arXiv:2104.05858v1 [cs.CV] 12 Apr 2021

Geometry-aware data augmentation for monocular 3D object detection

Qing Lian [1], Botao Ye[2], Ruijia Xu[1], Weilong Yao[3], Tong Zhang[1]
[1]The Hong Kong University of Science and Technology,
[2]Institute of Computing Technology, Chinese Academy of Sciences, China [3]AutoWise Inc
qlianab@connect.ust.hk, botao.ye@vipl.ict.ac.cn, rxuaq@connect.ust.hk,
yaoweilong@autowise.ai, tongzhang@ust.hk

Abstract

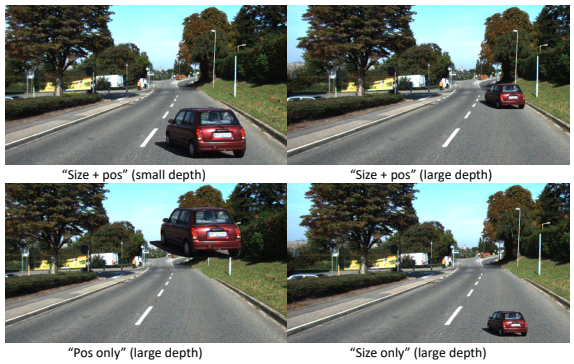
This paper focuses on monocular 3D object detection, one of the essential modules in autonomous driving systems. A key challenge is that the depth recovery problem is ill-posed in monocular data. In this work, we first conduct a thorough analysis to reveal how existing methods fail to robustly estimate depth when different geometry shifts occur. In particular, through a series of image-based and instance-based manipulations for current detectors, we illustrate existing detectors are vulnerable in capturing the consistent relationships between depth and both object apparent sizes and positions. To alleviate this issue and improve the robustness of detectors, we convert the aforementioned manipulations into four corresponding 3D-aware data augmentation techniques. At the image-level, we randomly manipulate the camera system, including its focal length, receptive field and location, to generate new training images with geometric shifts. At the instance level, we crop the foreground objects and randomly paste them to other scenes to generate new training instances. All the proposed augmentation techniques share the virtue that geometry relationships in objects are preserved while their geometry is manipulated. In light of the proposed data augmentation methods, not only the instability of depth recovery is effectively alleviated, but also the final 3D detection performance is significantly improved. This leads to superior improvements on the KITTI and nuScenes monocular 3D detection benchmarks with state-of-the-art results.

1. Introduction

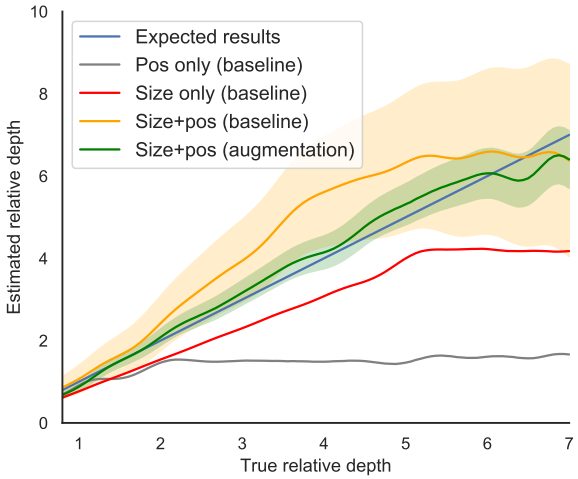
Monocular 3D object detection is the task that localizes foreground objects in a given image with location, orientation, and object dimension information. Compared to stereo or lidar equipment, a monocular camera requires a lower cost to perceive the surrounding environments. However, it suffers from the unreliable depth recovery problem in monocular data. To address this issue, data augmentation might be a promising direction which is widely used to improve the robustness of a machine learning system [28, 50, 37, 27]. Nonetheless, the design of 3D-aware augmentation methods is non-trivial and remains unexploited. In particular, the geometric relationships among different objects are supposed to be preserved when conducting the 3D-aware manipulations, which is not as straightforward as the case in 2D detection. To this end, we empirically study the robustness of monocular 3D detectors in capturing the geometry relationships and convert the analysis to four effective geometry-aware augmentation methods.

It is demonstrated in [10] that neural networks might rely on object apparent size and vertical position as two pictorial cues to predict depth. As visualized in Figure 1a, objects farther away from the camera have smaller apparent sizes and their vertical positions are closer to the vanishing points. To study if detectors are robust in utilizing the two visual cues, we manually distort them (e.g., shifting the object apparent size and vertical position) with three image-level manipulations (visualized in Figure 2) and one instance-level manipulation (visualized in Figure 1a). By evaluating the model robustness against the proposed manipulations, we observe that detectors cannot capture consistent relationships between depth with two pictorial visual cues, even the models can identify the variation of object sizes and positions. As shown in Figure 1b and Figure 3, the estimated depth from the baseline detectors has a strong deviation when the images are geometry manipulated. To better study how pictorial visual cues are used in estimating depth, we further follow [10] and only shift one of the visual cues in the manipulation. Compared to the results with "Size + pos" and "Size only" in Figure 1, the estimated depth in "Pos only" does not change as the visual cue changes. This observation is in contrast to the statement in dense depth estimation [10], where we show that detectors utilize the object's apparent size rather than vertical position to recover depth.

Based on the above analysis, we take a step further than [10] and convert the four geometry manipulations to



(a) Visualization of instance-level copy-paste manipulations.



(b) Visualization of baseline and augmentation-enhanced detectors under instance-level copy-paste manipulations (see the details in Sec 4.2).

Figure 1: We select one of the proposed augmentations (i.e., instance-level copy-paste) to illustrate the stability of depth prediction using both apparent size and vertical position cues. "Size+pos" denotes geometry-consistent manipulation that shifts the two visual cues, "Size only" and "Pos only" denote geometry-inconsistent manipulation that only shifts the vertical position or apparent size. Shaded region indicates the estimated deviation (std) in the "Size + pos" manipulation.

geometry-aware data augmentation techniques for training. At the image-level, we lift the random scale and random crop augmentation to 3D space by shifting the camera focal lengths and receptive field. With the help of a dense depth estimation model, we further propose an augmentation method which shifts the camera position. During the geometry manipulations, the relationships between depth and both object scales and positions are consistent. At the instance-level, we crop the foreground objects in the training data and paste them to a new location with the guidance of geometric relations. By considering the geometric constraints, the proposed augmentation techniques effectively generate new object instances and training images with accurate ground truth. Through adopting the proposed augmentation, the performance of both anchor-free and anchor-based detectors is enhanced and the networks' robustness under geometric manipulations is also improved. Compared to the baseline in Figure 1b, augmentation enhanced detectors are more robust and their outputs have less deviation under the geometric manipulation.

Our contributions are summarized as follows:

- Through a study of how monocular detectors estimate depth, we identified an instability problem of depth recovery under changes of object apparent sizes and positions.
- We provide four geometry-aware data augmentation techniques at the image-level and instance-level to address this problem. With the proposed techniques, we effectively generate more geometry-preserved training data. This approach effectively improves the stability under geometric shifts, leading to better 3D detection performance.
- Experimental results on the KITTI and nuScenes 3D detection benchmarks illustrate the effectiveness of the proposed data augmentation methods.

2. Related work

In this section, we first review the monocular 3D detection and depth estimation fields and then briefly introduce the data augmentation methods used in detection tasks.

2.1. Monocular 3D detection

Monocular 3D detection is the task that identifies the objects and their 3D information in a given image. Traditional approaches [1, 49, 38] estimate 3D detection by lifting image-based detectors to 3D space. MonoDis [38] proposes a loss disentangling module to smooth the optimization in 3D detection. M3D-RPN [1] redesigns the anchor generation module to better extract 3D information. In order to lift the 2D image to the 3D world, prior knowledge is widely used in existing approaches. RTM3D [25], KM3D-Net [26] and MonoPair [7] propose to use the geometric constraints to recovery depth from the constraints in single instance [25, 26] or pairwise instances [7]. Similar to MonoPair [7], RAR-Net [30] further proposes a reinforcement learning based post-processing strategy to refine the 3D information. Unlike the aforementioned approaches, another branch alleviates the 3D information with the help of external data. Mono3d [5] first adopts a semantic segmentation module to extract the contextual information for 3D detection. Later work [4, 11, 36] further utilizes external CAD model [4] and depth information [11, 36] to estimate 3D information. In addition to directly taking the monocular image as input, there are sev-

eral approaches [40, 33, 46, 35, 32] convert the image to pseudo point cloud data and then apply a lidar-based 3D detection on them. Although they achieve superior performance, the input transformation requires an extra depth estimation module during inference, leading to slow inference speed.

2.2. Monocular depth estimation

Unlike 3D detection, depth estimation is the task that assigns depth in pixel-level, which does not need to estimate the objects location and dimension information. With the rapid development of neural networks, there are multiple approaches proposed in recent years [14, 23, 18]. Current approaches can be split into two branches: supervised-based and unsupervised-based. In the supervised-based approaches, neural networks are trained with pixel-level depth ground truth, which have achieved superior performance in the autonomous driving benchmark [16]. Later approaches further alleviates the annotation burden and propose unsupervised-based approaches with the help of video [19, 24] or stereo data [18]. Recently, Dijk et al. [10] and Hu et al. [20] conduct empirical studies on explaining how do neural networks learn the depth information. In this work, we follow [10] and extend the analysis about objects' apparent sizes and vertical positions to 3D detection. Compared to [10], we further identify the instability problem in depth recovery and convert the analysis to four 3D-aware data augmentation methods to alleviate the issue.

2.3. Data augmentation in detection

Data augmentation is one of the most effective ways to boost the detection performance, which does not bring any extra computational cost during inference. Random scale, random crop, color distortion and other geometry and color augmentation techniques are widely adopted in 2D detection models [27, 37, 50, 49]. In addition to the commonly used data augmentation, copy-paste augmentation is also widely discussed in detection and segmentation tasks. Dvornik et al. and Zuo et al. [12, 40] propose to guide the object pasting by matching the visual context before and after augmentation. InstaBoost [13] proposes a probability heatmap to learn where to paste. In the 3D space, one of the recent work [48] proposes an occlusion-aware copy-paste approach for multi-modality 3D detection. In the lidar-based detection, data augmentation is also widely adopted [8, 22, 41]. Besides the common schemes used in object detection, there are several special augmentation methods based on the form of point cloud data, such as the random samples and pastes in SECOND [45], part aware data augmentation method in [9]. Disappointingly, nearly none of the monocular 3D detection methods take these aggressive data augmentation methods due to the violation of geometric constraints, thus making horizontal flip and color distortion are the only data augmentation methods used in this field for a long time. We hope the data augmentation techniques we provide can improve this embarrassing situation and enhance the baseline models.

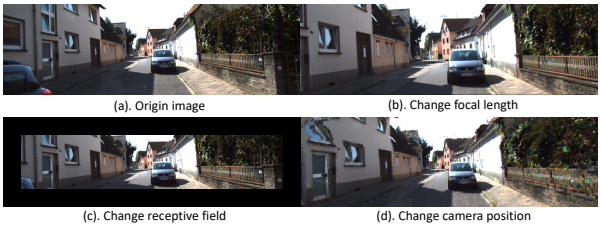


Figure 2: Visualization of the image level manipulation.

3. Preliminaries

3.1. Baselines

In this section, we first introduce the content in the monocular detectors. We use lower-case letters to represent the 2D image coordinate and upper-case letters for the 3D coordinate. Given the input image, the outputs of monocular 3D detectors contain seven components: 1) object category c; 2) 2D bounding box dimension (w, h); 3) 2D bounding box center coordinate (u_0, v_0); 4) 3D bounding box dimension (W, H, L); 5) orientation θ; 6) 3D bounding box center depth Z and 7) 3D bounding box center in 2D coordinate (u_1, v_1). When converting 3D coordinates to 2D coordinates, we follow the KITTI dataset protocol [15] and convert a 3D point $\mathbf{X} = (X, Y, Z, 1)^\intercal$ in camera coordinates to a image coordinate $\mathbf{x} = (u, v, 1)^\intercal$ by

$$Z\mathbf{x} = P_{rect}\mathbf{X}, \tag{1}$$

where the transformation matrix P_{rect} is

$$P_{rect} = \begin{pmatrix} f_u & 0 & c_u & 0 \\ 0 & f_v & c_v & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}. \tag{2}$$

In this work, we adopt both anchor-free (e.g. CenterNet [49]) and anchor-based (e.g., M3D-RPN [1]) models to demonstrate the efficacy of our proposed method.

3.2. Pictorial visual cues

In the field of human and machine perception, researchers [10, 17] suggest several pictorial visual cues that might be used for depth recovery, including object apparent size, vertical position, occlusion, shading, etc. In 3D object detection, the two most important cues are the object's apparent size and vertical position in the image, where the relationships between them with depth are visualized in Figure 4. As shown in Figure 4, the orange triangle displays the visual cue of 2D bounding box height h and 3D bounding

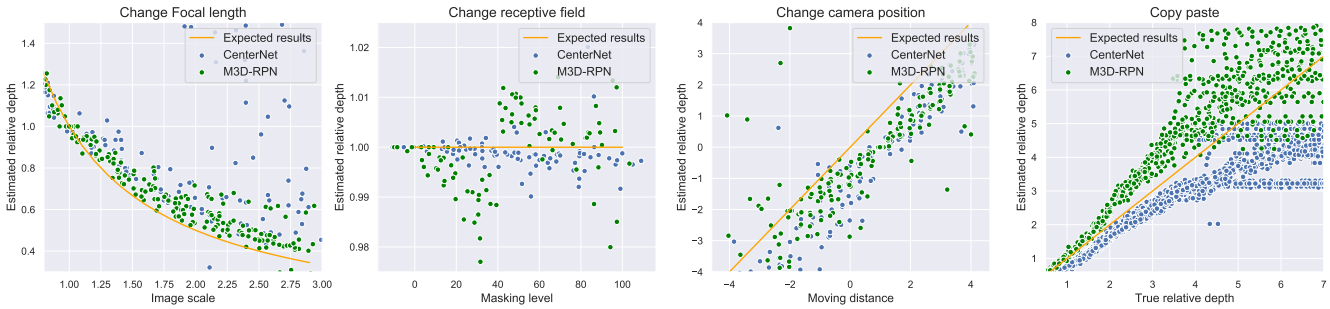


Figure 3: Empirical analysis of monocular detector under geometric manipulations.

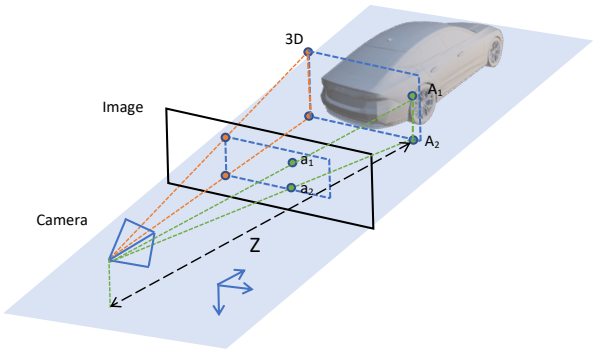


Figure 4: Visualization of the geometric relationships between depth and both object apparent sizes and positions. Figure is best viewed in color.

box height H with depth Z. Given the camera focal length in the v direction f_v, we can infer the depth as the following equation:

$$Z = f_v \frac{H}{h}. \tag{3}$$

The intuition behind this visual cue is that objects that are farther away from the camera tend to have smaller apparent sizes.

Except for the apparent size, detectors can use the vertical position of the object's ground contact points to estimate depth. Given the camera height Y_{cam} above the ground, the height of the horizon v_h in the image, the way to use the vertical position to infer depth is described as the following equation:

$$Z = f_v \frac{Y_{cam}}{v - v_h}. \tag{4}$$

In Figure 4, we visualize the relationship of vertical position with depth in the green triangle, where point A_1 represents one of the horizon lines projected in the object, point A_2 represents one of the object ground contact points. The points that A_1 and A_2 projected in image coordinate are a_1 and a_2, whose vertical positions are v and v_h, respectively. The intuition behind this visual cue is that objects that are closer to the camera tend to have a lower vertical position in the image. Although the two geometric relationships require several assumptions, most of them are satisfied in autonomous driving environments. We refer readers to [10] for a more thorough review of the pictorial cues.

Table 1: Experimental results of M3D-RPN and CenterNet under the proposed manipulations. Except the baseline setting, we replace the ground-truth with estimated results. For example, "- Dep" denotes replacing the ground truth depth with the estimated depth and setting all other metrics as ground truth. (Results of $AP|_{40}$ with IoU$\geq$0.5 on car (easy) category are reported.)

Network	Method	Base	- Dep	- 3D dim	- Pos
M3D-RPN	Origin	54.3	55.6	99.1	98.9
	Focal	31.3	34.8	98.2	98.4
	Recp field	40.2	42.3	95.6	96.7
	Cam pos	25.6	29.4	91.0	89.3
	Copy-paste	35.2	43.3	83.4	97.3
CenterNet	Origin	49.9	50.6	98.9	99.0
	Focal	23.3	27.3	97.8	97.9
	Recp field	38.8	41.0	94.7	94.2
	Cam pos	25.9	28.8	91.7	88.6
	Copy-paste	36.2	42.3	82.0	97.0

4. Analysis based on Geometric manipulations

In this section, we first introduce the proposed three image-level and one instance-level geometric manipulations and then describe the analysis based on them. We evaluate the aforementioned detectors on the KITTI validation set [6] with the proposed geometry manipulations.

4.1. Image-level

Camera focal length The first geometric manipulation is shifting the camera focal length with a scale s, which will change the image scale in 2D space. In the pinhole camera, this operation has the same effect as moving all objects towards or backwards the camera with a scale $\frac{1}{s}$ and a relative

offset in X and Y directions. Formally speaking, we represent this manipulation by using the transformation of 3D points as follows:

$$\mathbf{X_{new}} = \begin{pmatrix} 1 & 0 & (1-s)\frac{c_u}{f_u} & 0 \\ 0 & 1 & (1-s)\frac{c_v}{f_v} & 0 \\ 0 & 0 & s & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} X. \quad (5)$$

Camera receptive field Recently, one study [21] demonstrated that neural networks may learn the position information by image padding. Since image padding is related to the image edge, we randomly manipulate the camera receptive field, which corresponding to cropping and then padding the image edge in the 2D coordinate. Specifically, the manipulation is displayed as follows:

$$I(u,v) = \begin{cases} I(u,v) & u \in [u_a, u_b] \quad v \in [v_a, v_b] \\ 0 & otherwise \end{cases}, \quad (6)$$

where I denotes the input image, u_a, u_b, v_a and v_b are the crop index in the u and v directions, respectively.

Camera position The third geometric manipulation is moving the camera towards or backwards the objects. The transformation for the 3D point is adding a residual d in the Z direction. The transformation for the 2D point is represented as follows:

$$(Z+d)\mathbf{x_{new}} = P_{rect}\left(\begin{pmatrix} 0 & 0 & d & 0 \end{pmatrix}^\intercal + P_{rect}^{-1}\mathbf{x}\right). \quad (7)$$

We first convert it to a 3D point with the help of pixel level depth [14]. Then we add a residual d to shift the camera position change and project it back to the 2D coordinate.

4.2. Instance-level: copy-paste

In addition to the image-level manipulations, we also provide an instance-level manipulation: copy-paste. Copy-paste manipulation is widely used in 2D instance segmentation for generating new training instances. However, most of the approaches [12, 13, 40] focus on how to generate instances that match the real distribution in semantic context and visual appearance. In this work, we focus on providing a geometric consistent copy-paste manipulation that the relationships between depth and both object apparent sizes and positions are consistent.

4.2.1 Geometric consistent copy-paste

We split the manipulation approach into two stages: 1). what to copy and 2). how to paste.

What to copy. In this stage, we first collect an instance database from the training dataset. Since KITTI detection dataset does not contain segmentation annotations, we first apply an external instance segmentation model [44] to crop the objects in the training data. To increase the effectiveness of augmentation, we filter out the data with truncation and low visibility. Since the geometric relationships we used require the ground is flat, we further use the environment vanishing points [10] and object locations to filter the unqualified objects.

How to Paste. After collecting the instance database, we are ready for pasting. To generate new training instances, we randomly sample depth and then calculate the corresponding apparent size based on Eq 3. With the generated new instance, the pipeline for pasting it to an image is described in Algorithm 1.

Algorithm 1 Procedure of copy-paste augmentation.

1: **Input**:Original object with ground truth: $[(u_1, v_1, u_2, v_2), (X, Y, Z), (W, H, L), \theta]$.
2: Sample a new depth $\hat{Z}$.
3: Set the orientation $\hat{\theta} = \theta$.
4: Set the location as $\hat{X} = X\frac{\hat{Z}}{Z}$.
5: Calculate the location $\hat{Y}$ based on Eq 4.
6: Set the dimension as $\hat{W} = W, \hat{H} = H, \hat{Z} = Z$.
7: Calculate the new 2D coordinate $(\hat{x_1}, \hat{y_1}, \hat{x_2}, \hat{y_2})$ by projecting the corner points in 3D boxes to the image.
8: Check if the new instances satisfied the relationship about the apparent size based on Eq 3.
9: **Output**: the new instances with ground truth $[(\hat{x_1}, \hat{y_1}, \hat{x_2}, \hat{y_2}), (\hat{X}, \hat{Y}, \hat{Z}), (\hat{W}, \hat{H}, \hat{L}), \hat{\theta}]$.

Since we can not generate instance with new orientation angle, we fix the orientation as $\hat{\theta} = \theta$ in step 3. In step 5, we utilize the relationship between depth and vertical position to generate the point on the ground and then calculate the object center point with $\hat{Y}$. With the gotten orientation angle, location and dimension, we obtain the image coordinates by projecting the points in the 3D bounding boxes back to the image coordinate. The details in step 6 can be found in the supplementary material. Compared to the manipulation in [10], our instance-level manipulation further ensures the relationships between the new object's apparent size and vertical position with depth are consistent with the original images.

4.3. Stability under different manipulations

In figure 3, we first visualize the estimated depth and expected depth under different manipulations. As illustrated, the estimated depth in CenterNet and M3D-RPN have the trend as expected depth. However, both of them has large deviation, especially for CenterNet. To further evaluate if the detectors can identify the change of visual cues and learn consistent geometric relationships, we evaluate the estimated accuracy of different regression tasks (*e.g.* depth, 3D dimension, position). In Table 1, we display the experimental results under different manipulations, where "base"

denotes using all of the estimated results, "- depth", "- dim" and "- pos" denotes only using the estimated depth, dimension, position offset to replace the ground truth, respectively. We have the following observations. 1). In the origin setting, "- depth" has a larger performance drop than "- dim" and "- pos", showing that the depth recovery is a more difficult task. 2). Both detectors have a large performance drop under four kinds of manipulations, especially for CenterNet. 3). For the results of "- dim" and "- pos", they almost achieve 100% mAP, showing that the detectors can accurately estimate the dimensions and positions of the objects, even in the manipulated image. However, the results in "- depth" have a large gap with 100%, indicating that the detectors can not capture consistent geometry relationships under the manipulations. 4). Unlike the above three manipulations, detectors can not accurately estimate the object dimension for the inserted objects.

5. Geometry-aware data augmentation

After studying the detectors under the geometry manipulations, we take a step further than [10] and convert the geometric manipulations to four geometric consistent data augmentation methods for training.

Random Scale As aforementioned in Section 4.1, we distort the camera focal length to generate the image with a scale from 0.8 to 1.2 and convert the 3D ground truth based on Eq 3.

Random Crop As discussed in Section 4.1, the change of receptive field corresponds to cropping and padding in 2D space. In this augmentation technique, we first crop the images and further pad the image edge with zero value for the cropped region as in Eq 6. Through padding the cropped region, the object vertical position information is kept constant, which preserves the relationship between object depth and vertical position.

Moving camera position Regarding the moving camera position manipulation, we randomly move the camera in the Z direction with the range from -5 to 5. For the coordinate conversion in 2D and 3D coordinate, we adopt the same operation as in Section 4.1 During distorting the camera position, we fill the empty pixel with the nearest neighbor pixel value. Although the effectiveness of the augmentation method relies on the dense depth estimation model, it would not influence instance visual appearance too much. Through the manipulation, we can artificially generate the occlusion scenario and generate new training images. Experimental results also show the effectiveness of the proposed augmentation method.

Copy-Paste For the copy-paste augmentation, we adopt the geometric-consistent manipulation as discussed in Section 4.2. With considering the geometry relationships, both the apparent size and vertical position visual cues are matched the sample depth. As visualized in Figure 5, the

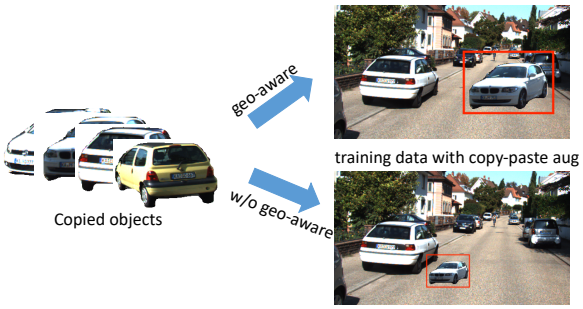


Figure 5: Visualization of Copy-paste data augmentation with and without geometry-aware.

Table 2: Comparison among different strategies of geometry augmentation methods. 3D detection results on the easy setting are reported. "2D only" denotes applying augmentation in the 2D task. "3D only" denotes applying in both 2D and 3D task but the geometric relationships are violated. "3D geo" denotes geometry-aware data augmentation methods.

Method	2D only	3D basic	3D geo
Baseline		13.8	
Random Scale	14.1	12.8	**16.7**
Random Crop	15.1	15.6	**18.5**
Camera Position	13.8	13.7	**17.4**
Copy-paste	13.9	13.2	**17.3**

Table 3: Experimental results of different augmentation methods on the KITTI dataset. $AP|_{40}$ is adopted.

Model		BEV ($AP|_{40}$)			3D($AP|_{40}$)		
		Easy	Mod.	Hard	Easy	Mod.	Hard
M3D	Base	19.75	14.98	12.89	14.53	11.07	8.64
	Geo	22.44	16.45	14.33	16.31	12.03	10.81
	Color	21.02	15.13	13.21	15.03	11.48	9.21
	All	22.65	16.78	14.56	16.92	12.56	11.03
Center	Base	20.96	15.38	13.21	13.84	9.98	8.42
	Geo	28.69	**20.91**	17.83	20.98	14.56	12.72
	Color	22.87	16.11	13.98	16.37	9.74	10.46
	All	**30.59**	20.34	**18.30**	**21.67**	**15.62**	**13.23**

3D geometric relationships are effectively preserved in our proposed geometry-aware augmentation.

6. Experiments

In this section, we first introduce the experimental setup in our work and then display the results with proposed data augmentation on the KITTI and nuScenes datasets.

6.1. Experimental setup

Dataset. We mainly evaluate the detection models on the KITTI 3D object detection benchmark [16]. It consists of 7,481 training images and 7,518 test images with an-

Table 4: **3D detection performance** for the **Car** category on the KITTI dataset. For the validation set, we report both $AP|_{40}$ and $AP|_{11}$ for better comparison. For the test set, we report the $AP|_{40}$ for performance comparison. The best results are highlighted in **bold**. * denotes that the model uses external right camera data in training.

Method	testing ($AP|_{40}$)			validation($AP|_{40}$)			validation ($AP|_{11}$)			RT
	Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard	(ms)
MonoDIS [38]	10.37	7.94	6.40	11.06	7.60	6.37	18.05	14.98	13.42	-
Movi3D [39]	15.19	10.90	9.26	14.28	11.13	9.68	-	-	-	45
MonoPair [7]	13.04	9.99	8.65	16.28	12.30	10.42	-	-	-	57
SMOKE [31]	14.03	9.76	7.84	-	-	-	14.76	12.85	11.50	30
KM3DNet [26]	-	-	-	-	-	-	19.19	16.70	16.14	55
KM3DNet* [26]	16.73	11.45	**9.92**	-	-	-	22.29	17.45	16.86	55
Kinematic [2] (image)	-	-	-	18.28	13.55	10.13	-	-	-	-
M3D-RPN (w/o aug)	14.76	9.71	7.42	14.53	11.07	8.64	20.27	17.06	15.21	161
M3D-RPN (w aug)	-	-	-	16.92	12.56	11.03	22.34	18.93	16.45	161
CenterNet (w/o aug)	-	-	-	13.84	9.98	8.42	20.46	16.86	16.35	47
CenterNet (w aug)	**17.46**	**11.67**	9.69	**21.67**	**15.62**	**13.23**	**25.79**	**21.23**	**18.20**	47

notated "car", "pedestrian", and "cyclist". To tune hyper-parameters and conduct ablation study, we follow [5] and split the training data into a training subset and validation subset with 3,712 and 3,619 images, respectively. The object instances of the KITTI dataset are split into easy, moderate, and hard based on the degrees of occlusion and truncation. To further evaluate the effectiveness of the proposed data augmentation methods on the large-scale dataset, we also conduct experiments on the nuScenes dataset [3]. It contains 40k annotated key frames from 6 different cameras with 4 different scene locations. Compared to the KITTI dataset, it has 7x as many annotations with 23 classes.

Implementation details. As we discussed in Session 3.1, we adopt the modified CenterNet [49] with DLA-34 backbone and M3D-RPN [1] with DenseNet-121 backbone as our baselines. During the training stage, the input images are pre-processed with the proposed data augmentation and fed to the network. During the inference stage, we feed the images to the network with the original resolution and do not adopt any test-time augmentation method.

Data augmentation. As stated in Section 5, we have four geometry-aware data augmentation methods, and the order in the pre-processing stage is: 1). Camera position change, 2). Copy-paste, 3). Random crop, 4). Random scale. For the final results, we also insert the color-based data augmentation to boost the performance, where random saturation, random jitter and random lightning are used. Due to the common use of random horizontal flip, we take it in all experiments.

Evaluation metrics. We follow [38] and adopt the $AP|_{40}$ evaluation metrics which uses 40-point interpolated average precision metric except the one where recall is 0 to avoid trigger bias. We adopt the bird-eye view 2D box AP_{bv} and 3D bounding box AP_{3D} for evaluating the car category.

6.2. Effectiveness of geometry-aware data augmentation

To evaluate the effectiveness of our geometry-aware strategy (denoted as "geo-aware") in the augmentation, we first conduct experiments to compare it with another two commonly used strategies: 1). "2D only": the augmentation methods are only applied for 2D task 2). "3D basic": the 3D ground truths are directly copied from the original data, which may violate the geometric consistency. We adopt CenterNet as the baseline and compare the performance of four augmentation approaches one by one. As illustrated in Table 2, the data augmentation methods that follow the geometry-aware strategy consistently improve the performance and outperform the other two strategies by a large margin. In the 2D only strategy, it brings limited improvement to the baseline, showing the need of lifting the augmentation to the 3D space. Moreover, the 3D basic strategy even hurts the performance in the random scale and copy-paste data augmentation. Compared with the limited power of copy-paste in 2D part [13, 47], our proposed copy-paste data augmentation with geometry-aware brings more than 2% improvements on the AP_{3D} metric.

In Table 3, we display the experimental results with different combinations of geometry-based and colored based augmentation. Compared to the baseline results, our proposed geometry-aware data augmentation improves them with a large margin. When further adding the colored-based augmentation, the augmentation enhanced CenterNet and M3D-RPN outperform the baseline with 7.83% and 2.39 in the easy 3D metrics, respectively.

6.3. Analysis proposed augmentation in M3D-RPN

Although our proposed augmentation methods improve CenterNet by a large margin, they bring limited improvement to the anchor-based M3D-RPN detector. As discussed in Section 5, M3D-RPN is not as vulnerable as CenterNet

under the manipulations. We guess the design of 3D anchor generation layer may cause M3D-RPN to learn the geometry relationship and robust under the manipulations. In M3D-RPN, it contains 32 anchors with different sizes and aspect ratios. Every anchors have predefined object statistics including dimension, location, and orientation, which are the average values of the matched ground truth instances. Different from the CenterNet that directly estimates the ground truth, M3D-RPN estimates the residual between the predefined anchors statistics with the ground truth. In the accumulated statistics, it implicit contains the geometric relationship (where anchors with large size have small predefined depth). To evaluate if this design make M3D-RPN robust under the manipulations, we modify the M3D-RPN to directly estimate the ground-truth label and use it to evaluate the effectiveness of our augmentation methods. As illustrated in Table 5, the direct-based detector has over 2% performance drop compared to residual-based M3D-RPN. When adopting the proposed augmentation methods during training, direct-based detector is effectively improved and achieves comparable performance with residual-based. This observation illustrates why M3D-RPN is relative robust to the proposed manipulations.

Table 5: Experimental results of different M3D-RPN.

Method	Easy	Mod	Hard
M3D-RPN (direct)	11.6	8.0	7.5
+ Geo aug	15.3	11.0	9.6
Improvement	+3.7	+3.0	+2.1
M3D-RPN (residual)	14.5	11.1	8.6
+ Geo Aug	16.3	12.0	10.8
Improvement	+1.8	+0.9	+2.2

6.4. Comparison with State-of-the-arts

To further evaluate the effectiveness of the proposed augmentation methods, we compare our augmentation enhanced detectors with recent state-of-the-art methods on both the KITTI validation and test sets, respectively. As shown in Table 4, our augmentation enhanced CenterNet obtains 17.46%, 11.67%, 9.69% 3D mAP on the KITTI test benchmark, achieving a new state of the art in the image-based methods with a very low inference time. Compared to other CenterNet based methods [7, 25, 31, 26], we effectively improve the baseline in a parameter-free way and do not use any geometric constraint during inference. For the second-best approach Movi-3D [39], it applies a virtual view training strategy and a test-time pre-processing strategy for the input image, which is orthogonal to our training time data augmentation. Regarding the inference speed, we follow MonoPair [7], RTM3D [25] and evaluate it in the 1080 Ti GPU, while SMOKE [31] is evaluated in the TITAN XP and Movi-3D [39] does not describe their accelerator. With the proposed augmentation, CenterNet achieves new state-of-the-arts with very fast inference speed.

Table 6: Comparison CenterNet with the detectors that use external information during inference on the KITTI test benchmark. $AP|40$ 3D metric is adopted.

Method	testing ($AP|_{40}$)			RT
	Easy	Mod.	Hard	(ms)
MonoPL (depth) [43]	10.76	7.50	6.10	400+
D4LCN (depth) [11]	16.65	11.72	9.51	400+
AM3D (depth) [34]	16.50	10.74	9.52	400+
PatchNet (depth) [32]	15.68	11.12	**10.17**	488
Kinematic (video) [2]	**19.07**	**12.72**	9.17	-
CenterNet	17.46	11.67	9.69	47

Table 7: Experimental results of CenterNet on the nuScenes dataset. mAP, mean translation error (mATE), mean size error (mASE), and the weighted (with weight 5 on mAP and 1 on others) average NDS are reported.

Setting	mAP↑	mATE↓	mASE↓	NDS↑
CenterNet	27.9	0.74	0.27	27.8
+ Geo aug	28.9	0.72	0.27	28.6

In Table 6, we also compare the augmentation enhanced CenterNet with the powerful pseudo-lidar and video-based approaches on the KITTI test benchmark. Note that we do not compare the KITTI validation set because the extra dense depth models [42] are trained on the KITTI depth estimation training data, which has some overlap with the detection validation data. We can observe that our image-based approach gets comparable results with pseudo-lidar and video based methods, but has faster inference speed.

6.5. Experimental results on the nuScenes dataset

Since KITTI dataset contains less than 10,000 training images, we further evaluate the proposed augmentation methods in the large scale nuScenes dataset. We adopt the CenterNet as our baseline. Due to the limited of computing resources, we only train the network with 70 epochs in the training set and evaluate the trained model in the official validation set. As illustrated in Table 7, our proposed geometry-aware augmentation methods consistently boost the baseline with more than 1% mAP.

7. Conclusion

In this work, we investigated the stability property of image-based monocular detectors under geometric shifts. Based on the observation that these detectors are not robust enough under geometric shifts, we proposed geometry-aware data augmentation from the image-level and at the instance-level. Our work provides a new way to improve the 3D detection performance by generating more training data with preserving the geometric properties. Experimental results on the KITTI and nuScenes dataset demonstrated the effectiveness of our proposed approach.

References

[1] Garrick Brazil and Xiaoming Liu. M3d-rpn: Monocular 3d region proposal network for object detection. In *ICCV*, 2019. 2, 3, 7

[2] Garrick Brazil, Gerard Pons-Moll, Xiaoming Liu, and Bernt Schiele. Kinematic 3d object detection in monocular video. In *ECCV*, 2020. 7, 8

[3] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *CVPR*, 2020. 7

[4] Florian Chabot, Mohamed Chaouch, Jaonary Rabarisoa, Céline Teulière, and Thierry Chateau. Deep manta: A coarse-to-fine many-task network for joint 2d and 3d vehicle analysis from monocular image. In *CVPR*, 2017. 2

[5] Xiaozhi Chen, Kaustav Kundu, Ziyu Zhang, Huimin Ma, Sanja Fidler, and Raquel Urtasun. Monocular 3d object detection for autonomous driving. In *CVPR*, 2016. 2, 7

[6] Xiaozhi Chen, Kaustav Kundu, Yukun Zhu, Andrew G Berneshawi, Huimin Ma, Sanja Fidler, and Raquel Urtasun. 3d object proposals for accurate object class detection. In *NeurIPS*, 2015. 4

[7] Yongjian Chen, Lei Tai, Kai Sun, and Mingyang Li. Monopair: Monocular 3d object detection using pairwise spatial relationships. In *CVPR*, 2020. 2, 7, 8

[8] Shuyang Cheng, Zhaoqi Leng, Ekin Dogus Cubuk, Barret Zoph, Chunyan Bai, Jiquan Ngiam, Yang Song, Benjamin Caine, Vijay Vasudevan, Congcong Li, Quoc V. Le, Jonathon Shlens, and Dragomir Anguelov. Improving 3d object detection through progressive population based augmentation. In *ECCV*, 2020. 3

[9] Jaeseok Choi, Yeji Song, and Nojun Kwak. Part-aware data augmentation for 3d object detection in point cloud. *arXiv preprint arXiv:2007.13373*, 2020. 3

[10] Tom van Dijk and Guido de Croon. How do neural networks see depth in single images? In *ICCV*, 2019. 1, 3, 4, 5, 6

[11] Mingyu Ding, Yuqi Huo, Hongwei Yi, Zhe Wang, Jianping Shi, Zhiwu Lu, and Ping Luo. Learning depth-guided convolutions for monocular 3d object detection. In *CVPR*, 2020. 2, 8

[12] Nikita Dvornik, Julien Mairal, and Cordelia Schmid. Modeling visual context is key to augmenting object detection datasets. In *ECCV*, 2018. 3, 5

[13] Hao-Shu Fang, Jianhua Sun, Runzhong Wang, Minghao Gou, Yong-Lu Li, and Cewu Lu. Instaboost: Boosting instance segmentation via probability map guided copy-pasting. In *ICCV*, 2019. 3, 5, 7

[14] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *CVPR*, 2018. 3, 5

[15] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013. 3

[16] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012. 3, 6

[17] James J. Gibson. *The perception of the visual world*. Hooughton Mifflin Boston, 1950. 3

[18] Clément Godard, Oisin Mac Aodha, and Gabriel J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *CVPR*, 2017. 3

[19] Ariel Gordon, Hanhan Li, Rico Jonschkowski, and Anelia Angelova. Depth from videos in the wild: Unsupervised monocular depth learning from unknown cameras. In *ICCV*, 2019. 3

[20] Junjie Hu, Yan Zhang, and Takayuki Okatani. Visualization of convolutional neural networks for monocular depth estimation. In *ICV*, 2019. 3

[21] Md Amirul Islam*, Sen Jia*, and Neil D. B. Bruce. How much position information do convolutional neural networks encode? In *ICLR*, 2020. 5

[22] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *CVPR*, 2019. 3

[23] Jin Han Lee, Myung-Kyu Han, Dong Wook Ko, and Il Hong Suh. From big to small: Multi-scale local planar guidance for monocular depth estimation. *arXiv preprint arXiv:1907.10326*, 2019. 3

[24] Hanhan Li, Ariel Gordon, Hang Zhao, Vincent Casser, and Anelia Angelova. Unsupervised monocular depth learning in dynamic scenes. *arXiv preprint arXiv:2010.16404*, 2020. 3

[25] Peixuan Li, Huaici Zhao, Pengfei Liu, and Feidao Cao. Rtm3d: Real-time monocular 3d detection from object keypoints for autonomous driving. In *ECCV*, 2020. 2, 8

[26] Peixuan Li, Huaici Zhao, Pengfei Liu, and Feidao Cao. Rtm3d: Real-time monocular 3d detection from object keypoints for autonomous driving, 2020. 2, 7, 8

[27] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017. 1, 3

[28] Hong Liu, Mingsheng Long, Jianmin Wang, and Michael Jordan. Transferable adversarial training: A general approach to adapting deep classifiers. In *ICML*, 2019. 1

[29] Lijie Liu, Jiwen Lu, Chunjing Xu, Qi Tian, and Jie Zhou. Deep fitting degree scoring network for monocular 3d object detection. In *CVPR*, 2019. 11

[30] Lijie Liu, Chufan Wu, Jiwen Lu, Lingxi Xie, Jie Zhou, and Qi Tian. Reinforced axial refinement network for monocular 3d object detection. In *ECCV*, 2020. 2

[31] Zechen Liu, Zizhang Wu, and Roland Tóth. Smoke: Single-stage monocular 3d object detection via keypoint estimation. In *CVPRW*, 2020. 7, 8

[32] Xinzhu Ma, Shinan Liu, Zhiyi Xia, Hongwen Zhang, Xingyu Zeng, and Wanli Ouyang. Rethinking pseudo-lidar representation. In *ECCV*, 2020. 3, 8

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

[34] Xinzhu Ma, Zhihui Wang, Haojie Li, Pengbo Zhang, Wanli Ouyang, and Xin Fan. Accurate monocular 3d object detection via color-embedded 3d reconstruction for autonomous driving. In *ICCV*, 2019. 8

[35] Rui Qian, Divyansh Garg, Yan Wang, Yurong You, Serge Belongie, Bharath Hariharan, Mark Campbell, Kilian Q. Weinberger, and Wei-Lun Chao. End-to-end pseudo-lidar for image-based 3d object detection. In *CVPR*, 2020. 3

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

[37] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015. 1, 3

[38] Andrea Simonelli, Samuel Rota Rota Bulò, Lorenzo Porzi, Manuel López-Antequera, and Peter Kontschieder. Disentangling monocular 3d object detection. *arXiv preprint arXiv:1905.12365*, 2019. 2, 7

[39] Andrea Simonelli, Samuel Rota Bulò, Lorenzo Porzi, Elisa Ricci, and Peter Kontschieder. Towards generalization across depth for monocular 3d object detection. In *ECCV*, 2020. 7, 8

[40] Hao Wang, Qilong Wang, Fan Yang, Weiqi Zhang, and Wangmeng Zuo. Data augmentation for object detection via progressive and selective instance-switching. *arXiv preprint arXiv:1906.00358*, 2019. 3, 5

[41] Kaixuan Wang and Shaojie Shen. MVDepthNet: real-time multiview depth estimation neural network. In *3DV*, 2018. 3

[42] Yan Wang, Wei-Lun Chao, Divyansh Garg, Bharath Hariharan, Mark Campbell, and Kilian Q Weinberger. Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving. In *CVPR*, 2019. 8

[43] Xinshuo Weng and Kris Kitani. Monocular 3d object detection with pseudo-lidar point cloud. In *ICCVW*, 2019. 8

[44] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. `https://github.com/facebookresearch/detectron2`, 2019. 5, 11

[45] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10):3337, 2018. 3

[46] Yurong You, Yan Wang, Wei-Lun Chao, Divyansh Garg, Geoff Pleiss, Bharath Hariharan, Mark Campbell, and Kilian Q Weinberger. Pseudo-lidar++: Accurate depth for 3d object detection in autonomous driving. *arXiv preprint arXiv:1906.06310*, 2019. 3

[47] Lingzhi Zhang, Tarmily Wen, Jie Min, Jiancong Wang, David Han, and Jianbo Shi. Learning object placement by inpainting for compositional data augmentation. In *ECCV*, 2020. 7

[48] Wenwei Zhang, Zhe Wang, and Chen Change Loy. Multi-modality cut and paste for 3d object detection, 2020. 3

[49] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. In *arXiv preprint arXiv:1904.07850*, 2019. 2, 3, 7

[50] Barret Zoph, Ekin D. Cubuk, Golnaz Ghiasi, Tsung-Yi Lin, Jonathon Shlens, and Quoc V. Le. Learning data augmentation strategies for object detection. In *ECCV*, 2019. 1, 3

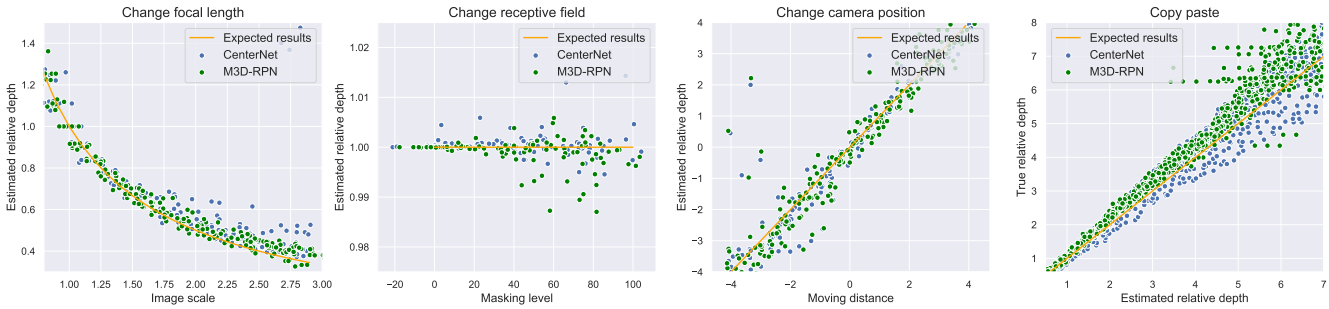


Figure 6: Empirical analysis of augmentation enhanced detectors under geometric shifts

Appendix

1. Hyper-parameters in data augmentation

The hyper-parameters for the data augmentation are represented as follows: 1). Random Crop: we randomly crop the image with size of 960×320. 2). Random Scale: we randomly resize the image with a range from 0.8 to 1.2, with fixing the size ratio. 3). Camera position: To alleviate generate artifact, we control the change distance of camera position from -5 to 5 meters. 4). Copy-paste: We first utilizes an instance segmentation method [44] to crop the foreground objects with around 12,581 instances. After that, we randomly select two cropped instances and insert them into every training image with sampling new depth from 0 - 80. 5). Color-based augmentation: We use Random Brightness, Random Contrast, Random Saturation, and Random Lighting. For the first three approaches, the intensity scale is from 0.6 to 1.4 and the scale of Random Lightning is 0.1. Code will be made available.

2. Details of Copy-paste

Generating bounding boxes For the step 7 in the Algorithm 1, we utilize the acquired object dimension, location, orientation to get the final bounding boxes 2D coordinates. The procedure is similar in [29]. We first calculate the rotation matrix R with using the egocentric orientation angle:

$$R = \begin{pmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{pmatrix}. \tag{8}$$

The 8 corner points in the object coordinate is:

$$P^{3d}_{4\times 8} = \begin{pmatrix} \frac{L}{2} & \frac{L}{2} & -\frac{L}{2} & -\frac{L}{2} & \frac{L}{2} & \frac{L}{2} & -\frac{L}{2} & -\frac{L}{2} \\ 0 & 0 & 0 & 0 & -H & -H & -H & -H \\ \frac{W}{2} & -\frac{W}{2} & -\frac{W}{2} & \frac{W}{2} & \frac{W}{2} & -\frac{W}{2} & -\frac{W}{2} & \frac{W}{2} \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}.$$

For the coordinate of point i, it is calculated as follows:

$$P^{2d}_{3\times 8} = K_{3\times 4} \begin{pmatrix} R & T \\ 0^T & 1 \end{pmatrix} P^{3d}_{4\times 8}, \tag{9}$$

Table 8: Comparisons among different strategies of geometry data augmentation methods. "2D only" denotes applying augmentation in the 2D task "3D only" denotes applying in both 2D and 3D task but the geometric relationships are violated. "3D geo" denotes geometry-aware data augmentation methods.

Method	Setting	Easy	Mod	Hard
Baseline	no aug	13.8	9.9	8.4
Random Scale	2D only	14.1	10.6	9.2
	3D basic	12.8	10.4	8.9
	3D geo	**16.7**	**12.9**	**10.7**
Random Crop	2D only	15.1	10.3	9.3
	3D basic	15.6	11.3	9.2
	3D geo	**18.5**	**13.2**	**11.3**
Cam Pos	2D only	13.8	9.9	8.3
	3D basic	13.7	10.0	8.5
	3D geo	17.3	11.6	9.1
Copy-paste	2D only	13.9	10.9	9.3
	3D basic	13.2	9.4	8.8
	3D geo	**17.3**	**12.6**	**11.0**

where T is the 3D location matrix with $[X, Y, Z]$, and $P^{2d}_{3\times 8}$ is the coordinates in the images.

3. More experimental results

3.1. Effectiveness of geometry-aware data augmentation

We display the experimental results of our geometry-aware augmentation with the CenterNet in Table 8. As illustrated, our proposed geometry-aware strategy effectively improves the baseline and other strategies with three different settings.

3.2. Stability of augmentation enhanced detectors

In Figure 6, we also display the empirical analysis we conducted in Section[4] to evaluate whether our proposed data augmentation methods can enhance the stability. Compared with the baseline results in Figure[4], the results from the augmentation enhanced detectors are more fixed with

the expected results and have less deviation.