Stereo Neural Vernier Caliper

Shichao Li[*1], **Zechun Liu**[1,2], **Zhiqiang Shen**[2,3,1], **Kwang-Ting Cheng**[1]

[1]Hong Kong University of Science and Technology
[2]Carnegie Mellon University
[3]Mohamed bin Zayed University of Artificial Intelligence
slicd@cse.ust.hk, timcheng@ust.hk

Abstract

We propose a new *object-centric* framework for learning-based stereo 3D object detection. Previous studies build *scene-centric* representations that do not consider the significant variation among outdoor instances and thus lack the flexibility and functionalities that an instance-level model can offer. We build such an instance-level model by formulating and tackling a *local update problem*, i.e., how to predict a refined update given an initial 3D cuboid guess. We demonstrate how solving this problem can complement *scene-centric* approaches in (i) building a *coarse-to-fine* multi-resolution system, (ii) performing model-agnostic object location refinement and, (iii) conducting stereo 3D tracking-by-detection. Extensive experiments demonstrate the effectiveness of our approach, which achieves state-of-the-art performance on the KITTI benchmark. Code and pre-trained models are available at https://github.com/Nicholasli1995/SNVC.

Introduction

Accurate perception of surrounding objects' 3D attributes is indispensable for autonomous driving, robot navigation, and traffic surveillance. Active range sensors such as Li-DAR measures the 3D scene geometry directly to perform precise 3D localization (Lang et al. 2019; Shi, Wang, and Li 2019). However, LiDAR sensors incur a high cost and can be limited in perception range where distant objects are only captured with very few points. On the other hand, passive sensors like cameras are inexpensive, yet the depth information is lost during the image formation process which makes 3D scene understanding a challenging inverse problem. Estimating depth from a *single* RGB image is ill-posed and leads to limited 3D object detection performance (Brazil and Liu 2019; Wang et al. 2021a; Lu et al. 2021). Stereo cameras, simulating a *binocular* human vision system, are the minimum sensor configuration that can exploit multi-view geometry for more reliable depth inference. Studying stereo 3D object detection (S3DOD), thus not only is a pursuit of the vision community that aims at visual scene understanding, but also offers practical value to complement active sensors through multi-sensor fusion (Liang et al. 2019).
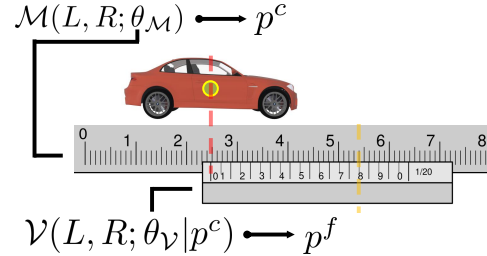
Figure 1: Stereo Neural Vernier Caliper (SNVC) consists of the main scale network $\mathcal{M}$ that models a coarse global scene and the Vernier scale network $\mathcal{V}$ that models a fine local scene. $\mathcal{M}$ predicts coarse location p^c while $\mathcal{V}$ *takes a closer look* around p^c and makes finer measurements.

Recent state-of-the-art (SOTA) S3DOD approaches take a *scene-centric* view and build a data representation for the **whole** scene. We use a representative work (Chen et al. 2020) as our baseline, which uses the estimated depth to build a voxel-based scene representation for object detection. In contrast to it, our study promotes an *object-centric* viewpoint and explores instance-level analysis for S3DOD. The following practical considerations motivate this study, which demand the attributes of this *object-centric* viewpoint that are not offered by the *scene-centric* counterpart.

- Cost-accuracy trade-off considering the depth variation: a distant object usually has lower-resolution (LR) feature representation compared to a nearby one, which makes it difficult to precisely recover its 3D attributes. Naively re-computing high-resolution (HR) features and using a finer voxel grid for the **whole** scene lead to intimidating computational cost and cubic memory growth. This is also unnecessary if the model performance for nearby instances already saturates. If an instance-level model is available, a multi-resolution (MR) system can be built to mitigate this problem. This system benefits from a *coarse-to-fine* design which computes an LR global representation and focuses on the tiny instances with complementary HR features[1].

- Capability for efficiently handling new frames: videos

---

[1]Such HR features can be obtained computationally or physically with an actively zooming camera (Bellotto et al. 2009).

are more prevalent in real-world scenarios than two static stereo images. The *scene-centric* approaches need to build a scene representation for every new pair of frames. With an instance-level model, one only needs to do so for some key frames and can conduct *tracking-by-detection* (Andriluka, Roth, and Schiele 2008), i.e., processing only a portion of new frames given region-of-interests (RoIs) implied by past detections.

- Flexibility in video applications: certain objects are more important in a 3D scene, e.g., a car heading towards a driver should draw more attention than a vehicle leaving the field of view. Instance-level analysis can offer the flexibility to prioritize certain regions when analyzing new frames.

To make an instance-level analysis model useful in all aforementioned scenarios, we can design it as a refinement model in a recurrent manner. Given an initial 3D bounding box guess, the model should reason about the 3D space around the cuboid guess and give an updated prediction. Note the initial guess can vary for different user scenarios, e.g., it can be a proposal from an LR global model or the prediction of the last frame. Thus our research question is *how to design such an instance-level refinement model in the stereo perception setting*? We tackle this problem by designing an instance-level neural network $\mathcal{V}$ which builds a voxel-based local scene representation and scores each voxel how likely it is a potential update of an object part. $\mathcal{V}$ can be combined with a global model $\mathcal{M}$ to form an MR S3DOD framework as illustrated in Fig. 1. $\mathcal{M}$ performs *scene-level* depth estimation and outputs coarse 3D object proposals. Conditioned on each 3D proposal, $\mathcal{V}$ further extracts HR features and refines its 3D attributes. We name this framework Stereo Neural Vernier Caliper (SNVC) since it resembles a Vernier caliper where $\mathcal{M}$ (the main scale) models the 3D scene with a coarse voxel grid while $\mathcal{V}$ (the Vernier scale) models an HR local scene conditioned on the initial guess. Compared to prior arts, our approach endows an S3DOD system with the advantages discussed above and can model fine-grained structures for important regions with a tractable number of voxels. Such ability leads to superior detection performance, especially for the hard instances, i.e., the tiny and occluded ones. This paper's contributions are summarized as:

- We propose the first MR framework for voxel-based S3DOD. The new instance-level model $\mathcal{V}$ within it, to our best knowledge, is also the first HR voxel representation learning approach tackling the *local update problem*.

- We study the transferability of $\mathcal{V}$ and demonstrate it can be used as a model-agnostic and plug-and-play refinement module for S3DOD that complements many existing *scene-centric* approaches.

- SNVC out-performs previously published results on the KITTI benchmark for S3DOD at the date of submission (Sep 8th, 2021).

Related Work

Our study is relevant to the following research directions while having distinct contributions.

Learning-based 3D object detection aims to learn a mapping from sensor input to 3D bounding box representations (Chen et al. 2015). Depending on the sensor modality, two parallel lines of research are vision-based methods (Roddick, Kendall, and Cipolla 2019; Ke et al. 2020; Weng and Kitani 2019; Wang et al. 2021b; Reading et al. 2021) and LiDAR-based approaches (Chen et al. 2017; Yan, Mao, and Li 2018; Zhou and Tuzel 2018; Qi et al. 2018; Li, Wang, and Wang 2021). Our approach lies in the former which does not require expensive range sensors. Compared with previous stereo vision-based studies (Chen et al. 2020; Garg et al. 2020) that focus on global scene modeling, we deal with a different *local update problem* and dedicate a new model for HR instance-level analysis. Our design can complement previous 3D object detectors that do not share the flexibility and high precision offered by our approach.

Instance-level analysis in 3D object detection builds a feature representation for an instance proposal to estimate its high-quality 3D attributes. FQ-Net (Liu et al. 2019) draws a projected cuboid proposal on an instance patch and regresses its 3D Intersection-over-Union (IoU) for location refinement. RAR-Net (Liu et al. 2020) formulates a reinforcement learning framework for iterative instance pose refinement. 3D-RCNN (Kundu, Li, and Rehg 2018) uses instance shape as auxiliary supervision yet requires extra annotations which are not needed by our approach. Notably, all these methods only consider the monocular case and *cannot* utilize stereo imagery. ZoomNet (Xu et al. 2020) and Disp R-CNN (Sun et al. 2020) construct point-based representations for each instance proposal and require extra mask annotation during training. Such representations also lose the semantic features and are less robust for distant and occluded objects which have few foreground points. We instead propose to learn a voxel-based representation to encode both semantic and geometric features.

Voxel-based representation is a classical and simple data structure encoding 3D features and is widely adopted in image-based rendering (Seitz and Dyer 1999) and multi-view reconstruction (Vogiatzis, Torr, and Cipolla 2005). Early studies utilize hand-crafted features and energy-based models to encode prior knowledge (Snow, Viola, and Zabih 2000), while recent deep learning-based approaches (Choy et al. 2016; Riegler, Osman Ulusoy, and Geiger 2017) directly learn such representations from data. Our approach learns a voxel-based neural representation for the unique *local update problem* under the stereo perception setting.

High-resolution neural networks was recently proposed (Sun et al. 2019) to model fine-grained spatial structure details to benefit tasks that involve precise localization. However, later studies only focus on monocular and 2D tasks (Wang et al. 2020; Cheng et al. 2020; Li et al. 2020, 2021). This work instead studies HR representation learning for S3DOD and can build an unprecedented fine 3D spatial resolution of 3 centimeters in the real self-driving scenarios.

Multi-resolution volumetric representation was previously studied for 3D shape representation (Riegler, Osman Ulusoy, and Geiger 2017) and reconstruction (Blaha et al. 2016) where less important region (e.g., free space) is represented with coarse voxel grid and a finer voxel grid

is used for regions close to object surface adaptively. The octree (Laine and Karras 2010) was a popular choice to generate an MR partition of a 3D region. We argue that there is also a large variation of importance for different regions in S3DOD where regions near objects have a larger influence on the detection performance. We thus introduce the idea of varying resolution to voxel-based S3DOD for the first time with a new MR system.

Cascaded 3D Object Detection

SNVC can be formulated as a cascaded model represented as the set $\{\mathcal{M}, \mathcal{V}\}$ consisting of the main scale network $\mathcal{M}$ and the Vernier scale network $\mathcal{V}$. Given an RGB image pair $(\mathcal{L}, \mathcal{R})$ captured by calibrated stereo cameras, $\mathcal{M}$ builds a coarse representation of the *global* 3D scene and predicts q 3D bounding box proposals $\{p_q^c\}_{q=1}^N$ as

$$\mathcal{M}(\mathcal{L}, \mathcal{R}; \theta_\mathcal{M}) = \{p_q^c\}_{q=1}^N. \qquad (1)$$

Conditioned on each coarse proposal p_q^c, $\mathcal{V}$ deals with the *local update problem* by constructing an HR local scene representation and infers an offset δp_q to obtain the refined pose $p_q^f = p_q^c + \delta p_q$.

$$\mathcal{V}(\mathcal{L}, \mathcal{R}; \theta_\mathcal{V} | p_q^c) = \delta p_q. \qquad (2)$$

This framework is general and does not enforce any assumption on the architecture design of $\mathcal{M}$. As we will show, one design of $\mathcal{V}$ can handle coarse predictions from different implementations of $\mathcal{M}$ and thus be *model-agnostic*. In light of this, we only detail our architecture of $\mathcal{V}$ in the main text. $\mathcal{M}$ used in our final system is sketched in Fig. 3 whose architecture details are in the supplementary material (SM).

High-resolution Instance-level Update

In this study, we propose to design $\mathcal{V}$ based on a three-step procedure:

- Space partitioning: we build a dense voxel grid in a 3D RoI conditioned on the coarse 3D proposal where each voxel is a candidate for a part location update.
- Deep voxel coloring: we aggregate high-level features extracted by a deep neural network for each voxel.
- Voting: we extract spatial structural information from the colored voxel grid and score each voxel how likely it is a location update of an object part. The pose update is then estimated via a vote based on the predicted object part locations and confidences.

The first step builds a very dense local 3D scene representation aiming at high-precision 3D object localization. In step two we study two strategies that correspond to two user scenarios and lead to two architecture variants. Our strategy in step three utilizes several pre-defined cuboid parts for robust inference considering occlusion. The following subsections describe each step in detail.

Candidate generation. Given a coarse 3D bounding box prediction p_q^c represented as a 7-tuple $p_q^c = (x_q, y_q, z_q, h_q, w_q, l_q, \theta_q)$, where (x_q, y_q, z_q), (h_q, w_q, l_q) and θ_q denotes its translation, size (height, width, length)
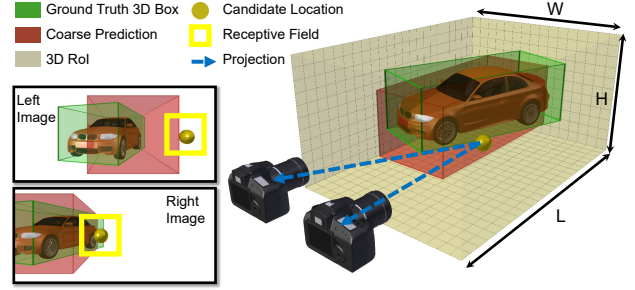

Figure 2: Illustration of the *local update problem*. A 3D region-of-interest (brown) is defined based on the coarse proposal (red). Each candidate (yellow ball) aggregates high-level stereo visual features. Such high-resolution volumetric features are used to infer the ground truth 3D bounding box (green). Best viewed in color. A coarse grid is drawn for visualization while the real grid is much finer.

and orientation respectively. These quantities are represented in the camera coordinate system (CCS). We define a 3D RoI r_q around p_q^c, within which $\mathcal{V}$ needs to predict the instance pose update. We use a cuboid RoI for convenience and represent r_q as $(x_q, y_q, z_q, H, W, L, \theta_q)$. r_q has the same 4-D pose $(x_q, y_q, z_q, \theta_q)$ as p_q^c and a pre-defined range (H, W, L). The *Vernier scale* is a fine partition of r_q represented as a 3D voxel grid $\{d_{i,j,k}\}_{i=1,j=1,k=1}^{N_H, N_W, N_L}$ with N_H, N_W, and N_L voxels uniformly sampled along the height, width and length directions respectively. Each voxel $d_{i,j,k}$ is a candidate for precise location update. Fig. 2 illustrates a r_q and its corresponding candidates.

We represent r_q with the camera coordinates of its center and the eight corners by applying a homography encoding translation and rotation as

$$\mathbf{H}_{4\times4} \begin{bmatrix} | \\ \mathbf{O}_{3\times9} \\ | \\ \mathbf{1} \end{bmatrix} = \begin{bmatrix} cos\theta_q & 0 & sin\theta_q & x_q \\ 0 & 1 & 0 & y_q \\ -sin\theta_q & 0 & cos\theta_q & z_q \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} | \\ \mathbf{O}_{3\times9} \\ | \\ \mathbf{1} \end{bmatrix}, \qquad (3)$$

where $\mathbf{O}_{3\times9}$ represent the 9 parts in the object coordinate system as

$$\begin{bmatrix} 0 & \frac{L}{2} & \frac{L}{2} & \frac{L}{2} & \frac{L}{2} & -\frac{L}{2} & -\frac{L}{2} & -\frac{L}{2} & -\frac{L}{2} \\ 0 & -\frac{H}{2} & \frac{H}{2} & -\frac{H}{2} & \frac{H}{2} & -\frac{H}{2} & \frac{H}{2} & -\frac{H}{2} & \frac{H}{2} \\ 0 & \frac{W}{2} & \frac{W}{2} & -\frac{W}{2} & -\frac{W}{2} & \frac{W}{2} & \frac{W}{2} & -\frac{W}{2} & -\frac{W}{2} \end{bmatrix}. \qquad (4)$$

Similarly, $d_{i,j,k}$ can be represented in the CCS as

$$d_{i,j,k} = \mathbf{H}_{4\times4} \begin{bmatrix} -N_L\Delta L/2 + (k-1)\Delta L \\ -N_H\Delta H/2 + (i-1)\Delta H \\ N_W\Delta W/2 - (j-1)\Delta W \\ 1 \end{bmatrix}, \qquad (5)$$

where $[\frac{-N_L\Delta L}{2}, \frac{-N_H\Delta H}{2}, \frac{N_W\Delta W}{2}]$ is the left-back-top corner of the 3D grid, and $[\Delta L, \Delta H, \Delta W] = [\frac{L}{N_L}, \frac{H}{N_H}, \frac{W}{N_W}]$ gives the grid resolution. In experiments, we specify $\Delta L, \Delta H, \Delta W$ as 3, 10, and 3 centimeters respectively and

$[N_L, N_H, N_W] = [192, 32, 128]$. Such 3-cm resolution is in contrast to a 20-cm or even coarser resolution used in our $\mathcal{M}$ and previous studies (Chen et al. 2020; Li, Su, and Zhao 2021; Garg et al. 2020).

Feature aggregation. We design two types of feature aggregation strategies depending on whether one can reuse features computed by $\mathcal{M}$. These two strategies result in two model variants ($\mathcal{V}$-A and $\mathcal{V}$-S).

For the $\mathcal{V}$-A (*model-agnostic*) models, we assume no knowledge of the architecture of $\mathcal{M}$ and do not reuse its features. This type of model is useful for dealing with proposals from different 3DOD models or handling new frames that have no features computed yet. In this case, we aggregate feature for each candidate from the left/right visual features $\mathcal{L}^F/\mathcal{R}^F$, which are obtained from a fully convolution network $\mathcal{L}^F/\mathcal{R}^F = \mathcal{N}^{2D}(\mathcal{L}/\mathcal{R}; \theta_{\mathcal{N}})$. Specifically, the feature vector for $d_{i,j,k}$ is

$$F_{i,j,k} = \mathcal{W}(\mathcal{L}^F, K_l d_{i,j,k}) \oplus \mathcal{W}(\mathcal{R}^F, K_r d_{i,j,k}), \quad (6)$$

where K_l, K_r are the intrinsic parameters of the left-/right cameras and $\mathcal{W}^F$ is a warping function and $\mathcal{W}(\mathcal{L}^F, K_l d_{i,j,k})$ extracts the feature at $K_l d_{i,j,k}$ from the corresponding location on the feature maps $\mathcal{L}^F$. $\mathcal{W}$ is implemented as bi-linear interpolation.

For the $\mathcal{V}$-S (*model-specific*) models, we assume that $\mathcal{M}$ also constructs a volumetric scene representation (e.g., a global cost-volume $\mathcal{C}_g$). This case happens when one builds a two-stage voxel-based detector where $\mathcal{V}$ has access to precomputed features. Here we sample $F_{i,j,k}$ from such features as $F_{i,j,k} = \mathcal{W}(\mathcal{C}_g, d_{i,j,k})$ to save computation.

After aggregation, the grid is colored with high-level visual features. This is in contrast to the low-level pixel intensity used in the classical voxel coloring study (Seitz and Dyer 1999). In addition, all voxels are colored instead of coloring only the consistent voxels (Seitz and Dyer 1999) because the Lambertian scene assumption does not hold. Subsequently a network $\mathcal{N}^{3D}$ processes the aggregated volumetric representation and predicts the output detailed below.

Output representation. For each p_q^c, we encode the ground truth update as dense multi-part confidence since we implement $\mathcal{N}^{3D}$ as a CNN which is known good at dense classification tasks. For part m ($m = 1, \ldots, K$), we assign each candidate a ground truth *confidence* $c_{i,j,k}^m$. In implementation we adopt the ground plane assumption (Roddick, Kendall, and Cipolla 2019) for autonomous driving datasets and ignore the offset in the height direction, which leads to a Bird's Eye View (BEV) confidence map as $c_{j,k}^m$. This does not reduce the generality of our framework and one can keep the original dimension if the height offset is significant in a different dataset. Denote the ground truth location for part m as $(j^\star, k^\star)$. The confidence map is defined as

$$c_{j,k}^m = e^{-\frac{(j-j^\star)^2 + (k-k^\star)^2}{\sigma^2}}. \quad (7)$$

In total, K=9 parts including the ground truth cuboid center and its 8 corners are used as similarly defined in Eq. (4). To reduce quantization errors, we transform the predicted confidences $\hat{c}_{j,k}$ into x-z coordinates $\hat{Y}_{K \times 2}$ using several convolution layers as in (Li et al. 2021).

Model instantiation. We have specified the inputs and outputs of $\mathcal{V}$, where detailed parameters and network components $\mathcal{N}^{2D}$ and $\mathcal{N}^{3D}$ are abstracted away. This makes the framework flexible and one can specify these parameters and sub-networks based on its computation budget. In our study, $\mathcal{N}^{2D}$ is implemented as HRNet-W32 (Sun et al. 2019). After coloring the voxel grid, we use 3D convolution layers and a 3D hourglass network to extract 3D spatial features. The 3D features are pooled and reshaped into BEV feature maps that are further transformed into confidence maps. The network architecture of $\mathcal{V}$ is depicted in Fig. 4 and detailed in the SM.

Error-statistics-agnostic Training

We train $\mathcal{M}$ and $\mathcal{V}$ separately. The training process of $\mathcal{M}$ can vary for different possible implementations. For the $\mathcal{M}$ used in our study, we use a similar procedure as the anchor-based baseline (Chen et al. 2020). The training supervision consists of a depth regression loss, an anchor classification loss, and an offset regression loss. The details are in the SM.

Training $\mathcal{V}$ is the key step in our framework, where the inputs and targets for training $\mathcal{V}$ are not readily available. We propose a simple *error-statistics-agnostic* (ESA) strategy to synthesize training data. This training strategy does not depend on $\mathcal{M}$ and can be used for any proposal model conveniently. Notably, PoseFix (Moon, Chang, and Lee 2019) proposed a refinement model for a different task and assumes there are certain types of errors from the predictions of the coarse models. Such error statistics are themselves collected from extra validation results. In contrast, our strategy assumes a Gaussian prior and does not require more specific knowledge of the error behavior.

Generating training data. For each ground truth 3D bounding box $\mathcal{B}_i = (x_i, y_i, z_i, h_i, w_i, l_i, \theta_i)$, we simulate a coarse prediction by adding a noise vector $n_i = (n_i^x, n_i^y, n_i^z, n_i^h, n_i^w, n_i^l, n_i^\theta)$ where $n \sim N(\mathbf{0}, \Sigma)$. We further assume that the noise for each attribute is independent and the covariance matrix Σ is diagonal. In experiments the standard derivations for the above attributes are 0.3m, 0m, 0.3m, 5cm, 5cm, 5cm and 5° respectively. Larger noise can be used if one assumes a weaker $\mathcal{M}$. The simulated coarse prediction $p_i^c = \mathcal{B}_i + n_i$, along with the ground truth confidence maps of $\mathcal{B}_i$, forms one training pair of $\mathcal{V}$. The Gaussian perturbation is added on-line for each instance in every iteration, which behaves like data augmentation so that $\mathcal{V}$ does not over-fit to a special subset of inputs.

Loss function. We penalize the predicted confidence maps of $\mathcal{V}$ with L_2 loss as $L_{conf} = L_2(\hat{c}_{j,k}, c_{j,k})$. The transformed coordinates are penalized with smooth L_1 loss $L_{coord} = SL_1(\hat{Y}_{K \times 2}, Y_{K \times 2})$. For training $\mathcal{V}$-S models, we have $L_{total}^S = L_{conf} + L_{coord}$.

For $\mathcal{V}$-A models, apart from supervising these regression targets, we add an extra intermediate supervision since we cannot reuse features encoding the scene depth. We add a 3D convolution head that classifies for each candidate $d_{i,j,k}$ whether it is a foreground or not. This serves to add depth cues to train $\mathcal{V}$ similar to (Chen et al. 2020) yet different in that we directly add supervision in the 3D space instead
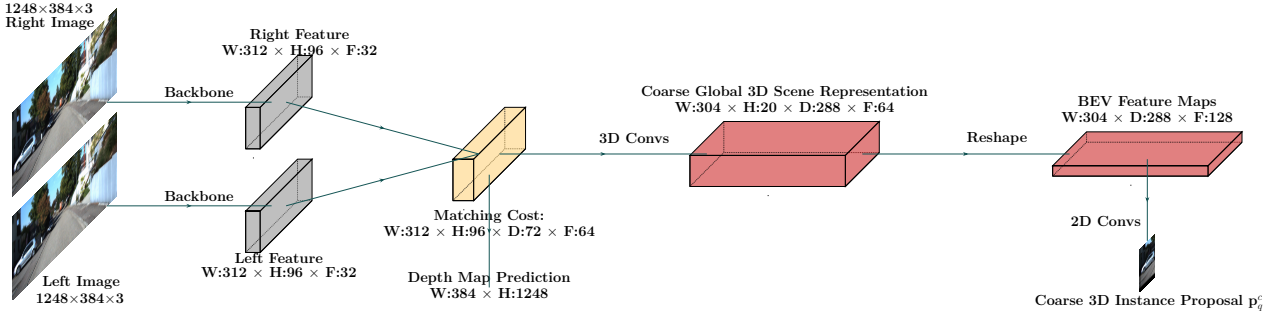
Figure 3: Diagram of the used main scale network $\mathcal{M}$ following a similar design of DSGN (Chen et al. 2020). A global voxel grid is used to sample image and cost volume features and construct a volumetric representation of the 3D scene in a predefined spatial range. This volumetric representation is converted into Bird's Eye View (BEV) feature maps. Object proposals are obtained by anchor classification and offset regression based on the BEV feature maps. Details are included in our SM.
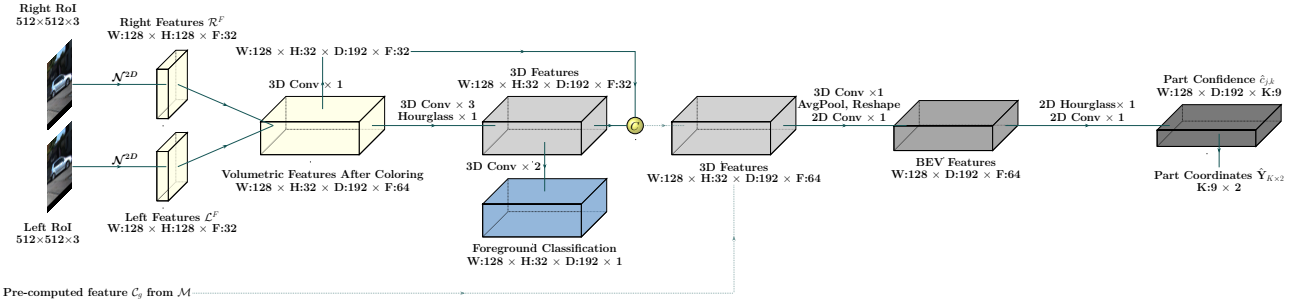

Figure 4: Architecture of the Vernier scale network $\mathcal{V}$. High-resolution features are extracted from left/right regions of interest (RoIs). A uniform 3D grid conditioned on the current prediction is sampled. Each grid point is projected back to the RoIs to aggregate image features. Pre-computed features from $\mathcal{M}$ are used if available. The grid features are processed by a 3D CNN to predict the part confidence maps which implies the pose update.

of the cost volume used in (Chen et al. 2020). For each point in the captured point cloud in r_q, the candidate it occupies after coordinate quantization is treated as a foreground. A candidate outside of the ground truth box $\mathcal{B}_q$ is treated as background. All other candidates are not assigned labels since they can be foreground (a foreground not recorded by the LiDAR) as well as background (free space). Since there are much more background candidates than the foreground ones, we use focal loss (Lin et al. 2017) to supervise this classification task as

$$L_{fg} = \begin{cases} -\alpha(1-\hat{p}_{i,j,k})^\gamma \log(\hat{p}_{i,j,k}), & p_{i,j,k} = 1, \\ -(1-\alpha)\hat{p}_{i,j,k}^\gamma \log(1-\hat{p}_{i,j,k}), & p_{i,j,k} = 0 \\ 0, & \text{else.} \end{cases}$$

where $\hat{p}_{i,j,k}$ is the predicted foreground probability of candidate $d_{i,j,k}$ and $p_{i,j,k}$ is the ground-truth one where $p_{i,j,k} = 1$ for foreground. We use the default parameters $\gamma = 2$ and $\alpha = 0.25$ and the total training loss $L_{total}^A = L_{conf} + L_{coord} + L_{fg}$.

Confidence-aware Robust Inference

During inference, the update prediction p_q^f derives from the predicted confidence maps $\hat{c}_{j,k}$ and x-z coordinates $\hat{Y}_{K \times 2}$,

which indicate the tentative update position for each part. Certain study (Peng et al. 2020) on instance-level S3DOD models only predicts the center of the object. This is similar to $K = 1$ in our framework, where the predicted new center is used to refine the instance translation. However, this could lead to sub-optimal results when the center is occluded or hard to estimate. We thus propose an update strategy by utilizing K=9 parts along with predicted confidence. The predicted coordinates $\hat{Y}_{K \times 2}$ may not define parallel and orthogonal edges. We thus employ a 9-point registration approach by estimating a rigid transformation $\{\hat{R}, \hat{T}\}$ as

$$\hat{R}, \hat{T} = \arg\min_{R,T} W||R\tilde{Y}_{K \times 2} + T - \hat{Y}_{K \times 2}||, \quad (8)$$

where $\tilde{Y}_{K \times 2}$ is the same part coordinates of the current proposal and W is the diagonal matrix with part confidences as its non-zero elements. The closed-form solution to Eq. (8) is

$$\hat{R} = V^T U^T \quad (9)$$

$$\hat{T} = -\hat{R}\tilde{Y}_{centroid} + \hat{Y}_{centroid} \quad (10)$$

where U, V gives the singular decomposition as USV $= \tilde{Y}^T W \hat{Y}$ where $\tilde{Y}_{centroid}$ is the average location of the K parts. This solution is the global optimum of Eq. (8) as detailed in the SM. The refined 3D box is then obtained by applying this estimated transformation to the current proposal.
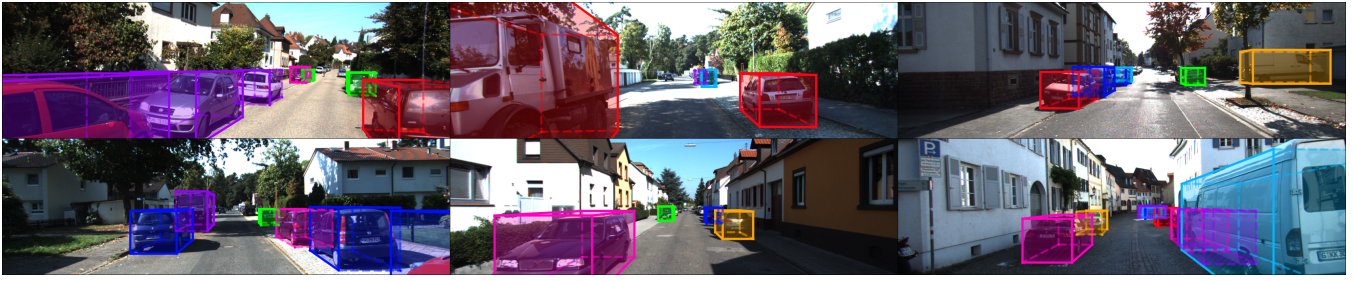
Figure 5: Qualitative results on KITTI *val* split. Best viewed in color. More results can be found in the supplementary material.

Method	Reference	$AP_{3D}@R_{11}$			$AP_{BEV}@R_{11}$		
		Easy	Moderate	Hard	Easy	Moderate	Hard
MLF (Xu and Chen 2018)	CVPR' 18	–	9.80	–	–	19.54	–
TLNet (Qin, Wang, and Lu 2019)	CVPR' 19	18.15	14.26	13.72	29.22	21.88	18.83
Stereo R-CNN (Li, Chen, and Shen 2019)	CVPR' 19	54.1	36.7	31.1	68.5	48.3	41.5
PL: F-PointNet (Wang et al. 2019)	CVPR' 19	59.4	39.8	33.5	72.8	51.8	33.5
PL++: AVOD (You et al. 2019)	ICLR' 19	63.2	46.8	39.8	77.0	63.7	56.0
IDA-3D (Peng et al. 2020)	CVPR' 20	54.97	37.45	32.23	70.68	50.21	42.93
Disp-RCNN (Sun et al. 2020)	CVPR' 20	64.29	47.73	40.11	77.63	64.38	50.68
DSGN (Chen et al. 2020)	CVPR' 20	72.31	54.27	47.71	83.24	63.91	57.83
ZoomNet (Xu et al. 2020)	AAAI' 20	62.96	50.47	43.63	78.68	66.19	57.60
RTS-3D (Li, Su, and Zhao 2021)	AAAI' 21	64.76	46.70	39.27	77.50	58.65	50.14
Disp-RCNN-flb* (Chen et al. 2021)	T-PAMI' 21	70.11	54.43	47.40	77.47	66.06	57.76
SNVC (Ours)	AAAI' 22	**77.29**	**63.75**	**56.81**	**87.07**	**72.95**	**66.77**

Table 1: Quantitative comparison of AP_{3D} and AP_{BEV} with SOTA stereo 3D object detection approaches on KITTI *val* split. Disp-RCNN-flb* requires extra mask annotation for training. 11 recall values are used to make the comparison consistent.

Experiments

We first introduce the used benchmark and evaluation metrics and then compare the overall performance of our MR SNVC with previously published approaches. We then demonstrate how our approach can improve existing 3D object detectors as a plug-and-play module. Finally, we present an ablation study on key design factors of $\mathcal{V}$.

Dataset. We employ the KITTI object detection benchmark (Geiger, Lenz, and Urtasun 2012) for evaluation, which contains outdoor RGB images captured with calibrated stereo cameras. The dataset is split into 7,481 training images and 7,518 testing images. The training images are further split into the *train* split and the *val* split containing 3,712 and 3,769 images respectively. We use the *train* split for training and conduct hyper-parameter tuning on the *val* split. When reporting the model performance on the testing set, both the *train* split and the *val* split are used for training.

Evaluation metrics. We conduct the evaluation for the car category. We employ the official average precision metrics to validate our approach. *3D Average Precision* (AP_{3D}) measures precision at 41 uniformly sampled recall values where a true positive is a predicted 3D box that has 3D intersection-over-union (IoU) > 0.7 with the ground truth. *Bird Eye's View Average Precision* (AP_{BEV}) instead uses 2D IoU > 0.7 as the criterion where the 3D boxes are projected to the ground plane and the object heights are ignored. The KITTI benchmark further defines three sets of ground truth labels with different difficulty levels as easy, moderate, and hard. The difficulty level of one ground truth label is determined

according to its 2D bounding box height, its occlusion level, and its truncation level. Evaluation is performed in parallel in these three different sets of ground truth labels. The hard set contains all the ground truth labels while the easy and moderate sets contain a fraction of easier objects. For consistency with previous works, results on the *val* split are reported using 11 recall values (denoted as $@R_{11}$).

Training details are attached in our SM.

Comparison with state-of-the-arts. Tab. 1 and Tab. 2 compares the 3D object detection performance of our approach with other previously published methods on the *val* split and the official testing set respectively. Fig. 5 shows 3D object detections of our system on the *val* split. Our system using $\mathcal{V}$-S outperforms previous approaches in all metrics with a clear margin. Compared with (Chen et al. 2020) that only builds coarse scene-level representation, our system benefits from the proposed HR instance-level model that leads to more precise 3D localization. Compared with the approaches that build instance-level point cloud (Xu et al. 2020; Sun et al. 2020), our voxel-based representation shows superior performance, especially for the hard category. We believe the reason is that the instance point cloud in (Xu et al. 2020; Sun et al. 2020) is sensitive to the depth estimation error and fails to handle distant and occluded objects due to a small number of points and lack of semantic visual features. In this camera-ready version paper, we also refer to a concurrent work (Guo et al. 2021) starting from the same baseline (DSGN) as us which regularizes voxel representation learning by distilling knowledge from an extra pre-trained
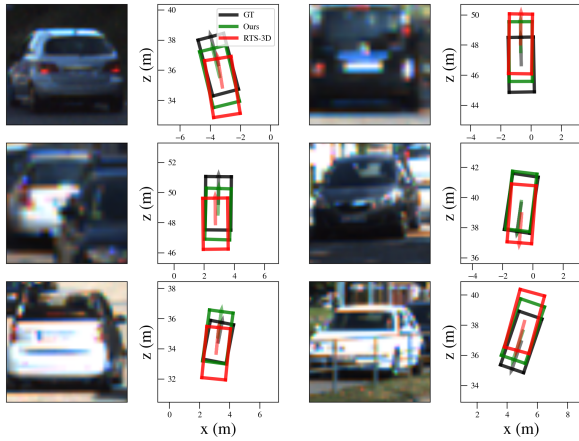
Figure 6: Qualitative comparison with RTS-3D (Li, Su, and Zhao 2021) for objects on KITTI *val* split. The image patches are RoIs on the left image and the 3D cuboid predictions are shown as bounding boxes in the bird's eye view plots. Note the objects are distant from the cameras.
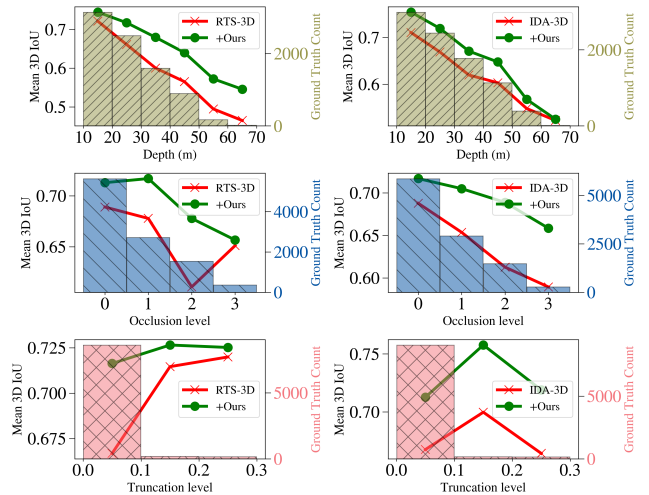

Figure 7: 3D object detection performance with respect to ground truth depth (top), occlusion level (middle) and truncation level (bottom) on the KITTI *val* split. Each bar indicates the number of ground truth (GT) objects that fall in the corresponding bin and are detected with 3D IoU > 0.3. A red cross shows the average 3D IoU for those detected GT objects in a bin before utilizing our $\mathcal{V}$ for refinement. A green dot shows such a result after using $\mathcal{V}$. The occlusion level and truncation level are annotated for each ground truth object in the KITTI benchmark.

network. The contribution of distillation is complementary to ours. None of any methods in Tab. 1/2 needs this process known as learning from privileged information (Lopez-Paz et al. 2015) or an extra teacher.

Model-agnostic refinement. Tab. 3 shows the result when utilizing our $\mathcal{V}$ along with other existing 3D object detectors. We download the pre-trained weights from the official implementation[2] of IDA-3D and RTS-3D to generate proposals and use our $\mathcal{V}$ to refine them. Note that using $\mathcal{V}$ leads to consistent and significant performance gain. This result validates that our $\mathcal{V}$ can be used as a model-agnostic refinement model. We show predictions for some distant and partially-occluded objects in Fig. 6 where our approach obtains better pose estimation performance compared to RTS-3D. RTS-3D can be used to generate coarse predictions in real-time applications while our $\mathcal{V}$ can complement it to obtain high-quality predictions only when necessary.

Which objects benefit more from $\mathcal{V}$? Tab. 3 shows the overall improvement in 3DOD performance, yet it does not reflect how the improvement is related to different object attributes. Fig. 7 instead shows which objects enjoy more performance improvements using the same proposals as in Tab. 3. One can use such knowledge to decide which coarse proposal to refine in practice. Each ground truth (GT) object is assigned to the corresponding bin based on an attribute such as its depth, if there is one matching predicted object. A predicted object matches a ground truth if their 3D IoU > 0.3. The match that has the largest 3D IoU is recorded for each GT object. The average of the matching 3D IoUs is shown as the line plots for GT objects in each bin.

The detection quality in terms of AP_{3D} improves for GT objects in all bins after using our $\mathcal{V}$ for refinement, in terms of all three attributes that influence the difficulty of detec-

tion. Note even if the objects are heavily occluded or partially truncated, using $\mathcal{V}$ still leads to a robust performance boost. RTS-3D also uses a voxel-based representation, and we can observe that the improvement over it for the middle-range and distant objects is more significant than the nearby objects. This validates our assumption that extracting complementary HR features are more helpful for the tiny objects that only have LR representations in the global feature maps.

Effect of learning multiple object parts. To validate our multi-part registration approach, we re-train another model to predict only the center confidence. This model uses the predicted center to update the proposal translation during inference. The performance comparison is shown in Tab. 4, where our multi-part strategy leads to consistently better performance since it is more robust to partially-occluded objects whose visible parts can be used to provide a more reliable estimate.

Effect of voxel size. To demonstrate the advantage of learning an HR voxel representation, we train $\mathcal{V}$ with a varying voxel resolution and the results of using these variants with RTS-3D are shown in Tab. 5. We can observe that using a smaller voxel size significantly improves the 3DOD performance which justifies our choice of parameters. While using a much coarser voxel grid (second row) leads to worse results, such performance is still better than using RTS-3D along without our $\mathcal{V}$. However, when the voxel grid is extremely coarse (first row), $\mathcal{V}$ cannot learn effective volumetric representation for S3DOD.

Comparison of space requirement. Building a uniform

---

[2]Different environments led to slight difference compared to the published results in Tab. 1.

Method	Reference	$AP_{3D}@R_{40}$			$AP_{BEV}@R_{40}$		
		Easy	Moderate	Hard	Easy	Moderate	Hard
TLNet (Qin, Wang, and Lu 2019)	CVPR' 19	7.64	4.37	3.74	13.71	7.69	6.73
Stereo R-CNN (Li, Chen, and Shen 2019)	CVPR' 19	47.58	30.23	23.72	61.92	41.31	33.42
ZoomNet (Xu et al. 2020)	AAAI' 20	55.98	38.64	30.97	72.94	54.91	44.14
IDA-3D (Peng et al. 2020)	CVPR' 20	45.09	29.32	23.13	61.87	42.47	34.59
Disp-RCNN (Sun et al. 2020)	CVPR' 20	59.58	39.34	31.99	74.07	52.34	43.77
DSGN (Chen et al. 2020)	CVPR' 20	73.50	52.18	45.14	82.90	65.05	56.60
CDN (Garg et al. 2020)	NeurIPS' 20	74.52	54.22	46.36	83.32	66.24	57.65
RTS-3D (Li, Su, and Zhao 2021)	AAAI' 21	58.51	37.38	31.12	72.17	51.79	43.19
Disp-RCNN-flb* (Chen et al. 2021)	T-PAMI' 21	68.21	45.78	37.73	79.76	58.62	47.73
SNVC (Ours)	AAAI' 22	**78.54**	**61.34**	**54.23**	**86.88**	**73.61**	**64.49**

Table 2: Overall system performance evaluated with AP_{3D} and AP_{BEV}, and compared with SOTA stereo 3D object detection approaches on KITTI *test* set (official KITTI leader-board).The official recall values are used.
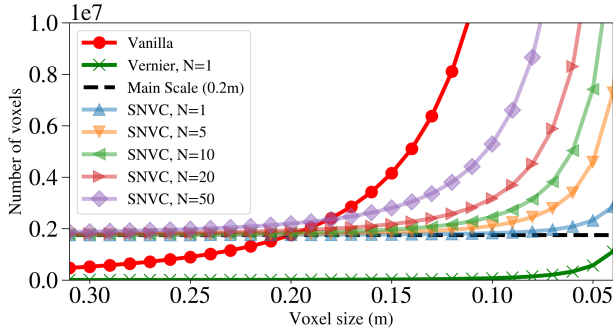

Figure 8: A comparison of needed number of voxels (NoVs) w and w/o our MR strategy. Vanilla: NoVs used for representing a global scene using a uniform resolution. Vernier: NoVs used to model a local scene in our $\mathcal{V}$. Main scale: NoVs used in our $\mathcal{M}$ with a resolution of 0.2m. SNVC, N=x: total NoVs as a sum of NoVs used in $\mathcal{M}$ plus that used in building local scene for x proposals.

voxel grid for the global scene as used in the *scene-centric* approaches needs intimidating memory that is not scalable to smaller voxel size. A comparison of the required number of voxels (NoVs) required in such systems and that needed in our proposed MR system is shown in Fig. 8. The NoVs used in the vanilla approach is $\mathcal{Q}(L_g, W_g, H_g, \Delta) = \frac{L_g * W_g * H_g}{\Delta^3}$ where $[L_g, W_g, H_g] = [60m, 60m, 4m]$ are the global spatial range and Δ is the voxel size. In contrast, the NoVs used in our approach is $N_L * N_H * N_W * N + \mathcal{Q}(L_g, W_g, H_g, \Delta_g)$ where Δ_g is the used global voxel size in the main scale network (0.2m) and N is the number of selected coarse proposals. Note that our SNVC uses significantly fewer NoVs than the vanilla approach to achieve a finer representation beyond Δ_g, even for a large number of proposals (N=50). Our framework can thus model the important regions at a high resolution and keep the total memory usage tractable at the same time.

More ablation studies and results can be found in our SM which includes how $\mathcal{V}$ can be employed for *tracking-by-detection* using the predictions from the previous frame as proposals.

Method	$AP_{3D}/AP_{BEV}@R_{11}$		
	Easy	Moderate	Hard
RTS-3D	59.31/74.49	41.61/53.98	34.67/46.66
RTS-3D + $\mathcal{V}$	69.25/82.71	52.92/65.75	45.82/57.47
IDA-3D	53.59/69.28	36.79/50.11	32.34/43.31
IDA-3D + $\mathcal{V}$	64.13/80.81	48.89/62.19	43.00/55.03

Table 3: AP_{3D}/AP_{BEV} evaluated on the KITTI *val* split when using $\mathcal{V}$-A as a model-agnostic refinement module with other 3D object detectors.

Method	$AP_{3D}/AP_{BEV}@R_{11}$		
	Easy	Moderate	Hard
Center-only (K=1)	64.94/77.89	47.22/64.00	43.73/55.84
Part-based (K=9)	69.25/82.71	52.92/65.75	45.82/57.47

Table 4: The same evaluation for RTS-3D + $\mathcal{V}$ as in Tab. 3 with varying number of parts K.

(N_L, N_H, N_W)	$AP_{3D}/AP_{BEV}@R_{11}$		
	Easy	Moderate	Hard
(24, 16, 16)	33.65/43.80	27.84/37.38	24.30/32.64
(48, 16, 32)	60.10/73.25	44.31/56.19	37.45/53.30
(192, 32, 128)	69.25/82.71	52.92/65.75	45.82/57.47

Table 5: The same evaluation for RTS-3D + $\mathcal{V}$ as in Tab. 3 with varying voxel size.

Conclusion

We introduce the idea of multi-resolution modeling to voxel-based stereo 3D object detection by modeling different regions with varying resolutions. This approach can keep the detection problem computationally tractable and can model important regions with smaller voxels to achieve high precision. A new instance-level model is designed, which samples candidate 3D locations and uses predicted object part coordinates to estimate a pose update. Our approach is validated to achieve state-of-the-art stereo 3D object detection performance and can perform model-agnostic refinement. For future study, instead of using a sampling grid with a fixed range, information from previous frames can be used to build a motion model that helps predict the future object location and provide a better clue on where to sample.

Acknowledgments

References

Andriluka, M.; Roth, S.; and Schiele, B. 2008. People-tracking-by-detection and people-detection-by-tracking. In *2008 IEEE Conference on computer vision and pattern recognition*, 1–8. IEEE.

Bellotto, N.; Sommerlade, E.; Benfold, B.; Bibby, C.; Reid, I.; Roth, D.; Fernández, C.; Van Gool, L.; and Gonzalez, J. 2009. A distributed camera system for multi-resolution surveillance. In *2009 Third ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC)*, 1–8. IEEE.

Blaha, M.; Vogel, C.; Richard, A.; Wegner, J. D.; Pock, T.; and Schindler, K. 2016. Large-scale semantic 3d reconstruction: an adaptive multi-resolution model for multi-class volumetric labeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3176–3184.

Brazil, G.; and Liu, X. 2019. M3d-rpn: Monocular 3d region proposal network for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 9287–9296.

Chen, L.; Sun, J.; Xie, Y.; Zhang, S.; Shuai, Q.; Jiang, Q.; Zhang, G.; Bao, H.; and Zhou, X. 2021. Shape Prior Guided Instance Disparity Estimation for 3D Object Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Chen, X.; Kundu, K.; Zhu, Y.; Berneshawi, A. G.; Ma, H.; Fidler, S.; and Urtasun, R. 2015. 3d object proposals for accurate object class detection. In *Advances in Neural Information Processing Systems*, 424–432. Citeseer.

Chen, X.; Ma, H.; Wan, J.; Li, B.; and Xia, T. 2017. Multi-view 3d object detection network for autonomous driving. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 1907–1915.

Chen, Y.; Liu, S.; Shen, X.; and Jia, J. 2020. Dsgn: Deep stereo geometry network for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12536–12545.

Cheng, B.; Xiao, B.; Wang, J.; Shi, H.; Huang, T. S.; and Zhang, L. 2020. Higherhrnet: Scale-aware representation learning for bottom-up human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5386–5395.

Choy, C. B.; Xu, D.; Gwak, J.; Chen, K.; and Savarese, S. 2016. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *European conference on computer vision*, 628–644. Springer.

Garg, D.; Wang, Y.; Hariharan, B.; Campbell, M.; Weinberger, K. Q.; and Chao, W.-L. 2020. Wasserstein Distances for Stereo Disparity Estimation. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M. F.; and Lin, H., eds., *Advances in Neural Information Processing Systems*, volume 33, 22517–22529. Curran Associates, Inc.

Geiger, A.; Lenz, P.; and Urtasun, R. 2012. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3354–3361. IEEE.

Guo, X.; Shi, S.; Wang, X.; and Li, H. 2021. Liga-stereo: Learning lidar geometry aware representations for stereo-based 3d detector. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 3153–3163.

Ke, L.; Li, S.; Sun, Y.; Tai, Y.-W.; and Tang, C.-K. 2020. Gsnet: Joint vehicle pose and shape reconstruction with geometrical and scene-aware supervision. In *European Conference on Computer Vision*, 515–532. Springer.

Kundu, A.; Li, Y.; and Rehg, J. M. 2018. 3d-rcnn: Instance-level 3d object reconstruction via render-and-compare. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3559–3568.

Laine, S.; and Karras, T. 2010. Efficient sparse voxel octrees. *IEEE Transactions on Visualization and Computer Graphics*, 17(8): 1048–1059.

Lang, A. H.; Vora, S.; Caesar, H.; Zhou, L.; Yang, J.; and Beijbom, O. 2019. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12697–12705.

Li, P.; Chen, X.; and Shen, S. 2019. Stereo r-cnn based 3d object detection for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7644–7652.

Li, P.; Su, S.; and Zhao, H. 2021. RTS3D: Real-time Stereo 3D Detection from 4D Feature-Consistency Embedding Space for Autonomous Driving. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 1930–1939.

Li, S.; Ke, L.; Pratama, K.; Tai, Y.-W.; Tang, C.-K.; and Cheng, K.-T. 2020. Cascaded deep monocular 3D human pose estimation with evolutionary training data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6173–6183.

Li, S.; Yan, Z.; Li, H.; and Cheng, K.-T. 2021. Exploring intermediate representation for monocular vehicle pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1873–1883.

Li, Z.; Wang, F.; and Wang, N. 2021. LiDAR R-CNN: An Efficient and Universal 3D Object Detector. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7546–7555.

Liang, M.; Yang, B.; Chen, Y.; Hu, R.; and Urtasun, R. 2019. Multi-task multi-sensor fusion for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7345–7353.

Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; and Dollár, P. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, 2980–2988.

Liu, L.; Lu, J.; Xu, C.; Tian, Q.; and Zhou, J. 2019. Deep fitting degree scoring network for monocular 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1057–1066.

Liu, L.; Wu, C.; Lu, J.; Xie, L.; Zhou, J.; and Tian, Q. 2020. Reinforced axial refinement network for monocular 3d object detection. In *European Conference on Computer Vision*, 540–556. Springer.

Lopez-Paz, D.; Bottou, L.; Schölkopf, B.; and Vapnik, V. 2015. Unifying distillation and privileged information. *arXiv preprint arXiv:1511.03643*.

Lu, Y.; Ma, X.; Yang, L.; Zhang, T.; Liu, Y.; Chu, Q.; Yan, J.; and Ouyang, W. 2021. Geometry uncertainty projection network for monocular 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 3111–3121.

Moon, G.; Chang, J. Y.; and Lee, K. M. 2019. Posefix: Model-agnostic general human pose refinement network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7773–7781.

Peng, W.; Pan, H.; Liu, H.; and Sun, Y. 2020. Ida-3d: Instance-depth-aware 3d object detection from stereo vision for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13015–13024.

Qi, C. R.; Liu, W.; Wu, C.; Su, H.; and Guibas, L. J. 2018. Frustum pointnets for 3d object detection from rgb-d data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 918–927.

Qin, Z.; Wang, J.; and Lu, Y. 2019. Triangulation learning network: from monocular to stereo 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7615–7623.

Reading, C.; Harakeh, A.; Chae, J.; and Waslander, S. L. 2021. Categorical depth distribution network for monocular 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8555–8564.

Riegler, G.; Osman Ulusoy, A.; and Geiger, A. 2017. Octnet: Learning deep 3d representations at high resolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3577–3586.

Roddick, T.; Kendall, A.; and Cipolla, R. 2019. Orthographic Feature Transform for Monocular 3D Object Detection. In Sidorov, K.; and Hicks, Y., eds., *Proceedings of the British Machine Vision Conference (BMVC)*, 59.1–59.13. BMVA Press.

Seitz, S. M.; and Dyer, C. R. 1999. Photorealistic scene reconstruction by voxel coloring. *International Journal of Computer Vision*, 35(2): 151–173.

Shi, S.; Wang, X.; and Li, H. 2019. Pointrcnn: 3d object proposal generation and detection from point cloud. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 770–779.

Snow, D.; Viola, P.; and Zabih, R. 2000. Exact voxel occupancy with graph cuts. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No. PR00662)*, volume 1, 345–352. IEEE.

Sun, J.; Chen, L.; Xie, Y.; Zhang, S.; Jiang, Q.; Zhou, X.; and Bao, H. 2020. Disp r-cnn: Stereo 3d object detection via shape prior guided instance disparity estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10548–10557.

Sun, K.; Xiao, B.; Liu, D.; and Wang, J. 2019. Deep high-resolution representation learning for human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5693–5703.

Vogiatzis, G.; Torr, P. H.; and Cipolla, R. 2005. Multi-view stereo via volumetric graph-cuts. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, 391–398. IEEE.

Wang, J.; Sun, K.; Cheng, T.; Jiang, B.; Deng, C.; Zhao, Y.; Liu, D.; Mu, Y.; Tan, M.; Wang, X.; et al. 2020. Deep high-resolution representation learning for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*.

Wang, L.; Du, L.; Ye, X.; Fu, Y.; Guo, G.; Xue, X.; Feng, J.; and Zhang, L. 2021a. Depth-conditioned Dynamic Message Propagation for Monocular 3D Object Detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 454–463.

Wang, Y.; Chao, W.-L.; Garg, D.; Hariharan, B.; Campbell, M.; and Weinberger, K. Q. 2019. Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8445–8453.

Wang, Y.; Yang, B.; Hu, R.; Liang, M.; and Urtasun, R. 2021b. PLUME: Efficient 3D Object Detection from Stereo Images. *arXiv preprint arXiv:2101.06594*.

Weng, X.; and Kitani, K. 2019. Monocular 3d object detection with pseudo-lidar point cloud. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 0–0.

Xu, B.; and Chen, Z. 2018. Multi-level fusion based 3d object detection from monocular images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2345–2353.

Xu, Z.; Zhang, W.; Ye, X.; Tan, X.; Yang, W.; Wen, S.; Ding, E.; Meng, A.; and Huang, L. 2020. Zoomnet: Part-aware adaptive zooming neural network for 3d object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 12557–12564.

Yan, Y.; Mao, Y.; and Li, B. 2018. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10): 3337.

You, Y.; Wang, Y.; Chao, W.-L.; Garg, D.; Pleiss, G.; Hariharan, B.; Campbell, M.; and Weinberger, K. Q. 2019. Pseudo-LiDAR++: Accurate Depth for 3D Object Detection in Autonomous Driving. In *ICLR*.

Zhou, Y.; and Tuzel, O. 2018. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4490–4499.