Triangulation Learning Network: from Monocular to Stereo 3D Object Detection

Zengyi Qin[*]
Tsinghua University
v-zeqin@microsoft.com

Jinglu Wang
Microsoft Research
jinglwa@microsoft.com

Yan Lu
Microsoft Research
yanlu@microsoft.com

Abstract

In this paper, we study the problem of 3D object detection from stereo images, in which the key challenge is how to effectively utilize stereo information. Different from previous methods using pixel-level depth maps, we propose employing 3D anchors to explicitly construct object-level correspondences between the regions of interest in stereo images, from which the deep neural network learns to detect and triangulate the targeted object in 3D space. We also introduce a cost-efficient channel reweighting strategy that enhances representational features and weakens noisy signals to facilitate the learning process. All of these are flexibly integrated into a solid baseline detector that uses monocular images. We demonstrate that both the monocular baseline and the stereo triangulation learning network outperform the prior state-of-the-arts in 3D object detection and localization on the challenging KITTI dataset.

1. Introduction

3D object detection aims at localizing amodal 3D bounding boxes of objects of specific classes in the 3D space [1, 32, 18, 29]. The detection task is relatively easier [7, 19, 15] when active 3D scan data are provided. However, the active scan data are of high cost and limited in scalability. We address the problem of 3D detection from passive imagery data, which require only low-cost hardware, adapt to different scales of objects, and offer fruitful semantic features.

Monocular 3D detection with a single RGB image is highly ill-posed because of the ambiguous mapping from 2D images to 3D geometries, but still attracts research efforts [2, 30] because of its simplicity of design. Adding more input views can provide more information for 3D reasoning, which is ubiquitously demonstrated in the traditional multi-view geometry [10] area by finding dense patch-level correspondences for points and then estimate their 3D locations by triangulation. The geometric methods deal with patch-level points with local features, while
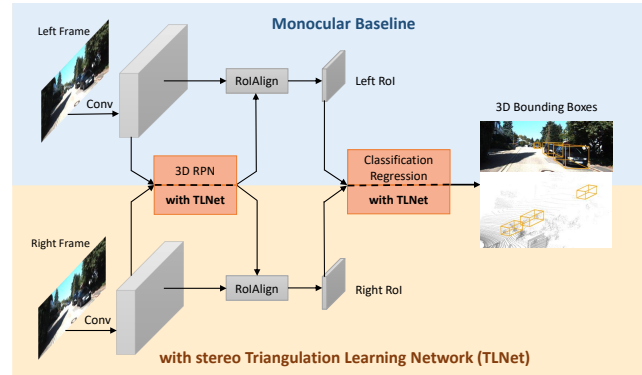

Figure 1. **Overview of the proposed 3D detection pipeline.** The baseline monocular network is indicated with blue background, and can be easily extended to stereo inputs by duplicating the baseline and further integrating with the proposed TLNet.

no semantic clues at object-level are considered.

Stereo data with pairs of images are more suitable for 3D detection, since the disparities between left and right images can reveal spacial variance, especially in the depth dimension. While extensive work [31, 24, 5] has been done on deep-learning based stereo matching, their main focus is on the pixel level rather than object level. 3DOP [3] uses stereo images for 3D object detection and achieves state-of-the-art results. Nevertheless, later we show that we can obtain comparable results using *only* a monocular image, by properly placing 3D anchors and extending the region-proposal network (RPN) [28] to 3D. Therefore, we are motivated to reconsider how to better exploit the potential of stereo images for accurate 3D object detection.

In this paper, we propose the stereo **T**riangulation **L**earning **N**etwork (TLNet) for 3D object detection from stereo images, which is free of computing pixel-level depth maps and can be easily integrated into the baseline monocular detector. The key idea is to use a 3D anchor box to explicitly construct object-level geometric correspondences of its two projections on a pair of stereo images, from which the network learns to triangulate a targeted object near the anchor. In the proposed TLNet, we introduce an

---
[*]Work done as an intern at Microsoft Research.

efficient feature reweighting strategy that strengthens informative feature channels by measuring left-right coherence. The reweighting scheme filters out the signals from noisy and mismatched channels to facilitate the learning process, enabling our network to focus more on the key parts of an object. Without any parameters, the reweighting strategy imposes little on computational burden.

To examine our design, we first propose a solid baseline monocular 3D detector, with an overview shown in Fig. 1. In combination with TLNet, we demonstrate that significant improvement can be achieved in 3D object detection and localization in various scenarios. Additionally, we provide quantitative analysis of the feature reweighting strategy in TLNet to have a better understanding of its effects. In summary, our contributions are three-fold:

- A solid baseline 3D detector that takes only a monocular image as input, which has comparable performance with its state-of-the-art stereo counterpart.

- A triangulation learning network that leverages the geometric correlations of stereo images to localize targeted 3D objects, which outperforms the baseline model by a significant margin on the challenging KITTI [8] dataset.

- A feature reweighting strategy that enhances informative channels of view-specific RoI features, which benefits triangulation learning by biasing the network attention towards the key parts of an object.

2. Related Work

Monocular 3D Detection. Due to the information loss in the depth dimension, 3D object detection is in particular difficult given only a monocular image. Mono3D [2] integrates semantic segmentation and context priors to generate 3D proposals. Extra computation is needed for semantic and instance segmentation, which slows down its running time. Xu et al.[30] utilizes a stand-alone disparity estimation model [22] to generate 3D point clouds from a monocular image, and then perform 3D detection using multi-level concatenated RoI features from RGB and the point cloud maps. However, it has to use extra data, i.e., ground truth disparities, to train the stand-alone model. Other methods [14] leverage 3D CAD models to synthesize 3D object templates to supervise the training and guide the geometric reasoning in inference. While the previous methods are dependent on additional data for 3D perception, the proposed monocular baseline model can be trained using only ground truth 3D bounding boxes, which saves considerations on data acquisition.

Multi-View based 3D Detection. MV3D [4] takes RGB images and multi-view projections of LIDAR 3D points as input, and then fuse the RoI pooled features for 3D bounding box prediction. By projecting the point clouds to bird's-eye-view (BEV) maps, it first generates 2D bounding box proposals in BEV, then fixes the height to obtain 3D proposals. RoIAlign [11] is performed in front-view and BEV point cloud maps and also in image to extract multi-view features, which are used to predict object classes and regress the final 3D bounding box. AVOD [17] aggregates features from front view and BEV of LIDAR point to jointly generate proposals and detect objects. It is an early-fusion method, where the multi-view features are fused before the region-proposal stage [11] to increase the recall rate. The most related approach to ours is 3DOP [3] that uses stereo images for object detection. However, 3DOP [3] directly relies on the disparity maps calculated from image pairs, resulting in its high computational cost and the imprecise estimation at distant regions. Our network is free of calculating pixel-level disparity maps. Instead, it learns to triangulate the target from left-right RoIs.

Learning based Stereo. Zbontar and LeCun [31] propose a stereo matching network to learning a similarity measure on small image patches to estimate the disparity at each location of the input image. Because comparing the image patches for each disparity would increase the computational cost combinationally, the authors propose to propagate the full-resolution images to compute the matching cost for all pixels in only one forward pass. DispNet [24] concatenates the stereo image pairs as input to learn the disparity and scene flow. Chen et al.[5] design a convolutional spatial propagation network to estimate the affinity matrix for depth estimation. While the existing studies mainly focus on pixel-level learning, we primarily explore the instance-level learning for 3D object detection using the intrinsic geometric correspondence of the stereo images.

Triangulation. Triangulation localizes a point by forming triangles to it from known points, which is of universal applications in 3D geometry estimation [6, 26, 12, 25]. In RGB-D based SLAM [6], triangulation can be utilized to create a sparse graph modeling the correlations of points and retrieve accurate navigation information in complex environment. It is also useful in camera calibration and motion tracking [12]. In stereo vision, typical triangulation [31] requires matching the 2D features in left and right frames at pixel level to estimate spacial dimensions, which can be computationally expensive and time consuming [24]. To avoid such computation and fully exploit stereo information for 3D object detection, we propose using 3D anchors as reference to detect and localize the object via learnable triangulation in a forward pass.

(a) Input image   (b) 2D objectness confidence map
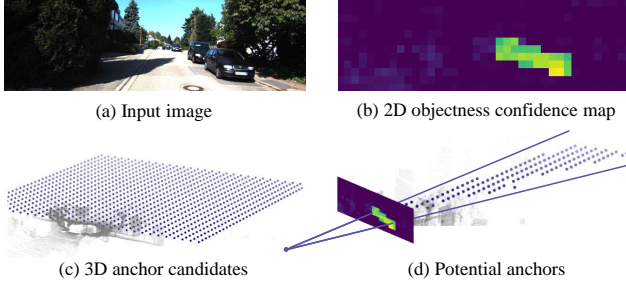
(c) 3D anchor candidates   (d) Potential anchors

Figure 2. **Front view anchor generation.** Potential anchors are of high objectness in the front view. Only the potential anchors are fed into RPN to reduce searching space and save computational cost.
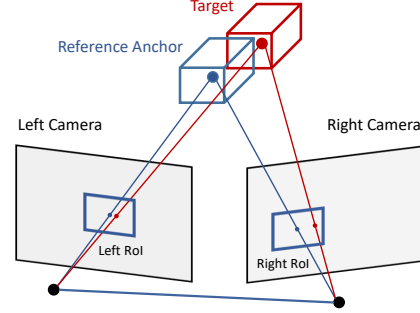

Figure 3. **Anchor triangulation.** By projecting the 3D anchor box to stereo images, we obtain a pair of RoIs. The left RoI establishes a geometric correspondence with the right one via the anchor box. The nearby target is present in both RoIs with slightly positional differences. Our TLNet takes the RoI pair as input and utilizes the 3D anchor as reference to localize the targeted object.

3. Approach

In this section, we first present the baseline model that predicts oriented 3D bounding boxes from a base image I, and then introduce the triangulation learning network integrated in the baseline for a pair of stereo image (I^l, I^r). An oriented 3D bounding box is defined as $B = (\mathbf{C}, \mathbf{S}, \theta)$, where $\mathbf{C} = (c_x, c_y, c_z)$ is the 3D center, $\mathbf{S} = (h, w, l)$ is the size along axis, and θ is the angle between itself and the vertical axis.

3.1. Baseline Monocular Network

The baseline network taking a monocular image as the input is composed of a backbone and three subsequent modules, i.e., the front view anchor generation, the 3D box proposal and refinement. The three-stage pipeline progressively reduces the searching space by selecting confident anchors, which highly reduces computational complexity.

3.1.1 Front View Anchor Generation

Following the dimension reduction principle, we first reduce the searching space in the 2D front view. The input image I is divided into a $G_x \times G_y$ grid, where each cell predicts its objectness. The output objectness represents the confidence how likely the cell is surrounded by 2D projections of targeted objects. An example of the confidence map is shown in Fig. 2. In training, we first calculate 2D projected centers of the ground truth 3D boxes and compute their least Euclidean distance to all cells in the $G_x \times G_y$ grid. Cells with distances less than 1.8 times their width are considered as foreground. For inference, foreground cells are selected to generate potential anchors.

Thus, we obtain 3D anchors located at the rays issued from the potential cells in an anchor pool, which contains a set of 3D anchors uniformly sampled at an interval of 0.25m on the ground plane within the view frustum and depth ranging $[0, 70]$ meters. Anchors are represented by its 3D center and the prior size along three axes. There are two anchors with BEV orientation $0°$ and $90°$ for each object class at

the same location. For different classes, we calculate the prior size by averaging corresponding samples in the training split. An example of generated anchors are shown in Fig. 2.

3.1.2 3D Box Proposal and Refinement

The multi-stage proposal and refinement mechanism in our baseline network is similar to Faster-RCNN [28]. In the 3D RPN, the potential anchors from front view generation are projected to the image plane to obtain RoIs. RoIAlign [11] is adopted to crop and resize the RoI features from the feature maps. Each crop of RoI features is fed to task specific fully connected layers to predict 3D objectness confidence as well as regress the location offsets $\Delta \mathbf{C} = (\Delta c_x, \Delta c_y, \Delta c_z)$ and dimension offsets $\Delta \mathbf{S} = (\Delta h, \Delta w, \Delta l)$ to the ground truth. Non-Maximum Supression (NMS) is adopted to keep the top K proposals, where $K = 1024$ for both training and inference. In the refinement stage, we project those top proposals to image and again use RoIAlign [11] to crop and resize the region of interest. The RoI features are passed to fully connected layers that classify the object and regress the 3D bounding box offsets $(\Delta \mathbf{C}, \Delta \mathbf{S})$ as well as the orientation vector $\mathbf{v}_\theta$ in local coordinate system as defined in [17].

3.2. Triangulation Learning Network

The stereo 3D detection is performed by integrating a triangulation learning network into the baseline model. In the following, we first introduce the mechanism of the TL-Net that focuses on object-level triangulation in opposite to computationally expensive pixel-level disparity estimation, and then present the details of its network architecture.

$i = 24, s_i = 0.898$ $i = 29, s_i = 0.851$ $i = 26, s_i = 0.846$ $i = 18, s_i = 0.742$ $i = 14, s_i = 0.706$ $i = 22, s_i = 0.589$ $i = 17, s_i = 0.429$

Left RoI

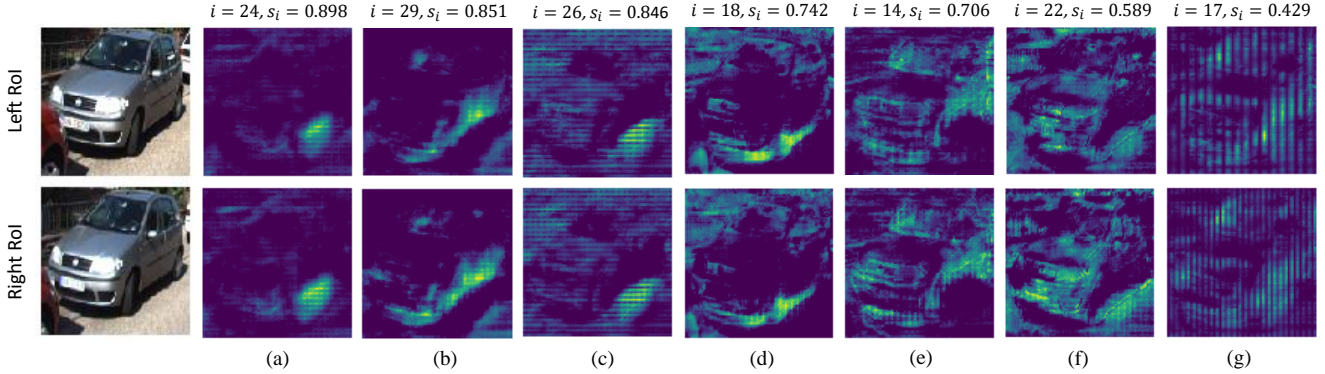Right RoI

(a) (b) (c) (d) (e) (f) (g)

Figure 4. **Activations of different channels with different coherence scores.** These small feature maps are cropped out using RoIAlign from the last convolutional layer, where the first row is from the left branch and the second row is from the right. Two branches of convolutional layers share the weights. Coherence score s_i is calculated for channel i. As we can see, channel 17 (g) and 22 (f) have noisy and less concentrated activations, while chanel 24 (a) is clear and informative of the key points of an object, e.g., wheels. Our objective is enhancing channels like (a) and weaken those like (g) and (f), so as to focus the network attention on specific parts of the object, which is empirically beneficial for discerning slight positional difference between where the object is present in left and right RoIs.
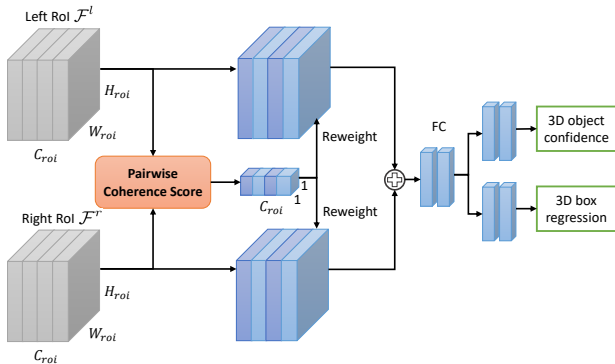

Figure 5. **TLNet architecture.** The inputs are a pair of RoIs obtained by projecting a 3D anchor box to the left and right feature maps with C_{roi} channels. The coherence score s_i is computed between each left channel $\mathcal{F}_i^l$ and right channel $\mathcal{F}_i^r$. We *reweight* the ith channel by multiplying with s_i. The features are fused and fed to fully-connected layers to predict objectness confidence and 3D bounding box offsets to the reference anchor.

3.2.1 Anchor Triangulation

Triangulation is known as localizing 3D points from multi-view images in the classical geometry fields, while our objective is to localize a 3D object and estimates its size and orientation from stereo images. To achieve this, we introduce an *anchor triangulation* scheme, in which the neural network uses 3D anchors as reference to triangulate the targets.

Considering the RPN stage, we project the pre-defined 3D anchor to stereo images and obtain a pair of left-right RoIs, as illustrated in Fig. 3. If the anchor tightly fits the target in 3D, its left and right projections can consistently bound the object in 2D. On the contrary, when the anchor fails in fitting the the object, their geometric differences in

3D are reflected in the visual disparity of the left-right RoI pair. The 3D anchor explicitly constructs correspondences between its projections in multiple views. Since the location and size of the anchor box are already known, modeling the anchor triangulation is conducive to estimating the 3D objectness confidence, i.e., how well the anchor matches a target in 3D, as well as regressing the offsets applied to the box to minimize its variance with the target. Therefore, we propose TLNet towards triangulation learning as follows.

3.2.2 TLNet Architecture

The TLNet takes as input a pair of left-right RoI features $\mathcal{F}^l$ and $\mathcal{F}^r$ with C_{roi} channels and size $H_{roi} \times W_{roi}$, which are obtained using RoIAlign [11] by projecting the same 3D anchor to the left and right frames, as shown in Fig. 5. We utilize the left-right *coherence scores* to *reweight* each channel. The reweighted features are fused using element-wise addition and passed to task-specific fully-connected layers to predict the objectness confidence and 3D bounding box offsets, i.e., the 3D geometric variance between the anchor and target.

Coherence Score. The positional difference between where the target is present in the left and right RoIs reveals the spacial variance between the target box and the anchor box, as illustrated in Fig. 3. The TLNet is expected to utilize such a difference to predict the relative location of the target to the anchor, estimate whether they are a good match, i.e., objectiveness confidence, and regress the 3D bounding box offsets. To enhance the discriminative power of spacial variance, it is necessary to focus on representational key points of an object. Fig. 4 shows that, though without explicit supervision, some channels in the feature maps has learned to

extract such key points, e.g., wheels. The coherence score s_i for the ith channel is defined as:

$$s_i = cos < \mathcal{F}_i^l, \mathcal{F}_i^r >= \frac{\mathcal{F}_i^l \cdot \mathcal{F}_i^r}{||\mathcal{F}_i^l|| \cdot ||\mathcal{F}_i^r||} \qquad (1)$$

where cos is the cosine similarity function for each channel, $\mathcal{F}_i^l$ and $\mathcal{F}_i^r$ are the ith pair of feature maps, i.e., features from the ith channel in left and right RoIs.

From Fig. 4, we observe that the coherence score s_i is lower when the activations are noisy and mismatched, while it is higher if the left-right activations are clear and coherent. In fact, from a mathematical perspective, $\mathcal{F}_i^l$ and $\mathcal{F}_i^r$ can be viewed as a pair of signal vectors. As you flatten along the row dimension, consecutive activations are more likely to align with other consecutive activations in a pair of RoIs, i.e., s_i is closer to 1.

Channel Reweighting. By multiplying the ith channel with s_i, we weaken the signals from noisy channels and bias the attention of the following fully-connected layers towards coherent feature representations. The reweighting is done in pairs for both the left and right feature maps, taking the form:

$$\mathcal{F}_i^{l,re} = s_i \mathcal{F}_i^l, \quad \mathcal{F}_i^{r,re} = s_i \mathcal{F}_i^r \qquad (2)$$

, and it is implemented in the TLNet as illustrated in Fig. 5. We will demonstrate that the reweighting strategy has a positive effect on triangulation learning in the experiments.

Network Generality. The proposed architecture can be easily integrated into the baseline network by replacing the fully-connected layers after RoIAlign, in both the RPN and the refinement stage. In the refinement stage, classification outputs are object classes instead of objectness confidence as is in RPN. For the CNN backbone, the left and right branches share their parameters. Computing cosine similarity and reweighting do not require any extra parameters, thus imposing little in memory burden. SENet [13] also involves reweighting feature channels. Their goal is to model the cross-channel relationships of a single input, while ours is to measure the coherence of a pair of stereo inputs and select confident channels for triangulation learning. In addition, their weights are learned by fully-connected layers, whose behavior is less interpretable. Our weights are the pairwise cosine similarities with clear physical significance. $\mathcal{F}_i^l$ and $\mathcal{F}_i^r$ can be viewed as two vectors, and s_i describes their included angle, i.e., correlation.

3.3. Implementation Details

Network Setup. We choose VGG-16 [23] as the CNN backbone, but without its fully-connected layers after $pool5$. Parameters are shared in the left and right branches.

All the input images are resized to 384×1248 so that they can be divided by 2 for at least 5 times. For the front-view objectness prediction, we use the left branch only. We apply 1×1 convolution to the output feature maps of $pool5$ and reduce the channels to 2, indicating the background and objectness confidence. Each pixel in the output feature maps represents a grid cell and yields a prediction.

For the region proposal and refinement stage, we start from the output of $conv4$ rather than $pool5$. To improve the performance on small objects, we leverage Feature Pyramid [21] to upsample the feature maps to the original resolution. Since the region proposal stage aims at filtering background anchors and keep the confident anchors in an efficient manner, channels of the feature maps are reduced to 4 using 1×1 convolution before RoIAlign [11] and fully connected layers to save computational cost. For the refinement stage, however, we use full channels containing more information to yield finer predictions.

Training. All the weights are initialized by Xavier initializer [9] and no pretrained weights are used. L2 regularization is applied to the model parameters with a decay rate of 5e-3. We first train the front-view objectness map for 20K iterations, then along with the RPN for another 40K iterations. In the next 60K iterations we add the refinement stage. For the above 120K iterations, the network is trained using Adam optimizer [16] at a learning rate of 1e-4. Finally, we use SGD to further optimize the network for 20K iterations at the same learning rate. The batchsize is always set to 1 for the input. The network is trained using a single GPU of NVidia Tesla P40.

4. Experiment

We evaluate the proposed network on the challenging KITTI dataset [8], which contains 7481 training images and 7518 testing images with calibrated camera parameters. Detection is evaluated in three regimes: easy, moderate and hard, according to the occlusion and truncation levels. We use the train1/val1 split setup in the previous works [2, 4], where each set contains half of the images. Objects of class Car are chosen for evaluation. Because the number of cars exceeds that of other objects by a significant margin, they are more suitable to assess a data-hungry deep-learning network. We compare our baseline network with state-of-the-art monocular 3D detectors, MF3D [30], Mono3D [2] and MonoGRNet [27]. For stereo input, we compare our network with 3DOP [3].

Metrics. For 3D detection, we follow the official settings of KITTI benchmark to evaluate the 3D Average Precision (AP_{3D}) at different 3D IoU thresholds. To evaluate the bird's eye view (BEV) detection performance, we use the

Method	Data	AP$_{3D}$(IoU=0.3)			AP$_{3D}$(IoU=0.5)			AP$_{3D}$(IoU=0.7)		
		Easy	Moderate	Hard	Easy	Moderate	Hard	Easy	Moderate	Hard
VeloFCN	LiDAR	/	/	/	67.92	57.57	52.56	15.20	13.66	15.98
Mono3D	Mono	28.29	23.21	19.49	25.19	18.20	15.22	2.53	2.31	2.31
MF3D	Mono	/	/	/	47.88	29.48	26.44	10.53	5.69	5.39
MonoGRNet	Mono	72.17	59.57	46.08	50.51	36.97	30.82	13.88	10.19	7.62
3DOP	Stereo	69.79	52.22	49.64	46.04	34.63	30.09	6.55	5.07	4.10
Ours (baseline)	Mono	72.91	55.72	49.19	48.34	33.98	28.67	13.77	9.72	9.29
Ours	Stereo	**78.26**	**63.36**	**57.10**	**59.51**	**43.71**	**37.99**	**18.15**	**14.26**	**13.72**

Table 1. **3D detection performance.** Average Precision of 3D bounding boxes on KITTI [8] validation set. The LiDAR based method VeloFCN [20] is listed for reference but not compared.

Method	Data	AP$_{BEV}$(IoU=0.3)			AP$_{BEV}$(IoU=0.5)			AP$_{BEV}$(IoU=0.7)		
		Easy	Moderate	Hard	Easy	Moderate	Hard	Easy	Moderate	Hard
VeloFCN	LiDAR	/	/	/	79.68	63.82	62.80	40.14	32.08	30.47
Mono3D	Mono	32.76	25.15	23.65	30.50	22.39	19.16	5.22	5.19	4.13
MF3D	Mono	/	/	/	55.02	36.73	31.27	22.03	13.63	11.60
MonoGRNet	Mono	73.10	60.66	46.86	54.21	39.69	33.06	24.97	19.44	16.30
3DOP	Stereo	71.41	57.78	51.91	55.04	41.25	34.55	12.63	9.49	7.59
Ours (baseline)	Mono	74.18	57.04	50.17	52.72	37.22	32.16	21.91	15.72	14.32
Ours	Stereo	**81.11**	**65.25**	**58.15**	**62.46**	**45.99**	**41.92**	**29.22**	**21.88**	**18.83**

Table 2. **BEV detection performance.** Average Precision of BEV bounding boxes on KITTI [8] validation set. Different from typical 2D detection evaluation, the bounding boxes are orientated, i.e., not necessarily aligned on each axis.

Method	Data	AP$_{LOC}$(<2m)			AP$_{LOC}$(<1m)			AP$_{LOC}$(<0.5m)		
		Easy	Moderate	Hard	Easy	Moderate	Hard	Easy	Moderate	Hard
Mono3D	Mono	47.21	35.82	35.40	17.74	14.80	14.05	4.78	4.06	4.03
MonoGRNet	Mono	83.09	64.82	55.79	56.29	42.29	35.01	28.38	19.73	18.06
3DOP	Stereo	84.53	65.37	64.10	60.84	45.11	40.35	27.53	19.39	17.64
Ours (baseline)	Mono	75.50	59.33	52.92	58.11	41.05	36.59	30.82	21.25	18.03
Ours	Stereo	**86.78**	**72.17**	**66.27**	**69.15**	**51.68**	**47.68**	**37.36**	**27.42**	**24.56**

Table 3. **3D localization performance.** Average Location Precision of 3D bounding boxes under different distance thresholds on KITTI [8] validation set.

BEV Average Precision (AP$_{BEV}$). We also provide results for Average Location Precision (AP$_{LOC}$), in which a ground truth object is recalled if there is a predicted 3D location within certain distance threshold. Note that we obtain the detection results of Mono3D and 3DOP from the authors so as to evaluate them with different IoU thresholds. The evaluation of MF3D follows their original paper.

4.1. 3D Object Detection

Monocular Baseline. 3D detection results are shown in Table 1. Our baseline network outperforms monocular methods MF3D [30] and Mono3D [2] in 3D detection. For *moderate* scenarios, our AP$_{3D}$ exceeds MF3D [30] by 4.50% at IoU = 0.5 and 4.65% at IoU = 0.7. For *easy* scenarios the gap is smaller. Since the raw detection results of MF3D [30] are not publicly available, detailed quantitative comparison cannot be made. But it is possible that a part of the performance gap comes from the object proposal stage, where MF3D [30] proposes on the image plane, while we directly propose in 3D. In *moderate* and *hard* cases where objects are occluded and truncated, 2D proposals can have difficulties retrieving the 3D information of a target that is partly unseen. In addition, the wide margin

with Mono3D [2] reveals a better expressiveness of learnt features than handcrafted features.

TLNet. By combining with the proposed TLNet, our method outperforms the baseline network and 3DOP [3] across all 3D IoU thresholds in *easy*, *moderate* and *hard* scenarios. Under IoU thresholds of 0.3 and 0.5, stereo triangulation learning brings $\sim 10\%$ improvement in AP$_{3D}$. In comparison with 3DOP [3], our method achieves $\sim 10\%$ better AP$_{3D}$ across all regimes. Note that 3DOP [3] needs to estimate the pixel-level disparity maps from stereo images to calculate depths and generate proposals, which is erroneous at distant regions. Instead, the proposed method utilizes 3D anchors to construct geometric correspondence between left-right RoI pairs, then triangulates the targeted objects using coherent features. According to the curves in Fig. 8 (a), our main method has a higher precision than 3DOP [3] and a higher recall than the baseline.

4.2. BEV Object Detection

Monocular Baseline. Results are presented in Table 2. Compared with 3D detection, we still evaluate the same set of 3D bounding boxes, but the vertical axis is disregarded
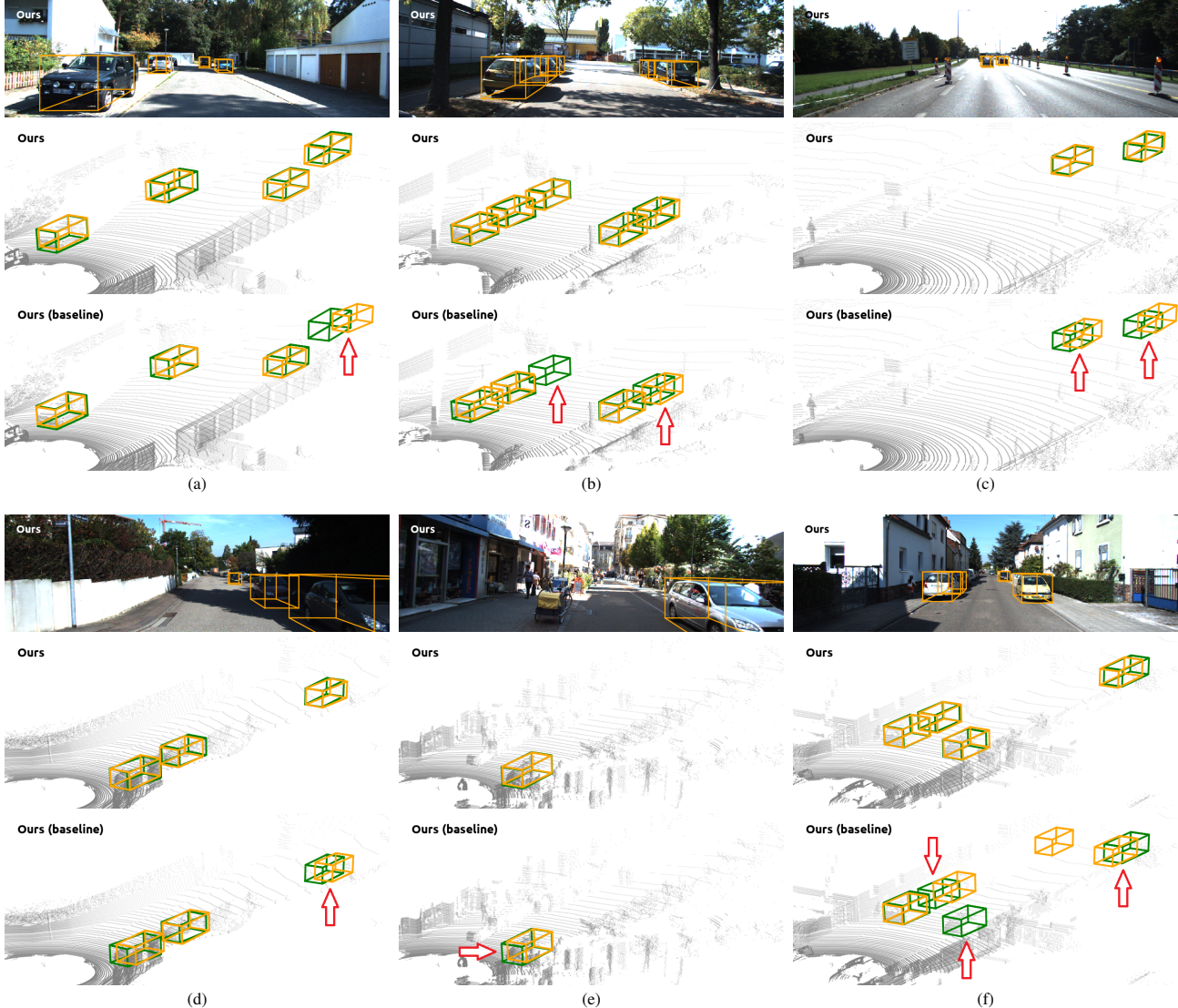
Figure 6. **Qualitative comparison.** Orange bounding boxes are detection results, while the green boxes are ground truths. For our main method, we also visualize the projected 3D bounding boxes in image, i.e., the first and forth rows. The lidar point clouds are visualized for reference but not used in both training and evaluation. It is shown that the triangulation learning method can reduce missed detections and improve the performance of depth prediction at distant regions.

for BEV IoU calculation. The baseline keeps its advantages over MF3D in *moderate* and *hard* scenarios. Note that MF3D uses extra data to train pixel-level depth maps. In *easy* cases, objects are clearly presented, and pixel-level predictions with sufficient local features can obtain more accurate results in statistics. However, pixel-level depth maps with high resolution are unfortunately not always available, indicating their limitations in real applications.

TLNet. Not surprisingly, by use of TLNet, we outperform the monocular baseline under all IoU thresholds in various scenarios. At IoU threshold 0.3 and 0.5, the triangulation learning yields $\sim 8\%$ increase in AP_{BEV}. Our method

also surpasses 3DOP by a notable margin, especially for strict evaluation criteria, e.g., under IoU threshold of 0.7, which reveals the high precision of our predictions. Such performance improvement mainly comes from two aspects: 1) the solid monocular baseline already achieves comparable results with 3DOP; 2) the stereo triangulation learning scheme further enhances the capability of our baseline model in object localization. Fig. 8 (b) further compares the Recall-Precision curves under IoU threshold of 0.3.

4.3. 3D Localization

Monocular Baseline. Results are shown in Table 3. Our monocular baseline achieve better results than 3DOP [3] un-

IoU	Method	AP$_{3D}$		
		Easy	Moderate	Hard
	Concat.	76.87	60.81	54.70
0.3	Add	75.74	59.89	53.57
	Reweight	**78.26**	**63.36**	**57.10**
	Concat.	56.32	41.20	36.41
0.5	Add	53.41	41.61	36.37
	Reweight	**59.51**	**43.71**	**37.99**
	Concat.	13.97	11.63	9.67
0.7	Add	16.60	13.59	11.20
	Reweight	**18.15**	**14.26**	**13.72**

Table 4. **Effect of reweighting on AP$_{3D}$.** Three fusion methods in Fig. 9 are compared, including concatenation, direct addition and the proposed reweighting strategy.

Figure 7. **Qualitative results for persons.**

(a) 3D object detection at IoU threshold of 0.3.

(b) BEV object detection at IoU threshold of 0.3.

(c) 3D localization at distance threshold of 1m.
Figure 8. **Recall-precision curves.**

der the strict distance threshold of 0.5m, indicating the high precision of our top predictions.

TLNet. TLNet boosts the baseline performance marginally, especially for distance threshold of 2m and 1m, where there is $\sim 10\%$ gain in AP$_{LOC}$. According to the Recall-Precision curves in Fig. 8 (c), TLNet increases both the precision and recall. The maximum precision is close to 1.0 because most of the top predictions are correct.

4.4. Qualitative Results

3D bounding boxes predicted by the baseline network and our stereo method are presented in Fig. 6. In general, the predicted orange bounding boxes matches the green ground truths better when TLNet is integrated into the baseline model. As shown in (a) and (c), our method can reduce depth error, especially when the targets are far away from the camera. Object targets missed by the baseline in (b) and (f) are successfully detected. The heavily truncated car in the right-bottom of (d) is also detected, since the object proposals are in 3D, regardless of 2D truncation.

4.5. Ablation Study

In TLNet, the small feature maps obtained by use of RoIAlign [11] are not directly fused. In order to focus more attention on the key parts of an object and reduce noisy signals, we first calculate the pairwise cosine similarity cos_i as coherence score, then reweight corresponding channels by multiplying with s_i. See 3.2.2 and Fig. 5 for details. In the followings, we examine the effectiveness of reweighting by replacing it with other two fusion methods, i.e., direct concatenation and element-wise addition, as is shown in Fig. 9.
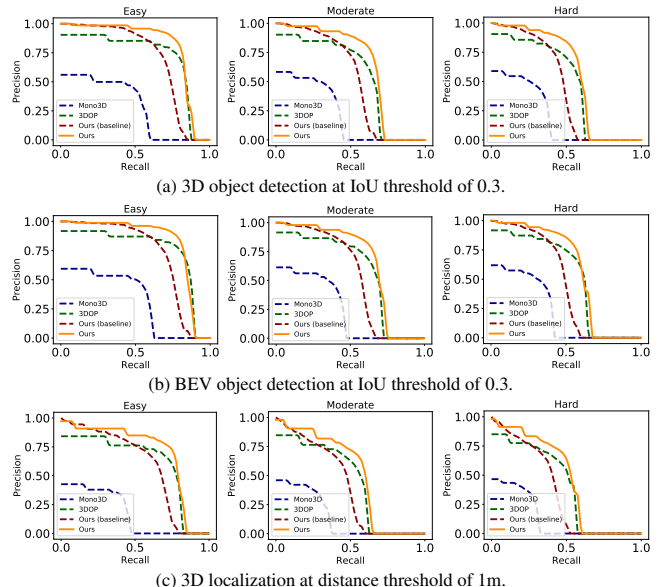
Comparisons between their AP$_{3D}$ and AP$_{BEV}$ are presented in Table 4. The evaluation corresponds with the empirical analysis in 3.2.2. Since the left and right branches share their weights in the backbone network, the same channels in left and right RoIs are expected to extract the same kind of features. Some of these features are more suitable for efficient triangulation learning, which are strengthened by reweighting. Note that the coherence score s_i is not fixed for each channel, but is dynamically determined by the RoI pairs cropped out at specific locations.
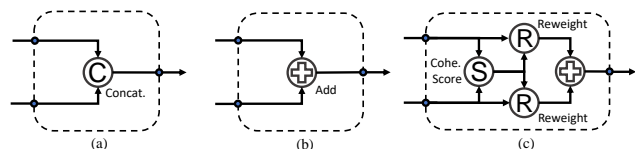

Figure 9. **Feature fusion methods in TLNet.** (c) is the proposed strategy. It first computes the coherence score of each pair of channels, then reweights the channels and adds them together.

5. Conclusion

We present a novel network for performing accurate 3D object detection using stereo information. We build a solid baseline monocular detector, which is flexibly extended to stereo by combining with the proposed TLNet. The key idea is to use 3D anchors to construct geometric correspondences between its projections in stereo images, from which the network learns to triangulate the targeted object in a forward pass. We also introduce an efficient channel reweighting method to strengthen informative features and weaken the noisy signals. All of these are integrated into our baseline detector and achieve state-of-the-art performance.

References

[1] H. Cai, T. Werner, and J. Matas. Fast detection of multiple textureless 3-d objects. In *Computer Vision Systems*, pages 103–112, 2013. 1

[2] X. Chen, K. Kundu, Z. Zhang, H. Ma, S. Fidler, and R. Urtasun. Monocular 3d object detection for autonomous driving. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2147–2156, 2016. 1, 2, 5, 6

[3] X. Chen, K. Kundu, Y. Zhu, A. G. Berneshawi, H. Ma, S. Fidler, and R. Urtasun. 3d object proposals for accurate object class detection. In *Advances in Neural Information Processing Systems*, pages 424–432, 2015. 1, 2, 5, 6, 7

[4] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia. Multi-view 3d object detection network for autonomous driving. In *IEEE CVPR*, volume 1, page 3, 2017. 2, 5

[5] X. Cheng, P. Wang, and R. Yang. Depth estimation via affinity learned with convolutional spatial propagation network. In *European Conference on Computer Vision*, pages 108–125, 2018. 1, 2

[6] W. Dai, Y. Zhang, P. Li, and Z. Fang. Rgb-d slam in dynamic environments using points correlations. *arXiv preprint arXiv:1811.03217*, 2018. 2

[7] 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 *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017. 1

[8] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Computer Vision and Pattern Recognition (CVPR)*, pages 3354–3361. IEEE, 2012. 2, 5, 6

[9] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Aistats*, 2010. 5

[10] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003. 1

[11] K. He, G. Gkioxari, P. Doll'ar, and R. Girshick. Mask r-cnn. *arXiv preprint arXiv:1703.06870*, 2017. 2, 3, 4, 5, 8

[12] C. D. Herrera, K. Kim, J. Kannala, K. Pulli, and J. Heikkil. Dt-slam: Deferred triangulation for robust slam. In *2014 2nd International Conference on 3D Vision*, pages 609–616, 2014. 2

[13] J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. 2018. 5

[14] W. Kehl, F. Manhardt, F. Tombari, S. Ilic, and N. Navab. Ssd-6d: Making rgb-based 3d detection and 6d pose estimation great again. In *Proceedings of the International Conference on Computer Vision (ICCV 2017)*, pages 22–29, 2017. 2

[15] J.-U. Kim and H.-B. Kang. A new 3d object pose detection method using lidar shape set. *Sensors*, 18(3):882, 2018. 1

[16] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *International Conference for Learning Representations*, 2014. 5

[17] J. Ku, M. Mozifian, J. Lee, A. Harakeh, and S. Waslander. Joint 3d proposal generation and object detection from view aggregation. *arXiv preprint arXiv:1712.02294*, 2017. 2, 3

[18] J. Lahoud and B. Ghanem. 2d-driven 3d object detection in rgb-d images. In *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017. 1

[19] B. Li. 3d fully convolutional network for vehicle detection in point cloud. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017. 1

[20] B. Li, T. Zhang, and T. Xia. Vehicle detection from 3d lidar using fully convolutional network. 2016. 6

[21] T.-Y. Lin, P. Doll'ar, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. *arXiv preprint arXiv:1612.03144v2*, 2017. 5

[22] R. Mahjourian, M. Wicke, and A. Angelova. Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5667–5675, 2018. 2

[23] D. Z. Matthew and F. Rob. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision*, pages 818–833, 2014. 5

[24] N. Mayer, E. Ilg, P. Hausser, and P. Fischer. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Computer Vision and Pattern Recognition(CVPR)*, pages 4040–4047, 2016. 1, 2

[25] T. Nir and A. M. Bruckstein. Causal camera motion estimation by condensation and robust statistics distance measures. In *European Conference on Computer Vision (ECCV)*, pages 119–131, 2004. 2

[26] E. Petroff, L. C. Oostrum, B. W. Stappers, M. Bailes, E. D. Barr, S. Bates, S. Bhandari, N. D. R. Bhat, M. Burgay, S. Burke-Spolaor, A. D. Cameron, D. J. Champion, R. P. Eatough, C. M. L. Flynn, A. Jameson, S. Johnston, E. F. Keane, M. J. Keith, L. Levin, V. Morello, C. Ng, A. Possenti, V. Ravi, W. van Straten, D. Thornton, and C. Tiburzi. A fast radio burst with a low dispersion measure. *arXiv preprint arXiv:1810.10773*, 2018. 2

[27] Z. Qin, J. Wang, and Y. Lu. Monogrnet: A geometric reasoning network for monocular 3d object localization. *AAAI*, 2019. 5

[28] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 2017. 1, 3

[29] Z. Ren and E. B. Sudderth. Three-dimensional object detection and layout prediction using clouds of oriented gradients. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1

[30] B. Xu and Z. Chen. Multi-level fusion based 3d object detection from monocular images. In *Computer Vision and Pattern Recognition (CVPR)*, pages 2345–2353, 2018. 1, 2, 5, 6

[31] J. Zbontar and Y. LeCun. Stereo matching by training a convolutional neural network to compare image patches. *Journal of Machine Learning Research*, 17(1-32):2, 2016. 1, 2

[32] A. Zeng, K.-T. Yu, S. Song, D. Suo, E. Walker, A. Rodriguez, and J. Xiao. Multi-view self-supervised deep learning for 6d pose estimation in the amazon picking challenge. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017. 1