Semantic Grid Estimation with Occupancy Grids and Semantic Segmentation Networks

Özgür Erkent, Christian Wolf, Christian Laugier

▶ **To cite this version:**

Özgür Erkent, Christian Wolf, Christian Laugier. Semantic Grid Estimation with Occupancy Grids and Semantic Segmentation Networks. ICARCV 2018 - 15th International Conference on Control, Automation, Robotics and Vision, Nov 2018, Singapore, Singapore. pp.1-6. hal-01933939

HAL Id: hal-01933939
https://inria.hal.science/hal-01933939

Submitted on 24 Nov 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Semantic Grid Estimation with Occupancy Grids and Semantic Segmentation Networks

Özgür Erkent[1], Christian Wolf[1,2,3], Christian Laugier[1],

***Abstract*— We propose a method to estimate the semantic grid for an autonomous vehicle. The semantic grid is a 2D bird's eye view map where the grid cells contain semantic characteristics such as *road, car, pedestrian, signage, etc.* We obtain the semantic grid by fusing the semantic segmentation information and an occupancy grid computed by using a Bayesian filter technique. To compute the semantic information from a monocular RGB image, we integrate segmentation deep neural networks into our model. We use a deep neural network to learn the relation between the semantic information and the occupancy grid which can be trained end-to-end extending our previous work on semantic grids. Furthermore, we investigate the effect of using a conditional random field to refine the results. Finally, we test our method on two datasets and compare different architecture types for semantic segmentation. We perform the experiments on KITTI dataset and Inria-Chroma dataset.**

I. Introduction

One of the challenges in self-driving vehicles is to perceive the environment precisely and accurately. Recent developments in machine learning methods, in particular deep learning, improved the perception capabilities of these vehicles significantly. However, complexities in the surroundings, weather and illumination differences and dynamic nature of the scenes are still hard problems to be considered. Although the performances have increased significantly, they are still not sufficient. To overcome the difficulties imposed by these challenges, we integrate the outcomes of the deep learning methods with a well-established area, occupancy grids obtained with a Bayesian filtering method.

As the name suggests, the occupancy grid is composed of cells/grids representing the environment as a 2D bird's eye view map. Each cell contains the probability about its state [1], [2]. These states can represent any property of the cell; such as the probability of containing free space or a static object. Depending on the size of the cells, the occupancy maps can be dense. Another important property of occupancy grids is that they can be adapted to be used with any kind of sensor easily (e.g. stereo cameras [3] or laser range sensors [4]). Despite their advantages, one of the shortcomings of the occupancy grids is that detailed semantic information is not available.

To overcome this disadvantage of the occupancy grids, we use a method which integrate the semantic information of the scene that can be obtained from an RGB camera with the occupancy grid and estimate the *semantic grid*

[1] INRIA, Chroma Team, Rhône-Alpes, France. [2] Université de Lyon, INSA-Lyon, CNRS, LIRIS, F-69621, France. [3] CITI, INSA-Lyon, F-69621, France. Correspondence: ozgur.erkent@inria.fr

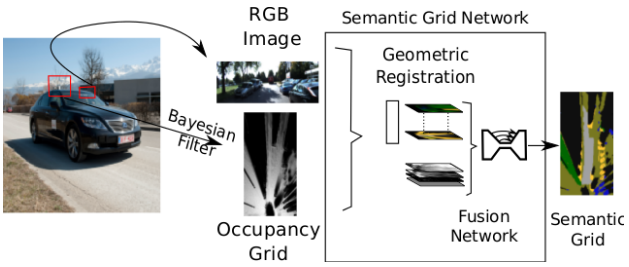


Fig. 1: An overview of the approach.

[5]. The semantic properties of the cells can contain classes *road, car, pedestrian, sidewalk, building, bicycle, etc.*. We propose a method that can be trained end-to-end with several semantic segmentation network architectures. Furthermore, we use conditional random fields (CRFs) [6] to refine the final details. Our network has three main parts: semantic segmentation of the scene, occupancy grid estimation, and integration of the occupancy grid with the semantic segmentation information.

It is possible to use any of the high performance segmentation networks for the semantic segmentation part. We compare three different semantic segmentation methods and evaluate their performances on two datasets, KITTI and Inria-Chroma dataset. The contributions of the proposed method can be listed as:

- A deep neural network which can be trained end-to-end to estimate the semantic grids by integrating the occupancy grids and a monocular RGB image of the scene.
- Refining the details of the semantic grid by applying a CRF.
- Comparison of three different semantic segmentation network architectures in the semantic grid estimation network.

II. Related Work

Semantic knowledge has been integrated into maps in a few previous studies. Tung et. al. [7] integrate the region proposal networks [8] to compute semantic segmentations. Babahajiani et. al. [9] removes the road surfaces and building facades from the point clouds and the remaining voxels are classified with the assumption of a strict restriction in the structure of the environment. All these approaches construct a 3D map of the surroundings which is generally computationally expensive. On contrary, some methods obtain a 2D representation of the environment with their semantic classes. Dequaire et. al. [10] use recurrent neural networks to obtain such a map; however, they don't consider to enrich

the LIDAR data with RGB information. Finally, Erkent et. al. [5] propose the semantic grids; however they do not consider the effects of using different network architectures in end-to-end training and do not integrate the architecture with CRFs, which is reported to increase the accuracy of the segmentation in recent methods (e.g. DeepLab v2 [11]).

Instead of directly constructing a map from sensor data, we use the output of Bayesian approaches which are capable of constructing an occupancy map efficiently. Occupancy maps are 2D bird's eye view maps of the environment providing information about the occupancy states of the cells which constitute the grid. Bayesian Occupany Filter (BOF) is an early successful example of such approaches [12]. Its success is a result of its capability to compute the occupancy and dynamic characteristics in parallel. Although it improved the computation time significantly, further studies provided a real-time computation power. Conditional Monte Carlo Dense Occupancy Tracker (CMCDOT) introduced four different states into the occupancy grid [1]. The introduction of four different states reduced the computation complexity significantly and provided a real-time speed. Therefore, we will use the dense occupancy grid provided by CMCDOT.

Although CMCDOT provides the occupancy states of the map, it is not capable of providing semantic characteristics of the cells. For this purpose, we will compute the semantic characteristics of the scene by using the RGB image of the environment. Long et. al. [13] is one of the pioneers at using deep neural networks for semantic segmentation. As a result of the cascaded layers to extract features, the resolution of the feature maps decreases. A solution to this problem is to use the encoder-decoder networks (e.g. SegNet [14] or U-Net [15]). Recent methods combine CRFs with segmentation networks ([11]). Another successful segmentation network is Grid Networks which segments as a flow through a grid. The resolution problem is solved via selecting a specific neural network according to the change of resolution during inference [16]. Although this method has a superior performance in segmentation accuracy, we do not consider this approach due to its computational complexity.

III. Semantic Grid Construction

We are going to integrate the output of an occupancy grid $\boldsymbol{o}$ with the 2D image $\boldsymbol{i}$ provided by an RGB camera. The final output will be the semantic grid $\boldsymbol{g}$ of the environment from a bird's eye view perspective.

Let $\boldsymbol{i}_{x,y}$ denote the value at position x and y, which corresponds to a pixel value for an image. The values of the semantically segmented grid $\boldsymbol{g}$ consist of different class values $\boldsymbol{c}$ where all the classes are elements of the alphabet Λ while the occupancy grid has a different probability value for each state.

The occupancy grid is constructed by using the LIDAR data $\boldsymbol{l}$ where $\boldsymbol{l}$ contains temporal and geometrical data. The Bayesian particle filter provides a dense map of the environment; however, the state classes are restricted to only four. For a more detailed semantic classification, we will fuse the occupancy grids with the output of the semantic segmentation part.

In the sub-sections, first we will explain the three semantic segmentation networks compared in this study, then occupancy grid estimation method, later the integration method of the occupancy grids with the output of the segmentation network part and finally the refinement method of the final output by using a CRF method.

A. Semantic Segmentation Methods for 2D Images

We are going to briefly explain three segmentation network architectures compared in this study. Each architecture takes an RGB input $\boldsymbol{i}$ and outputs the probability scores for each semantic class $\boldsymbol{s}$. These probabilities are fused with the occupancy grids to obtain the semantic grid estimations. In all of the architectures, we use the weights from a pre-trained model on a large scale dataset ImageNet/ILSVRC [17] and then fine-tune the weights by using our own dataset. We have selected the methods with a preferable runtime/accuracy trade-off.

SegNet [14] is an encoder-decoder network. The resolution degrades in the network due to down-sampling and pooling between successive layers. SegNet preserves the resolution of the image by applying upsampling. The encoder part has the architecture of VGG16 [18] with 13 layers. The decoder part also has 13 layers, with upsampling and unpooling with indices from the encoder layers. The soft-max is not applied and the outputs are fed to the fusion part of the network.

FCN [13] is a method where the early layers resemble the VGG16 [18] network and the pre-trained weights are taken from this network. Then a skip architecture is applied. This architecture integrates the semantic output of a deep coarse layer with a shallow one containing appearance information. Again, the soft-max is not applied and the outputs are fed to the fusion part of the network.

DeepLab v2 [11] uses convolution with upsampled filters called as "atrous convolution" to solve the resolution issue. It uses pre-trained parameters from ResNet [19]. Although CRF is used after the segmentation, we skip to use CRF.

B. Occupancy Grid Construction

The current sensor measurements and the previous grid cells are used to estimate the current state probabilities of the cells via using a Bayesian Filtering technique. The mathematical details of the problem can be found in [1]. We briefly explain the method to construct an occupancy grid by using CMCDOT approach.

In CMCDOT, each grid is associated with probabilities of four states: free, statically occupied, dynamically occupied or unknown. The value for the probability of free implies the probability of an area being empty; however, it should be noted that this does not mean that we can drive the vehicle towards that region. The value of statically occupied state for a cell implies an obstacle, while the value for a dynamically occupied cell indicates a moving object. An area with a high

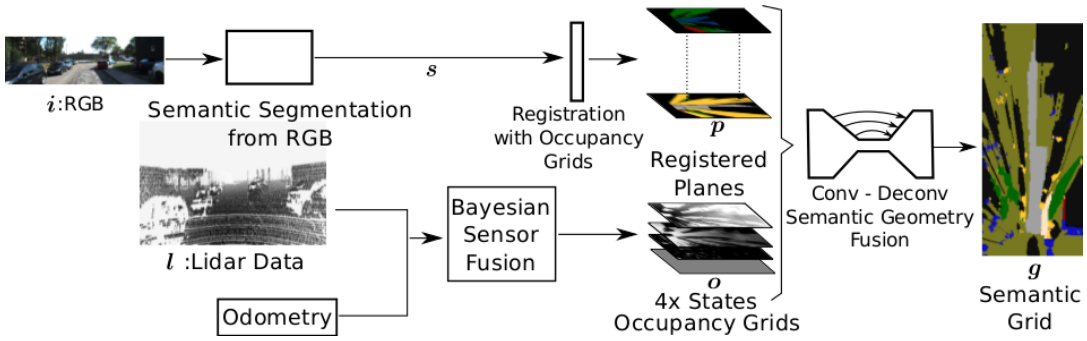


Fig. 2: Functional overview of the method. $\boldsymbol{i}$: RGB Image, $\boldsymbol{l}$: LIDAR data, $\boldsymbol{o}$: occupancy grids, $\boldsymbol{s}$: output of the segmentation network, $\boldsymbol{p}$: registered planes as inner representations, $\boldsymbol{g}$: semantic grid

value of unknown state shows that the corresponding area is not yet observed.

First, the current state probabilities are predicted by using the information from the previous states. A transition matrix is defined which transforms the previous states into the current state. Next, these updated state probabilities are evaluated by using an observation model based on the probabilistic sensor model [20]. Afterwards, state distributions are computed. Particle re-sampling is used to obtain new particles for new areas of the dynamically occupied regions. After this, the iteration continues again with the prediction step. The parameters of the occupancy grid construction are given in the Section IV.

C. Fusion of Occupancy and Semantic Information

Fusion of the occupancy grid $\boldsymbol{o}$ with the semantic information obtained from a semantic segmentation method applied on the RGB image $\boldsymbol{i}$ of the scene is non trivial unless precise segmentation is possible with the accompanying accurate depth information. Otherwise, any error resulting from the semantic segmentation approach would propagate to the semantic grid directly due to direct projection of semantic information via registration with depth information, also the sparsity of the depth data would result in a sparse semantic grid. Therefore we are assuming that depth information is not available during fusion avoiding computation of dense disparity maps from stereo and propose a method to learn the fusion method in an end-to-end manner without requiring to learn the semantic segmentation in RGB image plane.

It would be sub-optimal to learn the fusion process without any geometrical constraints, since the network would require huge amount of demonstrative data to learn a geometric transformation between the semantic view and the bird's eye view of the scene. Therefore, we provide this geometric transformation as an intermediate representation to the network. The output of the semantic segmentation part of the network provides the probability score $\boldsymbol{s}_{c,x,y}$ of each pixel belonging to a class $\boldsymbol{c}$ and $\boldsymbol{C}$ is the number of semantic classes. Therefore, we have $\boldsymbol{C}$ projective images with probability scores for each class at each pixel. We transform each semantic view $\boldsymbol{s}_c$ into intermediate representation planes $\boldsymbol{p}_{c,\delta_i}$ which are parallel to the plane of the occupancy grid $\boldsymbol{o}$ and $\boldsymbol{o}$ is generally perpendicular to the projective image (Fig. 3). $i \in \{1, ..., D\}$ and D is the number of total planes for class $\boldsymbol{c}$. δ_i is the distance of the i^{th} plane from occupancy grid $\emptyset$.

To overcome the ambiguity resulting from the lack of the depth information, we use a layered 3D map as a collection of $D \times \boldsymbol{C}$ planes $\{\boldsymbol{p}_{c,\delta_i}\}$ for $i \in \{1, ..., D\}$ and $c \in \{1, 2, ..., \boldsymbol{C}\}$. We assume a fixed distance between consecutive planes $d = ||\delta_i - \delta_{i-1}||$. The transformation between the projective image plane $\boldsymbol{i}$ and the registered planes is straight forward since we have access to the intrinsic and extrinsic calibration parameters of the camera and we know the parameters used to construct the occupancy grid. For a point $\{x_i^j, y_i^j\} \in \boldsymbol{p}_{c,\delta_i}$, the height is the distance from the occupancy grid $\boldsymbol{o}$, $z_i^j = \delta_i$. Therefore, the point can be defined as $\{x_i^j, y_i^j, z_i^j\}$ in occupancy grid $\boldsymbol{o}$ coordinates. First, we find the 3D coordinates of point j in image coordinates given the transformation ${}^{\boldsymbol{o}}_{\boldsymbol{i}}tf$ from occupancy grid coordinates to projective image coordinates $\left(\hat{x}_i^j, \hat{y}_i^j, \hat{z}_i^j, 1\right)^\intercal = {}^{\boldsymbol{o}}_{\boldsymbol{i}}tf \left(x_i^j, y_i^j, z_i^j, 1\right)^\intercal$. Then, the corresponding pixel in image plane can be found as

$$\begin{pmatrix} \bar{x}_i^j \\ \bar{y}_i^j \\ 1 \end{pmatrix} = K \begin{pmatrix} \hat{x}_i^j / \hat{z}_i^j \\ \hat{y}_i^j / \hat{z}_i^j \\ 1 \end{pmatrix} \tag{1}$$

where K is the intrinsic camera calibration matrix, $(\bar{x}_i^j, \bar{y}_i^j)$ is the image coordinate of the j^{th} point in the i^{th} plane $p_{.,\delta_i}$. Each point in each plane $\boldsymbol{p}_{c,\delta_i,x_i^j,y_i^j}$ has a corresponding value in the output of the semantic network $\boldsymbol{s}_{c,\bar{x}_i^j,\bar{y}_i^j}$. This transformation can be rewritten with a sampling kernel applied to have the value of a particular pixel for the cell $\boldsymbol{p}_{c,\delta_i,x_i^j,y_i^j}$:

$$\boldsymbol{p}_{c,\delta_i,x_i^j,y_i^j} = \sum_{n=1}^{\bar{H}} \sum_{m=1}^{\bar{W}} \boldsymbol{s}_{c,n,m} \boldsymbol{k}(x_i^j - m; \Phi_x) \boldsymbol{k}(y_i^j - n; \Phi_y) \tag{2}$$

$\forall i \in \{0, ..., D\}, c \in \{1, ..., \boldsymbol{C}\}, j \in \{1, ..., HW\}$ where (H, W) is the height and width of the occupancy grid and $(\bar{H}, \bar{W})$ is the height and width of $\boldsymbol{s}$. Any sampling kernel $k(.)$ can be used, we use a bilinear sampling kernel which is shown to be differentiable [21]. Therefore, we can

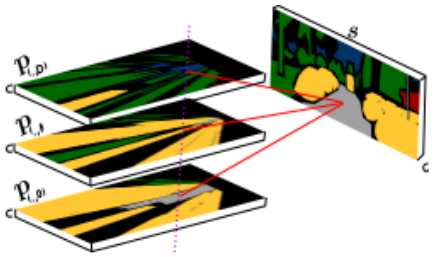


Fig. 3: A sample transformation of the projective semantic information into the planes that are parallel to the occupancy grid is shown. C represents the number of classes. The class probability values are colored for illustration purposes.

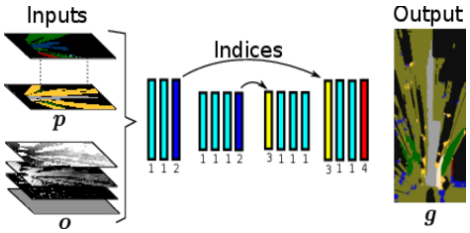


Fig. 4: The fusion network architecture. The number of layers are reduced with respect to a standard VGG16 network. The labels for the layers are as follows: 1: Convolution + Batch Normalization + ReLU, 2: Max-Pooling, 3: Upsampling, 4: Softmax

train our model end-to-end by using a bilinear sampling kernel for transforming the outputs of semantic network into the registered planes parallel to the occupancy grid. If the distance between any two registered planes δ_i is small enough, and the visible point is inside the limits of the occupancy grid, then at least one of the planes will contain the correct label of the corresponding cell.

For the semantic geometry fusion part of the network, we use the $D \times \boldsymbol{C}$ planes from the output of the semantic segmentation part and the four states of the occupancy grid as inputs to the fusion network which is a convolution-deconvolution network. Each layer performs a convolution to extract features, and then they are batch-normalized and an element-wise rectified-linear non-linearity (ReLU) is applied. The downsampling of the images is performed via max-pooling after which the indices of the maximum values are provided to the decoder part of the network [22]. The decoder part also has the same number of layers with the encoder. Each layer performs a convolution, batch-normalization and ReLU. The indices stored during encoding process is used to upsample the features. Finally, multi-class softmax is applied to classify the cells at the end of the operations. Loss is chosen to be cross-entropy [13] and it is equal to the sum all the errors in all of the cells. However, due to restrictions of the memory as we are already using a segmentation network in the first part of the model, we reduce the number of layers both in encoding and decoding layers. An illustration of the network is given in Fig. 4.

D. Refinement with Conditional Random Fields

One of the known issues of segmentation with deep neural networks is that deep models reduce resolution. One of the solutions to overcome this problem is to apply CRFs on the outcome of the deep neural networks [11]. CRFs have generally been used to smooth segmentations [23] due to their capability of coupling neighbor nodes which result in labeling the spatially close pixels with the same classes. Therefore, we are also encouraged to test the effect of CRFs in our model by using the fully connected CRF model proposed in [6]. The model utilizes the following energy function:

$$E(x) = \sum_i \phi_u(x_i) + \sum_{i<j} \phi_p(x_i, x_j) \tag{3}$$

$i, j \in \{1, ..., HW\}$. x_i are the label classes for the grids. The first part $\phi_u(x_i)$ is the unary potential obtained by a classifier which contains the distribution over label assignment x_i. In our case, it is the classification output of our network. The second part is called as the pairwise potentials and it can be computed in two parts as follows:

$$\phi_p(x_i, x_j) = \mu(x_i, x_j)[w_1 \exp(-\frac{|c_i - c_j|^2}{2\sigma_\alpha^2} - \frac{|o_i - o_j|^2}{2\sigma_\beta^2}) + w_2 \exp(-\frac{|c_i - c_j|^2}{2\sigma_\gamma^2})] \tag{4}$$

The first part is the bilateral kernel which is a Gaussian appearance kernel that depends on both the cell positions and the values of the grid. The second Gaussian kernel depends on locations only. $\sigma_\alpha, \sigma_\beta, \sigma_\gamma$ are scale parameters. Bilateral kernel enforces the close pixels with similar labels to have similar labels while the second one considers spatial relationship only. $\mu(x_i, x_j) = [x_i \neq x_j]$ is the simple label compatibility function given by Potts model and penalizes the grids with different labels. Higher-order potentials which can reason about more complex relations between the grid cells are not considered due their high computational complexity.

IV. Experiments

We use two different datasets which provide semantically segmented images acquired via sensors placed on a vehicle. One of the datasets don't contain labels for the bird's eye view; therefore, we transform the labels to the bird's eye view by using the segmented images from the frontal RGB images.

First, we use the **the KITTI Dataset** [24]. Zhang et. al. [25] provide a semantically segmented version of the dataset taken from the vehicle. The images are labeled at every 10 frames. The dataset contains 10 classes, 142 images for training and 110 images for evaluation. LIDAR data, camera calibration parameters and vehicle motion information is also provided. The segmented frontal camera view is the main groundtruth for this dataset. We obtain the groundtruth for the top view by using these segmented images. The

point clouds obtained with LIDAR are converted into RGB camera coordinates. Then, the pixels are registered with a depth value. For the pixels with more than one possible depth value, the smaller one is preferred. The registered points are back-projected onto the occupancy grid coordinates. However, since the pointcloud is sparse the registration results in a sparse representation. Another problem is the imperfections in the labeling process which results in false registrations. Some morphological techniques are applied to the images under the supervision of a human to reduce the errors in the registration process and obtain a dense semantic grid ground truth. For the regions where no labels are available, the grid state probabilities are compared. The empty cell gets the corresponding state label with the highest probability, i.e. free, statically occupied, dynamically occupied or unknown.

We also use the **the INRIA-Chroma Semantic Grid Dataset**. It contains 657 image with LIDAR information, 276 of which are labeled from 5 different route sequences. The labeling is made both in RGB view and the bird's eye view. LIDAR, camera calibrations and vehicle motion information are available. We use 146 images from 3 routes for training and 130 images from 2 routes for evaluation. Since the labels are available in the bird's eye view, we don't process data to obtain the groundtruth. The state labels of the occupancy grid are not added to this dataset.

We use CMCDOT [1] for the occupancy grid construction. We set the width of the area where the grid is constructed to be 31 m, the length to be 71 m and the grid size to be 0.2×0.2 m.

We compare the performance of three different network architectures for semantic segmentation of RGB images as discussed in Section. III-A, SegNet [14] , FCN [13] and DeepLab v2 [11]. For all of the three architectures that we compare, we use the parameters from a pre-trained network ILSVRC/Imagenet [17] for classification initially if the corresponding layers are available and later finetune it on our data, separately for both datasets. We use a learning rate of 1×10^{-3} and momentum of 0.9, a mini-batch size of 6, therefore it takes approximately 23 epochs for a complete pass over all training data. We train the data with 2000 iterations. Number of planes for intermediate representation is selected to be $D = 20$ and number of classes is $C = 14$.

The number of class labels are not balanced. This may result in a poor learning of less frequent classes. To overcome this problem, we use *median frequency balancing* [26]. The frequency of a class $f(c)$ is the ratio of the number of cells to the number of all cells in the semantic grid if the class is available in the grid. Each class has weight $\alpha_c = \frac{m_f}{f(c)}$ in the cross entropy loss function where m_f is the median of the frequencies. We use class-balancing in all of the evaluations. As discussed previously, we perform an end-to-end training, therefore we don't perform any evaluation on the output of the semantic segmentation of RGB images. We also analyze the effect of CRF refinement on output of the model (Table. I).

Results and Discussion — We discuss the results for different types of architecture types and with the existence of CRF refinement techniques for both of the datasets.

TABLE I: Results with CRF on the KITTI dataset with different architecture types.

Architecture Type	CRF	Pixel Acc.	Class Acc.	FmIoU
SegNet [14]		81.1	**49.4**	**69.8**
SegNet [14]	X	**81.2**	45.6	69.6
FCN [13]		79.8	47.5	68.6
FCN [13]	X	80.0	43.7	68.2
Deeplab v2 [11]		76.2	36.9	63.1
Deeplab v2 [11]	X	76.9	34.6	63.3

TABLE II: Results with CRF for INRIA-Chroma-Semantic Grid Dataset

Architecture Type	CRF	Pixel Acc.	Class Acc.	FmIoU
SegNet		78.6	**35.3**	65.2
SegNet	X	**78.9**	33.3	**65.4**

We use the following commonly used measures for comparison, pixel accuracy is the ratio of the correctly classified grid cells, class accuracy is the mean of the ratio of accuracy for each class and the mean of intersection over union based on frequency (**FmIoU**) which is also called as the frequency weighted average Jaccard index. The results are evaluated on the segmented grids only.

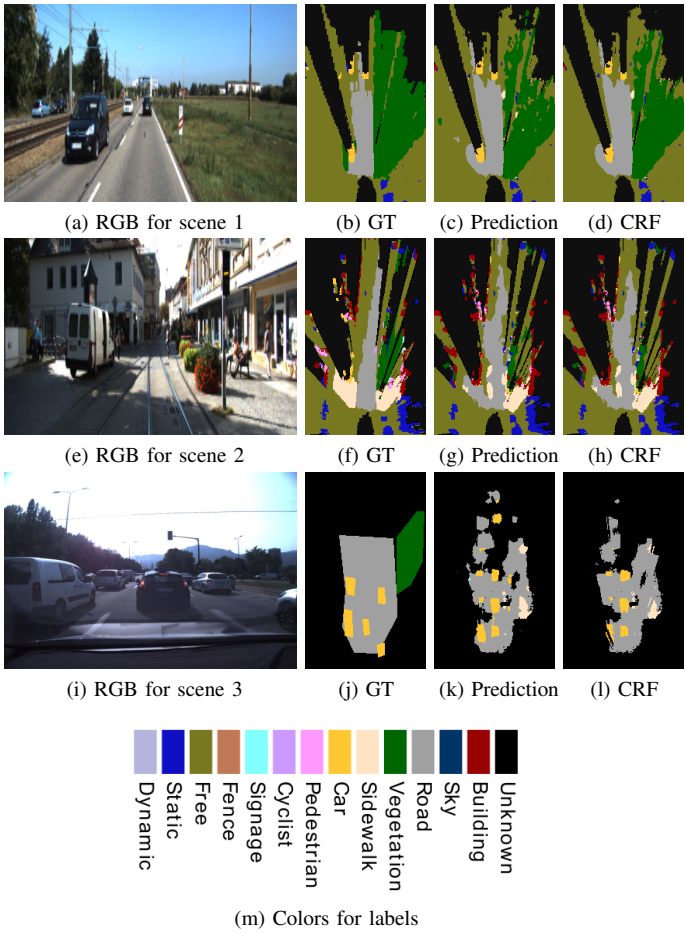
First, we analyze the different architecture types (Table. I). SegNet [14] performs the best with respect to all of the other architecture types. One of the reasons may be that it has less parameters than others; therefore, it can be trained with less number of training samples in a small number of iterations. A densely labeled data with higher number of samples can result in a different comparison.

Another point to analyze is the CRF refinement. The improvement after the CRF addition is not significant. There is a slight increase in the pixel accuracy; on the other hand, the class accuracy is decreasing. The reason may be related to the size of the semantic classes in the bird's eye view. Some of the classes may be viewed very small from the top view, which may be removed by CRF which results in a performance degrade on class accuracy. However, the holes may also be filled up, which increases the performance in pixel accuracy slightly.

We also test our model in Inria-Chroma dataset (Table. II). The results are similar to the ones obtained in KITTI dataset. Therefore, we can make similar conclusions. Addition of CRF slightly increases overall pixel accuracy, but decreases the class accuracy probably by removing some classes with small sizes. To use higher order potentials for CRF can improve the performance since it will learn the possible relations between classes, therefore it will also learn that some classes will have smaller sizes.

A few illustrations of the results are shown in Fig.5. **(a)**, **(e)**, **(i)** are RGB images of the scene as seen from the frontal camera of the vehicle. **(b)**, **(f)**, **(j)** are the ground truths for the semantic grid. **(c)**, **(g)**, **(k)** and **(o)** are the prediction outcomes from our model. Finally, **(d)**, **(h)**, **(l)** are the semantic grids after CRF refinement. We can observe some interesting behaviors of our model. For example, the

distinction between the road and the sidewalk can be made which is not possible in an occupancy grid. The vegetation can also be detected in some of the grids correctly. In scene 3, some of the grids containing pedestrians are detected correctly. The difference between a prediction and a semantic grid after CRF refinement is not perceived easily. For example, in **(h)** it can be observed that some of the false detections introduced by the neural network in **(g)** at the sides of the way is removed after the CRF refinement step.



(a) RGB for scene 1 (b) GT (c) Prediction (d) CRF

(e) RGB for scene 2 (f) GT (g) Prediction (h) CRF

(i) RGB for scene 3 (j) GT (k) Prediction (l) CRF

(m) Colors for labels

Fig. 5: Three scenes with RGB image, ground truth (GT), semantic segmentation predictions and results of the CRF refinement. Scene 1 and 2 are from KITTI dataset, whereas Scene 3 is from Inria-Chroma dataset.

V. Conclusion

We have proposed a method which is capable of fusing the Bayesian particle filter with the output of the neural networks that are used to semantically segment the scene. We have included a CRF process at the end of our model to analyze the effect of CRF refinement on the performance. Furthermore, we showed that the model can be trained end-to-end. We evaluated our method on two datasets which had similar results to each other.

Acknowledgement

This work was supported by Toyota Motor Europe. We thank Jean-Alix David and Jérôme Lussereau for their assistance with data collection.

References

[1] L. Rummelhard, A. Nègre, and C. Laugier, "Conditional monte carlo dense occupancy tracker," in *ITSC*. IEEE, 2015, pp. 2485–2490.
[2] H. P. Moravec, "Sensor fusion in certainty grids for mobile robots," *AI magazine*, vol. 9, no. 2, p. 61, 1988.
[3] M. Perrollaz, J.-D. Yoder, A. Spalanzani, and C. Laugier, "Using the disparity space to compute occupancy grids from stereo-vision," in *IROS*. IEEE, 2010, pp. 2721–2726.
[4] J. D. Adarve, M. Perrollaz, A. Makris, and C. Laugier, "Computing occupancy grids from multiple sensors using linear opinion pools," in *ICRA*. IEEE, 2012, pp. 4074–4079.
[5] O. Erkent, C. Wolf, C. Laugier, D. Gonzalez, and V. Cano, "Semantic grid estimation with a hybrid bayesian and deep neural network approach," in *IEEE IROS*, 2018, pp. 1–8.
[6] P. Krahenbuhl and V. Koltun, "Efficient inference in fully connected crfs with gaussian edge potentials," in *NIPS*, 2011.
[7] F. Tung and J. J. Little, "MF3D: Model-free 3D semantic scene parsing," in *ICRA*, 2017, pp. 4596–4603.
[8] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.
[9] P. Babahajiani, L. Fan, J. K. Kämäräinen, and M. Gabbouj, "Urban 3D segmentation and modelling from street view images and LiDAR point clouds," *Machine Vision and Applications*, vol. 28, no. 7, pp. 1–16, 2017.
[10] J. Dequaire, P. Ondrúška, D. Rao, D. Wang, and I. Posner, "Deep tracking in the wild: End-to-end tracking using recurrent neural networks," *The International Journal of Robotics Research*, p. 0278364917710543, 2017.
[11] L. C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE PAMI*, vol. PP, no. 99, pp. 1–1, 2017.
[12] C. Coué, C. Pradalier, C. Laugier, T. Fraichard, and P. Bessière, "Bayesian occupancy filtering for multitarget tracking: an automotive application," *The International Journal of Robotics Research*, vol. 25, no. 1, pp. 19–30, 2006.
[13] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *CVPR*, 2015.
[14] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE PAMI*, vol. 39, no. 12, pp. 2481–2495, 2017.
[15] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *MICCAI*, 2015.
[16] D. Fourure, R. Emonet, E. Fromont, D. Muselet, A. Tremeau, and C. Wolf, "Residual Conv-Deconv Grid Network for Semantic Segmentation," in *BMVC*, 2017.
[17] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, *et al.*, "Imagenet large scale visual recognition challenge," *IJCV*, vol. 115, no. 3, pp. 211–252, 2015.
[18] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.
[19] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE CVPR*, 2016, pp. 770–778.
[20] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*. MIT press, 2005.
[21] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, "Spatial transformer networks," in *NIPS*, 2015, pp. 2017–2025.
[22] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *ICCV*, 2015, pp. 1520–1528.
[23] C. Rother, V. Kolmogorov, and A. Blake, "Grabcut: Interactive foreground extraction using iterated graph cuts," in *ACM transactions on graphics (TOG)*, vol. 23, no. 3. ACM, 2004, pp. 309–314.
[24] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *CVPR*, 2012.
[25] R. Zhang, S. A. Candra, K. Vetter, and A. Zakhor, "Sensor Fusion for Semantic Segmentation of Urban Scenes," in *ICRA*, 2015, pp. 1850–1857.
[26] D. Eigen and R. Fergus, "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture," in *ICCV*, 2015, pp. 2650–2658.