

PR-1

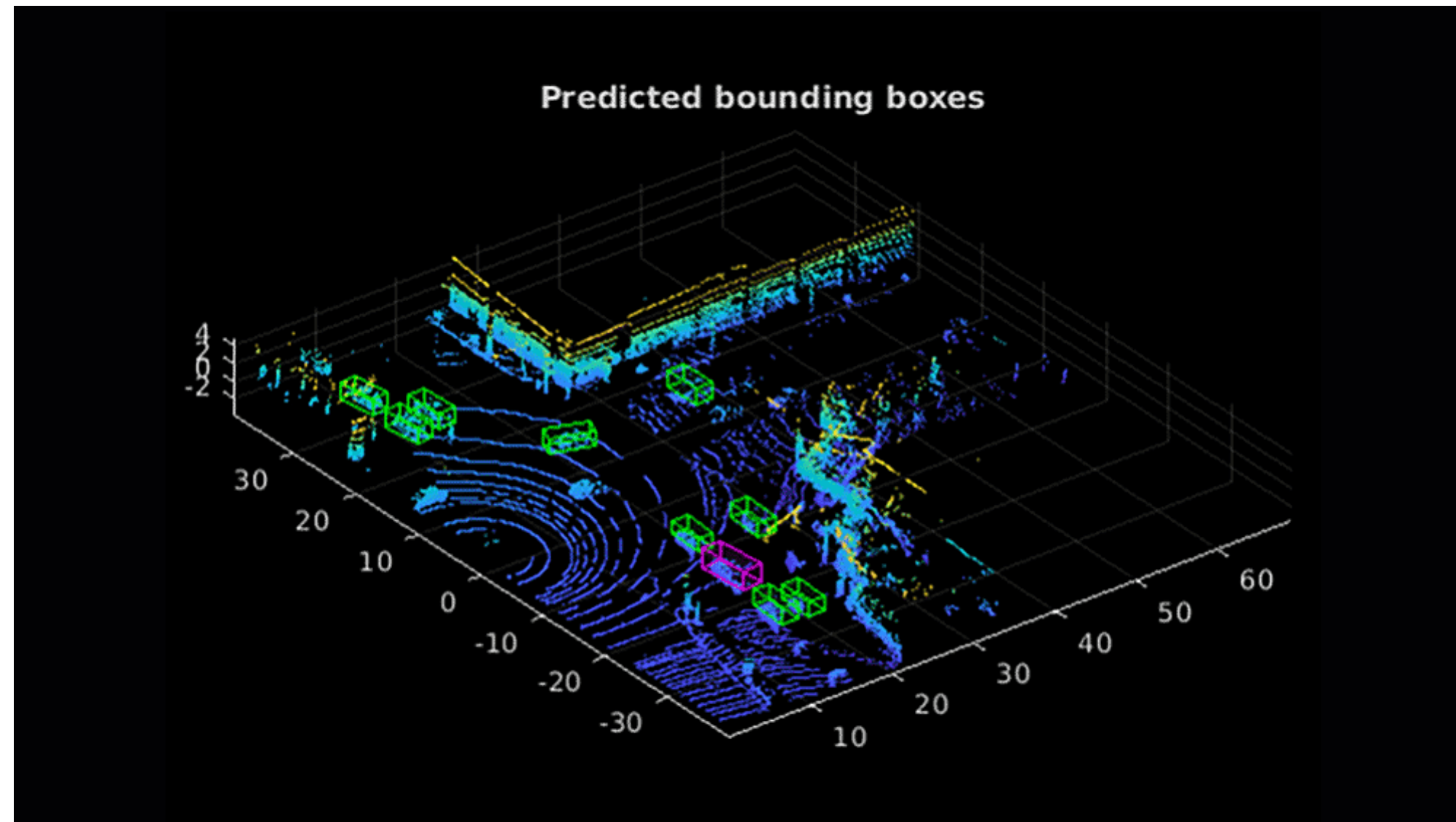
net. CenterPoint

Center-based 3D Object Detection and Tracking

박기완

Introduction

What is the Point Clouds



Point cloud is simply a set of data Points in a space

3D point cloud is represented by the XYZ coordinates of the points,
however, additional features such as surface normal, RGB values can also be used.

Introduction

The main Challenges of Deep learning on point Clouds

Irregularity (불규칙성) : Points는 어떤 영역에 있어 고르게 분포되어 있지 않다.

- 어느 공간은 Dense할 수 있고, 어느 공간은 Sparse할 수 있음

Unstructured (구조화되어 있지 않음) : Point Cloud data는 균일한 하나의 Grid로 이뤄져있지 않다.

- 2D image는 grid로 표현되는 반면에 point clouds는 그렇지 않음

Unorderliness (무질서) : LiDAR 같은 장비로 스캔하면 Points가 잡히는데, 순서 Index가 있는게 아니라 단순히 XYZ 좌표 정보만 가진다.

ABSTRACT

Lidar-based 3D Object Detection Method with Center Point tracking

[Exsting Papers]

3D Objects → Image-based Bounding Box 2D Detection 모방 → Point-clouds 3D Bounding Box

[Suggested Algorithm in this paper]

Center-based 3D object detection and tracking → Not using Bounding Box, but representing a target as one center-point

[Two Stages to implement this algorithm]

1. First-Stage :

Using "key-point Detector" → 객체에서 중심이 될 것 같은 특정 지점을 찾아 물체의 중심을 감지
→ 감지 후 중앙점으로 부터 회귀하여 해당 객체의 다른 속성을 추정 (3D Size, Direction, and Speed)

2. Second-Stage :

Using "Additional Point feature Extraction" → First-stage에서 얻은 객체의 속성에 대한 추정치를 세분화
→ Detection and Tracking 정확도를 개선

[Algorithm Performance]

LiDAR Dataset에서 대표적인 2개의 Dataset에서 가장 높은 성능 지표를 받아냄 (SOTA, the-state-of-the-art)

1. Waymo Open Dataset
2. nuScenes

Challenges in representing 3D object in Point Clouds

1. 일반적으로 3D 객체를 나타내는 데 사용되는 Point Cloud의 데이터가 희소한(Sparse) 경우가 많다.

- 3D 공간의 모든 지점에 대한 정보가 없을 수 있어 일부 영역이 누락되거나 불완전할 수 있음을 의미

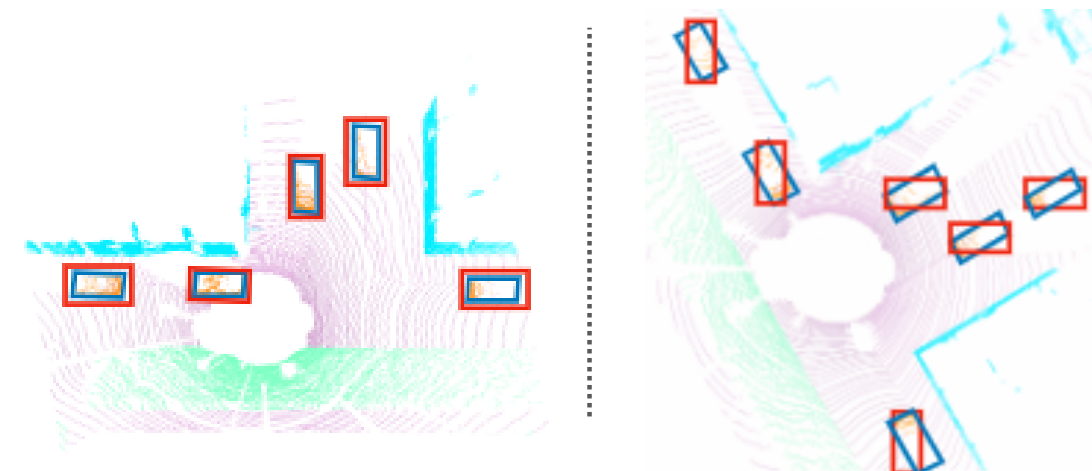


2. 3D Object를 Point Clouds에서 3D Box로 표현하는데 어려움이 많다.

- Box의 방향이 실제 물체의 방향과 일치하지 않을 수 있음을 의미

3. 3D Objects 마다의 Size, Shape 및 가로 세로 비율이 매우 다양하다.

- 버스와 리무진 같은 길이가 긴 차량이 있을 수 있고, 키가 크고 작은 보행자도 있듯이 다양함
- 즉, Object Detection을 위한 Single Template 또는 Anchor box는 모든 객체에 동작하지 않을 수 있음
- 그렇기 때문에 객체마다 다른 Template을 사용해야 할 수 있음 (Computation Increase)



Suggested Algorithm : CenterPoint

At the First-Stage

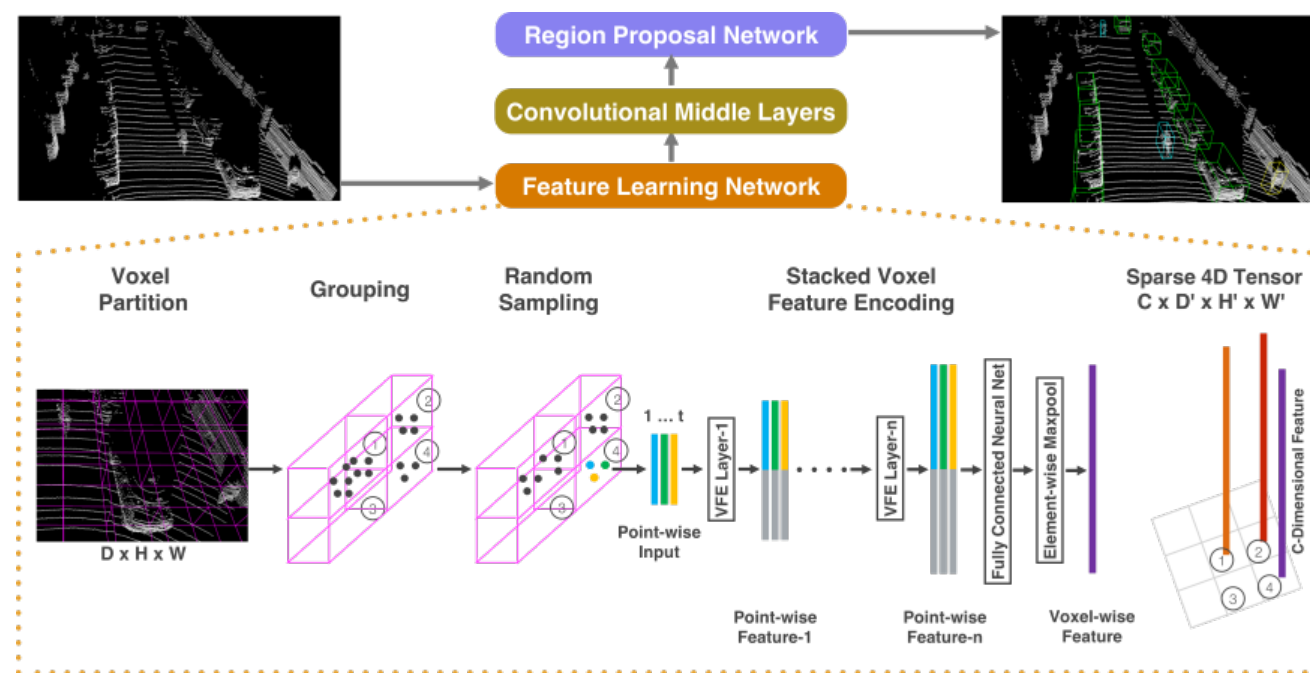
1. 3D Backbones : VoxelNet and Point-Pillars → Make a representation of Point Clouds
2. Using representation of Point Clouds → Make a overhead Map-View
3. Using "Key-point Detector" → Detect the Center Point of the objects from point clouds
4. Using the center point → Find all properties of the detected object

the Second-Stage

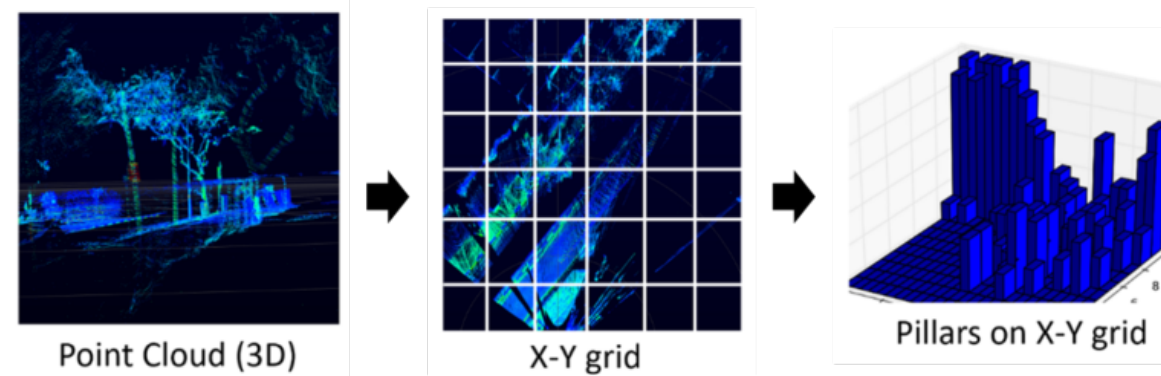
1. Extract the features of the center points from each face of the 3D bounding box of the estimated object.
 - a. (except for the top and bottom)
2. 객체 위치를 조정하여 전처리 과정에서 손실된 Geometrical Information을 복구하고 detection 성능을 개선

the First-Stage

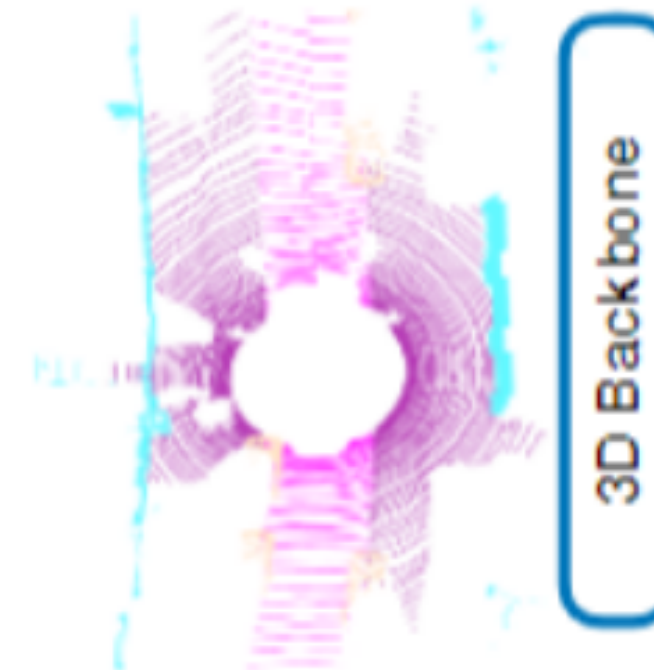
1. 3D Backbones : VoxelNet and Point-Pillars → Make a representation of Point Clouds (Feature Map)



ref. VoxelNet



ref. PointPillars



(a) Point. cloud

the First-Stage

2. Using representation of Point Clouds → Make a overhead Map-View (BEV)

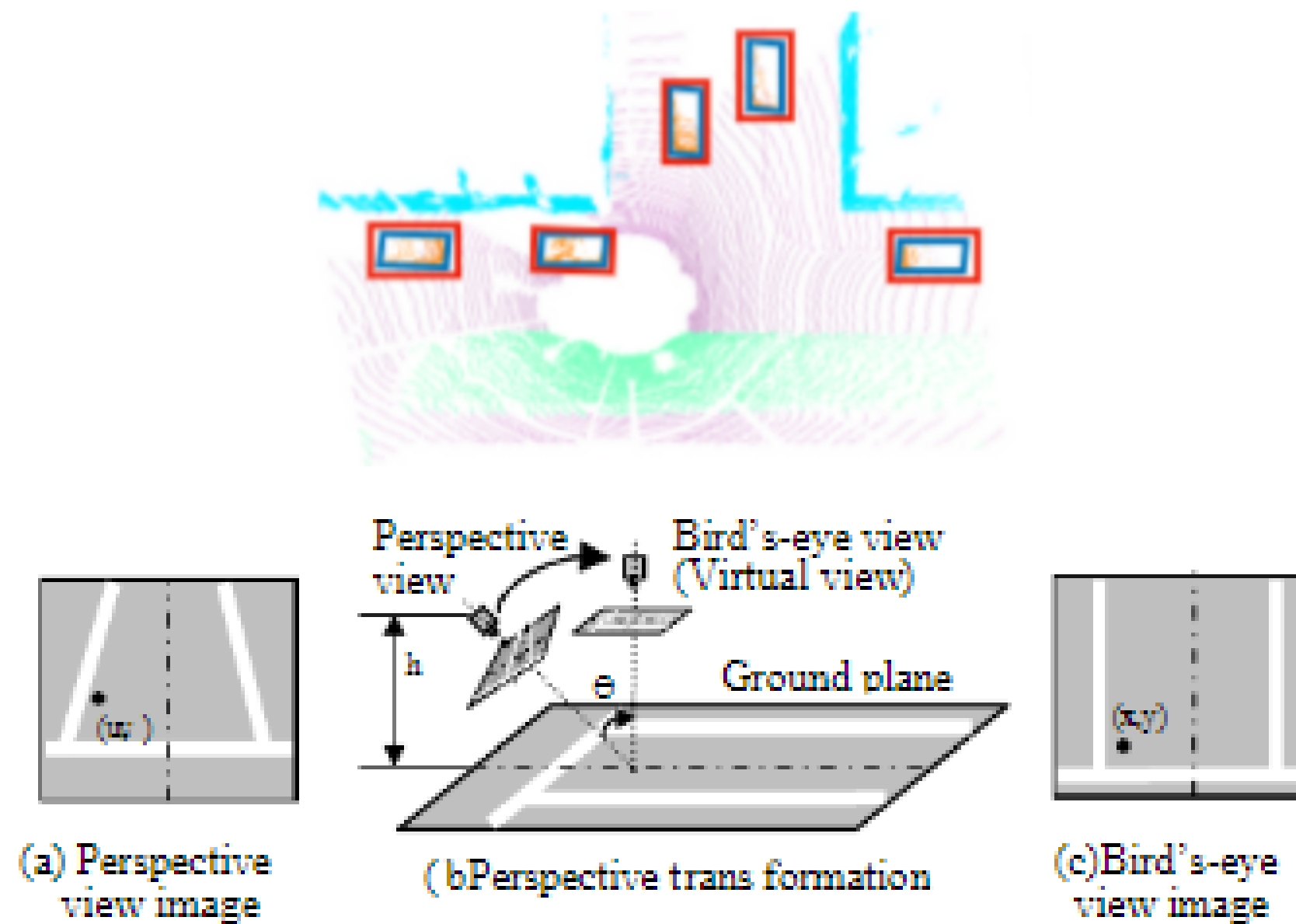
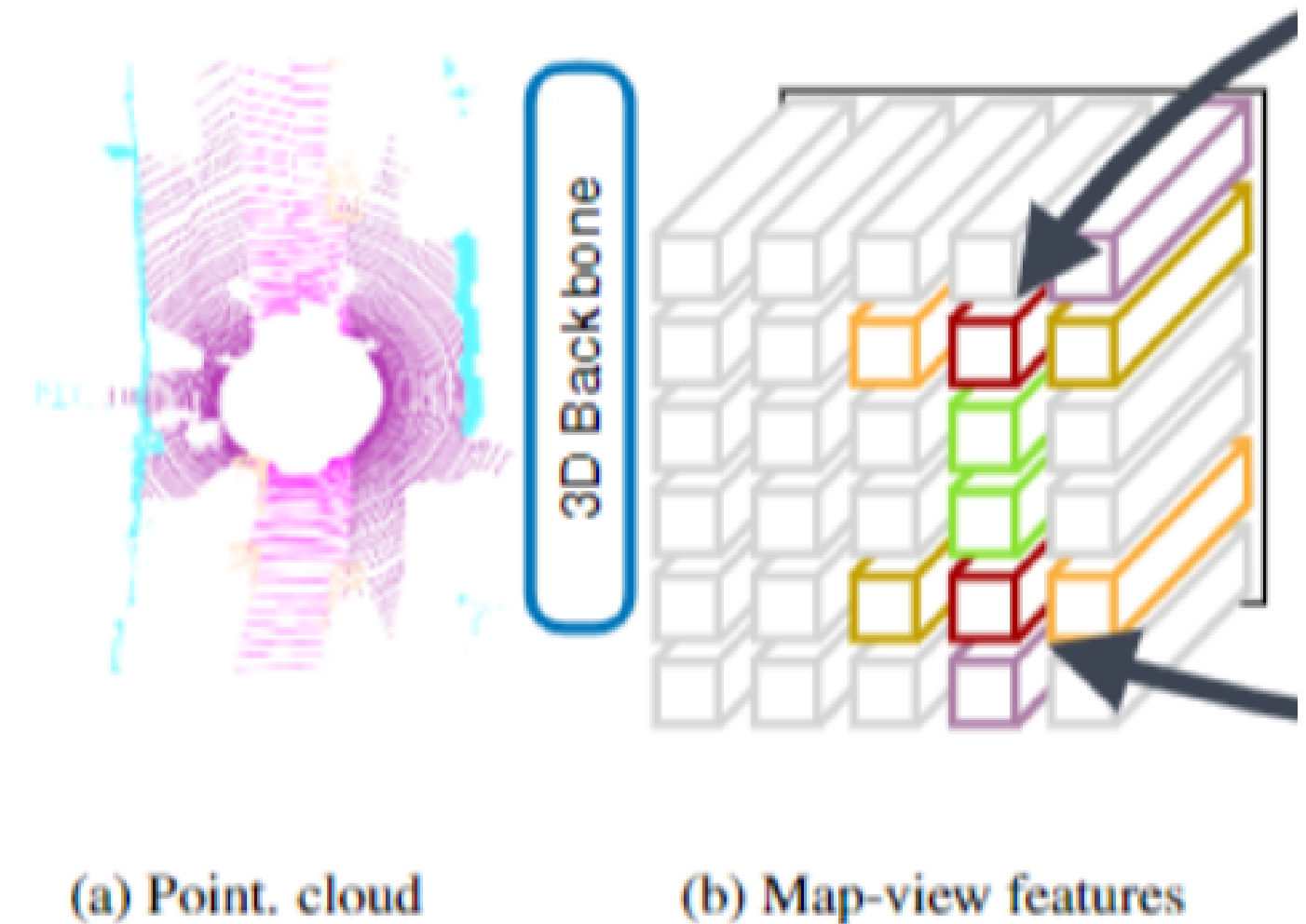


Fig. 1. Illustration of perspective transformation in a parking lot scene



ref. Low-cost Implementation of Bird's-eye View System for Camera-on-vehicle

the First-Stage

2. Using representation of Point Clouds → Make a overhead Map-View (BEV)

a. Center Heatmap Head : 중앙 지점으로부터 Gaussian Dist.를 사용하여 Heatmap을 생성한다.

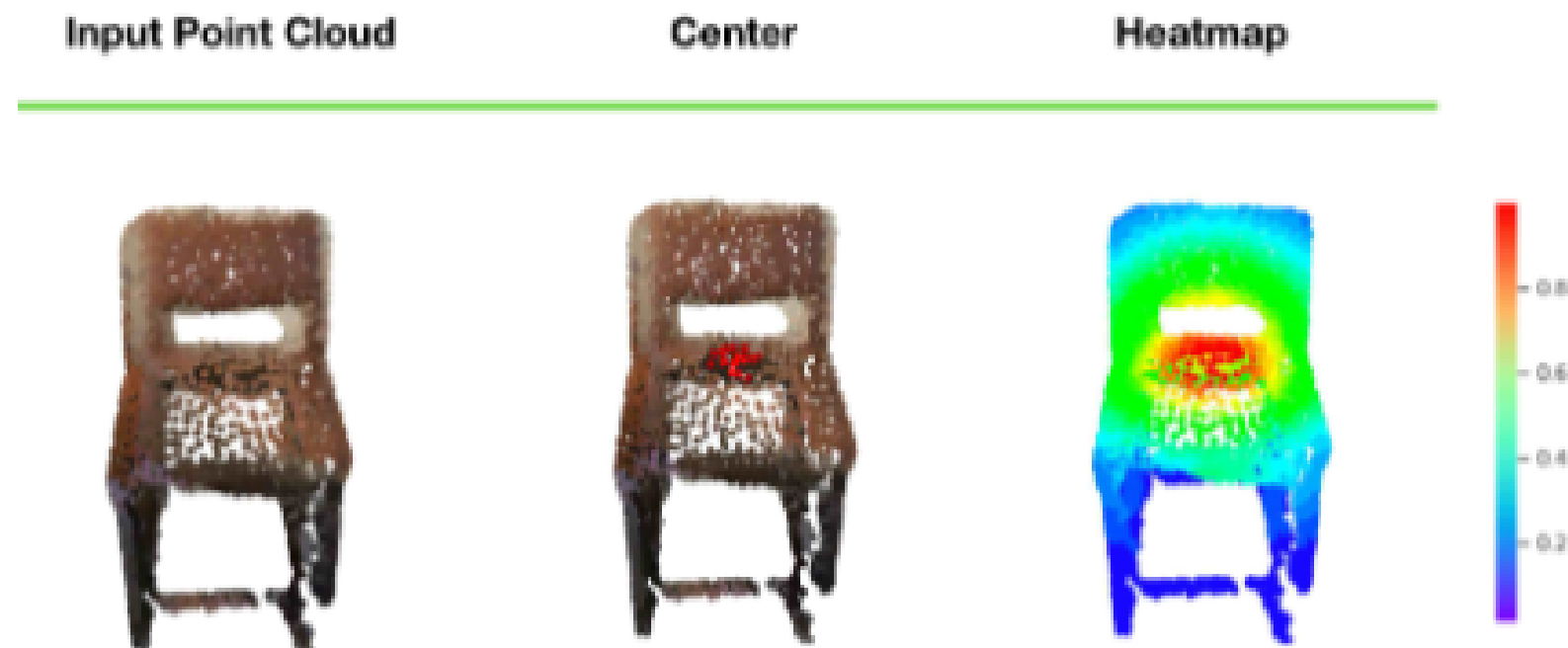


Fig. 7: An example of ground-truth center heatmap

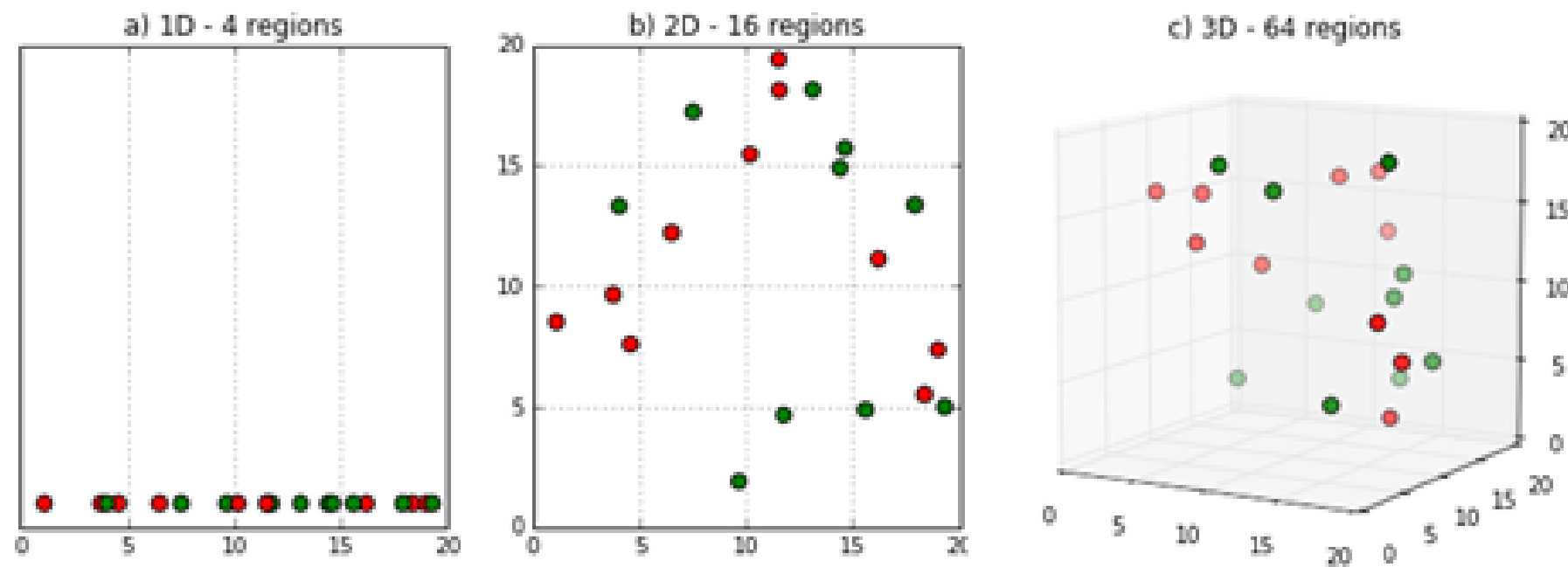
1. 감지된 물체의 중앙 위치에서 Heat map Peak를 생성하여 클래스 수 만큼 K 개의 채널을 가진 Heatmap, Y를 생성
2. Object Detection 작업에서 클래스 간 불균형 문제를 해결하기 위해 Focal Loss를 사용
 - Background 샘플의 수가 관심 대상 샘플의 수보다 훨씬 크기 때문

the First-Stage

2. Using representation of Point Clouds → Make a overhead Map-View (BEV)

b. Curse of Dimensionality (차원의 저주) / Dimension Reduction (차원 축소)

Map-view의 Feature Map에서 객체가 나타내는 Center-Points는 2D로 Projection할 때, 이미지 뷰에서 객체 중심이 원래 보다 훨씬 가까워지는 문제가 생길 수 있다.

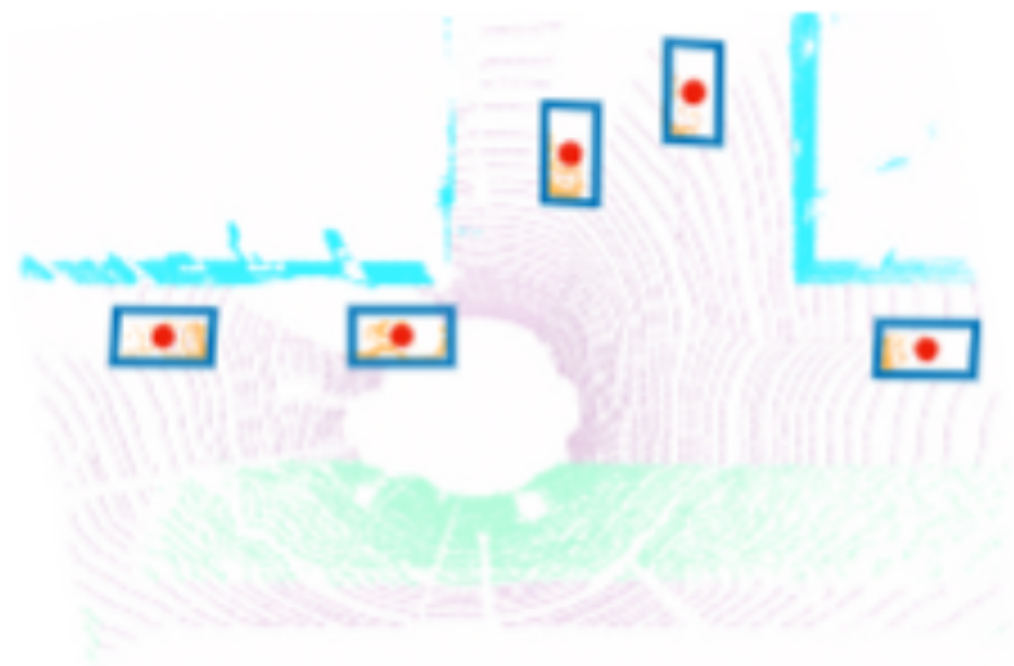


이를 해결하기 위해 Groud Truch Object의 중심에 렌더링되는 Gaussian Peak를 확대하여 Positive Heatmap에 주목하도록 알고리즘을 설계했다.

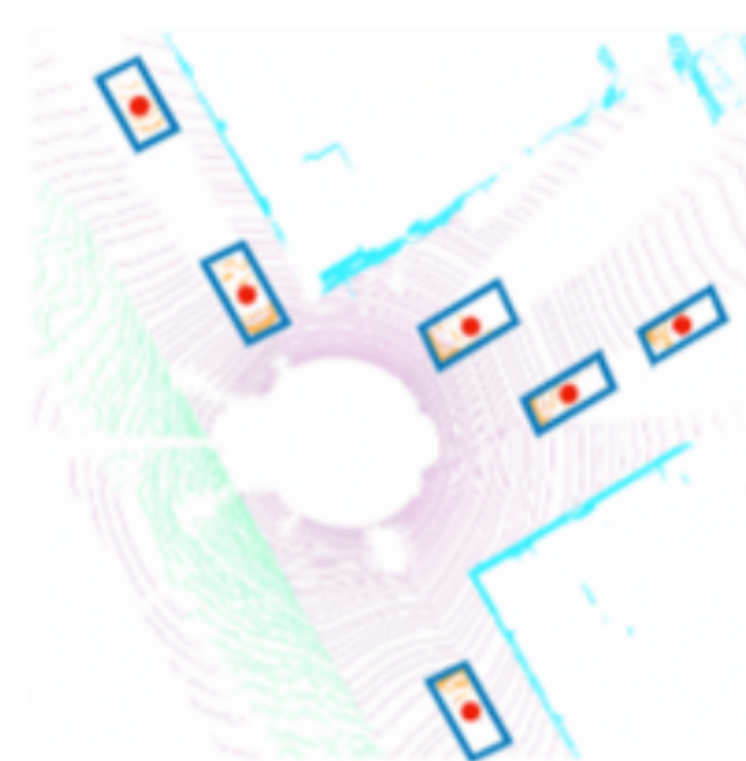
- Gaussian Radius, $\sigma = \max(f(wl), \tau)$,로 설정
 - 최소 Gaussian Radius는 $\tau = 2$ 이 가능한 구간
 - f 는 2D 모델의 ConerNet에서 정의된 반지름 함수

the First-Stage

3. Using "Key-point Detector" → Detect the Center Point of the objects from point clouds
4. Using the center point → Find all properties of the detected object



b) Center-based $t=1$



d) Center-based $t=2$

Detecting Center Points and predict the bounding boxes and all properties

the Second-Stage

2. 객체 위치를 조정하여 전처리 과정에서 손실된 Geometrical Information을 복구하고 detection 성능을 개선

⇒ predicted Bounding Box에서 Additional Point Feature Extraction

⇒ 이를 사용하여 3D Object Detection and Tracking을 세분화한다.

⇒ MLP는 Confidence(신뢰도) score and Box refinement(개선)을 예측

⇒ Detection and Tracking의 정확도를 개선한다.

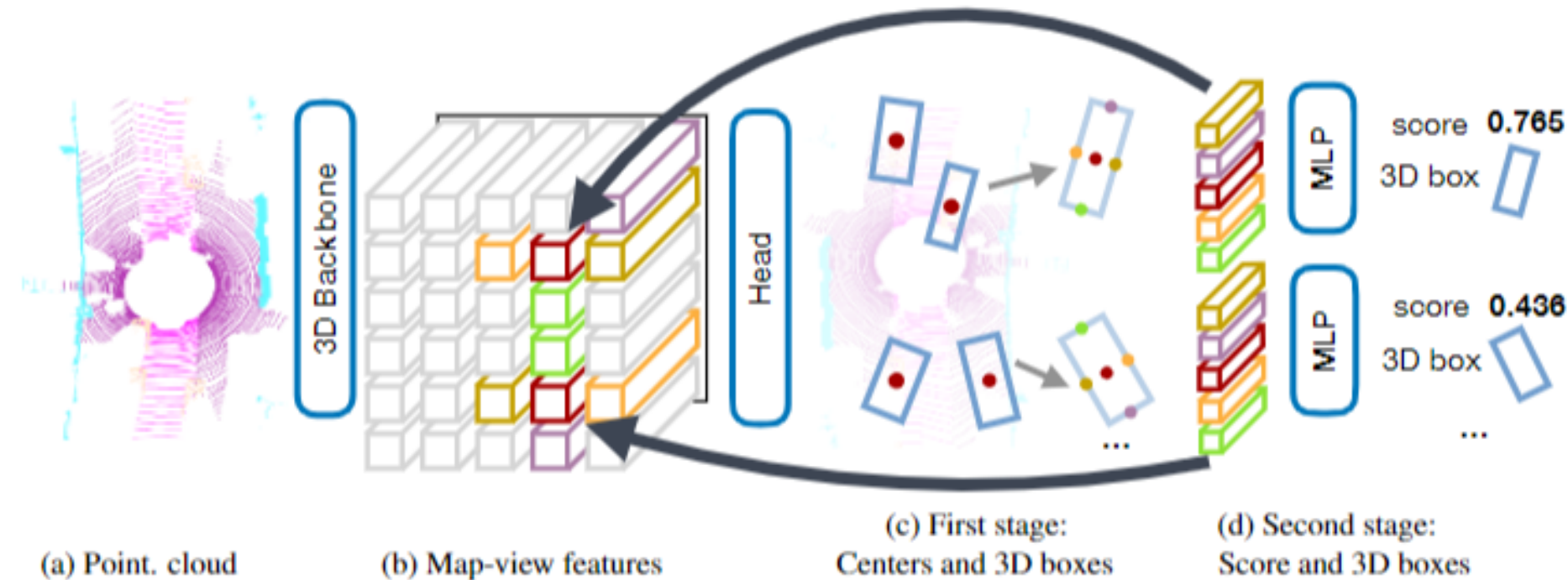


Figure 2: Overview of our CenterPoint framework. We rely on a standard 3D backbone that extracts map-view feature representation from Lidar point-clouds. Then, a 2D CNN architecture detection head finds object centers and regress to full 3D bounding boxes using center features. This box prediction is used to extract point features at the 3D centers of each face of the estimated 3D bounding box, which are passed into MLP to predict an IoU-guided confidence score and box regression refinement. Best viewed in color.

the Second-Stage

2. 객체 위치를 조정하여 전처리 과정에서 손실된 Geometrical Information을 복구하고 detection 성능을 개선

a. Regression heads : Input Point-Clouds 데이터를 기반으로 Object를 정확하게 감지하는 데 사용하는 객체 속성을 저장한다.

- 이때 말하는 객체 속성은 3D 크기, 방향 및 속도와 같은 속성들을 말함

sub-voxel location refinement : $o \in \mathbb{R}^2$

Backbone Network의 Voxelization 및 Striding 중 발생하는 양자화 오류를 줄이는 데 효율적

height-above-ground : $h_g \in \mathbb{R}$

3D 공간에서 물체의 고도에 대한 정보를 제공하여 객체를 3D로 위치시키는 데 도움

Map-View 투영(projection)으로 누락된 Height 정보를 추가

the 3D size : $s \in \mathbb{R}^3$

3D 공간에서의 Object 크기에 대한 정보를 제공한다.

물체를 정확하게 감지하고 추적하는 데 중요한 특성이라고 한다.

a yaw rotation angle : $(\sin(\alpha), \cos(\alpha)) \in \mathbb{R}^2$

3D 공간에서의 물체 방향에 대한 정보를 제공한다.

Training 중에는 Object의 Ground Truth Center에만 L_1 regression loss를 사용한다.

key advantages of Center-Based 3D Detection

1. Bounding Box와 달리 Point에는 Intrinsic orientation이 없다.

a. 즉, Point의 방향을 고려하지 않고 물체를 쉽게 감지할 수 있음

2. Center-Based Representation은 Object Tracking과 같은 작업을 Simplify할 수 있다.

a. 물체를 Point로 표현하면, 시간과 공간에서 물체의 경로(Tracklets)를 쉽게 추적할 수 있음

i. 단순히 경로를 Point의 Sequence로 표현하면 되기 때문

b. 영상 속 연속 프레임 사이의 객체 속도를 예측하므로 객체의 움직임을 쉽게 추적할 수 있음

3. Center Points 기반의 Feature Extraction을 통해 이전 기법에서 사용한 것보다 시스템의 성능을 높였다.

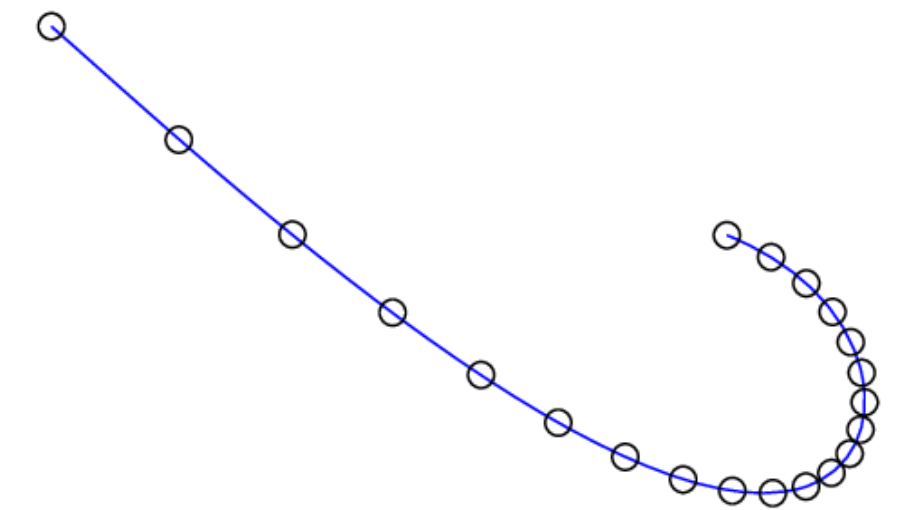
a. 데이터의 복잡성을 줄이고, 더 간결한 표현을 사용하여 학습 과정을 최적화

b. 이전 기법은 모델 학습에 리소스가 많이 소요됨 & 여전히 3D 표현이 복잡한 형태

→ 3D CNN : 2D 이미지 처리에 사용되는 CNN을 3D에 접목시켰으나, 계산 비용이 매우 높아 효율 X (computation cost ↑)

→ Voxel-Based : 3D 데이터의 구조를 간단하게 표현할 수 있었으나, 데이터가 커질 수록 성능 하락을 보임

→ Projection-Based : 3D 데이터를 2D로 투영하여 2D CNN으로 처리할 수 있으나, 정보 손실 등의 이유로 학습이 원활하지 않음



A Sequence of points

Main Results

1. 기존의 Bounding Box 표현에서 Center-Point 표현으로 전환할 경우 다양한 모델에서 Detection 성능이 향상되었다.

a. NeurlPS 2020 NuScenes 3D Detection Challenge : 3등

b. Waymo vehicle and pedestrian Detection : 기존 SOTA 대비 50% 성능 향상

Difficulty	Method	Vehicle		Pedestrian	
		mAP	mAPH	mAP	mAPH
Level 1	StarNet [36]	61.5	61.0	67.8	59.9
	PointPillars [28]	63.3	62.8	62.1	50.2
	PPBA [36]	67.5	67.0	69.7	61.7
	RCD [5]	72.0	71.6	-	-
	Ours	80.2	79.7	78.3	72.1
Level 2	StarNet [36]	54.9	54.5	61.1	54.0
	PointPillars [28]	55.6	55.1	55.9	45.1
	PPBA [36]	59.6	59.1	63.0	55.8
	RCD [5]	65.1	64.7	-	-
	Ours	72.2	71.8	72.2	66.4

Table 1: State-of-the-art comparisons for 3D detection on Waymo test set. We show the mAP and mAPH for both level 1 and level 2 benchmarks.

Method	mAP↑	NDS↑	PKL↓
WYSIWYG [23]	35.0	41.9	1.14
PointPillars [28]	40.1	55.0	1.00
CVCNet [7]	55.3	64.4	0.92
PointPainting [49]	46.4	58.1	0.89
PMPNet [62]	45.4	53.1	0.81
SSN [68]	46.3	56.9	0.77
CBGS [67]	52.8	63.3	0.77
Ours	58.0	65.5	0.69

Table 2: State-of-the-art comparisons for 3D detection on nuScenes test set. We show the nuScenes detection score (NDS), and mean Average Precision (mAP).

Main Results

2. 여러 물체들이 존재하는 경우에도 좋은 성능 지표를 보여준다.

a. MOTA : Multi-Object Tracking Accuracy / 시스템이 실시간으로 움직이는 객체를 얼마나 정확하게 추적할 수 있는 지를 측정하는 Metric

b. AMOTA : Average Multi-Object Tracking Accuracy

i. FP : 장면에 물체가 없지만 감지 알고리즘이 물체를 Detection하는 경우 (알고리즘은 존재하지 않는 Object를 Detectiton)

ii. FN : 장면에 물체가 있지만 감지 알고리즘이 이를 Detection하지 못하는 경우 (알고리즘이 Object를 놓치는 경우)

iii. IDS : 추적 알고리즘이 연속 프레임에서 Object에 다른 ID를 할당하는 경우 (서로 다른 두 Objects를 동일하다고 판단하거나 그 반대)

Difficulty	Method	MOTA↑		MOTP↓	
		Vehicle	Ped.	Vechile	Ped.
Level 1	AB3D [48, 53]	42.5	38.9	18.6	34.0
	Ours	62.6	58.3	16.3	31.1
Level 2	AB3D [48, 53]	40.1	37.7	18.6	34.0
	Ours	59.4	56.6	16.4	31.2

Table 3: State-of-the-art comparisons for 3D tracking on Waymo test set. We show MOTA, and MOTP. ↑ is for higher better and ↓ is for lower better.

Method	AMOTA↑	FP↓	FN↓	IDS↓
AB3D [53]	15.1	15088	75730	9027
Chiu et al. [10]	55.0	17533	33216	950
Ours	63.8	18612	22928	760

Table 4: State-of-the-art comparisons for 3D tracking on nuScenes test set. We show AMOTA, the number of false positives (FP), false negatives (FN), id switches (IDS), and per-category AMOTA. ↑ is for higher better and ↓ is for lower better.

Main Results

3. Object의 크기에 따라 성능 비교도 진행했고, 역시 Center-Based의 성능이 더 좋게 나왔다.

Encoder	Method	mAP	NDS
VoxelNet	Anchor-based	52.6	63.0
	Center-based	56.4	64.8
PointPillars	Anchor-based	46.2	59.1
	Center-based	50.3	60.2

Table 6: Comparison between anchor-based and center-based methods for 3D detection on nuScenes validation. We show mean average precision (mAP) and nuScenes detection score (NDS).

Main Results

4. 마지막으로 Waymo Validation 데이터셋을 이용한 3D Detection 알고리즘 중 가장 좋은 성능을 보여준다.

a. 특히 Level 2 pedestrian detection의 경우 기존 보다 +18.6 mAPH 높은 Detection 성능을 가짐

Difficulty	Method	Vehicle		Pedestrian	
		mAP	mAPH	mAP	mAPH
Level 1	DOPS [35]	56.4	-	-	-
	PointPillars [28]	56.6	-	59.3	-
	PPBA [36]	62.4	-	66.0	-
	MVF [65]	62.9	-	65.3	-
	Huang et al. [24]	63.6	-	-	-
	AFDet [14]	63.7	-	-	-
	CVCNet [7]	65.2	-	-	-
	Pillar-OD [52]	69.8	-	72.5	-
	PV-RCNN [44]	74.4	73.8	61.4	53.4
	CenterPoint-Pillar(ours)	76.1	75.5	76.1	65.1
	CenterPoint-Voxel(ours)	76.7	76.2	79.0	72.9
Level 2	PV-RCNN [44]	65.4	64.8	53.9	46.7
	CenterPoint-Pillar(ours)	68.0	67.5	68.1	57.9
	CenterPoint-Voxel(ours)	68.8	68.3	71.0	65.3

Table 11: State-of-the-art comparisons for 3D detection on Waymo validation.

Conclusion

Contributions :

1. Robust and Efficient Object Detection : Outperforming previous traditional approaches.
2. Simplified Representation & Easier Object Tracking: 객체를 점으로 표현함으로써 객체 인식과 추적 알고리즘을 단순화했다.

Limitations :

1. Sparse-point clouds data processing : 기본적으로 데이터의 밀도가 낮아지면 Center-Point를 찾기 힘들어 성능 저하가 발생할 수 있다.
2. Small Object Detection : 마찬가지로 작은 물체 역시 적은 수의 포인트 데이터로 구성되어 있어, Center-Point를 찾는 것이 어렵다.
3. Accurate Bounding Box generation
4. Real-time Processing speed

Improvements :

1. Data Augmentation techniques : Sparse-point clouds에 대해서도 Robust하게 설계한다.
2. Efficient Computation and Hardware Optimization : 실시간 처리 속도를 향상시켜 다양한 모델에 사용할 수 있도록 개선한다.
3. Fusion with Semantic segmentation or Depth Estimation, or something : 다른 정보를 추가로 줘서 성능을 높인다.

Thank you

Reference

- [1] Bello, S.A., Yu, S., & Wang, C. (2020). Deep learning on 3D point clouds. Fujian Key Laboratory of Sensing and Computing for Smart City, Xiamen University, China; Kano University of Science and Technology, Nigeria. Retrieved January 17, 2020.
- [2] Yin, T., Zhou, X., & Krähenbühl, P. (2021). Center-based 3D Object Detection and Tracking. UT Austin. Retrieved January 6, 2021.
- [3] <https://deepai.org/machine-learning-glossary-and-terms/curse-of-dimensionality>