instance, lost frames or odometry failures.

Let C_k be the coordinate frame of reference in time k and ${}^{k}\mathbf{p}^{k-1}$ the 3D coordinates of the nodes $\mathbf{x}$ in frame $k-1$ in C_k. Let ${}_{k-1}^{\;\;k}\mathscr{T}$ be the transformation from C_{k-1}



(a) $k = 0$

(b) MST over ${}^{0}\mathbf{p}^{0}$

(c) $k = 1$

(d) $[{}^{1}\mathbf{p}^{0}, {}^{1}\mathbf{p}^{1}]$

(e) MST over $[{}^{1}\mathbf{p}^{0}, {}^{1}\mathbf{p}^{1}]$

Fig. 2 Toy example for the sequence semantic segmentation process. (a) Suppose that the robot gathers the image and point cloud at time $k = 0$, the image is over-segmented and the nodes are placed in the clusters's centroids. (b) The Euclidean MST provide the graph structure for our CRF, where the inference takes place. (c) The same as (a) in the next time step. (d) We transform the centroids from the previous time step to the current one, the state of the corresponding nodes is now known (blue filled circles). (e) A new MST is computed between all the nodes, known and unknown. The new graphical model is a forest as the edges between known nodes (red lines) are not used in the inference process.

to C_k given by the robot odometry. In the first frame $k = 0$, we infer the state of each node in the graph as described before for a single frame. For $k = 1$ we follow the procedure as in single frame case, computing the superpixels, the features and the data-term. Then we transform the 3D coordinates of nodes at $k = 0$ to C_1 by computing ${}^{1}\mathbf{p}^{0} = {}^{1}_{0}\mathscr{T} \times {}^{0}\mathbf{p}^{0}$. With the set $[{}^{1}\mathbf{p}^{0}, {}^{1}\mathbf{p}^{1}]$ a new MST is computed. These steps are described in Fig. 2. Now, we can proceed to compute the pairwise potentials over the edges in this new graph and carry out the inference, estimating the state of ${}^{1}\mathbf{x}^{1}$ conditioned over the states of ${}^{0}\mathbf{x}^{0}$. This process is repeated for the next time k, conditioning only over the state in $k-1$. The nodes in $k-2$ are no more taken into account as we assume a filtering approach for the inference, Eq. 7.

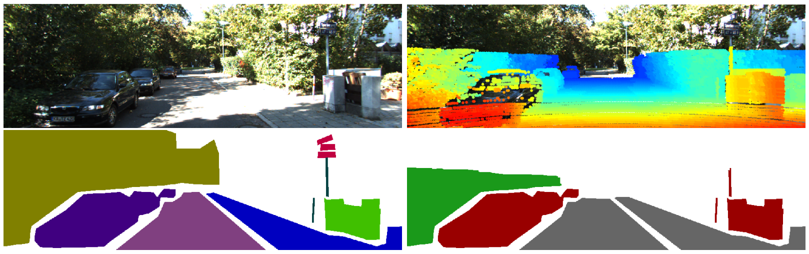


Fig. 3 Original ground truth as released by [23] and the effective ground truth with 3D information used in this paper.

$$p(\mathbf{x}^k|\mathbf{x}^{k-1},\mathbf{x}^{k-2},\ldots,\mathbf{x}^0,\mathbf{z}^k,\mathbf{z}^{k-1},\ldots,\mathbf{z}^0) = p(\mathbf{x}^k|\mathbf{x}^{k-1},\mathbf{z}^k) \quad (7)$$

Note that we have omitted the left superscript in Eq. 7 to denote the coordinate frame of reference as this choice does not affect the inference as long as all the nodes are expressed in a common reference frame.

Given the observed nodes the graph structure becomes a forest, see Fig. 2(e). We can find two extreme cases when we carry out the inference to compute $p(\mathbf{x}^k|\mathbf{x}^{k-1},\mathbf{z}^k)$. The first one is when the robot and the scene are static, in this case the spatial locations of $\mathbf{x}^k$ are expected to be very close to $\mathbf{x}^{k-1}$. Since the $\mathbf{x}^{k-1}$ is now treated as evidence, we would obtain at most a forest with n^k (number of nodes in time k) trees of size 2 connecting the corresponding nodes between $k-1$ and k. The inference for each node $\mathbf{x}_i^k$ is just a weighted average between the state from the local evidence $\mathbf{z}^k$ through the data term and the already inferred state $\mathbf{x}_i^{k-1}$ through the pairwise terms. The second case is when the motion between consecutive frames exceeds the range of the sensor, in which case would be only one edge connecting $\mathbf{x}^k$ and $\mathbf{x}^{k-1}$. In this case $p(\mathbf{x}^k|\mathbf{x}^{k-1},\mathbf{z}^k)$ approaches a single frame case $p(\mathbf{x}^k|\mathbf{z}^k)$ as the distance between frames increases.

Our MST connecting point clouds from two different timestamps gives us a robust way to avoid common errors from data association algorithms in dynamic environments. Even more, this MST graph structure is only telling us that there is a relation between the connected nodes and that it is likely, up to their 3D distance, they belong to the same category; but not, whether they are the same physical entity or a landmark.

4 Experiments

For our experiments we use the KITTI dataset [9], which contains images (1240x380) and 3D laser data taken for a vehicle in different scenarios. We demonstrate the performance of our approach on data from urban residential and city scenes. There are 70 manually labelled non-sequential images as ground truth made available by [23], 45 for training and 25 for testing. The original classes released by [23] are: road, building, vehicle, pedestrian, pavement, vegetation, sky, signal, post/pole and fence. We have mapped those to the four classes: ground (road and pavement), building, vegetation, and things (vehicle, pedestrian, signal, pole and fence). The sky class is omitted as we carry out the inference only in the portion of the image where we

Table 2 Semantic segmentation for single view in pixel-wise percentage recall accuracy.

	Ground	Objects	Building	Vegetation	Average	Global
Data-term: k-NN (Eq. 3)	96.8	75.9	80.7	77.6	82.8	83.5
CRF_MST_k-NN						
only Image Features	96.8	49.2	64.6	**95.5**	76.5	76.8
only 3D Features	95.9	84.2	80.5	46.7	76.8	78.8
All	**97.3**	82.9	82.8	86.9	**87.5**	**88.4**

Table 3 Results and timing for single view vs recursive segmentation.

		Ground	Objects	Bulding	Vegetation	Time MST	Time BP
Single view	Recall	0.973	0.829	0.828	0.869	21ms	164ms
	Precision	0.981	0.881	0.916	0.759		
	F_1	0.977	0.854	0.870	0.811		
Recursive Inference	Recall	0.975	0.836	0.832	0.855	57ms	69ms
	Precision	0.980	0.871	0.931	0.767		
	F_1	0.977	0.853	0.879	0.809		

have 3D data. For our system the effective manually labeled region in the image is reduced as the laser returns span up to 3.2m above the ground and the maximum range that we consider is 30m. As such we use only that ground truth image region for training and for the quantitative evaluations in testing, see Fig. 3.

We obtain the superpixel segmentation using SLIC implementation from the VLFeat library of [27], followed by the computation of the features described in Table 1. With the computed features in the training set we build a kd-tree using the implementation of [4] with the default parameters. We obtain the k-NN classification for the data using Eq. 4 with the $k = 10$ nearest neighbours.

Now, using the MST graph, the output of the local classifier in Eq. 3 and the pairwise potentials, Eq. 5, 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).[1]

At the testing time, to obtain the most likely label assignment for the superpixels we solve the MAP problem for the model. This problem does not require any threshold selection and all the parameters are 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. 1 we show several examples of the output of our approach in the single frame setting.

In Table 2 we show the pixel-wise recall accuracy along with the average and global accuracy for our approach: CRF_MST_k-NN which uses 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 *3D Features* show the corresponding performance when using one kind of features. The main contribution of the images features is placed in the *vegetation* class, while the 3D features improve the *objects* and *building* classes. Our full system, with both sets of features obtains the best trade off between all performance measures.

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

The row *data-term* in Table 2 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.

Given that the manually labeled testing frames belong to the same KITTI sequence (000015), we run our approach over the full sequence and compute the impact on the accuracy by our recursive inference processing. To that end we compute the odometry using the open-source libviso2 [10], using the default parameters provided with the library. Table 3 shows the results and the timing for the stages that change with respect to single view segmentation. We can see that there is no significant difference in the accuracy, and even when the cost of computing the MST with greater number of nodes has increased, the belief propagation is now even more efficient. Our approach is able to reach simultaneously high recall and precision for all the classes, with $F_{0.5}$ of 0.96 for *ground*, 0.87 for *objects*, 0.90 for *buildings*, and 0.78 for *vegetation*, compared for example with a $F_{0.5}$ of 0.62 and 0.52 for grass and bush, 0.91 and 0.67 for tarmac and dirt path, and 0.78 and 0.71 for textured and smooth walls, reported by [21] with a labeling system using spatio-temporal context and spending 4s per frame.

Although our approach is not directly comparable to [23] given the differences in the classes, sensor modalities and the effective region used as ground truth we can see similar performance in the average and global accuracies, 81.7% and 88.4% reported by [23] with respect to 87.5% and 88.4% for us.

The results over the full sequence 000015 of 1900 frames is shown in Fig. 4. The 3D laser returns are reprojected over the odometry. For clarity we split the point cloud in four views corresponding to each class. This trajectory corresponds to a loop in a residential part of the city. To detect the loop closures we use the DLoopDetector library [8], with the default vocabulary and parameters for BRIEF and geometrical checking with the epipolar constraint. To find the transformation between candidates to loop closures we use ICP over the two point clouds excluding those assigned as *objects*. This choice is because this class contains the entities that could be dynamic (cars, people, bikes), hence seems reasonable enough aligning two point clouds by their more stable components during one day: *ground, buildings, vegetation*. The result after optimize the pose graph with g2o [13] is shown in Fig. 4 left.

To test how our system performs facing new environments, we carry out the same semantic recursive inference over a sequence taken in the downtown with high density of dynamic objects, see Fig. 5. In Fig. 5(a) two images are shown, at the beginning and at the middle of the sequence. The probability of belonging to the *objects* class is shown in the second row. We can see how the pedestrians, cars, poles, chairs are all included in this class even when we do not have some of these specific instances in the training data. In the third row of Fig. 5(a) we show the final reprojected point cloud with textured of *objects* class from locations close images in the first row, the *ground* is also shown in grey only for reference. In the second column we can see the trajectories of the pedestrians. In Fig. 5(b) below, we show the reprojected point cloud over the odometry trajectory for classes *ground* and *building*. A zoom up in the second location of Fig. 5(a) is shown on the top

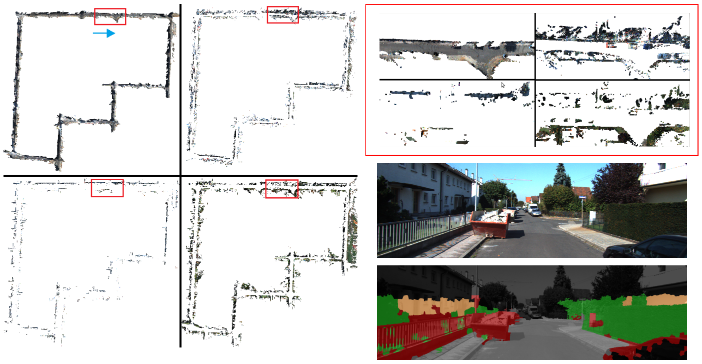
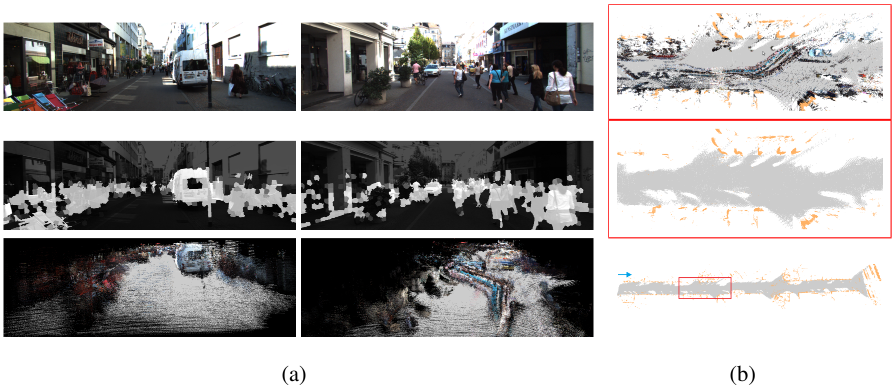


Fig. 4 Results in KITTI sequence 000015 in a residential environment after optimize the trajectory with the detected loop closures. In every image is shown the segmented points belonging to *ground, objects, building* and *vegetation* (left to right and up to down). On the right we show a zoom up along with a image of the corresponding region and the segmented results.



(a) (b)

Fig. 5 Segmentation by our recursive inference process in a high dynamic street in downtown. (a) A couple of frames with the highlighted *objects* class. (b) Reprojected point cloud for the *ground* and *building* class, on the top we show also the *objects* class.

and middle. In the first row the *objects* is included to highlight the importance of recognize this class to allow a better and reliable 3D mapping.

LEUVEN Dataset:

We have also tested our system on the LEUVEN dataset [17], where a set of 70 labeled images and disparity maps were released by [15]. The images were gathered in a residential neighbourhood at 316x216 pixels in resolution. We map the original eight classes to three as follows: ground (road and sidewalk), building, and things (car, person and bike). The sky again was omitted and the vegetation class was not present in the labeling data. We show some of our results in Fig. 6.

We obtain a pixel-wise recall of 97.6% (*ground*), 79.3% (*objects*) and 98.4% (*buildings*). While our average and global accuracies are 91.8% and 95.9%, [7] report 82.4% and 95.4%, and [15] report 84.9% and 95.8%, respectively. Note the dif-

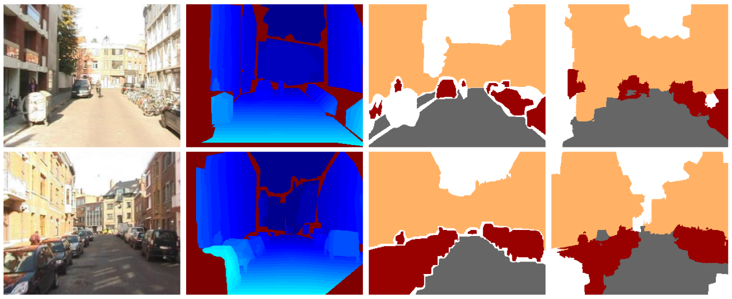
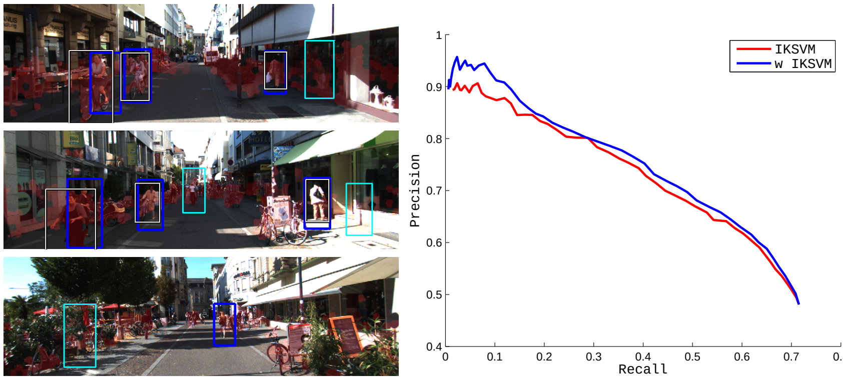


Fig. 6 LEUVEN dataset. From the left, original image, disparity map, ground truth labeling, MAP result from our proposal.



(a) Bounding box detections.

(b) Precision-Recall.

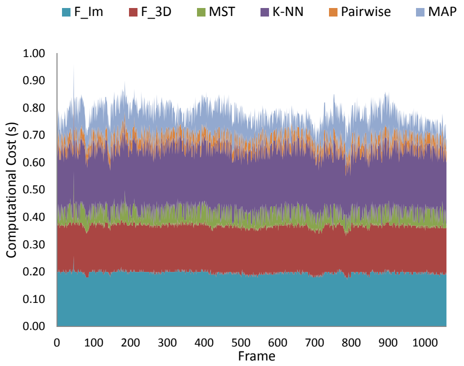
Fig. 7 Pedestrian detection on a high dynamic sequence. (a) Bounding box for ground truth (white), IKSVM [18] (blue and cyan) and weighted IKSVM with the evidence for the *objects* class inside the box (blue). The *objects* class is highlighted in red. (b) The precision-recall curve for this sequence with the IKSVM alone and weighted IKSVM by our objectness evidence.

ference for each system, [7] and [15] included the sky but omitted the person class. Floros and Leibe [7] use at least 5 frames for the 3D reconstruction and manually tune the parameters for the CRF. While Ladický et al. [15] solve a more complicated problem inferring the disparity map jointly with the segmentation.

4.1 Pedestrian Detection

We can combine our segmentation with different object detectors. As an example of this idea we present the results of combination of the efficient approximation of HOG based pedestrian detection with sliding windows (IKSVM) of [18], with our evidence of objects for the downtown sequence in Fig. 5. We compute the proportion of pixels inside the bounding box belonging to the *objects* class to measure the ”objectness” of the box. By weighting the score from the pedestrian detector with this objectness measure the precision is improved, see Fig. 7(b). We can see in Fig. 7(a) how detections in walls or vegetation are rejected (cyan) by the weighted IKSVM. Note in the second image the bounding box in the center is too large for

Fig. 8 Computational timing performance for a sequence in downtown with our recursive inference approach, see Fig. 5.



the actual size of the corresponding pedestrian resulting in rejection because a low objectness. We are currently working to integrate the outcome of our segmentation to guide the pedestrian (and other objects) detection, to improve the efficiency and performance of the sliding window approach.

4.2 Timing

We compute the timing on the sequence of Fig. 5; our proposal is implemented in Matlab. The computational cost is detailed in Fig. 8, excluding the superpixel over-segmentation and the odometry computation. 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 778ms and 960ms, respectively. Although the cost to obtain the SLIC superpixels in our current implementation is a bottleneck with almost 1.5 seconds, the GPU implementation (gSLIC) of Ren and Reid [22] takes only 86ms for images of size 1280x960. The libviso2 library spends 50ms per frame to compute the odometry. Solving the MAP problem has the same computational cost than obtaining the marginals with the BP algorithm. In the case of the LEUVEN dataset, our proposal spends in total only 231ms/282ms in average/max because of the lower resolution.

5 Discussion

We have presented a computationally efficient approach for semantic labeling of urban street view sequences into structural, natural and object categories. The proposed approach uses effectively 3D cues to generate evidence about the presence of objects.

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, this choice enables 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 $\mathscr{O}(nm^2)$, where n is the number of nodes in the graph, and m the number of classes.

Our recursive inference process consistently propagates the semantic segmentation over time in a efficient way, keeping a robust performance with not necessity of

expensive data association techniques. The alignment transformation between two consecutive frames is not needed to be perfect as we can assume with the current state of the art in (visual, laser, IMU) odometry systems a reasonable accuracy. Still, three possible failures could affect the recursive processing: frames are lost during the data acquisition, the odometry system fails but it is detectable, and finally the odometry fails resulting in a different motion model than the actual. In the two first scenarios, the next acquired frame or during the detected failure our approach can handle it as a single frame inference problem. In the last case, the data term is not affected by this failure but the propagation of evidence through the graph and hence the state estimation would be affected. An analysis of this influence is part of our future research. So far we have shown a basic research implementation for a very efficient recursive inference process and we expect to provide a C++ implementation with real time capabilities.
We demonstrated that our method can work in real scenarios, with dynamic objects, in a different kind of streets from the training and with objects never seen before.

We see our proposal as the first stage of a scalable semantic understanding system for mobile robots. The subsequent stages can use the obtained representation for finding objects or use it for isolating the stationary part from the dynamic part of the environment. This can further improve other tasks such long-term place recognition or dynamic objects detection/estimation. The presented model can be extended in a hierarchical manner to incorporate additional information about specific objects of interest if those become available. Our method can be used in conjunction with [20] to estimate the motion model of generic objects even if they are static in the scene.

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.
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.
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.
4. 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*, volume 1, pages 626–633, 2006.
5. 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.
6. 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.
7. G. Floros and B. Leibe. Joint 2d-3d temporally consistent semantic segmentation of street scenes. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2823–2830, 2012.
8. D. Galvez-Lopez and J. D. Tardos. Bags of binary words for fast place recognition in image sequences. *IEEE Transactions on Robotics*, 28(5):1188–1197, October 2012.
9. A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and PatternRecognition (CVPR)*, 2012.

10. A. Geiger, J. Ziegler, and C. Stiller. Stereoscan: Dense 3d reconstruction in real-time. In *Intelligent Vehicles Symposium (IV)*, 2011.
11. K. Klasing. *Aspects of 3D Perception, Abstraction, and Interpretation in Autonomous Mobile Robotics*. PhD thesis, Technical Univeristy of Munich, Munich, Germany, April 2010.
12. D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.
13. R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard. g2o: A general framework for graph optimization. In *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, Shanghai, China, May 2011.
14. 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.
15. L. Ladický, P. Sturgess, C. Russell, S. Sengupta, Y. Bastanlar, W. Clocksin, and P.H.S. Torr. Joint optimization for object class segmentation and dense stereo reconstruction. *International Journal of Computer Vision*, 100(2):122–133, 2012.
16. 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.
17. B. Leibe, N. Cornelis, K. Cornelis, and L. Van Gool. Dynamic 3d scene analysis from a moving vehicle. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pages 1–8, 2007.
18. S. Maji, A.C. Berg, and J. Malik. Classification using intersection kernel support vector machines is efficient. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, 2008.
19. B. Micusik and J. Kosecka. Semantic segmentation of street scenes by superpixel cooccurrence and 3d geometry. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pages 625 –632, Oct. 2009.
20. Frank Moosmann and Christoph Stiller. Joint self-localization and tracking of generic objects in 3d range data. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1138–1144, Karlsruhe, Germany, May 2013.
21. 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.
22. C.Y. Ren and I. Reid. gSLIC: a real-time implementation of SLIC superpixel segmentation. Technical report, University of Oxford, Department of Engineering, 2011.
23. S. Sengupta, E. Greveson, A. Shahrokni, and P.H.S. Torr. Urban 3D Semantic Modelling Using Stereo Vision. In *ICRA*, 2013.
24. J. Tighe and S. Lazebnik. Superparsing: Scalable nonparametric image parsing with superpixels. In *Computer Vision - ECCV 2010*. Springer Berlin Heidelberg, 2010.
25. J. Tighe and S. Lazebnik. Finding Things: Image Parsing with Regions and Per-Exemplar Detectors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013.
26. R.A. Triebel, R. Paul, D. Rus, and P. Newman. Parsing outdoor scenes from streamed 3d laser data using online clustering and incremental belief updates. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.
27. A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms. http://www.vlfeat.org/, 2008.
28. D. Wang, I. Posner, and P. Newman. What could move? finding cars, pedestrians and bicyclists in 3d laser data. In *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, Minnesota, USA, May 2012.
29. 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.
30. 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.