ImGeoNet: Image-induced Geometry-aware Voxel Representation for Multi-view 3D Object Detection

Tao Tu[1] Shun-Po Chuang[2] Yu-Lun Liu[3] Cheng Sun[1] Ke Zhang[4]
Donna Roy[4*] Cheng-Hao Kuo[4] Min Sun[1,4]

[1]National Tsing Hua University [2]National Taiwan University
[3]National Yang Ming Chiao Tung University [4]Amazon

ttaoretw@gmail.com f04942141@ntu.edu.tw yulunliu@cs.nycu.edu.tw
chengsun@gapp.nthu.edu.tw kezha@amazon.com donna.v.roy@gmail.com
chkuo@amazon.com sunmin@ee.nthu.edu.tw

Abstract

We propose ImGeoNet, a multi-view image-based 3D object detection framework that models a 3D space by an image-induced geometry-aware voxel representation. Unlike previous methods which aggregate 2D features into 3D voxels without considering geometry, ImGeoNet learns to induce geometry from multi-view images to alleviate the confusion arising from voxels of free space, and during the inference phase, only images from multiple views are required. Besides, a powerful pre-trained 2D feature extractor can be leveraged by our representation, leading to a more robust performance. To evaluate the effectiveness of ImGeoNet, we conduct quantitative and qualitative experiments on three indoor datasets, namely ARKitScenes, ScanNetV2, and ScanNet200. The results demonstrate that ImGeoNet outperforms the current state-of-the-art multi-view image-based method, ImVoxelNet, on all three datasets in terms of detection accuracy. In addition, ImGeoNet shows great data efficiency by achieving results comparable to ImVoxelNet with 100 views while utilizing only 40 views. Furthermore, our studies indicate that our proposed image-induced geometry-aware representation can enable image-based methods to attain superior detection accuracy than the seminal point cloud-based method, VoteNet, in two practical scenarios: (1) scenarios where point clouds are sparse and noisy, such as in ARKitScenes, and (2) scenarios involve diverse object classes, particularly classes of small objects, as in the case in ScanNet200. Project page: https://ttaoretw.github.io/imgeonet.

*Work done in Amazon.

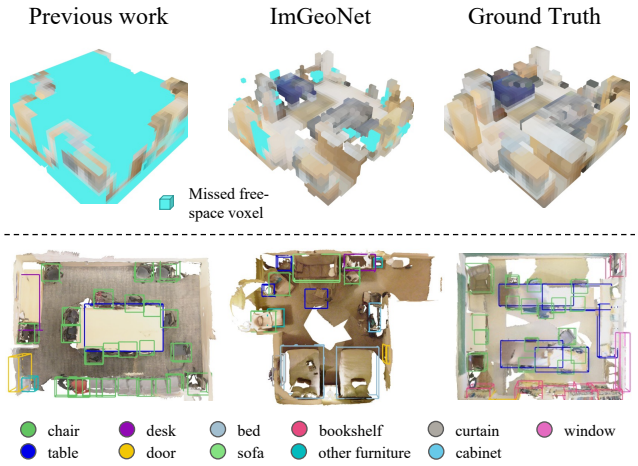

Figure 1. **Geometry-aware voxel representation.** (*Top part*) In contrast to prior works [56] (*top left*) that disregard the underlying geometry, our proposed ImGeoNet (*top center*) successfully preserves the geometric structure with respect to the ground truth (*top right*) while effectively reducing the number of voxels in free space. In the visualization of ImGeoNet, voxels with a surface probability exceeding a predefined threshold are retained, otherwise removed. The color of each voxel is determined by averaging the colors of ground truth point clouds within the voxel. Missed free-space voxels are marked as cyan. (*Bottom part*) We present the **detection results** using bounding cubes that are color-coded based on the predicted categories.

1. Introduction

Indoor 3D object detection has been an active area of computer vision research for over a decade, owing to its practical applications in robotics, augmented reality, and

mixed reality. In recent years, several studies [48, 16, 7, 85, 67, 34, 41, 14, 55] have demonstrated the effectiveness of methods based on point clouds in conjunction with deep learning techniques for indoor 3D object detection. However, the applicability of these methods is limited by their reliance on data acquired from expensive 3D sensors such as depth cameras, stereo cameras, or laser scanners. In contrast to point clouds, color images are more affordable and can capture semantically rich information akin to human vision. Therefore, image-based indoor 3D object detection is a promising research direction.

Image-based methods for indoor monocular 3D object detection [18, 19, 45, 81] have demonstrated a satisfactory level of accuracy. Nonetheless, monocular methods encounter challenges such as scale ambiguity, occlusion issues, and limited field of view. These issues can be mitigated by providing multiple perspectives of the scene, leading to a more robust and accurate 3D object detection result. Previous works [44, 56] employ multi-view images to construct a feature volume, which is subsequently utilized for conducting 3D object detection [56]. Although these methods have exhibited state-of-the-art performance, they neglect the underlying geometric characteristics during the feature volume construction.

In this work, we propose ImGeoNet, a multi-view 3D object detection framework that models a 3D space by an image-induced geometry-aware voxel representation. ImGeoNet learns to induce geometry from multi-view images to reduce the importance of voxels representing free space, and during the inference phase, only images from multiple views are required. Specifically, ImGeoNet predicts the likelihood of each voxel belonging to a surface, and subsequently weighting the feature volume according to this probability. The proposed approach exhibits a notable enhancement in detection performance owing to the successful alleviation of confusion arising from voxels in free space. Besides, a powerful pre-trained 2D feature extractor can be utilized by our representation, leading to more robust performance.

We conduct quantitative and qualitative experiments to evaluate the effectiveness of ImGeoNet on three indoor datasets, namely ARKitScenes [2], ScanNetV2 [10], and ScanNet200 [54]. The results demonstrate that ImGeoNet outperforms the state-of-the-art multi-view image-based method, ImVoxelNet [56], by 3.8%, 12.5% and 17.4% in mAP@0.25 on ARKitScenes, ScanNetV2, and ScanNet200, respectively. Additionally, ImGeoNet shows great data efficiency by achieving results comparable to ImVoxelNet with 100 views while utilizing only 40 input views. Furthermore, the results of the experiments indicate that our proposed image-induced geometry-aware representation can enable image-based methods to attain superior detection accuracy than the seminal point cloud-based method, VoteNet, in two practical scenarios: (1) scenarios where point clouds are sparse and noisy, such as in ARKitScenes, and (2) scenarios involve diverse object classes, particularly classes of small objects, as in the case in ScanNet200. Specifically, ImGeoNet outperforms VoteNet in these scenarios by at least 12.6% in terms of mAP@0.25.

The contributions of our work can be summarized as follows:

- We introduce a multi-view object detection framework that utilizes an image-induced geometry-aware voxel representation to enhance image-based 3D object detection substantially.
- Our method achieves state-of-the-art performance for image-based 3D object detection on ARKitScenes, ScanNetV2, and ScanNet200.
- Our studies demonstrate our proposed geometry-aware representation enables image-based methods to attain superior detection accuracy than the seminal point cloud-based method, VoteNet, in practical scenarios which consist of sparse and noisy point clouds or involve diverse classes.

2. Related Work

2.1. Point Cloud Based Object Detection

Since point clouds provide reliable geometric structure information, point cloud-based object detection has shown great performance on both indoor and outdoor scenes. There are two main branches, one is point-based methods directly sampling based on the set abstract and the feature propagation [50, 49, 78, 58, 77, 60, 46, 48, 85] while the other is grid-based methods based on grid representation [76, 91, 75, 25, 59, 12, 40, 14, 55].

As far as indoor object detection is concerned, the predominant methods [48, 7, 85, 67] are those relying on deep Hough voting [48] to shift surface points to their corresponding object centers. Transformer-based methods [41, 34] also deliver comparable results. Recently, sparse fully-convolutional detection methods [14, 55] have exhibited state-of-the-art performance with regard to both accuracy and efficiency. Despite the fact that point cloud-based methods perform well on object detection, they rely on costly 3D sensors, which narrows down their use cases.

2.2. Image-based Object Detection

Monocular Object Detection. There has been considerable attention paid to the field of monocular object detection due to its practicality and cost-effectiveness. Two-stage methods [62, 61, 51, 39, 35, 26] extend conventional two-stage 2D detection frameworks to estimate 3D object parameters. Single-stage anchor-based methods [3, 4, 24, 31, 36] and anchor-free methods [27, 33, 38, 42, 53, 70, 69, 83, 89, 90] predict object parameters in one stage.

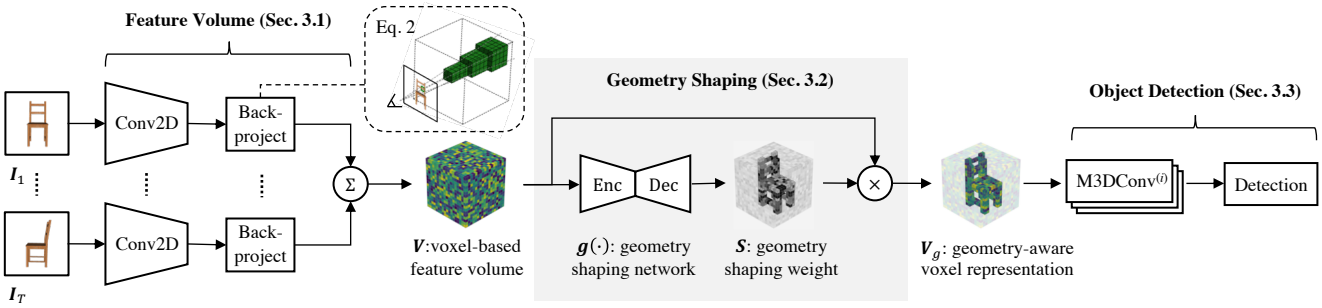

Figure 2. **An illustration of ImGeoNet framework** for 3D object detection. Given an arbitrary number of images, a 2D convolution backbone (Conv2D) is applied to extract visual features from each image, and then a 3D voxel feature volume ($\boldsymbol{V}$) is constructed by back-projecting (Eq. 2) and accumulating 2D features to the volume (Sec. 3.1). This feature volume is not ideal since the underlying geometry of the scene is not considered. Hence, geometry shaping (Sec. 3.2) is applied to weight the original feature volume by the predicted surface probabilities ($\boldsymbol{S}$), which preserves the geometric structure and removes voxels of free space. Finally, the geometry-aware volume ($\boldsymbol{V}_g$) is passed to the multiscale 3D convolutional layers (M3DConv) and the detection head (Sec. 3.3).

To mitigate the depth information loss, some approaches [52, 13, 47, 68, 74] use an additional backbone for depth map feature extraction, and other approaches [37, 71, 72] back-project depth images to 3D pseudo point clouds. The improvements highlight the significance of depth information in 3D object detection.

As for the indoor environment, some prior works [86, 87, 8, 29, 84] estimate the 3D bounding boxes based on geometry and 3D world priors. Other works [20, 22] utilize the category-specific 3D shape for detecting objects. More recently, several works [63, 19, 45, 81] view object detection as a component of scene understanding. However, they face challenges associated with monocular images, including scale ambiguity, occlusion problems and limited field of view.

Multi-view Object Detection. To better capture scene information, methods that consider multiple views has gained increasing attention in recent years. DETR-based methods [73, 32, 66] extend DETR [5] to 3D object detection. Besides, prior research [17, 28] has demonstrated that the bird-eye-view (BEV) representation is well-suited for object detection in autonomous driving scenarios. Although the aforementioned methods perform well in autonomous driving scenarios, they may not be applicable to indoor scenes, which often contain diverse object classes that are not necessarily situated on the ground. ImVoxelNet [56], on the other hand, has shown great performance in the domain of indoor 3D object detection by performing on a 3D voxel-based feature volume [44]. However, it does not properly preserve the underlying geometry of input scenes during the feature volume construction.

3. Approach

In this section, we first introduce the problem formulation, provide an overview of our method ImGeoNet and briefly explain the design concept. Next, we present the main steps of ImGeoNet (Fig. 2) in detail, which include feature volume construction (Sec. 3.1), geometry shaping (Sec. 3.2) and object detection (Sec. 3.3).

Problem Formulation. Given an arbitrary number of input images captured in the same scene $\{\boldsymbol{I}_t\} \subseteq \mathbb{R}^{H\times W\times 3}$ and their corresponding intrinsic matrices $\{\boldsymbol{K}_t\} \subseteq \mathbb{R}^{3\times 3}$ and poses $\{\boldsymbol{T}_t\} \subseteq \mathrm{SE}(3)$, the goal of 3D object detection is to identify target objects by predicted categories and enclosing bounding boxes $\{\boldsymbol{b}\} \subseteq \mathbb{R}^7$. A bounding box is parameterized by (x, y, z, w, h, l, ϕ), where (x, y, z) is the center, (w, h, l) is the size and ϕ is the yaw angle. We follow a common assumption [48, 56] that bounding boxes are on the ground plane, so only yaw angles are predicted.

Framework Overview. ImGeoNet aims to predict the 3D bounding boxes and corresponding object categories for the target objects present in the scene, using an arbitrary number of input images.

First of all, ImGeoNet constructs a 3D voxel feature volume by back-projecting and accumulating 2D features to the volume as in [44, 56]. The 2D feature of a pixel is duplicated to voxels along the ray emitted from the camera center through the pixel. The 3D voxel volume obtained in this process is suboptimal, as it can lead to contamination of voxels in free space and hinder the precision of detection.

To turn the constructed voxel volume into a geometry-aware representation, ImGeoNet performs geometry shaping. In this step, ImGeoNet weights each voxel feature according to the probability of that voxel being located on an object's surface. Consequently, voxels situated in free space will be assigned a lower weight, whereas those located on object surfaces will retain a higher weight and incorporate information from various viewpoints. This significantly improves the accuracy of the predicted bounding box. During training, we convert the ground-truth point clouds to surface

voxels to supervise geometry shaping network. Finally, we follow previous works [56, 65] to predict the bounding box for each voxel and perform non-maximal suppression to reduce redundant predictions.

3.1. Feature Volume

Attaining precise 3D object detection requires a thorough comprehension of the geometric structure inherent to a given scene. The feature volume approaches have been demonstrated to be highly effective in tasks that necessitate an extensive understanding of scene geometry, such as stereo matching [21, 79], surface reconstruction [11, 43, 80], and novel view synthesis [6, 64]. Consequently, we adopt the feature volume representation to describe a scene, and leverage a pre-trained 2D feature extractor for more robust performance.

We compute the feature volume $\boldsymbol{V} \in \mathbb{R}^{H_v \times W_v \times D_v \times C}$ from a sequence of images $\boldsymbol{I}_t$ with known camera intrinsics $\boldsymbol{K}_t$ and poses $\boldsymbol{T}_t$. Here H_v, W_v and D_v denote the side lengths of the volume in terms of the voxel size unit, while C represents the feature dimension. We first extract 2D features by 2D convolutional backbone from input images

$$\boldsymbol{F}_t = \text{Backbone2D}(\boldsymbol{I}_t), \tag{1}$$

where $\boldsymbol{F}_t \in \mathbb{R}^{H \times W \times C}$ and strides in convolution layers are ignored for simplicity. Next, the 2D features are back-projected to the volume by

$$\boldsymbol{V}_t[x, y, z, :] = \boldsymbol{F}_t[u, v, :], \tag{2}$$

where $[:]$ is the slice operator and the pixel coordinates (u, v) are computed from the voxel centers (x, y, z) by the pinhole camera model as follows:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \frac{1}{\lambda} \boldsymbol{K}_t \boldsymbol{\Pi}_0 \boldsymbol{T}_t \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}, \quad \text{where } \boldsymbol{\Pi}_0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \tag{3}$$

and λ is the distance along optical axis between the voxel center and the camera center. In practice, we scan over the voxel centers and retrieve the corresponding back-projected 2D features. If any voxels happen to be situated outside the view frustum, their features are assigned a value of zero. Finally, the back-projected volumes $\boldsymbol{V}_t$ computed from different views in Eq. 2 are averaged to construct the feature volume $\boldsymbol{V}$ by

$$\boldsymbol{V} = \left(\frac{1}{\sum_t \boldsymbol{M}_t}\right) \odot \left(\sum_t \boldsymbol{V}_t \odot \boldsymbol{M}_t\right), \tag{4}$$

where $\odot$ is the Hadamard product, and $\boldsymbol{M}_t$ is a binary mask indicating whether the voxels are in the view frustum of $\boldsymbol{I}_t$.

It is worth noting that the feature volume obtained through the mentioned construction process lacks information on the geometric structure of the scene. Specifically, each voxel along a camera ray is assigned the same feature value as its corresponding pixel, regardless of whether the voxel is located on the closest surface along that ray. Therefore, to properly incorporate the scene geometry, we introduce *Geometry Shaping*.

3.2. Geometry Shaping

One significant shortcoming of the derived feature volume from Sec. 3.1 is its geometry-unaware nature. In other words, even the voxels that are not on any object surface are still assigned values. The situation worsens when the voxels are in free space, where the detection module can be perturbed and generate false predictions. To address this, we propose *geometry shaping*, which leverages the multi-view image input to induce geometry structure and remove noisy voxels in free space by down-weighting unoccupied voxels while preserving voxels on surfaces.

Since appearance variance reveals certain geometric information [79, 6], we also take the feature variance into consideration. To be specific, we concatenate feature variance (refer to Eq. 8 in Sec. 4) with the feature volume to obtain $\boldsymbol{V}^{'}$. Subsequently, a geometry shaping volume $\boldsymbol{S}$ is generated via the geometry shaping network $g(\cdot)$ by

$$\boldsymbol{S} = g(\boldsymbol{V}^{'}), \tag{5}$$

where $\boldsymbol{S}$ shares the same grid size as the combined feature volume $\boldsymbol{V}^{'}$, and each element of $\boldsymbol{S}$ is the likelihood of the voxel being on an object surface. Note that $g(\cdot)$ employs the same feature volume (prior to geometry shaping) as the final 3D object detector, resulting in less overhead compared to computing Multi-View Stereo (MVS) from raw images. Afterward, the geometry-aware feature volume is derived by weighting the feature volume as follows:

$$\boldsymbol{V}_g = \boldsymbol{S} \odot \boldsymbol{V}. \tag{6}$$

Since the weights will be low in free space, the resulting geometry-aware feature volume mainly remains values on object surfaces, which could better describe the underlying geometry. As a result, the geometry shaping reduces the burden of the final detection to a great extent and improves precision.

In our implementation, we convert the RGB-D frames to point clouds and consider the voxels that contain at least one point as surface voxels. Besides, for each camera ray, we also consider locations neighboring surface voxels within a margin of ϵ as positive. Subsequently, we proceed to supervise the geometry shaping network through surface voxel prediction using focal loss [30] in an end-to-end way. It is worth noting that depth sensory data is solely employed to

supervise the geometry shaping network during the training phase, while only images from multiple views are utilized during the inference phase.

3.3. Object Detection

Even though the obtained feature volume in Eq. 6 is geometry-aware, it may still have limitations in capturing objects of varying scales. Therefore, we transform the geometry-aware volume by multiscale dense 3D convolution layers:

$$\boldsymbol{V}_h^{(i)} = \text{M3DConv}^{(i)}(\boldsymbol{V}_g), \tag{7}$$

where $i \in \{0, 1, ..., L-1\}$ is the scale index, and the volume of different scales will have different grid sizes.

As regards the detection head, we follow ImVoxelNet [56] to extend the single-stage anchor-free 2D detectors [65, 82] to 3D volume. All locations from L scales are considered, and for each location, a class probability, center-ness, and a 3D bounding box are predicted. However, only a few locations are selected as positive samples for supervision during training: 1) Locations not in any target bounding box are removed. 2) For each target object, only locations from the most fitting scale are kept. Specifically, we choose the smallest scale that contains more than M points. 3) For each target object, we only keep the top-k locations close to the bounding box center. 4) If a point corresponds to multiple targets, we choose the one with minimal volume. Finally, we use focal loss [30] for category classification, cross-entropy loss for center-ness estimation [65], and rotated 3D IoU loss [88] for bounding box prediction.

4. Implementation

Geometry Shaping Network. As illustrated in Fig. 3, our geometry shaping network has an encoder-decoder architecture with residual connections. Specifically, each encoder comprises of three 3D convolutional layers, while each decoder comprises of one transposed 3D convolutional layer and one 3D convolutional layer. We set the kernel sizes of all the 3D convolutional layers to 3, and set the kernel size of the transposed 3D convolutional layer to 2. Following each convolutional and transposed convolutional layer, we apply a batch normalization and ReLU activation. During encoding, the spatial sizes are reduced by a factor of 2, while the channel size is increased by a factor of 2. In contrast, during decoding, the spatial sizes are increased, and the channel size is reduced. Finally, we adopt a linear projection layer to reduce the channel size to 1, and we obtain the final output by passing the reduced tensor to a sigmoid function.

Framework Architecture. A ResNet50 [15] pre-trained on ImageNet [57] is used as the 2D feature extractor. The surface voxel margin ϵ is set to 4 voxels. The M3DConv network is the same as the 3D convolutional network before the detection head in ImVoxelNet [56]. As for the detection head, we follow the same previous work [56] and set the number of scales L to 3, the minimum fitting points M to 27 and k in the top-k selection to 18. It is worth mentioning that ImGeoNet is trained in an end-to-end way. Specifically, the geometry shaping network and all the remaining networks are jointly optimized for both surface voxel prediction and 3D object detection.

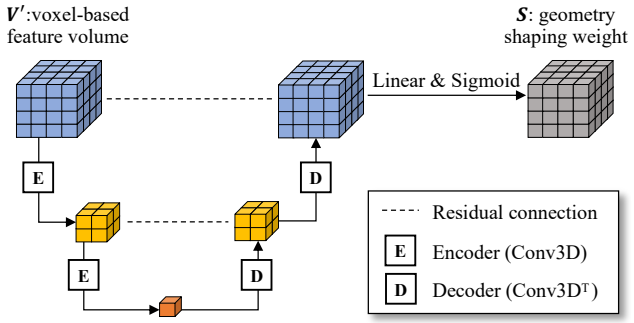

Figure 3. **The network architecture of geometry shaping network**. It is based on an encoder-decoder architecture with residual connections, followed by a linear projection layer and a sigmoid function.

Feature Variance. We compute the feature variance across different views for each voxel as follows:

$$\boldsymbol{V}_{\text{var}} = \left(\frac{1}{\sum_t \boldsymbol{M}_t} \odot \left(\sum_t (\boldsymbol{V}_t)^2 \odot \boldsymbol{M}_t \right) \right) - (\boldsymbol{V}_{\text{mean}})^2, \tag{8}$$

where $\boldsymbol{V}_{\text{mean}}$ is computed via Eq. 4 in the paper and we utilize the fact that $\text{Var}(X) = \text{E}[X^2] - \text{E}[X]^2$. Subsequently, our geometry shaping network utilizes the concatenated values of $\boldsymbol{V}_{\text{mean}}$ and $\boldsymbol{V}_{\text{var}}$ as input for predicting surface voxels.

Loss Configuration. Both the surface voxel prediction loss and category classification loss are focal losses with a gamma value of 2 and an alpha value of 0.25. The center-ness estimation loss is a cross-entropy loss, and the bounding box prediction loss is a rotated 3D IoU loss. All the loss weights are set to 1, except for the surface voxel prediction loss, which is assigned a weight of 10. It is worth mentioning that we employed identical loss hyperparameters across all experiments.

5. Experiment

We describe the datasets, evaluation metric, and implementation details in Sec. 5.1. Then, we present and analyze the results of our experiments in Sec. 5.2.

5.1. Setup

Dataset. Our approach is evaluated on three indoor multi-view datasets, each serving a distinct purpose. Firstly, to

examine the feasibility of the proposed method in a realistic scenario, we use ARKitScenes [2] which contains sensor data captured by popular mobile devices and is the most realistic of the three datasets. Secondly, ScanNetV2 [10] is used as it is the most widely adopted benchmark for comparing against other state-of-the-art methods. Lastly, to investigate the model performance in a setting with diverse classes of varying sizes and properties, ScanNet200 [54] is adopted since it comprises the largest number of object categories.

ARKitScenes [2] is a large real-world RGB-D video dataset captured with handheld 2020 Apple iPad Pros. It comprises 5,047 captures of 1,661 distinct scenes, and we follow the official split to separate it into 4,498 and 549 captures for training and testing, respectively. The image resolution is 192×256. We uniformly sample the views for each scene based on the frame indices. The number of sampled views for training is 50 and 200 for the image-based methods and point cloud-based methods, respectively. To acquire the point clouds, we back-project the sampled views to 3D space based on the supplied low-resolution depth maps. We filter the points through voxel downsampling where the voxel size is set to 0.02 meters. Finally, it is noteworthy that the quality of point clouds in ARKitScenes is inferior to that of ScanNetV2 since the depth maps in ARKitScenes are low-resolution while the point clouds in ScanNetV2 are high-resolution and derived from the 3D reconstructed meshes.

ScanNetV2 [10] is a richly annotated RGB-D video dataset of 3D reconstructed meshes of indoor scenes. It contains 2.5 million views in 1,513 room-level scenes and there are 18 classes available for classification. The resolution of the images is 968×1296. We follow the public train-test split originally proposed by ScanNet, which allocates 1,201 scenes for training and 312 scenes for testing. Since the standard release does not provide oriented bounding box annotation, we follow [16, 48] to create axis-aligned bounding boxes according to the semantic labels of mesh vertices. To train and evaluate the image-based methods, we evenly sample 50 views for each scene based on the frame indices and resize the images to 480 × 640.

ScanNet200 [54] extends ScanNetV2 to address a larger-vocabulary setting and provides 200 object categories for classification. The categories are divided into head, common, and tail groups, consisting of 66, 68, and 66 classes, respectively, based on the number of labeled surface points. Since smaller objects tend to have fewer surface points than larger ones, the average object size decreases from the head to the tail group (refer to Appendix. A.1). We adopt the same train-test split as the one employed in ScanNet, and only objects of categories existing in both the training and the test sets are considered as in the official benchmark script.

Evaluation Metric. We adopt the mean average precision (mAP) to evaluate the detection accuracy. Specifically, we use mAP@0.25 and mAP@0.5 where the numbers indicate the 3D intersection over union (IoU) thresholds. To explain, the threshold is the minimum IoU to determine a positive match, which means mAP@0.5 is stricter on the evaluation of object location than mAP@0.25.

Baselines. We compare ImGeoNet with ImVoxelNet [56], current state-of-the-art multi-view image-based 3D detector, and VoteNet [48], a seminal point cloud-based method. To reproduce ImVoxelNet, we use the code provided by the original authors [48]. As for VoteNet, we reproduce it with a common codebase [9] and implement the same hyperparameters and model architecture as the original work [48].

Optimization. The Adam optimizer [23] with a learning rate 0.0001 is used for training. The weight decay factor is 0.0001. Gradient clipping is also used and the max norm is set to 35. For ScanNetV2 [10] and ARKitScenes [2], the learning rate is reduced by ten times before the 9^{th} and 12^{th} epoch. For ScanNet200 [54], the learning rate is reduced by ten times before the 9^{th} and 30^{th} epoch. The total numbers of epochs are 12, 12 and 30 for ScanNetV2, ARKitScenes, and ScanNet200, respectively.

Voxel Volume. The feature volume is 6.4 × 6.4 × 2.56 meters. The voxel size is 0.16 meters for ScanNetV2 and ARKitScenes, but 0.08 meters for ScanNet200 for detecting small objects. For ScanNetV2 and ScanNet200, we shift the volume origin to the coordinate origin, whereas for ARKitScenes, we relocate it to the center of the sampled poses.

Table 1. **3D object detection results on ARKitScenes** are presented to demonstrate the practicability of ImGeoNet in real-world scenarios. We also include the result (a-4) reported by [2] for reference. The notations RGB and PC signify the use of color images and point clouds, respectively, during the inference phase.

	Input		Performance (mAP)	
Method	RGB	PC	@0.25	@0.5
(a-1) ImGeoNet (ours)	✓	-	**60.2**	**43.4**
(a-2) ImVoxelNet [56]	✓	-	58.0	38.8
(a-3) VoteNet [48]	-	✓	53.3	38.5
(a-4) VoteNet (from [2])	-	✓	35.8	-

5.2. Results

Realistic Mobile Capture. First of all, we conduct an experiment on ARKitScenes which is captured by popular mobile devices. The results are presented in Table. 1 and Fig. 4. In the present scenario, wherein a test set comprising 50 evenly sampled views, ImGeoNet (a-1) attains the best performance in mAP@0.25 and mAP@0.5. Through a comparison of ImGeoNet (a-1) with the current state-of-

Table 2. **3D object detection results on ScanNetV2**. In this benchmark, ImGeoNet outperforms the SOTA multi-view image-based method, ImVoxelNet, by 12.5% and 19.3% in mAP@0.25 and mAP@0.5, respectively. In addition, the performance of ImGeoNet with 50 views is not far from that of VoteNet, even though VoteNet leverages reconstructed meshes derived from typically more than 1000 viewpoints to obtain high-quality point clouds for its operations.

	Input		Performance (mAP)	
Method	RGB	PC	@0.25	@0.5
(s-1) ImGeoNet (ours)	✓	-	**54.8**	**28.4**
(s-2) ImVoxelNet [56]	✓	-	48.7	23.8
(s-3) 3D-SIS [16]	-	✓	25.4	14.6
(s-4) 3D-SIS [16]	✓	✓	40.2	22.5
(s-5) VoteNet [48]	-	✓	58.6	33.5

Table 3. **3D object detection results on ScanNet200** are presented to examine the robustness of ImGeoNet for diverse classes. As in [54], categories are divided into head, common and tail groups where the average object sizes decrease from the head to the tail group.

		Performance (mAP@0.25)			
Method	Input	Total	Head	Comm	Tail
(d-1) ImGeoNet (ours)	RGB	**22.3**	**38.1**	**17.3**	**9.7**
(d-2) ImVoxelNet [56]	RGB	19.0	34.1	14.0	7.7
(d-3) VoteNet [48]	PC	19.8	38.5	16.0	2.9

the-art multi-view image-based method, ImVoxelnet (a-2), we demonstrate that the geometry-aware representation utilized in ImGeoNet is effective in a practical mobile environment, where the depth maps used for training the geometry shaping network are not entirely accurate. Additionally, both image-based approaches (a-1 and a-2) exhibit superior performance compared to the seminal point cloud-based method, VoteNet (a-3). This highlights the practical preference for the use of images in real-world scenarios.

Comparison on ScanNetV2. Secondly, we compare our method with state-of-the-art methods on ScanNetV2, a well-known object detection benchmark for indoor scenes. The results are presented in Table. 2 and Fig. 4. It can be observed from the results that ImGeoNet (s-1) exhibits superior performance in comparison to ImVoxelNet (s-2), the current state-of-the-art multi-view image-based method. Specifically, ImGeoNet outperforms ImVoxelNet by 12.5% and 19.3% in mAP@0.25 and mAP@0.5, respectively. The efficacy of our proposed geometry-aware representation is empirically validated through the significant improvement in mAP achieved by ImGeoNet. On the other hand, ImGeoNet (s-1) outperforms the point cloud-based baseline 3D-SIS (s-3) and its variant incorporating image data (s-4). Furthermore, ImGeoNet successfully reduces the performance gap between the seminal point cloud-based method (s-5) and image-based methods to a significant extent. These are notable considering that point cloud-based methods rely on 3D reconstructed meshes that are obtained from a multitude of viewpoints (typically 1000+) to acquire high-quality point clouds for their operations, while ImGeoNet only utilizes 50 views.

Diverse Classes. To inspect the capability of the proposed method for a diverse range of object classes, we conduct an analysis on ScanNet200, and the results are presented in Table. 3 and Fig. 4. In this scenario, ImGeoNet (d-1) attains superior performance over ImVoxelNet (d-2) across all category groups, reaffirming the effectiveness of our geometry-aware volume representation in scenes containing objects of diverse classes. Furthermore, ImGeoNet (d-1) exhibits superior performance over VoteNet (d-3), particularly for the common and tail category groups, which are characterized by smaller average object sizes in contrast to the head category group (refer to Appendix. A.1). This achievement is remarkable, given that VoteNet utilizes vertices from high-quality 3D reconstructed meshes obtained from a great number of views (typically 1000+), while ImGeoNet directly takes images from 50 viewpoints as input. We conjecture there are two main reasons for it: (1) The downsampling technique employed in the point cloud-based baseline has a tendency to exclude small instances [77], thereby impeding the detection performance. (2) ImGeoNet primarily relies on the geometry-aware representation induced by images, which effectively leverages the visual features extracted from a 2D backbone pre-trained on examples containing objects of various sizes. As a result, ImGeoNet exhibits greater robustness for small objects.

Number of Views. In order to investigate the impact of varying numbers of views, we utilize different numbers of views in the construction of feature volume for image-based methods, as well as in the generation of point clouds for point cloud-based methods. We choose ARKitScenes as the dataset as its captured scenes are more representative of real-world scenarios. The results are presented in Table. 4. First of all, we can observe that ImGeoNet (v-1) with only 30 views outperforms VoteNet (v-3) with 100 views. Secondly, with only 40 views, the proposed geometry shaping enables ImGeoNet (v-1) to achieve comparable performance to ImVoxelNet (v-2) with 100 views. The effectiveness of ImGeoNet in utilizing data is particularly valuable in scenarios where only a limited number of views can be obtained.

Model Speed. Regarding inference time, we present a comparison between ImGeoNet and ImVoxelNet on ARKitScenes in Table. 5. The experiment is run by a single Nvidia 3090 GPU and the data loading time is ignored. It

Table 4. **3D object detection results for varying numbers of views** on ARKitScenes. ImGeoNet shows great data efficiency by achieving results comparable to ImVoxelNet with 100 views while utilizing only 40 input views.

Method	Input		Performance (mAP@0.25 / mAP@0.5)						
	RGB	PC	10 views	20 views	30 views	40 views	50 views	75 views	100 views
(v-1) ImGeoNet (ours)	✓	-	**39.0 / 21.9**	**53.1 / 34.3**	**57.1 / 39.2**	**59.5 / 42.7**	**60.2 / 43.4**	**61.8 / 45.0**	**62.4 / 45.7**
(v-2) ImVoxelNet [56]	✓	-	36.2 / 19.6	50.5 / 30.6	54.6 / 35.2	57.4 / 37.9	58.0 / 38.8	58.8 / 40.5	59.7 / 42.0
(v-3) VoteNet [48]	-	✓	30.2 / 20.8	45.9 / 31.5	50.2 / 34.0	51.1 / 36.8	53.3 / 38.5	53.6 / 38.3	53.9 / 39.0

Table 5. **Inference time for different numbers of views**. The experiment is run on ARKitScenes with a single Nvidia 3090 GPU. The data loading time is ignored and the inference times are averaged over all test scenes.

Method	Inference Time (ms)			
	20 views	40 views	50 views	100 views
(t-1) ImGeoNet (ours)	139.0	166.1	181.8	245.8
(t-2) ImVoxelNet [56]	113.1	140.0	155.9	219.7

Table 6. **Effectiveness of the geometry shaping**. We conduct this study on ScanNetV2. Despite that (g-3) ImVoxelNet with MaGNet, a SOTA multi-view depth estimator, takes 5 times more input views, has 42% larger model size and 15.7 times longer runtime, ImGeoNet has better detection results. The model size and the inference time of ImGeoNet on ScanNetV2 are 485.6 MB and 489.9 ms, respectively.

Method	Relative		mAP	
	Size	Runtime	@0.25	@0.5
(g-1) ImGeoNet (ours)	1.0	1.0	**54.8**	**28.4**
(g-2) ImVoxelNet [56]	0.82	0.85	48.7	23.8
(g-3) w/ MaGNet [1]	1.42	15.72	53.8	28.2
(g-4) w/ GT depth	0.82	0.89	58.8	33.4

can be observed that ImGeoNet (t-1) is slightly slower than ImVoxelNet (t-2) when using the same number of views. However, as previously mentioned, the performance of ImGeoNet (v-1) with only 40 views is comparable to that of ImVoxelNet (v-2) with 100 views. Notably, our method (t-1) achieves this level of performance with a significantly shorter inference time than ImVoxelNet (t-2) by 53.6 ms (a 24% relative speed-up). This finding highlights the effectiveness of the proposed geometry-aware representation, as it demonstrates that ImGeoNet can achieve a large performance improvement while only incurring a slight increase in running time.

Effectiveness of Geometry Shaping Network. To highlight the light overhead of the proposed Geometry Shaping Network ($g(\cdot)$ in Eq. 5), we extend ImVoxelNet with estimated depth (g-3) generated by MaGNet [1] (referred to as cascade baseline), the state-of-the-art multi-view depth estimator, and present the results on ScanNetV2 in Table. 6. Although both ImGeoNet (g-1) and the cascade baseline (g-3) achieved similar levels of performance, the cascade baseline requires more resources. Specifically, MaGNet [1] requires four close vicinity views to produce a reliable depth map, resulting in five times more input views than ImGeoNet. Additionally, the cascade baseline has a total model size that is 42% larger than ImGeoNet. Furthermore, since MaGNet does not share features with ImVoxelNet, the inference time of the cascade baseline is 15.7 times longer than ImGeoNet.

On the other hand, ImVoxelNet with ground-truth depth (g-4) can be regarded as the upper limit. A sizable gap between ImGeoNet (g-1) and the upper bound (g-4) can be observed, which indicates that there is room for improving Geometry Shaping Network in the future. Finally, we conduct a similar experiment on ARKitScenes, which shows the performance of the upper limit is 62.2 and 46.4 in mAP@0.25 and mAP@0.5, respectively. The performance gaps between the upper limit and ImGeoNet in ARKitScenes (2.0/3.0 for mAP@0.25/@0.5) is smaller than those (4.0/5.0 for mAP@0.25/@0.5) in ScanNetV2. This observation serves to corroborate the effectiveness of the Geometry Shaping Network in real-world scenarios where the point clouds exhibit sparsity and noise.

6. Conclusion

In this work, we have introduced ImGeoNet, a 3D object detection framework that utilizes a geometry-aware voxel representation induced by multi-view images to model a 3D space. Since ImGeoNet learns to predict geometry from multi-view images, a pre-trained 2D feature extractor can be leveraged and only images from multiple views are required during the inference phase. Through in-depth quantitative and qualitative experiments, we have demonstrated the effectiveness of our proposed geometry-aware representation by (1) achieving state-of-the-art results in image-based indoor 3D object detection, (2) showing great data efficiency by achieving great accuracy with fewer views, and (3) enabling image-based methods to attain superior detection accuracy than a seminal point cloud-based approach in practical scenarios with sparse and noisy point clouds or diverse object classes.

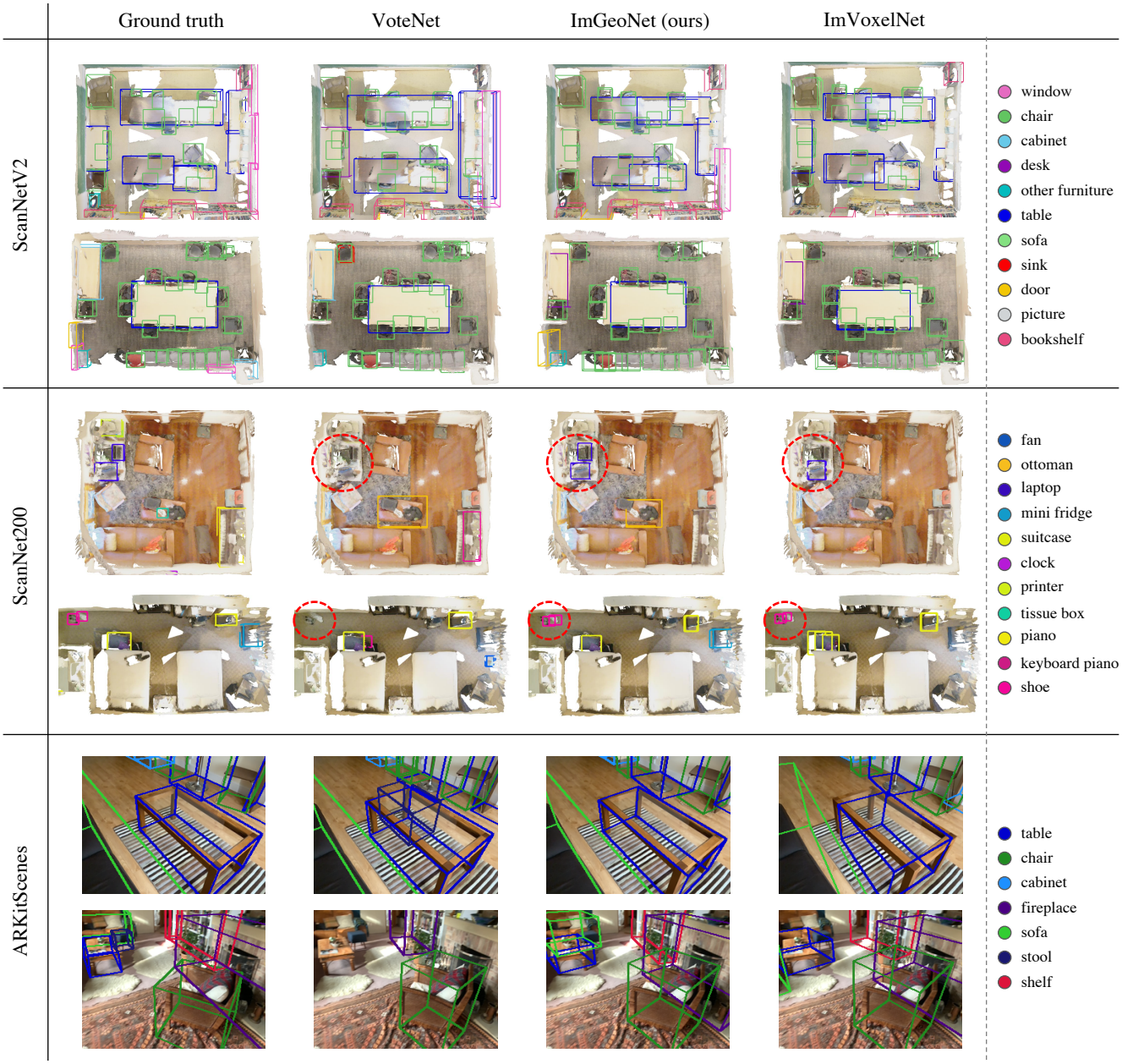

Figure 4. **Qualitative results of 3D object detection** on ScanNetV2, ScanNet200 and ARKitScenes. For ScanNet200, only objects in common or tail groups are presented. ImGeoNet outperforms the multi-view image-based SOTA method, ImVoxelNet, on all three datasets. Compared to the seminal point cloud-based method, VoteNet, ImGeoNet has superior results on small objects such as laptops and shoes (see the red circles in ScanNet200). Besides, in a more realistic mobile scenario (ARKitScenes), ImGeoNet yields the most precise outcome.

7. Acknowledgements

This work is supported in part by Ministry of Science and Technology of Taiwan (NSTC 111-2634-F-002-022). We thank National Center for High-performance Computing (NCHC) for computational and storage resource.

References

[1] Gwangbin Bae, Ignas Budvytis, and Roberto Cipolla. Multi-view depth estimation by fusing single-view depth probability with multi-view geometry. In *CVPR*, 2022. 8

[2] Gilad Baruch, Zhuoyuan Chen, Afshin Dehghan, Tal Dimry, Yuri Feigin, Peter Fu, Thomas Gebauer, Brandon Joffe, Daniel Kurz, Arik Schwartz, et al. Arkitscenes–a diverse

real-world dataset for 3d indoor scene understanding using mobile rgb-d data. *arXiv preprint arXiv:2111.08897*, 2021. 2, 6

[3] Garrick Brazil and Xiaoming Liu. M3d-rpn: Monocular 3d region proposal network for object detection. In *ICCV*, 2019. 2

[4] Garrick Brazil, Gerard Pons-Moll, Xiaoming Liu, and Bernt Schiele. Kinematic 3d object detection in monocular video. In *ECCV*, 2020. 2

[5] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020. 3

[6] Anpei Chen, Zexiang Xu, Fuqiang Zhao, Xiaoshuai Zhang, Fanbo Xiang, Jingyi Yu, and Hao Su. Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo. In *ICCV*, 2021. 4

[7] Bowen Cheng, Lu Sheng, Shaoshuai Shi, Ming Yang, and Dong Xu. Back-tracing representative points for voting-based 3d object detection in point clouds. In *CVPR*, 2021. 2

[8] Wongun Choi, Yu-Wei Chao, Caroline Pantofaru, and Silvio Savarese. Understanding indoor scenes using 3d geometric phrases. In *CVPR*, 2013. 3

[9] MMDetection3D Contributors. MMDetection3D: OpenMMLab next-generation platform for general 3D object detection. https://github.com/open-mmlab/mmdetection3d, 2020. 6

[10] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, 2017. 2, 6

[11] Angela Dai, Daniel Ritchie, Martin Bokeloh, Scott Reed, Jürgen Sturm, and Matthias Nießner. Scancomplete: Large-scale scene completion and semantic segmentation for 3d scans. In *CVPR*, 2018. 4

[12] Jiajun Deng, Shaoshuai Shi, Peiwei Li, Wengang Zhou, Yanyong Zhang, and Houqiang Li. Voxel r-cnn: Towards high performance voxel-based 3d object detection. In *AAAI*, 2021. 2

[13] Mingyu Ding, Yuqi Huo, Hongwei Yi, Zhe Wang, Jianping Shi, Zhiwu Lu, and Ping Luo. Learning depth-guided convolutions for monocular 3d object detection. In *CVPRW*, 2020. 3

[14] JunYoung Gwak, Christopher Choy, and Silvio Savarese. Generative sparse detection networks for 3d single-shot object detection. In *ECCV*, 2020. 2

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 5

[16] Ji Hou, Angela Dai, and Matthias Nießner. 3d-sis: 3d semantic instance segmentation of rgb-d scans. In *CVPR*, 2019. 2, 6, 7

[17] Junjie Huang, Guan Huang, Zheng Zhu, and Dalong Du. Bevdet: High-performance multi-camera 3d object detection in bird-eye-view. *arXiv preprint arXiv:2112.11790*, 2021. 3

[18] Siyuan Huang, Yixin Chen, Tao Yuan, Siyuan Qi, Yixin Zhu, and Song-Chun Zhu. Perspectivenet: 3d object detection from a single rgb image via perspective points. In *NeurIPS*, 2019. 2

[19] Siyuan Huang, Siyuan Qi, Yinxue Xiao, Yixin Zhu, Ying Nian Wu, and Song-Chun Zhu. Cooperative holistic scene understanding: Unifying 3d object, layout, and camera pose estimation. In *NeurIPS*, 2018. 2, 3

[20] Siyuan Huang, Siyuan Qi, Yixin Zhu, Yinxue Xiao, Yuanlu Xu, and Song-Chun Zhu. Holistic 3d scene parsing and reconstruction from a single rgb image. In *ECCV*, 2018. 3

[21] Sunghoon Im, Hae-Gon Jeon, Stephen Lin, and In So Kweon. Dpsnet: End-to-end deep plane sweep stereo. *arXiv preprint arXiv:1905.00538*, 2019. 4

[22] Hamid Izadinia, Qi Shan, and Steven M Seitz. Im2cad. In *CVPR*, 2017. 3

[23] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[24] Abhinav Kumar, Garrick Brazil, and Xiaoming Liu. Groomed-nms: Grouped mathematically differentiable nms for monocular 3d object detection. In *CVPR*, 2021. 2

[25] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *CVPR*, 2019. 2

[26] Buyu Li, Wanli Ouyang, Lu Sheng, Xingyu Zeng, and Xiaogang Wang. Gs3d: An efficient 3d object detection framework for autonomous driving. In *CVPR*, 2019. 2

[27] Peixuan Li, Huaici Zhao, Pengfei Liu, and Feidao Cao. Rtm3d: Real-time monocular 3d detection from object keypoints for autonomous driving. In *ECCV*, 2020. 2

[28] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Yu Qiao, and Jifeng Dai. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. In *ECCV*, 2022. 3

[29] Dahua Lin, Sanja Fidler, and Raquel Urtasun. Holistic scene understanding for 3d object detection with rgbd cameras. In *ICCV*, 2013. 3

[30] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017. 4, 5

[31] Lijie Liu, Jiwen Lu, Chunjing Xu, Qi Tian, and Jie Zhou. Deep fitting degree scoring network for monocular 3d object detection. In *CVPR*, 2019. 2

[32] Yingfei Liu, Tiancai Wang, Xiangyu Zhang, and Jian Sun. Petr: Position embedding transformation for multi-view 3d object detection. In *ECCV*, 2022. 3

[33] Zechen Liu, Zizhang Wu, and Roland Tóth. Smoke: Single-stage monocular 3d object detection via keypoint estimation. In *CVPRW*, 2020. 2

[34] Ze Liu, Zheng Zhang, Yue Cao, Han Hu, and Xin Tong. Group-free 3d object detection via transformers. In *ICCV*, 2021. 2

[35] Yan Lu, Xinzhu Ma, Lei Yang, Tianzhu Zhang, Yating Liu, Qi Chu, Junjie Yan, and Wanli Ouyang. Geometry uncertainty projection network for monocular 3d object detection. In *ICCV*, 2021. 2

[36] Shujie Luo, Hang Dai, Ling Shao, and Yong Ding. M3dssd: Monocular 3d single stage object detector. In *CVPR*, 2021. 2

[37] Xinzhu Ma, Zhihui Wang, Haojie Li, Pengbo Zhang, Wanli Ouyang, and Xin Fan. Accurate monocular 3d object detection via color-embedded 3d reconstruction for autonomous driving. In *ICCV*, 2019. 3

[38] Xinzhu Ma, Yinmin Zhang, Dan Xu, Dongzhan Zhou, Shuai Yi, Haojie Li, and Wanli Ouyang. Delving into localization errors for monocular 3d object detection. In *CVPR*, 2021. 2

[39] Fabian Manhardt, Wadim Kehl, and Adrien Gaidon. Roi-10d: Monocular lifting of 2d detection to 6d pose and metric shape. In *CVPR*, 2019. 2

[40] Jiageng Mao, Yujing Xue, Minzhe Niu, Haoyue Bai, Jiashi Feng, Xiaodan Liang, Hang Xu, and Chunjing Xu. Voxel transformer for 3d object detection. In *ICCV*, 2021. 2

[41] Ishan Misra, Rohit Girdhar, and Armand Joulin. An end-to-end transformer model for 3d object detection. In *ICCV*, 2021. 2

[42] Arsalan Mousavian, Dragomir Anguelov, John Flynn, and Jana Kosecka. 3d bounding box estimation using deep learning and geometry. In *CVPR*, 2017. 2

[43] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM TOG*, 2022. 4

[44] Zak Murez, Tarrence Van As, James Bartolozzi, Ayan Sinha, Vijay Badrinarayanan, and Andrew Rabinovich. Atlas: End-to-end 3d scene reconstruction from posed images. In *ECCV*, 2020. 2, 3

[45] Yinyu Nie, Xiaoguang Han, Shihui Guo, Yujian Zheng, Jian Chang, and Jian Jun Zhang. Total3dunderstanding: Joint layout, object pose and mesh reconstruction for indoor scenes from a single image. In *CVPR*, 2020. 2, 3

[46] Xuran Pan, Zhuofan Xia, Shiji Song, Li Erran Li, and Gao Huang. 3d object detection with pointformer. In *CVPR*, 2021. 2

[47] Dennis Park, Rares Ambrus, Vitor Guizilini, Jie Li, and Adrien Gaidon. Is pseudo-lidar needed for monocular 3d object detection? In *ICCV*, 2021. 3

[48] Charles R Qi, Or Litany, Kaiming He, and Leonidas J Guibas. Deep hough voting for 3d object detection in point clouds. In *ICCV*, 2019. 2, 3, 6, 7, 8

[49] Charles R Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J Guibas. Frustum pointnets for 3d object detection from rgb-d data. In *CVPR*, 2018. 2

[50] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *NeurIPS*, 2017. 2

[51] Zengyi Qin, Jinglu Wang, and Yan Lu. Monogrnet: A geometric reasoning network for monocular 3d object localization. In *AAAI*, 2019. 2

[52] Cody Reading, Ali Harakeh, Julia Chae, and Steven L Waslander. Categorical depth distribution network for monocular 3d object detection. In *CVPR*, 2021. 3

[53] Thomas Roddick, Alex Kendall, and Roberto Cipolla. Orthographic feature transform for monocular 3d object detection. *arXiv preprint arXiv:1811.08188*, 2018. 2

[54] David Rozenberszki, Or Litany, and Angela Dai. Language-grounded indoor 3d semantic segmentation in the wild. In *ECCV*, 2022. 2, 6, 7

[55] Danila Rukhovich, Anna Vorontsova, and Anton Konushin. Fcaf3d: fully convolutional anchor-free 3d object detection. In *ECCV*, 2022. 2

[56] Danila Rukhovich, Anna Vorontsova, and Anton Konushin. Imvoxelnet: Image to voxels projection for monocular and multi-view general-purpose 3d object detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022. 1, 2, 3, 4, 5, 6, 7, 8

[57] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *IJCV*, 2015. 5

[58] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *CVPR*, 2019. 2

[59] Shaoshuai Shi, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. From points to parts: 3d object detection from point cloud with part-aware and part-aggregation network. *IEEE TPAMI*, 2020. 2

[60] Weijing Shi and Raj Rajkumar. Point-gnn: Graph neural network for 3d object detection in a point cloud. In *CVPR*, 2020. 2

[61] Xuepeng Shi, Qi Ye, Xiaozhi Chen, Chuangrong Chen, Zhixiang Chen, and Tae-Kyun Kim. Geometry-based distance decomposition for monocular 3d object detection. In *ICCV*, 2021. 2

[62] Andrea Simonelli, Samuel Rota Bulo, Lorenzo Porzi, Manuel López-Antequera, and Peter Kontschieder. Disentangling monocular 3d object detection. In *ICCV*, 2019. 2

[63] Shuran Song, Samuel P Lichtenberg, and Jianxiong Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *CVPR*, 2015. 3

[64] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *CVPR*, 2022. 4

[65] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: A simple and strong anchor-free object detector. *IEEE TPAMI*, 2020. 4, 5

[66] Ching-Yu Tseng, Yi-Rong Chen, Hsin-Ying Lee, Tsung-Han Wu, Wen-Chin Chen, and Winston Hsu. Crossdtr: Cross-view and depth-guided transformers for 3d object detection. *arXiv preprint arXiv:2209.13507*, 2022. 3

[67] Haiyang Wang, Shaoshuai Shi, Ze Yang, Rongyao Fang, Qi Qian, Hongsheng Li, Bernt Schiele, and Liwei Wang. Rbgnet: Ray-based grouping for 3d object detection. In *CVPR*, 2022. 2

[68] Li Wang, Liang Du, Xiaoqing Ye, Yanwei Fu, Guodong Guo, Xiangyang Xue, Jianfeng Feng, and Li Zhang. Depth-conditioned dynamic message propagation for monocular 3d object detection. In *CVPR*, 2021. 3

[69] Tai Wang, ZHU Xinge, Jiangmiao Pang, and Dahua Lin. Probabilistic and geometric depth: Detecting objects in perspective. In *Conference on Robot Learning*, 2022. 2

[70] Tai Wang, Xinge Zhu, Jiangmiao Pang, and Dahua Lin. Fcos3d: Fully convolutional one-stage monocular 3d object detection. In *ICCV*, 2021. 2

[71] Xinlong Wang, Wei Yin, Tao Kong, Yuning Jiang, Lei Li, and Chunhua Shen. Task-aware monocular depth estimation for 3d object detection. In *AAAI*, 2020. 3

[72] Yan Wang, Wei-Lun Chao, Divyansh Garg, Bharath Hariharan, Mark Campbell, and Kilian Q Weinberger. Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving. In *CVPR*, 2019. 3

[73] Yue Wang, Vitor Campagnolo Guizilini, Tianyuan Zhang, Yilun Wang, Hang Zhao, and Justin Solomon. Detr3d: 3d object detection from multi-view images via 3d-to-2d queries. In *Conference on Robot Learning*, 2022. 3

[74] Bin Xu and Zhenzhong Chen. Multi-level fusion based 3d object detection from monocular images. In *CVPR*, 2018. 3

[75] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. *Sensors*, 2018. 2

[76] Bin Yang, Wenjie Luo, and Raquel Urtasun. Pixor: Real-time 3d object detection from point clouds. In *CVPR*, 2018. 2

[77] Zetong Yang, Yanan Sun, Shu Liu, and Jiaya Jia. 3dssd: Point-based 3d single stage object detector. In *CVPR*, 2020. 2, 7

[78] Zetong Yang, Yanan Sun, Shu Liu, Xiaoyong Shen, and Jiaya Jia. Std: Sparse-to-dense 3d object detector for point cloud. In *ICCV*, 2019. 2

[79] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for unstructured multi-view stereo. In *ECCV*, 2018. 4

[80] Zehao Yu, Songyou Peng, Michael Niemeyer, Torsten Sattler, and Andreas Geiger. Monosdf: Exploring monocular geometric cues for neural implicit surface reconstruction. *arXiv preprint arXiv:2206.00665*, 2022. 4

[81] Cheng Zhang, Zhaopeng Cui, Yinda Zhang, Bing Zeng, Marc Pollefeys, and Shuaicheng Liu. Holistic 3d scene understanding from a single image with implicit representation. In *CVPR*, 2021. 2, 3

[82] Shifeng Zhang, Cheng Chi, Yongqiang Yao, Zhen Lei, and Stan Z Li. Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. In *CVPR*, 2020. 5

[83] Yunpeng Zhang, Jiwen Lu, and Jie Zhou. Objects are different: Flexible monocular 3d object detection. In *CVPR*, 2021. 2

[84] Yinda Zhang, Shuran Song, Ping Tan, and Jianxiong Xiao. Panocontext: A whole-room 3d context model for panoramic scene understanding. In *ECCV*, 2014. 3

[85] Zaiwei Zhang, Bo Sun, Haitao Yang, and Qixing Huang. H3dnet: 3d object detection using hybrid geometric primitives. In *ECCV*, 2020. 2

[86] Yibiao Zhao and Song-Chun Zhu. Image parsing with stochastic scene grammar. In *NeurIPS*, 2011. 3

[87] Yibiao Zhao and Song-Chun Zhu. Scene parsing by integrating function, geometry and appearance models. In *CVPR*, 2013. 3

[88] Dingfu Zhou, Jin Fang, Xibin Song, Chenye Guan, Junbo Yin, Yuchao Dai, and Ruigang Yang. Iou loss for 2d/3d object detection. In *2019 International Conference on 3D Vision (3DV)*, 2019. 5

[89] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019. 2

[90] Yunsong Zhou, Yuan He, Hongzi Zhu, Cheng Wang, Hongyang Li, and Qinhong Jiang. Monocular 3d object detection: An extrinsic parameter free approach. In *CVPR*, 2021. 2

[91] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *CVPR*, 2018. 2