Enabling spatio-temporal aggregation in Birds-Eye-View Vehicle Estimation

Avishkar Saha[1], Oscar Mendez[1], Chris Russell[2], Richard Bowden[1]

Abstract— **Constructing Birds-Eye-View (BEV) maps from monocular images is typically a complex multi-stage process involving the separate vision tasks of ground plane estimation, road segmentation and 3D object detection. However, recent approaches have adopted end-to-end solutions which warp image-based features from the image-plane to BEV while implicitly taking account of camera geometry. In this work, we show how such instantaneous BEV estimation of a scene can be learnt, and a better state estimation of the world can be achieved by incorporating temporal information. Our model learns a representation from monocular video through factorised 3D convolutions and uses this to estimate a BEV occupancy grid of the final frame. We achieve state-of-the-art results for BEV estimation from monocular images, and establish a new benchmark for single-scene BEV estimation from monocular video.**

I. Introduction

Autonomous vehicles require spatially and semantically-rich representations of their environment and doing this from cameras alone is challenging. While semantic segmentation in the image-plane is a good initial step, it lacks the spatial layout that would make it directly useful for downstream tasks such as trajectory forecasting and path planning. A semantically segmented birds-eye-view (BEV) map provides a compact method of capturing the spatial configuration of a scene and the agents within it.

We formulate BEV estimation from video as predicting an occupancy grid for each semantic category, for each frame, and incorporating temporal cues from the past into our spatial representation of the present. To this end, we learn a spatiotemporal representation that aggregates both local and global dynamics. Our network builds on single-image BEV prediction and by using 3D spatiotemporal convolutions, creates a temporally-aware BEV predictor. Integrating the past in this way and aggregating over time leads to better BEV estimations compared to relying solely on a single image.

The contributions of this paper are (1) We combine best practices across the 2D scene understanding literature to obtain a new state-of-the-art single-image BEV prediction approach. (2) We demonstrate the importance of learning dynamics in the BEV-plane rather than the image-plane and (3) We introduce a temporally-aware BEV predictor which aggregates spatiotemporal information across multiple scales.

II. Related Work

Semantic segmentation in the image-plane offers spatially dense scene semantics. While it can be applied to complex outdoor scenes, 2D image-plane representations lack the spatial relationships needed by self-driving vehicles. However, 3D scene understanding is typically demonstrated on indoor scenes [1]–[3], where strong geometric priors are available. For outdoor scenes, where geometric complexity is greater, layered representations are used to reason about spatial layout and semantics, with a particular focus on occlusion handling [4]–[6]. Such representations are often inadequate for spatial reasoning. In contrast, a birds-eye-view representation [7]–[14], provides a metric spatial description.

Prior work that builds BEV representations from images can be categorised by the image-plane to BEV transform method used: some explicitly use camera geometry [7], [8], [10]–[12], [14], while others learn the transformation implicitly [9], [13].

Approaches that exploit camera geometry can be categorised by the extent to which they use it to guide their models. When transforming an image from perspective-view to BEV, [10]–[12] use pixel-level depth and semantic segmentation maps to backproject segmented objects from the image-plane into BEV. These sparse intermediate representations act as priors upon which the model generates its output. Although object frustums provide helpful cues regarding their horizontal direction and depth, the models require depth and image-plane segmentation maps as additional input. Prior work of [7], [8], [14] instead infers depth and semantics implicitly, thereby foregoing additional annotations.

While the aforementioned methods work well on single-images, with the exception of [11], they do not exploit the temporal relationships in the video. Instead, our spatiotemporal model is specifically designed to utilize video cues to improve BEV prediction accuracy, by conditioning on a sequence as opposed to a single-image.

Learning temporal representations is central to the task of scene understanding in videos, with many approaches tackling the problem from different directions. Srivastava et al. [15] employ LSTMs [16] to propagate features across frames for the task of future frame prediction, a similar task to ours. Their LSTM implicitly learns motion, making it suitable for scenes where the background remains constant due to the lack of ego-motion from the camera. However, self-driving scenarios contain far more ego motion, making the implicit learning of motion insufficient. We use 3D convolutions to build our temporal representation as they

[1] Centre for Vision Speech and Signal Processing, University of Surrey, Guildford, UK, {a.saha, o.mendez, r.bowden}@surrey.ac.uk

[2] *Amazon*, Tubingen, Germany. Work was done prior to joining Amazon. cmruss@amazon.com

provide the freedom to learn motion-specific kernels. One drawback of conventional 3D convolutions is their larger number of parameters, which increase their computational cost and susceptibility to overfitting. Researchers address this speed-accuracy trade-off by factorising 3D convolutions into a spatial convolution followed by a temporal one, which has led to state-of-the-art temporal representations [17]–[19]. In particular, the separable 3D Inception blocks of Xie et al. [19] — based on [20] — demonstrate the best speed-accuracy trade-off. We draw upon these separable 3D convolutions (as well as those of Tran et al. [18]) to learn motion-specific kernels.

Our framework for constructing temporal representations is perhaps closest to the future prediction approach proposed by Hu et al. [21]. Specifically, the authors learn motion between frames using 3D convolutions and hierarchically aggregate them. However, our approaches differ in that we aggregate spatiotemporal features across an input sequence using progressively larger temporal receptive fields. This allows us to build a temporal representation for a single time step, instead of building a representation for an entire sequence.

Importantly, these temporal approaches learn motion in the image-plane; in contrast, our dynamics module operates solely in the BEV-plane.

III. Model

Given a sequence of monocular images captured while driving, our goal is to estimate a BEV spatial layout of the last frame. To represent our spatial BEV maps, we use an occupancy grid parametrisation [22] extended to multiple semantic categories. As with standard occupancy grids, every grid cell m_i is occupied ($m_i = 1$) or free ($m_i = 0$). By extending this formalisation to multiple classes K, the probability that a class occupies a grid cell is $p(m_i^k), k \in K$. Our objective is to predict a set of multi-class binary variables given a sequence of images $\mathbf{I}_{1:t}$:

$$P(\hat{\mathbf{m}_t^k}|\mathbf{I}_{1:t}) = f(\mathbf{I}_{1:t}, \theta) \tag{1}$$

where f is a neural network with weights θ which map perspective space images in the image-plane $\mathbb{P}^I$ to orthographic BEV semantic maps in the BEV-plane $\mathbb{P}^{BEV}$. Hereafter we will use $\mathbb{P}^I$ interchangeably with the image-plane and $\mathbb{P}^{BEV}$ with the XZ or BEV-plane.

A. Predicting semantic BEV maps from perspective images

Our BEV prediction model f (Fig. 1) is composed of a series of sub-networks which are trained together in an end-to-end fashion. It consists of the following steps:

1) **Encode spatial features in the image plane:** given a sequence of images $\mathbf{I}_{1:t}$, extract image features $\mathbf{s}^I_{1:t}$ in $\mathbb{P}^I$ (Eq. 2)
2) **Transform spatial features from the image-plane to BEV-plane:** transform spatial features $\mathbf{s}^I_{1:t}$ into $\mathbb{P}^{BEV}$ to obtain spatial BEV features $\mathbf{s}^{BEV}_{1:t}$ (Eq. 3)
3) **Encode spatiotemporal features in the BEV-plane:** extract dynamic features across a sequence of features $\mathbf{s}^{BEV}_{1:t}$ to obtain a spatiotemporal representation $\mathbf{d}^{BEV}_t$ of the last frame t (Eq. 4)
4) **Decode spatiotemporal representation in BEV-plane:** decode dynamic features $\mathbf{d}^{BEV}_t$ of the final time step t into BEV occupancy grids $\hat{\mathbf{m}}^k_t$ for each semantic category k (Eq. 5)

Thus our overall model is defined as:

$$\mathbf{s}^I_{1:t} = \mathcal{E}(\mathbf{I}_{1:t}) \tag{2}$$

$$\mathbf{s}^{BEV}_{1:t} = \mathcal{T}(\mathbf{s}^I_{1:t}) \tag{3}$$

$$\mathbf{d}^{BEV}_t = \mathcal{D}(\mathbf{s}^{BEV}_{1:t}) \tag{4}$$

$$\hat{\mathbf{m}}^k_t = \mathcal{B}(\mathbf{d}^{BEV}_t), k \in K \tag{5}$$

We emphasise the change in coordinate system: the first two steps operate on perspective features in $\mathbb{P}^I$, and the last two operate on orthographic features in $\mathbb{P}^{BEV}$. Similarly, the first two steps learn spatial-features by operating on each time step individually through 2D convolutions. Step 3 then builds a spatiotemporal representation by processing the entire sequence using factorised 3D convolutions to produce dynamic features for the last time step in the sequence. Step 4 uses 2D convolutions to decode the dynamic features into the final semantic BEV occupancy grids.

The inclusion of spatiotemporal features in step 3 means the approach is conditioned on a sequence of frames. When conditioning on a single image, step 3 can be omitted.

While the encoder (Eq. 2) and decoder (Eq. 5) represent more general functions, the uniqueness of this approach lies in the way the transformation from $\mathbb{P}^I$ to $\mathbb{P}^{BEV}$ is carried out and how the subsequent dynamics are learnt.

B. Image-plane to birds-eye-view transformation

The transformation module in (Eq. 3) warps image-based features $s^I \in \mathbb{R}^{C \times H_\delta \times W}$ into BEV features $s^{BEV} \in \mathbb{R}^{C \times Z \times X}$, where C is the number of channels. As the transformation is done for each frame in the sequence individually, we omit the time step t from the notation for clarity. The transformation process is based on the following premise: given a perspective image in the image plane, inferring an object's depth (z-axis distance) from the camera requires vertical context. However, its position along the x-axis can be determined using camera geometry. With this in mind, the transformation is carried out in the following steps:

1) Each feature map $s^I \in \mathbb{R}^{C \times H_\delta \times W}$ is vertically trimmed to discard redundant context.
2) Every trimmed feature map $s^I \in \mathbb{R}^{C \times H \times W}$ is vertically condensed using a fully-connected layer, resulting in an encoding $s^X \in \mathbb{R}^{C \times 1 \times W}$
3) The encoding s_X is expanded along the z-axis using 1×1 convolutions, resulting in a BEV encoding on a polar spatial grid $s^{\phi(BEV)} \in \mathbb{R}^{C \times Z \times W}$.
4) As the BEV encoding $s^{\phi(BEV)}$ lies on a polar spatial grid, it must be converted to a rectilinear coordinate system for downstream convolutional operations. This

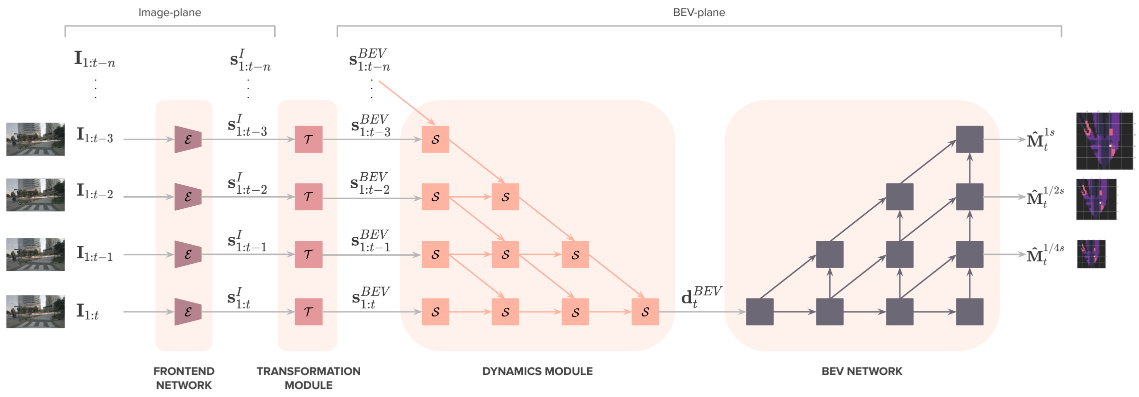

Fig. 1. The architecture of our spatiotemporal model. A **Frontend Network (Eq. 2)** encodes spatial features in the image-plane at multiple scales. The **Transformation Module (Eq. 3)** transforms the extracted spatial features from the image-plane to the BEV-plane. The **Dynamics Module** (Eq. 4) aggregates spatiotemporal features across features of adjacent frames in the BEV-plane to obtain a spatiotemporal representation of frame t. The **Semantic BEV Network (Eq. 5)** processes the BEV spatiotemporal representation and predicts the final occupancy grid probabilities for each class, at multiple scales.

is because regular convolution kernels are not suited to polar representations as the space-varying distortion of the grid makes translational weight sharing ineffective [23].

C. Multi-scale BEV feature transformation

The transformation process described above must occur for image-based features at multiple scales. In the perspective space, objects of the same semantic category appear at various scales in the image depending on their distance to the camera. Thus detecting objects at varying depths (and therefore scale) necessitates high-level feature maps with receptive fields of varying scale — where larger receptive fields capture objects closer to the camera. See Section III-F for more details.

D. Learning dynamics in BEV with 3D convolutions

Once BEV features are generated for every image, we then learn the dynamics over the entire sequence. This spatiotemporal representation allows us to capture the evolution of both static and dynamic scene objects, which aids BEV prediction by overcoming intermittent occlusions.

Autonomous vehicles typically operate in highly-structured environments where the motion in world space of scene objects is typically along two orthogonal axes: parallel to the ego-vehicle, or perpendicular to it. Effectively, during driving, the principal patterns of motion are parallel, perpendicular and occasionally biaxial. The orthogonality of these axes however does not hold true in the image-plane, where the axis for motion parallel to the ego-vehicle is dependent upon the object's vertical position in the image. Hence, we learn motion in the BEV-plane as its perspective-free space is geometrically simpler. As the principal directions of motion in BEV are distinct and well-specified, we use 3D convolutions as their grid-like structure allows us to learn patterns of motion independently.

We explicitly learn these patterns of motion using 3D convolutions factorised into spatial and temporal components. We integrate these filters into the model using a spatiotemporal block as shown in Fig.2. Each block contains both 3D convolution and pooling operations constructed to learn dynamics at both local and global scales. The relationship between the patterns of motion and the axes of the XZ-plane guides the design of our factorised 3D convolution kernels:

1) motion parallel to ego-vehicle is along the z-axis
2) motion perpendicular to ego-vehicle is along the x-axis
3) biaxial motion comprises x and z-axis (the XZ-plane)

With this in mind, the spatiotemporal block takes BEV features $\mathbf{s}^{BEV}_{t-1:t} \in \mathbb{R}^{C\times 2\times Z\times X}$ for a sequence of two consecutive frames and performs the following separate operations:

1) Local spatio-temporal features: three independent streams learn parallel, perpendicular and biaxial motion using 3D convolutions factorised into a spatial convolution followed by a temporal convolution. Spatial features are learned using a kernel of size $(1, k_z, k_x)$. Temporal features meanwhile use kernel sizes $(k_t, 1, k_x)$, $(k_t, k_z, 1)$, (k_t, k_z, k_x) to capture motion along the x-axis, z-axis and on XZ, respectively.

2) Global context: spatio-temporal context is learnt at multiple scales of 1, 1/2 and 1/4 using 3D average pooling layers with sizes (k_t, D, W), $(k_t, \frac{D}{2}, \frac{W}{2})$ and $(k_t, \frac{D}{4}, \frac{W}{4})$.

The seven independent streams from the local and global context layers are preceded by $1 \times 1 \times 1$ convolutions to reduce their channel dimension. Once processed, the output of the layers are then finally concatenated along their channel dimension along with the BEV feature $\mathbf{s}^{t}_{BEV}$ of the last frame to output a single spatio-temporal encoding of two frames $\mathbf{d}^{BEV}_{t} \in \mathbb{R}^{C\times 1\times Z\times X}$.

E. Positional uncertainty and loss

The Dice-coefficient has been shown to outperform cross-entropy loss for semantic segmentation [24], [25]. However, as we have positional uncertainty, we approximate an Earth-Mover's Distance (EMD) by computing the Dice-coefficient at multiple scales, following [26], [27].

F. Spatiotemporal Model Architecture

Frontend network (Eq. 2). This module encodes spatial features in the image plane. A ResNet-50 [28] with 2D

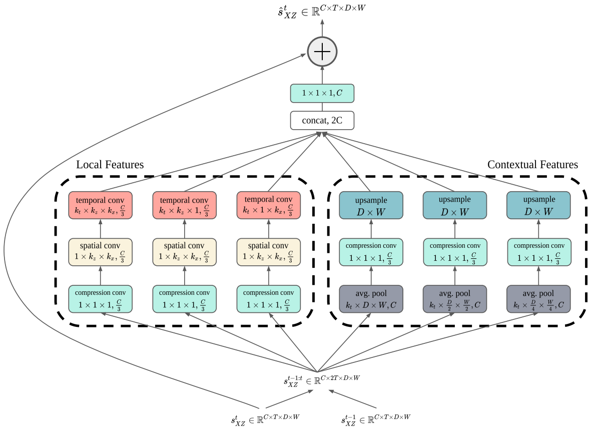
$\hat{s}^t_{XZ} \in \mathbb{R}^{C \times T \times D \times W}$
$1 \times 1 \times 1, C$
concat, 2C
Local Features
Contextual Features
temporal conv $k_t \times k_z \times k_x, \frac{C}{3}$
temporal conv $k_t \times k_z \times 1, \frac{C}{3}$
temporal conv $k_t \times 1 \times k_x, \frac{C}{3}$
upsample $D \times W$
upsample $D \times W$
upsample $D \times W$
spatial conv $1 \times k_z \times k_x, \frac{C}{3}$
spatial conv $1 \times k_z \times k_x, \frac{C}{3}$
spatial conv $1 \times k_z \times k_x, \frac{C}{3}$
compression conv $1 \times 1 \times 1, \frac{C}{3}$
compression conv $1 \times 1 \times 1, \frac{C}{3}$
compression conv $1 \times 1 \times 1, \frac{C}{3}$
compression conv $1 \times 1 \times 1, \frac{C}{3}$
compression conv $1 \times 1 \times 1, \frac{C}{3}$
compression conv $1 \times 1 \times 1, \frac{C}{3}$
avg. pool $k_t \times D \times W, C$
avg. pool $k_t \times \frac{D}{2} \times \frac{W}{2}, C$
avg. pool $k_t \times \frac{D}{4} \times \frac{W}{4}, C$
$s^{t-1:t}_{XZ} \in \mathbb{R}^{C \times 2T \times D \times W}$
$s^t_{XZ} \in \mathbb{R}^{C \times T \times D \times W}$
$s^{t-1}_{XZ} \in \mathbb{R}^{C \times T \times D \times W}$

Fig. 2. Our spatio-temporal block.

convolution kernels takes a single-input image and extracts spatial features at four scales of 1/8, 1/16, 1/32 and 1/64. These features are passed through a feature pyramid network (FPN) [29], supplementing the lower-level, high-resolution features with rich semantic context. This creates semantically strong feature maps $\mathbf{s}^I_{1:t}$ at scales $k \in K$:

$$\mathbf{s}^I_{1:t} = \{\mathbf{s}^I_{1:t,k} \in \mathbb{R}^{C \times h_k \times w_k}\} = \mathcal{E}(\mathbf{I}_{1:t}) \tag{6}$$

where $\mathcal{E}$ denotes the ResNet combined with a feature pyramid.

Transformation Module (Eq. 3). This module transforms spatial features from the image-plane to BEV using the process described in III-B. Coarser features correspond to smaller depth-intervals closer to the camera, while higher-resolution features map to larger depth-intervals further away. As described in III-C, the relationship between feature maps and the depths at which they capture objects is determined by their receptive fields. Table I shows the depth interval that each scale corresponds to and the ResNet layer it is extracted from.

TABLE I

RESNET LAYERS FEATURE MAP SCALES AND THE MAXIMUM DEPTH THEY TRANSFORM TO IN BEV.

Scales, k	1/8	1/16	1/32	1/64
ResNet layer name	conv2_x	conv3_x	conv4_x	conv5_x
Maximum depth, Z	Z	$4Z/5$	$2Z/5$	$Z/6$

Each feature $\mathbf{s}^I_{1:t,k}$ is transformed to a unique depth interval $\mathbf{s}^{BEV}_{1:t,k}$ and the output of the Transformation module is a concatenation of BEV features along depth-axis z:

$$\mathbf{s}^{BEV}_{1:t} = concat_z(\{\mathbf{s}^{BEV}_{1:t,k} \in \mathbb{R}^{C \times z_k \times X} | Z = \textstyle\sum_k^K z_k\}) \tag{7}$$

See Fig.3 for the Frontend and Transformation module.

Dynamics Module (Eq. 4). This module extracts spatiotemporal features from BEV maps. Given a sequence of spatial BEV features $\mathbf{s}^{BEV}_{1:t}$, our Dynamics Module processes it in partitions of two consecutive time steps using the Spatiotemporal Block from section III-D. These blocks are aggregated in a hierarchical manner to produce a single spatiotemporal representation $\mathbf{d}^{BEV}_t$ of the final time step.

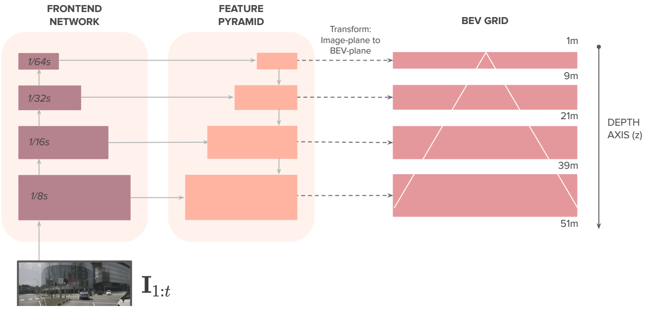

Fig. 3. Frontend network with BEV Transformation module.

The complete structure can be seen in Fig.1. Our aggregated dynamics function $\mathcal{D}_t : \mathbb{R}^{t \times C \times Z \times X} \rightarrow \mathbb{R}^{1 \times C \times Z \times X}$, for a sequence of spatial BEV features $\mathbf{s} = \{\mathbf{s}_1, ..., \mathbf{s}_t\}$, is formulated as:

$$\mathcal{D}_t(\mathbf{s}) = R^{t-2}_t(\mathbf{s}), t \geq 2 \tag{8}$$

where R is defined as

$$R^m_t(\mathbf{s}) = \begin{cases} \mathcal{S}(\mathbf{s}_t, \mathbf{s}_{t-1}), & \text{if } m = 1 \\ \mathcal{S}(R^{m-1}_t(\mathbf{s}), R^{m-1}_{t-1}(\mathbf{s})), & \text{otherwise} \end{cases} \tag{9}$$

and $\mathcal{S}$ is a Spatiotemporal Block.

Semantic BEV network (Eq. 5). This module decodes spatiotemporal features $\mathbf{d}^{BEV}_t$ into semantic BEV occupancy grids $\hat{\mathbf{M}}_\mathbf{t} = \{\hat{\mathbf{m}}^k_t \ |\forall k \in K\}$. The network consists of ResNet blocks [28] in an encoder-decoder structure with deep layer aggregation [30] as shown in Fig.4. We choose an aggregated structure to improve the network's spatial and semantic awareness; aggregating across channels and depths improves inference of *what*, while aggregating across resolutions and scales improves inference of *where* [30].

Although the module's forward-pass is similar to the Dynamics Module defined in (8) and (9), we also extract features from intermediate nodes at multiple-scales.

The spatiotemporal features $\mathbf{d}^{BEV}_t$ are first encoded into a series of layers $\mathbf{x}_1, ..., \mathbf{x}_n$ with progressively richer semantic information; upon which our BEV module $\mathcal{B}_n$ is formulated:

$$\mathcal{B}_n(\mathbf{x}) = \begin{cases} R^{n-2}_n & \text{if } n = 2 \\ \{R^{n-2}_n, ..., R^{n-n}_n\} & \text{if } n \geq 2 \end{cases} \tag{10}$$

where R is defined as in (9) except with a ResNet block instead of the Spatiotemporal block $\mathcal{S}$.

Multi-scale Dice Loss. We supervise different layers of the decoder in the Semantic BEV network, thus injecting gradients deeper into the network and at multiple-scales. At each scale s, the mean Dice Loss across classes K is formulated as:

$$\mathcal{L}^s_{dice} = 1 - \frac{1}{|K|} \sum_{k=1}^{K} \frac{2 \sum_i^N \hat{m}^k_i m^k_i}{\sum_i^N \hat{m}^k_i + m^k_i + \epsilon} \tag{11}$$

where m^k_i is the ground truth binary variable grid cell, $\hat{m}^k_i$ the predicted output of the BEV network passed through a sigmoid, and ϵ is a constant used to prevent division by zero.

TABLE II

IoU(%) ON THE NUSCENES VALIDATION SPLIT OF [8]. WE COMPARE AGAINST BASELINE RESULTS OF PRIOR WORK REPORTED IN [8].

Method	Drivable	Crossing	Walkway	Carpark	Bus	Bike	Car	Cons.Veh.	Motorbike	Trailer	Truck	Ped.	Cone	Barrier	Mean
PON [8]	60.4	28.0	31.0	18.4	20.8	9.4	24.7	**12.3**	7.0	**16.6**	16.3	**8.2**	5.7	8.1	19.1
VED [13]	54.7	12.0	20.7	13.5	0.0	0.0	8.8	0.0	0.0	7.4	0.2	0.0	0	4.0	8.7
VPN [9]	58.0	27.3	29.4	12.3	20.0	4.4	25.5	4.9	5.6	**16.6**	17.3	7.1	4.6	10.8	17.5
Our Spatial	**71.1**	**31.5**	**32.0**	**28.0**	**22.8**	**14.6**	**34.6**	10.0	**7.1**	11.4	**18.1**	7.4	**5.8**	**10.8**	**21.8**
Our Spatiotemp.	**70.7**	**31.1**	**32.4**	**33.5**	**29.2**	**12.1**	**36.0**	**12.1**	**8.0**	**13.6**	**22.8**	**8.6**	**6.9**	**14.2**	**23.7**

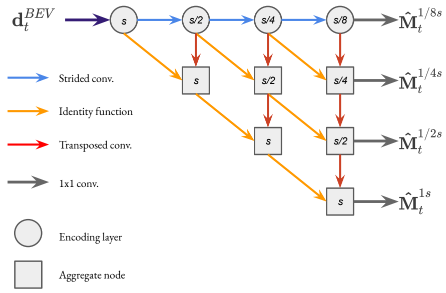

Fig. 4. Semantic BEV network with intermediate supervision at multiple scales. The network has an encoder-decoder structure with deep layer aggregation. Blue arrows represent downsampling by strided convolution, green upsampling by strided transposed convolution, and orange an identity function.

IV. EXPERIMENTS AND RESULTS

We evaluate our models on the nuScenes dataset [31]. We start by demonstrating the effect of multi-scale intermediate supervision on our spatial models. We then evaluate our spatiotemporal models, where we compare learning dynamics in BEV to the image-plane. Finally, we evaluate our spatial and spatiotemporal models against other published methods.

Dataset: The NuScenes dataset [31] consists of 1000 short 20-second clips captured across Boston and Singapore. Each scene is fully annotated with 3D bounding boxes for 23 object classes. It also provides detailed vectorised maps which include road lanes, sidewalks, carparks and more. We follow [8] and select four map categories and seven object categories; we also use their training and validation split.

Evaluation metrics: We evaluate Intersection-Over-Union (IoU) accuracy. Given the model's sigmoid output, we create a binary map for each class based on a threshold of $p(m_i) > 0.5$, as done by [8]. Unless otherwise specified, all IoUs reported correspond to the largest/final layer, which for evaluation is resized to 200×200 to be comparable with [8].

Implementation details: For our frontend we use a ResNet-50 [28] with a feature pyramid [29] on top of it, with both having been pretrained together. BEV feature maps built by the Transformation Module have a resolution of 100×100 pixels, with each pixel being 0.5m. For our spatiotemporal model, the Dynamics Module takes a 12Hz sequence of 6 images, where the last frame in the sequence is the time step we are making a BEV prediction for. Our BEV network is constructed as an encoder-decoder, with layers at scales of $1, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}$. In between the encoder and decoder layers, we aggregate features across scales as shown in Fig. 4. Our largest scale output is 100×100 pixels, which we resize to 200×200 for evaluation. We use Adam as our optimizer, with a weight decay of 0.0001. For our spatial models, we use a batch size of 7 with gradients accumulated over 7 iterations. In our spatiotemporal models, every batch is a sequence of size 6, with gradients accumulated over 50 iterations. We exponentially decay an initial learning rate of 5×10^{-5} by 0.99 every epoch, and train for 60 epochs.

Spatial model ablation studies: Table III shows model accuracy when varying the depth and scale of supervision provided to the Semantic BEV Network. These models were run at a BEV representation size of 50×50 pixels, and on a 50% subset of NuScenes; the IoU reported corresponds to the model's 50×50 output resized to 200×200 for evaluation.

TABLE III

IoU(%) WHEN VARYING DEPTH OF SUPERVISION TO THE BEV NETWORK.

Supervision scales	Static Classes	Dynamic Classes
1s	31.2	5.0
1s, 1/4s	32.5	5.5
1s, 1/4s, 1/8s	34.4	7.5
1s, 1/4s, 1/8s, 1/16s	**35.5**	**10.8**

Adding supervision at multiple scales progressively increases model IoU, particularly for the dynamic, smaller classes which see a twofold increase between single and quadruple-scale supervision. This difference in accuracy can be attributed to the change in behaviour of the training signal when going from single to multi-scale. Supervising at a single scale using the Dice coefficient makes the loss particularly sparse for small classes. By adding supervision at smaller scales, where object scale is larger in the image, we can emulate the behaviour of an earth movers distance. This means a prediction that misses by a few pixels at the largest scale may infact intersect with the ground truth at smaller scales. Thus, the multi-scale supervision provides the model with cues akin to a distance metric on its positional uncertainty.

Just as the supervision pyramid aids training, its effect on the positional certainty of the network is clear in Table IV. The dynamic, smaller classes in particular increase in IoU by a relative factor of 35% at the smallest scales. We believe this is one of the strengths of our approach: even if the model has less true positives at larger resolutions (or equally, is unable to render the shape details accurately at larger resolutions), its lower resolution outputs have a better sense of position.

Spatiotemporal model ablation studies: In Table V we demonstrate the effectiveness of learning dynamics in BEV compared to the image-plane. To learn dynamics in the image-plane, we amend the model by shifting the Dynamics Module before the Transformation Module; the configuration of the motion-specific kernels in the Spatiotemporal blocks

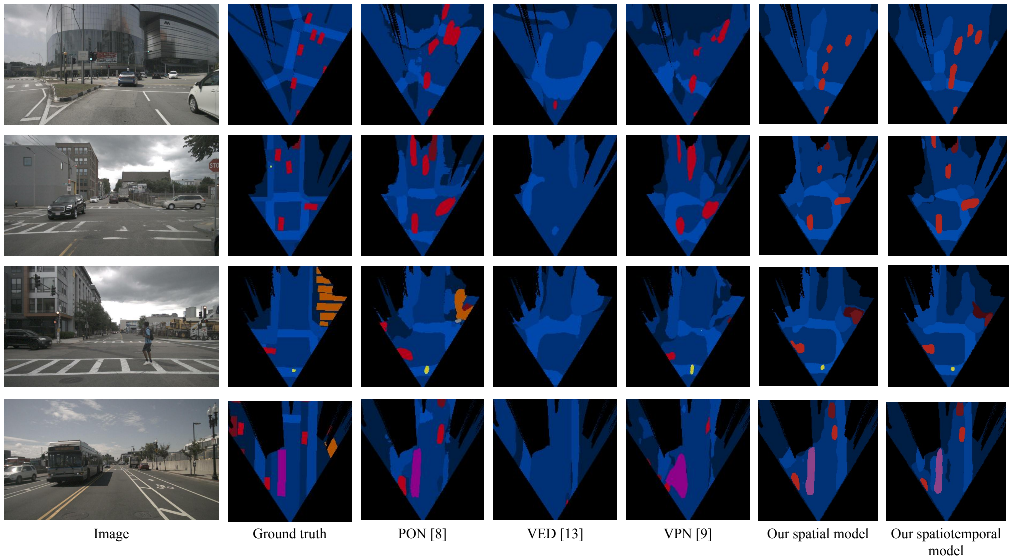

Fig. 5. Qualitative results on the NuScenes validation set. Like the quantitative assessment, we compare against baseline results of prior work reported in [8] and follow their colour scheme. For fair comparison, we apply the ground truth visibility mask (black) to the predicted images as was done in [8].

TABLE IV

IoU(%) AT MULTIPLE-SCALES AND DEPTHS WITHIN THE BEV NETWORK. THE IoU REPORTED IN THIS TABLE CORRESPONDS TO EVALUATION PERFORMED AT THE SCALE OF SUPERVISION.

Supervision scales	Static Classes	Dynamic Classes
100×100	38.0	17.9
50×50	40.0	20.0
25×25	42.8	23.6
13×13	**48.2**	**27.5**

is kept the same as our normal spatiotemporal model. With the large increase in IoU, it is clear that our motion-specific kernels are better suited to the grid-like motion seen in the BEV-plane as opposed to the much larger variation seen in image-plane. By allowing each spatiotemporal kernel in the BEV-plane to look for motion in a specific direction, they become easier to train than using kernels that are agnostic to the direction of motion.

TABLE V

IoU(%) FOR SPATIOTEMPORAL MODELS WITH DYNAMICS LEARNT IN THE IMAGE-PLANE, OR THE BEV-PLANE.

Dynamics Plane	Static Classes	Dynamic Classes
Image-plane	29.8	9.9
BEV-plane	**41.8**	**17.6**

Baselines: Given that semantic BEV prediction from monocular images is a relatively new task — for both static and dynamic classes — there are only a few established baselines that we can compare our work to: the Pyramid Occupancy Network (PON) of [8], the Variational AutoEncoder (VED) of [13] and the View Parsing Network (VPN) of [9]. We do not include the results of Philion and Fidler [14] as they use different subsets of NuScenes data, different class labels, and a different quantization of the BEV-grid.

As shown in Table II, our spatial model outperforms all previous methods, including the state-of-the-art PON method of [8] by an average of 2.7%. It is the dynamic, smaller classes on which we show significant improvement e.g. the car class demonstrates a 10% increase and bicycles a 5% increase.

However, our spatiotemporal model outperforms our spatial model by a further 1.9%. While the larger classes see some gain, it is the smaller, dynamic classes where the performance improvement is most evident. This can be seen in Fig. 5, where our spatiotemporal model displays better positional and shape accuracy than its spatial counterpart.

Limitations and improvements: One of the clear drawbacks of our Dynamics Module is that the depth of its aggregate structure scales with the length of its input sequence. Although our spatiotemporal model uses sequence lengths of 6 frames at 12Hz, it may well be that some frames are redundant. Future work could be identifying which are the most useful frames for building compact spatiotemporal representations. As discussed previously, another area for improvement is the difference between the IoU at the smallest and largest scales of the Semantic BEV Network, highlighted in Table IV. Future work could find ways to ensure positional certainty does not decrease at the larger resolution outputs.

V. Conclusion

We have presented a framework for instantaneous BEV estimation of a scene from both monocular images and video. In particular we have demonstrated an approach to integrating temporal information which results in a better state estimation of the world. One of our key insights is that spatiotemporal convolutions are better suited to the BEV-plane than the image-plane. Our models set a new state-of-the-art for BEV estimation from monocular images while establishing a new benchmark for monocular video.

Acknowledgement

This work was supported by the EPSRC project ROSSINI (EP/S016317/1) and iCASE Studentship (19000091).

References

[1] C. Liu, A. Schwing, K. Kundu, R. Urtasun, and S. Fidler, "Rent3d: Floor-plan priors for monocular layout estimation," in *CVPR*, 2015.

[2] I. Armeni, O. Sener, A. R. Zamir, H. Jiang, I. Brilakis, M. Fischer, and S. Savarese, "3d semantic parsing of large-scale indoor spaces," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1534–1543.

[3] S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. Funkhouser, "Semantic scene completion from a single depth image," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1746–1754.

[4] J. Tighe, M. Niethammer, and S. Lazebnik, "Scene parsing with object instances and occlusion ordering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 3748–3755.

[5] R. Guo and D. Hoiem, "Beyond the line of sight: labeling the underlying surfaces," in *European Conference on Computer Vision*. Springer, 2012, pp. 761–774.

[6] S. Tulsiani, R. Tucker, and N. Snavely, "Layer-structured 3d scene inference via view synthesis," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 302–317.

[7] T. Roddick, A. Kendall, and R. Cipolla, "Orthographic feature transform for monocular 3d object detection," 2019.

[8] T. Roddick and R. Cipolla, "Predicting semantic map representations from images using pyramid occupancy networks," in *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[9] B. Pan, J. Sun, H. Y. T. Leung, A. Andonian, and B. Zhou, "Cross-view semantic segmentation for sensing surroundings," *IEEE Robotics and Automation Letters*, 2020.

[10] Z. Wang, B. Liu, S. Schulter, and M. Chandraker, "A parametric top-view representation of complex road scenes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10 325–10 333.

[11] B. Liu, B. Zhuang, S. Schulter, P. Ji, and M. Chandraker, "Understanding road layout from videos as a whole," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 4414–4423.

[12] S. Schulter, M. Zhai, N. Jacobs, and M. Chandraker, "Learning to look around objects for top-view representations of outdoor scenes," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 787–802.

[13] C. Lu, M. J. G. van de Molengraft, and G. Dubbelman, "Monocular semantic occupancy grid mapping with convolutional variational encoder–decoder networks," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 445–452, 2019.

[14] J. Philion and S. Fidler, "Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d," in *Proceedings of the European Conference on Computer Vision*, 2020.

[15] N. Srivastava, E. Mansimov, and R. Salakhudinov, "Unsupervised learning of video representations using lstms," in *International conference on machine learning*, 2015, pp. 843–852.

[16] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[17] L. Sun, K. Jia, D.-Y. Yeung, and B. E. Shi, "Human action recognition using factorized spatio-temporal convolutional networks," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 4597–4605.

[18] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 4489–4497.

[19] S. Xie, C. Sun, J. Huang, Z. Tu, and K. Murphy, "Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 305–321.

[20] J. Carreira and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," in *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6299–6308.

[21] A. Hu, F. Cotter, N. Mohan, C. Gurau, and A. Kendall, "Probabilistic future prediction for video scene understanding," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[22] A. Elfes, "Occupancy Grids: A Stochastic Spatial Representation for Active Robot Perception," in *Proceedings of the Sixth Conference on Uncertainty in AI*, vol. Vol. 2929, 1990, p. 6.

[23] T. S. Cohen, M. Geiger, J. Köhler, and M. Welling, "Spherical CNNs," in *International Conference on Learning Representations*, 2018. [Online]. Available: https://openreview.net/forum?id=Hkbd5xZRb

[24] M. A. Rahman and Y. Wang, "Optimizing intersection-over-union in deep neural networks for image segmentation," in *International symposium on visual computing*. Springer, 2016, pp. 234–244.

[25] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-net: Fully convolutional neural networks for volumetric medical image segmentation," in *2016 fourth international conference on 3D vision (3DV)*. IEEE, 2016, pp. 565–571.

[26] S. Shirdhonkar and D. W. Jacobs, "Approximate earth mover's distance in linear time," in *2008 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2008, pp. 1–8.

[27] Haibin Ling and K. Okada, "Diffusion distance for histogram comparison," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 1, 2006, pp. 246–253.

[28] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[29] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117–2125.

[30] F. Yu, D. Wang, E. Shelhamer, and T. Darrell, "Deep layer aggregation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 2403–2412.

[31] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 621–11 631.