2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

FastOcc: Accelerating 3D Occupancy Prediction by Fusing the 2D Bird's-Eye View and Perspective View

Jiawei Hou[1*], Xiaoyan Li[2*], Wenhao Guan[1*], Gang Zhang[3], Di Feng[3], Yuheng Du[1] Xiangyang Xue[1], and Jian Pu[4†]

***Abstract*— In autonomous driving, 3D occupancy prediction outputs voxel-wise status and semantic labels for more comprehensive understandings of 3D scenes compared with traditional perception tasks, such as 3D object detection and bird's-eye view (BEV) semantic segmentation. Recent researchers have extensively explored various aspects of this task, including view transformation techniques, ground-truth label generation, and elaborate network design, aiming to achieve superior performance. However, the inference speed, crucial for running on an autonomous vehicle, is neglected. To this end, a new method, dubbed FastOcc, is proposed. By carefully analyzing the network effect and latency from four parts, including the input image resolution, image backbone, view transformation, and occupancy prediction head, it is found that the occupancy prediction head holds considerable potential for accelerating the model while keeping its accuracy. Targeted at improving this component, the time-consuming 3D convolution network is replaced with a novel residual-like architecture, where features are mainly digested by a lightweight 2D BEV convolution network and compensated by integrating the 3D voxel features interpolated from the original image features. Experiments on the Occ3D-nuScenes benchmark demonstrate that our FastOcc achieves state-of-the-art results with a fast inference speed.**

***Index Terms*— Autonomous Driving, Semantic Scene Completion, 3D Occupancy Prediction**

I. INTRODUCTION

Understanding the 3D geometry and semantic information of the surrounding scene is a crucial problem for autonomous driving. Recently, camera-based perception methods have gained widespread concerns due to their lower costs than the LiDAR-based methods. Several approaches have reached remarkable achievements in the 3D perception tasks, such as 3D object detection [1]–[3], bird's-eye-view (BEV) semantic segmentation [4]–[7], *etc*. However, tasks such as 3D object detection are plagued by the long-tail issue and have difficulty recognizing objects with arbitrary shapes or unexpected categories in real-world scenarios.

Camera-based 3D occupancy prediction task takes the multi-camera images as inputs and estimates the occupancy status and semantic label of each 3D voxel of the entire surrounding. Unlike 3D object detection and other perception tasks, it provides denser perception results and demonstrates greater robustness against the weird objects [8], such as buses with bending connections or construction vehicles with long mechanical arms. Moreover, the voxel-based representation has the potential to be extended to various tasks, such as 3D semantic segmentation. At the same time, predicting the occupancy voxels is more efficient than reconstructing the whole 3D scene in detail because most autonomous driving tasks do not need over-elaborate details, such as tree leaves, windows of buildings, the texture of sidewalk tiles, and so on.

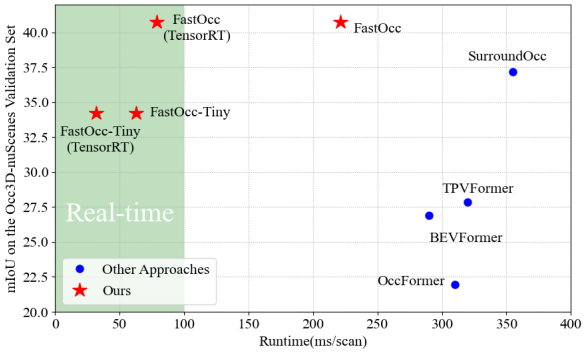


Fig. 1: Comparisons of the mIoU and runtime of various 3D occupancy prediction methods on the Occ3D-nuScenes [8] validation set.

Despite the advantages mentioned above, 3D occupancy prediction is a highly challenging task that demands robustness, accuracy, and practical real-time efficiency. The previous works [8]–[12] investigated various aspects of 3D occupancy prediction tasks, including the feature representation, transformation from the image view to the voxel view, elaborate networks and ground-truth label generation, to improve the prediction accuracy. However, as shown in Fig. 1, many existing methods suffer from a significant computational burden during the prediction process, making them unsuitable for the real-time perception requirements, which is vital for autonomous driving.

To this end, we propose FastOcc, a new 3D occupancy prediction method with the real-time inference speed and competitive accuracy compared with the state-of-the-art approaches. The network effect and latency of different approaches are extensively evaluated and illustrated in the ablation study according to four parts, including the input image resolution, image backbone, view transformation, and occupancy prediction head. From these experimental results,

* These authors contributed equally to this work
† Corresponding author
[1] School of Computer Science, Fudan University, Shanghai, China. {jwhou23, whguan21, yhdu22}@m.fudan.edu.cn, xyxue@fudan.edu.cn
[2] Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China. xiaoyan.li@bjut.edu.cn
[3] Mogo Auto Intelligence and Telematics Information Technology Co., Ltd. zhanggang11021136@gmail.com
[4] Institute of Science and Technology for Brain-Inspired Intelligence, Fudan University, Shanghai, China. jianpu@fudan.edu.cn

it is observed that the 3D convolution or deconvolution used in the occupancy prediction head has considerable potential for optimizing the speed-accuracy trade-off. While most existing methods lift image features to 3D voxel features and straightly decode them in 3D representation, our proposed method first employs a fast approach to obtain volume features. Then the 3D form feature is collapsed to the 2D BEV representation and decoded in the BEV form. To address the absence of z-axis information in the BEV representation, a fast and simple interpolation sampling method is applied to extract 3D features with height information from the image features. Subsequently, the BEV features and the interpolated features are integrated for the final prediction results. Essentially, our method simplifies the process of a 3D perception task as the feature is compressed to BEV representation and decoded in 2D form, and then interpolated 3D features are employed to refine and enhance the 2D features. Supervision is applied both on BEV features and final voxel features. Our proposed method achieves state-of-the-art results with high efficiency compared to other methods. Furthermore, to adapt our method to the real-time perception requirements of autonomous driving, the network structure and setups are optimized and accelerated while ensuring precision. TensorRT SDK [13] is also employed for further acceleration.

Our contributions can be summarized as follows:

- A detailed comparison of the network effect and latency is conducted on four parts in the occupancy prediction task, including the input image resolution, image backbone, view transformation, and occupancy prediction head. Results are presented in the ablation study.
- A novel efficient approach named FastOcc is proposed, which accelerates the 3D occupancy prediction process by simplifying 3D convolution blocks to a 2D BEV convolution network and completing the BEV features with the interpolated voxel features.
- FastOcc achieves the state-of-the-art mIoU of 40.75 while running much faster compared to other methods on the Occ3D-nuScenes [8] dataset. The latency of a single inference is reduced to 63 ms and can be further reduced to 32 ms with the TensorRT SDK [13] acceleration.

II. Related Work

A. Traditional Visual Perception

In recent years, there has been a growing interest in the perception of autonomous vehicles to understand the surrounding environment. BEV perception [1], [5], [14]–[16] has been one of the focal points. Various methods aimed to transform the individual feature representations from RGB cameras into a unified representation, which facilitates modeling of the surrounding environment. LSS [5] estimated per-pixel depth and used the depth feature to place features at their estimated 3D locations. Simple-BEV [14] proposed to project the pre-defined 3D coordinates into images and rise bilinearly sampled features to 3D volume grids. BEVFormer [16] used deformable attention operations to integrate image features into 3D grid coordinates.

3D object detection [1], [17]–[19] has emerged as a simple and effective approach for perception, leveraging input from surround-view RGB cameras. Various works [17], [19]–[21] have reached great effect on this task, which allows for the accurate estimation of objects using 3D bounding box with dimensions, positions, and orientation. The bounding box has been widely accepted as a suitable representation for autonomous driving tasks, especially for objects in traffic environments that exhibit rigid body attributes, such as vehicles. However, some objects with unique shapes and irregular structures are not well-suited for this format.

B. 3D Occupancy Prediction

3D occupancy perception [8], [9], [12], [22], [23] is a task that can obtain more detailed scene perception results while demonstrating good scalability and adaptability to downstream tasks. The pioneering Monoscene [22] utilized a monocular camera as input for semantic scene completion. It employed a continuous 2D-3D UNet [24] to map the image feature to a 3D representation. However, due to the monocular perspective limitation, inferring fine-grained and accurate results with a simple framework is challenging and vulnerable to occlusion, distortion, and ghosting issues. TPVFormer [9] incorporated surround multi-camera input and lifted features to a tri-perspective view space using a transformer-based approach. As it relied on sparse LiDAR points for supervision, the predicted results were also sparse. SurroundOcc [12] generated 3D voxel features at multiple scales using a transformer-based approach and combined them through deconvolutional upsampling. It also proposed a pipeline to obtain dense semantic occupancy supervision from sparse LiDAR information, resulting in a dense prediction. CTF-Occ [8] gradually refined the 3D voxel features from various scales in a coarse-to-fine manner and constructed a dense visibility-aware benchmark. However, the prediction process in these methods is time-consuming and far from the real-time perception requirements of autonomous driving. For example, the network of SurroundOcc [12] takes more than 300 ms for a single inference. While most methods directly enhance the feature transformation from images to dense 3D voxel representations using carefully designed approaches to achieve better results, our approach converts image features to BEV features in a straightforward manner and employs a fast interpolation method to complement the missing height dimension features of BEV, resulting in equally accurate occupancy prediction results with significantly reduced computational overhead.

III. Methodology

In this section, first, we illustrate the visual 3D occupancy prediction task and provide a formulaic expression of the entire process in III-A. Subsequently, as shown in Fig. 2, the pipeline of the proposed FastOcc can be divided into three parts, including image feature extraction, view transformation, and occupancy prediction head. III-B shows the

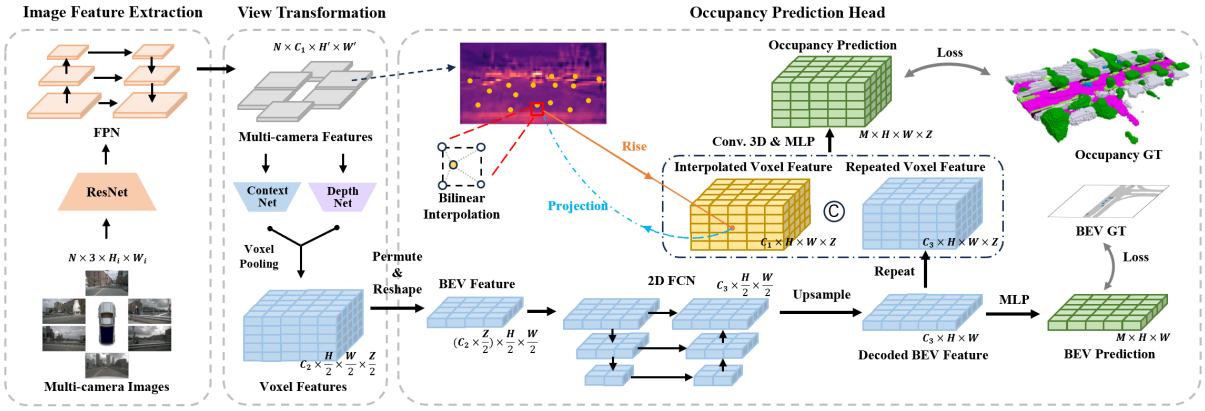


Fig. 2: The pipeline of the proposed method. First, multi-camera features are extracted from image inputs with a backbone network. Then image features are transformed to the 3D space following the LSS [5] strategy. The voxel feature is collapsed to the BEV form and decoded in the 2D representation. Subsequently, the BEV features are upsampled, repeated, and supplemented with the voxel features interpolated from image features. BEV semantic segmentation is supervised as an auxiliary loss.

employed feature extraction backbone. In III-C, widely-used 2D-to-3D view transformation methods are evaluated and the strategy used in our approach is illustrated. Most importantly, our novel occupancy prediction head is illustrated in III-D, where the 3D convolution blocks are simplified by a 2D BEV convolution network, and 2D features are fused with the interpolated voxel features for further fine-tuning. III-E introduces the training loss function.

A. Problem Formulation

In this work, the 3D surrounding scene to be predicted is divided by voxels. Assuming that the autonomous ego is placed at the origin of the real-world coordinates, the scene perception range is denoted as $[H_s, W_s, Z_s, H_e, W_e, Z_e]$. Given that the shape of 3D volume grids is $[H, W, Z]$, each voxel v has the shape of

$$[\frac{W_e - W_s}{W}, \frac{H_e - H_s}{H}, \frac{Z_e - Z_s}{Z}], \tag{1}$$

and the semantic occupancy labels can be defined as $\mathbf{Y}^* \in \mathbb{R}^{M \times H \times W \times Z}$, where M is the number of semantic labels, including the unoccupied voxels denoted as *empty*. Taking multi-camera images $\mathbf{X} = \{\mathbf{X}^1, \mathbf{X}^2, \cdots, \mathbf{X}^N\}$ from N cameras as input, a neural network $\mathscr{G}$ is developed to tackle the semantic occupancy prediction task, which is represented as:

$$\mathbf{Y} = \mathscr{G}(\mathbf{X}^1, \mathbf{X}^2, \cdots, \mathbf{X}^N), \tag{2}$$

where $\mathbf{Y}$ is the predicted result.

B. Image Feature Extraction

The image feature extraction process takes multi-camera images $\mathbf{X} \in \mathbb{R}^{N \times 3 \times H_i \times W_i}$ as inputs, where $[H_i, W_i]$ is the shape of input images. Then a UNet-like [24] backbone is employed to extract multi-camera features $\mathbf{F} = \{\mathbf{F}^1, \mathbf{F}^2, \cdots, \mathbf{F}^N\}$. In our implementation, ResNet-like [25] blocks are employed to encode image features to $1/16$ of the origin shape and the feature pyramid network (FPN) [26] is applied to aggregate features into scale $[H', W']$. The output feature can be denoted as $\mathbf{F} \in \mathbb{R}^{N \times C_1 \times H' \times W'}$.

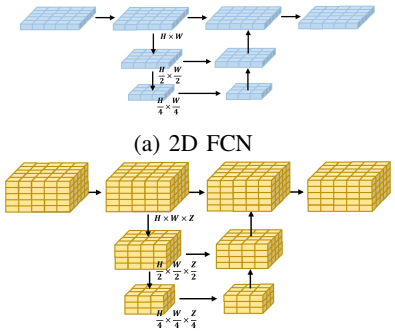
C. View Transformation

In the view transformation process, image features $\mathbf{F}$ from multiple cameras are lifted to a unified 3D form to represent the 3D scene uniquely. The transformed feature can be denoted as $\mathbf{V_B} \in \mathbb{R}^{C_2 \times \frac{H}{2} \times \frac{W}{2} \times \frac{Z}{2}}$, where C_2 is the embedding dim, and to lower the cost, features are transformed to a rather coarse grid size $[\frac{H}{2}, \frac{W}{2}, \frac{Z}{2}]$. Many previous occupancy prediction methods [8], [12], [27] build 3D volume queries and apply the cross-view attention [16] to integrate the multi-view 2D image features into 3D space. However, for high efficiency, the principle proposed by LSS [5] is employed as our view transformation strategy. The LSS [5] approach estimates the depth and context features simultaneously and applies a voxel-pooling mechanic to integrate the 2D features into 3D representation. Moreover, we adopt the BEVDepth [15], which introduces point clouds to supervise the depth feature predicted by the depth net of LSS [5]. By estimating the depth of each pixel, the image features are projected with depth uncertainty accounted for. The transformation strategy, which applies the depth supervision together with the depth-context correspondence, is demonstrated to have a better performance and faster speed in our experiments.

D. Occupancy Prediction Head

To get the 3D prediction output efficiently and effectively, the original 3D feature decoding process is replaced by a residual-like architecture, which is composed of the BEV feature decoding process, the image feature interpolation sampling for compensating the z-axis information, and the final feature integration. These components are introduced as follows.

BEV Feature Decoding. Most of the existing methods directly decode the volume features in 3D form. Taking the 3D fully convolutional network (FCN) as an example, for



(a) 2D FCN

(b) 3D FCN

Fig. 3: The comparison of applying 2D FCN and 3D FCN. It is obvious that 2D FCN is highly efficient in terms of time and memory cost.

the j^{th} 3D convolution layer, the number of floating point operations (FLOPs) can be calculated as

$$FLOPs_j^{3D} = C_j^{in} \times {k_j}^3 \times C_j^{out} \times H_j \times W_j \times Z_j, \tag{3}$$

where in layer j, C_j^{in} is the number of the input channels, k_j is the convolution kernel size, C_j^{out} is the number of the output channels, and $[H_j, W_j, Z_j]$ is the shape of the 3D feature map.

Compared with straightly decoding the lifted voxel feature in 3D space, the proposed method employs a lightweight 2D BEV decoder. Given the previous view transformation outputs $\mathbf{V_B} \in \mathbb{R}^{c_2 \times \frac{H}{2} \times \frac{W}{2} \times \frac{Z}{2}}$, the proposed method first combine the z dim of 3D voxel features $\mathbf{V_B}$ with its embedding channel to get the 2D BEV features $\mathbf{B'} \in \mathbb{R}^{(C_2 \times \frac{Z}{2}) \times \frac{H}{2} \times \frac{W}{2}}$. Then $\mathbf{B'}$ is decoded with a 2D FCN to the BEV feature $\mathbf{B} \in \mathbb{R}^{C_3 \times \frac{H}{2} \times \frac{W}{2}}$, as shown in Fig. 3. This reduces the computational complexity to a large extent. The FLOPs in each 2D convolution layer j can be calculated as

$$FLOPs_j^{2D} = C_j^{in} \times {k_j}^2 \times C_j^{out} \times H_j \times W_j. \tag{4}$$

Consequently, in the first layer, $C_1^{in} = C_2 \times \frac{Z}{2}$, the 2D convolution layer is theoretically k times faster than the 3D convolution layer. In the subsequent layer $j\,(j > 1)$, the 2D convolution layers is s_j times faster than 3D ones, s_j can be computed as

$$s_j = \frac{FLOPs_j^{3D}}{FLOPs_j^{2D}} = k_j \times Z_j. \tag{5}$$

Image Feature Interpolation Sampling. To augment the absent z-axis information in the BEV form and minimize the computational complexity, a simple and efficient approach is designed to acquire 3D features.

To be more specific, first, a 3D volume coordinate is created according to the voxel space shape $[H, W, Z]$ and assigned to the ego coordinate, defined as C_{ego}. Then the transformation from the ego to the image can be computed as $T_{e2i} = T_{c2i} \times T_{e2c}$, where T_{c2i} is the camera intrinsic matrix and T_{e2c} is the transformation from ego car to camera. The coordinate C_{ego} is projected to the images to get the correspondence between the grid coordinate and perspective-view features, and the projected grid can be defined as

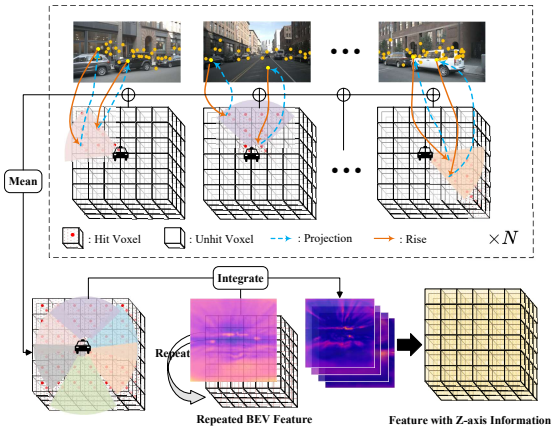


Fig. 4: In the upper dashed box, the volume grids are projected to multiple perspective images. Features of hit voxels on the sub-pixels are bilinearly interpolated and lifted to corresponding 3D space. Below, the absence of the z-axis of repeated BEV features can be completed by the interpolated features.

$C_{image} = T_{e2i} \times C_{ego}$. After that, points that exceed the image range or have a negative depth are filtered out. Subsequently, we apply bilinear sampling to interpolate features from projected sub-pixel coordinates on multiple cameras and compute the mean value after masking out unobserved voxels. Fig. 4 illustrates the detailed process. The FLOPs of the interpolation sampling process is

$$FLOPs^{inter} = 4N \times C \times H \times W \times Z, \tag{6}$$

where 4 neighbor pixels are referred for bilinear sampling sub-pixel features with dim C from N cameras.

Feature Integration. To integrate the 2D BEV feature with interpolated 3D voxel feature, decoded BEV features $\mathbf{B}$ at scale $[\frac{H}{2}, \frac{W}{2}]$ are upsampled to a fine-grained scale $[H, W]$ and repeated at the z-axis, denoted as $\mathbf{B_z} \in \mathbb{R}^{C_3 \times H \times W \times Z}$. The interpolated voxel feature $\mathbf{P} \in \mathbb{R}^{C_1 \times H \times W \times Z}$ is obtained in a fast manner directly at the fine-grained scale with more detailed information. $\mathbf{B_z}$ and $\mathbf{P}$ are concatenated together and integrated by a convolution layer to get the output voxel feature $\mathbf{V}$.

Moreover, to ensure that the decoded BEV feature $\mathbf{B}$ contains enough information for further fine-tuning, it is processed by a UNet-like [24] semantic segmentation head and supervised by the BEV ground truth $\mathbf{B_{gt}} \in \mathbb{R}^{M \times H \times W}$. To generate the BEV ground truth $\mathbf{B_{gt}} \in \mathbb{R}^{M \times H \times W}$ from occupancy ground truth $\mathbf{V_{gt}} \in \mathbb{R}^{M \times H \times W \times Z}$, we simply count the voxels occupied by each class at the z-axis and assign the BEV grid as occupied by each class using a binary multi-class vector.

Rather than simply repeating BEV features to 3D form, which results in redundancy on the z-axis, integrating with the interpolated voxel features incorporates multiple perspective images and achieves better scene understanding, as shown in Fig. 4.

For the entire occupancy prediction head. If a 3D FCN is

TABLE I: 3D semantic occupancy prediction performance on the validation set of Occ3D-nuScenes [8]. For a fair comparison, we train SurroundOcc [12] on the Occ3D-nuScenes dataset with its origin setups, denoted as SurroundOcc*.

Method	others	barrier	bicycle	bus	car	const. veh.	motorcycle	pedestrian	traffic cone	trailer	truck	drive. surf.	other flat	sidewalk	terrain	manmade	vegetation	mIoU
MonoScene [22]	1.75	7.23	4.26	4.93	9.38	5.67	3.98	3.01	5.90	4.45	7.17	14.91	6.32	7.92	7.43	1.01	7.65	6.06
TPVFormer [9]	7.22	38.90	13.67	40.78	45.90	17.23	19.99	18.85	14.30	26.69	34.17	55.65	35.47	37.55	30.70	19.40	16.78	27.83
BEVDet [1]	4.39	30.31	0.23	32.36	34.47	12.97	10.34	10.36	6.26	8.93	23.65	52.27	24.61	26.06	22.31	15.04	15.10	19.38
OccFormer [28]	5.94	30.29	12.32	34.40	39.17	14.44	16.45	17.22	9.27	13.90	26.36	50.99	30.96	34.66	22.73	6.76	6.97	21.93
BEVFormer [16]	5.85	37.83	17.87	40.44	42.43	7.36	23.88	21.81	20.98	22.38	30.70	55.35	28.36	36.0	28.06	20.04	17.69	26.88
CTF-Occ [8]	8.09	39.33	20.56	38.29	42.24	16.93	24.52	22.72	21.05	22.98	31.11	53.33	33.84	37.98	33.23	20.79	18.0	28.53
SurroundOcc* [12]	8.97	**46.33**	17.08	**46.54**	52.01	20.05	21.47	23.52	18.67	**31.51**	37.56	81.91	41.64	50.76	**53.93**	**42.91**	**37.16**	37.18
FastOcc(Ours)	**12.06**	43.53	**28.04**	44.80	**52.16**	**22.96**	**29.14**	**29.68**	**26.98**	30.81	**38.44**	**82.04**	**41.93**	**51.92**	53.71	41.04	35.49	**39.21**
SurroundOcc*-TTA [12]	9.42	43.61	19.57	**47.66**	53.77	21.26	22.35	24.48	19.36	32.96	39.06	83.15	43.26	52.35	55.35	**43.27**	**38.02**	38.69
FastOcc-TTA(Ours)	**12.86**	**46.58**	**29.93**	46.07	**54.09**	**23.74**	**31.10**	**30.68**	**28.52**	**33.08**	**39.69**	**83.33**	**44.65**	**53.90**	**55.46**	42.61	36.50	**40.75**

TABLE II: Comparisons of the mIoU and latency of the proposed components. The SurroundOcc* is progressively refined to the proposed FastOcc. The input image size is 640×1600 and image backbone is the ResNet-101.

Method	View Transformation	Occupancy Prediction Head	mIoU	Latency(ms)
SurroundOcc*	BEVFormer [16]	Deconv.	37.18	355
Baseline	LSS [5]	Deconv.	38.44	306
Baseline$^+$	LSS [5]	3D FCN	**41.02**	342
FastOcc	LSS [5]	2D FCN	40.75	**221**

TABLE III: The ablation study of the image backbones and input resolutions. TRT means the acceleration of the TensorRT SDK [13].

	Method	Backbone	Input Res.	mIoU	Latency(ms) 2D / 2D-to-3D / 3D / Total
Pytorch	FastOcc-Tiny	ResNet-50	320×800	34.21	26.32 / 3.59 / 32.89 / **62.80**
	FastOcc-Small	ResNet-101	320×800	37.21	53.86 / 3.59 / 32.89 / 90.34
	FastOcc	ResNet-101	640×1600	**40.75**	176.82 / 11.57 / 32.89 / 221.28
TRT	FastOcc-Tiny	ResNet-50	320×800	34.21	14.11 / 1.80 / 16.02 / **31.94**
	FastOcc	ResNet-101	640×1600	**40.75**	58.42 / 3.62 / 16.21 / 78.25

applied, the computation complexity is of $O(k^3 C_{in} C_{out} HWZ)$. In our method, the cost of interpolation sampling is much less than multiple convolution layers, consequently, the computational complexity is dominated by $O(k^2 C_{in} C_{out} HW)$.

E. Loss Function

To train the model, we apply the focal loss L_f following M2BEV [29], the affinity loss L_{sem}, L_{geo}, and dice loss L_{dice} introduced in MonoScene [22], the lovasz-softmax loss L_{ls} from OpenOccupancy [10]. As mentioned above, to ensure that the features are transformed of high quality, we supervise the perspective depth with L_d and BEV feature map with binary cross-entropy loss L_b. The final loss is composed of:

$$Loss = L_f + L_{sem} + L_{geo} + L_{dice} + L_{ls} + L_d + L_b. \quad (7)$$

TABLE IV: The ablation study of the BEV supervision and interpolated feature fusion. The decoded BEV features are straightly repeated and regressed to occupancy if interpolated features are not fused with.

BEV Supervision	Interpolated Feature Fusion	mIoU
√	-	31.67
-	√	33.08
√	√	**34.21**

IV. EXPERIMENTS

A. Experimental Setups

Dataset and Evaluation Metrics. Occ3D-nuScenes [8], [30] provides the ground truth of a voxelized representation of the 3D space, with the occupancy state and semantic labels jointly estimated. The benchmark contains 28,130 train samples, 6,019 validation samples, and 6,008 test samples. The perception range is $[-40m, -40m, -1m, 40m, 40m, 5.4m]$ and is divided by voxels with size $0.4m$. The voxels are classified into 18 semantic categories.

For evaluation, following the previous works [9], [12], [23], [27], the mean intersection over union (mIoU) of all semantic classes is employed for the 3D semantic occupancy prediction task.

Implementation Details. For our best result, ResNet-101 [25] pretrained on FCOS3D [19] is employed as the image backbone, and the input image size is cropped to 640×1600. The employed FPN [26] has three levels of layers. Image features are transformed to voxel features with shape $[100 \times 100 \times 8]$. The collapsed BEV feature has the shape of $[100 \times 100]$, and the 2D FCN decoder is composed of a ResNet-18 [25] and a 3-level FPN [26]. The decoded BEV features are upsampled and repeated to $[200 \times 200 \times 16]$, which is the same shape as interpolated features. The AdamW [31] optimizer and cosine annealing [32] learning rate scheduler with a warm-up is employed, and the learning rate is initialized to $2e-4$. Data augmentation on both input images and 3D voxels is employed. Test-time augmentation and camera masks that ignore those invisible voxels are also applied. Temporal information from the previous 16 frames is

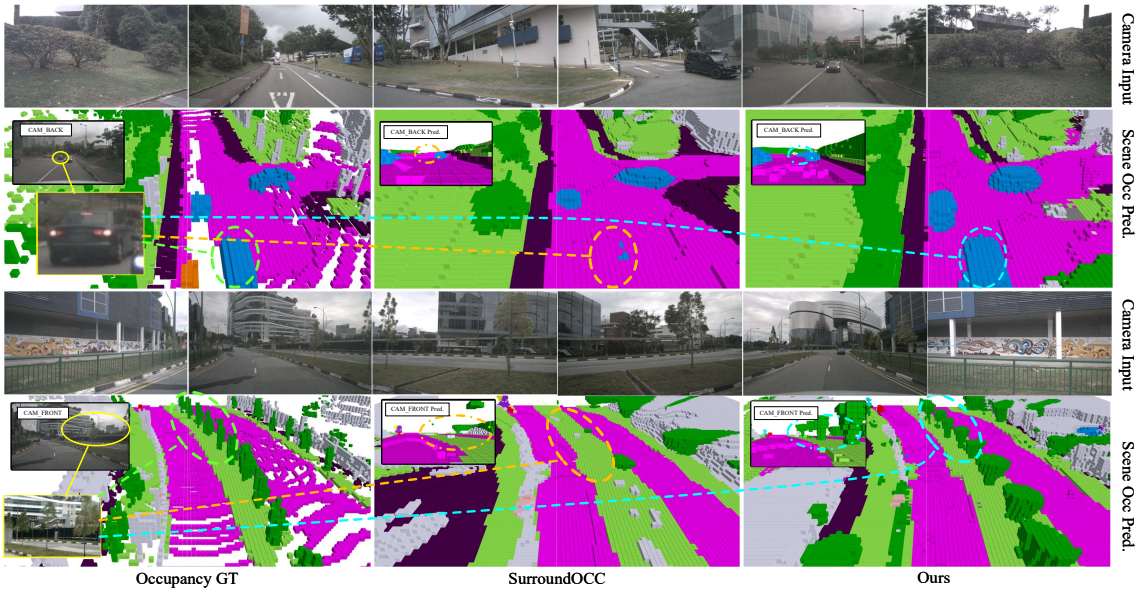


Fig. 5: Visualization of the occupancy prediction results on the validation set of Occ3D-nuScenes [8].

considered for better results. The experiments are conducted on four Tesla V100 GPUs.

B. Evaluation Comparisons

Table I illustrates the comparison of mIoU scores among our method and other relevant approaches for the 3D occupancy prediction task. It is evident that our method achieves high performance on mIoU and most of the categories. Fig. 5 shows the prediction results of FastOcc compared with SurroundOcc [12]. It is obvious that FastOcc fills the blank grids of the ground truth in a more reasonable manner and avoids perception failures on distant cars and blurry trees.

C. Ablation Study

Effects of View Transformation and Occupancy Prediction Head. The transformation method to lift 2D features to 3D space has always been a popular topic. We compare the efficiency and the resulting mIoU scores of the transformer-based method [16] and LSS [5] strategy on our baseline work. Table II shows the results. SurroundOcc* [12] employs multi-scale cross-view attention [16] as the view transformation method and decodes the features using 3D deconvolution network. We implement the Baseline applying the LSS [5] strategy following [33]. Compared with SurroundOcc [12], the Baseline model results in better results with faster speed. Moreover, the occupancy prediction head is ablated to show the efficiency of the proposed method. In Baseline^{+}, a 3D FCN is applied to get better results compared to the multi-scale deconvolution network used in Baseline, but the computation cost increases obviously. In FastOcc, the 2D FCN network is used as the occupancy prediction head, which retains the mIoU with a much faster inference speed. From the comparisons, it is obvious that the depth-supervised LSS [5] and 2D FCN with interpolated features completion present both effectiveness and efficiency.

Effects of Input Resolution and Image Backbone. We also evaluate the impact of the input image resolutions and image backbones. As shown in Tabel III, both higher image resolution and stronger image backbone lead to more accurate results (higher mIoU). Besides, the proposed FastOcc is further accelerated by the TensorRT SDK [13]. Specifically, FastOcc-Tiny and FastOcc run 31.94 ms and 78.25 ms, respectively, to meet the real-time inference requirement.

Effects of BEV Supervision and Interpolation. Recovering the complete 3D voxel information from the 2D BEV features is a challenging task since the z-axis is absent. To tackle this problem, we propose two strategies: 1) the BEV supervision imposes the 3D information on the 2D BEV features; 2) the interpolated voxel features sampling from the images serve as a supplement. The results in Table IV demonstrate the effectiveness of the two strategies.

V. CONCLUSIONS

In this paper, FastOcc is proposed for efficient 3D semantic occupancy prediction. 3D voxel features are compressed to be 2D BEV features after view transformation, where a 2D FCN is applied for efficient feature extraction. Subsequently, the absent z-axis of the BEV features is compensated by the interpolated voxel features from the image, resulting in the complete 3D voxel information with efficiency. Comparisons with other methods on the Occ3D-nuScenes [8] dataset demonstrate the advantages of the proposed components. The proposed FastOcc achieves a leading mIoU of 40.75 and the FastOcc-Tiny runs 32 ms with the TensorRT SDK [13] acceleration.

ACKNOWLEDGMENT

This work was supported in part by NSFC Project (62176061), STCSM Project (No.22511105000) and Shanghai Platform for Neuromorphic and AI Chip under Grant 17DZ2260900 (NeuHelium).

REFERENCES

[1] J. Huang, G. Huang, Z. Zhu, and D. Du, "Bevdet: High-performance multi-camera 3d object detection in bird-eye-view," *ArXiv preprint*, vol. abs/2112.11790, 2021.

[2] C. Yang, Y. Chen, H. Tian, C. Tao, X. Zhu, Z. Zhang, G. Huang, H. Li, Y. Qiao, L. Lu, J. Zhou, and J. Dai, "Bevformer v2: Adapting modern image backbones to bird's-eye-view recognition via perspective supervision," *ArXiv preprint*, vol. abs/2211.10439, 2022.

[3] S. Wang, Y. Liu, T. Wang, Y. Li, and X. Zhang, "Exploring object-centric temporal modeling for efficient multi-view 3d object detection," *ArXiv preprint*, vol. abs/2303.11926, 2023.

[4] L. Peng, Z. Chen, Z. Fu, P. Liang, and E. Cheng, "Bevsegformer: Bird's eye view semantic segmentation from arbitrary camera rigs," *ArXiv preprint*, vol. abs/2203.04050, 2022.

[5] J. Philion and S. Fidler, "Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d," in *Computer Vision – ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds. Cham: Springer International Publishing, 2020, vol. 12359, pp. 194–210.

[6] B. Zhou and P. Krähenbühl, "Cross-view transformers for real-time map-view semantic segmentation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*. IEEE, 2022, pp. 13 750–13 759.

[7] X. Zhu, X. Cao, Z. Dong, C. Zhou, Q. Liu, W. Li, and Y. Wang, "Nemo: Neural map growing system for spatiotemporal fusion in bird's-eye-view and bdd-map benchmark," *ArXiv preprint*, vol. abs/2306.04540, 2023.

[8] X. Tian, T. Jiang, L. Yun, Y. Wang, Y. Wang, and H. Zhao, "Occ3d: A large-scale 3d occupancy prediction benchmark for autonomous driving," *ArXiv preprint*, vol. abs/2304.14365, 2023.

[9] Y. Huang, W. Zheng, Y. Zhang, J. Zhou, and J. Lu, "Tri-perspective view for vision-based 3d semantic occupancy prediction," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*. IEEE, 2023, pp. 9223–9232.

[10] X. Wang, Z. Zhu, W. Xu, Y. Zhang, Y. Wei, X. Chi, Y. Ye, D. Du, J. Lu, and X. Wang, "Openoccupancy: A large scale benchmark for surrounding semantic occupancy perception," *ArXiv preprint*, vol. abs/2303.03991, 2023.

[11] Y. Wang, Y. Chen, X. Liao, L. Fan, and Z. Zhang, "Panoocc: Unified occupancy representation for camera-based 3d panoptic segmentation," *ArXiv preprint*, vol. abs/2306.10013, 2023.

[12] Y. Wei, L. Zhao, W. Zheng, Z. Zhu, J. Zhou, and J. Lu, "Surroundocc: Multi-camera 3d occupancy prediction for autonomous driving," *ArXiv preprint*, vol. abs/2303.09551, 2023.

[13] H. Vanholder, "Efficient inference with tensorrt," in *GPU Technology Conference*, vol. 1, 2016, p. 2.

[14] A. W. Harley, Z. Fang, J. Li, R. Ambrus, and K. Fragkiadaki, "Simple-bev: What really matters for multi-sensor bev perception?" *ArXiv preprint*, vol. abs/2206.07959, 2022.

[15] Y. Li, Z. Ge, G. Yu, J. Yang, Z. Wang, Y. Shi, J. Sun, and Z. Li, "Bevdepth: Acquisition of reliable depth for multi-view 3d object detection," *arXiv preprint arXiv:2206.10092*, 2022.

[16] Z. Li, W. Wang, H. Li, E. Xie, C. Sima, T. Lu, Y. Qiao, and J. Dai, "Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers," *arXiv preprint arXiv:2203.17270*, 2022.

[17] Y. Wang, V. Guizilini, T. Zhang, Y. Wang, H. Zhao, and J. Solomon, "Detr3d: 3d object detection from multi-view images via 3d-to-2d queries," *ArXiv preprint*, vol. abs/2110.06922, 2021.

[18] Z. Luo, C. Zhou, G. Zhang, and S. Lu, "Detr4d: Direct multi-view 3d object detection with sparse attention," *ArXiv preprint*, vol. abs/2212.07849, 2022.

[19] T. Wang, X. Zhu, J. Pang, and D. Lin, "FCOS3D: fully convolutional one-stage monocular 3d object detection," in *IEEE/CVF International Conference on Computer Vision Workshops, ICCVW 2021, Montreal, BC, Canada, October 11-17, 2021*. IEEE, 2021, pp. 913–922.

[20] Y. You, Y. Wang, W. Chao, D. Garg, G. Pleiss, B. Hariharan, M. E. Campbell, and K. Q. Weinberger, "Pseudo-lidar++: Accurate depth for 3d object detection in autonomous driving," in *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

[21] Y. Liu, J. Yan, F. Jia, S. Li, A. Gao, T. Wang, X. Zhang, and J. Sun, "Petrv2: A unified framework for 3d perception from multi-camera images," *ArXiv preprint*, vol. abs/2206.01256, 2022.

[22] A. Cao and R. de Charette, "Monoscene: Monocular 3d semantic scene completion," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*. IEEE, 2022, pp. 3981–3991.

[23] Z. Li, Z. Yu, D. Austin, M. Fang, S. Lan, J. Kautz, and J. M. Alvarez, "Fb-occ: 3d occupancy prediction based on forward-backward view transformation," *ArXiv preprint*, vol. abs/2307.01492, 2023.

[24] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *ArXiv preprint*, vol. abs/1505.04597, 2015.

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. IEEE Computer Society, 2016, pp. 770–778.

[26] T. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie, "Feature pyramid networks for object detection," in *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*. IEEE Computer Society, 2017, pp. 936–944.

[27] Y. Li, Z. Yu, C. Choy, C. Xiao, J. M. Alvarez, S. Fidler, C. Feng, and A. Anandkumar, "Voxformer: Sparse voxel transformer for camera-based 3d semantic scene completion," *ArXiv preprint*, vol. abs/2302.12251, 2023.

[28] Y. Zhang, Z. Zhu, and D. Du, "Occformer: Dual-path transformer for vision-based 3d semantic occupancy prediction," *ArXiv preprint*, vol. abs/2304.05316, 2023.

[29] E. Xie, Z. Yu, D. Zhou, J. Philion, A. Anandkumar, S. Fidler, P. Luo, and J. M. Alvarez, "M^2bev: Multi-camera joint 3d detection and segmentation with unified birds-eye view representation," *ArXiv preprint*, vol. abs/2204.05088, 2022.

[30] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*. IEEE, 2020, pp. 11 618–11 628.

[31] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

[32] ——, "SGDR: stochastic gradient descent with warm restarts," in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

[33] Z. Li, Z. Yu, W. Wang, A. Anandkumar, T. Lu, and J. M. Alvarez, "Fb-bev: Bev representation from forward-backward view transformations," *ArXiv preprint*, vol. abs/2308.02236, 2023.