What does the topic cover?

Why it is important?

3D object detection is the ability to define and identify objects which is usually present in 2D images. Different from 2D object detection, 3D object detection requires bounding box an extra dimension which could imply the object’s volume. With the help of modern sensors, such as LIDAR could contain depth information, while camera could provide image data, 3D detection in point clouds should be able to achieve. However, accurately detecting 3D object is still a challenging problem in computer vision, due to the lack of object features and invisible parts. Despite of that, it plays an important role in autonomous applications, especially in field of autonomous driving. Because of the rapid demand on intelligent transport, self-driving vehicles not only aim to make driving easier but also pursue high-standard protection. In the context of self-driving transportation, sensors like LIDAR are ubiquitous since its high accuracy is required for safety, but the major problem is that the output data is in a form of a point cloud that is unstructured, which means that it may put great challenge on computation. To solve this, efficient algorithms for detecting 3D objects in point clouds are greatly in need of.

What are applications of the topic?

What is the societal significance of the research?

Environment perception is extremely important in modern self-driving vehicles, self-navigating robots and many other collision-avoidance-like systems. Using sensors to detect the nearby objects is common in many autonomous driving applications. Normally, the navigating system contains LINDAR, RADAR and cameras, which could interpret the detected static and dynamic objects into data. LINDAR, one of the most accurate remote sensing methods, detects surroundings and returns the data with depth in the form of point cloud. Unlike images in RGB pixels, point cloud is consist of sparse points with variable density. In order to extract the objects’ features, algorithms play a significant role to turn the clouds into 3D voxel and encode it with features. It is the fundamental of modern autonomous system which is based on computer vision. Due to the large amount of the points (normally ~100K points) and difficulty in features, detecting and recognition 3D object in point cloud poses a great challenge to computational and memory requirement. If there are some algorithms could balance the speed and accuracy on detecting objects, meanwhile optimize the calculation and memory usage, autonomous navigating system not only be used to sense the surroundings, but also predict and avoid the possible dangers. In a word, it should be able to set up in industrial and daily situation as a pre-warning, or in advance a fully automatic system to make sure the operators’, pedestrians’ and properties’ safety. Furthermore, if the sensors could be miniaturization, it may be possible to be used in the field of robotics and medical use, including robot navigation and assistance for people who are visually impaired.

Pick an area of focus that interests you in the topic

Literature review

As comprehensive as you can, research the different approaches and solutions in research community and industry

3D object detection was developed from 2D object detection, with the rise of the autonomous driving. Combined with 2D images, 3D object detection are commonly assisted with depth sensers which normally provide point cloud information. Starting from 2017, 3D object detection has seen rapid progress thanks to the continuously advances in machine learning on point cloud.

In 2017, ‘PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation’ Charles R. Qi and Hao Su designed a new type of neural network that provides a unified architecture for applications. It is the first time using the original point information as the input data, meanwhile it focuses more on transformation invariance and rotation invariance. 1

In 2017, ‘VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection’ written by Yin Zhou and Oncel Tuzel present an end-to-end architecture for 3D object detection called VoxelNet. VoxelNet could directly use the sparse points without manually designed object features largely reducing data restructure process.2

In 2018, ‘PIXOR: Real-time 3D Object Detection from Point Clouds’ written by Bin Yang, Wenjie Luo and Raquel Urtasun proposed a real-time 3D project detector called PIXOR. PIXOR is a proposal-free Single-stage object detector which could directly predict the finals detections. It uses the BEV(Bird’s Eye View) to decrease the calculation workload. Unlike the previous two-stage detection, this single-stage detection redesigns the input representation, network architecture, and output parameterization, meanwhile, removes the hyper parameter of pre-defined object anchors by re-defining the objective function of object localization, which leads to a simpler detection framework. 3

In 2019, ‘PointRCNN: 3D Object Proposal Generation and Detection from Point Cloud’ written by Shaoshuai Shi, Xiaogang Wang and Hongsheng Li proposed a 2-stage detector namely: PointRCNN. The stage-1 is for the bottom-up 3D proposal generation and stage-2 is for refining proposals in the canonical coordinates to obtain the final detection results. The paper’s greatest attribution should be the stage-1 process which could generate relatively small number of high-quality points largely increasing the accuracy of bis refinement and confidence prediction.4

In 2019, ‘PV-RCNN: Point-Voxel Feature Set Abstraction for 3D Object Detection’ written by Shaoshuai Shi, Chaoxu Guo and Li Jiang first published on arXiv, and updated in 2021. They present a high-performance 3D object detection framework, named PointVoxel-RCNN (PV-RCNN), for accurate 3D object detection from point clouds. In this paper, they combine grid-based method’s and point-based method’s advantages making their approach based on either 3D voxel CNN with sparse convolution or PointNet-based networks as the backbone, since voxel with sparse convolution are more efficient and the point-based methods can extract more accurate information.5

Open Source research

Research the different open source projects that touch the topic of your interest

In 2021, Wu zheng, Weiliang Tang and Li Jiang published a paper ‘SE-SSD: Self-Ensembling Single-Stage Object Detector From Point Cloud’. By using improved teacher-student network, their method ranked high in the KITTI benchmark. And their code is available at: <https://github.com/Vegeta2020/SE-SSD>. Their method starts with inputting cloud point to teacher to produce relatively precise bounding boxes and confidence, taking the predictions as soft target to supervise the students, which could assist the detection effectively.6

Their code is mainly based on Det3D, the first 3D Object Detection toolbox which provides off the box implementations of many 3D object detection algorithms, and CIA-SSD, a new single-stage detector which could align the localization accuracy and classification confidence.

The code requires Linux, Python 3.6+, PyTorch 1.1-1.6, CUDA 10.0/10.1, CMake 3.13.2 or higher, spconv, nuscenes-devkit

The file structure is as follow:

# For KITTI Dataset

└── KITTI\_DATASET\_ROOT

├── training <-- 7481 train data

| ├── image\_2 <-- for visualization

| ├── calib

| ├── label\_2

| ├── velodyne

| └── velodyne\_reduced <-- empty directory

└── testing <-- 7580 test data

├── image\_2 <-- for visualization

├── calib

├── velodyne

└── velodyne\_reduced <-- empty directory

# For nuScenes Dataset

└── NUSCENES\_TRAINVAL\_DATASET\_ROOT

├── samples <-- key frames

├── sweeps <-- frames without annotation

├── maps <-- unused

└── v1.0-trainval <-- metadata and annotations

└── NUSCENES\_TEST\_DATASET\_ROOT

├── samples <-- key frames

├── sweeps <-- frames without annotation

├── maps <-- unused

└── v1.0-test <-- metadata

# For Lyft Dataset

└── LYFT\_DATASET\_ROOT

├── trainval

| ├── data

| ├── lidar

| └── maps

└── test

├── data

├── lidar

└── maps

Installation

$ git clone https://github.com/Vegeta2020/SE-SSD.git

$ cd ./SE-SSD/det3d/core/iou3d

$ python setup.py install

$ cd ./SE-SSD

$ python setup.py build develop

$ git clone https://github.com/jackd/ifp-sample.git

$ pip install -e ifp-sample

Please follow Det3D for installation of other related packages and data preparation.

Train and Eval

Configure the model in

$ /SE-SSD/examples/second/configs/config.py

Please use our code to generate ground truth data:

$ python ./SE-SSD/tools/create\_data.py

Train the SE-SSD:

$ cd ./SE-SSD/tools

$ python train.py # Single GPU

$ python -m torch.distributed.launch --nproc\_per\_node=4 train.py # Multiple GPU

Evaluate the SE-SSD:

$ cd ./SE-SSD/tools

$ python test.py

1. C. R. Qi, L. Yi, H. Su and L. J. Guibas, arXiv preprint arXiv:1706.02413 (2017).

2. Y. Zhou and O. Tuzel, presented at the Proceedings of the IEEE conference on computer vision and pattern recognition, 2018 (unpublished).

3. B. Yang, W. Luo and R. Urtasun, presented at the Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, 2018 (unpublished).

4. S. Shi, X. Wang and H. Li, presented at the Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019 (unpublished).

5. S. Shi, C. Guo, L. Jiang, Z. Wang, J. Shi, X. Wang and H. Li, presented at the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020 (unpublished).

6. W. Zheng, W. Tang, L. Jiang and C.-W. Fu, presented at the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021 (unpublished).