Learning to Look around Objects for Top-View Representations of Outdoor Scenes

Samuel Schulter[1†], Menghua Zhai[2†], Nathan Jacobs[2], and Manmohan Chandraker[1,3]

[1] NEC-Laboratories, Cupertino CA 95014, USA
[2] University of Kentucky, Lexington KY 40506, USA
[3] University of California San Diego, La Jolla CA 92093, USA

Abstract. Given a single RGB image of a complex outdoor road scene in the perspective view, we address the novel problem of estimating an occlusion-reasoned semantic scene layout in the top-view. This challenging problem not only requires an accurate understanding of both the 3D geometry and the semantics of the visible scene, but also of occluded areas. We propose a convolutional neural network that learns to predict occluded portions of the scene layout by looking around foreground objects like cars or pedestrians. But instead of hallucinating RGB values, we show that directly predicting the semantics and depths in the occluded areas enables a better transformation into the top-view. We further show that this initial top-view representation can be significantly enhanced by learning priors and rules about typical road layouts from simulated or, if available, map data. Crucially, training our model does not require costly or subjective human annotations for occluded areas or the top-view, but rather uses readily available annotations for standard semantic segmentation in the perspective view. We extensively evaluate and analyze our approach on the KITTI and Cityscapes data sets.

Keywords: 3D scene understanding · Occlusion reasoning · Semantic top-view representations

1 Introduction

Visual completion is a crucial ability for an intelligent agent to navigate and interact with the three-dimensional (3D) world. Several tasks such as driving in urban scenes, or a robot grasping objects on a cluttered desk, require innate reasoning about unseen regions. A top-view or bird's eye view (BEV) representation[4] of the scene where occlusion relationships have been resolved is useful in such situations [11]. It is a compact description of agents and scene elements with semantically and geometrically consistent relationships, which is intuitive for human visualization and precise for autonomous decisions.

[4] We use the terms "top-view" and "bird's eye view" interchangeably.
† indicates equal contribution

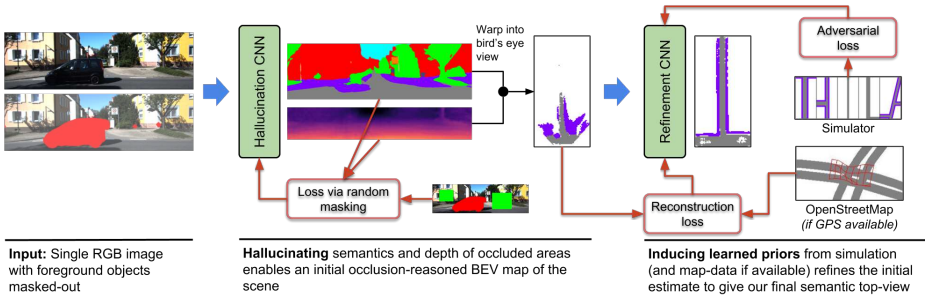


Fig. 1. Given a single RGB image of a typical street scene (left), our approach creates an **occlusion-reasoned semantic map of the scene layout in the bird's eye view**. We present a CNN that can hallucinate depth and semantics in areas occluded by foreground objects (marked in red and obtained via standard semantic segmentation), which gives an initial but noisy and incomplete estimate of the scene layout (middle). To fill in unobserved areas in the top-view, we further propose a refinement-CNN that induces learning strong priors from simulated and OpenStreetMap data (right), which comes at no additional annotation costs.

In this work, we derive such top-view representations through a novel framework that simultaneously reasons about geometry and semantics from just *a single RGB image*, which we illustrate in the particularly challenging scenario of outdoor road scenes. The focus of this work lies in the estimation of the scene layout, although foreground objects can be placed on top using existing 3D localization methods [24, 38]. Our learning-based approach estimates a geometrically and semantically consistent spatial layout even in regions hidden behind foreground objects, like cars or pedestrians, without requiring human annotation for occluded pixels or the top-view itself. Note that human supervision for such occlusion-reasoned top-view maps is likely to be subjective and of course, expensive to procure. Instead, we derive supervisory signals from readily available annotations for semantic segmentation in the perspective view, a depth sensor or stereo (for visible areas) and a knowledge corpus of typical road scenes via simulations and OpenStreetMap data. Fig. 1 provides an illustration.

Specifically, in Sec. 3.1, we propose a novel CNN that takes as input an image with occluded regions (corresponding to foreground objects) masked out, and estimates the segmentation labels and depth values over the entire image, essentially *hallucinating distances and semantics in the occluded regions*. In contrast to standard image in-painting approaches, we operate in the semantic and depth spaces rather than the RGB image space. Sec. 3.1 shows how to train this CNN without additional human annotations for occluded regions. The hallucinated depth map is then used to map the hallucinated semantic segmentation of each pixel into the bird's eye view, see Sec. 3.2.

This initial prediction can be incomplete and erroneous, for instance, since BEV pixels far away from the camera can be unobserved due to limited image resolution or due to imperfect depth estimation. Thus, Sec. 3.3 proposes a refine-

ment and completion neural network to leverage easily obtained training data from *simulations that encode general priors and rules* about road scene layouts. Since there is no correspondence between actual images and simulated data, we employ an *adversarial loss* for teaching our CNN a generative aspect about typical layouts. When GPS is available for training images, we also show how *map data provides an additional training signal* for our models. We demonstrate this using OpenStreetMap (OSM) [19]. Maps provide rough correspondence with RGB images through the GPS location, but it can be noisy and lacks information on scene scale, besides mislabels in the map itself. We handle these issues by *learning a warping function* that aligns OSM data with image evidence using a variant of spatial transformer network [13]. Note that a single RGB image is used at test time, with simulations or OSM limited to training.

In Sec. 4, we evaluate our proposed semantic BEV synthesis on the KITTI [8] and Cityscapes [4] datasets. For a quantitative evaluation, we manually annotate validation images with the scene layout in both the perspective and the top-view, which is a time-consuming and error-prone process but again highlights the benefit of our method that resorts only to readily available annotations. Since, to the best of our knowledge, no prior work exists solving this problem in a similar setup to allow a fair comparison, we comprehensively evaluate with several baselines to study the role of each module. Our experiments consider roads and sidewalks for layout estimation, with cars and persons as occluding foreground objects, although extensions to other semantic classes are straightforward in future work. While not our focus, we visualize a simple application in Sec. 4.3 to include foreground objects such as cars and pedestrians in our representation. We observe qualitatively meaningful top-view estimates, which also obtain low errors on our annotated test set.

2 Related Work

General scene understanding is one of the fundamental goals of computer vision and many approaches exist that tackle this problem from different directions.

Indoor: Recent works like [2, 16, 26] have shown great progress by leveraging strong priors about indoor environments obtained from large-scale data sets. While these approaches can rely on strong assumptions like a Manhattan world layout, our work focuses on less constrained outdoor driving scenarios.

Outdoor: Scene understanding for outdoor scenes has received a lot of interest in recent years [5, 10, 27, 30, 37], especially due to applications like driver assistance systems or autonomous driving. Wang et al. [29] propose a conditional random field that infers 3D object locations, semantic segmentation as well as a depth reconstruction of the scene from a single geo-tagged image, which also enables the use of OSM data. At test time, their approach requires as input accurate GPS and map information. In contrast, we require only the RGB image at test time. Seff and Xiao [21] leverage OpenStreetMap (OSM) data to predict several road layout attributes from a single image, like the distance to an intersection, drivable directions, heading angle, etc. While we also leverage OSM

for training our models and make predictions only from a single RGB image, we infer a full semantic map in the top view instead of a discrete set of attributes.

Top-view representations: Sengupta et al. [22] derive a top-view representation by relating semantic segmentation in perspective images to a ground plane with a homography. However, this is a simplifying (flat-world) assumption where non-flat objects will produce artifacts in the ground plane, like shadows or cones. To alleviate these artifacts, they aggregate semantics over multiple frames. However, removing all artifacts would require viewing objects from many different angles. In contrast, our approach enables reasoning about occlusion from just a single image, which is enabled by automatically learned and context-dependent priors about the world. Geiger et al. [7] represent road scenes with a complex model in the bird's eye view. However, input to the model comes from multiple sources (vehicle tracklets, vanishing points, scene flow, etc.) and inference requires MCMC, while our approach efficiently computes the BEV representation from just a single image. Moreover, their hand-crafted parametric model might not account for all possible scene layouts, whereas our approach is non-parametric and thus more flexible. Máttyus et al. [18] combine perspective and top-view images to estimate road layouts and Zhai et al. [34] predict the semantic layouts of top-view images by learning the transformation between the perspective and the top-view. Gupta et al. [11] demonstrate the suitability of a BEV representation for mapping and planning, even though it is not explicitly learned.

Occlusion reasoning: Most recent works in this area focus on occlusions of foreground objects and use complex hand-crafted models [5, 30, 37, 38]. In contrast, we estimate the layout of a scene occluded by foreground objects. Guo and Hoiem [10] employ a scene parsing approach that retrieves existing shapes from training data based on visible pixels. Our approach learns to hallucinate occluded areas and does not rely on an existing and fixed set of polygons from training data. Liu et al. [17] also hallucinate the semantics and depth of regions occluded by foreground objects. However, (i) their approach relies on a hand-crafted graphical model while ours is learning-based and (ii) they assume sparse depth from a laser scanner as input, while we estimate depth from a single RGB image (the sparse depth maps are actually ground truth for training our models).

3 Generating bird's eye view representations

We now present our approach for transforming a single RGB image in the perspective view into an occlusion-reasoned semantic representation in the bird's eye view, see Fig. 1. We take as input an image $I \in \mathbb{R}^{h \times w \times 3}$ with spatial dimension h and w and a semantic segmentation $S^{\text{fg}} \in \mathbb{R}^{h \times w \times C}$ of the *visible* scene, where C is the number of categories. Note that any semantic segmentation method can be used and we rely on the recently proposed pyramid scene parsing (PSP) network [35]. S^{fg} provides the location of foreground objects that occlude the scene. In this work, we consider foreground objects like cars or pedestrians as occluders but other definitions are possible as well.

To reason about these occlusions, we define a masked image I^M, where pixels of foreground objects have been removed. In Sec. 3.1, we propose a CNN that takes I^M as input and hallucinates the depth as well as the semantics of the entire image, including occluded pixels. The occlusion-reasoned depth map D^{bg} allows us to map the occlusion-reasoned semantic segmentation S^{bg} into 3D and then into the bird's eye view (BEV), see Sec. 3.2.

While this initial BEV map B^{init} is already better than mapping the non-occlusion reasoned semantic map S^{fg} into 3D, there can still be unobserved or erroneous pixels. In Sec. 3.3, we thus propose a CNN that learns priors from simulated data to further improve our representation. If a GPS signal is available, OpenStreetMap (OSM) data can be additionally included as supervisory signal.

3.1 Learning to see around foreground objects

An important step towards an occlusion-reasoned representation of the scene is to infer the semantics and the geometry behind foreground objects.

Masking: Given the semantic segmentation S^{fg}, we define the mask of foreground pixels as $M \in \mathbb{R}^{h \times w}$, where a pixel in the mask M_{ij} is 1 if and only if the segmentation at that pixel S^{fg}_{ij} belongs to any of the foreground classes. Otherwise, the pixel in the mask is 0. In order to inform the CNN about which pixels have to be in-painted, we apply the mask on the input RGB image and define each pixel in the masked input I^M as

$$I^M_{ij} = \begin{cases} \bar{m}, & \text{if } M_{ij} = 1 \\ I_{ij}, & \text{otherwise,} \end{cases}$$

where $\bar{m}$ is the mean RGB value of the color range, such that after normalization the input to the CNN is zero for those pixels. Given I^M, we extract a feature representation by applying ResNet-50 [12]. Similar to recent semantic segmentation literature [35], we use a larger stride in convolutions and dilation [32] to increase the feature map resolution from $\frac{h}{32} \times \frac{w}{32}$ to $\frac{h}{8} \times \frac{w}{8}$.

In addition to masking the input image, we explicitly provide the mask as input to the CNN for two reasons: (i) While the value m becomes 0 after centering the input of the CNN, other visible pixels might still share the same value and confuse the training of the CNN. (ii) An explicit mask input allows encoding more information like the category of the occluded pixel. We thus define another mask $M^{\text{cls}} \in \mathbb{R}^{h \times w \times C^{\text{fg}}}$, where C^{fg} is the number of foreground classes and each channel corresponds to one of them. We encode M^{cls} with a small CNN and fuse the resulting feature with the one from the masked image, see Fig. 2a.

Hallucination: We then put two decoders on the fused feature representation of I^M and M^{cls} for predicting semantic segmentation and the depthmap of the occlusion-free scene. For semantic segmentation, we again use the PSP module [35], which is particularly useful for in-painting where contextual information is crucial. For depth prediction, we follow [15] in defining the network architecture. Both decoders are followed by a bilinear upsampling layer to provide the

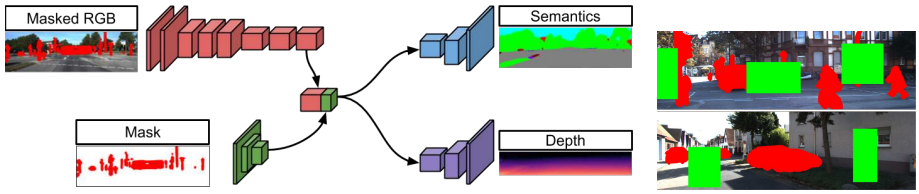


Fig. 2. (a) The *inpainting CNN* first encodes a masked image and the mask itself. The extracted features are concatenated and two decoders predict semantics and depth for visible and occluded pixels. **(b)** To train the inpainting CNN we ignore foreground objects as no ground truth is available (red) but we *artificially add masks (green)* over background regions where full annotation is already available

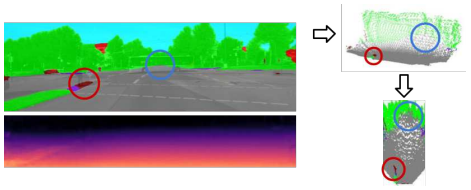


Fig. 3. The process of mapping the semantic segmentation with corresponding depth first into a 3D point cloud and then into the bird's eye view. The red and blue circles illustrate corresponding locations in all views.

output at the same resolution as the input, see Fig. 2a. While traditional in-painting methods fill missing pixels with RGB values, note that we directly go from an RGB image to the in-painted semantics and the geometry of the scene, which has two benefits: (1) The computational costs are smaller as we avoid the (in our case) unnecessary detour in the RGB space. (2) The task of in-painting in the RGB space is presumably harder than in-painting semantics and depth as there is no need for predicting any texture or color.

Training: We train the proposed CNN in a supervised way. However, as mentioned before, it would be very costly to annotate the semantics and particularly the geometry behind foreground objects. We thus resort to an alternative that only requires standard semantic segmentation and depth ground truth. Because our desired ground truth is unknown for real foreground objects in the masked input image I^M, we do not infer any loss at those pixels. However, we augment I^M with additional randomly sampled masks, but for which we still have ground truth, see Fig. 2b. In this way, we can teach our CNN to hallucinate occluded areas of the input image without acquiring costly human annotations. Note that an alternative to masking regions in the input image is to paste real foreground objects into the scene. However, this strategy requires separate instances of foreground objects cropped at the semantic boundaries and a good understanding of the scene geometry for generating a realistic looking training image.

3.2 Mapping into the bird's eye view

Given the depth map D^{bg} and the intrinsic camera parameters $\mathbf{K}$, we can map each coordinate of the perspective view into the 3D space. We drop the z-

coordinate (height axis) for each 3D point and assign x and y coordinates to the closest integer, which gives us a mapping into bird's eye view representation. We use this mapping to transfer the class probability distribution of each pixel in the perspective view, i.e., S^{bg}, into the bird's eye view, which we denote $B^{\text{init}} \in \mathbb{R}^{k \times l \times C^{\text{bg}}}$, where C^{bg} is the number of background classes and k and l are the spatial dimensions. Throughout the paper, we use $k = 128$ and $l = 64$ pixels that we relate to 60×30 meters in the point cloud. For all points that are mapped to the same pixel in the top view, we average the corresponding class distribution. Fig. 3 illustrates the geometric transformation.

Note that B^{init} is our first occlusion-reasoned semantic representation in the bird's eye view. However, B^{init} also has several remaining issues. Some pixels in B^{init} will not be assigned any class probability, especially those far from the camera due to image foreshortening in the perspective view. Imperfect depth prediction is also an issue because it may assign a well classified pixel in the perspective view a wrong depth value, which puts the point into a wrong location in top-view. This can lead to unnatural arrangements of semantic classes in B^{init}.

3.3 Refinement with a Knowledge Corpus

To remedy the above mentioned issues, we propose a refinement CNN that takes B^{init} and predicts the final output $B^{\text{final}} \in \mathbb{R}^{k \times l \times C^{\text{bg}}}$, which has the same dimensions as B^{init}. The refinement CNN has an encoder-decoder structure with a fully-connected bottleneck layer, see Fig. 4b. The main difficulty in training the refinement CNN is the lack of semantic ground truth data in the bird's eye view, which is very hard and costly to annotate. In the following we present two sources of supervisory signals that are easy to acquire.

Simulation: The first source of information we leverage is a simulator that renders the semantics of typical road scenes in the bird's eye view. The simulator models roads with different types of intersections, lanes and sidewalks, see Fig. 4a for some examples. Note that it is easy to create such a simulator as we do not need to model texture, occlusions or any perspective distortions in the scene. A simple generative model about road topology, number of lanes, radius for curved roads, etc. is enough. Since there is no correspondence with the real training data, we rely on an adversarial loss [1] between predictions of the refinement CNN B^{final} and data from the simulator B^{sim}

$$\mathcal{L}^{\text{sim}} = \sum_{i=1}^{m} d\left(B_i^{\text{final}}; \Theta_{\text{discr}}\right) - \sum_{i=1}^{m} d\left(B_i^{\text{sim}}; \Theta_{\text{discr}}\right) \ ,$$

where m is the batch size and $d\left(.; \Theta_{\text{discr}}\right)$ is the discriminator function with parameters Θ_{discr}. Note that $d\left(.; \Theta_{\text{discr}}\right)$ needs to be a K-Lipschitz function [1], which is enforced by gradient clipping on the parameters Θ_{discr} during training. While any other variant of adversarial loss is possible, we found [1] to provide the most stable training. The adversarial loss injects prior information about typical road scene layouts and remedies errors of B^{init} like unobserved pixels or unnatural shapes of objects due to depth or semantic prediction errors.

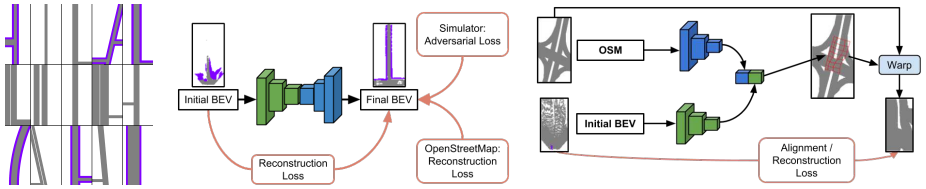


Fig. 4. (a) Simulated road shapes in the top-view. **(b) The refinement-CNN** is an encoder-decoder network receiving three supervisory signals: self-reconstruction with the input, adversarial loss from simulated data, and reconstruction loss with aligned OpenStreetMap (OSM) data. **(c) The alignment CNN** takes as input the initial BEV map and a crop of OSM data (via noisy GPS and yaw estimate given). The CNN predicts a warp for the OSM map and is trained to minimize the reconstruction loss with the initial BEV map.

Since $\mathcal{L}^{\text{sim}}$ operates without any correspondence, the refinement network needs additional regularization to not deviate too much from the actual input, i.e., B^{init}. We add a reconstruction loss between B^{init} and B^{final} to define the final loss as $\mathcal{L} = \mathcal{L}^{\text{sim}} + \lambda \cdot \mathcal{L}^{\text{reconst}}$ with

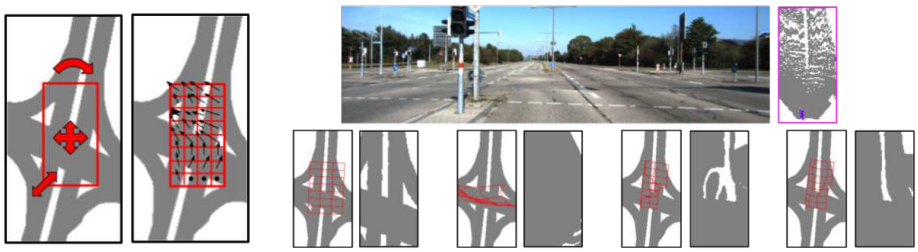
$$\mathcal{L}^{\text{reconst}} = \frac{\|(B^{\text{init}} - B^{\text{final}}) \odot \mathbf{M}\|^2}{\sum_{ij} \mathbf{M}} ,$$

where $\odot$ is an element-wise multiplication and $\mathbf{M} \in \mathbb{R}^{k \times l}$ is a mask of 0's for unobserved pixels in B^{init} and 1's otherwise.

OpenStreetMap data: Driving imagery often comes with a GPS signal and an estimate of the driving direction, which enables the use of OpenStreetMap (OSM) data as another source of supervisory signal for the refinement CNN. The most simple approach is to render the OSM data for the given location and angle, B^{osm}, and define a reconstruction loss with B^{final} as $\mathcal{L}^{\text{OSM}} = \|B^{\text{final}} - B^{\text{osm}}\|^2$. This loss can be included into the final loss $\mathcal{L}$ in addition to or instead of $\mathcal{L}^{\text{reconst}}$. In any case, $\mathcal{L}^{\text{OSM}}$ ignores noise in the GPS and the direction estimate as well as imperfect renderings due to annotation noise and missing information in OSM.

We therefore propose to align the initial OSM map B^{osm} with the semantics and geometry observed in the actual RGB image with a warping function $\hat{B}^{\text{osm}} = w\,(B^{\text{osm}}; \theta)$ parameterized by θ. We use a composition of a similarity transformation implemented as a parametric spatial transformer (handling translation, rotation, and scale; denoted "Box") and a non-parametric warp implemented as bilinear sampling (handling non-linear misalignments due to OSM rendering; denoted "Flow") [13], see Fig. 5. We minimize the masked reconstruction between $\hat{B}^{\text{osm}}$ and the initial BEV map B^{init},

$$\theta^* = \arg\min_{\theta} \frac{\|(B^{\text{init}} - w\,(B^{\text{osm}}; \theta)) \odot \mathbf{M}\|^2}{\sum_{ij} \mathbf{M}} + \lambda_2 \Gamma(w\,(B^{\text{osm}}; \theta)) + \lambda_3 \|\theta\|_2^2 ,$$



(a) (b)

Fig. 5. (a) We use a composition of similarity transform (left, "box") and a non-parametric warp (right, "flow") to align noisy OSM with image evidence. **(b, top)** Input image and the corresponding B^{init}. **(b, bottom)** Resulting warping grid overlaid on the OSM map and the warping result for 4 different warping functions, respectively: "box", "flow", "box+flow", "box+flow (with regularization)". Note the importance of composing the transformations and the induced regularization.

where $w(.;\theta)$ is differentiable [13], and $\Gamma(.)$ is a low-pass filter similar to [36, 28], and $\|.\|_2^2$ the squared ℓ_2-norm, both acting as regularizing functions. The hyper-parameters λ_2 and λ_3 are manually set.

To minimize the alignment error the first choice is non-linear optimization, e.g., LBFG-S [3]. However, we found this to produce satisfactory results only for parts of the data, while a significant portion would require hand-tuning of several hyper-parameters. This is mostly due to noise in the initial BEV map B^{init} as well as the rendering B^{osm}. An alternative, which proved to be more stable and easy to realize, is to learn a function that predicts the warping parameters, which has the benefit that the predictive function can implicitly leverage other examples of $(B^{\text{init}}, B^{\text{osm}})$ pairs in the training corpus. We thus train a CNN that takes B^{init} and B^{osm} as inputs and predicts the warping parameters θ by minimizing the alignment error. Also, we can either train this CNN separately or jointly with the refinement CNN, thus providing different training signals for the refinement module. We evaluate these options in Sec. 4.2. Fig. 4c illustrates the process of aligning the OSM data.

4 Experiments

Our quantitative and qualitative evaluation focuses on occlusion reasoning via hallucination in the perspective view (Sec. 4.1) and scene completion via the refinement network in the bird's eye view (Sec. 4.2).

Datasets: Creating the proposed BEV representation requires data for learning the parameters of the modules described above. Importantly, the only supervisory signal that we need is semantic segmentation (human annotation) and depth (LiDAR or stereo), although not both are required for the same input image. Both KITTI [8] and Cityscapes [4] fulfill our requirements. Both data

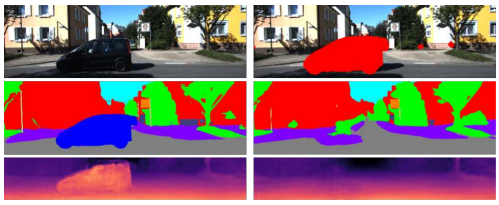


Fig. 6. Qualitative example of our hallucination CNN: Semantics and depth without (left) and with (right) hallucination.

Table 1. Hallucination results for two general in-painting strategies and different mask encodings

Method	random-boxes hidden		human-gt visible	human-gt hidden
	IoU	ARD	IoU	IoU
RGB-inpaint	**68.83**	.1428	79.25	55.79
Direct	64.63	**.1413**	**81.12**	**60.06**
RGB-only	63.07	.1440	79.71	60.77
+ mask	63.47	**.1435**	80.14	60.24
+ cls-encode	**64.79**	.1453	**80.63**	**61.06**

sets come with a GPS signal and a yaw estimate of the driving direction, which allows us to additionally leverage OSM data during training.

The KITTI [8] data set contains many sequences of typical driving scenarios and contains accurate GPS location and driving direction as well as a 3D point cloud from a laser scanner. However, annotation for semantic segmentation is scarce. We create two versions of the data set based on segmentation annotation: *KITTI-Ros* consists of 31 sequences (14201 frames) for training, where 100 of them have semantic annotation, and of 9 sequences (4368) for validation, where 46 images are annotated for segmentation. The segmentation ground truth comes from [20]. *KITTI-RAW* consists of 31 sequences (16273 frames) for training and 9 sequences (2296 frames) for validation. 1074 images from the training set and 233 images from the validation set have ground truth annotations for semantic segmentation, which we collected on our own.

The *Cityscapes* data set [4] contains 2975 training and 500 validation images, all of which are fully annotated for semantic segmentation and are provided as stereo image pairs. For ease of implementation, we rely on a strong stereo method [33] to serve as our training signal for depth, although unsupervised methods exists for direct training from stereo images [6, 9]. GPS location and heading are also provided, although accuracy is lower compared to KITTI.

Validation data for occlusion-reasoning: For a quantitative evaluation of occlusion reasoning in the perspective view as well as in the bird's eye view, we manually annotated all validation image of the three data sets that also have semantic segmentation ground truth. We asked annotators to draw the scene layout by hand for the categories "road" and "sidewalk". Other pixels are annotated as "background".

Implementation details: We train our in-painting models with a batch size of 2 for 80k iterations with ADAM [14]. The initial learning rate is 0.0002, which is decreased by a factor of 10 for the last 20k iterations. The refinement network is trained with a batch size of 64 for 80k iterations and a learning rate of 0.0001.

4.1 Occlusion Reasoning by Hallucination

Here we analyze our hallucination CNN proposed in Sec. 3.1, which targets at in-painting the semantics and depth of areas occluded by foreground objects.

To the best of our knowledge, there is no prior art that can serve as a *fair* comparison point. Although [17] addresses the same task, their approach assumes sparse depth information as input, which serves as ground truth in our approach. Nevertheless, we have created fair baselines that justify our design choices.

Evaluation protocol: We split our evaluation protocol into two parts. First, we follow [17] by randomly masking out background regions in the input and evaluate the predictions of the hallucination CNNs (random-boxes). For this case, note that evaluation can be done for all semantic classes and depth. While this is the only possible evaluation without human annotation for occluded areas, the sampling process may not resemble objects realistically. Thus, we also evaluate with our newly acquired annotations (human-gt) for the categories "road" and "sidewalk", which was not done in [17]. We measure mean IoU for segmentation and absolute relative distance (ARD) for depth estimation as in [15].

Semantics & depth space versus RGB space: We compare our hallucination CNN with a baseline that takes the traditional approach of in-painting and operates in the RGB pixel space. This baseline consists of two CNNs, one for in-painting in the RGB space and one for semantic and depth prediction. For a fair comparison, we equip both CNNs with the same ResNet-50 feature extractor. For RGB-space in-painting, we use the same decoder structure as for depth prediction but with 3 output channels and train it with the random mask sampling strategy. The second CNN has the exact same architecture as our hallucination CNN and is trained without masking inputs but instead uses the already in-painted RGB images. From Table 1 we can see that the proposed direct hallucination network outperforms in-painting in the RGB space for depth prediction and segmentation with the human-provided ground truth while it trails for segmentation of all categories with random boxes. The reason for the inferior performance might be missing context information that is available to the baseline by the RGB-space supervision. However, note that the proposed architecture is twice as efficient, since in-painting and prediction of semantics and depth are obtained in the same forward pass. Qualitative examples of our direct hallucination CNN are given in Fig. 6.

Mask-encoding: We also analyze different variants of how to encode the foreground mask as input to the proposed hallucination CNN. Table 1 demonstrates the beneficial impact of explicitly encoding the foreground mask ("+mask") in addition to masking the RGB image ("RGB-only"), as well as providing the class information of the foreground objects inside the mask ("+cls-encode").

4.2 Refining the BEV representation

We now evaluate the refinement model described in Sec. 3.3 on all three data sets with the acquired annotations in the bird's eye view. The evaluation metric again is mean IOU for the categories "road" and "sidewalk". We compare four models: (1) The initial BEV map, without refinement. (2) A refinement heuristic, where missing semantic information at pixel (i, j) is filled with the

Table 2. (a) Results on the KITTI-RAW data set showing the impact of the refinement module with simulated and OSM data compared to B^{init} and a simple refinement heuristic. We also show the impact of hallucination and depth prediction. **(b)** Results for KITTI-Ros and Cityscapes

(a)

Setting (KITTI-RAW)	Road	Sidewalk	Mean
BEV-init	58.13	29.33	43.73
Refine-heuristic	67.93	30.12	49.02
Simulation	66.98	29.73	48.36
Simulation+OSM	**68.89**	**30.35**	**49.62**
no halluc.	51.85	24.76	38.31
no halluc. (refine)	65.67	25.91	45.79
no depth pred.	44.54	8.61	26.58
no depth pred. (refine)	46.11	7.73	26.92

(b)

Dataset	Setting	Road	Sidewalk	Mean
KITTI-Ros	BEV-init	56.93	40.71	48.82
	Refine-heuristic	69.59	41.31	55.45
	Simulation	62.96	43.19	53.08
	Simulation+OSM	**71.82**	**44.77**	**58.29**
Cityscapes	BEV-init	51.40	17.47	34.43
	Refine-heuristic	52.06	17.22	34.64
	Simulation	52.89	17.89	35.39
	Simulation+OSM	**56.46**	**19.60**	**38.03**

Experiment	Setting	Road	Sidewalk	Mean
Warping-method	Box	64.77	30.51	47.64
	Flow	66.03	**30.74**	48.39
	Box+Flow	**68.89**	30.35	**49.62**
Warp-optimization	LBFGS	22.31	29.24	25.78
	CNN	63.91	29.19	46.55
	CNN-joint	**68.89**	**30.35**	**49.62**

Table 3. An ablation study of the proposed BEV-refinement module. We analyze different types of warping functions and OSM alignment optimization strategies

semantics of the closest pixels in y-direction towards the camera. (3) The proposed refinement module with simulated data and the self-reconstruction loss. (4) The refinement module with the additional OSM-reconstruction loss. Table 2 clearly shows that the combination of simulated and aligned OSM data provides the best supervisory signal for the refinement module on all three data sets. Interestingly, the refinement heuristic is a strong competitor but this is probably because evaluation is limited to only "road" and "sidewalk", where simple rules are often correct. This heuristic will likely fail for classes like "vegetation" and "building". Importantly, all refinement strategies improve upon the initial BEV map. Because no fair comparison point to prior art is available to us, we further analyze two alternative baselines on the KITTI-RAW data set.

Importance of hallucination: We train a refinement module that takes as input BEV maps that omit the hallucination step ("No halluc."). To create this BEV map, we train a joint segmentation and depth prediction network (same architecture as for hallucination) with standard foreground annotation and map the semantics of background classes into the BEV map as described in Sec. 3.2. Table 2 shows that avoiding the hallucination step hurts the performance. Note that the proposed refinement CNN recovers most errors for roads, while the relative performance drop for sidewalks is larger. We believe this is due to long stretches of non-occluded roads in the KITTI data set. Sidewalks, on the other hand, are typically more occluded due to parked cars and pedestrians.

Importance of depth prediction: We train a CNN that takes as input the RGB image in the perspective view and directly predicts the BEV map, without

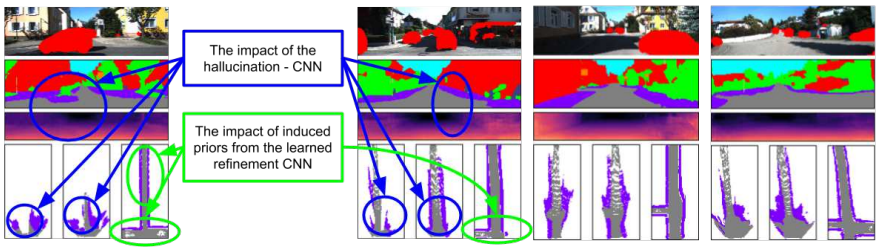


Fig. 7. Examples of our BEV representation. Each one shows the masked RGB input, the hallucinated semantics and depth, as well as three BEV maps, which are (from left to right), The BEV map without hallucination, with hallucination, and after refinement.

depth prediction ("No depth pred."). The CNN extracts basic features with ResNet-50 [12], applies strided convolutions for further down-sampling, a fully-connected layer resembling a transformation from 2D to 3D, and transposed convolutions for up-sampling into the BEV dimensions. To create a training signal for this network, we map ground truth segmentation with the ground truth depth data (LiDAR) into the bird's eye view. On top of the output of this CNN, we still apply the proposed refinement module for a fair comparison. The importance of depth prediction becomes clearly evident from Table 2. In this case, not even the refinement-CNN is able to recover. While there can be better architectures for directly predicting a semantic BEV map from the perspective view than our baseline, it is important to note that depth is an intermediary that clearly eases the task by enabling the use of known geometric transformations.

Warping OSM data: In Table 3, we compare different warping functions and optimization strategies for aligning the OSM data, as described in Sec. 3.3. Our results show that the composition of "Box" (translation, scale and rotation) and "Flow" (displacement field) is superior to individual warps. We can also see that the proposed alignment CNN trained jointly with the refinement module provides the best training signal from OSM data. As already mentioned in Sec. 3.3, LBFG-S alignment failed for around 30% of the training data, which explains the superiority of the proposed CNN for predicting warping parameters.

Qualitative results: Fig. 7 demonstrates the beneficial impact of both the hallucination and refinement modules with several qualitative examples. In the first three cases, we can observe the learned priors of the hallucination CNN that correctly handles largely occluded areas, which is evident from both the hallucinated semantics and the difference in the first two illustrated BEV maps (before and after hallucination). Other examples illustrate how the refinement CNN completes unobserved areas and even completes whole side roads and intersections.

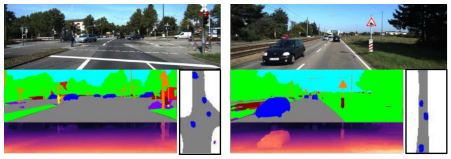


Fig. 8. Two examples of a BEV map including foreground objects, like cars here. For each example, we also shows the input image, the semantic segmentation and the predicted depth map.

4.3 Incorporating Foreground Objects into the BEV map

Finally, we show how foreground objects like cars or pedestrians can be handled in the proposed framework. Since it is not the main focus of this paper, we use a simple baseline to lift 2D bounding boxes of cars into the BEV map. Importantly, we demonstrate that our refinement module is able to handle foreground objects as well. First we leverage the 3D ground truth annotations of the KITTI data set and estimate the mean dimensions of a 3D bounding box. Then, for a given 2D bounding box in the perspective view, we use the estimated depth map to compute the 3D point of the bottom center of the bounding box, which is then used to translate our prior 3D bounding box in the BEV map. The refinement network takes the initial BEV map that now includes foreground objects. We extend the simulator to render objects as rectangles in the top-view and employ a self-reconstruction loss since OSM cannot provide such information. Fig. 8 gives two examples of the obtained BEV-map with foreground objects for illustrative purpose. A full quantitative evaluation for localization accuracy and consistency with background requires significant extensions to be studied in our future work.

5 Conclusion

Our work addresses a complex problem in 3D scene understanding, namely, occlusion-reasoned semantic representation of outdoor scenes in the top-view, using just a single RGB image in the perspective view. This requires solving the canonical challenge of hallucinating semantics and geometry in areas occluded by foreground objects, for which we propose a CNN trained using only standard annotations in the perspective image. Further, we show that adversarial and warping-based refinement allow leveraging simulation and map data as valuable supervisory signals to learn prior knowledge. Quantitative and qualitative evaluations on the KITTI and Cityscapes datasets show attractive results compared to several baselines. While we have shown the feasibility of solving this problem using a single image, incorporating temporal information might be a promising extension for further gains. We finally note that with the use of indoor data sets like [23, 25], along with simulators [31] and floor plans [16], a similar framework may be derived for indoor scenes, which will be the subject of our future work.

Acknowledgement: This material is based upon work supported by the National Science Foundation under Grant No. (IIS-1553116). The work was part of M. Zhai's internship at NEC Labs America, in Cupertino.

References

1. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein Generative Adversarial Networks. In: ICML (2017)
2. Armeni, I., Sener, O., Zamir, A.R., Jiang, H., Brilakis, I., Fischer, M., Savarese, S.: 3D Semantic Parsing of Large-Scale Indoor Spaces. In: CVPR (2016)
3. Byrd, R.H., Lu, P., Nocedal, J., Zhu, C.: A limited memory algorithm for bound constrained optimization. SIAM Journal on Scientific Computing **16**(5), 1190–1208 (1995)
4. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The Cityscapes Dataset for Semantic Urban Scene Understanding. In: CVPR (2016)
5. Dhiman, V., Tran, Q.H., Corso, J.J., Chandraker, M.: A Continuous Occlusion Model for Road Scene Understanding. In: CVPR (2016)
6. Garg, R., G, V.K.B., Carneiro, G., Reid, I.: Unsupervised CNN for Single View Depth Estimation: Geometry to the Rescue. In: ECCV (2016)
7. Geiger, A., Lauer, M., Wojek, C., Stiller, C., Urtasun, R.: 3D Traffic Scene Understanding from Movable Platforms. PAMI (2014)
8. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets Robotics: The KITTI Dataset. International Journal of Robotics Research (IJRR) (2013)
9. Godard, C., Aodha, O.M., Brostow, G.J.: Unsupervised Monocular Depth Estimation with Left-Right Consistency. In: CVPR (2017)
10. Guo, R., Hoiem, D.: Beyond the line of sight: labeling the underlying surfaces. In: ECCV (2012)
11. Gupta, S., Davidson, J., Levine, S., Sukthankar, R., Malik, J.: Cognitive Mapping and Planning for Visual Navigation. In: CVPR (2017)
12. He, K., Zhang, X., Ren, S., Sun, J.: Deep Residual Learning for Image Recognition. In: CVPR (2016)
13. Jaderberg, M., Simonyan, K., Zisserman, A., Kavukcuoglu, K.: Spatial Transformer Networks. In: NIPS (2015)
14. Kingma, D.P., Ba, J.: Adam: A Method for Stochastic Optimization. In: ICLR (2015)
15. Laina, I., Christian Rupprecht, V.B., Tombari, F., Navab, N.: Deeper Depth Prediction with Fully Convolutional Residual Networks. In: 3DV (2016)
16. Liu, C., Schwing, A.G., Kundu, K., Urtasun, R., Fidler, S.: Rent3D: Floor-Plan Priors for Monocular Layout Estimation. In: CVPR (2015)
17. Liu, M., He, X., Salzmann, M.: Building Scene Models by Completing and Hallucinating Depth and Semantics. In: ECCV (2016)
18. Máttyus, G., Wang, S., Fidler, S., Urtasun, R.: HD Maps: Fine-grained Road Segmentation by Parsing Ground and Aerial Images. In: CVPR (2016)
19. OpenStreetMap contributors: Planet dump retrieved from https://planet.osm.org . https://www.openstreetmap.org (2017)
20. Ros, G., Ramos, S., Granados, M., Bakhtiary, A., Vazquez, D., Lopez, A.M.: Vision-based Offline-Online Perception Paradigm for Autonomous Driving. In: WACV (2015)
21. Seff, A., Xiao, J.: Learning from Maps: Visual Common Sense for Autonomous Driving. arXiv:1611.08583 (2016), https://arxiv.org/abs/1611.08583
22. Sengupta, S., Sturgess, P., Ladický, L., Torr, P.H.S.: Automatic Dense Visual Semantic Mapping from Street-Level Imagery. In: IROS (2012)

23. Silberman, N., Hoiem, D., Kohli, P., Fergus, R.: Indoor Segmentation and Support Inference from RGBD Images. In: ECCV (2012)
24. Song, S., Chandraker, M.: Robust Scale Estimation in Real-Time Monocular SFM for Autonomous Driving. In: CVPR (2014)
25. Song, S., Lichtenberg, S.P., Xiao, J.: SUN RGB-D: A RGB-D Scene Understanding Benchmark Suite. In: CVPR (2015)
26. Song, S., Yu, F., Zeng, A., Chang, A.X., Savva, M., Funkhouser, T.: Semantic Scene Completion from a Single Depth Image. In: CVPR (2017)
27. Sturgess, P., Alahari, K., Ladický, L., Torr, P.H.S.: Combining Appearance and Structure from Motion Features for Road Scene Understanding. In: BMVC (2009)
28. Vijayanarasimhan, S., Ricco, S., Schmid, C., Sukthankar, R., Fragkiadaki, K.: Sfm-net: Learning of structure and motion from video. CoRR **abs/1704.07804** (2017)
29. Wang, S., Fidler, S., Urtasun, R.: Holistic 3D Scene Understanding from a Single Geo-tagged Image. In: CVPR (2015)
30. Wojek, C., Walk, S., Roth, S., Schindler, K., Schiele, B.: Monocular Visual Scene Understanding: Understanding Multi-Object Traffic Scenes. PAMI **36**, 882–897 (2013)
31. Wu, Y., Wu, Y., Gkioxari, G., Tian, Y.: Building Generalizable Agents With a Realistic And Rich 3D Environment. CoRR **abs/1801.02209** (2018)
32. Yu, F., Koltun, V.: Multi-Scale Context Aggregation by Dilated Convolutions. In: ICLR (2016)
33. Zbontar, J., LeCun, Y.: Stereo matching by training a convolutional neural network to compare image patches. JMLR **17**, 1–32 (2016)
34. Zhai, M., Bessinger, Z., Workman, S., Jacobs, N.: Predicting Ground-Level Scene Layout from Aerial Imagery. In: CVPR (2017)
35. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid Scene Parsing Network. In: CVPR (2017)
36. Zhou, T., Brown, M., Snavely, N., Lowe, D.G.: Unsupervised learning of depth and ego-motion from video. In: CVPR (2017)
37. Zia, M.Z., Stark, M., Schindler, K.: Explicit Occlusion Modeling for 3D Object Class Representations. In: CVPR (2013)
38. Zia, M.Z., Stark, M., Schindler, K.: Towards Scene Understanding with Detailed 3D Object Representations (2015)