國立臺灣大學電機資訊學院電信工程學研究所

碩士論文

Graduate Institute of Communication Engineering
College of Electrical Engineering and Computer Science
National Taiwan University

Master Thesis

# 基於可變形模板匹配之弱監督三維物體檢測

# Weakly Supervised 3D Object Detection via Deformable Template Matching

紀彥仰

Yan-Yang Ji

指導教授：王鈺強 博士

Advisor: Yu-Chiang Frank Wang, Ph.D.

中華民國 112 年 7 月

July 2023

doi:10.6342/NTU202302077

# 中文摘要

三維物體偵測是三維視覺的一個熱門研究領域，近年來受到廣泛關注。然而，訓練用於三維物體偵測的深度學習模型通常需要大量帶有三維邊界框註釋的數據，這是一項耗時的任務並且存在重大挑戰。爲了應對這一挑戰，我們提出了一種通過可變形模板匹配（DTMNet）進行弱監督三維物體偵測的方法，該方法在圖像和二維實例遮罩的弱監督下，通過將可變形形狀模板與輸入的LiDAR 點雲進行匹配，生成弱監督的三維虛擬邊界框。生成的三維虛擬邊界框可以用於訓練基於圖像或基於LiDAR 的三維物體偵測器。我們的DTMNet 顯著降低了註釋成本，提高了三維物體偵測的效率。對KITTI 基準數據集的實驗結果在定量和定性上證明了我們提出的模型的有效性和實用性



doi:10.6342/NTU202302077

# Abstract

3D object detection is an active research topic for 3D vision and has been widely studied in recent years. However, training deep learning models for 3D object detection typically requires extensive data with 3D bounding box annotations, which is a time-consuming task and presents a significant challenge. To address this challenge, we propose a weakly supervised 3D object detection method via deformable template matching (DTMNet), which generates weakly supervised 3D pseudo-bounding boxes by matching a deformable shape template with the input LiDAR point clouds under the weak supervision of images and 2D instance masks. The generated 3D pseudo-bounding boxes can be used to train either image-based or LiDAR-based 3D object detectors. Our DTMNet significantly reduces annotation costs and improves the efficiency of 3D object detection. Experimental results on the KITTI benchmark dataset quantitatively and qualitatively demonstrate the effectiveness and practicality of our proposed model.



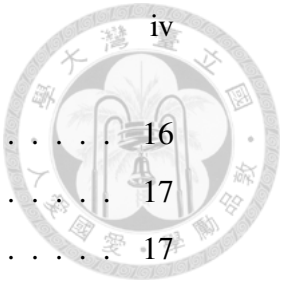
doi:10.6342/NTU202302077

# Contents





doi:10.6342/NTU202302077





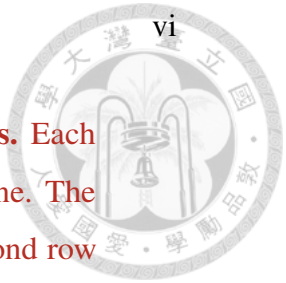
**doi:10.6342/NTU202302077**

# List of Figures





doi:10.6342/NTU202302077





doi:10.6342/NTU202302077

# List of Tables





doi:10.6342/NTU202302077

# Chapter 1

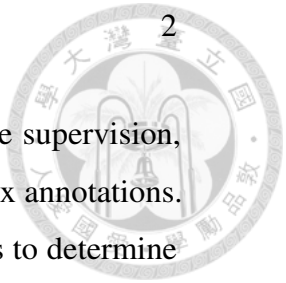
# Introduction

3D object detection has gained substantial attention in recent years, owing to its potential applications in diverse fields such as autonomous driving, robotics, and augmented reality [2, 3, 4]. However, to develop models that accurately detect 3D objects [5, 6, 7], it is typically necessary to gather extensive datasets annotated with 3D bounding boxes. The time-consuming nature of this annotation process has been identified as a bottleneck. For example, it takes an average of 114 seconds to annotate a single 3D bounding box in SUN-RGBD [8]. This limitation has impeded progress of 3D object detection. To overcome this challenge, researchers have proposed semi-supervised [9, 10, 11] and weakly supervised methods [12, 1, 13, 14, 15, 16] for reducing the annotation cost.

Semi-supervised techniques for 3D object detection aim to generate pseudo labels from unlabeled data using teacher-student strategies for self-training. To enhance the quality of these pseudo labels, certain approaches [9, 10, 11] eliminate unreliable ones. On the other hand, WS3D [13] employs a limited number of object-center annotations in bird's eye view (BEV) to train cylindrical object proposals, and approximately 3% 3D bounding box annotations to refine the proposals, producing cuboids and confidence scores. However, despite utilizing these semi-supervised methods, a small amount of 3D bounding box annotation remains necessary, and thus its cost remains a concern.



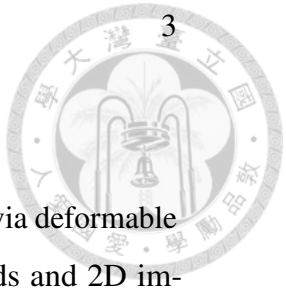
doi:10.6342/NTU202302077



In contrast, weakly supervised methods rely on less expensive supervision, such as 2D annotations, to eliminate the need for 3D bounding box annotations. For instance, VS3D [12] utilizes pre-trained 2D teacher networks to determine the objectness and 3D orientation of the 3D bounding box proposals generated from LiDAR point cloud based on normalized density. Zakharov *et al.* [1] employs pre-trained normalized object coordinate spaces (NOCS) [17] and DeepSDF [18] networks to match a 3D car shape with 2D mask and LiDAR point cloud. Nevertheless, these weakly supervised methods generally rely on strong 3D-related priors that necessitate additional datasets rich in 3D information, which may not be available in some situations. To address this issue, recent approaches such as WeakM3D [16] generate virtual rays from the camera to the LiDAR point cloud for matching the surface of a fixed-size 3D bounding box with the LiDAR point cloud. Conversely, McCraith *et al.* [15] leverage a fixed 3D car template to match with the points in the frustum of a 2D mask. Nevertheless, despite the efforts to avoid relying on strong 3D-related priors, the use of a single fixed bounding box or template may not be sufficient to precisely match with the wide range of real-world cars, making it challenging to achieve accurate 3D object detection solely through weak supervision without the aid of such priors.

In this paper, we introduce a novel weakly supervised 3D object detection method via deformable template matching (DTMNet), which is capable of generating 3D pseudo-bounding boxes suitable for training image-based or LiDAR-based 3D object detectors. The proposed framework utilizes LiDAR point cloud data, 2D images, and 2D instance masks of cars in the images to accurately match a deformable shape template with the point cloud data filtered by each 2D mask. In addition, we incorporate dense 2D edge and color supervision to enhance the learning process and improve the matching of the deformable shape template. Our experimental results demonstrate that our approach outperforms current state-of-the-art weakly supervised methods and achieves comparable performance to fully supervised learning methods.

doi:10.6342/NTU202302077



We now summarize the contributions of this work below:

- We propose a weakly supervised 3D object detection network via deformable template matching (DTMNet), utilizing LiADR point clouds and 2D images with instance masks to generate 3D pseudo-bounding boxes without observing 3D ground truth annotation.
- By incorporating edge supervision, DTMNet matches 2D edges of the deformable template with those of the instance mask, properly resulting in detailed geometry of the object template with target instances.
- With image color supervision, our DTMNet ensures projected point clouds and the target instances in the input image share similar color representations, avoiding detecting non-target objects.
- Through extensive experiments, we demonstrate that our proposed DTMNet achieves state-of-the-art performance in various settings, including camera-based and LiDAR-based 3D object detection tasks.

doi:10.6342/NTU202302077

# Chapter 2

# Related Work

## 2.1 Supervised 3D object detection

Supervised 3D object detection methods aim to predict 3D bounding boxes by using 3D bounding box annotations as supervision, given either RGB images or LiDAR point clouds as input. RGB image-based methods rely solely on RGB images during training and inference, while LiDAR-based methods use point clouds. For the image-based method, PatchNet [7] incorporates depth information predicted by a pre-trained depth predictor into the image representation, dividing samples from different distance levels into separate branches for 3D detection. For LiDAR-based methods, PointRCNN [5] adopts a two-stage pipeline for point cloud analysis, generating region proposals using a point-based backbone network in the first stage, performing RoI pooling and feature refinement in the second stage for accurate object localization and classification. Another example is PointPillars [6], which processes point clouds with a pillar-based representation, followed by 2D networks for feature extraction and object detection. Despite their effectiveness, supervised methods have high annotation costs, limiting their practical applicability.

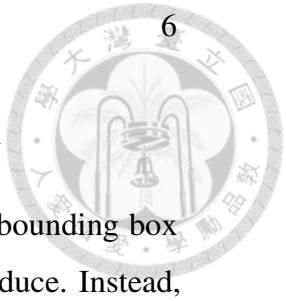


doi:10.6342/NTU202302077

## 2.2 Semi-supervised 3D object detection

Semi-supervised 3D object detection techniques have been developed to address the issue of high annotation costs associated with 3D bounding boxes. These methods leverage unlabeled data to generate pseudo labels, reducing the need for a large number of annotated examples. One prevalent strategy involves using a teacher-student framework, in which a teacher model trained on a small labeled dataset generates pseudo labels for a student model to learn. For instance, 3DIoUMatch [9] predicts IoU-based pseudo labels, which can filter out unreliable samples and enhance the quality of 3D object detection outcomes. Another approach proposed by Wang *et al.* [10] employs a Graph Neural Network (GNN) to improve the consistency of pseudo labels. Their GNN constructs a graph consisting of nodes representing the features of pseudo-3D boxes in LiDAR video frames and edges connecting similar nodes, carrying their differences as edge features. By learning the temporal and spatial consistency of the graphs, the GNN predicts the confidence of each pseudo-3D box. Alternatively, Yin *et al.* [11] use a spatial-temporal ensemble module and a clustering-based bounding box voting module to handle false negatives and false positives in pseudo labels, respectively, and introduce a soft supervision signal through box-wise contrastive learning. Despite the impressive performance achieved by these semi-supervised techniques, they still require approximately 10% 3D bounding box annotations to achieve 90% of the performance of fully supervised methods, making them less practical in real-world scenarios. To address this, some methods, such as WS3D [13] exploit weak object-center annotations in bird's eye view (BEV) to train cylindrical object proposals, using only a limited number of 3D bounding box labels to refine them. With this approach, WS3D generates cuboids and confidence scores with only about 3% 3D bounding box annotations, demonstrating the potential of weak supervision in reducing annotation costs even further.

doi:10.6342/NTU202302077



## 2.3 Weakly supervised 3D object detection

Weakly supervised methods aim to eliminate the reliance on 3D bounding box annotations, which are time-consuming and labor-intensive to produce. Instead, these methods utilize other types of supervision that can be labeled much more quickly, such as 2D bounding boxes or semantic segmentation masks. For example, VS3D [12] generates 3D bounding box proposals from LiDAR point cloud data using a normalized point cloud density approach. It then leverages pre-trained 2D teacher networks to predict objectness and 3D orientation confidence from images, which guide the point cloud-based student network to detect cars in the LiDAR point cloud. Another method [1] utilizes pre-trained normalized object coordinate spaces (NOCS)[17] and DeepSDF [18] networks to initialize a 3D car shape and optimizes the 3D car shape using RANSAC [19] to match it precisely with the 2D mask and LiDAR point cloud. However, these weakly supervised methods rely on pre-trained models that require additional datasets rich in 3D information, such as the pre-trained 3D orientation network in VS3D and the pre-trained 3D pose predictor in [1]. To overcome this challenge, some proposed methods rely on assumptions related to the properties of the vehicles in the scene. For example, FGR [14]assumes that the LiDAR point clouds of a car are always matched with two edges of a bounding box in the bird's eye view (BEV), and optimization is employed to determine the optimal orientation of the bounding box. However, FGR may struggle to accurately identify the correct orientation when the LiDAR sensor observes only one side of a car. WeakM3D [16] estimates the vehicle size by computing statistics from annotated datasets and generates virtual rays from the camera to the LiDAR point cloud filtered by a 2D bounding box to match the surface of a fixed-size 3D bounding box (according to the calculated vehicle size) with the LiDAR point cloud. Meanwhile, McCraith *et al.* [15] employs a fixed 3D shape template to match with the points in the frustum of 2D segmentation masks. However, fitting the point cloud of a vehicle with a single fixed-size bounding box or template may not accurately match with the varying sizes of real-world vehicles.

doi:10.6342/NTU202302077

| Method | Input | | 2D annotation | Auxiliary 3D annotations | Assumption |
|---|---|---|---|---|---|
| | Training | Inference | | | |
| VS3D [12] | LiDAR + image | LiDAR/image | - | ✓ | - |
| Zakharov *et al.* [1] | LiDAR + image | LiDAR/image | mask | ✓ | - |
| FGR [14] | LiDAR + image | LiDAR/image | box | - | BEV edges |
| McCraith1 *et al.* [15] | LiDAR + image | LiDAR | mask | - | fixed 3D shape |
| WeakM3D [16] | LiDAR + image | image | mask | - | fixed 3D box |
| Ours | LiDAR + image | LiDAR/image | mask | - | *deformable* 3D shape |

Table 2.1: **Comparisons between different weakly supervised 3D object detection methods.** Note that VS3D requires a pre-trained 2D classifier to determine objectness and a pre-trained network for 3D orientation prediction. On the other hand, Zakharov *et al.* . require a pre-trained 3D pose predictor.

Compared with the methods discussed above, our method relies solely on 2D segmentation masks of cars as annotated data and does not rely on auxiliary 3D annotations. Our approach is similar to FGR, McCraith *et al.* , and WeakM3D in terms of weakly-supervised settings. However, our method uses deformable shape template to match the various appearances of cars in LiDAR scenes, which is different from the fixed-size bounding boxes or templates used in McCraith *et al.* and WeakM3D. In Table 2.1, we provide a comparison of our proposed method with the aforementioned weakly supervised 3D object detection methods.

doi:10.6342/NTU202302077

# Chapter 3

# Proposed Method

## 3.1 Problem formulation and model overview

We first define the problem definition and the notations used in this paper. For simplicity, we take only *one* car as an example target to explain our approach. In our weakly supervised detection task, one observes a LiDAR point cloud $L \in \mathbb{R}^{N\times 3}$ in an outdoor scene, where $N$ represents the number of points in the scene, and each point is represented in three-dimensional coordinates. An RGB image $I \in \mathbb{Z}^{H\times W\times 3}$, where $H$ and $W$ denote the height and width of the image, as well as a 2D car mask $m \in \{0,1\}^{H\times W}$ of the same outdoor scene are also inputs observed during training. As auxiliary supervision, a deformable 3D car template $M(\cdot) \in \mathbb{R}^{N_T\times 3}$ composed of a mean shape $M_0$ and $r$ basis $[\boldsymbol{p}_1, \boldsymbol{p}_2, ..., \boldsymbol{p}_r]$ is provided, where $N_T$ is the number of points in the template. To be more specific, given a set of parameters $\boldsymbol{s} = [s_1, s_2, ..., s_r]$, the corresponding deformed template $M(\boldsymbol{s})$ is calculated as:

$$M(\boldsymbol{s}) = M_0 + \sum_{k=1}^{r} s_k \boldsymbol{p}_k, \tag{3.1}$$

where $\boldsymbol{p}_k$ and $s_k$ are the $k^{th}$ basis and the associated coefficient. Inspired from [20], the basis is obtained by applying Principal Component Analysis (PCA) [21] to the sampled point clouds of multiple car CAD models for training. Given the above training inputs and supervision, our goal is to predict the optimal deformation



doi:10.6342/NTU202302077

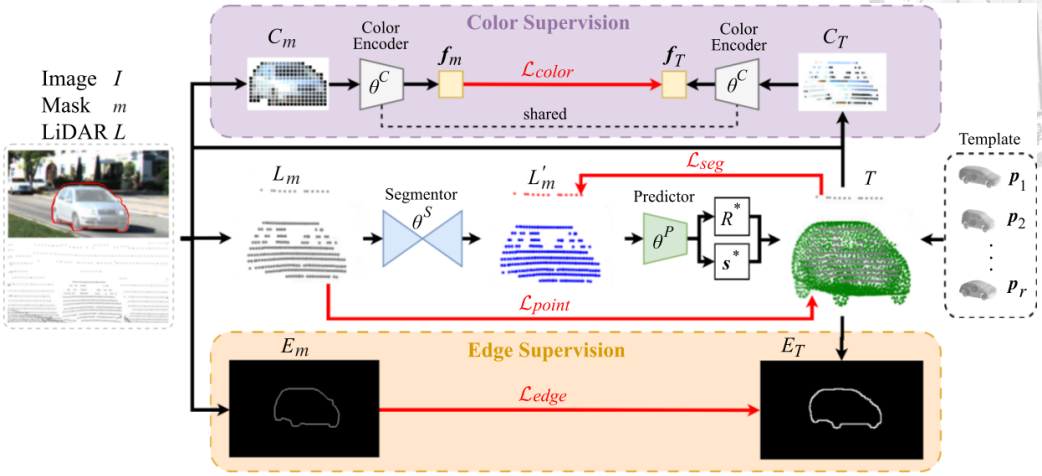
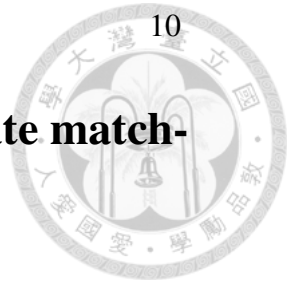



Figure 3.1: Architecture of our proposed DTMNet, which contains a Segmentor $\theta_S$, a Predictor $\theta^P$, and a Color Encoder $\theta^C$. Our DTMNet matches the car instances using LiDAR and image inputs, generating a 3D bounding box by matching a pre-collected car template $[\boldsymbol{p}_1, \boldsymbol{p}_2, ..., \boldsymbol{p}_r]$ without additional 3D annotation.

coefficients $\boldsymbol{s}^*$ and rigid transformation $R^*$ of the 3D car template $M(\cdot)$, where $R^*$ includes a $3 \times 3$ yaw rotation matrix and a $3 \times 1$ translation vector, to match the template with the target car object in the LiDAR scene and to generate the corresponding 3D pseudo bounding box without any 3D bounding box annotation.

To tackle the above problem, we propose a novel weakly supervised 3D object detection method via deformable template matching (DTMNet). Our architecture is illustrated in Figure 3.1, with a *Segmentor* $\theta^S$ designed to segment the points belonging to the target car in the masked LiDAR point cloud $L_m$ (i.e., masking the input point cloud $L$ with the 2D car mask $m$) and a *Predictor* $\theta^P$ which aims to match the template with the segmented point clouds. Moreover, our proposed DTMNet architecture utilizes 2D supervision, including edge supervision from the 2D car mask $m$ and color supervision from the RGB image $I$. By incorporating these designs, our DTMNet can accurately match the template with the car in the LiDAR scene and produce a 3D pseudo-bounding box, allowing for the training of 3D object detectors in the absence of 3D annotations.

doi:10.6342/NTU202302077



## 3.2 Weakly supervised deformable template matching

### 3.2.1 Segmentor and Predictor

Provided with the LiDAR point cloud $L$, the car mask $m$, and the deformable car template $M(\cdot)$, the Segmentor $\theta^S$ in Figure 3.1 aims to segment points corresponding to the car from the masked point cloud $L_m$, where $L_m$ is defined as:

$$L_m = \{\boldsymbol{z} \mid \forall \boldsymbol{z} \in L \ni m_{K\boldsymbol{z}} = 1\}. \tag{3.2}$$

Here, $K$ denotes the projection function that maps a 3D point $\boldsymbol{z}$ in $L$ onto its corresponding 2D image coordinate, and the binary mask value $m_{K\boldsymbol{z}}$ corresponds to the nearest binary mask value of the projected point $K\boldsymbol{z}$ on the 2D car mask $m$. By applying $\theta^S$, we are able to further filter out the outliers and noises in $L_m$ caused by the mismatch of 2D masks or LiDAR scanning through transparent parts of the car such as windows (as addressed in [15]). On the other hand, the Predictor $\theta^P$ in Figure 3.1 aims to predict deformation coefficients $\boldsymbol{s}^*$ and rigid transformation $R^*$ that transform the deformable template $M(\cdot)$ to match with the point cloud segmented by $\theta^S$, denoted as $L'_m$. The transformed template $T$ can be expressed as follows:

$$T = \{R^*\boldsymbol{x} \mid \forall \boldsymbol{x} \in M(\boldsymbol{s}^*)\}, \tag{3.3}$$

where $T$ represents the set of all points obtained by transforming the deformed 3D car template $M(\boldsymbol{s}^*)$ using $R^*$.

Since we are not allowed to obtain the ground truth segmentation label for $\theta^S$, we use the transformed template $T$ as a weak supervision for $\theta^S$ by treating points in $L_m$ that are close enough to $T$ as segmentation label $\boldsymbol{y}$ for $\theta^S$. The segmentation label $\boldsymbol{y}$ is denoted as:

$$\boldsymbol{y} = \{y_{\boldsymbol{x}} \mid \forall \boldsymbol{x} \in L_m\}, \tag{3.4}$$

doi:10.6342/NTU202302077

where $y_{\boldsymbol{x}}$ is defined as:

$$y_{\boldsymbol{x}} = \begin{cases} 1 & \text{if } d(\boldsymbol{x}, T) < th_{3D}, \\ 0 & \text{otherwise.} \end{cases} \tag{3.5}$$

In the above equation, $th_{3D}$ is a pre-defined threshold and the minimum Euclidean distance $d(\boldsymbol{x}, T)$ between a point $\boldsymbol{x}$ and $T$, defined as:

$$d(\boldsymbol{x}, T) = \min_{\boldsymbol{x'} \in T} \|\boldsymbol{x'} - \boldsymbol{x}\| . \tag{3.6}$$

To this end, we can define the learning objective $\mathcal{L}_{seg}$ for $\theta^S$ as:

$$\mathcal{L}_{seg} = \mathcal{L}_{BCE}(\boldsymbol{y}^*, \boldsymbol{y}), \tag{3.7}$$

where $\boldsymbol{y}^*$ denotes the predicted probabilities of points belonging to the car, and $\mathcal{L}_{BCE}(\cdot)$ denotes the binary cross entropy loss. As for the objective for $\theta^P$, we propose a point matching loss to encourage $T$ to be closer to each point in $L_m$:

$$\mathcal{L}_{point} = \frac{1}{|L_m|} \sum_{\boldsymbol{x} \in L_m} \frac{d_{clip3D}(\boldsymbol{x}, T)}{N(\boldsymbol{x})}, \tag{3.8}$$

where $N(\boldsymbol{x})$ denotes the normalization factor to balance the influence of dense and sparse regions in $L_m$. It is computed as the number of points in the local neighborhood of each point $\boldsymbol{x}$ in $L_m$. Additionally, $d_{clip3D}(\boldsymbol{x}, T)$ is the clipped distance between the point $\boldsymbol{x}$ and the transformed template $T$ to reduce the negative impact of noises and outliers in $L_m$ by clipping the distance that exceeds a certain threshold. $d_{clip3D}(\boldsymbol{x}, T)$ is computed as:

$$d_{clip3D}(\boldsymbol{x}, T) = \begin{cases} d(\boldsymbol{x}, T) & \text{if } d(\boldsymbol{x}, T) < th_{3D}, \\ th_{3D} & \text{otherwise,} \end{cases} \tag{3.9}$$

where $th_{3D}$ is a threshold for clipping. By learning from $\mathcal{L}_{point}$, $\theta^P$ can roughly match the template with the car in the scene.

doi:10.6342/NTU202302077

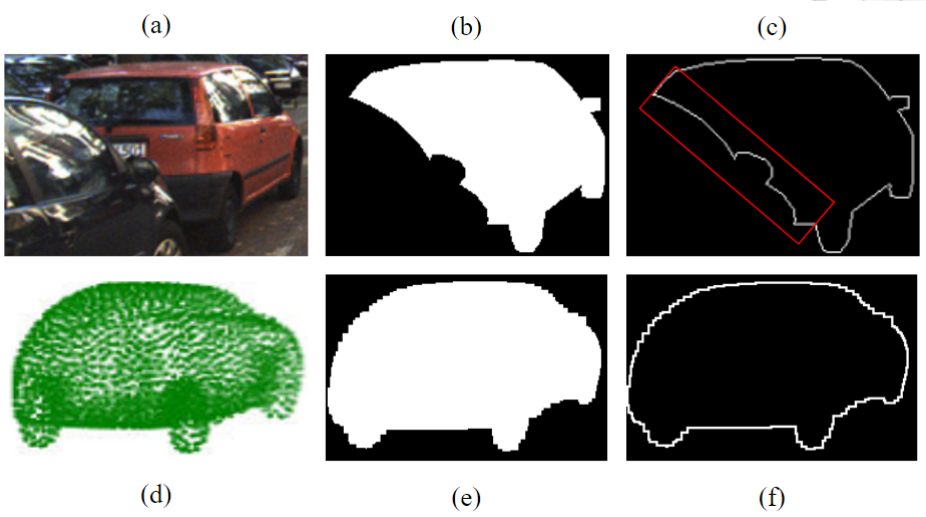



Figure 3.2: **Example comparisons between occlusion edges and the edges captured from the transformed template $T$.** We show the example of (a) the car in the image, (b) its corresponding mask $m$, (c) the edges of the mask $E_m$, (d) the transformed template $T$, (e) the 2D mask projected from $T$, and (f) the 2D edge of projection of transformed template $E_T$.

### 3.2.2 Edge and color supervision

Despite the effectiveness of filtering out noises using $\theta^S$ and matching transformed templates with the target object using $\theta^P$, we still face challenges in accurately matching fine-grained shapes, such as side mirrors, and may erroneously match with points that partially share similar shapes with cars. To overcome these limitations, we propose the use of 2D supervision, comprising edge and color supervision, to capture the detailed geometry of the object, including its fine-grained shapes, while avoiding matching errors with non-target objects, respectively.

**Edge supervision.** To improve deformable template matching with fine-grained shapes, such as side mirrors, we incorporate dense 2D edge information to supervise the deformable template matching. The edge supervision is achieved by matching the 2D projection edges $E_T$ of $T$ with the edges $E_m$ of the 2D car mask $m$. To be

doi:10.6342/NTU202302077

more specific, $E_m$ is obtained by applying a Sobel filter $Sobel(\cdot)$ to $m$ for edge detection, and $E_T$ can be represented using the following formula:

$$E_T = Sobel(mask(KT)), \tag{3.10}$$

where $K$ is the same projection matrix in Eqn. 3.2 and $mask(\cdot)$ represents the mask construction operation (please refer to the supplementary materials for further details). However, matching $E_T$ with $E_m$ is challenging when the car is occluded by other objects in the scene as depicted in Figure 3.2, where the occlusion edges highlighted in (c) should not be matched with $E_T$. To address these challenges, we propose the edge matching loss to encourage $E_T$ to be closer to each pixel in $E_m$ while mitigating the influence of the occlusion edges. The edge matching loss is defined as:

$$\mathcal{L}_{edge} = \frac{1}{|E_m|} \sum_{\boldsymbol{u} \in E_m} d_{clip2D}(\boldsymbol{u}, E_T), \tag{3.11}$$

where $d_{clip2D}(\boldsymbol{u}, E_T)$ denotes the distance between a 2D edge pixel $\boldsymbol{u}$ in $E_m$ and the entire $E_T$. In order to address the negative effects of occlusion edges in $E_m$, this distance is clipped by setting a threshold $th_{2D}$. The clipped distance is defined as:

$$d_{clip2D}(\boldsymbol{u}, E_T) = \begin{cases} d_{2D}(\boldsymbol{u}, E_T) & \text{if } d_{2D}(\boldsymbol{u}, E_T) < th_{2D}, \\ th_{2D} & \text{otherwise}, \end{cases} \tag{3.12}$$

where $d_{2D}(\boldsymbol{u}, E_T)$ is computed as:

$$d_{2D}(\boldsymbol{u}, E_T) = \min_{\boldsymbol{u'} \in E_T} \|\boldsymbol{u'} - \boldsymbol{u}\|, \tag{3.13}$$

The incorporation of edge supervision in our proposed framework enables the extraction of detailed shape information of the vehicle in the input scene. This, in turn, facilitates the more precise prediction of the deformation coefficients $\boldsymbol{s}^*$ and rigid transformation $R^*$ by the Predictor network $\theta^P$.

**Color supervision.** While the use of edge supervision and point matching can be valuable in accurately identifying objects, they may not be adequate in distinguishing between objects with similar shapes. Therefore, the inclusion of color

doi:10.6342/NTU202302077

supervision is essential as it can improve the accuracy of the deformable template matching by ensuring that $T$ is matched with the intended target object and not with something similar in shape. The primary objective of the color supervision is to facilitate similarity between the global representations of color values $C_T$, which correspond to the points that are matched by the template, and the color values $C_m$ of the car present in the image $I$ that has been cropped by $m$. To be more specific, $C_T$ is obtained by:

$$C_T = \{I_{K\boldsymbol{x}} \mid \forall \boldsymbol{x} \in L_m \ni d(\boldsymbol{x}, T) < th_{3D}\}, \tag{3.14}$$

where $K$ refers to the same projection matrix as in Eqn. 3.2 and $I_{K\boldsymbol{x}}$ is the nearest pixel value on $I$ with respect to the projected point $K\boldsymbol{x}$. On the other hand, the color values in the masked image $C_m$ are obtained by selecting the colors of pixels in the masked image based on $m$.

To encourage similarity between the global representations of $C_T$ and $C_m$, we introduce a *Color Encoder* $\theta^C$. The Color Encoder is combined with the color contrastive loss, which is used to promote similarity between positive pairs of representations. The color contrastive loss, denoted by $\mathcal{L}_{color}$, is defined as follows:

$$\mathcal{L}_{color} = \mathcal{L}_{con}(\boldsymbol{f}_m, \boldsymbol{f}_T), \tag{3.15}$$

where $\mathcal{L}_{con}(\cdot)$ is a contrastive loss. Although we do not impose any restrictions on the selection of this contrastive loss, we acknowledge that some contrastive learning-based techniques may necessitate a large number of negative pairs, which can be computationally expensive. Therefore, we adopt the Barlow Twins loss function [22] in our methodology, which does not necessitate the observation of negative examples. The global color representations of $C_m$ and $C_T$ are denoted as $\boldsymbol{f}_m$ and $\boldsymbol{f}_T$, respectively, and can be computed as follows:

$$\boldsymbol{f}_m = \theta^C(C_m), \tag{3.16}$$

$$\boldsymbol{f}_T = \theta^C(C_T). \tag{3.17}$$

doi:10.6342/NTU202302077

With the color supervision design, our method is able to accurately distinguish between objects with similar shapes, as the color information helps match $T$ with the intended target object.

## 3.3 Training and obtaining pseudo-bounding box

During training, we summed up $\mathcal{L}_{point}$, $\mathcal{L}_{seg}$, $\mathcal{L}_{edge}$, and $\mathcal{L}_{color}$ to form the full objective of our proposed DTMNet framework. The objective allows for accurate matching of the template with the car present in the scene and generates a 3D pseudo-bounding box (i.e., directly calculate the width, height, and length of the template as the size of the pseudo-bounding box, with $R^*$ applied as the transformation), which in turn facilitates the training of a 3D object detector without the need for 3D annotations. Our experimental results demonstrate that our proposed framework outperforms existing methods in various settings, including image-based and LiDAR-based 3D object detection tasks.

$$\mathcal{L} = \mathcal{L}_{point} + \mathcal{L}_{seg} + \mathcal{L}_{edge} + \mathcal{L}_{color}, \tag{3.18}$$

doi:10.6342/NTU202302077

# Chapter 4

# Experiments

## 4.1 Dataset and implementation details

### 4.1.1 Dataset

We utilize the KITTI Object Detection dataset [23] for our experiments. This benchmark contains 7481 training pairs of RGB images and point clouds along with 2D instance mask and 3D bounding box annotations of three different classes, i.e., cars, cyclists, and pedestrians. Note that the only annotation used in our approach is the 2D instance mask. We follow the standard protocol in [24] and split the dataset into a training set (3,712 samples) and a validation set (3,769 samples). In our experiments, we evaluate our approach on the car category samples only.

### 4.1.2 Implementation Details

We utilize the PyTorch library [25] to implement our proposed approach. To construct the *Predictor* $\theta^P$, *Segmentor* $\theta^S$, and *Color Encoder* $\theta^C$, we employ PointNet-like architectures [26]. We set the number of points in the car template $M(\cdot)$, denoted as $N_T$, to 3072. The threshold $th_{3D}$ defined in Eqn. 3.9 is initialized at 1.5 meters and is gradually reduced by 0.2 meters per epoch until it reaches 0.7 meters. On the other hand, threshold $th_{2D}$ in Eqn. 3.12 is set to 10. The value of



doi:10.6342/NTU202302077

| Method | Inference | Supervision | | $AP_{BEV}/AP_{3D}(IoU=0.5)$ | | |
|---|---|---|---|---|---|---|
| | | 2D | 3D | easy | moderate | hard |
| Patchnet [7] | Mono | | ✓ | 71.70 / 68.26 | 50.60 / 47.93 | 43.29 / 41.38 |
| VS3D (Mono) [12] | Mono | ✓ | | - / 31.35 | - / 23.92 | - / 19.34 |
| WeakM3D [16] | Mono | ✓ | | 58.20 / 50.16 | 38.02 / 29.94 | 30.17 / 23.11 |
| Patchnet (Ours) | Mono | ✓ | | **71.07 / 64.53** | **50.16 / 42.30** | **42.63 / 38.26** |
| PointRCNN [5] | LiDAR | | ✓ | 97.74 / 97.70 | 89.88 / 89.84 | 89.35 / 89.25 |
| PointPillars [6] | LiDAR | | ✓ | 97.46 / 97.23 | 93.70 / 93.70 | 91.37 / 88.99 |
| VS3D (LiDAR) | LiDAR | ✓ | | - / 42.43 | - / 41.58 | - / 32.74 |
| Zakharov *et al.* [1] | LiDAR | ✓ | | 94.90 / 90.70 | 88.50 / 71.10 | - / - |
| McCraith *et al.* [15] | LiDAR | ✓ | | 86.52 / 83.45 | 86.22 / 79.53 | 75.53 / 71.01 |
| PointRCNN (Ours) | LiDAR | ✓ | | 95.71 / **94.84** | 88.37 / 87.60 | **87.51 / 86.15** |
| PointPillars (Ours) | LiDAR | ✓ | | **96.83** / 90.33 | **89.49 / 88.48** | 85.13 / 79.46 |

Table 4.1: **Performance comparisons on KITTI validation set for cars.**In this table, the *Inference* column refers to the input modality for inference, where *Mono* denotes monocular images and *LiDAR* denotes LiDAR point clouds.

$N(\boldsymbol{x})$ is determined by counting the number of points that are situated within a distance of 0.1 meters from the point $\boldsymbol{x}$. During training, we use a single NVIDIA RTX 3090Ti GPU with a batch size of 32 and a learning rate of 0.01 using the ADAM optimizer [27] for 30 epochs.

## 4.2 Weakly supervised 3D object detection

### 4.2.1 Quantitative evaluation

We now present a quantitative evaluation of our proposed DTMNet. To show the effectiveness and reliability of our approach, we train several state-of-the-art 3D object detectors with our 3D pseudo-bounding boxes, including both image-based (PatchNet [7]) and LiDAR-based (PointRCNN [5], PointPillars [6]) methods. We

doi:10.6342/NTU202302077

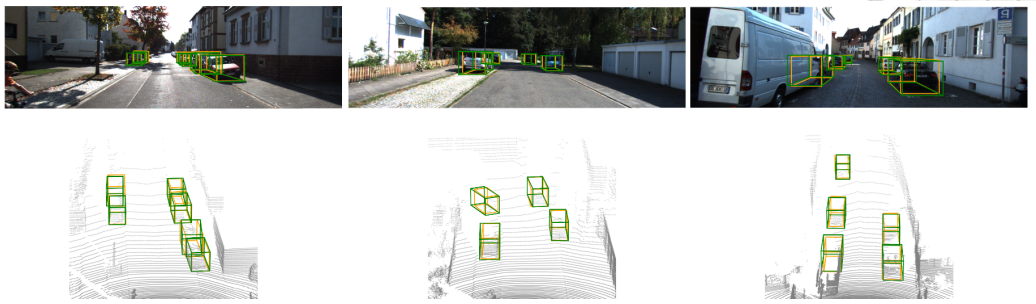


Figure 4.1: **Qualitative results of 3D object detection.** The green boxes are the 3D object detection results of our method, and the orange boxes are the ground truth 3D bounding boxes.

compare our results against several state-of-the-art weakly supervised approaches, including VS3D [12], WeakM3D [16], Zakharov *et al.* [1], and McCraith *et al.* [15]. To ensure a fair comparison, we utilize the average precision for both BEV and 3D boxes with a 0.5 IoU threshold, as suggested by prior relevant works [12, 1], in all difficulty categories. We summarize our quantitative comparisons in Table 4.1, which shows that our proposed DTMNet method achieves the best scores in all metrics, surpassing many other existing approaches by a significant margin, particularly in the *hard* category. It is worth noting that Zakharov *et al.* used their method to generate pseudo labels for training PointPillars, which should be compared with the performance of PointPillars trained with our 3D pseudo-bounding boxes. Moreover, we also compare our proposed DTMNet approach with supervised methods by training the aforementioned 3D object detectors with ground truth 3D bounding boxes. The experimental results reveal that our method achieves 90% of the performance of the supervised methods in all metrics.

### 4.2.2 Qualitative result

To further demonstrate the effectiveness of our proposed method, we provide qualitative results of 3D object detection in Fig. 4.1 to illustrate its superior performance. The results show that the detector trained with our 3D pseudo-bounding

doi:10.6342/NTU202302077

Figure 4.2: **Qualitative results of 3D pseudo-bounding boxes.** The green shapes are the transformed templates, and the green boxes are the corresponding 3D pseudo-bounding boxes. The orange boxes are the ground truth 3D bounding boxes.

boxes is able to detect 3D car objects occluded by other objects. Furthermore, we showcase some of the transformed templates and their corresponding 3D pseudo-bounding boxes in Fig. 4.2 to verify the effectiveness of our method. We also present the qualitative comparison of 3D pseudo-bounding box results generated by our proposed DTMNet and Zakharov *et al*. [1]. The results are shown in Fig. 4.3. From this figure, we observe that although Zakharov *et al*. optimizes the 3D car shape to match it with the 2D mask and LiDAR point cloud, such a method is insufficient in matching cars located far away from the sensors due to the sparsity of the LiDAR point cloud. On the other hand, our DTMNet is able to precisely detect distant cars, which validates our claim that the edge and color supervision in our proposed approach is capable of extracting detailed shape information of distant cars, even in situations where the LiDAR point cloud is too sparse to be recognized. The visualizations demonstrate that our approach generates accurate and precise 3D pseudo-bounding boxes, which in turn leads to improved object detection performance.

doi:10.6342/NTU202302077

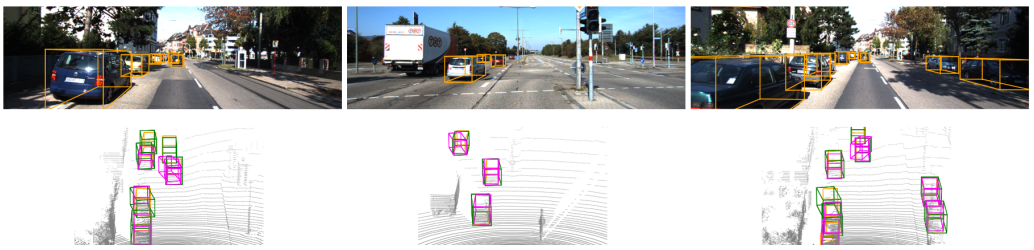


Figure 4.3: **Qualitative comparison of 3D pseudo-bounding boxes.** Each column in the figure represents the result of a single scene. The first row shows the images of each scene, while the second row displays the LiDAR point cloud of each scene. The 3D pseudo-bounding boxes predicted by our DTMNet are represented by the green boxes, while the purple boxes show the 3D pseudo-bounding boxes generated by Zakharov et al. [1]. The ground truth 3D bounding boxes are represented by the orange boxes.

## 4.3 Ablation Study

In this study, we first present an ablation analysis of the architecture of DTMNet in Table 4.2. Our objective is to investigate the contribution of each component to the overall performance of the proposed method. We start by establishing the baseline method A, which employs only the point matching and a mean car template to generate 3D pseudo-bounding boxes for training a PointRCNN [5] model without considering filtering out noise, edge supervision, or color supervision. We then filter out noise with segmentor $\theta^S$ (method B) to reduce noise and improve the overall performance. By adding back the deformable car template (method C), we obtain an improvement in average precision. In method D, we introduce edge supervision, which leads to a further improvement in performance. Finally, the inclusion of color supervision (method E) results in the best overall performance. Our ablation study shows that each component of the proposed method contributes to its effectiveness, and incorporating all components results in the highest performance.

As an essential component of our approach, we compare the methods of

doi:10.6342/NTU202302077

| Method | $\mathcal{L}_{point}$ | $\mathcal{L}_{seg}$ | deform. | $\mathcal{L}_{edge}$ | $\mathcal{L}_{color}$ | $AP_{BEV}/AP_{3D}(IoU=0.5)$ | | |
|---|---|---|---|---|---|---|---|---|
| | | | | | | **easy** | **moderate** | **hard** |
| A | ✓ | - | - | - | - | 81.53 / 63.89 | 72.38 / 57.38 | 63.92 / 50.17 |
| B | ✓ | ✓ | - | - | - | 87.27 / 73.61 | 81.44 / 68.53 | 72.19 / 60.37 |
| C | ✓ | ✓ | ✓ | - | - | 94.48 / 87.95 | 87.43 / 85.65 | 85.98 / 83.11 |
| D | ✓ | ✓ | ✓ | ✓ | - | 94.75 / 88.48 | 88.29 / 86.64 | 87.45 / 84.23 |
| E | ✓ | ✓ | ✓ | ✓ | ✓ | **95.71 / 94.84** | **88.37 / 87.60** | **87.51 / 86.15** |

Table 4.2: **Ablation study of our proposed framework.** In this table, *deform.* denotes the usage of deformable car template. All ablation studies use PointRCNN as the 3D object detector. Please refer to Sect. 4.3 for the details.

| Method | $AP_{BEV}/AP_{3D}(IoU=0.5)$ | | |
|---|---|---|---|
| | easy | moderate | hard |
| mask | 90.06 / 86.84 | 86.93 / 84.05 | 78.33 / 75.86 |
| edge | **95.71 / 94.84** | **88.37 / 87.60** | **87.51 / 86.15** |

Table 4.3: **Comparisons of edge matching and mask matching.** Note that we use PointRCNN as the 3D object detector.

edge supervision with mask supervision in Table 4.3 to verify the effectiveness of edge supervision. Specifically, the mask supervision method in Table 4.3 ensures matching with 2D car mask $m$ and the projection mask $mask(KT)$ of the transformed template $T$ instead of only matching between their edges as in our proposed edge supervision. The results demonstrate that the edge matching method outperforms the mask matching method, indicating that explicitly focusing on the edges is better suited for guiding our predictor $\theta^P$.

We also conduct an experiment using a contrastive loss proposed in NT-Xent [28] as our color supervision loss to compare with the Barlow Twins loss used in our approach. The results in Table 4.4 show that while our method is not

doi:10.6342/NTU202302077

| Method | $AP_{BEV}/AP_{3D}(IoU = 0.5)$ | | |
|---|---|---|---|
| | easy | moderate | hard |
| NT-Xent [28] | 95.33 / 94.59 | 87.88 / 87.05 | 86.21 / 84.36 |
| Barlow Twins [22] | **95.71 / 94.84** | **88.37 / 87.60** | **87.51 / 86.15** |

Table 4.4: **Comparisons of different contrastive losses.** Note that we use PointR-CNN as the 3D object detector.

restricted to using a specific contrastive loss, Barlow Twins outperforms NT-Xent by a small margin. Note that NT-Xent loss function requires a large number of negative samples, and therefore the performance gap in Table 4.4 is reasonable.

Finally, We perform qualitative ablation analysis. As depicted in Figure 4.4, we compare the template matching results with the mean car template $M_0$ and the deformable car template $M(\cdot)$ (as mentioned in Sect. 3.1 of our main paper). It is evident that the mean car template $M_0$ fails to match with the diverse range of real-world cars, thus affirming the justification for our approach. Our proposed weakly supervised deformable template matching technique effectively enables accurate matching with cars of various shapes. With the deformable template matching technique, we further compare the template matching results using edge and color supervision versus no edge and color supervision. As illustrated in Figure 4.5, The results show that integrating both edge and color supervision yields more robust and accurate matching outcomes when dealing with car occlusions.

## 4.4 Additional Experiment Results

We now provide more experiment results of our proposed method. we conduct additional experiments for *pedestrians* from KITTI, which consists of 2104 training samples and 2172 validation samples. Table 4.5 shows that our weakly supervised method was able to achieve comparable performance as the fully-supervised

doi:10.6342/NTU202302077


Figure 4.4: **Template Matching with Various Templates** Each row in the figure corresponds to a single car and depicts the outcomes obtained using different templates. The *Image* column presents the 2D bounding box of the car. The *Mean* column denotes the results using the mean car template $M_0$, while the *Deform* column displays the outcomes achieved using a deformable car template $M(\cdot)$. The transformed template predicted by our DTMNet is represented by the green points, whereas the red edges correspond to the edges of 2D car masks.

method did. This result also supports the effectiveness of our approach in handling deformable objects from noisy point cloud data with weak supervision.

| Method | Inference | Supervision | | $AP_{BEV}/AP_{3D}(IoU = 0.25)$ | | |
|---|---|---|---|---|---|---|
| | | 2D | 3D | easy | moderate | hard |
| PointRCNN | LiDAR | | ✓ | 67.41 / 67.40 | 60.40 / 60.26 | 54.51 / 54.45 |
| PointRCNN (Ours) | LiDAR | ✓ | | 63.63 / 63.51 | 57.18 / 57.01 | 51.13 / 50.90 |

Table 4.5: **Performance comparisons on KITTI validation set for pedestrians.** The *Inference* column refers to the input modality for inference, where *LiDAR* denotes LiDAR point clouds.

doi:10.6342/NTU202302077


Figure 4.5: **Comparison of Using Edge and Color Supervision** Each row in the figure depicts the outcomes obtained for a single car and compares the use of edge and color supervision versus no edge and color supervision. The *Image* column presents the 2D bounding box of the car. The green points depict the transformed template predicted by our DTMNet, while the red edges indicate the edges of the 2D car masks.

doi:10.6342/NTU202302077

# Chapter 5

# Conclusion

In this paper, we proposed a weakly supervised 3D object detection method via deformable template matching. Our DTMNet enables 3D object detection without 3D annotated data during training. The proposed method utilizes LiDAR point cloud, RGB image of the scene, and 2D car mask in the image to derive 3D pseudo-bounding boxes for training the 3D detector. Our DTMNet leverages edge supervision and color supervision to match the detailed geometry of the object and prevent matching errors with non-target objects, respectively. Experimental results demonstrated that our DTMNet outperformed state-of-the-art weakly supervised methods and achieved comparable performance to recent methods trained in fully supervised fashion. Thus, the use of our proposed DTMNet for 3D object detection can be sufficiently supported.



doi:10.6342/NTU202302077

# Reference


[1] S. Zakharov, W. Kehl, A. Bhargava, and A. Gaidon, "Autolabeling 3d objects with differentiable rendering of sdf shape priors," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. vi, 1, 2, 6, 7, 17, 18, 19, 20

[2] J. Leonard, J. How, S. Teller, M. Berger, S. Campbell, G. Fiore, L. Fletcher, E. Frazzoli, A. Huang, S. Karaman *et al.*, "A perception-driven autonomous urban vehicle," *Journal of Field Robotics*, 2008. 1

[3] H. Gao, B. Cheng, J. Wang, K. Li, J. Zhao, and D. Li, "Object classification using cnn-based fusion of vision and lidar in autonomous vehicle environment," *IEEE Transactions on Industrial Informatics*, 2018. 1

[4] W. Lee, N. Park, and W. Woo, "Depth-assisted real-time 3d object detection for augmented reality," in *Proceedings of the International Conference on Artificial Reality and Telexistence*, 2011. 1

[5] S. Shi, X. Wang, and H. Li, "Pointrcnn: 3d object proposal generation and detection from point cloud," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 4, 17, 20

[6] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "Pointpillars: Fast encoders for object detection from point clouds," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 4, 17




doi:10.6342/NTU202302077

[7] X. Ma, S. Liu, Z. Xia, H. Zhang, X. Zeng, and W. Ouyang, "Rethinking pseudo-lidar representation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 1, 4, 17

[8] S. Song, S. P. Lichtenberg, and J. Xiao, "Sun rgb-d: A rgb-d scene understanding benchmark suite," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 1

[9] H. Wang, Y. Cong, O. Litany, Y. Gao, and L. J. Guibas, "3dioumatch: Leveraging iou prediction for semi-supervised 3d object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 1, 5

[10] J. Wang, H. Gang, S. Ancha, Y.-T. Chen, and D. Held, "Semi-supervised 3d object detection via temporal graph neural networks," in *International Conference on 3D Vision (3DV)*, 2021. 1, 5

[11] J. Yin, J. Fang, D. Zhou, L. Zhang, C.-Z. Xu, J. Shen, and W. Wang, "Semi-supervised 3d object detection with proficient teachers," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022. 1, 5

[12] Z. Qin, J. Wang, and Y. Lu, "Weakly supervised 3d object detection from point clouds," in *Proceedings of the ACM Conference on Multimedia (MM)*, 2020. 1, 2, 6, 7, 17, 18

[13] Q. Meng, W. Wang, T. Zhou, J. Shen, L. Van Gool, and D. Dai, "Weakly supervised 3d object detection from lidar point cloud," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 1, 5

[14] Y. Wei, S. Su, J. Lu, and J. Zhou, "Fgr: Frustum-aware geometric reasoning for weakly supervised 3d vehicle detection," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2021. 1, 6, 7

**doi:10.6342/NTU202302077**

[15] R. McCraith, E. Insafutdinov, L. Neumann, and A. Vedaldi, "Lifting 2d object locations to 3d by discounting lidar outliers across objects and views," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2022. 1, 2, 6, 7, 10, 17, 18

[16] L. Peng, S. Yan, B. Wu, Z. Yang, X. He, and D. Cai, "Weakm3d: Towards weakly supervised monocular 3d object detection," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2022. 1, 2, 6, 7, 17, 18

[17] H. Wang, S. Sridhar, J. Huang, J. Valentin, S. Song, and L. J. Guibas, "Normalized object coordinate space for category-level 6d object pose and size estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2, 6

[18] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove, "Deepsdf: Learning continuous signed distance functions for shape representation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2, 6

[19] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, 1981. 6

[20] F. Lu, Z. Liu, X. Song, D. Zhou, W. Li, H. Miao, M. Liao, L. Zhang, B. Zhou, R. Yang *et al.*, "Permo: Perceiving more at once from a single image for autonomous driving," *arXiv preprint arXiv:2007.08116*, 2020. 8

[21] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics and intelligent laboratory systems*, 1987. 8

[22] J. Zbontar, L. Jing, I. Misra, Y. LeCun, and S. Deny, "Barlow twins: Self-supervised learning via redundancy reduction," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2021. 14, 22

**doi:10.6342/NTU202302077**

[23] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 16

[24] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3d object detection network for autonomous driving," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 16

[25] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," 2019. 16

[26] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 16

[27] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014. 17

[28] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2020. 21, 22

**doi:10.6342/NTU202302077**