Understanding Road Layout from Videos as a Whole

Buyu Liu[1] Bingbing Zhuang[1] Samuel Schulter[1] Pan Ji[1] Manmohan Chandraker[1,2]
[1]NEC Laboratories America [2]UC San Diego

Abstract

In this paper, we address the problem of inferring the layout of complex road scenes from video sequences. To this end, we formulate it as a top-view road attributes prediction problem and our goal is to predict these attributes for each frame both accurately and consistently. In contrast to prior work, we exploit the following three novel aspects: leveraging camera motions in videos, including context cues and incorporating long-term video information. Specifically, we introduce a model that aims to enforce prediction consistency in videos. Our model consists of one LSTM and one Feature Transform Module (FTM). The former implicitly incorporates the consistency constraint with its hidden states, and the latter explicitly takes the camera motion into consideration when aggregating information along videos. Moreover, we propose to incorporate context information by introducing road participants, e.g. objects, into our model. When the entire video sequence is available, our model is also able to encode both local and global cues, e.g. information from both past and future frames. Experiments on two data sets show that: (1) Incorporating either global or contextual cues improves the prediction accuracy and leveraging both gives the best performance. (2) Introducing the LSTM and FTM modules improves the prediction consistency in videos. (3) The proposed method outperforms the SOTA by a large margin.

1. Introduction

Understanding 3D properties of road scenes from single or multiple images is an important and challenging task. Semantic segmentation [2, 3, 41], (monocular) depth estimation [9, 16, 40] and road layout estimation [10, 25, 28, 38] are some well-explored directions for single image 3D scene understanding. Compared to image-based inputs, videos provide the opportunity to exploit more cues such as temporal coherence, dynamics and context [19], yet 3D scene understanding in videos is relatively under-explored, especially with long-term inputs. This work takes a step towards 3D road scene understanding in videos through a holistic consideration of local, global and consistency cues.

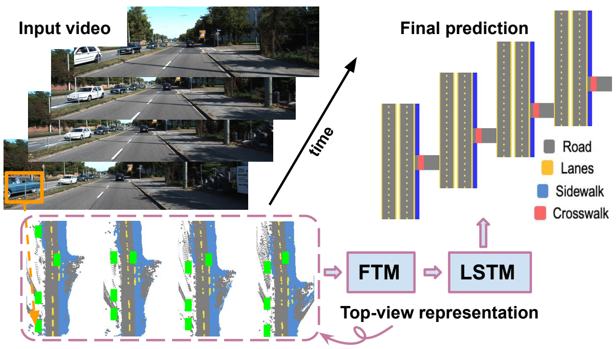

Figure 1: Given perspective images (top left) that captures a 3D scene, our goal is to predict the layout of complex driving scenes in top-view both accurately and coherently.

To this end, we formulate this video scene understanding problem as a road layout estimation problem. We are particularly interested in estimating top-view representations [25, 27] as working in the top-view is beneficial for higher-level reasoning and decision making for downstream applications, e.g. path planning and autonomous driving. Given a video sequence, our goal is to predict the road layout of each frame both accurately and coherently.

To achieve that, we first propose a novel network structure to enforce temporal smoothness in videos. Specifically, our network consists of (i) an LSTM [13] that incorporates the long-range prediction coherency and (ii) a Feature Transform Module (FTM) that explicitly aggregates information w.r.t. estimated camera motion and encourages consistency on the feature level. By explicitly and implicitly encoding temporal smoothness constraints, our proposed network is able to improve the prediction accuracy and consistency. Apart from incorporating temporal coherency in videos, we further propose to exploit context cues in scene attribute prediction in the top-view. More specifically, such cues are represented and obtained with 3D object detection [17, 18] and provide useful priors and constraints for layout estimation in the top-view. For instance, by looking at only the green rectangles in the "top-view representation" in Fig. 1, one can obtain a rough estimate of the drivable areas.

The above mentioned proposals are applicable for both offline and online scene understanding in videos. When the online scenario is not required, we can further improve our model by leveraging even more powerful temporal cues from future frames. Applications like forensic analysis of traffic scenes and driver behavior studies using commodity cameras are examples where road layout estimation is required while the entire videos can be available. For such cases, we further propose to combine information obtained locally from individual frames with that obtained globally from entire video sequences. Specifically, we utilize structure from motion (SfM) [23] and multiple view stereo (MVS) [24] to obtain a dense 3D reconstruction of the road scene from the video sequence. The multiview geometric constraints encoded in such methods naturally aggregate the temporal information from multiple frames, thus permitting a more coherent scene representation as compared to individual views. This allows us to build the model input that is of better representation power and smoother, which boosts the prediction accuracy as well as coherence (see Fig. 1).

We conduct experiments to validate our ideas on two public driving data sets, KITTI [8] and NuScenes [22] (Sec. 4). Our results demonstrate the effectiveness of the global and context cues in terms of prediction accuracy, and the importance of the LSTM as well as FTM for consistent outputs. To summarize, our key contributions are:

- A novel neural network that includes (i) an LSTM that implicitly enforces prediction coherency in videos and (ii) a Feature Transform Module that explicitly encodes the feature level consistency w.r.t. camera motion.
- An input representation that considers information obtained (i) locally from individual frames, (ii) globally from entire video sequence and (iii) from contextual cues.
- Experimental evaluation that outperforms state-of-the-art on public datasets by a large margin.

2. Related Work

Scene understanding is an important yet challenging task in computer vision, which enables us to perform applications such as robot navigation [11], autonomous driving [7, 15], augmented reality [1] or real estate [20, 31].

3D scene understanding: 3D scene understanding is most frequently explored for indoor scenes [1, 20, 30], which is typically formulated as a room layout estimation problem. However, unlike the indoor scenes where strong priors, e.g. Manhattan world assumption, are available, scene understanding for outdoor scenarios is less constrained, thus can be more challenging. To this end, many non-parametric approaches have been proposed [10, 34, 35], where layered representations are utilized to reason about the geometry as well as semantics in occluded areas. More recent work [25, 28] propose top-view representations to provide a more detailed description for outdoor scenes in 3D. As for parametric models, Seff and Xiao [27] propose a model that consists of certain road scene attributes and further utilize a neural network to directly predict them from a single perspective RGB image. However, these pre-defined attributes are not rich enough to capture various types of road layout. A more recent work [15] presents a graph-based road representation that includes lanes and lane markings, from partial segmentation of an image. However, this representation focuses on straight roads only. Similarly, an interesting model is proposed by Máttyus *et al.* [21] to augment existing map data with richer semantics. Again, this model only handles straight roads and requires additional input from aerial images. To further handle complex road layouts with traffic participants, Geiger *et al.* [7] propose to utilize multiple modalities as input, such as vehicle tracklets, vanishing points and scene flow. Recently, a richer model [38] is proposed to handle more complex road scenarios, e.g. multiple lanes and different types of intersections. Our work follows the parametric representation proposed in [38]. Unlike [38] that only takes local information, e.g. pixel-level depth and semantic segmentation from a single frame, we propose to explore multiple aspects in videos, e.g. global information and context cues, to obtain accurate and coherent predictions.

Scene understanding in videos: Apart from accuracy, scene understanding in videos further requires consistent predictions between consecutive frames. Spatio-temporal probabilistic graphical models [19, 38] are widely explored for outdoor scenes. Nowadays, recurrent neural networks (RNNs) like LSTMs [4, 13, 32] are used to propagate feature representations from still frames over time. However, LSTMs implicitly enforce the prediction consistency without the explicit knowledge of motion. To this end, more recent work [5, 29] propose to combine features extracted from both images and motion, or optical flow, to boost the representational power. Although motion is fed as additional input for networks in these above mentioned networks, it is not utilized to transform features over time to explicitly enforce the feature level consistency. Maybe [36, 42] are the most closest work to ours in terms of feature warping and aggregation. Specifically, [42, 43] explicitly warp feature maps [6] between frames and propose to aggregate them in a more well-aligned manner. To address the fixed temporal width problem in [42], more recent work [36] introduces a feature memory that is warped from one frame to another, which enables a longer temporal horizon without looking into future frames. In contrast, we introduce a Feature Transform Module (FTM) that warps features w.r.t camera motion and aggregates them between consecutive frames. And we further propose to combine LSTM and FTM to implicitly and explicitly enforce temporal smoothness in predictions. More importantly, unlike prior methods that estimate flow in the perspective view of the scene, all our modules work in the top-view space.

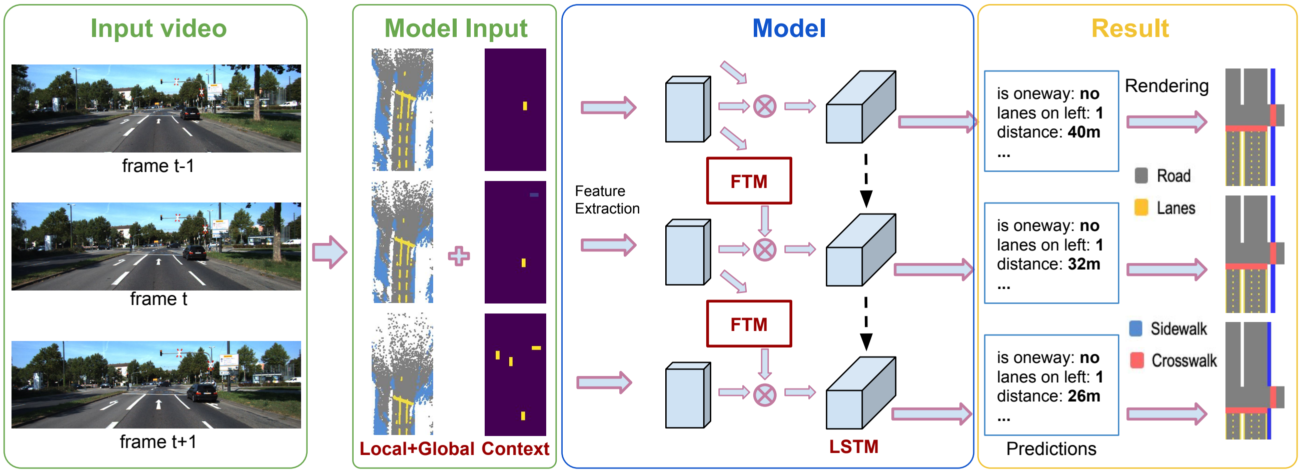

Figure 2: **Overview of our proposed framework:** Given videos as input, top-view maps aggregated from local, global and context information, are fed into our FTM/LSTM network to predict the parametric road scene layout.

3. Our Framework

We briefly introduce the parameterized scene model in Sec. 3.1. Then we describe our proposed LSTM with FTM in Sec. 3.2, followed by the utlization of context information and global information in Sec. 3.3. An overview of our framework is shown in Fig. 2.

3.1. Scene Representation

Given a monocular video sequence capturing a road in the perspective view, our goal is to obtain a coherent yet accurate road layout estimation in the top-view for each frame. To describe the top-view layout, we follow the parametric representation of [38], which consists of three different types of scene attributes/variables, including $M^{\mathrm{b}} = 14$ binary variables Θ_{b}, $M^{\mathrm{m}} = 2$ multi-class variables Θ_{m} and $M^{\mathrm{c}} = 22$ continuous variables Θ_{c}. These parameters can represent a large variety of road scene layouts [1]. See Fig. 1 for examples of predicted results. Denoting the scene model parameters $\Theta = \{\Theta_{\mathrm{b}}, \Theta_{\mathrm{m}}, \Theta_{\mathrm{c}}\}$, we aim to predict Θ^t, the scene parameters of the t-th frame, coherently and accurately for all $t \in \{1, \ldots, T\}$, where T is the number of frames in a video. However, instead of making predictions from single image, we propose to augment the underlying feature representation and aggregate temporal information from videos for more temporally consistent and robust predictions.

3.2. Our LSTM-based Model

In order to exploit long-term temporal information, we propose to utilize LSTM for road layout prediction. In addition, we propose a Feture Transform Module (FTM) to more explicitly aggregate temporal information. We will discuss more details below.

We denote the representation of the road scene at a certain view as $\mathbf{x} \in \mathbb{R}^{H \times W \times C}$, and hence $\mathbf{x}^t$ at the t-th frame in a video. Here, $\mathbf{x}$ can be regarded as any generic representation such as RGB perspective image or top-view image of the road; our proposed novel top-view representation will be discussed shortly in Sec. 3.3. Given $\mathbf{x}^t$, our overall model is defined as following:

$$\begin{aligned} y^t &= f_{com}(g_i(\mathbf{x}^t), g_{ft}(g_i(\mathbf{x}^{t-1}))),\\ \Theta^t &= h(g_{lstm}(g_j(y^t))), \end{aligned} \tag{1}$$

where h, g_* and f_* are neural networks, with weights γ_h, γ_{g*} and γ_{f*} respectively, that we want to train. y^t is the auxiliary intermediate feature representation. Our network structure is illustrated in Fig. 3. The architecture of g_i is a shallow (one-layer) convolutional neural network (CNN) and g_j is a deeper (seven-layer) network. We firstly pass the individual input $\mathbf{x}^t$ to g_i and receive feature $g_i(\mathbf{x}^t)$. Then this feature is combined with $g_{ft}(g_i(\mathbf{x}^{t-1}))$, which is obtained by feeding $g_i(\mathbf{x}^{t-1})$, the feature obtained from previous frame, to FTM g_{ft}, and feed to g_j. The output of g_j is a 1-dimensional feature vector (for each frame) that is further fed into our LSTM module g_{lstm}. g_{lstm} then outputs features that implicitly encode information from previous frame by incorporating hidden states from $\mathbf{x}^{t-1}$ and send to h. Then, the function h is defined as a multi-layer perceptron (MLP) predicting the scene attributes Θ^t with features obtained from LSTM module. Specifically, h is implemented as a multi-task network with three separate predictions for each of the parameter groups of the scene model [2].

3.2.1 Feature transform module

Assuming that after g_i, we are able to extract a feature map $\mathbf{F} = g_i(\mathbf{x}) \in \mathbb{R}^{h_f \times w_f \times c_f}$, where h_f, w_f, and c_f denote the height, width and feature dimension. We would like

[1] We refer to our supplementary material and [38] for more details.

[2] In experiments, we also tried both bi-directional LSTM [26] and ConvLSTM [39]. We did not observe significant improvements over traditional LSTM.

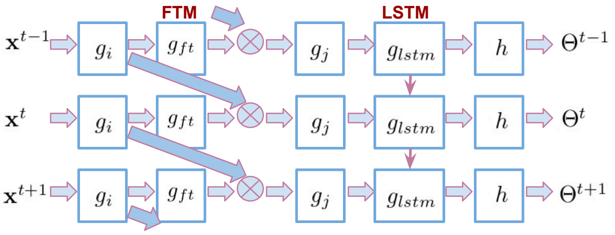

Figure 3: Overview of our network.

the network to learn more robust feature representations by encouraging temporal consistency. To this end, we propose to aggregate the features that correspond to the same location in the scene from the nearby frames, and feed the composited features into the later prediction layers. Intuitively, we would like the network to make use of the features at the same point but from different views for more robust prediction, as well as encourage more temporally consistent feature learning.

Feature correspondence: In order to aggregate the features corresponding to the same spatial location, we need to find a correspondence between two feature maps in neighbouring frames. One obvious way to achieve this is by computing optical flow [33] between neighbouring perspective images and transforming the correspondence w.r.t. the flow field on the feature map. We later show in Sec. 3.3.2 that such computation can be obviated here as we have the camera poses and 3D reconstruction of the road built from the video.

Feature transformation: Given the correspondence between feature maps, we can warp [36, 42] the feature map at time step $t-1$ to the current time step t, denoted as:

$$g_{ft}(\mathbf{F}^{t-1}) = \phi(\mathbf{F}^{t-1},\ S^{t-1}(\mathbf{x}^t, \mathbf{x}^{t-1})) \qquad (2)$$

where $\phi(*)$ is the bilinear sampling function [14] and $S^{t-1}(\mathbf{x}^t, \mathbf{x}^{t-1}) \in \mathbb{R}^{h_f \times w_f \times 2}$ is a displacement (or flow) field between frames t and t-1.

Feature composition: One can choose various ways to aggregate the feature map $g_{ft}(\mathbf{F}^{t-1})$ and $\mathbf{F}^t$. Here, we adopt the following simple weighted summation of the two,

$$f_{com}(\mathbf{F}^t, g_{ft}(\mathbf{F}^{t-1})) = \alpha \cdot \mathbf{F}^t + (1-\alpha) \cdot g_{ft}(\mathbf{F}^{t-1}), \quad (3)$$

where α can either be a scalar, e.g. $\alpha = \frac{1}{2}$, or a matrix, $\alpha \in \mathbb{R}^{h_f \times w_f}$. Note that in both cases, α can be automatically learned together with the rest of the network without any additional supervision. For special cases where $t = 1$, we simply assume $g_{st}(\mathbf{F}^{t-1})$ equals to $\mathbf{F}^t$. It's simple but it works well in practice, see supp for more discussion.

Discussion: Note that we can easily extend our model by aggregating features from frames that are further away or even from future frames in offline settings. Also, FTM can be extended for multiple feature maps at different scales, which can potentially further improve the performance.

3.2.2 Loss functions on scene attribute annotation:

Given the annotated data sets D^{r} that consist of N^{r} videos, we define our overall loss function $\mathcal{L}^{\mathrm{r}}_{\text{sup}}$ as:

$$\sum_{i,t}^{N^{\mathrm{r}},T} \mathrm{BCE}(\Theta^t_{\mathrm{b},i}, \hat{\Theta}^t_{b,i}) + \mathrm{CE}(\Theta^t_{\mathrm{m},i}, \hat{\Theta}^t_{m,i}) + \ell_1(\Theta^t_{\mathrm{c},i}, \hat{\Theta}^t_{c,i}),$$

where CE denotes cross-entropy and BCE represents binary cross-entropy. We further denote the ground-truth layout parameters of the t-th frame in the i-th video sequence in the data set as $\hat{\Theta}^t_{*,i}$. Note that we discretize continuous variables for the regression task – each variable is discretized into K=100 bins by convolving a Dirac delta function centered at Θ_{c} with a Gaussian of fixed variance [38].

3.3. Model Input

As suggested by [38], the form of our model input $\mathbf{x}$ has a large impact on our model. Although a perspective RGB image is a natural representation of $\mathbf{x}$, it does not work well in top-view layout estimations. Instead, [38] proposes to convert each single perspective image to semantic top-view with pixel-level depth and semantic segmentation. In our case, since we are working on videos, there are more aspects that can be explored, like long-term temporal smoothness. In this section, we propose to exploit both context and global information in videos to improve the representation of $\mathbf{x}$.

Local information: single frame Following [38], given a single perspective image, together with its semantic segmentation, CNN-predicted dense depth, and camera intrinsics, we could obtain a top-view image of the road by (1) back-projecting all the road pixels into a 3D point cloud and (2) projecting all the 3D points onto the x-y plane. We use this mapping to transfer the semantic class probability distribution of each pixel from perspective- into top-view. We call it *bev* and denote it as $\mathbf{x} \in \mathbb{R}^{H \times W \times C}$, where C is the number of semantic classes. $H = 128$ and $W = 64$ pixels, which relates to 60×30 meters in the point cloud. See Fig. 6 for some examples of the *bev* map obtained with single images.

3.3.1 Context information

Considering that incorporating context cues proves to be very beneficial in many scene understanding tasks, unlike [38] that only includes $C = 4$ classes (i.e. road, sidewalk, lane, crosswalk), we propose to further encode an object class in $\mathbf{x}$ (so $C = 5$). Specifically, we observe that traffic participants, i.e. vehicles like cars, are commonly present in driving scenarios and can be very useful for layout estimation, e.g. side-faced cars are informative in terms of predicting the existence of or even the distance to side-road. One naive way to incorporate context cues is to follow what is done for other classes to directly map the pixel-level probabilities of objects to top-view. However, we observe that this naive

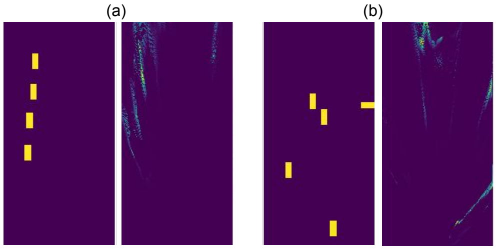

Figure 4: Two examples comparing the box representation and the point cloud representation of the cars. We can observe four frontal-faced cars in both (a) and (b) and one side-faced car in (b).

method would lead to diffused representations[3] (See Fig. 4 for examples) and can be very noisy. Instead, we propose to utilize the bounding box representation. Specifically, we apply existing 3D object detectors [17, 18] on our videos and map the detection results into top-view. In order to reduce the impact of angle predictions, we further propose to categorize angles into two types, frontal-faced and side-faced. As long as the predicted yaw angle is within $[-\frac{\pi}{4}, \frac{\pi}{4}]$ or $[\frac{3\pi}{4}, \frac{5\pi}{4}]$ w.r.t. camera z-axis (forward direction), we assume this predicted object is frontal-faced. Otherwise, it is side-faced (Fig. 4).

3.3.2 Global information from video sequence

As can be viewed in individual frame examples in Fig. 6, there are clearly some artifacts with *bev*, e.g. the V-shape in the bottom and the missing details where pixels are far away. This is in essence due to the inherent limitation of single-view 3D reconstruction in terms of the resolution for far away points in the scene.

Top-view map from COLMAP: When the input is an entire video sequence, we are able to address the above mentioned artifacts in an individual frame by generating a more complete 3D reconstruction as well as the subsequent *bev*. Specifically, we make use of state-of-the-art SfM and MVS algorithms from COLMAP [23, 24], as well as the semantic segmentation in 2D image, in order to generate the top-view map (denoted as *bev-col*) from a video. The pipeline is illustrated in Fig. 5. We first perform 3D semantic segmentation on the dense point cloud returned from COLAMP. We project each 3D point into the images as per the visibility map to obtain the semantic candidates from the 2D semantic segmentation; a simple winner-take-all strategy is applied to determine the final semantic label for each point (Fig. 5(b)). The 3D semantic segmentation allows us to extract the 3D points for the road part, to which we fit a 2D plane via RANSAC (Fig. 5(c)). Finally, we can generate the *bev-col* by cropping a rectangular box on the 2D road plane according to the camera pose and our pre-defined top-view image size and resolution; the 3D points on the plane are converted into the pixels in the image (Fig. 5(d)). After obtaining the global plane parameters, we utilize the camera pose to transform the global plane parameters to the local parameters with respect to the camera. Note that in a well-calibrated case where the global plane can be the same as local plane, the global lane estimation process can be skipped. See Fig. 6 for two more examples. Here, one can see that the dense correspondence between two *bev-col* can be trivially obtained given the known position and orientation of each *bev-col* in the 2D road plane. Such mapping is transferred to the correspondence on the feature map and fed into FTM (Sec. 3.2.1). Here, we do not handle the mapping of dynamic objects, but find FTM to still work well [4]. Finally, we also note that the absolute scale of the 3D reconstruction here can be retrieved from either GPS or camera-ground height in driving scenes.

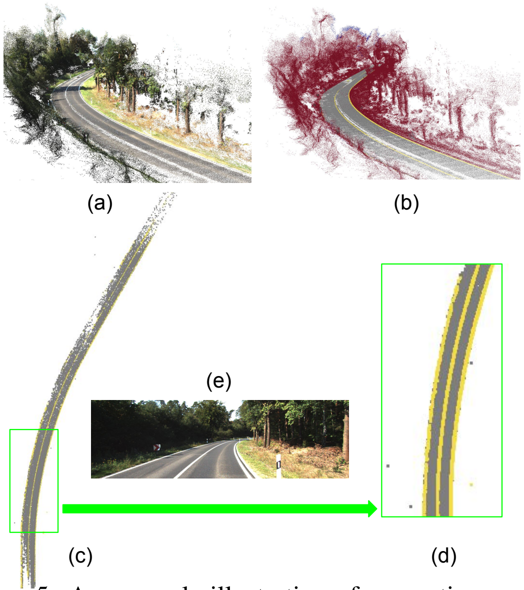

Figure 5: An example illustration of generating semantic *bev-col*. (a)(b) 3D scene reconstruction and semantic segmentation. (c) road extraction and plane fitting. (d) *bev-col* for the view (e).

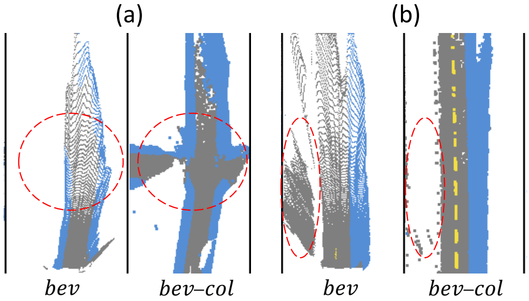

Figure 6: Two examples of *bev* and *bev-col* on the background classes obtained from individual frame and COLMAP. Red circles highlight the differences.

Discussion: Note that in our case, rather than presenting

[3] This is mainly due to the small size of the objects, making it less robust to semantic and depth prediction noise when mapping to top-view.

[4] Having access to GT masks for dynamic objects may help to improve performance, but in their absence, even our current handling of dynamic objects yields promising results.

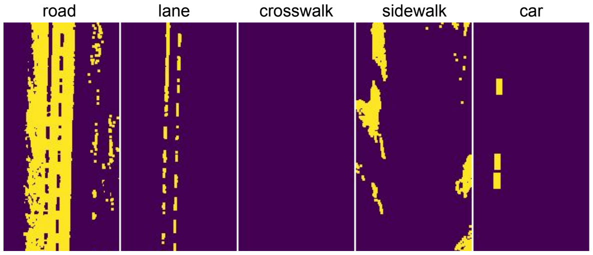

Figure 7: An example visualizing the layers in our $bev\text{-}final$.

what can be actually seen in the current frame, our *bev-col* obtained with COLMAP models what really exists. For instance, a sideroad that is far away or occluded can hardly be noticed in *bev* generated from an individual frame only due to the lack of observations in perspective view. In contrast, *bev-col* generated with global information is able to recover the sideroad properly as long as we have observations in this video (see comparison in Fig. 6(a)).

Final input: Our final input incorporates local information from individual frame, global information from videos and object-level context information. Note that local and global cues can be mutually informative, e.g. geometry based 3D reconstruction might be inaccurate or incomplete in some challenging cases like texture-less regions, but in such cases the top-view information from CNN-predicted depth could be more informative towards the layout prediction (See comparison in Fig. 6(b)). In summary, our proposed top-view semantic map $bev\text{-}final$ is in $\mathbb{R}^{H\times W\times C}$, where $C=5$ and the layers represents four background classes and car; We overlay *bev-col* on top of *bev* to fuse them together. Specifically, for a binary mask B with $B_{i,j}{=}1$ if $bev\text{-}col_{i,j}{\neq}0$, we have $bev\text{-}final = bev\text{-}col{\odot}B + bev{\odot}(1{-}B)$, where $\odot$ indicates element-wise product. See an example visualization of each layer in Fig. 7.

4. Experiments

In this section, we demonstrate the effectiveness of our proposed augmented input representation and LSTM/FTM by conducting several experiments on two data sets.

Datasets: We validate our ideas on two data sets, KITTI [8] and NuScenes [22]. Similarly, we utilize the annotated data in [38], which includes around 17000 annotations for KITTI [8] and about 1200 annotations for NuScenes [22] in terms of scene layout annotation. To train our segmentation network, we further have 1243 images for training and 382 for testing on KITTI. We refer the readers to [38, 37] for more details of data sets and implementation details can be found in supplementary.

Evaluation metrics: Since our output space consists of three types of predictions and involves both discrete and continuous variables, we follow the metrics defined in [38] and summarize them below.

	NuScenes [22]			
Method	Accu.-Bi. ↑	Accu.-Mc. ↑	MSE ↓	IOU ↑
BEV	.846	.485	.073	.217
BEV-C	.856	.471	.069	.211
BEV-J	**.872**	.486	.036	.230
BEV-J-O	.858	**.543**	.027	**.313**
+LSTM	.859	.536	**.023**	.311
+LSTM+FTM	**.863**	**.547**	**.023**	**.328**

Table 1: Results on road scene layout estimation on [22].

As for binary variables Θ_{b} (like the existence of crosswalk on the main road or left side-walk) and for multi-class variables Θ_{m} (like the number of lanes on your left), we report the prediction accuracy as Accu.-Bi $= \frac{1}{14}\sum_{k=1}^{14}[p_k = \Theta_{\mathrm{b}k}]$ and Accu.-Mc $= \frac{1}{2}\sum_{k=1}^{2}[p_k = \Theta_{\mathrm{m}k}]$. More importantly, since we observe that the binary classes are extremely biased, we also report the F1 score on Θ_{b} which better informs the overall performance. Specifically, F1.-Bi $= \frac{1}{14}\sum_{k=1}^{14} 2\times\frac{p_k\times r_k}{p_k+r_k}$, where p_k and r_k are the precision and recall rate on k-th variable of binary attributes. For regression variables we use the mean square error (MSE).

In addition, we also report Intersection-over-Union (IoU) as a overall performance evaluation metric. Specifically, we assume that we can render four-class semantic top-view maps with either the predicted results or the ground-truth annotations. Then we report the average IoU score over all test images. More details on IoU are presented in supplementary.

4.1. Evaluations on Global and Context Cues

In this section, we would like to explore the impact of $bev\text{-}final$. To this end, we aim to validate the effectiveness of our proposed context and global information on road layout prediction. We use the basic model without LSTM and FTM, unless otherwise specified. Details for this basic model can be found in *basic* of Fig. 8.

Baselines: We compare our proposed method with several competing methods presented in [38]. The **RGB** method takes a single perspective RGB image as input and trains a model [12] that directly outputs predictions for scene attributes. The **RGB+D** method shares the same model architecture w.r.t. **RGB** but outputs both per-pixel depth map and attribute predictions. Both **BEV** and the SOTA **H-BEV-DA** output scene attributes with *bev* as input, which means they are all online methods. The differences lie in: 1) the former uses only *bev* from real images during training while the latter utilizes additional simulated *bev*; 2) the former uses *basic* model and the later has a hybrid model as in [38]. Note that +GM means a spatial-temporal graphical model is added as post-processing.

Our proposals: Apart from baselines, we further propose the following methods with different input representations:

	KITTI [8]				
Method	Accu.-Bi. ↑	Accu.-Mc. ↑	MSE ↓	F1.-Bi ↑	IOU ↑
RGB [27]	.811	.778	.230	.176	.327
RGB [27]+D	.818	.819	.154	.109	.334
BEV [38]	.820	.797	.141	.324	.345
BEV [38]+GM	.831	.802	.136	.344	.357
H-BEV-DA[38]+GM	.834	.831	**.134**	.435	.404
BEV-C	.826	.779	.175	.456	.317
BEV-J	.840	.832	.135	.458	**.416**
BEV-J-O	.831	**.837**	.142	.494	.388
+LSTM	**.845**	.825	**.134**	**.537**	.382
+LSTM+FTM	**.842**	**.841**	**.134**	**.534**	**.405**

Table 2: Main results on road scene layout estimation on KITTI [8]. We can see that: 1) The global (BEV-C, BEV-J) and context (BEV-J-O) information do benefits the single image performance compared to their single image competitors. 2) Introducing LSTM and FTM can further improve the accuracy compared to single image method.

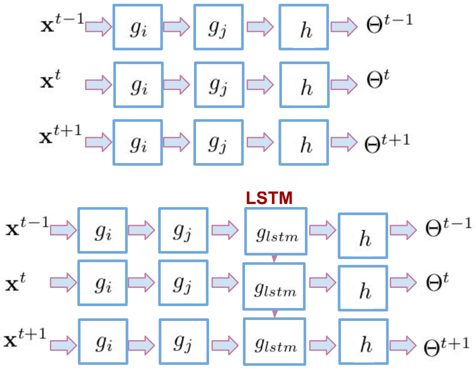

Figure 8: Illustrations of baseline model architectures. We denote the upper one as $basic$ and lower one as $blstm$.

- **BEV-COLMAP (BEV-C):** We propose bev-col with global information from videos, as described in Sec. 3.3.2. Once we get the bev-col, we feed it to the $basic$ model.
- **BEV-JOINT (BEV-J):** We combine the bev and bev-col together and obtain the input for BEV-J. Then we feed the joint input to the $basic$ model.
- **BEV-JOINT-OBJ (BEV-J-O):** As described in Sec. 3.3.1, we further add object information and obtain bev-$final$ as the input for BEV-J-O. Again, a $basic$ model is learned.

Quantitative results: Tab. 2 summarizes our main results on KITTI [8]. We have the following observations. Firstly, comparing BEV-C to BEV, we can see that obtaining global information from 3D point clouds generated with the entire video sequence is better than relying on individual depth and semantic segmentation results, e.g. large improvement in F1 score. This improvement might come from higher quality of $bevinit$, especially in terms of both higher accuracy in regions that are far away from the camera and more consistent results. Second, if we compare BEV-J to BEV-C and BEV, we can see that combining them together can further boost the performance for all attributes, e.g. 1.4%, 3.3% and 0.04 improvement for binary, multiclass and regression tasks. More importantly, we can improve the IoU by around 10%. One reason might be that the input from BEV-C focuses more on texture regions that can be both close by and far away while input from BEV keeps more details for all close by regions. They are mutual beneficial in terms of representing scene layout in top-view. Finally, once adding object information (BEV-J-O), we can further boost the F1 score by about 5%. In conclusion, we can observe a clear improvement when adding individual components.

We also report partial results on NuScenes [22] in Tab. 1 and refer the readers to supplementary for complete ones. As for NuScenes [22], we again observe that combining individual and global information provides better results than either one in isolation. Also, adding context information further introduces improvements on almost all tasks.

	KITTI [8]			
Method	Accu.-Bi. ↑	Accu.-Mc. ↑	MSE ↓	F1.-Bi. ↑
BEV-C	.826	.779	.175	.456
*-denseC	.826	.783	.148	.496
*-C-O	.836	.812	.146	.507
*+LSTM	.831	.812	.136	.360

Table 3: Ablation study on single image based road layout prediction. We abbreviate BEV as $*$ in this table.

Ablation study: Since the main improvement of the performance boost comes from information of the full video sequence as well as objects, in this section, we conduct ablation studies on: 1) the impact of the quality of top-view maps generated with full video sequences on models, and 2) the impact of the representation of objects on models. Tab. 3 summarizes our results, where *-denseC means we change 3D reconstruction parameter of COLMAP to generate much denser bev-col as inputs to train our model. And *-C-O means we combine the input from BEV-C with object information and learn to predict the road attributes.

As can be seen in this table, denser bev-col can definitely improve the performance for all tasks and measurements, which indicates that there can be more room to boost prediction accuracy by improving the quality of bev-col. Additionally, further performance boosts can be obtained with object information, which again proves that context information can be very useful in our layout prediction task.

Qualitative results: We show several qualitative results in Fig. 9. We can see from the examples that our model successfully describes a diverse set of road scenes. Moreover, our predictions are generally consistent w.r.t. object detection results.

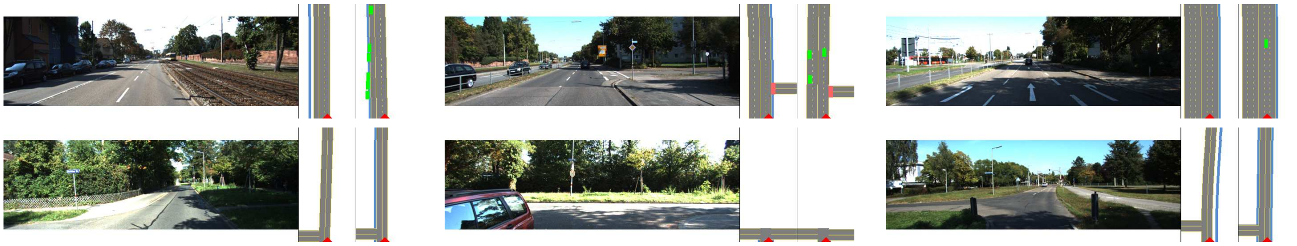

Figure 9: Qualitative results of full model on individual frames from KITTI. Each example shows perspective RGB, ground truth and predicted semantic top-view (including the object bounding box as green rectangular), respectively.

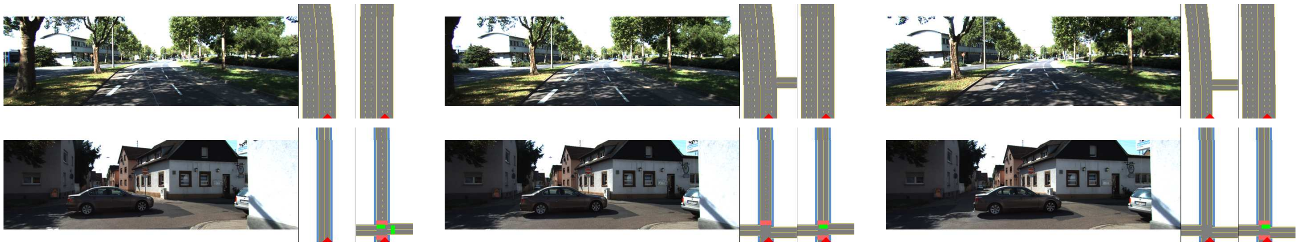

Figure 10: Qualitative results comparing BEV-J-O and full model in **consecutive** frames of two example sequences from KITTI. We visualize for each view the perspective RGB image, prediction from BEV-J-O and from full model, respectively.

	KITTI [8]	
Method	seman.↓	temp.↓
BEV [38]	2.65	3.99
H-BEV-DA [38]+GM	1.77	1.93
BEV-J-O	1.26	2.78
+LSTM+FTM	1.26	1.96

Table 4: Main results on consistency measurements.

4.2. Evaluations on LSTM and FTM

We further demonstrate the effectiveness of LSTM and FTM in Tab. 2. More specifically, +LSTM model is illustrated in Fig. 8 ($blstm$ part) and +LSTM+FTM is our full model. Both models receive $bev\text{-}final$ as input.

As can be seen in this table, with the help of LSTM, we can increase performance on the binary and regression tasks. Note that we can further boost the F1 score by at least 4% compared to single image methods. More importantly, our full model, or +LSTM+FTM, provides the best overall performance. As highlighted in the table, it almost always performs the best or second best among all tasks and measurements. Even compared to previous SOTA that is trained with both real and simulated data and includes a spatio-temporal graphical model as post-processing, we can clearly observe that the proposed method provides better results, specifically for F1 score. In conclusion, we demonstrate the effectiveness of FTM and LSTM in terms of improving prediction accuracy. We also report the BEV+LSTM in Tab. 3. We can observe that adding LSTM ($blstm$) can improve the performance even with bev inputs, which demonstrates the effectiveness of temporal information on locally obtained top-view representations. Such trend is also shared on NuScenes in Tab. 1.

Consistency: Since we are working in videos, we also analyze the consistency in predictions. To this end, we utilize the following metrics as suggested in [38] and refer the readers to supplementary material for more details:

- *Semantic consistency:* we report the average conflicts in attribute predictions.
- *Temporal consistency:* we also report the average number of changes in our prediction.

We show in Tab. 4 quantitative results for the temporal as well as semantic consistency metrics defined on the two data sets. Compared to the single image baseline, we can observe significant improvement from the proposed BEV-J-O in consistency for both data sets. Also, if we compare with video-based existing work, combined with the prediction accuracy shown in Tab.2, we can see that the proposed method outperforms previous SOTA, e.g. boost the accuracy while not sacrificing temporal smoothness. Results and our analysis on NuScenes [22] can be found in the supplementary.

Finally, we visualize qualitative results of consecutive frames in two test sequences from KITTI in Fig. 10. As can be seen, our model successfully enforces temporal smoothness. We can also observe more consistent predictions, *e.g.*, width of side-road and delimiter width, with the help of the LSTM and FTM. Again, our predictions are consistent w.r.t. object detection results.

5. Conclusion

In this work, we present a scene understanding framework to estimate the parametric road layout in each view in a video. In constrast to the case with only single image input, we propose to make use of the temporal information in videos by leveraging LSTM/FTM model, context information as well as global 3D reconstruction, altogether leading to very promising results.

References

[1] Iro Armeni, Ozan Sener, Amir R. Zamir, Helen Jiang, Ioannis Brilakis, Martin Fischer, and Silvio Savarese. 3D Semantic Parsing of Large-Scale Indoor Spaces. In *CVPR*, 2016.

[2] Samuel Rota Bulò, Lorenzo Porzi, and Peter Kontschieder. In-Place Activated BatchNorm for Memory-Optimized Training of DNNs. In *CVPR*, 2018.

[3] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. In *ECCV*, 2018.

[4] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2625–2634, 2015.

[5] Christoph Feichtenhofer, Axel Pinz, and Richard P Wildes. Spatiotemporal multiplier networks for video action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4768–4777, 2017.

[6] Philipp Fischer, Alexey Dosovitskiy, Eddy Ilg, Philip Häusser, Caner Hazırbaş, Vladimir Golkov, Patrick Van der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. *arXiv preprint arXiv:1504.06852*, 2015.

[7] Andreas Geiger, Martin Lauer, Christian Wojek, Christoph Stiller, and Raquel Urtasun. 3D Traffic Scene Understanding from Movable Platforms. *PAMI*, 2014.

[8] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets Robotics: The KITTI Dataset. *International Journal of Robotics Research (IJRR)*, 2013.

[9] Clément Godard, Oisin Mac Aodha, and Gabriel J. Brostow. Unsupervised Monocular Depth Estimation with Left-Right Consistency. In *CVPR*, 2017.

[10] Ruiqi Guo and Derek Hoiem. Beyond the line of sight: labeling the underlying surfaces. In *ECCV*, 2012.

[11] Saurabh Gupta, James Davidson, Sergey Levine, Rahul Sukthankar, and Jitendra Malik. Cognitive Mapping and Planning for Visual Navigation. In *CVPR*, 2017.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *CVPR*, 2016.

[13] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[14] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial Transformer Networks. In *NIPS*, 2015.

[15] Lars Kunze, Tom Bruls, Tarlan Suleymanov, and Paul Newman. Reading between the Lanes: Road Layout Reconstruction from Partially Segmented Scenes. In *International Conference on Intelligent Transportation Systems (ITSC)*, 2018.

[16] Iro Laina, Vasileios Belagiannis Christian Rupprecht, Federico Tombari, and Nassir Navab. Deeper Depth Prediction with Fully Convolutional Residual Networks. In *3DV*, 2016.

[17] Buyu Li, Wanli Ouyang, Lu Sheng, Xingyu Zeng, and Xiaogang Wang. Gs3d: An efficient 3d object detection framework for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1019–1028, 2019.

[18] Peiliang Li, Xiaozhi Chen, and Shaojie Shen. Stereo r-cnn based 3d object detection for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7644–7652, 2019.

[19] Buyu Liu and Xuming He. Multiclass semantic video segmentation with object-level active inference. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4286–4294, 2015.

[20] Chenxi Liu, Alexander G. Schwing, Kaustav Kundu, Raquel Urtasun, and Sanja Fidler. Rent3D: Floor-Plan Priors for Monocular Layout Estimation. In *CVPR*, 2015.

[21] Gellért Máttyus, Shenlong Wang, Sanja Fidler, and Raquel Urtasun. HD Maps: Fine-grained Road Segmentation by Parsing Ground and Aerial Images. In *CVPR*, 2016.

[22] NuTonomy. The NuScenes data set. `https://www.nuscenes.org`, 2018.

[23] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4104–4113, 2016.

[24] Johannes L Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision*, pages 501–518. Springer, 2016.

[25] Samuel Schulter, Menghua Zhai, Nathan Jacobs, and Manmohan Chandraker. Learning to Look around Objects for Top-View Representations of Outdoor Scenes. In *ECCV*, 2018.

[26] Mike Schuster and Kuldip K Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681, 1997.

[27] Ari Seff and Jianxiong Xiao. Learning from Maps: Visual Common Sense for Autonomous Driving. *arXiv:1611.08583*, 2016.

[28] Sunando Sengupta, Paul Sturgess, L̀ubor Ladický, and Philip H. S. Torr. Automatic Dense Visual Semantic Mapping from Street-Level Imagery. In *IROS*, 2012.

[29] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*, pages 568–576, 2014.

[30] Shuran Song, Fisher Yu, Andy Zeng, Angel X. Chang, Manolis Savva, and Thomas Funkhouser. Semantic Scene Completion from a Single Depth Image. In *CVPR*, 2017.

[31] Shuran Song, Andy Zeng, Angel X. Chang, Manolis Savva, Silvio Savarese, and Thomas Funkhouser. Im2Pano3D: Extrapolating 360 Structure and Semantics Beyond the Field of View. In *CVPR*, 2018.

[32] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhudinov. Unsupervised learning of video representations using lstms. In *International conference on machine learning*, pages 843–852, 2015.

[33] Deqing Sun, Stefan Roth, and Michael J Black. Secrets of optical flow estimation and their principles. In *2010 IEEE*

computer society conference on computer vision and pattern recognition, pages 2432–2439. IEEE, 2010.

[34] Joseph Tighe, Marc Niethammer, and Svetlana Lazebnik. Scene Parsing with Object Instances and Occlusion Ordering. In *CVPR*, June 2014.

[35] Shubham Tulsiani, Richard Tucker, and Noah Snavely. Layer-structured 3D Scene Inference via View Synthesis. In *ECCV*, 2018.

[36] Tuan-Hung Vu, Wongun Choi, Samuel Schulter, and Manmohan Chandraker. Memory warps for learning long-term online video representations. *arXiv preprint arXiv:1803.10861*, 2018.

[37] Ziyan Wang, Buyu Liu, Samuel Schulter, and Manmohan Chandraker. A dataset for high-level 3d scene understanding of complex road scenes in the top-view. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshop*, June 2019.

[38] Ziyan Wang, Buyu Liu, Samuel Schulter, and Manmohan Chandraker. A parametric top-view representation of complex road scenes. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[39] SHI Xingjian, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *Advances in neural information processing systems*, pages 802–810, 2015.

[40] Dan Xu, Wei Wang, Hao Tang, Hong Liu, Nicu Sebe, and Elisa Ricci. Structured Attention Guided Convolutional Neural Fields for Monocular Depth Estimation. In *CVPR*, 2018.

[41] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid Scene Parsing Network. In *CVPR*, 2017.

[42] Xizhou Zhu, Yujie Wang, Jifeng Dai, Lu Yuan, and Yichen Wei. Flow-guided feature aggregation for video object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 408–417, 2017.

[43] Xizhou Zhu, Yuwen Xiong, Jifeng Dai, Lu Yuan, and Yichen Wei. Deep feature flow for video recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2349–2358, 2017.