Center-based 3D Object Detection and Tracking

Tianwei Yin, et al. 2021, CVPR

Introduction

- 3D detection on point-clouds offers a series of interesting challenges
  1. First, point-clouds are sparse, and most parts of 3D objects are without measurements
  2. Second, the resulting output is a three dimensional box that is often not well aligned with any global coordinate frame
  3. Third, 3D objects come in a wide range of sizes, shapes, and aspect ratios

- These marked differences between 2D and 3D detection made a transfer of ideas between the two domains harder

- We argue that the main underlying challenge in linking up the 2D and 3D domains lies in this representation of objects

- We show how representing objects as points greatly simplifies 3D recognition

Introduction

- Our two-stage 3D detector, CenterPoint, finds centers of objects and their properties using a keypoint detector, a second-stage refines all estimates

- The center-based representation has several key advantages:
    1. Unlike bounding boxes, points have no intrinsic orientation. This dramatically reduces the object detector's search space
    2. A center-based representation simplifies downstream tasks such as tracking
    3. Point-based feature extraction enables us to design an effective two-stage refinement module that is much faster than the previous approaches

Preliminaries – 2D CenterNet

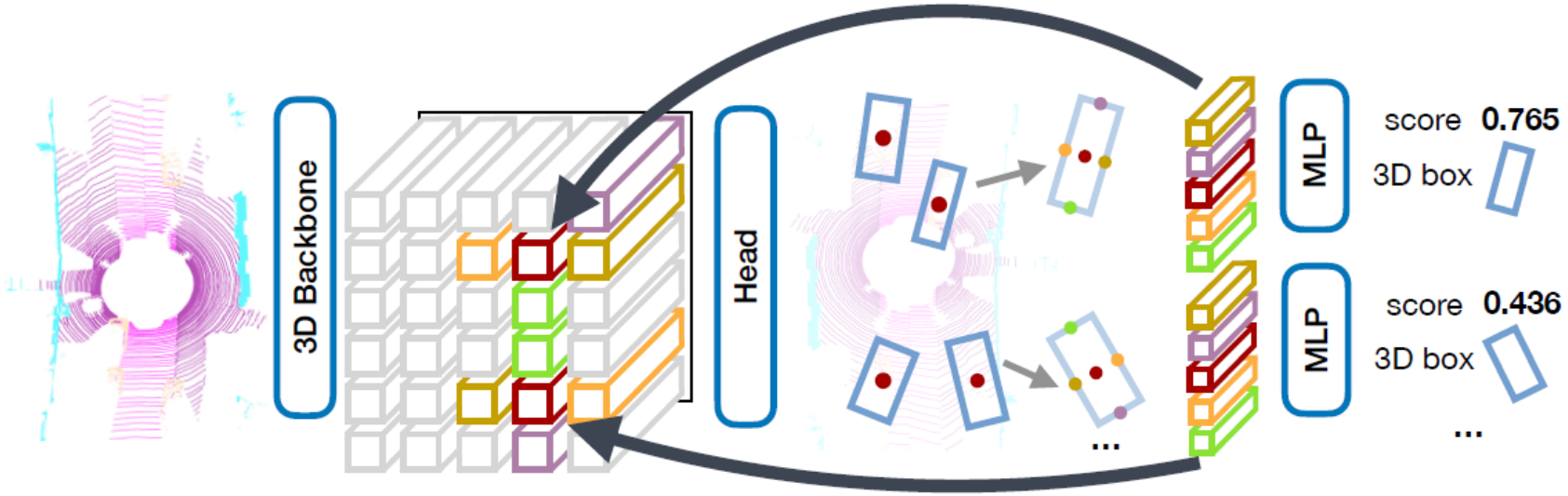
- 2D CenterNet rephrases object detection as keypoint estimation
- It takes an input image and predicts a $w \times h$ heatmap $\hat{Y} \in [0,1]^{w \times h \times \bar{K}}$ (K classes)
  - Each local maximum (i.e., pixels whose value is greater than its eight neighbors) in the output heatmap corresponds to the center of a detected object
- To retrieve a 2D box, CenterNet regresses to a size map $\hat{S} \in \mathbb{R}^{w \times h \times 2}$ shared between all categories

Preliminaries – 3D Detection

- Let $\mathcal{P} = \{(x, y, z, r)_i\}$ be an orderless point cloud of 3D location (x, y, z) and reflectance r measurements -> 3D object detection aims to predict a set of 3D object bounding boxes $\mathcal{B} = \{b_k\}$, $b = (u, v, d, w, l, h, \alpha)$
- Modern 3D object detectors uses a 3D encoder that quantizes the point-cloud into regular bins
- The output of a backbone network is a map-view feature-map $\mathbf{M} \in \mathbb{R}^{W \times L \times F}$ of width W and length L with F channels
- With a map-view feature map M, a detection head then produces object detections from some predefined anchor boxes

CenterPoint

- **Center heatmap head**
  - Goal : Produce a heatmap peak at the center location of any detected object
  - This head produces a **K-channel heatmap $\hat{Y}$**, one channel for each of K classes $\hat{Y} \in [0,1]^{w \times h \times \bar{K}}$
  - It targets a 2D Gaussian produced by the projection of 3D centers of annotated bounding boxes into the map-view



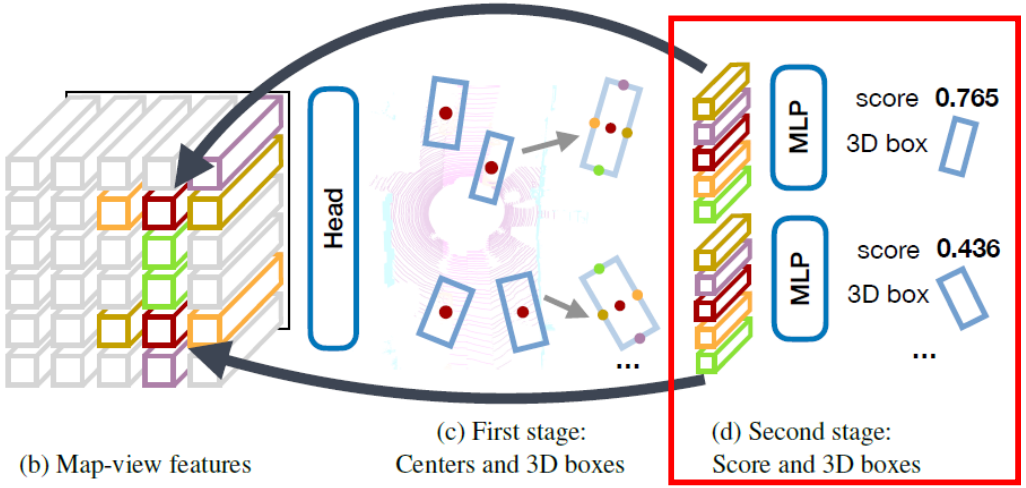
CenterPoint

- **Regression heads**
  - We store several object properties at center-features of objects
    1. Sub-voxel location refinement
    2. Height-above-ground
    3. The 3D size
    4. A yaw rotation angle
  - Combined with box size, these regression heads provide the full state information of the 3D bounding box

- **Velocity head and tracking**
  - To track objects through time, we learn to predict a two-dimensional velocity estimation $\mathbf{v} \in \mathbb{R}^2$ for each detected object as an additional regression output

Two-Stage CenterNet

- The second stage extracts additional point-features from the output of the backbone
- We extract one point-feature from the 3D center of each face of the predicted bounding box
  - Only consider the four outward-facing box-faces together with the predicted object center
- We concatenate the extracted point-features and pass them through an MLP
- The second stage predicts a class-agnostic confidence score and box refinement on top of one-stage CenterPoint's prediction results
  - CenterPoint and computes the final confidence score as the geometric average of the two scores $\hat{Q}_t = \sqrt{\hat{Y}_t * \hat{I}_t}$
  - For box regression, the model predicts a refinement on top of first stage proposals



Experiments – Main Results

- Evaluate CenterPoint on Waymo Open Dataset and nuScenes dataset
- **3D Detection**
  - Our model displays a consistent performance improvement over all categories

Difficulty	Method	Vehicle		Pedestrian	
		mAP	mAPH	mAP	mAPH
Level 1	StarNet [34]	61.5	61.0	67.8	59.9
	PointPillars [27]	63.3	62.8	62.1	50.2
	PPBA [34]	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 [34]	54.9	54.5	61.1	54.0
	PointPillars [27]	55.6	55.1	55.9	45.1
	PPBA [34]	59.6	59.1	63.0	55.8
	RCD [5]	65.1	64.7	_	_
	Ours	**72.2**	**71.8**	**72.2**	**66.4**

<3D detection on Waymo dataset>

Method	mAP↑	NDS↑	PKL↓
WYSIWYG [22]	35.0	41.9	1.14
PointPillars [27]	40.1	55.0	1.00
CVCNet [7]	55.3	64.4	0.92
PointPainting [47]	46.4	58.1	0.89
PMPNet [60]	45.4	53.1	0.81
SSN [66]	46.3	56.9	0.77
CBGS [65]	52.8	63.3	0.77
Ours	**58.0**	**65.5**	**0.69**

<3D detection on NuSenes dataset>

Experiments – Main Results

- **3D Tracking**
  - Our velocity-based closest distance matching significantly outperforms the official tracking baseline in the Waymo paper, which uses a Kalman-filter based tracker
  - On nuScenes, our framework outperforms the last challenge winner Chiu et al. by 8.8 AMOTA

Difficulty	Method	MOTA↑		MOTP↓	
		Vehicle	Ped.	Vechile	Ped.
Level 1	AB3D [46, 51]	42.5	38.9	18.6	34.0
	Ours	**62.6**	**58.3**	**16.3**	**31.1**
Level 2	AB3D [46, 51]	40.1	37.7	18.6	34.0
	Ours	**59.4**	**56.6**	**16.4**	**31.2**

<3D tracking on Waymo dataset>

Method	AMOTA↑	FP↓	FN↓	IDS↓
AB3D [51]	15.1	**15088**	75730	9027
Chiu et al. [10]	55.0	17533	33216	950
Ours	**63.8**	18612	**22928**	**760**

<3D tracking on NuSenes dataset>

Experiments – Ablation studies

- **Center-based vs Anchor-based**

Encoder	Method	Vehicle	Pedestrain	mAPH
VoxelNet	Anchor-based	66.1	54.4	60.3
	Center-based	**66.5**	**62.7**	**64.6**
PointPillars	Anchor-based	64.1	50.8	57.5
	Center-based	**66.5**	**57.4**	**62.0**

<Comparison between anchor-based and center-based methods on Waymo dataset>

Encoder	Method	mAP	NDS
VoxelNet	Anchor-based	52.6	63.0
	Grid Point-based	53.1	62.8
	Center-based	**56.4**	**64.8**
PointPillars	Anchor-based	46.2	59.1
	Grid Point-based	47.1	58.8
	Center-based	**50.3**	**60.2**

<Comparison between anchor-based, Grid Point-based and center-based methods on NuScenes dataset>

	Vehicle			Pedestrian		
Rel. yaw	0°-15°	15°-30°	30°-45°	0°-15°	15°-30°	30°-45°
# annot.	81.4%	10.5%	8.1%	71.4%	15.8%	12.8%
Anchor-based	67.1	**47.7**	45.4	55.9	32.0	26.5
Center-based	**67.8**	46.4	**51.6**	**64.0**	**42.1**	**35.7**

<Comparison between anchor-based and center based methods>

Method	Vehicle			Pedestrian		
	small	medium	large	small	medium	large
Anchor-based	58.5	**72.8**	64.4	29.6	60.2	60.1
Center-based	**59.0**	72.4	**65.4**	**38.5**	**69.5**	**69.0**

<Effects of object size for the performance of anchor-based and center-based methods>

Experiments – Ablation studies

- **One-stage vs Two-stage**
  - Two-stage refinement with multiple center features gives a large accuracy boost to both 3D encoders with small overheads (6ms-7ms)
  - Center-based feature aggregation achieved comparable performance but is faster and simpler than ROIAlign

Encoder	Method	Vehicle	Ped.	$T_{proposal}$	T_{refine}
VoxelNet	First Stage	66.5	62.7	71ms	-
	+ Box Center	68.0	64.9	71ms	5ms
	+ Surface Center	**68.3**	65.3	71ms	6ms
	Dense Sampling	68.2	**65.4**	71ms	8ms
PointPillars	First Stage	66.5	57.4	56ms	-
	+ Box Center	67.3	57.4	56ms	6ms
	+ Surface Center	**67.5**	57.9	56ms	7ms
	Dense Sampling	67.3	**57.9**	56ms	8ms

<Comparison between one-stage and two-stage methods on Waymo dataset>

Experiments – Ablation studies

- **3D Tracking**
  - Compare with last year's challenge winner Chiu et al. [10], which uses mahalanobis distance-based Kalman filter to associate detection results of CBGS
  - Using our simple velocity-based closest point distance matching outperforms the Kalman filter-based Mahalanobis distance matching
  - Improvements
    1. Model the object motion with a learned point velocity, rather than modeling 3D bounding box dynamic with a Kalman filter
    2. Match objects by center point-distance instead of a Mahalanobis distance
    3. A simple nearest-neighbor matching without any hidden-state computation

Detector	Tracker	AMOTA↑	AMOTP↓	T_{track}	T_{tot}
CenterPoint-Voxel	Point	**63.7**	**0.606**	1ms	62ms
CBGS [65]	Point	59.8	0.682	1ms	$> 182ms$
CenterPoint-Voxel	M-KF	60.0	0.765	73ms	135ms
CBGS [65]	M-KF	56.1	0.800	73ms	>254ms

<Ablation studies for 3D tracking on nuScenes validation>

Conclusion

- CenterPoint : a center-based framework for simultaneous 3D object detection and tracking from the Lidar point-clouds

- Detection is a simple local peak extraction with refinement, and tracking is a closest-distance matching

- CenterPoint is simple, near real-time, and achieves state-of-the-art performance on the Waymo and NuScenes benchmarks