Monocular BEV Perception of Road Scenes via Front-to-Top View Projection

Wenxi Liu, Qi Li, Weixiang Yang, Jiaxin Cai, Yuanlong Yu, Yuexin Ma, Shengfeng He, Senior Member, IEEE, and Jia Pan

Abstract—HD map reconstruction is crucial for autonomous driving. LiDAR-based methods are limited due to expensive sensors and time-consuming computation. Camera-based methods usually need to perform road segmentation and view transformation separately, which often causes distortion and missing content. To push the limits of the technology, we present a novel framework that reconstructs a local map formed by road layout and vehicle occupancy in the bird’s-eye view given a front-view monocular image only. We propose a front-to-top view projection (FTVP) module, which takes the constraint of cycle consistency between views into account and makes full use of their correlation to strengthen the view transformation and scene understanding. In addition, we apply multi-scale FTVP modules to propagate the rich spatial information of low-level features to mitigate spatial deviation of the predicted object location. Experiments on public benchmarks show that our method achieves various tasks on road layout estimation, vehicle occupancy estimation, and multi-class semantic estimation, at a performance level comparable to the state-of-the-arts, while maintaining superior efficiency.

Index Terms—BEV perception, autonomous driving, segmentation.

I. INTRODUCTION

W
ITH the rapid progress of autonomous driving technologies, there have been many recent efforts on related research topics, e.g., scene layout estimation [1], [2], [3], [4], [5], [6], [7], 3D object detection [8], [9], [10], [11], [12], [13], vehicle behavior prediction [14], [15], [16], [17], and lane detection [18], [19], [20].

Among these tasks, high-definition map (HD map) reconstruction is fundamental and critical for perception, prediction, and planning of autonomous driving. Its major issues are concerned with the estimation of a local map including the road layout as well as the occupancies of nearby vehicles in the 3D world. Existing techniques rely on expensive sensors like LiDAR and require time-consuming computation for cloud point data.

In addition, camera-based techniques usually need to perform road segmentation and view transformation separately, which causes distortion and the absence of content.

To push the limits of the technology, our work aims to address the realistic yet challenging problem of estimating the road layout and vehicle occupancy in top-view, or bird’s-eye view (BEV), given a single monocular front-view image (see Fig. 1). However, due to the large view gap and severe view deformation, understanding and estimating the top-view scene layout from the front-view image is an extremely difficult problem, even for a human observer. The same scene has significantly different appearances in the images of bird’s-eye-view and the front-view. Thus, parsing and projecting the road scenes of front-view to top-view require the ability to fully exploit the information of the front-view image and innate reasoning about the unseen regions.

Traditional methods (e.g., [21], [22]) focus on investigating the perspective transformation by estimating the camera parameters and performing image coordinate transformation, but gaps in the resulting BEV feature maps caused by geometric warping lead to poor results. Recent deep learning-based approaches [23], [24] mainly rely on the hallucination capability of deep Convolutional Neural Networks (CNNs) to infer the unseen regions between views. In general, instead of modeling the correlation between views, these methods directly leverage CNNs to learn the view projection models in a supervised manner. These models require deep network structures to propagate and transform the features of the front-view through multiple layers to spatially align with the top-view layout. However, due to the locally confined receptive fields of convolutional layers, it is difficult to fit a view projection model and identify the vehicles of small scales.

To address these concerns, we derive a novel framework to estimate the road layout and vehicle occupancies from the

Fig. 1. Given a front-view monocular image, we propose a front-to-top view projection module consisting of a cycle structure that bridges the features of the front-view and top-view in their respective domains, as well as a cross-view transformer that correlates views attentively to facilitate the road layout estimation.
top-view given a single monocular front-view image. To handle the large discrepancy between views, we present a Front-to-Top View Projection (FTVP) module in our network, which is composed of two sub-modules: Cycled View Projection (CVP) module bridges the view features in their respective domains and Cross-View Transformer (CVT) correlates the views, as shown in Fig. 1. Specifically, the CVP projects views using a multi-layer perceptron (MLP), which overcomes the standard information flow passing through convolutional layers, and involves the constraint of cycle consistency to retain the features relevant for view projection. In other words, transforming front-views to top-views requires a global spatial transformation over the visual features. However, standard CNN layers only allow local computation over feature maps, and it takes several layers to obtain a sufficiently large receptive field. On the other hand, fully connected layers can better facilitate the front-to-top view projection. CVT explicitly correlates the features of the views before and after projection obtained from CVP, which can significantly enhance the features after view projection. We involve a feature selection scheme in CVT, which leverages the associations of both views to extract the most relevant information.

However, at the core of our network model, the low-resolution encoded top-view features may result in the spatial deviation of the predicted object location, because of the information loss of heavily compressed features. To alleviate this issue, low-level visual features need to be exploited to refine the upsampled features. To bridge the low-level visual features from the front-view and the upsampled features, we propose applying multi-scale FTVP modules to project and propagate the low-level features. Concretely, the front-view features from the downsampling stream are first projected and transformed to top-view ones via the FTVP modules. Then all the projected features are concatenated with the corresponding top-view features of the upsampling stream. Moreover, to further enhance the representation of top-view features from different scales, deep supervision is employed to supervise the segmentation heads.

In comprehensive experiments, we demonstrate that our front-to-top view projection module can effectively elevate the performance of road layout and vehicle occupancy estimation. For both tasks, we compare our model against the state-of-the-art methods on public benchmarks and demonstrate that our model is superior to all the other methods. It is worth noting that, for estimating vehicle occupancies, our model achieves a significant advantage over the competing methods by at least 34.8% in the KITTI 3D Object dataset and by at least 46.4% in the Argoverse dataset. Furthermore, we also validate our framework by predicting semantic segmentation results with more than seven classes in both the Argoverse and NuScenes datasets. We show that our method surpasses the latest off-the-shelf methods, which reflects the generality and rationality of our proposed model. Last but not least, we show that our framework is able to process 1024 × 1024 images at 25 FPS using a single Titan XP GPU, and it is applicable for real-time reconstruction of a panorama HD map.

Overall, the contributions of our paper are:

- We propose a novel front-to-top view projection module, denoted as FTVP, which consists of a cycled view projection structure and a cross-view transformer. Specifically, FTVP leverages the cycle consistency between the frontal views and Bird’s Eye View to establish their correlation and strengthen view projection.
- We also employ multi-scale FTVP modules that project and propagate front-view features of different scales to refine the representation of top-view features, providing multi-scale details for a more precise estimation of object locations.
- On public benchmarks, we demonstrate that our model achieves tasks such as road layout, vehicle occupancy, and multi-class semantic estimation at a performance level comparable to state-of-the-art methods, while maintaining superior efficiency.

Note that a preliminary version of this work was presented as [25]. This submission extends [25] the methodology and experiment in the following aspects. First, we renovate our model by adjusting the structure of the front-to-top view projection module in which we rectify the projected features by utilizing the raw front-view features for supplementing visual information. This improves the expressive ability of the final top-view features and enhances the robustness of the module structure (Section III-B). Second, our previous model [25] tends to cause the spatial deviation in vehicle occupancy prediction and road layout estimation, due to upsampling the low-resolution features to higher-resolution ones. To address this concern, we re-design the framework by directly projecting multi-scale features from the downsampling stream via the multi-scale FTVP modules to strengthen the upsampling process using the skip connection (Section III-C). With the above improvements, our proposed model performs better than preliminary version on all datasets. Especially on the KITTI 3D Object dataset, our method achieves 4.9% and 17.5% improvements in terms of mIOU and mAP. Third, we extend our model to address multi-class segmentation problems with more than seven classes, which shows our model is able to handle scene parsing for diverse objects not limited to vehicles and road (Section IV-C). Finally, we evaluate and discuss the effectiveness of our proposed framework and conduct additional experiments including using different backbones to compare against the state-of-the-arts for multi-class BEV segmentation the Argoverse and NuScenes datasets (Sections IV-C and IV-D). We also analyze the performance of our model for recognizing and localizing small or distant objects (Section IV-E). Furthermore, we demonstrate that our proposed network has superior efficiency over prior works, and it can achieve real-time performance on a single Titan XP GPU.

In the remainder of this paper, we first review the related work in Section II. We will then describe the overall framework and introduce the front-to-top view projection module in Section III. In Section IV, the experimental results are demonstrated and discussed. Finally, we summarize our work in Section V.

II. RELATED WORK

In this section, we survey the related literature on road layout estimation, vehicle detection, and street view synthesis on top-view representation. We also introduce the recent progress of transformers on vision tasks.
BEV-Based Road Layout Estimation: Most road scene parsing works focus on semantic segmentation [26], [27], [28], [29], [30], [31], [32], while there are a few attempts that derive top-view representation for road layout [1], [6], [7], [33], [34], [35], [36], [37]. Among these methods, Schulter et al. [6] propose estimating an occlusion-reasoned road layout on top-view from a single color image by depth estimation and semantic segmentation. VED [37] proposes a variational autoencoder (VAE) model to predict road layout from a given image, but without attempting to reason about the unseen layout from observation. PON [3] directly transforms features from images to 3D space and finally to bird’s-eye-view (BEV) grids. STA [38] and Stitch [39] aggregate the temporal information to produce the final segmentation results through the transformation module. I2M [40] and GitNet [41] formulate map generation from an image as a set of sequence-to-sequence translations. Most related to our work, [42] presents a unified model to tackle the task of road layout (static scene) and traffic participant (dynamic scene) estimations from a single image. Unlike prior methods, we propose a novel model to explicitly model the large view projection that learns the spatial information to produce high-quality BEV segmentation results.

In contrast to our task, some works leverage six different views of images to perform BEV segmentation. VPN [43] presents a cross-view semantic segmentation by transforming and fusing the observation from multiple cameras. Lift-Splat [44] learns a depth distribution over pixels to lift camera images to a 3D point cloud, and project the latter into BEV using camera geometry. FIERY [45] combines the perception, sensor fusion, and prediction components of traditional autonomous driving. CVT [46] uses a camera-aware cross-view attention mechanism that equips each camera-view feature with positional embeddings. BAEFormer [47] proposes the bi-directional interaction that uses unshared attention maps to update BEV and image features simultaneously.

Monocular Image-Based 3D Vehicle Detection: Many monocular image-based 3D vehicle detection techniques have been developed (e.g., [8], [13], [48], [49], [50], [51], [52]). Several methods handle this problem by mapping the monocular image to the top-view. For instance, [10] proposes mapping a monocular image to the top-view representation and treats 3D object detection as a task of 2D segmentation. 2D-Lift [53] proposes the BEV feature transform layer to transform 2D image features to the BEV space, which exploits depth maps and 3D point cloud. BirdGAN [9] also leverages adversarial learning for mapping images to bird’s-eye-view. BAEFormer [54] first introduces sequential temporal modeling into multi-view 3D object detection and applies temporal self-attention. PETR [55] converts the multi-view 2D features to 3D position-aware features by adding the 3D position embedding. PETRv2 [56] extends the PETR with temporal modeling and the support for multi-task learning.

View Transformation and Synthesis: Traditional methods (e.g. [21], [22], [57]) have been proposed to handle the perspective transformation in traffic scenes. With the progress of deep learning-based methods, [23] proposes a pioneering work to generate the bird’s-eye-view based on the driver’s view. They treat cross-view synthesis as an image translation task and adopt a GAN-based framework to accomplish it. Due to the difficulty in collecting annotation for real data, their model is trained from video game data. [58] focuses exclusively on warping camera images to BEV images without performing any downstream tasks such as object detection. Recent attempts [24], [59] on view synthesis aim to convert aerial images to street view images, or vice versa. Compared with these works, our purpose is quite different and requires not only the implicit view projection from front-view to top-view, but also the estimation of road layout and vehicle occupations under a unified framework.

Transformer for Vision Tasks: Convolutional neural networks(CNNs) are regarded as the most basic component in vision tasks. However, with the recent success of the Transformer [60], its ability to explicitly model pairwise interactions for elements in a sequence has been leveraged in many vision tasks, such as image classification [61], object detection [62], [63], activity recognition [64], and image super-resolution [65]. ViT [66] first applies the Transformer framework with non-overlapping image patches in the vision task. Swin Transformer [67] jointly leverages the inner transformer block and outer transformer block to enhance information exchange. Swin Transformer [68] obtains a larger receptive field by shifting the windows over the image. PVT [69] proposes a spatial reduction attention to reduce computational complexity. These models all show even more impressive modeling capabilities and achieve excellent performance. Inspired by these transformer-based models, our proposed cross-view transformer attempts to establish the correlation between the features of views. In addition, we incorporate a feature selection scheme along with the non-local cross-view correlation scheme, significantly enhancing the representativeness of the features.

III. OUR PROPOSED METHOD

A. Network Overview

The goal of our work is to estimate the road scene layout and vehicle occupancies on the bird’s-eye view in the form of semantic masks given a monocular front-view image.

Our network architecture is shown in Fig. 2. The front-view image $I$ is first passed through the encoder based on ResNet [70] as the backbone network to extract multi-scale visual features. After that, our proposed front-to-top view projection modules project the encoded features of different scales and propagate them to the decoder to produce the top-view semantic mask $M$. In the following subsections, we will elaborate the details of our front-to-top view projection module and its multi-scale deployments on various scales.

B. Front-to-Top View Projection Module

Due to the large gap between front-views and top-views, a large proportion of image content may be lost during view projection, so the traditional view projection techniques lead to defective results. To this end, the hallucination ability of CNN-based methods has been exploited to address the problem,
but the patch-level correlation of both views is not trivial to model within deep networks.

In order to strengthen the view correlation while exploiting the capability of deep networks, we introduce a front-to-top view projection module into our framework, which enhances the extracted visual features for projecting front-view to top-view. The structure of our proposed FTVP module is shown in Fig. 2, and it is composed of two parts: cycled view projection and cross-view transformer.

**Cycled View Projection (CVP):** Since the features of front-views are not spatially aligned with those of top views due to their large gap, we follow [43] and deploy an MLP structure consisting of two fully-connected layers to project the features of front-view to top-view, which can overtake the standard information flow of stacking convolution layers. As shown in Fig. 2, X and \( X' \) represent the feature maps before and after view projection, respectively. Hence, the view projection can be achieved by:

\[
X' = \mathcal{F}_{\text{MLP}}(X),
\]

where \( X \) refers to the features extracted from the ResNet backbone.

However, such a simple view projection structure cannot guarantee that the information of front-views is effectively delivered. Here, we introduce a cycled self-supervision scheme to consolidate the view projection, which projects the top-view features back to the domain of front-views. As illustrated in Fig. 2, \( X'' \) is computed by cycling \( X' \) back to the front-view domain via the same MLP structure, i.e., \( X'' = \mathcal{F}'_{\text{MLP}}(X') \).

To guarantee the domain consistency between \( X \) and \( X'' \), we incorporate a cycle loss, i.e., \( \mathcal{L}_{\text{cycle}} \), as expressed below:

\[
\mathcal{L}_{\text{cycle}} = \|X - X''\|_1. \tag{1}
\]

The benefits of the cycle structure are two-fold. First, similar to the cycle consistency-based approaches [71], [72], cycle loss can innate improve the representativeness of features, since cycling back the top-view features to the front-view domain will strengthen the connection between both views. Second, when the discrepancy between \( X \) and \( X'' \) cannot be further narrowed, it represents the missing pixels that are not needed for BEV segmentation and thus \( X'' \) retains the crucial cues for view projection. In Fig. 3, we show two examples by visualizing the features of the front-view and top-view. Specifically, we visualize them by selecting the representative channels of the feature maps (i.e., the 7th and 92nd for two examples of Fig. 3) and aligning them with the input images. As observed, \( X \) and \( X'' \) are similar, but quite different from \( X' \) due to the domain difference. We can also observe that, via cycling, \( X'' \) concentrates more on the road and the vehicles. \( X \), \( X' \), and \( X'' \) will be fed into the cross-view transformer.

**Cross-View Transformer (CVT):** The main purpose of CVT is to correlate the features before view projection (i.e., \( X \)) and the features after view projection (i.e., \( X' \)) to strengthen the latter ones. Since \( X'' \) contains the substantial information of the front-view for view projection, it can be used to further enhance the features as well. As illustrated in Fig. 4, CVT can be roughly divided into two schemes: the cross-view correlation scheme that explicitly correlates the features of views to achieve an attention map \( W \) to strengthen \( X' \) and the feature selection scheme that extracts the most relevant information from \( X'' \).

Specifically, \( X, X', \) and \( X'' \) serve as the key \( K (K \equiv X) \), the query \( Q (Q \equiv X') \), and the value \( V (V \equiv X'') \) of CVT, respectively. In our model, the dimensions of \( X, X', \) and \( X'' \) are...
are set as the same. \(X'\) and \(X\) are both flattened into patches, and each patch is denoted as \(x'_i \in X'(i \in \{1, \ldots, hw\})\) and \(x_j \in X(j \in \{1, \ldots, hw\})\), where \(hw\) refers to the width of \(X\) times its height. Thus, the relevance matrix \(R\) between any pairwise patches of \(X\) and \(X'\) can be estimated, i.e., for each patch \(x'_i\) in \(X'\) and \(x_j\) in \(X\), their relevance \(r_{ij}(\forall r_{ij} \in R)\) is measured by the normalized inner-product:

\[
    r_{ij} = \frac{x'_i \cdot x_j}{||x'_i|| \cdot ||x_j||}.
\]  

With the relevance matrix \(R\), we create two vectors \(W = \{w_i, \forall i \in \{1, \ldots, hw\}\}\) and \(H = \{h_i, \forall i \in \{1, \ldots, hw\}\}\) based on the maximum value and the corresponding index for each row of \(R\), respectively:

\[
    w_i = \max_j r_{ij}, \forall r_{ij} \in R, \quad (3)
\]

\[
    h_i = \arg \max_j r_{ij}, \forall r_{ij} \in R. \quad (4)
\]

Each element of \(W\) implies the degree of correlation between each patch of \(X'\) and all the patches of \(X\), which can serve as an attention map. Each element of \(H\) indicates the index of the most relevant patch in \(X\) with respect to each patch of \(X'\).

To this end, we introduce a feature selection scheme \(F_{fs}\). With \(H\) and \(X''\), \(F_{fs}\) can produce new feature maps \(T (T = \{t_i, \forall i \in \{1, \ldots, hw\}\})\) by retrieving the most relevant and compact features from \(X''\) according to the indices of the largest values in the relevance matrix \(R\):

\[
    t_i = F_{fs}(X'', h_i), \forall h_i \in H, \quad (5)
\]

where \(F_{fs}\) retrieves the feature vector \(t_i\) from the \(h_i\)-th position of \(X''\).

In overall, the motivation of our CVT design in the key, query, and value is to leverage the frontal view to enhance the BEV features. In a typical transformer design, \(X'\) should have been employed as the value \(V\). Yet, the view projection during training is imperfect, leading to inaccurate \(X'\). Thus, we choose to employ \(X''\) to approach the role of \(X'\), but \(X''\) is not perfect either. Recall that, both \(X\) and \(X''\) represent the frontal-view features, except that \(X\) contains its complete yet excess information, while \(X''\) retain the cues for view projection only. Therefore, as shown in Fig. 4, we allow \(X\) to concatenate with \(T\) to supplement each other. Afterwards, the concatenated features will be weighted by the attention map \(W\) and finally aggregated with \(X\) via a residual structure. To sum up, the process can be formally expressed as:

\[
    X_{out} = X' + F_{conv}(\text{Concat}(X, T)) \odot W, \quad (6)
\]

where \(\odot\) denotes the element-wise multiplication and \(F_{conv}\) refers to a convolutional layer with \(3 \times 3\) kernel size. \(X_{out}\) is the final output of CVT and will then be passed to the decoder network to produce the segmentation mask of the top-view.

C. Multi-Scale FTVP Modules

As mentioned above, our proposed FTVP module can acquire the enhanced top-view features \(X_{out}\) to generate refined segmentation results. However, the resolution of \(X_{out}\) is too low to completely retain the information on visual details, which tends to cause spatial deviation of object location from top-views.

To remedy the shortcomings, we introduce multi-scale FTVP modules to bridge the low-level features and the upsampled top-view features in the decoder. As shown in Fig. 2, the extracted front-view features \(X_i\) of the \(i\)-th scales are passed through the FTVP modules for view projection in order to gain the corresponding top-view features \(\hat{X}_i\). In practice, we employ the three innermost encoded features for projection to avoid unnecessary computational burden. Next, the projected features \(\hat{X}_i\) and the top-view features \(X'_i\) in the upsampling stream are concatenated, i.e., \(X'_i = \text{Concat}(\hat{X}_i, X'_i)\), so that the projected features \(\hat{X}_i\) manage to deliver rich low-level information to the decoder. Finally, as illustrated in Fig. 2, we employ the deep supervision scheme that supervises the segmentation results generated by the top-view features \(X'_i\) of different scales. Specifically, the multi-scale segmentation losses are calculated for \(X'_i\) at different scales via additional segmentation heads.

We visualize the segmentation masks generated from the top-view features of different scales in Fig. 5. When the resolutions of the features increase, the false negatives (i.e., red areas) and false positives (i.e., green areas) of estimating the locations of vehicles gradually decrease due to the involvement of low-level information. In addition, for the features of 3rd, 4th, and 5th scales (i.e., the three highest resolution intermediate feature maps in the encoder), their produced results are almost the same, so it is sufficient to employ three FTVP modules only.

D. Loss Function

Overall, the loss function of our framework is defined as:

\[
    L = \sum_{i=0}^{n} L_{seg}^i + \lambda L_{cycle}, \quad (7)
\]

where \(L_{seg}\) is the cross-entropy loss for supervising the segmentation results produced by the features of the \(i\)-th scale. The main objective of the network is to narrow the gap between the
predicted semantic mask and the ground-truth mask. However, real-world scenes often include small objects such as vehicles and pedestrians, which lead to class-imbalance problems. Thus, in practice, we use the square root of the inverse class frequency to weight unbalanced loss for small classes. In addition, \( \lambda \) is the balance weight of the cycle loss and it is set as 0.001.

IV. EXPERIMENTAL RESULTS

To evaluate our proposed model, we conduct several experiments over a variety of challenging scenarios and compare our results against state-of-the-art methods on public benchmarks. We also perform extensive ablation experiments to delve into our network structure.

A. Implementation Details

We implement our framework using Pytorch on a workstation with a single NVIDIA Titan XP GPU card. In most experiments, the input images are normalized to \( 1024 \times 1024 \) by bilinear interpolation and the output size is \( 256 \times 256 \). We adopt the pretrained ResNet-18 or ResNet-50 [70] without bottleneck layers as our encoder backbone. Specifically, after the encoder, we employ two maxpooling layers and \( 3 \times 3 \) convolution layers to reduce the computation overhead of the FTVP module. Thus, the resolution size of the smallest feature map coming out of the encoder is \( 8 \times 8 \) for an input image of \( 1024 \times 1024 \) resolution. In practice, we use the features with the dimensions of \( 8 \times 8, 16 \times 16, \) and \( 32 \times 32 \) to feed the multi-scale FTVP modules. In the CVP, MLP contains two fully-connected layers and ReLU activation, and each input of CVT utilizes one convolutional layer with kernel size \( 1 \times 1 \). The number of channels of MLP is set as 128 for ResNet-18 and 256 for ResNet-50, respectively. The decoder is composed of five convolutional blocks and upsampling layers with the spatial resolution increasing by a factor of 2. In the decoder, the projected features \( \hat{X}_i \) from the FTVP modules and the top-view features \( X'_i \) from the \( 2 \times \) upsampling stream are concatenated along channel dimension to generate the new top-view features \( \hat{X}'_i \) with \( 16 \times 16 \) and \( 32 \times 32 \) and the output size is \( 256 \times 256 \). Last, deep supervision loss is calculated at the segmentation heads of each scale, which is a \( 3 \times 3 \) convolutional layer.

The network parameters are randomly initialized, and we adopt the Adam optimizer [73] and use a mini-batch size of 6. The initial learning rate is set to \( 1 \times 10^{-3} \) and is decayed by a poly learning rate policy where the initial learning rate is multiplied by \( (1 - \frac{iter}{total \_ iter})^{0.9} \) after each iteration. In practice, it takes 50 epochs to converge our model, and our model can run in real-time (25 FPS) on our single-GPU platform.

B. Datasets

We evaluate our approach on three datasets: KITTI [74], Argoverse [75], and Nuscenes [76]. For performance assessment, we adopt the mean of Intersection-over-Union (mIOU) and Average Precision (mAP) as the evaluation metrics.

KITTI: Since KITTI has not provided sufficient annotation for road layout or vehicles to be used in our task, we generally follow the practice of [42], in which the results are categorized in the following datasets. For comparison with state-of-the-art 3D vehicle detection approaches, we evaluate performance on the KITTI 3D object detection (KITTI 3D Object) split of Chen et al. [8], i.e., 3712 training images and 3769 validation images. The KITTI Odometry dataset is used to evaluate the road layout, the annotation of which comes from the Semantic KITTI dataset [77]. In addition to the previous two datasets, we evaluate the performance on the KITTI Raw split used in [6], i.e., 10156 training images and 5074 validation images. Since its ground-truths are produced by registering the depth and semantic segmentation of Lidar scans that are not sufficiently dense, we apply image dilation and erosion to produce better ground-truth annotations.

Argoverse: Furthermore, we also compare methods on the dataset Argoverse (Version 1.1) following the settings of [3], which provides a high-resolution semantic occupancy grid and vehicle detection in top-view for evaluating the spatial layout of roads and vehicles, with 6723 training images and 2418 validation images. For multiple semantic categories, we follow the evaluation protocol of [3], which selects 1 static class (i.e., Drivable) and 7 dynamic object classes (i.e., Vehicle, Pedestrian, Large Vehicle, Bicycle, Bus, Trailer, and Motorcycle) to conduct comparison experiments.

Nuscenes: The NuScenes dataset consists of 1000 twenty-second video clips collected in Boston and Singapore. It provides 3D bounding box annotations for 23 object classes including walkway, car, trailer, and so on. In our experiments, we also follow the protocol of [3]. We choose 4 static classes (i.e.,
Drivable, Pedestrian crossing, Walkway, and Carpark) and 10 dynamic object classes (i.e., Car, Truck, Bus, Trailer, Construction vehicle, Pedestrian, Motorcycle, Bicycle, Traffic cone, and Barrier) for evaluation. In addition, we also use their training and validation split, i.e., 28,008 training images and 5,981 validation images.

C. Comparison Methods and Performance Evaluation

Comparison Methods: For evaluation, we compare our model with some of the state-of-the-art methods for road layout estimation and vehicle occupancy estimation, including VED [37], MonoLayout [42], VPN [43], Mono3D [8], OFT [10], and PYVA [25]. Among these methods, Mono3D [8] and OFT [10] are specifically used to detect vehicles in top-view. For the quantitative results, we follow the ones reported in [42]. For MonoLayout [42], we compare with their latest online reported results, which are generally better than the ones reported in their original paper. VPN [43] originally adopted multiple views from different cameras to generate the top-view representation. We adapt their model for single-view input and then retrain it using the same training protocol for our task. Likewise, VED [37] is retrained for the benchmarks of road layout estimation, and we obtain comparable or better results than the ones reported in [42]. PYVA [25] is our preliminary version, which correlates the features of the views before and after projection and performs feature selection.

In addition, we also compare our model against the state-of-the-art methods for multi-class semantic estimation. These methods include VED [37], VPN [43], PON [3], OFT [10], 2D-Lift [53], Stitch [39], STA [38], and 12M [40]. Most of these works focus on estimating the top-view semantic segmentation maps, and the rest use detection methods. We follow the single frame quantitative results reported in [53], [39], and [40] for the ArgoVerse and Nuscenes datasets.

Road Layout Estimation: To evaluate the performance of our model on the task of road layout estimation, we compare our model against VED [37], MonoLayout [42], VPN [43], and PYVA [25] on the ArgoVerse Road and ArgoVerse Odometry datasets. Note that, since we post-process the ground-truth annotations of ArgoVerse Raw, we retrain all the comparison methods under the same training protocol. The comparison results are demonstrated in Table I. Additionally, we also compare them on ArgoVerse Vehicle, as shown in Table II. As observed, in these three benchmarks, our model shows advantages over the competitors in both mIOU and mAP. In addition, compared with our preliminary version (i.e., PYVA [25]), the performance also exceeds about +1%, especially in mAP, because the spatial deviation has been rectified using multi-scale FTVP. Examples are shown in Fig. 6. Note that the ground-truths may contain noise, since they are converted from the Lidar measurements. Even so, our approach can still produce satisfactory results.

Vehicle Occupancy Estimation: Compared with road layout estimation, estimating vehicle occupancies is a more challenging task, since the scales of vehicles vary and there exist mutual occlusions in the scenes. For evaluation, we perform comparison experiments on the KITTI 3D Object and ArgoVerse Vehicle benchmarks against VED [37], Mono3D [8], OFT [10], MonoLayout [42], VPN [43], and PYVA [25]. The results are shown in Tables II and III. In Table III, our model demonstrates superior performance against the comparison methods. Since KITTI 3D Object contains several challenging scenarios, most comparison methods barely obtain 30% mIOU and 50% mAP, while our model gains 40.69% and 59.07%, respectively, which shows at least 34.8% and 28.7% improvement over prior methods. Our model achieves significant advantage over PYVA by at least 4.9% and 17.5% in mIOU and mAP, brought by the structure improvement of FTVP as well as the multi-scale FTVP modules, which beeps up the model and corrects the spatial deviation. For the evaluation on ArgoVerse Vehicle in Table II, our model outperforms others by a large margin, i.e., at least 46.4% and 29.9% boost over the comparison methods in mIOU and mAP, respectively. On the first three rows of Fig. 7, we show the examples on vehicle occupancy estimation on KITTI 3D Object. For the challenging cases with multiple vehicles parked on the sides of roads, our model can still perform well. The last four rows of Fig. 7 show examples of the joint estimation for roads and vehicles on ArgoVerse, and we highlight the advantages of our results.

Multi-Class Semantic Estimation: We extend our model to address multi-class semantic segmentation problems on ArgoVerse with 8 classes and on Nuscenes with 14 classes. Compared

| Methods       | KITTI Raw | KITTI Odometry |
|---------------|-----------|----------------|
|               | mIOU (%)  | mAP (%)        |
| VED [37]      | 58.41     | 66.01          |
| VPN [43]      | 59.58     | 79.07          |
| MonoLayout [42]| 66.02     | 75.73          |
| PYVA [25]     | 68.34     | 80.78          |
| Ours          | 68.42     | 81.78          |

| Methods       | Road     | Vehicle |
|---------------|----------|---------|
|               | mIOU (%) | mAP (%) |
| VED [37]      | 72.84    | 78.11   |
| VPN [43]      | 71.07    | 86.83   |
| MonoLayout [42]| 73.25    | 84.56   |
| PYVA [25]     | 76.51    | 87.21   |
| Ours          | 78.14    | 88.05   |

| Methods       | mIOU (%) | mAP (%) |
|---------------|----------|---------|
| VED [37]      | 20.45    | 22.59   |
| Mono3D [8]    | 17.11    | 26.62   |
| OFT [10]      | 25.24    | 34.69   |
| VPN [43]      | 16.80    | 35.54   |
| MonoLayout [42]| 30.18    | 45.91   |
| PYVA [25]     | 38.79    | 50.26   |
| Ours          | 40.69    | 59.05   |

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
Fig. 6. Comparison results of road layout estimation on KITTI Odometry.

TABLE IV

| Model | Backbone | Resolution | Driva. | P.C. | Walkway | Carpark | Car | Truck | Bus | Train. | C.V. | Ped. | Motor. | Bicyc. | T.C. | Barrier | Mean |
|-------|----------|------------|--------|------|---------|--------|----|-------|-----|--------|-----|------|-------|-------|-----|---------|------|
| VED [37] | VGG16 | 384x768 | 54.7 | 12.0 | 20.7 | 13.5 | 8.8 | 0.2 | 0.0 | 7.4 | 0.0 | 0.0 | 0.0 | 4.0 | 0.0 | 8.7 |
| VPN [43] | ResNet18 | 464x800 | 58.0 | 27.3 | 29.4 | 12.9 | 25.5 | 17.3 | 20.0 | 16.6 | 4.9 | 7.1 | 5.6 | 4.4 | 1.4 | 10.8 | 17.5 |
| OPT [10] | ResNet18 | 900x1600 | 62.4 | 30.9 | 34.5 | 23.5 | 34.7 | 17.4 | 23.2 | 18.2 | 3.7 | 1.2 | 6.6 | 4.6 | 1.1 | 12.9 | 19.6 |
| Ours | ResNet18 | 640x640 | 73.4 | 30.3 | 36.5 | 29.0 | 30.2 | 16.4 | 26.0 | 11.5 | 0.5 | 3.9 | 7.4 | 7.8 | 3.3 | 15.0 | 20.8 |
| Ours | ResNet18 | 768x768 | 74.0 | 31.2 | 37.0 | 29.0 | 29.5 | 17.8 | 26.5 | 12.5 | 0.5 | 4.1 | 8.7 | 8.8 | 3.3 | 15.2 | 21.3 |
| Ours | ResNet18 | 1024x1024 | 74.1 | 31.8 | 37.2 | 31.1 | 31.4 | 18.1 | 27.9 | 12.9 | 0.7 | 4.0 | 8.9 | 10.0 | 3.3 | 17.2 | 22.0 |
| PON [3] | ResNet50 | 600x800 | 60.4 | 28.0 | 31.0 | 18.4 | 24.7 | 16.8 | 20.8 | 16.6 | 12.3 | 8.2 | 7.0 | 9.4 | 5.7 | 8.1 | 19.1 |
| 2D-Lift [53] | ResNet50 | 900x1600 | 62.3 | 31.8 | 37.3 | 25.2 | 37.4 | 18.7 | 24.8 | 16.4 | 4.7 | 3.4 | 7.9 | 7.2 | 3.9 | 13.6 | 21.0 |
| Stitch [39] | ResNet50 | 448x800 | 71.7 | 27.2 | 34.9 | 32.1 | 32.9 | 15.3 | 23.1 | 15.2 | 4.4 | 5.8 | 6.3 | 6.8 | 4.8 | 17.2 | 21.4 |
| STA [38] | ResNet50 | 900x1600 | 71.1 | 31.5 | 32.0 | 28.0 | 34.6 | 18.1 | 22.8 | 11.4 | 10.0 | 7.4 | 7.1 | 14.6 | 5.8 | 10.8 | 21.8 |
| I2M [40] | ResNet50 | 900x1600 | 72.6 | 36.3 | 32.4 | 30.5 | 37.4 | 24.5 | 32.5 | 15.5 | 13.8 | 8.7 | 8.1 | 15.1 | 7.4 | 15.1 | 25.0 |
| I2M* [40] | ResNet50 | 900x1600 | 71.2 | 32.8 | 33.4 | 34.9 | 35.8 | 19.5 | 33.3 | 14.0 | 1.5 | 8.1 | 6.7 | 16.5 | 10.0 | 15.6 | 23.8 |
| Ours | ResNet50 | 640x640 | 74.7 | 32.2 | 37.4 | 28.4 | 32.2 | 18.4 | 30.4 | 12.2 | 0.7 | 5.0 | 9.2 | 8.5 | 4.0 | 16.9 | 22.2 |
| Ours | ResNet50 | 768x768 | 74.8 | 32.2 | 38.2 | 29.2 | 33.0 | 20.2 | 31.5 | 12.2 | 1.0 | 5.1 | 10.3 | 10.6 | 3.8 | 18.9 | 22.9 |
| Ours | ResNet50 | 1024x1024 | 78.7 | 34.0 | 39.2 | 30.5 | 34.4 | 20.4 | 31.9 | 17.2 | 0.8 | 6.3 | 10.4 | 11.3 | 6.2 | 16.5 | 23.9 |

Note that all methods take a single frame as input for fair comparison. The best results are highlighted in bold and the second best results are underlined. * represents our reproduced results.

with road layout estimation and vehicle occupancy estimation, multi-class semantic estimation is a more challenging task because it must accommodate the broad roads and a wide variety of vehicles with different scales. Specifically, for the tasks of road layout and vehicle occupancy, it produces the BEV outcomes with the resolution of 256 × 256. For the multi-class semantic estimation, following the previous methods (e.g., PON [3] and I2M [40]), we apply the same BEV lattice grid with 0.25 m resolution that spans a region of [−25,25]m in lateral direction and [1,50]m in longitudinal direction. In these tasks, all the comparison models produce the BEV map using the resolution of 200 × 200. We conduct comparison experiments against VED [37], VPN [43], PON [3], Stitch [39], and I2M [40] for the Argoverse dataset. For Nuscenes, we introduce more comparison methods including OPT [10], 2D-Lift [53], and STA [38]. The comparison results are depicted in Tables IV and V respectively. For I2M, we provide the original results reported in [40], and we also reproduce its results denoted as I2M*. Specifically, I2M* is trained based on its official released training codes.\(^1\) The batch size is set as 6 and we train it for 40 epochs. The initial learning rate is set to \(5 \times 10^{-5}\) and is decayed by a polynomial
learning rate policy where the initial learning rate is multiplied by $(1 - \frac{\text{epoch}}{\text{total_epoch}})^{0.9}$ after each epoch. The reproduced model will also be used in our visualization and ablation study. As observed, our model shows superior performance over other methods. Among the models using ResNet-18 as backbone, our model performs significantly better than previous methods by at least 2% mIOU with similar input resolution. Among the ones with ResNet-50 as backbone, our model also outperforms PON and Stitch using smaller resolution input images. Compared to I2M, the mIOU of our method is slightly lower, but there is a significant difference for each class. Specifically, our method outperforms I2M for static classes, and underperforms it for dynamic classes. This is probably because I2M has a much larger model capacity than ours. For the Construction Vehicle (C. V.) class, our performance is severely worse than I2M, which is attributed to the limited number of training samples, thereby resulting in confusion with the Truck class. However, the disparities between both methods are marginal when it comes to smaller objects such as motorcycle, barrier, and pedestrian. They demonstrate comparable performance in capturing these smaller targets. On the other hand, our model offers significant improvement over the state-of-the-art methods on Argoverse for the “trailer” class. In Fig. 8, we illustrate the examples of multi-class semantic estimation on Nuscenes. We observe that our model performs better than other methods on position and shape estimation for both static and dynamic classes.

### D. Ablation Study

To delve into our network structure, we conduct several ablation experiments for the front-to-top view projection module.

**Front-to-Top View Projection Module:** Recall that our front-to-top view projection module consists of CVP and CVT. Specifically, CVP can be divided into the MLP and the cycle structure. CVT can be decomposed into a cross-view correlation scheme and a feature selection scheme. In the following, we will investigate the necessity of these modules based on the dataset KITTI 3D Object in Table VI.

First, the baseline is the vanilla encoder-decoder network using the same encoder and decoder as our model. Then we insert the MLP structure to the baseline. As shown in Table VI, it obviously improves the effectiveness of view projection. Next, we add a cross-view correlation scheme into the network, which measures the correlation of $X$ and $X'$ and applies it as the attention map to enhance $X'$. As observed, with the involvement

---

**Table VI**

Effectiveness of the Front-to-Top View Projection Module

| Model Structure | mIOU (%) | mAP (%) |
|-----------------|----------|---------|
| Baseline        | 22.31    | 34.58   |
| + MLP           | 27.42    | 37.44   |
| + Cross-view Correlation | 35.03 | 46.33   |
| + Cycle Structure | 35.54    | 47.29   |
| + Feature Selection | 39.97 | 54.33   |
| + Multi-scale PTVPs | 37.83    | 55.02   |
| + Deep Supervision | 40.69    | 59.05   |

---

**Table V**

Comparison Results on Argoverse

| Model | Driva. | Veh. | Ped. | Lat. veh. | Bicyc. | Bar. | Trail. | Motor. | Mean. |
|-------|--------|------|------|-----------|--------|------|-------|--------|-------|
| VED [37] | 62.9 | 14.0 | 1.0 | 3.9 | 0.0 | 12.3 | 1.3 | 0.0 | 11.9 |
| VPN [43] | 64.9 | 23.9 | 6.2 | 9.7 | 0.9 | 3.0 | 0.4 | 1.9 | 13.9 |
| PON [3] | 65.4 | 31.4 | 7.4 | 11.1 | 3.6 | 11.0 | 0.7 | 5.7 | 17.0 |
| Stitch [39] | 79.8 | 28.2 | 4.8 | 11.4 | 0.0 | 22.5 | 1.1 | 0.4 | 18.5 |
| I2M [40] | 75.9 | 35.8 | 5.7 | 14.9 | 3.7 | 30.2 | 12.2 | 2.6 | 22.6 |
| Ours-R18 | 83.3 | 36.5 | 4.2 | 13.7 | 2.4 | 0.9 | 60.7 | 1.1 | 25.2 |
| Ours-R50 | **83.5** | **37.5** | **4.9** | **12.0** | **4.1** | **0.8** | **60.9** | **3.1** | **25.9** |

We follow the quantitative results of prior works reported in [39], [40]. Ours-R18 and ours-R50 refer to the models using ResNet-18 and ResNet-50 as backbone.
Fig. 8. Examples of static and dynamic object location estimation on Nuscenes.
of the cross-view correlation scheme, the performance is significantly boosted. We then introduce the cycle structure as well as the cycle loss into the network, in which $X'\prime\prime$ will be fed into the cross-view correlation scheme. Finally, we insert the feature selection scheme, which further strengthens the performance of the model.

In Fig. 9, we show several examples of the ablation study on the FTVP module. The examples are selected from the results of KITTI 3D Object dataset. With the addition of structures (e.g., MLP, cross-view correlation, cycle structure, and feature selection) in the FTVP module, our model can effectively extract the masks of the individual vehicles, remove noises, and refine their shapes.

Cross-View Transformer: We validate different input combinations of $K$, $Q$, $V$ for CVT. We demonstrate the results in Table VII. For all test cases, the query (i.e., $Q$) is assigned to a feature after view projection $X'$. As the most trivial case, we use $X'$ as $K$ and $V$ of CVT as well, which self-correlates all the non-local patches of $X'$. Since $X'$ may lose some information via view projection, CVT does not perform well. Because both $K$ and $V$ are assigned to $X$ or $X''$, it involves the features before view projection, but $X$ contains richer information than $X''$, which leads to better performance. Moreover, with $X$ and $X''$ corresponding to $K$ and $V$, the substantial information for view projection is implicitly introduced by $X''$ to strengthen the model. More specifically, using $X$ as the key is better for generating a precise relevance embedding, while applying $X''$ as the value encourages the involvement of most relevant features, leading to the best results.

Multi-Scale FTVP Modules: Based on the model with a single FTVP module, we upgrade the network by integrating the multi-scale FTVP modules. As observed in Table VI, the upgraded model improves mAP while decreasing mIOU. This is perhaps because the features of different scales deliver rich, detailed information but their learning lacks guidance. To address this, we employ the deep supervision scheme. As shown, our model achieves the best performance (40.69% mIOU and 59.07% mAP). Note that, for reference, we also add deep supervision for the model with a single FTVP module, and its results are almost unchanged (39.48% versus 39.97% on mIOU and 54.78% versus 54.53% on mAP). In the case of a single FTVP, all the decoded features are upsampled from the deepest top-view features with a small resolution, so the multi-scale details can hardly be delivered to the decoder and thus the deep supervision alone cannot strengthen the model. Moreover, we show the representative visualization results with respect to the multi-scale FTVP modules in Fig. 10. It is observed that the spatial deviation effect can be reduced in the presence of FTVPs.

To show the advantage of employing three FTVPs only, we compare our model against other variants with deep supervision, including the models with one or more than three FTVPs. The comparison results in terms of mIOU, mAP, and FPS are demonstrated in Table VIII. A single FTVP is more efficient but lacks multi-scale spatial information, while FTVP modules

| Model Structure | mIOU (%) | mAP (%) | FPS |
|------------------|---------|---------|-----|
| Naive Conv       | 39.99   | 58.40   | 23  |
| Window FTVPs     | 39.09   | 57.37   | 19  |
| 1 FTVP           | 39.48   | 54.78   | 35  |
| 4 FTVPs          | 39.69   | 57.70   | 21  |
| 5 or 6 FTVPs     | N/A     | N/A     | N/A |
| Ours (3 FTVPs)   | 40.69   | 59.05   | 25  |

N/A represents out of memory during training.
Fig. 10. We illustrate that, while our model will cause spatial deviation with a single FTVP module, the proposed multi-scale FTVP modules can propagate the rich spatial information to alleviate this problem. In addition, we highlight the pixels that are discrepancies between our estimation and the ground truth. The red and green pixels represent False Negatives and False Positives, respectively. Please zoom in for a better view.

Table IX

| Dist. (m) | Model  | Driva. | Walkway | Ped. | Motor | Bicyc. | Mean |
|----------|--------|--------|---------|------|-------|--------|------|
|          | I2M [40] |        |         |      |       |        |      |
| [0, 50)  | w/o FTVPs | 85.3   | 35.8    | 18.4 | 14.5  | 35.9   | 35.6 |
|          | Ours   | 88.8   | 45.6    | 15.6 | 22.8  | 24.3   | 37.8 |
| [50, 100)| w/o FTVPs | 77.4   | 38.5    | 11.7 | 11.7  | 15.2   | 29.0 |
|          | Ours   | 81.5   | 44.9    | 5.4  | 11.5  | 8.6    | 27.9 |
| [100, 150]| w/o FTVPs | 67.4   | 34.6    | 5.2  | 5.1   | 3.9    | 20.3 |
|          | Ours   | 71.4   | 38.8    | 2.9  | 4.6   | 3.1    | 19.5 |
| [150, 200]| w/o FTVPs | 63.6   | 29.7    | 0.9  | 0     | 3.2    | 16.8 |
|          | Ours   | 66.0   | 31.4    | 1.4  | 0.9   | 0.7    | 13.8 |

Fig. 11. We illustrate the examples of pedestrians from different distances to demonstrate the effectiveness of multi-scale FTVPs.

E. Analysis on Small or Distant Targets

To evaluate the ability of identifying and localizing distant or small objects, we measure the mIOU of several representative classes (e.g., Pedestrians, Motorcycle, and Bicycle) and overall performance for all classes in Nuscenes for different distance ranges, i.e., [0,50), [50,100), [100,150), [150,200] meters, as illustrated in Table IX. As observed, our model generally outperforms I2M in close ranges and underperforms it in the range of [150,200]. The advantage of I2M mainly comes from its deployed on the shallow layers lack semantic information and demand a large number of parameters and a lot of computational resources for computing higher-resolution features. In addition, based on three FTVP modules, we added two FTVPs with $12 \times 12$ local window or two naive $3 \times 3$ convolutional layers at the third and fourth scales to increase the spatial information, costing less computational consumption. They both use higher-resolution front-view features $X$ as input and output the top-view features $X'$. However, local window FTVPs and convolutional layers are both local operations without considering global information. Thus, these variants all lead to sub-optimal performance, so we choose to use three FTVPs in our network, enabling real-time inference.
cascaded transformer blocks, which provide exceptional ability to capture distance objects, but also bring in a large amount of parameters. In contrast to the model without FTVPs (i.e., [25]), the capability of identifying distance objects can be obviously enhanced. We visualize examples in Fig. 11. On the first row, we observe that the positions of pedestrians are refined with FTVPs and thus False Positives can be reduced. However, at excessive distances, these categories suffer from positioning failure, with little improvements. We have observed that, in the second row of Fig. 11, the distant pedestrian stands besides the vehicle can hardly be localized. Even with FTVPs, it still results in spatial deviation and unsatisfactory performance.

In addition, to delve into each module of our model and analyze their effect on small or distant objects, we conduct ablation study on the Nuscenes dataset in Table X. First of all, we ablate our model by removing CVP, concatenation with $X$, feature selection, applying softmax operator, and FTVPs. As observed, CVP and FTVPs severely influence the performance on small objects, since CVP controls the view projection and FTVPs provide multi-scale details. Taking Fig. 12 for examples, the distant pedestrian and vehicles can be detected with assistance of the CVP module. Softmax operator and the removal of feature selection also lead to sub-optimal results, due to the introduction of redundant information into $X''$. Besides, even without $X$ concatenation (i.e., with $X''$ alone), our model can still perform well on small objects like Motorcycle and Bicycle, which means that $X''$ carries the important information. In addition, we attempt to incorporate the absolute position embedding (Abs. PE) [66], the conditional position embedding (Con. PE) [78]. Since our CVT only selects the most relevant information, position embedding plays a minor role. Last, we also assess the model with dice loss, which well recognizes the class of Pedestrians, but their performance on the other classes is undesired, since the training samples of Pedestrians are much more than Motorcycles and Bicycles.

Moreover, we illustrate the successful and failure cases of localizing the distant objects as well as their corresponding attention maps in Fig. 13. On the first and second rows, when the activated areas of attention maps cover the target objects and their surroundings, it is likely that our model can successfully perceive the pedestrians or motorcyclists at a distance. On the
Fig. 14. We montage the estimated road layout from the image sequences of Argoverse to produce three panorama HD maps (on the right side of the figure) containing the road layout and vehicle occupancies. Specifically, (a) and (b) respectively showcase the reconstruction of an HD map captured by a vehicle traversing a straight road with two intersections. Whereas, (c) illustrates the reconstructed scene of a vehicle making a right turn at an intersection.

third row, only the small central region with the car is activated, resulting in that the pedestrian behind the vehicle cannot be recognized. As we can conclude, the intensity of attention maps from front-view features correlates to the results of BEV segmentation for small objects.

F. Panorama HD Map Generation

We showcase the application of our model on the Argoverse dataset for generating a panorama HD map via stitching the road layout estimation given the consecutive front-view images. The generated HD map is shown in Fig. 14, highlighting the potential of our approach for generating panorama HD maps.

| Model   | Backbone | Resolution | FPS | Params | FLOPs | Mean |
|---------|----------|------------|-----|--------|-------|------|
| VED [37] | VGG16    | 384×768    | 77  | 45.61M | 140.58G | 8.7  |
| VPN [43] |          | 464×800    | 100 | 18.29M | 37.61G  | 17.5 |
| Ours    | ResNet18  | 640×640    | 72  | 20.64M | 18.33G  | 20.8 |
| Ours    | ResNet18  | 768×768    | 71  | 21.37M | 26.55G  | 21.3 |
| Ours    | ResNet18  | 1024×1024  | 25  | 24.43M | 48.08G  | 22.0 |
| PON [3]  |          | 600×800    | 17  | 38.64M | 96.12G  | 19.1 |
| Stitch [39] |          | 448×800    | 6   | 52.32M | 589.25G | 21.4 |
| I2M [40] | ResNet50  | 900×1600   | 7   | 52.20M | 268.28G | 25.0 |
| Ours    | ResNet50  | 640×640    | 44  | 32.34M | 35.86G  | 22.2 |
| Ours    | ResNet50  | 768×768    | 33  | 33.07M | 51.68G  | 22.9 |
| Ours    | ResNet50  | 1024×1024  | 19  | 36.13M | 92.29G  | 23.9 |

G. Network Efficiency

We use the different input image resolutions to measure FPS, the number of parameters, and FLOPs for the competing methods (i.e., VED [37], VPN [43], PON [3], Stitch [39], and I2M [40]) in Table XI. All methods are tested on the same platform using a single NVIDIA Titan XP GPU. For a fair comparison, we have provided results in three resolutions (i.e., 640×640, 768×768, and 1024×1024) for our model. As observed, our model achieves real-time performance with few parameters using ResNet-18. With ResNet-50 as the backbone, our model efficiency is able to outperform other models at similar input resolution. Thus, our model efficiency is comparable to these methods without using any model compression techniques.

V. Conclusion

In this paper, we present a novel framework to estimate road layout and vehicle occupancy in top-views given a front-view monocular image. We propose a front-to-top view projection module that is composed of a cycled view projection structure and a cross-view transformer in which the features of the views before and after projection are explicitly correlated and the most relevant features for view projection are fully exploited in order to enhance the transformed features. In addition, we introduce the multi-scale FTVP modules to compensate for the information loss in the encoded features. We also demonstrate that our proposed model achieves the state-of-the-art performance and runs at 25 FPS on a single GPU, which is efficient and applicable for real-time panorama HD map reconstruction.

Nevertheless, the images or videos captured from real life for the vision tasks are bound to cause class-imbalance problems. Especially for our task, the sample ratio between some classes reaches 100 to 1000 times. Thus, it remains very challenging for our model to predict those classes with small samples, which leads to sub-optimal performance in certain classes for our results. In addition, errors may occur when vehicles are very close to each other, encounter the occlusions, or make sharp turns due to the large view gap. For our task, the major issues rest in the class imbalance and large view gap. As the future work, we will consider to exploit synthetic data to supplement the small sample classes for data balance to strengthen our model performance. On the other hand, we will further exploit the information from the front-view images and explore more effective transformation
between different views. Additionally, another limitation of our proposed method rests in the requirement for the same aspect ratio of the input image and the corresponding output BEV mask, due to the design of our FTVP structure. Since the aspect ratio of BEV mask is usually 1:1, we have to practically employ bilinear interpolation to normalize the input images to the same aspect ratio. Thus, it may cause the undesired deformation of objects, resulting in a noticeable impact on the performance of small objects in particular.

**REFERENCES**

[1] A. Geiger, M. Lauer, C. Wojek, C. Stiller, and R. Urtasun, “3D traffic scene understanding from movable platforms,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 36, no. 5, pp. 1012–1025, May 2014.

[2] T. Sun, Z. Di, P. Che, C. Liu, and Y. Wang, “Leveraging crowdsourced GPS data for road extraction from aerial imagery,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2019, pp. 7501–7510.

[3] T. Roeddick and R. Cipolla, “Predicting semantic map representations from images using attentional pyramidal networks,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2020, pp. 1135–1144.

[4] B. Liu, B. Zhuang, S. Schulte, P. Ji, and M. Chandraker, “Understanding road layout from videos as a whole,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2020, pp. 4413–4422.

[5] J. Liang, N. Homayounfar, W.-C. Ma, S. Wang, and R. Urtasun, “Convolutional recurrent network for road boundary extraction,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2019, pp. 9504–9513.

[6] S. Schulte, M. Zhai, N. Jacobs, and M. Chandraker, “Learning to look around objects for top-view representations of outdoor scenes,” in Proc. Eur. Conf. Comput. Vis., 2018, pp. 815–831.

[7] Z. Wang, B. Liu, S. Schulte, and M. Chandraker, “A parametric top-view regression,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2019, pp. 10317–10325.

[8] X. Chen, K. Kundu, Z. Zhang, H. Ma, and R. Urtasun, “Monocular 3D object detection for autonomous driving,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 2147–2156.

[9] S. Srivastava, F. Jurie, and G. Sharma, “Learning 2D to 3D lifting for object detection in 3D for autonomous vehicles,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst., 2019, pp. 4504–4511.

[10] T. Roeddick, A. Kendall, and R. Cipolla, “Orthographic feature transform for monocular 3D object detection,” in Proc. Brit. Mach. Vis. Conf., 2019, pp. 59.1–59.13.

[11] A. Simonelli, S. R. Bulo, L. Porzi, M. Lépez-Antequera, and P. Kontschieder, “Semantics-Guided 3D object detection,” in Proc. IEEE Int. Conf. Comput. Vis., 2019, pp. 1991–1999.

[12] M. Ding et al., “Learning depth-guided convolutions for monocular 3D object detection,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops, 2020, pp. 1000–1001.

[13] B. Li, W. Ouyang, L. Sheng, X. Zeng, and X. Wang, “OS3D: An efficient 3D object detection framework for autonomous driving,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2020, pp. 1019–1028.

[14] S. Mozaffari, O. Y. Al-Jarrah, M. Dianati, P. Jennings, and A. Mouzakitis, “Deep learning-based vehicle behavior prediction for autonomous driving applications: A review,” IEEE Trans. Intell. Transp. Syst., vol. 23, no. 1, pp. 33–47, Jan. 2022.

[15] J. Hong, B. Sapp, and J. Philbin, “Rules of the road: Predicting driving behavior using a convolutional model of semantic interactions,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2020, pp. 8446–8454.

[16] J. Kim, S. Moon, A. Rohrbach, T. Darrell, and J. Canny, “Advisable learning for self-driving vehicles by internalizing observation-to-action rules,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Pattern Recognit., 2020, pp. 9658–9667.

[17] Y. Ma, X. Zhu, S. Zhang, R. Yang, W. Wang, and D. Manocha, “TrafficPredict: Trajectory prediction for heterogeneous traffic-agents,” in Proc. AAAI Conf. Artif. Intell., 2019, pp. 6120–6127.

[18] Y. Hou, Z. Ma, C. Liu, and C. C. Loy, “Learning lightweight lane detection CNNs by self attention distillation,” in Proc. IEEE Int. Conf. Comput. Vis., 2019, pp. 1013–1022.

[19] L. Peng, Z. Chen, Z. Fu, P. Liang, and E. Cheng, “BEVSegFormer: Bird’s eye view semantic segmentation from arbitrary camera rigs,” in Proc. Winter Conf. Appl. Comput. Vis., 2023, pp. 5924–5932.

[20] S. Gourj, Y. Chen, Z. Fu, P. Liang, and E. Cheng, “BEV-SegFormer: Bird’s eye view semantic segmentation from arbitrary camera rigs,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2019, pp. 3146–3154.

[21] Q. Li, Y. Wang, Y. Wang, and H. Zhao, “HDMapNet: An online HD map construction and evaluation framework,” in Proc. IEEE Int. Conf. Robot. Autom., 2022, pp. 4628–4634.

[22] M. Zhai, Z. Bessinger, S. Workman, and N. Jacobs, “Predicting ground-level scene layout from aerial imagery,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2017, pp. 4132–4140.

[23] S. Gupta, V. Tolani, J. Davidson, S. Levine, R. Sukthankar, and J. Malik, “Cognitive mapping and planning for visual navigation,” Int. J. Comput. Vis., vol. 128, no. 4, pp. 1311–1330, 2020.

[24] C. Lu, M. J. G. van de Molengraft, and D. Gubbelman, “Monocular semantic occupancy grid mapping with convolutional variational encoder-decoder networks,” IEEE Trans. Robot. Autom., vol. 4, no. 2, pp. 445–452, Apr. 2019.

[25] A. Saha, O. Mendez, C. Russell, and R. Bowden, “Enabling spatio-temporal aggregation in birds-eye-view vehicle estimation,” in Proc. IEEE Int. Conf. Robot. Autom., 2021, pp. 5133–5139.

[26] Y. B. Can, A. Liniger, O. Unal, D. Paedal, and L. Van Gool, “Understanding bird’s-eye view of road semantics using an onboard camera,” IEEE Trans. Robot. Autom., vol. 33, no. 2, pp. 3302–3309, Apr. 2022.

[27] A. Saha, O. Mendez, C. Russell, and R. Bowden, “Translating images into maps,” in Proc. IEEE Int. Conf. Robot. Automat., 2022, pp. 9200–9206.

[28] S. Gong et al., “GitNet: Geometric prior-based transformation for birds-eye-view segmentation,” in Proc. Eur. Conf. Comput. Vis., 2022, pp. 396–411.

[29] A. Mani, S. Daga, S. Garg, N. S. Shankar, K. M. Jatavallabhula, and K. M. Krishna, “MonoLayout: Amodal scene layout from a single image,” in Proc. Winter Conf. Appl. Comput. Vis., 2020, pp. 1678–1686.

[30] B. Pan, J. Sun, H. Y. T. Leung, A. Andonian, and B. Zhou, “Cross-view semantic segmentation for sensing surroundings,” IEEE Trans. Robot. Autom., vol. 5, no. 3, pp. 4867–4873, Jul. 2020.

[31] B. Li and S. Schulte, “Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3D,” in Proc. Eur. Conf. Comput. Vis., 2020, pp. 194–210.

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
[45] A. Hu et al., “FIERY: Future instance prediction in bird’s-eye view from surround monocular cameras,” in Proc. IEEE Int. Conf. Comput. Vis., 2021, pp. 15 273–15 282.

[46] B. Zhou and P. Krähenbühl, “Cross-view transformers for real-time map-view semantic segmentation,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2022, pp. 13 760–13 769.

[47] C. Pan, Y. He, J. Peng, Q. Zhang, W. Sui, and Z. Zhang, “BAEFormer: Bidirectional and early interaction transformers for bird’s-eye view semantic segmentation,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2023, pp. 9590–9599.

[48] M. Mousavian, D. Anguelov, J. Flynn, and J. Kosecka, “3D bounding box estimation using deep learning and geometry,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2017, pp. 5632–5640.

[49] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, “PointPillars: Fast encoders for object detection from point clouds,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2019, pp. 12 697–12 705.

[50] J. Park et al., “Time will tell: New outlooks and a baseline for temporal multi-view 3D object detection,” 2022, arXiv:2210.02443.

[51] Y. Jiang et al., “PolarFormer: Multi-camera 3D object detection with polar transformer,” in Proc. AAAI Conf. Artif. Intell., 2023, pp. 1042–1050.

[52] J. Lu, Z. Zhou, X. Zhu, H. Xu, and L. Zhang, “Learning ego 3D representation as ray tracing,” in Proc. Eur. Conf. Comput. Vis., Springer, 2022, pp. 129–144.

[53] I. Dwivedi, S. Malla, Y.-T. Chen, and B. Dariush, “Bird’s eye view segmentation using lifted 2D semantic features,” in Proc. Brit. Mach. Vis. Conf., 2021, pp. 6985–6994.

[54] Z. Li et al., “BEVFormer: Learning bird’s-eye-view representation from multi-camera images via spatiotemporal transformers,” in Proc. Eur. Conf. Comput. Vis., Springer, 2022, pp. 1–18.

[55] Y. Liu, T. Wang, X. Zhang, and J. Sun, “PETR: Position embedding transformation for multi-view 3D object detection,” in Proc. Eur. Conf. Comput. Vis., Springer, 2022, pp. 531–548.

[56] Y. Liu et al., “PETRv2: A unified framework for 3D perception from multi-camera images,” in Proc. IEEE Int. Conf. Comput. Vis., 2023, pp. 3262–3272.

[57] Y. Huang, Y. Li, X. Hu, and W. Ci, “Lane detection based on inverse perspective transformation and Kalman filter,” KSI Trans. Internet Inf. Syst., vol. 12, no. 2, pp. 643–661, 2018.

[58] S. A. Abbas and A. Zisserman, “A geometric approach to obtain a bird’s eye view from an image,” in Proc. IEEE Int. Conf. Comput. Vis. Workshops, 2019, pp. 4095–4104.

[59] H. Tang, D. Xu, N. Sebe, Y. Wang, J. J. Corso, and Y. Yan, “Multi-channel attention selection GAN with cascaded semantic guidance for cross-view image translation,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2019, pp. 2417–2426.

[60] A. Vaswani et al., “Attention is all you need,” in Proc. Int. Conf. Neural Inf. Process. Syst., 2017, pp. 5998–6008.

[61] D. Zhang, H. Zhang, J. Tang, M. Wang, X. Hua, and Q. Sun, “Feature pyramid transformer,” in Proc. Eur. Conf. Comput. Vis., 2020, pp. 323–339.

[62] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, “End-to-end object detection with transformers,” in Proc. Eur. Conf. Comput. Vis., 2020, pp. 213–229.

[63] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, “Deformable DETR: Deformable transformers for end-to-end object detection,” 2020, arXiv:2010.04159.

[64] K. Gavrilyuk, R. Sanford, M. Javan, and C. G. Snoek, “Actor-transformers for group activity recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2020, pp. 839–848.

[65] F. Yang, H. Yang, J. Fu, H. Lu, and B. Guo, “Learning texture transformer network for image super-resolution,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2020, pp. 5790–5799.

[66] A. Dosovitskiy et al., “An image is worth 16x16 words: Transformers for image recognition at scale,” 2020, arXiv:2010.11929.

[67] K. Han, A. Xiao, E. Wu, J. Guo, C. Xu, and Y. Wang, “Transformer in transformer,” in Proc. Int. Conf. Neural Inf. Process. Syst., 2021, pp. 15 908–15 919.

[68] Z. Liu et al., “Swin transformer: Hierarchical vision transformer using shifted windows,” in Proc. IEEE Int. Conf. Comput. Vis., 2021, pp. 9992–10002.

[69] W. Wang et al., “Pyramid vision transformer: A versatile backbone for dense prediction without convolutions,” in Proc. IEEE Int. Conf. Comput. Vis., 2021, pp. 548–558.
Yuanlong Yu received the PhD degree in electrical engineering from the Memorial University of Newfoundland, St. Johns, NL, Canada, in 2010. Since 2011, he worked as a postdoctoral fellow respectively with the Memorial University of Newfoundland and Dalhousie University, Halifax, NS, Canada. Since 2013, he has been working as a professor with Fuzhou University, Fuzhou, China. His main interests are computer vision, machine learning, visual attention, and cognitive robotics.

Yuexin Ma received the PhD degree from the Department of Computer Science, University of Hong Kong, in 2019. Now, she is an assistant professor with ShanghaiTech University, China, leading 4DV Lab. Her research interests lie on the computer vision and deep learning. Particularly, she is interested in 3D scene understanding, multi-modal perception, human-machine cooperation, and autonomous driving. She has published dozens of papers on the IEEE Transactions on Pattern Analysis and Machine Intelligence, International Journal of Computer Vision, CVPR, ICCV, ECCV, IJCAI, AAAI, etc.

Shengfeng He (Senior Member, IEEE) received the BSc and MSc degrees from the Macau University of Science and Technology, in 2009 and 2011, respectively, and the PhD degree from the City University of Hong Kong, in 2015. He is an associate professor with the School of Computing and Information Systems, Singapore Management University. He was on the faculty of the South China University of Technology, from 2016 to 2022. His research interests include visual understanding and generative models. He is a senior member of CCF. He serves as the lead guest editor of the International Journal of Computer Vision and the associate editor of the Neurocomputing.

Jia Pan received the PhD degree in computer science from the University of North Carolina at Chapel Hill, Chapel Hill, NC, USA, in 2013. He is currently an associate professor with the Department of Computer Science, University of Hong Kong, Hong Kong. He is also a member of the Centre for Garment Production Limited, Hong Kong. His research interests include robotics and artificial intelligence as applied to autonomous systems, particularly for navigation and manipulation in challenging tasks such as effective movement in dense human crowds and manipulating deformable objects for garment automation.