Short-Term Prediction and Multi-Camera Fusion on Semantic Grids

Lukas Hoyer
Bosch Center for Artificial Intelligence
lukas.hoyer@outlook.com

Patrick Kesper
Bosch Center for Artificial Intelligence
patrick.kesper@bosch.com

Anna Khoreva
Bosch Center for Artificial Intelligence
anna.khoreva@bosch.com

Volker Fischer
Bosch Center for Artificial Intelligence
volker.fischer@bosch.com

Abstract

An environment representation (ER) is a substantial part of every autonomous system. It introduces a common interface between perception and other system components, such as decision making, and allows downstream algorithms to deal with abstract data without knowledge of the used sensor. In this work, we propose and evaluate a novel architecture that generates an egocentric, grid-based, predictive, and semantically-interpretable ER, which we call semantic grid. We show that our approach supports the spatio-temporal fusion of multiple camera sequences and short-term prediction in such an ER. Our design utilizes a strong semantic segmentation network together with depth and egomotion estimates to first extract semantic information from multiple camera streams and then transform these separately into egocentric temporally-aligned bird's-eye view grids. A deep encoder-decoder network is trained to fuse a stack of these grids into a unified semantic grid and to predict the dynamics of its surrounding. We evaluate this representation on real-world sequences of Cityscapes and show that our architecture can make accurate predictions in complex sensor fusion scenarios and significantly outperforms a model-driven baseline in a category-based evaluation.

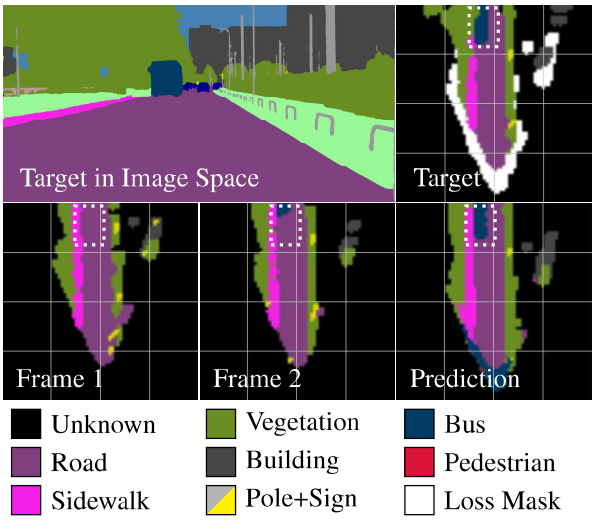

Figure 1. (best viewed in color) Semantic grid prediction: Our framework is based on semantic segmentations in image space (top left), which are transformed into the proposed bird's eye view *semantic grid* representation (top right). We feed our architecture with a sequence of input frames (bottom left and bottom center) to predict the subsequent frame (bottom right), which is compared with the target frame (top right). In the shown example, a bus (dark blue) is entering the grid at the top (bottom row). Our architecture is able to predict its movement further into the grid even though the vehicle became just visible in frame 2. For details on the loss mask and prediction artifacts, see Section 5.

1. Introduction

In recent years, deep learning methods have been investigated to control autonomous systems, such as self-driving cars or robots. An important property of such systems is their capability to perceive complex situations using multiple sensors and to act accordingly in a fast and reliable way. To enable intelligent decision making, a common environment representation (ER) as interface between different sensors and the downstream control has to be provided.

Such an ER should have certain properties. First, it should unify different sensor representations to support sensor fusion. In that way, downstream algorithms can work with this abstract representation without knowledge of the used sensors. Second, the ER should be interpretable to modularize the system and enhance human accessibility. So, the system can be debugged and understood more easily, improving its reliability. Third, the representation should be adaptable to the dynamically changing environment (e.g. not restricted to the predefined number of objects in the scene). And fourth, it should be predictive to compensate for system-inherent latencies caused by sensor mea-

surements or signal processing, which is particularly important in the context of recent computationally expensive computer vision algorithms.

Due to these reasons, we present and evaluate a proof of concept for generating an egocentric, interpretable, predictive, and grid-based representation of the changing environment, most importantly including the spatio-temporal fusion of semantic information of multiple cameras and short-term prediction. We call this ER *semantic grid*. It is a bird's eye view 2D projection of semantic features of the environment (see Fig. 1 top row).

Our concrete architecture for producing semantic grids can be disentangled into two major parts (see I and II in Fig. 2): First, semantic information is extracted from each camera signal using deep neural networks for semantic segmentation (see $\mathcal{S}$ in Fig. 2). The semantic information is then spatially transformed into a top-down, bird's eye view semantic grid, using depth information provided by the stereo camera (see $\mathcal{P}$ in Fig 2). Grids from different cameras and past time frames are temporally aligned to a certain future time τ using the agent's egomotion (see $\mathcal{T}$ in Fig. 2). Note that independently moving objects are not taken into account, as there is no motion model of other objects than the agent itself, yet.

Second, these spatio-temporally aligned representations are combined into a single grid using a deep encoder-decoder (ED) neural network. This network contains all trainable parameters of our architecture and has the non-trivial task to fuse the grids from different cameras and past times, making assumptions about the environment, as well as predicting the motion of potentially multiple dynamic objects. Such a prediction is shown in Fig. 1. Even though in this work we focus on multi-camera fusion, our architecture can be extended to other modalities such as LiDAR given an appropriate algorithm for extracting spatial semantic features from the sensor.

We have evaluated our architecture with respect to different semantic categories on the camera-based Cityscapes dataset of real-world driving scenarios. Our approach significantly outperforms solely model-driven baselines for single camera and multi-camera prediction, considering missing egomotion information, different sequence length, and varying prediction horizons. To the best of our knowledge, we are the first to investigate *joint* short-term prediction and multi-camera fusion on *semantic grids* using deep convolutional neural network. Our approach enables autonomous systems to perceive the environment with multiple cameras and helps to mitigate the problem of long runtimes of recent computer vision algorithms.

2. Related Work

Due to the recent discovery of certain cells dedicated to spatial referencing in brains [1], grid-based representations

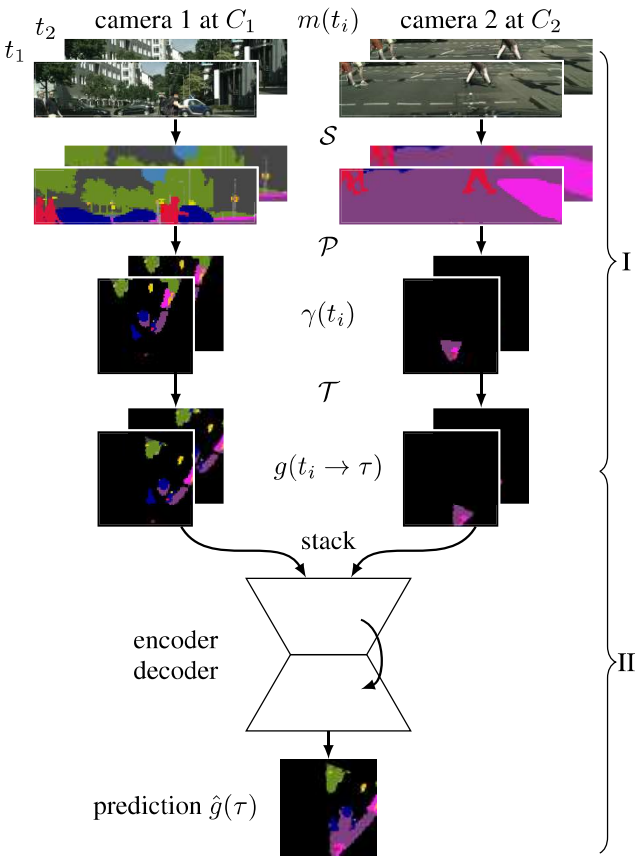

Figure 2. (best viewed in color) Schematic overview of our semantic grid fusion and prediction architecture. The framework is fed with image sequences $m(t_i)$ of different cameras from past times t_i. These are semantically segmented ($\mathcal{S}$), projected ($\mathcal{P}$) into a agent-centric top-down view 2D point cloud $\gamma(t_i)$ and transformed ($\mathcal{T}$) to the same time τ resulting in spatio-temporally aligned grids $g(t_i \to \tau)$. The encoder-decoder deep neural network (ED) fuses these grids and predicts environment dynamics. The two-step description in Section 3 is indicated as I and II on the right. Note that the point cloud $\gamma(t_i)$ is represented as grid for visualization purpose.

seem biologically plausible. One well known example of interpretable grid-based ER are single- and multi-sensor occupancy grids [2, 3], which have also gained interest in recent deep learning studies [4, 5, 6]). Due to the lack of suitable feature extractors before the raise of deep learning, these grids normally only consisted of one semantic feature encoding the occupancy property. However, recently semantic grid representations have become a subject of interest. For instance, [7, 8, 9] deal with the generation of single-camera, non-predictive semantic grids, while [10] utilizes them for decision-making in autonomous driving as semantic grids are easier to simulate than raw sensor data. Other approaches [11, 12] generate sparse bird's eye view repre-

sentations of the environment by detecting objects in image space and transforming them to the top-down view.

We want to differentiate and hence disentangle two different but both desired predictive properties of autonomous agents. First, the agent should be able to reason about future events, e.g., for planning, and in this sense should be able to form mid- or long term predictions. For instance, [10, 13, 14] perform explicit long-term vehicle trajectory prediction using RNNs. From this, we differentiate a second form of prediction, intrinsic to perception. Sensors and their down-stream signal processing (e.g. semantic segmentation) induce a temporal delay between the actual present state of the world and the agent's belief about this state. Our aim is to design an environment representation that can compensate these short-term system-inherent latencies and synchronize data from different time horizons.

For short-term prediction of camera data, there are approaches predicting extracted image features such as object bounding boxes [15], semantic segmentation [16], or instance segmentation [17]. However, all of these work are in the sensor-dependent image space, which unnecessarily complicates the task considering downstream sensor fusion. In contrast, [4, 18, 19, 20] and [21] employ an end-to-end trainable recurrent architecture to directly predict an unoccluded occupancy grid from laser data, capable to track multiple objects. We distinguish ourselves by investigating the capacity of our architecture in the context of sensor fusion and further, by using camera data instead of laser data, enabling semantically richer representations. Moreover, we combine semantic predictions with a moving sensor instead of analyzing both scenarios separately.

3. Method

In this section, we describe our architecture that generates semantic grids from semantic segmentations of camera images (see Fig. 2). A *semantic grid* is a $g \in \mathbb{R}^{W_G \times H_G \times F}$, with $W_G, H_G, F \in \mathbb{N}$ denoting the grid's width, height, and number of semantic features (e.g. different object classes) respectively. Each grid cell represents a certain area in space and hence the spatial coordinate is determined by the cell's indices. The semantic grid is *egocentric* in the sense that the agent is constantly located at the same grid cell.

Generating synchronized semantic grids (Fig. 2, I): Given a RGB image $m \in \mathbb{R}^{W_I \times H_I \times 3}$, a depth map $d \in \mathbb{R}^{W_I \times H_I}$, where $W_I, H_I \in \mathbb{N}$ denote the pixel resolution, and the agent's egomotion, consisting of the translational $\frac{dq}{dt}$ and the angular velocity $\frac{d\alpha}{dt}$, we first use a semantic segmentation network $\mathcal{S}$ to generate semantic information in image space $s = \mathcal{S}(m) \in [0,1]^{W_I \times H_I \times F}$.

Second, this sensor-dependent representation (in image space) is transformed into a spatially aligned (w.r.t. the agent) continuous 2D point cloud $\gamma \in \mathbb{R}^{W_I \times H_I \times (2+F)}$, using the depth map d to project the semantic information from pixel space to the floor plane of the camera coordinate system at recording time t_i:

$$\gamma(t_i) = \mathcal{P}(s(t_i), d(t_i)) \tag{1}$$

For that purpose, each segmentation pixel $s_{uv} \in \mathbb{R}^F$ with $u \in \{1, \dots, W_I)$ and $v \in \{1, \dots, H_I\}$ is transformed to a 2D point with semantic information $\gamma_{uv} = (\gamma^x_{uv}, \gamma^z_{uv}, \gamma^F_{uv})^T \in \mathbb{R}^2 \times \mathbb{R}^F$ in the agent coordinate system A. This transformation is achieved using the intrinsic camera matrix K and the extrinsic parameters $R_{C\to A}$ and $t_{C\to A}$ describing the camera pose C with respect to the agent (see [22] for details on those matrices):

$$\begin{pmatrix} \gamma^x_{uv} \\ \gamma^y_{uv} \\ \gamma^z_{uv} \end{pmatrix} = (R_{C\to A}, t_{C\to A}) \cdot K^{-1} \cdot d_{uv} \cdot \begin{pmatrix} u \\ v \\ 1 \end{pmatrix}. \tag{2}$$

Using these agent-centric 3D coordinates, the semantic features s are projected onto the ground floor. For simplicity, we define $\mathcal{P}$ as well as the following operators on a single point while they are applied to the whole point cloud:

$$\mathcal{P}(s_{uv}, d_{uv}) := (\gamma^x_{uv}, \gamma^z_{uv}, s_{uv})^T. \tag{3}$$

Third, the semantic grids are spatially aligned based on the agent's current orientation $\alpha^{t_i}_{t_0} = \sum_{j=1}^{i} \frac{d\alpha}{dt}(t_j) \cdot (t_j - t_{j-1})$, the integrated angular component of the egomotion, and the point cloud is discretized to the grid representation:

$$g(t_i) = \mathcal{D}(\mathcal{T}_1(\gamma(t_i), \alpha^{t_i}_{t_0})), \tag{4}$$

$$\mathcal{T}_1(\gamma_{uv}, \alpha) := \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & \\ & 0 & \mathrm{Id}_F \end{bmatrix} \cdot \begin{pmatrix} \gamma^x_{uv} \\ \gamma^z_{uv} \\ \gamma^F_{uv} \end{pmatrix}. \tag{5}$$

Thereby, rotations of the agent do not cause rotation of the entire grid, but only result in a change of the agent's internal orientation $\alpha^{t_i}_{t_0}$ and a rotation of potential new data, reducing quantization errors. The discretization $\mathcal{D}$ assigns the 2D points γ_{uv} to grid cells g_{kl}. If two points are allocated to the same grid cell, they are prioritized preferring small and dynamic classes. In that way, we ensure that no important information is occluded by another class. If no point is assigned to a grid cell, it is classified as "unknown" (black) to model the lack of sensor data in that area.

And fourth, the integrated translational component of egomotion $q^{\tau=t_k}_{t_i} = \sum_{j=i}^{k-1} \frac{dq}{dt}(t_j) \cdot (t_{j+1} - t_j)$ is used for the parameter-free temporal extrapolation (grid translation) into the future time τ:

$$g(t_i \to \tau) = \mathcal{T}_2(g(t_i), q^{\tau}_{t_i}), \tag{6}$$

$$\mathcal{T}_2(g_{kl}, q) := g_{k-q^x, l-q^z}, \tag{7}$$

assuming that q is represented in pixel coordinates. If $k + q^x > W_G$ or $l + q^z > H_G$, g_{kl} is assigned the class "unknown" (black).

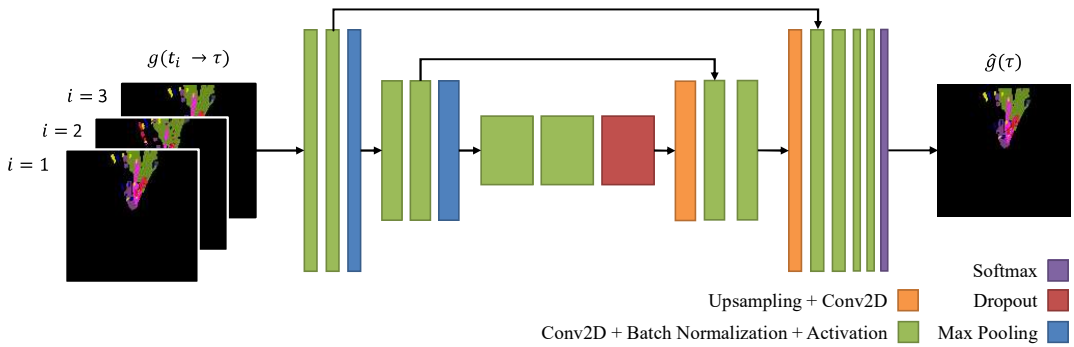

Figure 3. (best viewed in color) Encoder-decoder CNN (ED). The ED combines synchronized semantic grids into one prediction including dynamic object transformation, temporal filtering, and multi-camera fusion.

Predictive fusion architecture (Fig. 2, II): Let $\{m(t_1), \ldots, m(t_n)\}$ denote an input sequence of length n of past images at times $t_1, \ldots, t_n$ obtained from a single sensor, then $\{g(t_1 \rightarrow \tau), \ldots, g(t_n \rightarrow \tau)\}$ are accordingly synchronized to the same future time τ using the parameter-free (and hence not trained) transformations $\mathcal{P}$ and $\mathcal{T} = \mathcal{T}_2 \circ \mathcal{D} \circ \mathcal{T}_1$. These, now synchronized single-sensor semantic grids, are stacked and fed into an encoder-decoder deep neural network (ED) (see Fig. 2 stack). Its task is to fuse the grids from multiple sensors and different original time steps and to incorporate object motion. The ED network yields the predicted semantic grid $\hat{g}(\tau)$ at time τ. In case $\tau - t_n$ covers the system-inherent latency, $\hat{g}(\tau)$ represents the agents belief about the actual present situation. The ED is trained using self-supervised learning. It is provided with grid input sequences $\{g(t_1 \rightarrow \tau), \ldots, g(t_n \rightarrow \tau)\}$ for multiple sensors, while withholding the grid $g(\tau)$, measured at the time $\tau > t_n$, as ground truth.

Loss function: For training we use the categorical cross-entropy loss function. The area M that the sensors have already seen in past frames, but where no ground truth information is available for the target frame, should be ignored during the loss computation. For this purpose, we compute a masked loss

$$\mathcal{L}_{\text{masked}} = f(g(\tau), (1 - M) \cdot \hat{g}(\tau) + M \cdot g(\tau)), \qquad (8)$$

where $g(\tau)$ and $\hat{g}(\tau)$ are the ground truth and predicted semantic grids, respectively, f is some distance measure (e.g. categorical cross entropy), and M is the mask of the covered area without ground truth information $M = M_{\text{covered}} - M_{\text{target}}$. Here, M_{target} denotes the area with ground truth information $M_{\text{target}} = \text{known}(g(\tau))$ and M_{covered} the area the sensors have seen during the entire synchronized sequence (including the target frame $g(\tau)$), $M_{\text{covered}} = \text{known}(g(\tau) + \sum_{i=1}^{n} g(t_i \rightarrow \tau))$. The function "known" determines the area within a grid where the classification is not the class "unknown". In the area determined through M, the network is not penalized for any predictions.

In that way, we want to encourage the network to remember areas, which are occluded in the target frame but were already observed in the past (e.g. the waiting car in Fig. 7b that is newly occluded by another car in the target frame; M visualized in white color), instead of predicting the class "unknown" (black).

4. Experiment Setup

Dataset: For our experiments, we use the Cityscapes dataset of driving scenarios [23], which provides real-world RGB image sequences (30 frames, 17 Hz) including the pre-processed disparity of the stereo camera, the egomotion of the vehicle, as well as training data for semantic segmentation. In that way, Cityscapes allows the generation of semantic grids with various and diverse classes. For evaluation, we use the intersection over union (IoU) [24]

As some classes are comparatively rare due to their small 2D projection during the semantic grid generation for training, we have combined large vehicles (bus, truck, and train) as well as small static objects (poles, traffic signs, and traffic lights), which have a similar semantic meaning and behave similarly in the semantic grid representation. Therefore, the color coding for the classes in a semantic grid does not exactly match with the Cityscapes colors.

To evaluate the performance of the semantic grid prediction with respect to multi-camera fusion, we split the camera frames into separate image sections. While the *Split 1* scenario just divides the images into a lower and an upper part, the *Split 2* scenario additionally has a left and right black margin, as well as a blind area between the lower and upper image section (see Fig. 4). These splits are especially challenging due to the vertical split direction, as there are ambiguities when merging the resulting semantic grids. Moreover, the image is split close below the horizon, which is the most crucial part of the image. In that way, we want to underline the capabilities of the ED to solve ambiguities, track objects between both cameras, and conclude information about blind spots.

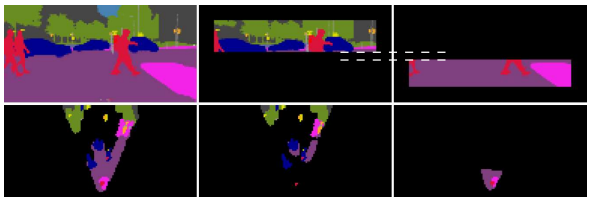

Figure 4. (best viewed in color) Sensor Split 2 for simulating multiple sensors. The semantic segmentation of the camera image (top left) is cropped to simulate two separate sensors (top center and right). In the lower row the associated egocentric semantic grids are visualized. The white dashed lines mark the blind spot.

Semantic Grid Generation: We generate the semantic segmentation of the Cityscapes sequences using the state-of-the-art DeepLabv3 segmentation network [25], which achieves 80.31 IoU on Cityscapes. The semantic labels are transformed to the semantic grid using the ego motion of the vehicle and the disparity of the stereo camera (see Section 3). For the semantic grid, we choose a size of 128×128 pixels, which is the equivalent of 100×100 meters.

Dataset of Grid Sequences: For training and evaluation of our semantic grid prediction framework, the frames $o, o+s, o+2s, \ldots, o+(n-1)s$, where o is the sequence offset and s the step size, are used as input frames, while frame $o + n \cdot s$ represents the target frame. For our experiments, we use the step size $s = 5$, which corresponds to approximately 300 ms, matching typical processing times of semantic segmentation networks. The training sequences are overlappingly sampled (o is chosen arbitrarily), in order to provide as much training samples as possible for the prediction task. As in Cityscapes only every 30$^{\text{th}}$ frame has a manually labeled ground truth, we use the semantic segmentation maps predicted by the network as our ground truth. In contrast to the training sequences, the validation sequences are disjoint and their ground truth frames are aligned for our experiments. In case of $n = 2$, we work with 59500 training sequences and 500 validation sequences. All results are reported on the validation set.

Encoder-Decoder CNN: We use an encoder-decoder convolutional network (ED) similar to U-Net [26] (see Fig. 3). The encoder with depth d contains d downsampling blocks. Each of them reduces the spatial resolution by half and doubles the initial number of features f. The decoder consists of $d - 1$ upsampling blocks and a final softmax layer. Between corresponding down- and upsampling block there are skip connections to enable dense predictions [24].

5. Experiments

In this section, the behavior of the proposed framework is studied. For this purpose, we have designed several experiments, each analyzing certain aspects of our architecture. For all experiments, the results were analyzed according to categories that contain similarly behaving semantic classes: static (unknown, building, road, sidewalk, vegetation), vehicles (car, bus, truck, train), small static (pole, traffic light, traffic sign) and small dynamic (person, bicycle). Note that bus, truck, and train as well as pole, traffic light, and traffic sign were already combined into one class in the dataset. The class-wise mean intersection over union (IoU) for each of those categories is plotted in Fig. 5. The experiment labels (DC, NT, SF,...) are associated with the following paragraphs.

DC *Default config:* We compared our framework utilizing a single sensor (ED-DC) with a simple baseline (BL-DC), which transforms the last sensor frame into the target time.

NT *No explicit translation:* We trained the ED to estimate and apply the translational ego motion $q^{\tau}_{t_n}$ without additional external sensor input or the explicit translation step $\mathcal{T}_2$ (ED-NT).

SF *Sensor fusion:* We analyze the performance of the framework to fuse grids from multiple simulated cameras (ED-SplX).

SL *Sequence length:* We varied the sequence length to study its relation with the prediction performance (ED-SLX).

PH *Prediction horizon:* We evaluated how far the network is able to predict into the future (ED-PHX).

Baselines: To assess the performance of the proposed semantic grid prediction, we designed several baselines that leave out the ED. The simplest baseline BL-NT just replicates the last input grid as prediction $\hat{g}(\tau) = g(t_n)$, while the baseline BL-DC transforms the last input frame using the vehicle's egomotion into the time of the prediction $\hat{g}(\tau) = \mathcal{T}_2(g(t_n), q^{\tau}_{t_n})$. To have baselines that consider the difficulties of the sensor split (Sp), we also designed BL-Sp1 and BL-Sp2, which overlay the semantic grid of the lower sensor with the grid of the upper sensor.

Default config (DC): We compared our framework utilizing a single sensor (ED-DC) with the baseline (BL-DC). As default ED configuration, we have used $n = 2$, $d = 3$, and $f = 64$, as this configuration provides a good trade-off between performance and runtime. In Fig. 5a and 5c column DC, it can be seen that ED-DC significantly outperforms its baseline BL-DC for large static objects and vehicles. On the one hand, the higher performance for large static objects demonstrates the ED's ability to combine the information of the input frames and predict an improved combined grid, which also may contain information about newly occluded areas (see white area of target frame in Fig. 7b). On the other hand, the high performance for ve-

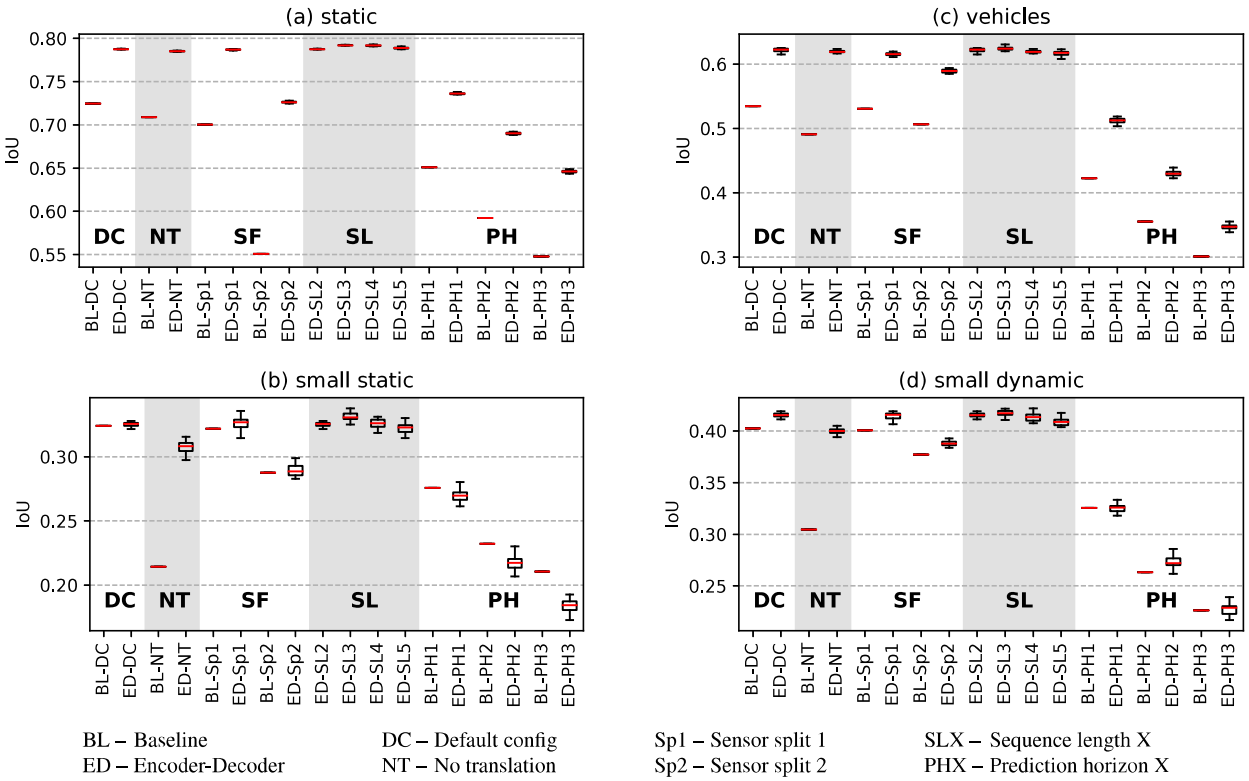

Figure 5. Category-wise mean IoU reported on the validation set with respect to different framework configurations. Further details on the experiments (columns within the plots) are provided in the paragraphs associated with the corresponding references (DC, NT, SF,...).

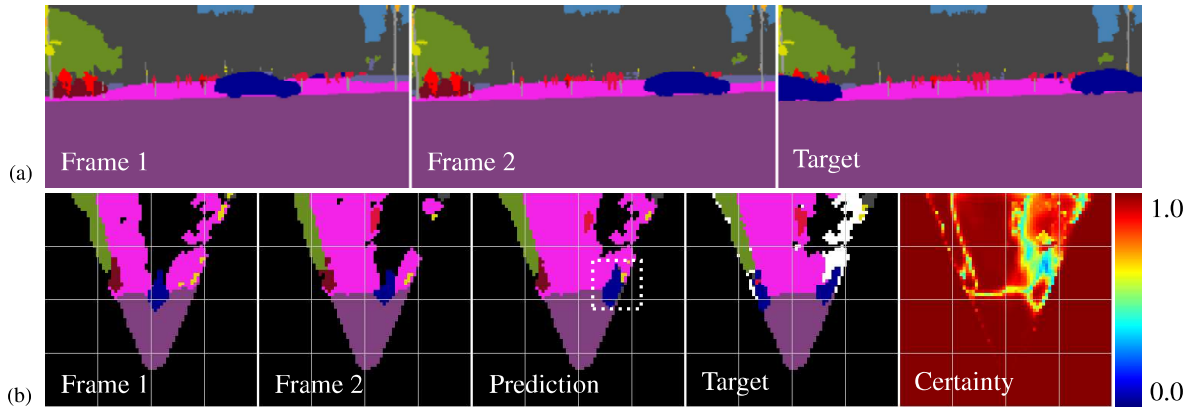

Figure 6. (best viewed in color) Example prediction sequence using ED-DC. (a) shows the sequence of semantic segmentations while (b) presents the associated semantic grids. The framework is fed with frame 1 and 2. The prediction is compared with the target frame. The maximum softmax activation of the prediction is visualized on the right. In this example, a car (dark blue) is moving to the right. Black areas represent unknown parts of the grid and the white color in the target frame visualizes the loss mask M. Note that the left car in the target frame cannot be predicted as it was not visible in frame 1 and 2.

hicles indicates that the ED is able to predict the motion of dynamic objects and use it for the prediction (see Fig. 6b).

Even though classes covering a small area are supposedly quite challenging for ED architectures, our framework is able to outperform the baseline for small dynamic (Fig. 5d) and small static (Fig. 5b) objects (compare ED-DC with BL-DC). However, the absolute IoU is relatively low for both ED-DC and BL-DC. During qualitative analysis of the data, we found that the effect may be due to temporal noise caused by the alignment of RGB and depth image as well as the grid discretization. Small objects sometimes vanish and appear again making it difficult to predict them. This effect can be mitigated by excluding segmentation borders and regions with a minimum depth difference as well

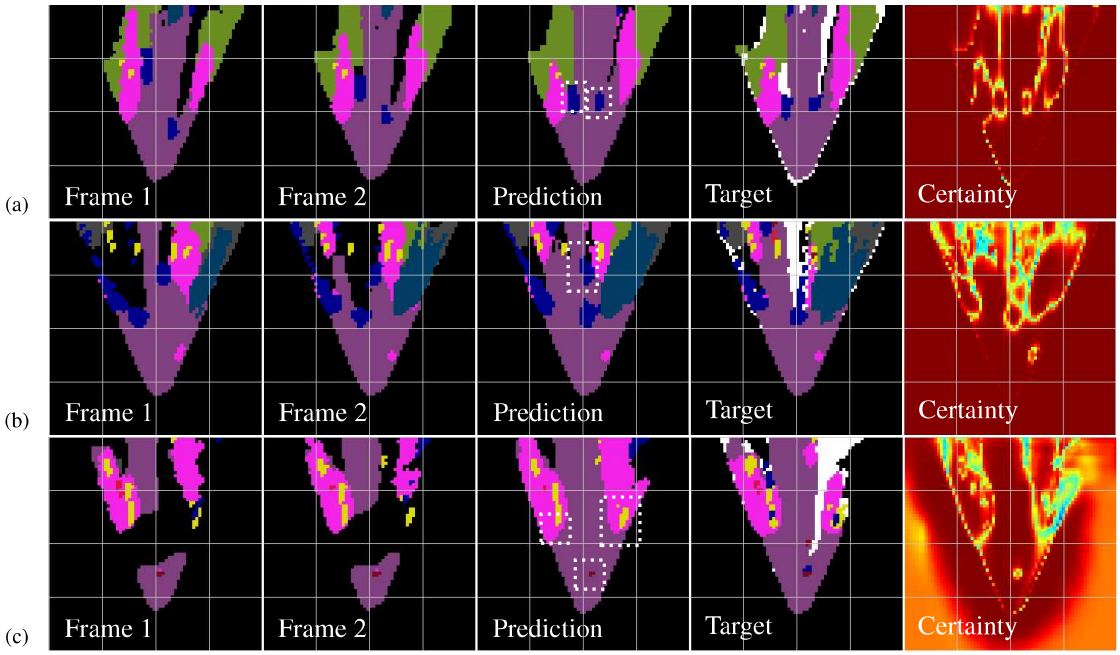

Figure 7. (best viewed in color) Series of sequences and their associated predictions. (a) The left car (dark blue) is approaching while the right one drives away (ED-DC). (b) The ED is able to remember the waiting car, which is occluded in the white area of the target frame (ED-DC) (c) A bicycle (maroon) is moving to the right (ED-Sp2). In contrast to the other examples, it uses the Split 2 dataset. Frame 1 and frame 2 are visualized as combination of both sensors. Moreover, the curvature of the left sidewalk towards the blind spot is recognized and correctly completed. Also, the CNN concludes that there is sidewalk around the pole on the right side.

as by increasing the resolution of the semantic grid.

To underline the abilities of the ED as well as the straightforward human-interpretability of the semantic grid, Fig. 1, 6, and 7 visualize selected predictions from the validation dataset. In Fig. 6, a car (dark blue) is moving perpendicular to the camera. The ED is provided with frame 1 and frame 2. It can be seen that it translates the car according to its velocity to the right, which is quite accurate compared with the target frame. On the right side of Fig. 6b, the maximum softmax activation of the ED is visualized. It can be interpreted as certainty of the CNN's predictions. While the network is certain (red) about already seen static objects such as roads (purple), the maximum softmax activation indicates its uncertainty (green) in the area behind the car and at the borders between classes. The ED is even able to differentiate between instances of the same class and translate them according to their different velocities. This can be seen in Fig. 7a, in which the left car (dark blue) is approaching while the right one is driving away. Moreover, the ED can correctly predict areas that would have been occluded in the target frame while they were visible in the frames before (see waiting car in Fig 7b and compare with white area M in target frame). While the network is able to predict correct classes in most of M, it often fails at the bottom of the grid (see Fig. 1) as there is never ground truth information available during the training process. For a deployable system, we would remove this region from M.

No explicit translation (NT): To study the ED's ability to estimate the translational egomotion and the motion of dynamic objects simultaneously and superimpose both, we disabled the explicit translation $\mathcal{T}_2$ of our framework. However, the ED is still provided with the orientationally aligned semantic grids $g(t_i)$. The framework with disabled translation (ED-NT) reaches the performance of the framework with explicit transformation (ED-DC) for large static areas and vehicles (see Fig. 5a and 5c column DC and NT). Hence, the network is able to solve both tasks in general. However, it struggles with small objects (see Fig. 5b and 5d), probably, as a more precise estimate of the egomotion is needed to predict them correctly due to their small size.

Sensor fusion (SF): In this experiment, the ability of the framework to fuse grids from multiple simulated cameras is evaluated. When using the Split 1 dataset, the framework (ED-Sp1) achieves as good results as without a camera split (ED-DC), as can be seen in Fig. 5 in the corresponding columns. This shows, that the ED is able to fuse multiple sensors and solve ambiguities. However, a CNN trained on Split 2 (ED-Sp2) performs worse than the other configurations so far, as there is information missing in the input data due to the margins left and right of the sensor and the blind area between both sensors (see Fig. 4 and 7c). This fact can also clearly be seen in the baseline of Split 2 (see Fig. 5 BL-Sp2). Even though the missing information decreases the absolute performance, the ED is still able to

maintain the difference to its associated baseline compared to other models (see BL-DC/ED-DC and BL-Sp2/ED-Sp2 in Fig. 5). For static objects, the CNN is even able to increase the difference as it is able to make assumptions about the environment. For instance, ED-Sp2 concludes that there is sidewalk around the pole on the right side of the grid in Fig. 7c even though it has never seen the sidewalk before. Note that during training, the semantic grid generated of the whole camera image, instead of the limited image sections for the virtual cameras, was provided as target frame.

Sequence length (SL): To analyze the influence of the sequence length, we varied n. It can be seen in Fig. 5 column SL that one additional input frame (ED-SL3) improves the performance of all categories in comparison with ED-SL2 as it probably allows better denoising and speed estimation. However, a longer sequence length also decreases the number of sequences that can be sampled from the training set, leading to a drop in performance for ED-SL4/5.

Prediction horizon (PH): In the last experiment, we have analyzed, how far our framework is able to predict into the future. We have maintained the same step size of about 300 ms between the input frames and evaluated target frames one, two, and three steps after the last input frame (ED-PHX). As expected, the performance of the ED drops with increasing prediction horizon (see Fig. 5 column PH). Still, the CNN is able to outperform its associated baselines (BL-PHX) for large and medium objects. However, using a long prediction horizon, the ED struggles with small objects, which is probably caused by the increased sensitivity towards spatial mismatches.

6. Discussion

Grid-based representations are potentially beneficial in applications where very fast reaction times are required and where the duration to compute and update the ER should be independent of the number of objects. However, one limitation of grid-based representations is their linear scalability with respect to the number of semantic features. This is especially true for objects carrying a high variety of semantic information, such as road signs. However, the grid could provide an attention mechanism to identify relevant areas for further processing (e.g. road sign recognition).

In contrast to other representations, which are not interpretably modularized, our approach provides direct interfaces to verify and test the trained encoder-decoder fusion architecture. Normally, for camera or laser input signals, it is hard to synthesize new, especially critical, scenarios. Whereas for our approach, it is comparably easy to generate such sequences of semantic input and respective output grid representations. Further, the interpretable interface functions as a human-readable monitor which can for example be used for debugging or determining corner cases.

Even though we have concentrated on semantic features in this work, the semantic grid can be extended to support a variety of other information. As the semantic grid already provides an abstract, scale-invariant, and low resolution representation of the dynamically changing environment, down-stream algorithms can easily extract further information such as correspondence information or object tracks. If more detailed information is necessary, additional semantic maps encoding local features provided by preprocessing steps can be added to the semantic grid (e.g. local velocities, pedestrian pose, instance segmentation, or uncertainty). It is also possible to train the model to predict semantic grids of multiple time horizons simultaneously to cover short-, mid-, and long-term predictions. Other interesting subjects for future work include: multi-scale or dynamic-resolution grid representations, multi-modal sensor inputs (e.g., LiDAR or radar), sensor signals with different dynamics (e.g., frame rates, offsets), and alternative ED architectures for fusion.

In this work, we omitted the estimation of the actual system-inherent latency. The latency depends on the used hardware and other choices, such as the architecture to extract the semantic information from sensors or the chosen encoder-decoder architecture. Possible solutions are estimating the delay on the final real-world system and using this estimate for the temporal synchronization of the grid representation before fusion or integrating an additional adaptive component that estimates the current latency.

The presented architecture should be thought of as part of a larger system. For example, the provided environment representation could be combined with a recurrent representation to store a mid-term ER, a SLAM approach for global mapping, or with a decision making module. In this context, the semantic grid can act as interpretable interface, which can be used for debugging or determining corner cases. Moreover, this interface simplifies synthesizing artificial training data for new, especially critical, scenarios in comparison to the simulation of raw sensor signals.

7. Conclusion

We presented and evaluated a concept for generating an environment representation supporting multi-camera fusion on autonomous systems. We designed the proposed architecture as grid-based to be independent of the number of objects, egocentric to support sensor fusion, interpretable to modularize the system and enhance human accessibility, and finally predictive to compensate for system-inherent latencies. The architecture was evaluated on the real-world Cityscapes dataset. We demonstrated its superiority to several model-based baselines, its capability to model independent motion of multiple objects, and to fuse ambiguous and incomplete sensor signals. We think that the proposed architecture and design ideas can further be used as a flexible part of a larger framework to control autonomous systems.

References

[1] T. Hafting, M. Fyhn, S. Molden, M.-B. Moser, and E. I. Moser, "Microstructure of a spatial map in the entorhinal cortex," *Nature*, vol. 436, no. 7052, p. 801, 2005.

[2] T. Colleens and J. Colleens, "Occupancy grid mapping: An empirical evaluation," in *Control & Automation, 2007. MED'07. Mediterranean Conference on*. IEEE, 2007, pp. 1–6.

[3] H. P. Moravec, "Sensor fusion in certainty grids for mobile robots," in *Sensor devices and systems for robotics*. Springer, 1989, pp. 253–276.

[4] 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*, vol. 37, no. 4-5, pp. 492–512, 2018.

[5] S. Gupta, J. Davidson, S. Levine, R. Sukthankar, and J. Malik, "Cognitive mapping and planning for visual navigation," *arXiv preprint arXiv:1702.03920*, vol. 3, 2017.

[6] J. Lundell, F. Verdoja, and V. Kyrki, "Deep network uncertainty maps for indoor navigation," *arXiv preprint arXiv:1809.04891*, 2018.

[7] Ö. Erkent, C. Wolf, and C. Laugier, "Semantic grid estimation with occupancy grids and semantic segmentation networks," in *2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV)*. IEEE, 2018, pp. 1051–1056.

[8] Ö. Erkent, C. Wolf, C. Laugier, D. S. González, and V. R. Cano, "Semantic grid estimation with a hybrid bayesian and deep neural network approach," in *IROS*, 2018.

[9] C. Lu, G. Dubbelman, and M. J. G. van de Molengraft, "Monocular semantic occupancy grid mapping with convolutional variational auto-encoders," *arXiv preprint arXiv:1804.02176*, 2018.

[10] M. Bansal, A. Krizhevsky, and A. Ogale, "Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst," *arXiv preprint arXiv:1812.03079*, 2018.

[11] A. Palazzi, G. Borghi, D. Abati, S. Calderara, and R. Cucchiara, "Learning to map vehicles into bird's eye view," in *International Conference on Image Analysis and Processing*. Springer, 2017, pp. 233–243.

[12] D. Wang, C. Devin, Q.-Z. Cai, P. Krähenbühl, and T. Darrell, "Monocular plan view networks for autonomous driving," *arXiv preprint arXiv:1905.06937*, 2019.

[13] N. Lee, W. Choi, P. Vernaza, C. B. Choy, P. H. Torr, and M. Chandraker, "Desire: Distant future prediction in dynamic scenes with interacting agents," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 336–345.

[14] S. Park, B. Kim, C. M. Kang, C. C. Chung, and J. W. Choi, "Sequence-to-sequence prediction of vehicle trajectory via lstm encoder-decoder architecture," *arXiv preprint arXiv:1802.06338*, 2018.

[15] A. Bhattacharyya, M. Fritz, and B. Schiele, "Long-term on-board prediction of people in traffic scenes under uncertainty," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4194–4202.

[16] P. Luc, N. Neverova, C. Couprie, J. Verbeek, and Y. LeCun, "Predicting deeper into the future of semantic segmentation," in *IEEE International Conference on Computer Vision (ICCV)*, vol. 1, 2017.

[17] P. Luc, C. Couprie, Y. Lecun, and J. Verbeek, "Predicting future instance segmentations by forecasting convolutional features," *arXiv preprint arXiv:1803.11496*, 2018.

[18] J. Dequaire, D. Rao, P. Ondruska, D. Wang, and I. Posner, "Deep tracking on the move: learning to track the world from a moving vehicle using recurrent neural networks," *arXiv preprint arXiv:1609.09365*, 2016.

[19] P. Ondrúška, J. Dequaire, D. Z. Wang, and I. Posner, "End-to-end tracking and semantic segmentation using recurrent neural networks," *arXiv preprint arXiv:1604.05091*, 2016.

[20] P. Ondrúška and I. Posner, "Deep tracking: Seeing beyond seeing using recurrent neural networks," *arXiv preprint arXiv:1602.00991*, 2016.

[21] M. Schreiber, S. Hoermann, and K. Dietmayer, "Long-term occupancy grid prediction using recurrent neural networks," *arXiv preprint arXiv:1809.03782*, 2018.

[22] M. Cordts and N. Schneider, "Cityscapes calibration," https://github.com/mcordts/cityscapesScripts/blob/master/docs/csCalibration.pdf.

[23] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 3213–3223.

[24] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[25] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *arXiv preprint arXiv:1706.05587*, 2017.

[26] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.