Structured Output Prediction for Semantic Perception in Autonomous Vehicles

Rein Houthooft, Cedric De Boom, Stijn Verstichel, Femke Ongenae, Filip De Turck
Department of Information Technology (INTEC)
Ghent University - iMinds, Gaston Crommenlaan 8 box 201, B-9050 Ghent, Belgium
{rein.houthooft, cedric.deboom, stijn.verstichel, femke.ongenae, filip.deturck}@ugent.be

Abstract

A key challenge in the realization of autonomous vehicles is the machine's ability to perceive its surrounding environment. This task is tackled through a model that partitions vehicle camera input into distinct semantic classes, by taking into account visual contextual cues. The use of structured machine learning models is investigated, which not only allow for complex input, but also arbitrarily structured output. Towards this goal, an outdoor road scene dataset is constructed with accompanying fine-grained image labelings. For coherent segmentation, a structured predictor is modeled to encode label distributions conditioned on the input images. After optimizing this model through max-margin learning, based on an ontological loss function, efficient classification is realized via graph cuts inference using α-expansion. Both quantitative and qualitative analyses demonstrate that by taking into account contextual relations between pixel segmentation regions within a second-degree neighborhood, spurious label assignments are filtered out, leading to highly accurate semantic segmentations for outdoor scenes.

1 Introduction

Outdoor settings present many challenges to autonomous vehicle operation due to the lack of structured elements in the environment, such as walls or doors (Reina, Milella, and Underwood 2012). Beyond the identification of traversable areas and object detection (Sivaraman and Trivedi 2013; Bernini et al. 2014), environmental sensing can be approached by determining a semantic class for each point of the environment, enabling further high-level processing such as automatic vehicle steering based on road segments. In this paper, we present a semantic perception model for autonomous vehicles based on image segmentation.

In computer vision, two main trends can be discerned, namely segmentation models and bounding box models. The latter attempt at identifying objects by drawing bounding rectangles. Although this approach performs very well at recognizing objects with a regular shape, its performance drops significantly when irregular regions, such as vegetation or sky, have to be identified (Nowozin and Lampert 2011). Segmentation models are not hindered by irregular shapes as they label each individual image pixel. Because outdoor scenes consist largely of irregular shapes, the segmentation approach fits best for autonomous vehicles.

Many segmentation techniques are based on a graphical model that embodies interactions between neighboring over-segmentation regions, i.e., coherent pixel clusters, conditionally-dependent on the input image (see (Nowozin and Lampert 2011) for an overview). We propose a structured prediction model that takes into account contextual relations between both first- and second-degree region neighbors by means of distinct interaction functions. This model is optimized according to an ontological loss function via max-margin learning. Towards our goal of autonomous vehicle perception, a dataset is constructed in which recorded dashboard camera images are labeled in a fine-grained manner. Effective and efficient inference is possible by means of the α-expansion (Boykov, Veksler, and Zabih 2001) algorithm, yielding accurate semantic image segmentations.

2 Related work

This section explores literature related to perception in outdoor autonomous vehicles and outdoor scene understanding, and how our work differs from previous approaches.

Byun et al. (Byun et al. 2015) used a Markov random field for predicting road regions and obstacles that fall out of the perception sensor range. Their approach relies on manually engineered unary and pairwise potential functions. In contrast, our work makes use of structured learning through structural support vector machines (SSVMs) to simultaneously tune all graphical model potentials, allowing the model to adapt to previously recorded data. Bosch et al. (Bosch, Muoz, and Freixenet 2007) investigated the use of segmentation in outdoor environments by means of a probabilistic pixel map and fuzzy classifiers. In contrast, we discard probabilities and focus on the model's discriminative aspect by using an energy-based formulation of our model. Armbrust et al. (Armbrust et al. 2009) built an off-road autonomous vehicle with the goal of operating in highly vegetated terrain. Their system integrates multiple manually-engineered sensor processing systems, and is responsible for traversable region and object detection, omitting any form of semantics. Kelly et al. (Kelly et al. 2006) and Leonard et al. (Leonard et al. 2008) similarly focused on an integration of sensor processing systems in which traversable regions are detected. Geiger et al. (Geiger et al. 2014) proposed a combined scene flow, vanishing point, and scene segmentation approach for 3-D traffic understanding using geometric features. Contrary to their approach, we focus on semantic scene understanding through an SSVM, which enables accurate segmentations based on contextual information. Several authors tackle the problem of outdoor scene segmentation by means of convolutional neural networks (Hadsell et al. 2009; Sermanet et al. 2009; Hadsell et al. 2007). Kuthirummal et al. (Kuthirummal, Das, and Samarasekera 2011) built a traversable region map using a 3-D grid. Other work proposed the use of radar for traversable region detection (Reina, Milella, and Underwood 2012; Milella et al. 2014; 2011), however, this does not permit high-level reasoning about objects and their semantics. Furthermore, Nourani-Vatani et al. (Nourani-Vatani, Lpez-Sastre, and Williams 2015) propose the use of SSVMs with a hierarchical loss function for seafloor imagery classification in autonomous underwater vehicles. In contrast, we focus on segmentation rather than classification.

3 Preprocessing

Before the images are fed to the prediction model, they pass through a preprocessing pipeline in order to extract uniform feature representations. In this section, we first describe the region segmentation approach, after which we explain how features are extracted.

3.1 Region over-segmentation

Classifying every pixel is computationally challenging. Therefore, we first segment the image into visually coherent regions, or superpixels, as shown in Fig. 1. This is done via the SLIC algorithm (Achanta et al. 2012), which clusters all pixels in a 5-D space composed of the CIELab L-, a-, and b-values, and the pixel coordinates. First, a grid is defined on top of the image, whose nodes serve as starting positions for K distinct clusters. The clustering distance used is a weighted sum of both the distance in the (L, a, b)-space and the (x, y)-space. The weights allow for tuning the fuzziness of the region boundaries. By means of expectation-maximization, the clusters evolve until convergence.

3.2 Feature extraction

The next step is the feature extraction process in which each distinct region is assigned a uniformly-sized feature vector based on both gradient and color information. Gradient information is extracted as 200-D features through the DAISY (Tola, Lepetit, and Fua 2010) algorithm, while color information is modeled by the HSV-value of each pixel, leading to 3-D color features. What we obtain now is a set of features for each image pixel for the whole training dataset.

Because the segmentation regions vary in size, the number of extracted features also varies. However, we want to obtain a uniform feature vector for each region, independent of its size. Therefore, after we have extracted features from all pixels, we apply k-means clustering to all extracted gradient features, and to all extracted color features from all image pixels. Gradient features are clustered into G clusters, while color features are clustered into C clusters.

To reduce the clustering complexity, the features are not extracted for each image pixel, but only for pixels that lie on a regular grid. The feature cluster centers now form so-called *words*. After these words have been identified, the extracted features within the same segmentation region are mapped to the closest word in terms of Euclidean distance in the feature space, forming bags-of-words.

Because the number of feature clusters, namely G and C, is fixed, a histogram can be built for each segmentation region that contains the frequency of occurrence of each of the words. Moreover, the relative position of the region center, i.e., the median of its pixel coordinates, is added as a feature. A uniform representation is obtained for each segmentation region by concatenating the two histograms and the center into a single vector in $\mathbb{R}^{(G+C+2)}$, which acts as the input to our prediction model. In contrast to the grid-based feature extraction process used for constructing the words of the bag-of-words model, the features used to construct the uniform feature vector for each region are densely sampled.

4 Structured output prediction

Traditional machine learning models for classification predict a single output through a function $f : \mathcal{X} \to \mathbb{R}$. In contrast, structured prediction models (Nowozin and Lampert 2011) are defined by a function $f : \mathcal{X} \to \mathcal{Y}$ of which the output is arbitrarily structured. In this work, this structure is shaped as a vector in $\mathcal{Y} = \mathcal{L}^V = \{1, \ldots, K\}^V$, with V the number of output variables to be predicted. In our use case, V is the number of regions present in the input image, as explained in Section 3.1. Structured models maximize a compatibility function $g : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ to obtain the predicted value $f(x)$, defined as

$$f(x) = \arg\max_{y \in \mathcal{Y}} g(x, y), \tag{1}$$

which is called inference. In this work, we use a linearly parametrized function $g(x, y) = \langle w, \Psi(x, y) \rangle$, in which w is learned from data and $\Psi(x, y)$ is a joint feature vector of both x and y. Because the domain $\mathcal{Y}$ is very large, as it is a combination of label assignments to all regions, Ψ has to be defined in such a way that underlying structures can be exploited. In this work, we ensure that Ψ identifies with the energy function of an SSVM (Nowozin and Lampert 2011), for which efficient inference techniques exist that correspond to maximizing the compatibility g.

4.1 Structural support vector machines (SSVMs)

We first explain conditional random fields (CRFs), and afterwards the SSVM model as used in this paper is derived. A CRF is a type of probabilistic graphical model, namely the conditional variant of a Markov random field. A graphical model defines a probability distribution in a compact way by making explicit dependencies between random variables. CRFs in particular define not the full joint distribution over all random variables, but the conditional distribution of the labels, given the input. This allows for tractable inference within the model (Nowozin and Lampert 2011).

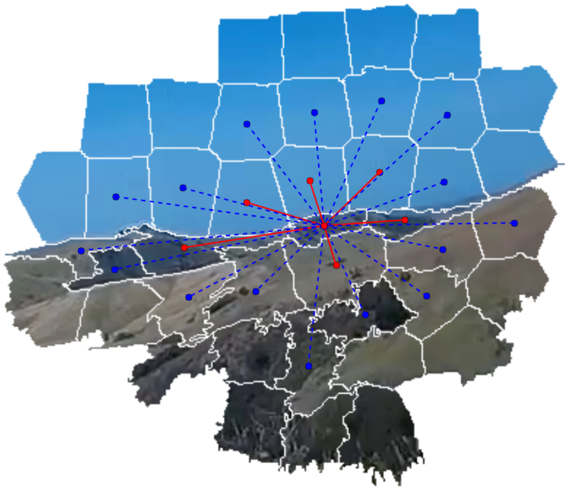

Figure 1: Neighborhood connectivity with both first-degree (solid red) and second-degree (dashed blue) neighbor connections for an actual region over-segmentation

A CRF models the conditional probability distribution $p(y|x)$, with x an observation, and y an assignment of labels in $\mathcal{Y}$. This distribution can be written as

$$p(y|x) = \frac{1}{Z(x)} \prod_{F \in \mathcal{F}} \psi_F(y_F, x_F), \text{ with} \tag{2}$$

$$Z(x) = \sum_{y \in \mathcal{Y}} \prod_{F \in \mathcal{F}} \psi_F(y_F, x_F) \tag{3}$$

called the partition function, which normalizes the distribution. Herein, $\mathcal{F}$ represents the set of factors in the CRF, which model relations between variables. The model as presented here can be split up into two types of factors, namely unary and pairwise factors, as follows:

$$p(y|x) = \frac{1}{Z(x)} \prod_{i \in \mathcal{V}} \psi_i(y_i, x_i) \prod_{(i,j) \in \mathcal{E}} \psi_{ij}(y_i, y_j), \tag{4}$$

for a CRF represented by a graph $G = (\mathcal{V}, \mathcal{E})$. Fig. 2 depicts the graphical model as used in this work, while Fig. 1 shows the second-order neighborhood connections for an actual image over-segmentation. The unary terms $\psi_i(y_i, x_i)$ represent relations between input features x_i and region labels y_i, while the pairwise terms $\psi_{ij}(y_i, y_j)$ model relations between regions. For example, when x_i is a green-colored region, then ψ_i is large if $y_i = \texttt{grass}$ and low if $y_i = \texttt{road}$. For the pairwise factors, ψ_{ij} is high if $y_i = y_j$ and low otherwise. This leads to a smoothing effect by ensuring that proximal and similar regions favor equal labels. In our scenario, interlabel relations are linked together based on each region's first- and second-degree neighborhood[1], as will be explained later on.

Because the normalization function $Z(x)$ is difficult to compute (Nowozin and Lampert 2011) and we are foremost

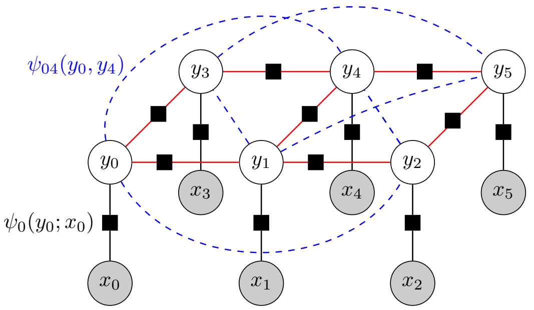

Figure 2: A CRF is shown with second-degree neighborhood connectivity graph, represented as a factor graph with observations x_i representing region feature vectors, and random variables representing label assignments y_i. The blocks represent the factors/potentials (not drawn for second-degree edges), while their incident nodes represent their arguments.

interested in retrieving the most-likely labels, we convert the CRF into an energy-based model by dropping the normalization function $Z(x)$ and defining energy potentials as $E_F(y_F, x_F) = -\log \psi_F(y_F, x_F)$. By altering the CRF in this way, we obtain a model called a structural support vector machine (SSVM) (Nowozin and Lampert 2011), which formulates the total energy as

$$E(y, x) = \sum_{i \in V} E_i(y_i, x_i) + \sum_{(i,j) \in \mathcal{E}} E_{ij}(y_i, y_j). \tag{5}$$

Computing $y^* = \arg\max_{y \in \mathcal{Y}} p(y|x)$, called maximum a priori (MAP) inference, requires maximization of $p(y|x)$, which is the same as minimizing $E(y, x)$. The factor energies are then linearly parametrized (Lucchi et al. 2012) as

$$E_i(y_i, x_i; w^U) = \langle w^U, \psi_i(y_i, x_i) \rangle \text{ and} \tag{6}$$

$$E_{ij}(y_i, y_j; w^P) = \langle w^P, \psi_{ij}(y_i, y_j) \rangle, \tag{7}$$

with w^U and w^P the unary and pairwise parameters respectively. Constructing a weight vector $w = ((w^U)^\top, (w^P)^\top)^\top$, and similarly a feature vector $\Psi(y, x) = ((\Psi^U)^\top, (\Psi^P)^\top)^\top$, with Ψ^U and Ψ^P respectively the sum of all unary features $\psi_i(y_i, x_i)$ and the sum of all pairwise features $\psi_{ij}(y_i, y_j)$, leads to a linear parametrization of E :

$$E(y, x; w) = \langle w, \Psi(y, x) \rangle. \tag{8}$$

In this formulation, the unary features are defined as

$$\psi_i(y_i, x_i) = \left(h\left(x_i\right)^\top \left[y_i = m\right] \right)^\top_{(m \in \mathcal{L})}, \tag{9}$$

with $[\cdot]$ the Iverson brackets, and $h(x_i)$ the probabilistic output of a logistic regression function. This multiclass logistic regression function provides the unary potential input and has been trained on the bag-of-words features $x_i \in \mathbb{R}^{(G+C+2)}$ extracted from the i-th segmentation region, as presented in the Section 3.2. Therefore, $\mathcal{X} = (\mathbb{R}^{(G+C+2)})^V$, with V the variable number of regions

[1] Second-degree neighbors are defined as neighbors of neighbors, based on a region connectivity graph.

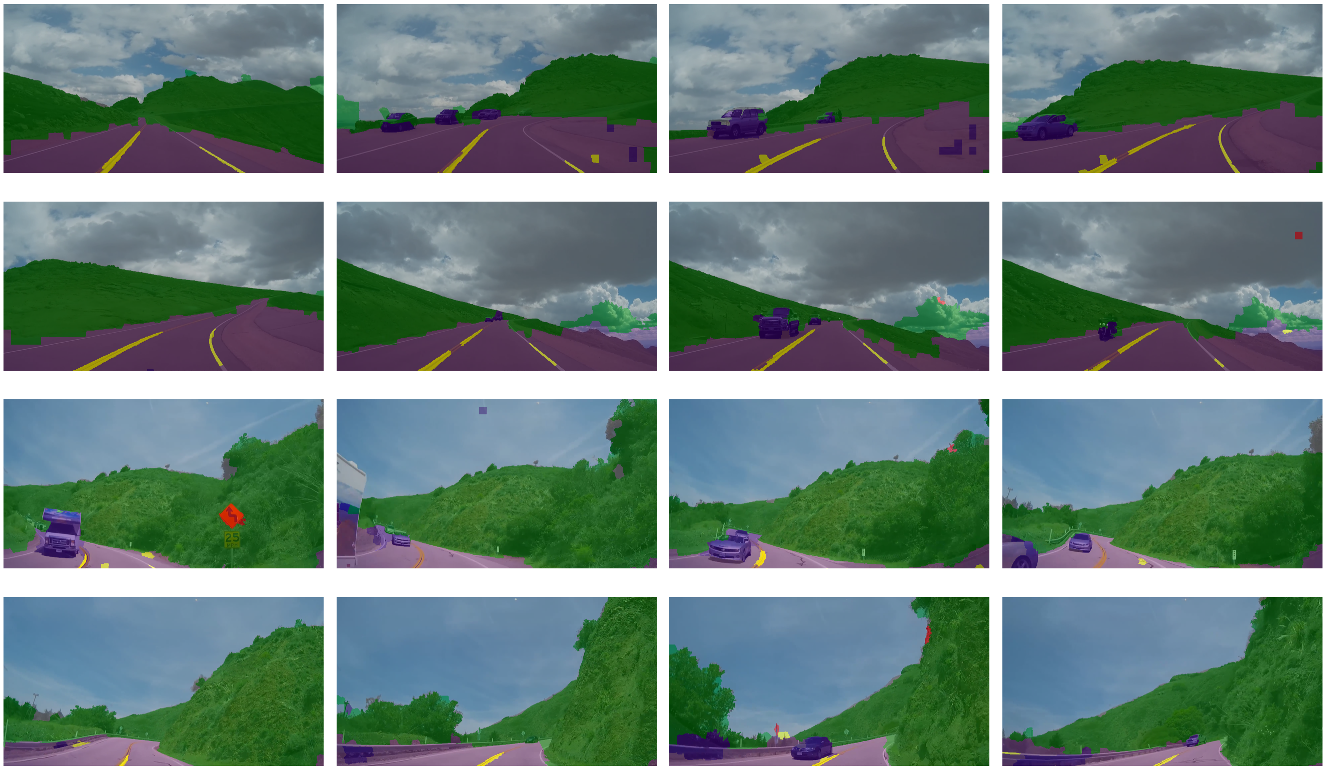

Figure 3: Unseen samples from two dashboard camera videos (time flows from left to right, top to bottom) segmented. Legend: *road* (light purple/pink), *vehicle* (dark purple), *vegetation* (green), *sky* (gray), *road marking* (yellow), and *road sign* (red).

in an image. The function $h : \mathbb{R}^{(G+C+2)} \to [0,1]$ outputs a probability for each class, for each region independently.

To obtain a fine-grained segmentation of the image, e.g., to detect vehicles and signs from a distance, we choose a fine-grained region segmentation of 1000 regions. Because neighboring interactions span a low distance, this leads to noisy predictions. For this reason, both first- and second-degree neighboring regions are connected by means of pairwise potentials, as shown in Fig. 2. Contrary to the unary factors that depend linearly on the inputs $h(x_i)$, the pairwise factors are uniform for all region interactions. However, a different interaction factor is used for first- and second-degree connections, which allows for differentiation between short- and medium-distance interactions. This is encoded into the pairwise features, as defined by

$$\psi_{ij}(y_i, y_j) = \Big(\pi(i,j)[y_i = m \wedge y_j = n]\Big)^{\top}_{((m,n)\in\mathcal{L}^2)}, \quad (10)$$

with $\pi(i,j) = (1,0)$ if the incident nodes of edge $(i,j) \in \mathcal{E}$ are first-degree neighbors, and $\pi(i,j) = (0,1)$ if they are second-degree neighbors. This effectively encodes the separate pairwise table for both first- and second-degree links into the first and second column of w^P.

In Eq. (8), it can be noticed that $E(y,x;w)$ has a form similar to $g(x,y)$, the compatibility function g defined at the beginning of this section. Computing Eq. (1) is therefore equivalent to MAP inference in the SSVM model. Doing so allows us to avoid the intractable computation of g over all y-values in $\mathcal{Y}$ by the use of efficient SSVM MAP inference methods, as will be explained in Section 4.3.

4.2 Max-margin learning

Training the SSVM means tuning the parameters w of the energy function based on a training dataset, such that its predictions generalize well on a test set. This can be done through so-called max-margin learning methods, using quadratic programming. In structured prediction, we minimize a regularized structured risk rather than the Bayes' risk, namely

$$R(w) + \frac{\lambda}{N}\sum_{n=1}^{N} \Delta(y^n, f(x^n)), \quad (11)$$

with $\Delta : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}^+$ the loss function, $R(\cdot)$ a regularization function, λ the inverse regularization strength, and a training set $\{(x^n, y^n)\}_{n\in\{1,\dots,N\}} \subset \mathcal{X} \times \mathcal{Y}$. Due to the piecewise nature of this function, caused by the loss function shape, gradient-based optimization techniques are unusable (Nowozin and Lampert 2011). Therefore, we minimize a convex upper bound of this function

$$w^* = \operatorname*{arg\,min}_{w\in\mathbb{R}^D} \left[R(w) + \frac{\lambda}{N}\sum_{n=1}^{N} u(y^n, x^n; w) \right], \quad (12)$$

with u a function that extracts the maximal difference between the prediction loss and the energy loss for a data sam-

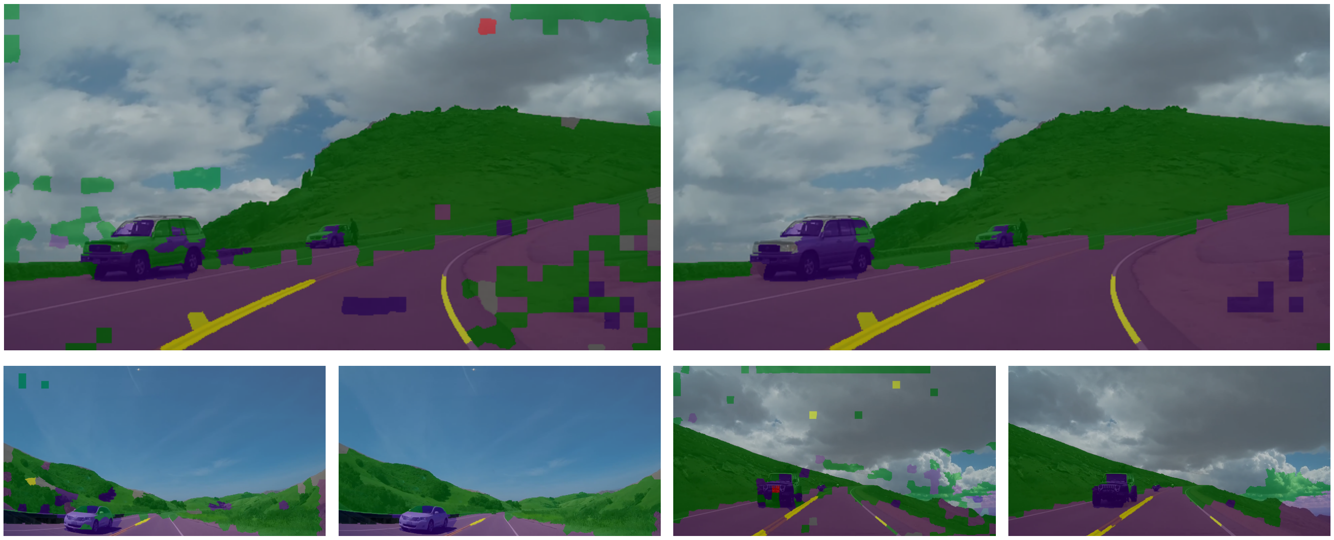

Figure 4: Qualitative comparison between independent logistic regression classification (left images) and structured prediction (right images), using the same classes as in Fig. 3. The top two images feature two vehicles, the bottom left images one vehicle (black colored regions represent unknown labels), and the bottom right images two vehicles. The logistic regression classifier leads to spurious labels, which are filtered out by the structured predictor. The image legend is equal to the legend in Figure 3.

ple (x^n, y^n), defined as

$$u(y^n, x^n; w) = \max_{y \in \mathcal{Y}} \left[\Delta(y, y^n) - \left(E(y, x^n; w) - E\left(y^n, x^n; w\right)\right)\right]. \quad (13)$$

In this work, we optimize this objective function by means of the N-slack cutting plane method (Joachims, Finley, and Yu 2009) with L_2-regularization, which reformulates the above optimization problem as

$$w^* = \arg\min_{w, \xi_1, \ldots, \xi_N} \left[\frac{1}{2}\|w\|^2 + \frac{\lambda}{N}\sum_{i=1}^{N}\xi_i\right], \text{ s.t.}$$
$$E(y, x^n; w) - E(y^n, x^n; w) \geq \Delta(y, y^n) - \xi_n, \quad (14)$$

with $n \in \{1, \ldots, N\}$ and $y \in \mathcal{Y}$. Herein, ξ_n are slack variables, which are introduced to allow for linearly punished constraint violation. By doing this, the maximization function in Eq. (13) is translated into linear constraints. Moreover, the objective function becomes quadratic in w. This allows quadratic optimization, for which various optimization libraries exist. However, a downside is the large number of constraints due to the high dimensionality of $\mathcal{Y}$.

To counter this, we use cutting plane optimization (Joachims, Finley, and Yu 2009), in which a working set of constraints W is utilized. Cutting plane optimization starts by solving the optimization problem with $W = \emptyset$, and iteratively adds additional constraints. Only those constraints which are maximally violated, for each training sample (x^n, y^n), are added to W. This allows fast optimization at the start of the procedure, combined with strong results.

For Δ, we chose a weighted ontological loss function based on the distance between classes in an ontology, as

$$\Delta(y, y^n) = \frac{1}{V}\sum_{i=1}^{V}\theta_{y_i^n}\,\delta\left(y_i^n, y_i\right), \quad (15)$$

with V the number of segmentation regions. The weights $\theta_{y_i^n}$ are set to the inverse of the square root of label frequency in the training set, normalized over all labels, to correct for some class imbalance. The function $\delta : \mathcal{L} \times \mathcal{L} \rightarrow \mathbb{R}^+$ represents the distance along the spanning tree defined by the ontology. As such, the loss of misclassification between two classes that are conceptually far apart is high.

4.3 Reasoning

In order to infer a set of labels that maximally correspond to the regions of an input image, we have to calculate $f(x)$, as defined in Eq. (1). However, brute-force calculation of $f(x)$ is intractable as it involves due to the combinatorial complexity of the set of possible labelings, we rely on using tractable approximate reasoning. An inference technique called α-expansion (Boykov, Veksler, and Zabih 2001) is used. α-expansion breaks up the energy minimization problem of Eq. (1), based on Eq. (5), into sequential subproblems. In a subproblem, SSVM nodes have the possibility to alter their label y_i to a different (but fixed) label α.

Each subproblem is converted into an auxiliary graph of which the nodes and edges are constructed in such a manner, that a graph cut results in an assignment of labels in which y_i remains constant, or changes to α. Because the edge weights are set to the SSVM energies, finding a minimal cut corresponds to a labeling change that results in a minimal energy value (for a particular α switch). Solving the subproblems sequentially for different α-values, yields an approximately optimal labeling. A more in-depth

Table 1: Per-class and average F_1 scores (in %) of the models on the CamVid dataset (Brostow, Fauqueur, and Cipolla 2009)

	animal	archway	cyclist	bridge	building	car	cart/pram	child	pole	fence	drive mark	misc. text	other moving	parking block	pedestrian	road	road shoulder	sidewalk	sign	sky	SUV	traffic light	tree	truck/bus	vegetation	wall	$\bar{F_1}$
SSVM 2-table	**25**	**39**	**58**	**62**	81	**70**	**04**	11	08	**46**	**57**	01	**35**	01	17	89	00	**75**	**15**	90	19	30	**79**	**47**	**32**	**45**	**39.8**
SSVM default	17	26	57	00	**83**	69	**04**	**12**	07	42	**57**	00	32	01	15	**90**	01	**75**	14	**91**	**21**	**31**	**79**	37	30	42	35.9
unary-only	01	03	29	04	56	36	02	02	**10**	28	52	**09**	16	**09**	**23**	80	**08**	59	06	86	20	24	63	11	13	22	25.8

Table 2: Per-class and average F_1 scores (in %) of the models on the KITTI dataset (Ros et al. 2015; Kundu et al. 2014)

	sky	building	road	side walk	fence	vegetation	pole	car	$\bar{F_1}$
SSVM 2-table	**80**	**82**	**79**	**52**	**40**	**79**	02	**70**	**60.7**
SSVM default	**80**	**82**	78	51	35	78	01	69	59.4
unary-only	63	74	65	41	24	73	**10**	45	49.5

analysis, including energy formulation requirements and restrictions, is given in (Boykov, Veksler, and Zabih 2001; Nowozin and Lampert 2011).

5 Results and discussion

To evaluate our model qualitatively in an actual autonomous vehicle setting, we constructed a segmentation dataset using recorded dashboard camera images, accompanied by a fine-grained labeling. Furthermore, we add quantitative performance results based on two publicly available datasets to objectively evaluate our model. Specifically, the autonomous vehicle segmentation dataset KITTI (Ros et al. 2015) is used, augmented with 49 additional train images by (Kundu et al. 2014), leading to 149 train and 46 test images, and the CamVid dataset (Brostow, Fauqueur, and Cipolla 2009), split into 525 train and 175 test images. For both datasets, classes that are severely under-represented are omitted.

Model hyperparmeters are optimized via k-fold cross-validation on the training set. Input images are first segmented into 1000 regions with a SLIC compactness of 15. The gradient and the color bag-of-words vectors dimensions are set to $G = 300$ and $C = 150$ (further increment of these dimensions did not lead to additional performance gains). The dataset is translated into a uniform representation of features, regions, and region connections. The input representation is thus equal for the different models for both datasets, allowing for an objective quantitative comparison. The results of the optimized models on the test set are shown in Tables 1 and 2 by means of F_1 scores, defined as $F_1 = \frac{2pr}{(p+r)}$ with p the precision and r the recall of a particular class, averaged over all of its corresponding regions.

These results show the improved overall accuracy of structured prediction over independent logistic regression, evidenced by a higher average $\bar{F_1}$ score. However, one may also observe a decrement in accuracy for some of the labels corresponding to smaller objects, e.g., poles in both datasets. Visual inspection shows that the SSVM merges these regions with their surroundings. This could be due to the variance in context in which these classes appear. The labels 'misc. text', 'pedestrian', and 'road shoulder' also suffer from a decreased F_1 score when using contextual relations. Upon visual inspection, both classes 'misc. text' and 'road shoulder' are hard to identify and appear in a plethora of different contexts, similar to poles. Pedestrians on the other hand are easy to identify by humans, but visual inspection also indicates that they are prone to being filtered out. The underlying reason could be that the SSVM is unable to learn how to lever context in an effective manner due to varying context. Using a second-degree connectivity structure over a default SSVM, and thus larger contextual cues, increases the classification accuracy of some classes substantially. Although some classes suffer from a marginal decrement in accuracy, the average $\bar{F_1}$ score still improves.

Fig. 3 shows the performance of our model on our own autonomous vehicle dataset. These qualitative results show the ability of the structured predictor to segment the dataset in a very fine-grained fashion. The hardest classes to detect are smaller objects, e.g., vehicles and road signs. The predictor is especially confused in case of road signs, as these are not surrounded by a fixed label class. Combined with the low occurrence frequency of the road sign class in the training data, the structured predictor is unable to leverage contextual information to undo the errors of the underlying logistic regression classifier. This is consistent with, and could explain, the lower detection F_1 scores of particular classes in Tables 1 and 2. Road marks on the other hand, also small in size, are better detected. An explanation is that road marks have similar appearances, and are consistently surrounded by road regions in our dataset, allowing this contextual information to be captured by the pairwise potentials.

Fig. 4 shows a qualitative comparison of independent classification by logistic regression (left images), and structured prediction (right images). In this figure, it can be noticed that the classification results of independent logistic regression is very noisy, leading to a high number of spurious predictions. Although the SSVM is based on the output of this logistic regression function, it is able to correct many of the underlying errors by making use of the first- and second-degree neighbor interactions. Moreover, the smoothing effect caused by these pairwise interaction potentials leads to a more visually consistent segmentation. These results demonstrate that the use of contextual information leads to higher labeling accuracy by filtering out erroneous labels.

6 Conclusion

Advanced perception systems that understand the environment are essential in enabling autonomous vehicles. We modeled a structured output machine learning model that takes into account visual contextual cues between over-segmentation regions within a second-order neighborhood. The structured model is formulated as an energy-based function using feature-dependent unary potentials, and pairwise potentials that differentiate between first- and second-degree region neighbors. After optimization by means of max-margin learning, most-probable label assignments are obtained via graph cuts inference using α-expansion. Quantitative and qualitative results both indicate that using second-degree contextual information allows for a higher labeling accuracy by better filtering out spurious labels.

7 Acknowledgment

We would like to thank Brecht Hanssens for his insightful comments. Rein Houthooft and Cedric De Boom are supported by a Ph.D. Fellowship of the Research Foundation - Flanders (FWO).

References

Achanta, R.; Shaji, A.; Smith, K.; Lucchi, A.; Fua, P.; and Susstrunk, S. 2012. SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Trans. Pattern Anal. Mach. Intell.* 34(11):2274–2282.

Armbrust, C.; Braun, T.; Föhst, T.; Proetzsch, M.; Renner, A.; Schäfer, B.-H.; and Berns, K. 2009. RAVON–The robust autonomous vehicle for off-road navigation. In *Proc. IARP Int. Workshop on Robotics for Risky Interventions and Environmental Surveillance*, 12–14.

Bernini, N.; Bertozzi, M.; Castangia, L.; Patander, M.; and Sabbatelli, M. 2014. Real-time obstacle detection using stereo vision for autonomous ground vehicles: A survey. In *Proc. IEEE Int. Conf. Intelligent Transportation Systems (ITSC)*, 873–878.

Bosch, A.; Muoz, X.; and Freixenet, J. 2007. Segmentation and description of natural outdoor scenes. *Image Vision Comput.* 25(5):727 – 740.

Boykov, Y.; Veksler, O.; and Zabih, R. 2001. Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Anal. Mach. Intell.* 23(11):1222–1239.

Brostow, G. J.; Fauqueur, J.; and Cipolla, R. 2009. Semantic object classes in video: A high-definition ground truth database. *Pattern Recognition Letters* 30(2):88–97.

Byun, J.; Na, K.-i.; Seo, B.-s.; and Roh, M. 2015. Drivable road detection with 3D point clouds based on the MRF for intelligent vehicle. In *Proc. Int. Conf. Field and Service Robotics*, 49–60.

Geiger, A.; Lauer, M.; Wojek, C.; Stiller, C.; and Urtasun, R. 2014. 3d traffic scene understanding from movable platforms. *IEEE Trans. Pattern Anal. Mach. Intell.* 36(5):1012–1025.

Hadsell, R.; Erkan, A.; Sermanet, P.; Ben, J.; Kavukcuoglu, K.; Muller, U.; and LeCun, Y. 2007. A multi-range vision strategy for autonomous offroad navigation. In *Proc. Robotics and Applications (RA)*, 1–7.

Hadsell, R.; Sermanet, P.; Ben, J.; Erkan, A.; Scoffier, M.; Kavukcuoglu, K.; Muller, U.; and LeCun, Y. 2009. Learning long-range vision for autonomous off-road driving. *J. Field Robot.* 26(2):120–144.

Joachims, T.; Finley, T.; and Yu, C.-N. J. 2009. Cutting-plane training of structural SVMs. *Mach. Learn.* 77(1):27–59.

Kelly, A.; Stentz, A.; Amidi, O.; Bode, M.; Bradley, D.; Diaz-Calderon, A.; Happold, M.; Herman, H.; Mandelbaum, R.; Pilarski, T.; et al. 2006. Toward reliable off road autonomous vehicles operating in challenging environments. *Int. J. Robot. Res.* 25(5-6):449–483.

Kundu, A.; Li, Y.; Dellaert, F.; Li, F.; and Rehg, J. 2014. Joint semantic segmentation and 3D reconstruction from monocular video. In *Proc. European Conf. Computer Vision (ECCV)*. 703–718.

Kuthirummal, S.; Das, A.; and Samarasekera, S. 2011. A graph traversal based algorithm for obstacle detection using lidar or stereo. In *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, 3874–3880.

Leonard, J.; How, J.; Teller, S.; Berger, M.; Campbell, S.; Fiore, G.; Fletcher, L.; Frazzoli, E.; Huang, A.; Karaman, S.; et al. 2008. A perception-driven autonomous urban vehicle. *J. Field Robot.* 25(10):727–774.

Lucchi, A.; Li, Y.; Smith, K.; and Fua, P. 2012. Structured image segmentation using kernelized features. In *Proc. European Conf. Computer Vision (ECCV)*. 400–413.

Milella, A.; Reina, G.; Underwood, J.; and Douillard, B. 2011. Combining radar and vision for self-supervised ground segmentation in outdoor environments. In *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, 255–260.

Milella, A.; Reina, G.; Underwood, J.; and Douillard, B. 2014. Visual ground segmentation by radar supervision. *Robot. Auton. Syst.* 62(5):696–706.

Nourani-Vatani, N.; Lpez-Sastre, R.; and Williams, S. 2015. Structured output prediction with hierarchical loss functions for seafloor imagery taxonomic categorization. In *Proc. Iberian Conference on Pattern Recognition and Image Analysis*. 173–183.

Nowozin, S., and Lampert, C. H. 2011. Structured learning and prediction in computer vision. *Found. Trends. Comput. Graph. Vis.* 6(3–4):185–365.

Reina, G.; Milella, A.; and Underwood, J. 2012. Self-learning classification of radar features for scene understanding. *Robot. Auton. Syst.* 60(11):1377–1388.

Ros, G.; Ramos, S.; Granados, M.; Bakhtiary, A.; Vazquez, D.; and Lopez, A. M. 2015. Vision-based offline-online perception paradigm for autonomous driving. In *Proc. IEEE Winter Conf. Applications Computer Vision (WACV)*, 231–238.

Sermanet, P.; Hadsell, R.; Scoffier, M.; Grimes, M.; Ben, J.; Erkan, A.; Crudele, C.; Miller, U.; and LeCun, Y. 2009. A multirange architecture for collision-free off-road robot navigation. *J. Field Robot.* 26(1):52–87.

Sivaraman, S., and Trivedi, M. 2013. Looking at vehicles on the road: A survey of vision-based vehicle detection, tracking, and behavior analysis. *IEEE Trans. Intell. Transp. Syst.* 14(4):1773–1795.

Tola, E.; Lepetit, V.; and Fua, P. 2010. DAISY: An efficient dense descriptor applied to wide baseline stereo. *IEEE Trans. Pattern Anal. Mach. Intell.* 32(5):815–830.