Range Conditioned Dilated Convolutions for Scale Invariant 3D Object Detection

Alex Bewley[*1], **Pei Sun**[*2], **Thomas Mensink**[1], **Dragomir Anguelov**[2], **Cristian Sminchisescu**[1]
[1]Google Research, [2]Waymo LLC
{bewley, mensink, sminchisescu}@google.com, {peis, dragomir}@waymo.com

Abstract:
This paper presents a novel 3D object detection framework that processes LiDAR data directly on its native representation: *range images*. Benefiting from the compactness of range images, 2D convolutions can efficiently process dense LiDAR data of a scene. To overcome scale sensitivity in this perspective view, a novel range-conditioned dilation (RCD) layer is proposed to dynamically adjust a continuous dilation rate as a function of the measured range. Furthermore, localized soft range gating combined with a 3D box-refinement stage improves robustness in occluded areas, and produces overall more accurate bounding box predictions. On the public large-scale Waymo Open Dataset, our method sets a new baseline for range-based 3D detection, outperforming multiview and voxel-based methods over all ranges with unparalleled performance at long range detection.

Keywords: Autonomous Driving, 3D Detection, Range Image.

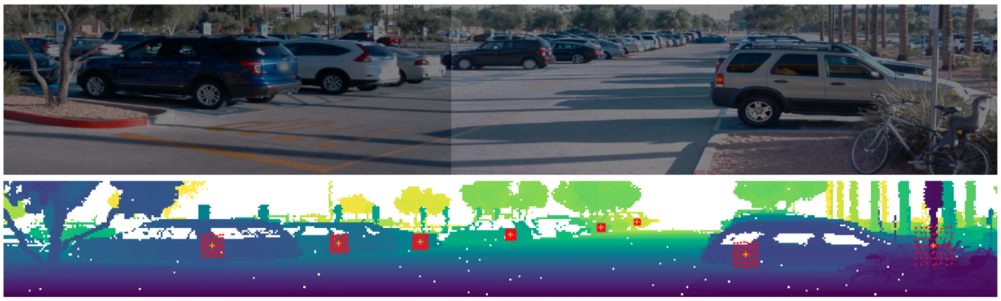


Figure 1: **Top:** Reference RGB images of a scene. *For illustration purposes only.* **Bottom:** Range image showing the dynamic sampling of our proposed RCD layer at selected positions. Here the measured range at the yellow + is used to govern a scale of the local receptive field towards a geometrically consistent sample density at any range.

1 Introduction

One of the most exciting opportunities at the intersection of robotics and machine learning is autonomous driving, where the detection of objects is of critical importance. Catalyzed by the proliferation of high-density LiDAR sensors, algorithms for 3D detection from pointcloud data have gained significant attention in recent years. Several meta-architectures are used for addressing 3D object detection. Firstly, voxelization methods [1, 2, 3] typically bin sparse Cartesian coordinates into discrete voxels and later process them with subsequent 3D convolutions. While such approaches perform well in practice, they are impeded by the memory and computational demands of 3D convolutions in large scenes. A related meta-architecture is projecting the sparse points into a birds-eye view (BEV) [4, 5] in order to reduce the scene back to a 2D space while maintaining scale invariance at the price of lost information through quantization. Other meta-architectures are built upon

[*] Indicates equal contribution. Ordered alphabetically.

4th Conference on Robot Learning (CoRL 2020), Cambridge MA, USA.

the PointNet framework [6, 7] but suffer from issues around point sparsity at long range and often require inefficient custom operations for defining point neighborhoods [8].

This paper focuses on an a less explored alternative (in the context of 3D detection) using range images that exploits the intrinsic 2.5D manifold structure [9, 10] of raw 3D point data in its native spherical coordinate form, enabling efficient and direct 3D object detection. This *range image* representation is characterised as 3D Cartesian points projected onto unique pixels in a 2D spherical image where range is encoded as pixel values and their row and column indices correspond to inclination and azimuth angles respectively. Operating on range images enjoys the benefits of applying mature 2D convolutional architectures and is naturally efficient due to the intrinsically compact representation [11]. Crucially, it does not suffer from the issue of sparsity at long range. However, known challenges for learning such as scale variation and occlusion need further consideration. This paper addresses these issues by proposing a novel convolutional layer with a scale aware dilation rate for efficient reuse of filter weights at different scales. This directly leverages the measure distance in range images to compensate for the corresponding scale change. Combined with soft range-based gating, both scale, and occlusion are appropriately handled within this framework. Occlusion is further addressed through the use of a second stage local box refinement module.

In this work, we present an efficient range image-based two-stage 3D object detector with the following key contributions. Firstly, a novel range conditioned dilated (RCD) convolutional operator is introduced that is capable of dynamically adjusting the local receptive field to provide a consistent scale relative to the convolutional kernel at any distance (see Figure 1). Second, a region convolutional neural network (RCNN) based second stage network is investigated in the context of range image-based 3D object detection; Finally, a new baseline is set for range image-based 3D object detection on a public dataset [12]. The introduced RCD based model performs especially well at long ranges (large distances), where voxel and sparse-point cloud based approaches suffer from point sparsity issues. Therefore, we believe this is the first work to combine a range image-based network with a RCNN second stage for 3D object detection.

2 Related Work

2.1 3D LiDAR Detection

While many works combine color images with LiDAR data [13, 14, 15], here we restrict our review to works that only process 3D LiDAR data.

Rasterized and Voxel methods. A popular way to do 3D object detection is to first project the points to birds-eye view (BEV) and constructs a 2D multi-channel image. The image is then processed by 2D CNN to get either BEV or 3D boxes. The transformation process is usually hand-crafted, some selected works MV3D [4], PIXOR [16], Complex YOLO [17]. VoxelNet [3] divides the point cloud into a 3D voxel grid and uses a PointNet-like network [6] to learn an embedding of the points inside each voxel. PointPillars [5], a compute efficient method, that divides the point cloud into 3D pillars and then extracts features similar as VoxelNet [3], is the most prolific detector in this category and serves as a comparison in our experiments.

Point based methods. Another paradigm of methods are point based detection. It processes the raw point cloud with point cloud feature extraction methods like PointNet++ [7], Sparse Convolution [18], and then regresses 3D boxes in either downsampled BEV view or 3D point view directly. Some representative works are PointRCNN [19], PVRCNN [20], STD [21] or SA-SSD [22] which uses pointwise supervision for training a voxel based backbone. Our method is benchmarked against PVRCNN which is currently the top detector on the KITTI dataset and also benchmarked on Waymo Open Dataset [12].

Range image based methods. The range image is compact and does not suffer from sparsity related issues which is the main challenge when developing 3D algorithms. This representation is under-explored because a) range image based detectors require more data to train [11]. b) generating high quality range images is non-trivial without knowing raw sensor information such as laser scan pattern, relative position at each laser shot. Both of these are addressed by the Waymo Open Dataset [12]. The primary representative work is LaserNet [11] which benchmarks on a private dataset. Range image based detection algorithms need to deal with scale variance (near range objects are larger) and occlusions.

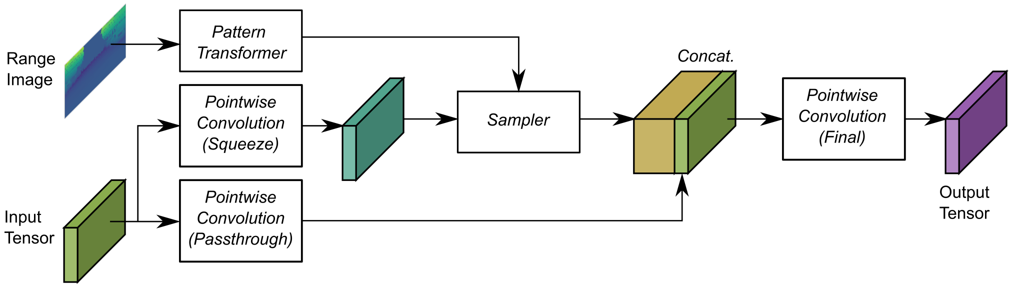

Range Image
Pattern Transformer
Pointwise Convolution (Squeeze)
Sampler
Concat.
Pointwise Convolution (Final)
Input Tensor
Pointwise Convolution (Passthrough)
Output Tensor

Figure 2: An overview of the range-conditioned dilation block detailing the inputs and outputs the various modules (illustrated with white boxes). The sampler is solely responsible for the spatial processing of the input tensor where the receptive field is driven by the input range image.

2.2 Adaptive and Dilated Convolutions

The seminal work of Jaderberg *et al.* on transforming the input signal using spatial transformers [23] has led to several subsequent methods. For example, for dilated convolutions, the dilation rate could be learned per filter and layer [24], or related to the size of the 3D environment using RGBD as input [25]. Wang *et al.* [26] use depth as a constant weighting factor of the influence of neighboring pixels in convolutional layers and max-pooling layers. More generically, dynamic functions could be learned to generate the weights of the convolutional layer conditioned on its input values [27, 28, 29, 30]. Ding *et al.* [30] propose a dynamic filter framework which simulates a dilated convolution with integer shifts to the feature maps, these are then combined using weights from RGB image features. Here a fixed set of dilations are chosen a priori similar the atrous spatial pyramid pooling (ASPP) in DeepLab [31]. In contrast, our method takes advantage of the available range observations to rescale the dilation rate enabling the reuse of kernel weights across multiple scales. This can be viewed as an extension of the 1D distance based input filtering of Beyer *et al.* [32] to 3D pointclouds represented as a range image. Furthermore, our sampler is continuous and adapts through the course of training mitigating the need to select a set of fixed dilation rates.

3 Range Condition to Learn Scale Invariant Features

This section describes the proposed range conditioned dilated convolutional block. This block can be placed anywhere in a larger network where a *range image* is available.

3.1 Range Conditioned Dilated Convolution

The RCD block accounts for the scale variation when observing objects at different distances by dynamically adjusting the spatial extent of a convolution via its dilatation rate. The dilation is scaled using a *continuous* function of the measure range requiring the replacement of discrete pixel index lookup (used in regular kernels) with a form of spatial sampling that is simultaneously sparse, local and conditioned on the observed range. As the dilation rate is continuous as opposed to integer-valued, a specialization of the spatial transformer [23] is adapted to adjust the spatial scale of the sampling pattern. This sampling is applied densely to every pixel to create a form of deformable convolution [31, 33] that is conditioned on the input range. This section describes the RCD applied to 2D range-images relevant for this paper, however without loss of generality the equivalent operations can be applied to other dimensional inputs. Figure 2 provides a high-level overview of the RCD block with the individual components detailed below.

Pattern Transformer: takes in a range-image $R \in \mathbb{R}^{H \times W}$ with height H and width W. Internally parameters in the form of N sparse 2D points $\mathcal{G} \in \mathbb{R}^{N \times 2}$ are maintained to represent the relative spatial sampling pattern for the RCD convolutional kernel. The sampling pattern $\mathcal{G}$ is a set of learnable parameters which is initialized to be a uniform grid with mean $(0, 0)$. This relative sampling pattern $\mathcal{G}$ is shared for all pixel locations and then individually transformed using the input range image as follows:

$$\mathcal{S} = \sigma(R, \lambda) \cdot \mathcal{G} + \mathcal{P}, \tag{1}$$

where (via broadcasting[1]) $\mathcal{S} \in \mathbb{R}^{H \times W \times N \times 2}$, $\mathcal{P} \in \mathbb{R}^{H \times W \times 2}$ represents pixel coordinates and $\sigma(R, \lambda)$ is the following trigonometric function applied to all range values $r_i = R(i), i \in \mathcal{P}$:

$$\sigma(r_i, \lambda) = \arctan(\lambda / r_i), \tag{2}$$

where $\lambda \in \mathbb{R}^+$ is a learnable scalar parameter representing the nominal width of the cross-sectional area covered by the receptive field at any range r_i. The dilation multiplier $\sigma(r_i, \lambda)$ is visualized for several values of λ in Figure 3.

Sampler: At the core of the RCD block is a sampler that is responsible for gathering spatial information from its input analogous to the dilated kernel sampling, ready for an inner product with kernel weights as with conventional convolutions. Given input feature tensor $X_s \in \mathbb{R}^{H \times W \times C}$, input sampling is performed N times for all pixel locations to produce a resampled tensor $\hat{X}_s \in \mathbb{R}^{H \times W \times N \times C}$, where C is the dimensionality of pixelwise features in X_s. While any sampling kernel can be applied, the bilinear sampling is chosen for its efficiency [27] and its spatial partial derivatives [23] allow for loss gradients to flow back to the size and spatial parameters λ and $\mathcal{G}$.

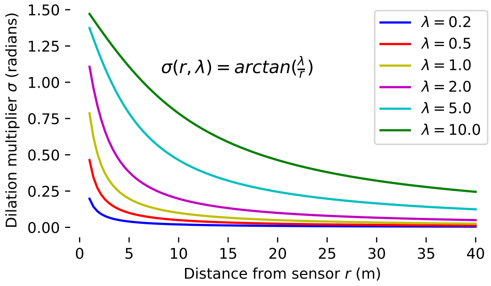


Figure 3: The rate of the convolutional filter dilation as a function of both the distance and the nominal object size λ in meters. Note multiplier is in units of radians and should be adjusted to pixels using the appropriate LiDAR angular resolution.

The coordinates contained in $\mathcal{S}$ may include samples extending beyond the width or height of the input range image. Samples extending above and below image boundaries are clamped to the first and last row respectively. For samples beyond the left and right boundaries, horizontal angular wrapping is performed, taking advantage of the range imageś 360° view.

In parallel to scaling the perceptive field, the range is also used to weight the magnitude of sampled features. This mechanism serves as a form of soft range gating (SRG) to prevent down-stream training difficulties caused by distractors from near-occlusion where neighboring pixels have a substantial difference in range. The weighting for SRG follows a Gaussian distribution, where for a given pixel location i with corresponding sample location $j \in \mathcal{S}(i)$, a feature masking weight:

$$\bar{\boldsymbol{x}}_{ij} = \hat{x}_{ij}\, \mathcal{N}(\hat{r}_j; r_i, \gamma), \tag{3}$$

where $\mathcal{N}$ denotes the Gaussian probability density function, evaluated at the range of the integer spatial location r_i, with mean $\hat{r}_j = R(j)$ being the bilinearly interpolated range value at the sample location j. The variance γ is a learnable parameter that controls the length-scale of the soft range gate (initialized to 1 meter). This decreases the importance of distant points, which are likely from a different object to the point at the center of the convolution.

Pointwise Convolutions: Also known as 1×1 convolutions [34] are used in multiple parts of the RCD block. See Figure 2 for their position within the RCD block. Firstly, two pointwise convolutions (PConv) partition the input tensor for the sampler input and concatenation with its output. This arrangement is inspired by the two stream hypothesis [35] where the sampler with its input PConv and the pass-through PConv are responsible for spatial and recognition processes respectively. On a practical note, the PConv feeding into the spatial sampler projects the input into lower dimensional features, significantly reducing the computation and memory requirements (in all RCD experiments $C = 3$). The output of the sampler is reshaped to $\mathbb{R}^{H \times W \times NC}$ and then concatenated along the last dimension of the result of the pass-through PConv. The final PConv completes the convolutional operation as the input contains the dilated resampling of the input and is reshaped to pack the spatial samples into the channel dimension. The weights in this final PConv are essentially the weights of the RCD convolution making the final output equal to:

$$\mathbf{h}_i = f(W_f^s \mathbf{x}_i^s + W_f^p \mathbf{x}_i^p), \tag{4}$$

where $\mathbf{x}_i^s, \mathbf{x}_i^p$ are the flattened spatially sampled features and passthrough feature respectively at pixel location i and $f(\cdot)$ represents Layer Normalization [36] followed by the a non-linear exponential linear unit [37].

[1] Singular tensor shapes are omitted for brevity.

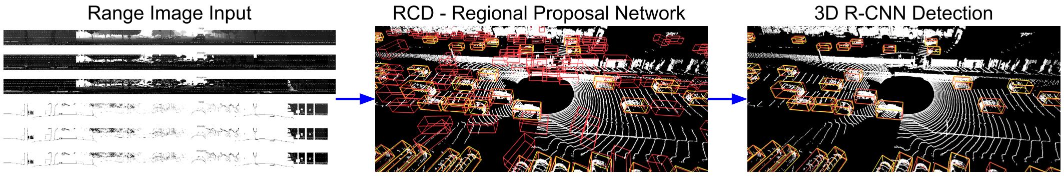


Figure 4: Two stage design: the input range images (size $64 \times 2650 \times 8$) are first provided to the RCD-RPN which generates a 3D box proposal corresponding to each point. High scoring proposals (red boxes) are sent to the second 3D RCNN stage for refinement.

4 Two Stage Detection Network

The RCD block is embedded within the region proposal network (RPN) stage of a two stage detection network, see Figure 4. The first stage is an adaption of the LaserNet [11] architecture, with RCD blocks, containing both a foreground classification and box regressor head. The second stage refines high-scoring box proposals similar in spirit to [19, 38].

4.1 First stage: RCD-RPN

The first stage is a fully convolutional deep layer aggregation network architecture [39, 11], with RCD blocks replacing the initial convolutional layers at multiple scales (see supplementary). To achieve a larger receptive field and faster compute, we downsample more aggressively along the horizontal axis only with a $[1, \cdot]$ pooling kernel. This befits the typical structure of range images with few rows and many columns. The RCD-RPN network yields four outputs per point: a foreground classification score, two additional scores representing the probability its top and left neighboring pixels lay on the same object, and a predicted 3D box represented by a 7D vector $(x, y, z, h, w, l, \theta)$.

The focal loss [40]: $L_f(p_i) = -\alpha_i(1 - p_i)^\gamma \log(p_i)$ is employed, where $p_i = s_i$ if i is positive, and $p_i = 1 - s_i$ otherwise, to handle class imbalance (using α_i) and focus on difficult boxes (using focus parameter $\gamma = 2$), for the three classification scores:

$$L_\text{f} = \frac{1}{|\mathcal{P}|} \sum_i [L_f(p_i) + L_f(p_i^\text{t}) + L_f(p_i^\text{l})], \tag{5}$$

where $|\mathcal{P}|$ is the number of valid points in the range image, and $p_i, p_i^\text{t}, p_i^\text{l}$ are the scores for foreground class and top and left similarity classification respectively.

For regressing towards the 3D box parameters, the bin-loss (L_{bin}) [19] divides the distance to the center point (x, y, z) and the heading of the ground truth box into bins and performs bin classification first, followed by a regression within each bin (see [19] for more details). It has been shown that the bin-loss converges faster and achieves higher recall than regression based loss. For each point i in the range image, we use its predicted 3D box b_i as follows:

$$L_\text{b} = \frac{1}{N} \sum_i \tfrac{1}{n_i} L_{bin}(b_i, b_j), \tag{6}$$

where N is the number of ground-truth boxes, we sum over all pixels in the range image, n_i is the number of points in the target ground-truth box b_j that contains point i.

4.2 Second stage: 3D RCNN

The second stage refines the initial box proposals, by using an RCNN stage similar to [41, 19, 38]. In general a two stage design greatly improves box prediction accuracy and further mitigates the occlusion effect. For each box, the raw points are extracted and the per point feature embeddings, predicted box parameters and semantic classification scores are reused from RCD-RPN. Each box is transformed into the canonical box frame [19] and divided into a a fixed 3D grid ($12 \times 8 \times 6$). All the features in each grid cell are pooled to yield a single feature per grid cell. For the semantic features (points, classification score and RCD-RPN box parameters) average pooling is used, while the feature embeddings are max pooled. Then a 3D convolution layer, followed by downsampling and a fully connected layer generates the final box parameters and classification score.

Method	**Easy**	**Moderate**	**Hard**
LaserNet	79.19	74.52	68.45
RCD (Ours)	82.26	75.83	69.91
RCD-FT (Ours)	85.37	82.61	77.80

Table 1: Comparison to official LaserNet BEV results on KITTI Car testset. No test-time augmentation or additional training data is used for RCD. RCD method is trained from scratch on KITTI while RCD-FT is finetuned from WOD pretraining.

During training, it is important to sub-sample boxes from the first stage to have an efficient training pipeline. First, the range image is divided into a top and bottom half, and per column the highest scoring box-proposal is kept. This coarse partitioning maintains spatial diversity for training the second stage. The remaining proposals are then further reduced to 50 positive (with intersection over union $IoU \geq .5$) and 50 negative ($IoU < .5$) proposals. During inference, the top 400 boxes are kept after running non-maximum-suppression on the RCD-RPN box proposals.

The second stage 3D RCNN also contains classification loss which is averaged over all box proposals ($M = 100$ during training) selected from RCD-RPN:

$$L_{\text{cls}} = \frac{1}{M} \sum_k L_{ce}(s_k, y_k), \tag{7}$$

where s_k denotes the refined classification score, y_k the ground-truth class for box k, and $L_{ce}(\cdot)$ the cross-entropy loss. For the refined bounding box parameters b_k a combination of losses is used, including a residual loss similar to [41] for box centers (x, y, z) and box dimensions (w, h, l), while heading (θ) regression uses the bin loss [19]. These losses are averaged over all proposals:

$$L_{\text{reg}} = \frac{1}{M} \sum_k L_{\text{box}}(b_j, b_k), \tag{8}$$

where the box regression loss L_{box} is a sum of smooth $L1$ loss for center positions and normalized length, width and height prediction and a bin loss for heading prediction.

4.3 Joint Training:

Both the RPN and RCNN networks are trained jointly with the final objective as:

$$L = \underbrace{L_{\text{f}} + L_{\text{b}}}_{\text{RCD-RPN}} + \underbrace{L_{\text{cls}} + L_{\text{reg}}}_{\text{3D RCNN}}. \tag{9}$$

5 Experiments

We primarily benchmark on the Waymo Open Dataset (WOD) [12] as it released its raw data in range image format while other datasets such as KITTI [42] or nuScenes [43] provide pointclouds. Converting a pointcloud back into a range image requires known laser angles and accurate point-wise timing to offset for the relative vehicle pose when in motion. Simply projecting Cartesian points to their spherical counterparts [44] results in either significant pixel collisions or many holes depending on the choice of range-image resolution. For the KITTI LiDAR with 64 individual lasers, their unique vertical and inclination offsets are recovered with the Hough-transform and points are grouped to form the rows in the range image. Table 1 shows the BEV results on the KITTI detection testset. See Section E.2 in Appendix for more details. As reported in [11], we also observe that small datasets are prone to overfitting for range image detectors and resume our evaluation using the larger WOD dataset.

WOD captures multiple major cities in the U.S., under a variety of weather conditions and across different times of the day. The dataset provides a total number of 1000 sequences of 20s duration each, sampled at 10Hz, with train/validation split of 798/202 sequences. The effective annotation radius for 6M vehicles across these sequences is 75 meters. For our experiments, we evaluate both Average Precision (AP) and AP weighted by heading (APH) [12] in 3D and BEV for vehicles on the WOD validation set and 3D detection metrics using the public evaluation server for the test set.

	3D AP (IoU=0.7)				3D APH (IoU=0.7)				BEV AP (IoU=0.7)				BEV APH (IoU=0.7)			
	All	$r_{\leq 30}$	r_{30-50}	$r_{\geq 50}$	All	$r_{\leq 30}$	r_{30-50}	$r_{\geq 50}$	All	$r_{\leq 30}$	r_{30-50}	$r_{\geq 50}$	All	$r_{\leq 30}$	r_{30-50}	$r_{\geq 50}$
P.Pillars [5]	56.62	81.0	51.8	27.9	-	-	-	-	75.57	92.1	74.1	55.5	-	-	-	-
DynVox [45]	59.29	84.9	56.1	31.1	-	-	-	-	77.18	93.0	76.1	57.7	-	-	-	-
MVF [45]	62.93	86.3	60.0	36.0	-	-	-	-	80.40	93.6	79.2	63.1	-	-	-	-
PV-RCNN [20]	**70.30**	**91.9**	**69.2**	42.2	**69.69**	**91.3**	**68.5**	41.3	82.96	**97.4**	**83.0**	65.0	82.06	**96.7**	80.0	63.2
LaserNet*	52.11	70.9	52.9	29.6	50.05	68.7	51.4	28.6	71.19	83.9	71.4	54.5	67.66	80.7	68.4	51.4
RCD (Ours)	69.59	87.2	67.8	**46.1**	69.16	86.8	67.4	**45.5**	**83.35**	93.5	82.3	**67.9**	**82.64**	93.0	**81.6**	**66.6**

Table 2: Comparison of methods for vehicle detection on the Waymo Open Dataset (WOD) validation set for 3D detection with 7DOF boxes. The **best** and second best results are highlighted in bold and underlined respectively. (*) Our implementation of LaserNet [11]. Columns with r show breakdown of metrics by range (in meters).

5.1 Baseline Methods

The full RCD model is compared to an equivalent two-stage detector baseline without SRG and RCD layers replaced with a fixed 7×7 dilated convolution (set to a dilation rate of 3 which is shown to have best RPN performance). Furthermore, a selection of state of art methods from each mainstream category in 3D object detection algorithms are compared on WOD.

LaserNet [11]: LaserNet is a 2D CNN-based singleshot 3D object detector operating on LiDAR range-images. It showed improvements on a large private dataset. With no publicly available implementation for this method, we use a variant of our RPN sub-network with normal 2D convolutions, ResNet blocks, adaptive NMS, and trained with multi-model box regression loss as described in [11].

Point Pillars (P.Pillars) [11]: Another single stage detector which utilizes PointNets [6] to encode a pointcloud scene representation organized in vertical columns in the BEV. Metrics from [45].

Multi-View Fusion (MVF) [45]: This method fuses Cartesian view features and spherical view features. It shows significant improvements on long range detection because of the spherical view features. We share the same findings in our method as our method is perspective only.

Point-Voxel (PV-RCNN) [20]: a recently proposed method combining PointNets [7] and a sparse convolution RPN backbone with a similar second stage RCNN refinement network.

5.2 Implementation details

For the RCD layer we always use $N = 64$ number of samples initialized as a 8×8 grid. With the number of channels kept at 64 for the inputs and outputs of the RCD block, the entire block consumes 23K FLOPs per pixel which is significantly lower than 262K FLOPs for an equivalent 2D convolutional layer, primarily due to PConv squeeze with only 3 filters. Our RPN network takes 74ms per frame compared to 301ms for our best-effort implementation of the RPN from PV-RCNN on a V100 GPU. All validation experiments use the Adam optimizer [46] for 350K iterations of batch-size 8 (or 17.5 epochs) with a learning rate starting from 6e-3 with a cosine decay end to end from scratch without any data augmentation. For submission to the WOD test-server our RCD model is trained for a total of 1 million iterations.

5.3 Discussion of Results

Among the baseline methods, only LaserNet [11] uses the range imagery directly, this makes LaserNet our direct comparison. The results of this are shown in Table 2. Even without RCNN our RPN achieves 57.2 AP, a significant improvement over the LaserNet range image detector. This is mainly because of the way RCD and SRG handle scale variance and occlusion in the range image view. As our method processes the range image in the perspective view it has fundamentally different characteristics compared to voxel or BEV projection based methods. Compared with the voxel based methods reported on WOD, our method shows greatest improvements in long range. This is due to the issue around voxel sparsity for distant objects corroborating with the findings of [45]. Additionally, RCD is better able to utilize contextual information in the range image to distinguish distant objects with few points. As a result of this, our method exhibits complementary performance to the state-of-the-art PV-RCNN method which has strongest performance in close range where point density is highest for voxel based detection.

	Level 1		Level 2	
	AP	APH	AP	APH
Second [47]	50.11	49.63	42.88	42.48
P.Pillars [5]	54.94	54.47	48.61	48.18
StarNet [48]	61.68	61.23	55.17	54.76
SA-SSD [22]	70.24	69.54	61.79	61.17
RCD 1M (Ours)	**71.97**	**71.59**	**65.06**	**64.70**

Table 3: Comparison of single frame methods for 3D vehicle detection on the Waymo Open Dataset (WOD) test set. Values are provided by the test server and divided into two levels of difficulty where Level 1 has at least five points per ground truth object and Level 2 can have as few as a single point.

Fixed dilation (rate=3)		✓						✓	
ASPP			✓						✓
RCD at start				✓	✓	✓	✓		
With SRG					✓	✓	✓		
RCD at multiple scales						✓	✓		
3D RCNN							✓	✓	✓
3D AP	52.3	53.8	45.8	54.9	55.0	57.2	**69.6**	63.6	64.4
3D APH	50.3	51.8	45.2	52.5	52.9	56.7	**69.2**	63.1	63.9

Table 4: Ablation study showing the progressive improvements with proposed contributions. Measured 3D AP and APH are from the WOD validation sequences. RCD at multiple scales uses 4 blocks with one at the start of the RPN and others at downsampled resolutions (see supplementary).

WOD Leaderboard: Recently, an unlabelled test set was released to the public by Waymo for benchmarking 3D object detectors via an online evaluation service. Table 3 shows the performance of our best model trained for 1 million iterations compared to other published single LiDAR frame vehicle detectors on the leaderboard[2]. The public leaderboard divides the detection results into two difficulty levels based on the number of points within the annotated boxes. Level 1 has at least five points per ground truth object and Level 2 boxes may only have a single point. Our range based RCD model significantly outperforms other methods for both difficulty levels on this public benchmark.

Ablation Study: The merits of the proposed RCD block is compared with standard dilated convolution. Furthermore, the proposed RCD layer relates to the exhausted dilation combination of differently sized receptive kernels in ASPP [31]. The key difference is that ASPP merges multiple fixed dilated convolutions together while RCD reuses the same set of filter weights over a continuum of dilations. We found that substituting our RCD for an ASPP block often resulted in overfitting or unstable training with only a single variant producing comparable results shown as the third column of results in Table 4. As ASPP uniformly applies multiple rates of dilation simultaneously its large receptive field is susceptible to distracting object in the periphery. Using the RCD with SRG at the start and throughout the backbone network achieves the best single stage performance while ASPP exhibited the worst performance. The last three columns show the effect of the two-stage framework. Interestingly, the 3D RCNN second stage is able to substantially recover from the under-performing performing ASPP based RPN. The RCNN also capitalizes on the high-quality proposals from the RPN network endowed with multiple RCD blocks.

6 Conclusions

We introduce RCD, a method for dynamically adjusting the dilation rate for use with LiDAR range images for scale invariant 3D object detection. An improved system relying on our new RCD block representation, and based on the two stage RCNN method [19], is the top performing range image based detection method, over all ranges, on the Waymo Open Dataset [12] benchmark for BEV and achieves competitive results in other tests. Specifically, our approach sets a new state-of-the-art for detecting vehicles at long distances as it benefits from the dense nature of the range image. Two directions of future research include, applying the proposed method with other common robotic depth sensors, like structured-light or time-of-flight cameras which also capture a 2D range image, and designing a hybrid PV-RCNN and RCD approach to efficiently obtain the best performance at all ranges.

[2] waymo.com/open/challenges/3d-detection/, results gathered on July 27^{th}, 2020

References

[1] M. Engelcke, D. Rao, D. Z. Wang, C. H. Tong, and I. Posner. Vote3deep: Fast object detection in 3d point clouds using efficient convolutional neural networks. In *ICRA*. IEEE, 2017.

[2] D. Z. Wang and I. Posner. Voting for voting in online point cloud object detection. In *Robotics: Science and Systems*, 2015.

[3] Y. Zhou and O. Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *CVPR*, pages 4490–4499, 2018.

[4] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia. Multi-view 3d object detection network for autonomous driving. In *CVPR*, 2017.

[5] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *CVPR*, pages 12697–12705, 2019.

[6] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, pages 652–660, 2017.

[7] C. R. Qi, L. Yi, H. Su, and L. J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *NeurIPS*, 2017.

[8] F. Engelmann, T. Kontogianni, and B. Leibe. Dilated Point Convolutions: On the Receptive Field Size of Point Convolutions on 3D Point Clouds. In *ICRA*, 2020.

[9] A. Bewley and B. Upcroft. Advantages of exploiting projection structure for segmenting dense 3d point clouds. In *Australian Conference on Robotics and Automation*, 2013.

[10] B. Wu, A. Wan, X. Yue, and K. Keutzer. Squeezeseg: Convolutional neural nets with recurrent crf for real-time road-object segmentation from 3d lidar point cloud. In *ICRA*, 2018.

[11] G. P. Meyer, A. Laddha, E. Kee, C. Vallespi-Gonzalez, and C. K. Wellington. Lasernet: An efficient probabilistic 3d object detector for autonomous driving. In *CVPR*, 2019.

[12] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *CVPR*, 2020.

[13] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas. Frustum pointnets for 3d object detection from rgb-d data. In *CVPR*, pages 918–927, 2018.

[14] J. Ku, M. Mozifian, J. Lee, A. Harakeh, and S. L. Waslander. Joint 3d proposal generation and object detection from view aggregation. In *IROS*. IEEE, 2018.

[15] Z. Wang and K. Jia. Frustum convnet: Sliding frustums to aggregate local point-wise features for amodal 3d object detection. In *IROS*. IEEE, 2019.

[16] B. Yang, W. Luo, and R. Urtasun. Pixor: Real-time 3d object detection from point clouds. In *CVPR*, 2018.

[17] M. Simony, S. Milzy, K. Amendey, and H.-M. Gross. Complex-yolo: An euler-region-proposal for real-time 3d object detection on point clouds. In *ECCV Workshops*, 2018.

[18] B. Graham, M. Engelcke, and L. van der Maaten. 3d semantic segmentation with submanifold sparse convolutional networks. *CVPR*, 2018.

[19] S. Shi, X. Wang, and H. Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *CVPR*, pages 770–779, 2019.

[20] S. Shi, C. Guo, L. Jiang, Z. Wang, J. Shi, X. Wang, and H. Li. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In *CVPR*, pages 10529–10538, 2020.

[21] Z. Yang, Y. Sun, S. Liu, X. Shen, and J. Jia. Std: Sparse-to-dense 3d object detector for point cloud. In *ICCV*, pages 1951–1960, 2019.

[22] C. He, H. Zeng, J. Huang, X.-S. Hua, and L. Zhang. Structure aware single-stage 3d object detection from point cloud. In *CVPR*, 2020.

[23] M. Jaderberg, K. Simonyan, A. Zisserman, et al. Spatial transformer networks. In *NeurIPS*, pages 2017–2025, 2015.

[24] Y. He, M. Keuper, B. Schiele, and M. Fritz. Learning dilation factors for semantic segmentation of street scenes. In *German Conference on Pattern Recognition*, pages 41–51. Springer, 2017.

[25] Y. Chen, T. Mensink, and E. Gavves. 3d neighborhood convolution: Learning depth-aware features for rgb-d and rgb semantic segmentation. In *3DV*, pages 173–182. IEEE, 2019.

[26] W. Wang and U. Neumann. Depth-aware cnn for rgb-d segmentation. In *ECCV*, 2018.

[27] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei. Deformable convolutional networks. In *ICCV*, pages 764–773, 2017.

[28] F. Strub, M. Seurin, E. Perez, H. De Vries, J. Mary, P. Preux, and A. CourvilleOlivier Pietquin. Visual reasoning with multi-hop feature modulation. In *ECCV*, pages 784–800, 2018.

[29] X. Zhu, H. Hu, S. Lin, and J. Dai. Deformable convnets v2: More deformable, better results. In *CVPR*, pages 9308–9316, 2019.

[30] M. Ding, Y. Huo, H. Yi, Z. Wang, J. Shi, Z. Lu, and P. Luo. Learning depth-guided convolutions for monocular 3d object detection. In *CVPR*, 2020.

[31] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *PAMI*, 40(4), 2017.

[32] L. Beyer, A. Hermans, and B. Leibe. DROW: Real-Time Deep Learning based Wheelchair Detection in 2D Range Data. *RA-L*, 2016.

[33] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. In *ICLR*, 2016.

[34] M. Lin, Q. Chen, and S. Yan. Network in network. *ICLR*, 2014.

[35] M. A. Goodale, A. D. Milner, et al. Separate visual pathways for perception and action. *Trends in Neurosciences*, 15, 1992. ISSN 01662236. doi:10.1016/0166-2236(92)90344-8.

[36] D.-A. Clevert, T. Unterthiner, and S. Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*, 2015.

[37] J. L. Ba, J. R. Kiros, and G. E. Hinton. Layer normalization. *preprint arXiv:1607.06450*, 2016.

[38] S. Shi, Z. Wang, X. Wang, and H. Li. Part-aˆ 2 net: 3d part-aware and aggregation neural network for object detection from point cloud. *PAMI*, 2019.

[39] F. Yu, D. Wang, E. Shelhamer, and T. Darrell. Deep layer aggregation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2403–2412, 2018.

[40] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. In *ICCV*, pages 2980–2988, 2017.

[41] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015.

[42] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013.

[43] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom. nuscenes: A multimodal dataset for autonomous driving. *arXiv preprint arXiv:1903.11027*, 2019.

[44] A. Milioto, I. Vizzo, J. Behley, and C. Stachniss. Rangenet++: Fast and accurate lidar semantic segmentation. In *IROS*. IEEE, 2019.

[45] Y. Zhou, P. Sun, Y. Zhang, D. Anguelov, J. Gao, T. Ouyang, J. Guo, J. Ngiam, and V. Vasudevan. End-to-end multi-view fusion for 3d object detection in lidar point clouds. In *CoRL*, 2019.

[46] D. Kingma and J. Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[47] Y. Yan, Y. Mao, and B. Li. Second: Sparsely embedded convolutional detection. *Sensors*, 2018.

[48] J. Ngiam, B. Caine, W. Han, B. Yang, Y. Chai, P. Sun, Y. Zhou, X. Yi, O. Alsharif, P. Nguyen, et al. Starnet: Targeted computation for object detection in point clouds. *arXiv preprint arXiv:1908.11069*, 2019.

[49] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[50] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

A RCD-RPN Backbone Network

As briefly described in the main paper the backbone of the RCD-RPN is a 2D convolutional network inspired by the deep layer aggregation network [39] used in LaserNet[11]. The architecture is composed of ResNet convolutional bottleneck units [49] with multiple steps of horizontal downsampling as shown in the upper blue section of Figure 5 (see caption for the number of ResNet units per downsampled scale). The aggregator modules take in two feature maps at different resolutions, upsampling the lower resolution via transposed convolutions.

Beyond substituting an initial convolution applied to the LiDAR input, the RCD block can be applied anywhere within the convolutional network. For example, replacing a ResNet block with an RCD block is considered at Res1, Res2, and Res3 of our multi-resolution backbone. The range image used to condition the amount of dilation at each pixel is coarsely downsampled using max-pooling, with the intention to favour distant objects with fewer pixels.

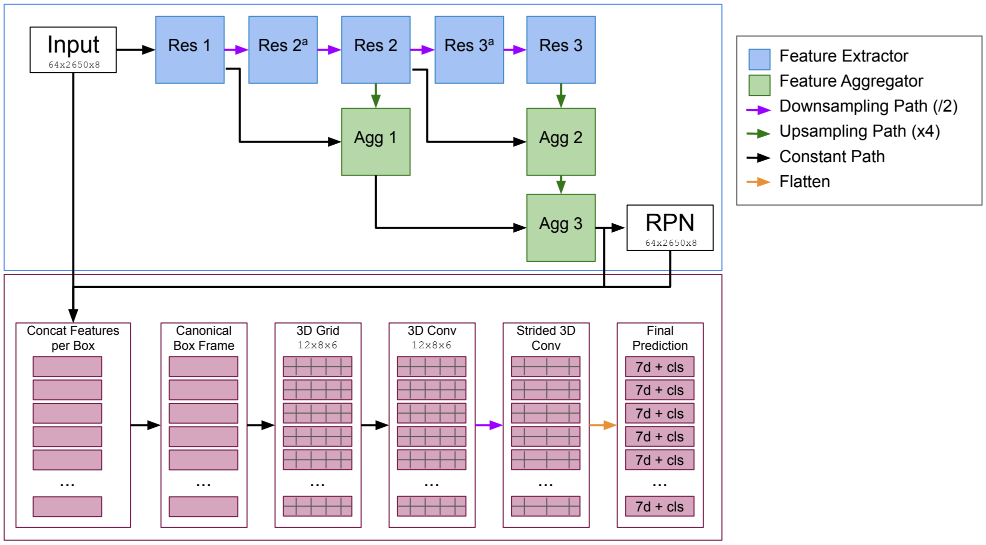


Figure 5: Illustrative detail of our detection pipeline. **RPN:** The upper blue section shows our 2D CNN backbone where the Resnet blocks and aggregation blocks contain either RCD layers or convolutational layers for the baseline. Number of bottlenecks units in each block: (Res1: 5), (Res2,Res2a: 7), (Res3,Res3a: 9), (Agg1,Agg2,Agg3: 4). Strides in each block: (Res1: 2), (Res2, Res2a: 2), (Res3, Res3a: 2). The bottleneck is the one described in [49] with bottleneck depth set to the bottleneck input channel size divided by 4. The output channel size from each layer are 64, 128, 256 corresponding to the block numbers. The final RPN head predicts box parameters and score. **RCNN:** The second stage operates independently per surviving box. The Cartesian point coordinates are combined with box parameters to transform features range-image into their box canonical frame. For each box, point features are voxelized and followed by 3D convolutions and final predictions as shown.

B Learnable Sampling Pattern Analysis

To further assess the effect of using a learnable, continuous dilated kernel sampling, Figure 6 shows the relative locations of the learnt spatial samples with their initial positions marked. Compared to the uniform sampling at initialization, inner sample points tend to increase the local concentration around the center with the outer points showing negligible or slight spread away from the central location. This indicates that the RCD model attempts to over-sample in the central region with sparser sampling in the extremities, similar to the Fovea in the human eye. Figure 7 plots the evolution of the learnable nominal width parameter λ from Equation 2 of the main paper.

Effect of Nominal Width: For assessing the behaviour of the nominal width in all experiments we set it as a learnable parameter initialized to the value of 1m. Over the course of training, this

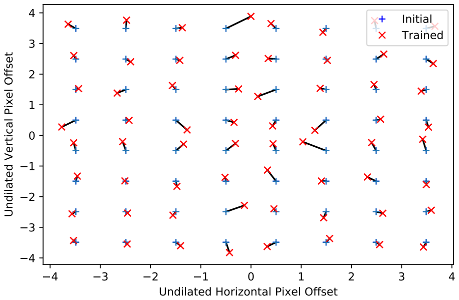


Figure 6: Movement of a 8 × 8 kernel sampling pattern over the course of training. With blue '+' showing the uniform grid used to initialize offsets, and red '×' showing the pattern after training.

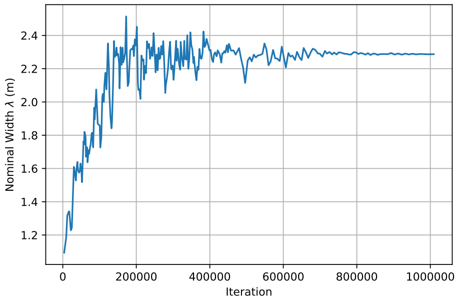


Figure 7: Nominal width free parameter λ over the course of training for the input RCD block. λ is used to model the nominal width (in meters) of an object at any range.

Method	Dilation	AP	APH	R@50P
Standard Convolution	Fixed rate=1	52.25	50.28	59.22
	Fixed rate=3	53.75	51.76	60.05
	Fixed rate=5	53.61	51.3	59.93
	Fixed rate=7	51.58	48.06	60.23
ASPP* [31]	(12, 24, 36, G)	45.8	45.2	-
Ours	Range conditioned	54.87	52.47	**61.03**
Ours with SRG	Range conditioned	**55.01**	**52.89**	60.83

Table 5: Performance comparison for first stage RPN with a single RCD or standard convolution applied to the input tensor with fixed dilation rates. Average precision (AP), average precision weighted by heading (APH) and Recall at 50% precision (R@50P) is measured on WOD validation set after 350k iterations of training. (*) ASPP - with 16 filter channels per dilation layer - peaked near 240K iterations with values shown in table. RCD improves over using only standard convolutional layers, and adding soft range gating (SRG), improves AP and APH even further.

parameter generally increases and then settles between 2 and 3m which is comparable to the average dimensions of a vehicle.

Effect of SRG: We found that the addition of a range gate improved training stability leading to faster convergence with a slightly improved performance. Table 5 shows the benefits provided to the first stage detector compared to adding a standard square kernel with different dilation rates. For a fair comparison we substitute the RCD block with a standard 2D convolutional layer with 7 × 7 kernel and 64 channels matching the RCD output.

C Comparison to Fixed Dilation Rates

Table 5, expands the experiments in the main paper with a more exhaustive set of dilation rates. Here we explore the effect of increasing the dilation rate of the kernel for a standard convolutional operation [31, 33] in the first region proposal network (RPN). We also compare to the atrous spatial pyramid pooling (ASPP) [31] framework by replacing our RCD layer with the ASPP module. Here we used fixed strides of 12, 24, 36 and global average features denoted by 'G'[3].

With dilation rates of 3 and 5, a small improvement over the standard convolution (with default dilation rate of 1) is observed. However, further increasing the dilation results in a sharp drop in performance as seen with the dilation rate of 7. This drop in performance could be due to: insufficient sampling of small distant objects; or an increase in the amount of padding needed given

[3] Following implementation of [50] from: github.com/rishizek/tensorflow-deeplab-v3

# Channels	Dilations	RPN-AP	RPN-APH	RCNN-AP	RCNN-APH	Peak@iteration
64	(2, 4, 8, G)	28.6	28.0	59.2	58.7	83K
64	(6, 12, 18, G)	27.1	26.4	58.4	58.0	62K
64	(12, 24, 36, G)	30.4	29.6	58.7	58.2	60K
16	(12, 24, 36, G)	45.8	45.2	64.4	63.9	240K
8	(12, 24, 36, G)	27.5	27.0	59.4	59.0	76K

Table 6: Parameter search for the ASPP [31] baseline where all methods were trained for 350K iterations. Peak@iteration shows the checkpoint which the highest second stage RCNN performance on WOD validation. The low and sporadic RPN performance and early peak iteration suggests that ASPP easily overfits and generally results in unstable training.

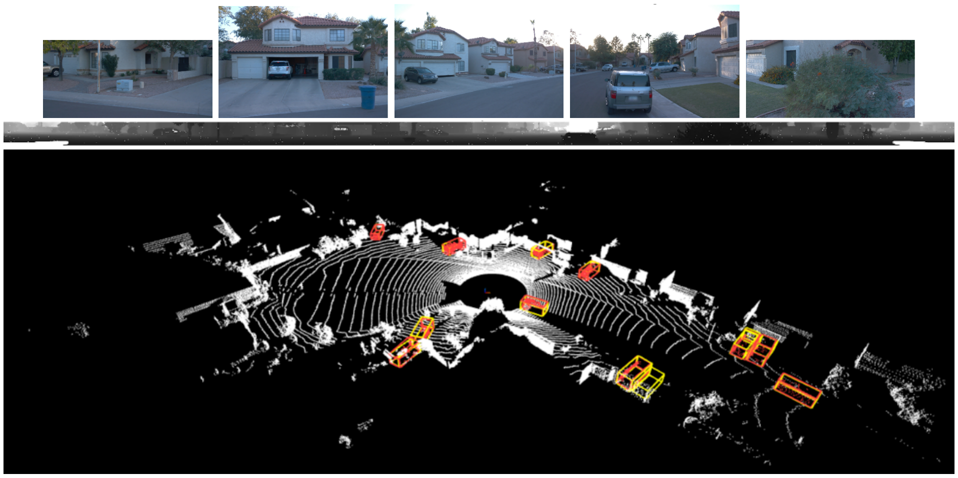


Figure 8: Example visualization of 3D alignment. Top row shows colour images of Left, Front-Left, Front, Front-Right and Right view respectively. Middle row shows the corresponding range image for the scene. Bottom row shows the corresponding 3D pointcloud with our detections in red against the ground truth boxes in yellow. *Note the RGB images are only for illustration purposes.*

the height of the range image is limited to 64 pixels. ASPP with various dilations significantly under-performed compared to the standard dilated convolutions. Having fixed and overly large dilation rates increases the susceptibility of ASPP to be distracted by sporadic activations caused by high frequency detail. We tried to vary the receptive field and the number of channels (parameters) to prevent overfitting but found the training to be generally highly unstable. See Table 6. The RCD models, with their ability to appropriately adjust their rate of dilation for both near and far, leads to the best overall performance.

D Qualitative Results

Figures 8 and 10 provide a qualitative view of our detections on WOD validation sequences. In top left of the third row of Figure 10 the RPN conservatively predicts a proposal in the occlusion shadow of a vehicle which is later correctly removed by the second stage RCNN refinement. A failure case is shown in the first row, where the RCNN mistakenly classifies a proposal as background. More detailed visualizations are provided in the accompanying video supplementary.

E Comparison to LaserNet on KITTI

While the KITTI dataset is a common benchmark for 3D detection, it has been shown that its small size can lead to overfitting and poor generalization for range image based methods [11]. While we also observe similar issues, the KITTI dataset is used as an indication of how our method compares to the official results of LaserNet in the small dataset regime.

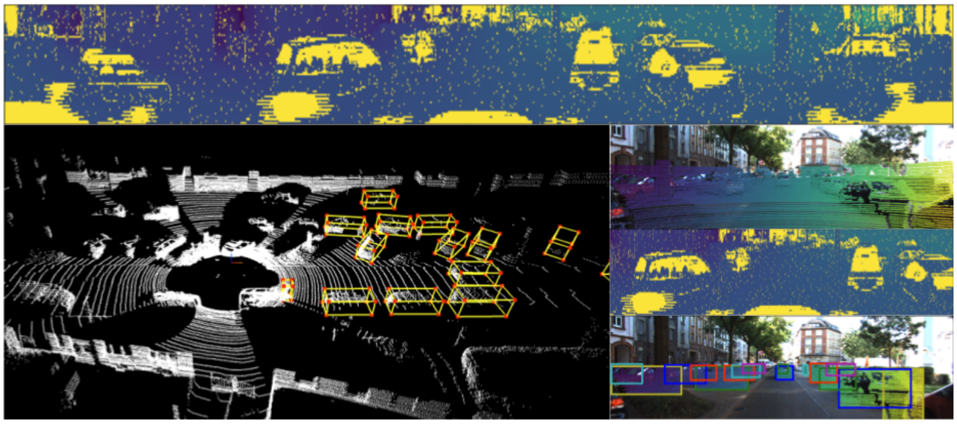


Figure 9: Illustration of extracted range image (**top**) from KITTI point cloud (**left**), with projection of the point cloud onto the RGB image (**middle right**), a detail of the extracted range image, and (**bottom right**) an illustration of the visible points per annotated bounding box.

E.1 Range Images from Point Clouds

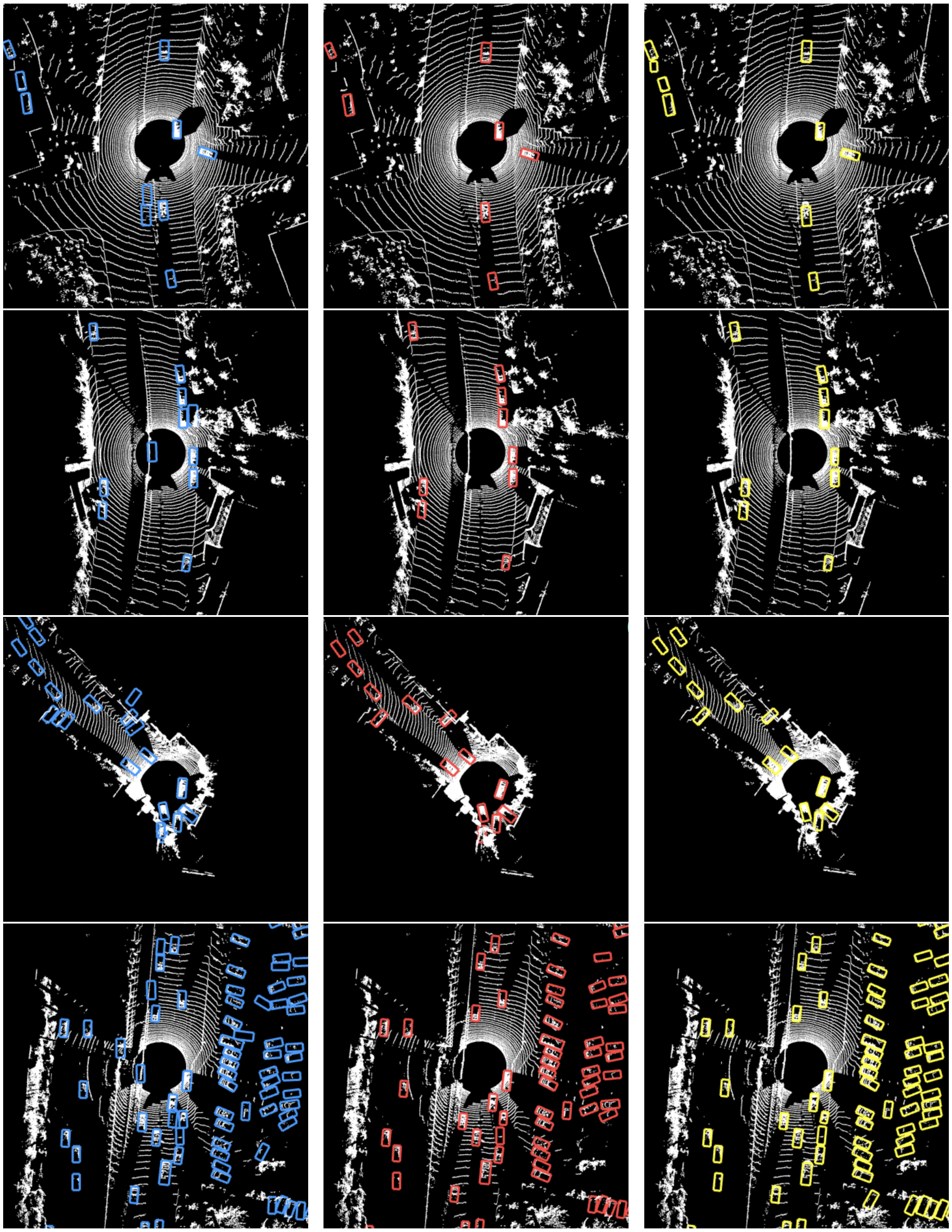
Despite the range-image being a native representation for LiDAR, KITTI distributes data from the Velodyne LiDAR sensor as a Cartesian pointcloud with an additional intensity channel. The straightforward conversion of Cartesian to spherical points results in poor quality range-images due to the 64 lasers of the Velodyne not sharing a common origin. To overcome this issue, dense and compact range images are reconstructed as follows:

1. For each of the lasers in the Velodyne sensor its inclination and height are determined using Hough voting, in a discretized height and inclination space.
2. For each point in a point cloud its corresponding laser-id l and azimuth angle a is estimated by minimizing the reconstruction / quantization error.
3. The range image is then constructed as a $L \times A$ image, where L denotes the number of lasers (64) and A is the number of azimuth steps (set to 2048). The pixel value at location (l, a) reflects the observed range, *i.e.* $r = \sqrt{x^2 + y^2 + z^2}$, for point $p = (x, y, z)$). The addition of the laser intensity is similarly added as another channel. When multiple observations map to the same pixel location (l, a) in the range image, the closest point is kept.

An illustration of the compact reconstructed range-image from KITTI Velodyne data is shown in Figure 9.

E.2 Results

We report performance on KITTI test set after a fixed number of 100K iterations, using flipping and horizontal pixel shifting as data augmentation. Since the KITTI data is only labelled in the 90° around the forward driving direction, only that part of the range image is fed into the network. Table 1 shows the performance of our method against the primary representative for range-image detectors LaserNet [11] on this small dataset. This table is extended with a RCD model pretrained on WOD and finetuned on KITTI for 100K iterations, denoted as RCD-FT. While these numbers are not intended to be directly compared to other published works, they serve to high-light the importance of large and diverse datasets for training range-image detectors.



(a) Proposals (b) Detections (c) Ground Truth

Figure 10: Birds-eye-view visualization of four different scenes from the WOD validation sequences. Each row shows a different scene and each column from left to right shows: output of the RPN in blue, output of RCNN in red, and ground truth boxes in yellow respectively. All predictions are filtered with minimum confidence of 0.5. Best viewed digitally with zoom.