# Leveraging Camera-Based Methods for Enhanced Feature-to-World Mapping

Jaeha Song*
*Department of Automotive Engineering*
*Hanyang University*
Seoul, Republic of Korea
archiiive99@hanyang.ac.kr

Sungjin Park*
*Department of Automotive Engineering*
*Hanyang University*
Seoul, Republic of Korea
shihtzu333@hanyang.ac.kr

Soonmin Hwang†
*Department of Automotive Engineering*
*Hanyang University*
Seoul, Republic of Korea
soonminh@hanyang.ac.kr

***Abstract*—Scene representation in autonomous driving relies heavily on extracting meaningful features from images and accurately mapping them to 3D world coordinates. Traditional methods, such as ResNet-based backbones pretrained on ImageNet, provide a robust foundation for feature extraction but are increasingly viewed as limited when it comes to aligning features with the 3D world. This paper explores the integration of advanced segmentation models as backbones, focusing on how feature quality at the extraction stage directly impacts downstream scene representation tasks. Preliminary experiments demonstrate the potential for improved feature alignment and semantic consistency, highlighting the importance of robust backbone design in modern 3D perception pipelines.**

***Index Terms*—3D occupancy prediction, BEV perception, scene representation, autonomous driving**

## I. Introduction

In autonomous driving, one of the most critical challenges lies in extracting high-quality features from image inputs and effectively mapping them to a 3D coordinate system. This process, often determined by the backbone network, serves as the cornerstone for tasks like 3D scene understanding and navigation. Over time, backbones such as ResNet50 and ResNet101 [14], pretrained on ImageNet [13], have been the default choices due to their simplicity and robustness. However, these conventional architectures often fall short in aligning extracted features accurately with real-world locations, limiting their suitability for autonomous driving scenarios.

A well-designed image backbone not only enhances the quality of extracted features but also simplifies the subsequent processing stages. If features are accurately aligned with the 3D world at the backbone stage, the downstream layers—such as attention modules, decoders, or scene representation heads—can be significantly lighter while maintaining high performance. This streamlined pipeline enables a more efficient and interpretable learning process for understanding 3D scenes, whether in the context of BEV perception or 3D occupancy prediction. Both paradigms rely heavily on robust feature extraction, underscoring the universal applicability of optimized backbones across these tasks.

* Equal contribution.
† Corresponding author.

This paper takes a practical approach to evaluate how advanced segmentation models, typically designed for image tasks, perform when their outputs are projected onto LiDAR data for downstream 3D perception tasks. Rather than proposing a complete overhaul of backbone design, we focus on a preliminary investigation into whether leveraging segmentation models for this purpose can offer meaningful insights or improvements. By aligning segmentation outputs with 3D world coordinates, we aim to highlight the potential benefits and limitations of such an approach, leaving further exploration and refinement for future work.

## II. Related Works

Recent advancements in camera-based methods have significantly impacted feature extraction and 3D scene representation. Historically, most approaches have relied on standard backbone networks like ResNet [14], often pretrained on ImageNet [13]. While these backbones provide robustness and simplicity, the ImageNet [13] domain differs significantly from autonomous driving scenarios. This domain gap raises concerns about the ability of such backbones to capture features relevant to autonomous driving tasks. Despite their widespread use, little attention has been given to adapting or optimizing these backbones specifically for the unique requirements of autonomous driving datasets, leaving room for improvement in domain-specific feature extraction.

### A. BEV Perception

BEV perception has been widely adopted for spatial understanding in autonomous driving. BEVFormer [7] employs ResNet-50 and ResNet-101 with DCN to extract image features, which are projected into the bird's-eye-view (BEV) space using a spatiotemporal transformer. BEVDepth [8] integrates depth estimation into the BEV framework, relying on ResNet-50 for robust image feature extraction. Similarly, BEVDet [9] leverages ResNet-101 and Swin Transformer backbones to generate BEV representations through multi-camera inputs and view transformations. These methods underscore the centrality of ResNet-101 in BEV tasks, while primarily focusing on downstream processing and view transformations.

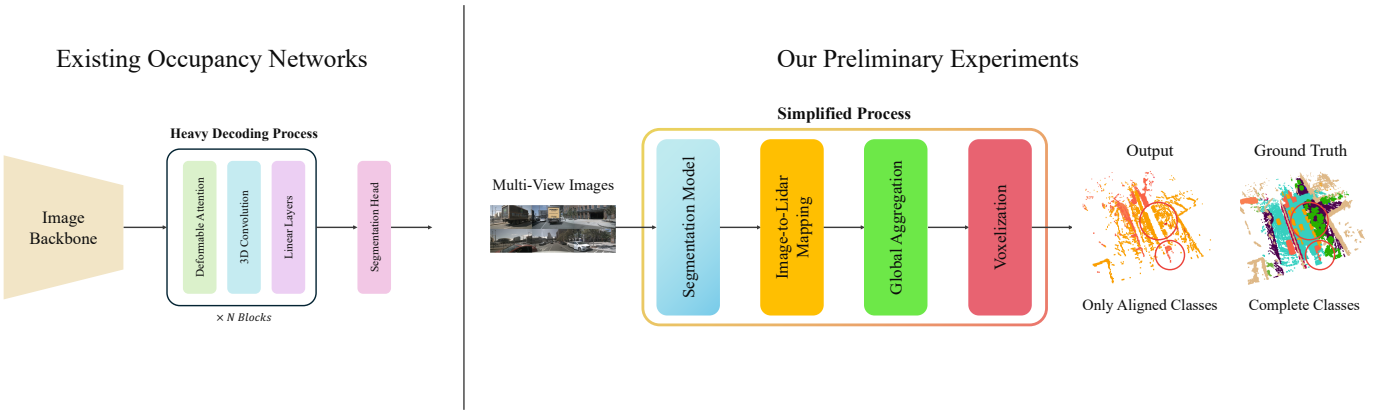


Fig. 1. Comparison of Existing Occupancy Networks and Our Preliminary Experiments. On the left, existing networks rely on an image backbone followed by a heavy decoding process, including deformable attention, 3D convolutions, and linear layers. On the right, our approach simplifies the pipeline by utilizing multi-view images and applying segmentation followed by an image-to-LiDAR mapping, global aggregation, and voxelization. Despite only aligning specific classes from the Occ3D dataset, our results show promising quality and semantic richness in the 3D space.

### B. 3D Occupancy Prediction

3D occupancy prediction aims to label the environment with semantic and geometric information. MonoScene [5] uses ResNet to process monocular images for semantic scene completion. TPVFormer [1] extends the BEV paradigm by using ResNet to lift image features into a tri-perspective view (TPV) representation for richer scene understanding. GaussianFormer [8] adopts ResNet to support probabilistic Gaussian modeling, enabling sparse and efficient 3D occupancy predictions. Similarly, GaussianOcc [9] builds on ResNet to introduce self-supervised learning for scalable occupancy estimation. SurroundOcc [2] utilizes a ResNet, while OctreeOcc [3] employs the same backbone with a focus on hierarchical queries.

Across these methods, ResNet [14] remains the backbone of choice, providing reliable feature extraction while leaving downstream modules to handle task-specific complexities. This reliance on a generic backbone, however, suggests a missed opportunity to explore domain-adaptive designs tailored to autonomous driving. By revisiting this foundational aspect, future work could unlock significant improvements in feature extraction and overall system efficiency.

## III. Methods

This section describes the methodology used to explore the feasibility of leveraging segmentation models for 2D image segmentation and LiDAR mapping in autonomous driving tasks. Figure 1 provides an overview of the proposed pipeline in comparison to previous occupancy networks. While previous approaches rely on a complex decoding process involving deformable attention, 3D convolutions, and linear projections, our method simplifies the pipeline significantly. By leveraging segmentation models and direct image-to-LiDAR mapping, we aim to achieve comparable performance without resorting to heavy post-processing steps.

### A. 2D Image Segmentation

For 2D image segmentation, let $\mathcal{I} = I_1, I_2, \ldots, I_6$ represent the set of six input images captured from surrounding cameras. These images are processed by a segmentation model, denoted as $\mathcal{S}$, which outputs per-pixel semantic probabilities for each image. Mathematically, this can be expressed as:

$$\mathcal{S}(I_i) = P_i, \quad P_i \in \mathbb{R}^{H \times W \times C}, \quad \forall i \in 1, 2, \ldots, 6, \tag{1}$$

where $H$, $W$, and $C$ represent the height, width, and number of classes, respectively.

For each semantic class $c \in 1, \ldots, C$, the per-class probability map $P_i^c$ is extracted, and the segmentation mask is defined as:

$$M_i^c(u, v) = \begin{cases} 1, & \text{if } c = \arg\max_k P_i^k(u, v), \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

where $(u, v)$ are pixel coordinates in the image $I_i$.

This segmentation model $\mathcal{S}$ is pretrained on the COCO dataset and fine-tuned on an autonomous driving dataset to ensure domain relevance. Data augmentation techniques, including color jittering, random cropping, and flipping, are applied during training to improve robustness.

### B. Image-to-LiDAR Mapping

The segmented outputs $\mathcal{P} = P_1, P_2, \ldots, P_6$ are projected onto the LiDAR coordinate system through a series of transformations. The mapping process assumes access to accurate camera calibration parameters and ground-truth depth information, and it involves the following steps:

Depth Association: Each pixel $(u, v)$ in $P_i$ is assigned a depth value $D_i(u, v)$ from the ground-truth depth map $D_i$. This depth association can be expressed as:

$$Q_i(u, v) = \begin{bmatrix} u & v & D_i(u, v) & 1 \end{bmatrix}, \tag{3}$$

where $Q_i(u, v)$ represents the homogeneous coordinate in the image space.

Camera to Ego Transformation: Using the camera intrinsic matrix $K_i$ and extrinsic parameters $T_{\text{camera}\to\text{ego}}$, the 2D pixel coordinates are transformed into the ego coordinate system:

$$X_{\text{ego}} = T_{\text{camera}\to\text{ego}} \cdot K_i^{-1} \cdot Q_i(u, v). \tag{4}$$

Ego to LiDAR Transformation: The coordinates in the ego system are further transformed into the LiDAR coordinate system using $T_{\text{ego}\to\text{LiDAR}}$:

$$X_{\text{LiDAR}} = T_{\text{ego}\to\text{LiDAR}} \cdot X_{\text{ego}}. \tag{5}$$

Semantic Projection: The semantic probabilities $P_i(u, v)$ are assigned to the corresponding 3D points $X_{\text{LiDAR}}$ in the LiDAR coordinate system. The semantic class for each LiDAR point is determined by projecting the 3D points back into the image space and retrieving the corresponding semantic labels:

$$S_{\text{LiDAR},j} = \arg\max_c \frac{1}{|\mathcal{N}_j|} \sum_{(u,v)\in\mathcal{N}_j} P_i^c(u, v), \tag{6}$$

where $\mathcal{N}_j$ represents the set of image pixels that project to the 3D point $X_{\text{LiDAR},j}$, and $j$ denotes the unique identifier for each 3D LiDAR point.

This process produces a semantically enriched 3D point cloud with each point labeled by its most probable class.

### C. Visibility Filtering for LiDAR Points

In this step, LiDAR points that cannot be projected into the field of view of any of the six cameras are excluded from the final representation. This ensures that only visible points are considered for semantic enrichment. The filtering process is described as follows:

Projection into Camera Space: For each LiDAR point $X_{\text{LiDAR},j}$, project it into the camera coordinate system for all six cameras:

$$X_{\text{camera},j} = K_i \cdot T_{\text{global}\to\text{camera}} \cdot X_{\text{LiDAR},j}, \tag{7}$$

where $T_{\text{global}\to\text{camera}}$ is the transformation from the global coordinate system to the camera coordinate system.

Visibility Check: For each projected point $X_{\text{camera},j}$, check if the point falls within the image boundaries and satisfies the depth constraint:

$$0 \le u < W, \quad 0 \le v < H, \quad \text{and } D_j > 0, \tag{8}$$

where $(u, v)$ are the image coordinates derived from $X_{\text{camera},j}$ and $D_j$ is the depth value.

Exclusion of Non-visible Points: Points that fail the visibility check for all six cameras are excluded from further processing:

$$X_{\text{LiDAR},j} \notin \mathcal{V} \implies \text{exclude } X_{\text{LiDAR},j}, \tag{9}$$

where $\mathcal{V}$ is the set of visible points across all cameras.

This filtering ensures that only LiDAR points with valid projections into at least one camera field of view are retained, improving the semantic mapping's accuracy and relevance.

### D. Global Aggregation

To account for temporal information and achieve a denser 3D point cloud, multiple frames are aggregated in the global coordinate system. Specifically, for a time window of $2k$ frames (i.e., $k$ frames before and after the current time $t$), the LiDAR points are transformed and merged as follows:

$$X_{\text{global}} = \bigcup_{\tau=t-k}^{t+k} T_{\tau\to\text{global}} \cdot X_{\text{LiDAR},\tau}, \tag{10}$$

where $T_{\tau\to\text{global}}$ represents the transformation from the local LiDAR frame at time $\tau$ to the global coordinate system. By aggregating points across this temporal duration, a denser and temporally consistent 3D representation of the scene is achieved, capturing information that might be missed in a single frame.

### E. Voxelization

For spatial discretization, the aggregated LiDAR points in the global coordinate system are voxelized into a 3D grid. The voxelization process is defined as:

$$V(x, y, z) = \begin{cases} \text{majority}(S_\ell(p, q, r)), & \text{if } (x, y, z) \in \mathcal{P}, \\ 0, & \text{otherwise} \end{cases} \tag{11}$$

where $V(x, y, z)$ represents the voxel value at the coordinate $(x, y, z)$, $(x, y, z)$ is the voxel's coordinate index, $\mathcal{P}$ denotes the set of LiDAR points within the voxel, and $S_\ell(p, q, r)$ represents the semantic labels of these points.

The voxel grid's dimensions are determined by the predefined point cloud range and voxel size:

$$I = \left\lfloor \frac{X_g - R_{\text{min}}}{s} \right\rfloor, \tag{12}$$

where $I$ is the voxel index, $X_g$ denotes the global coordinates of the LiDAR points, $R_{\text{min}}$ is the minimum bound of the point cloud range, and $s$ is the voxel resolution.

This step ensures a structured and compact representation of the 3D scene, suitable for downstream processing.

## IV. Experiments

### A. Configurations

The configurations used in the experiments are summarized in Table I. These configurations were chosen based on prior research and empirical tuning to optimize performance.

TABLE I
Experimental Configurations

| Configuration | Value |
|---|---|
| Segmentation Model | YOLO11l-seg |
| Pretraining Dataset | COCO |
| Input Image Resolution ($H \times W$) | $384 \times 640$ |
| Number of Classes ($C$) | 5 |
| Voxel Size ($s$) | 0.4 m |
| Point Cloud Range ($R_{\text{min}}, R_{\text{max}}$) | $[-40, -40, -1], [40, 40, 5.4]$ m |
| Temporal Frames ($k$) | 10 |

In this setup, the number of classes ($C$) was chosen as 5, representing the subset of COCO [12] categories that align with the Occ3D-nuScenes [11] dataset. The input image resolution was fixed at $384 \times 640$, ensuring compatibility with the YOLO11l-seg [15] model.


Fig. 2. Qualitative results of the proposed method. Each row shows input images, ground truth, and prediction (left to right).

### B. Experimental Setup

The experiments were conducted to evaluate the performance of the proposed pipeline in terms of both accuracy and computational efficiency.

The segmentation model, YOLO11l-seg [15], was pretrained on the COCO [12] dataset and directly used for zero-shot inference on the Occ3D-nuScenes [11] dataset. No fine-tuning was performed, ensuring that the results strictly reflect the model's generalization capability without additional dataset-specific adjustments.

The hardware setup comprised a single NVIDIA A6000 GPU running PyTorch. Specific hyperparameters, such as voxel size ($s$), point cloud range ($R_{\text{min}}, R_{\text{max}}$), and temporal frame aggregation ($k$), were optimized based on validation

performance.

The evaluation was performed using Intersection over Union (IoU) and runtime as the primary metrics. These metrics were selected to assess both the semantic segmentation accuracy and the computational efficiency of the pipeline.

It is important to note that due to unknown errors encountered during evaluation on the Occ3D-nuScenes [11] validation set, 30 out of 150 scenes could not be assessed. While the remaining scenes are substantial enough to provide meaningful insights into the pipeline's performance, this limitation slightly reduces the reliability of direct comparisons with other models.

### C. Quantitative Results

The experimental results are presented in Table II, which reports IoU scores for five key semantic classes: *bicycle, car, motorcycle, bus, and truck*. These categories are critical for autonomous driving tasks and provide a meaningful evaluation metric for assessing semantic occupancy prediction models. The evaluation is conducted on a subset of the Occ3D-nuScenes [11] dataset, specifically focusing on representative samples that reflect real-world complexities.

Table II compares our proposed method, referred to as "Ours" with several existing methods. Although our method is not a top-performing approach, it achieves competitive results across various metrics.

In Table II, the red highlights represent the upper bounds achieved by existing methods, while the blue highlights denote the lower bounds. Scores achieved by "Ours" that fall between the lower and upper bounds are underlined, showcasing that our method operates effectively within the competitive range.

TABLE II
QUANTITATIVE RESULTS ON OCC3D-NUSCENES

| Method | Bicycle | Car | Motorcycle | Bus | Truck |
|---|---|---|---|---|---|
| MonoScene [5] | 4.26 | 9.38 | 3.98 | 4.93 | 7.17 |
| OccFormer [4] | 13.13 | 37.12 | 14.02 | 20.37 | 20.64 |
| TPVFormer [1] | 13.67 | 45.90 | 19.99 | 40.78 | 34.17 |
| FB-OCC [6] | 30.00 | 51.54 | 29.13 | 46.62 | 39.36 |
| **Ours** | 14.32 | 16.70 | 12.15 | 15.64 | 17.48 |

These findings illustrate that while our method does not consistently surpass state-of-the-art approaches, it performs competitively across most categories. For instance, in the *bicycle* category, our method achieves results that are similar to OccFormer [4] and TPVFormer [1], showcasing its robustness in this class. Additionally, for the *motorcycle* category, our method's performance is comparable to that of OccFormer [4], further demonstrating its effectiveness in this specific context.

In conclusion, our method highlights a promising approach to 3D semantic occupancy prediction by balancing computational simplicity with semantic accuracy. These results suggest that accurate alignment of 2D semantics into the 3D domain can yield competitive results with reduced computational overhead.

### D. Qualitative Results

To further illustrate the performance of our proposed method, qualitative results are provided in Figure 2. The results highlight the strengths and limitations of our approach. Static objects such as buildings and road surfaces are generally well-predicted, aligning closely with the ground truth. However, challenges arise in the handling of dynamic objects like vehicles, where predictions occasionally exhibit elongated or distorted shapes. For pedestrians, the primary limitation lies in the lack of sufficient LiDAR points, which often results in an inability to match the image semantics to corresponding 3D points, preventing proper voxelization. Additionally, some predictions for cluttered regions or overlapping objects exhibit noisy outputs, leading to inaccuracies in object boundaries and semantics. These visualizations demonstrate that while the method shows promising results in strucfred environments with clear object boundaries, further refinement may be needed to robustly handle more dynamic, cluttered, and noisy scenarios, particularly in cases where LiDAR sparsity limits semantic alignment.

## V. CONCLUSIONS

This study demonstrates that 2D image segmentation, when executed effectively, can significantly reduce the computational burden associated with 3D semantic occupancy prediction. By focusing on segmentation and projection methods, we have shown that it is possible to bypass heavy decoding processes while still achieving results comparable to those produced by complex models. Figure 1 highlights this approach, where a lightweight pipeline delivers meaningful semantic representations.

The simplicity of this method is its primary strength, as it avoids the intricate attention mechanisms and 3D convolutions often employed in existing frameworks. However, it does have limitations. Dynamic objects, for instance, appear stretched or misaligned due to temporal aggregation, and certain categories, such as pedestrians, are challenging to map because they often lack sufficient LiDAR points. These gaps in representation highlight areas for future improvement.

Nonetheless, this study establishes an important baseline: robust 2D segmentation alone can serve as a foundational step for high-quality 3D occupancy prediction. By focusing on lightweight and interpretable methods, this work opens up possibilities for more computationally efficient solutions, particularly in scenarios where real-time performance and scalability are critical. Future research could address these limitations by integrating enhanced temporal dynamics and better handling of sparse object classes to further improve the accuracy and applicability of the proposed approach.

## ACKNOWLEDGMENT

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT)(RS-2024-00409492).

## REFERENCES

[1] Y. Huang, W. Zheng, Y. Zhang, J. Zhou, and J. Lu, "Tri-Perspective View for Vision-Based 3D Semantic Occupancy Prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 9223–9232.

[2] Y. Wei, L. Zhao, W. Zheng, Z. Zhu, J. Zhou, and J. Lu, "SurroundOcc: Multi-Camera 3D Occupancy Prediction for Autonomous Driving," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023, pp. 1000–1009.
[3] Y. Lu, X. Zhu, T. Wang, and Y. Ma, "OctreeOcc: Efficient and Multi-Granularity Occupancy Prediction Using Octree Queries," *arXiv preprint arXiv:2312.03774*, 2023.
[4] Y. Zhang, Z. Zhu, and D. Du, "OccFormer: Dual-path Transformer for Vision-based 3D Semantic Occupancy Prediction," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023, pp. 1000–1009.
[5] A.-Q. Cao and R. de Charette, "MonoScene: Monocular 3D Semantic Scene Completion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 3991–4001.
[6] 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 arXiv:2307.01492*, 2023.
[7] Z. Liu, T. Tang, and S. Zhang, "BEVFormer: Learning Bird's-Eye-View Representations from Multi-Camera Images via Spatiotemporal Transformers," *arXiv preprint arXiv:2203.17270*, 2022.
[8] Y. Huang, W. Zheng, B. Zhang, J. Zhou, and J. Lu, "GaussianFormer: Scene as Gaussians for Vision-Based 3D Semantic Occupancy Prediction," *Proceedings of the European Conference on Computer Vision (ECCV)*, 2024.
[9] W. Gan, F. Liu, H. Xu, N. Mo, and N. Yokoya, "GaussianOcc: Fully Self-Supervised and Efficient 3D Occupancy Estimation with Gaussian Splatting," *arXiv preprint arXiv:2408.11447*, 2024.
[10] Y. Huang, A. Thammatadatrakoon, W. Zheng, Y. Zhang, D. Du, and J. Lu, "GaussianFormer-2: Probabilistic Gaussian Superposition for Efficient 3D Occupancy Prediction," *arXiv preprint arXiv:2412.04384*, 2024.
[11] 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 arXiv:2304.14365*, 2023.
[12] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Proc. ECCV*, 2014.
[13] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," in *Proc. CVPR*, 2009.
[14] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. CVPR*, 2016.
[15] Ultralytics Github.