3D Object Detecting With ARKit

Xufeng Shen
BUID:U61121388

Abstract:This is the Boston University EC601 course first project which is a description for the 3D object detecting with ARKit. The project is aim to figure out what ARKit can do and explore the object detecting capability. In this section, I will focus on how the 3D object detection can be done and then I will talk about things can be done with ARKit and 3D detection which is interesting.

Motivation:

Recent years, lots of new concepts were introduced to public including self-driving cars, mobile augmented reality and different kind of robots. A big part of these technologies is detecting objects. Knowing where the objects are and what the object is became the first step, and base on these information, machines can do right decisions. While lots of works focus on 2D detection which means takes images as input, for some cases the 3D detection is important because 2D object detection can not provide three-dimensional information like depth, size, physical world parameter, which makes it can not solve the problems in real world. To achieve the goal, people started to focus on how to detect the 3D objects by using different kinds of methods.

Problem Statement:

So the next problem is what the 3D objects detection is and how can we do it. There is a survey[1] provides lots of information.

2D objects detection

Before going into 3D world, there are lots of methods are used for 2D objects detection which means detect objects from images and videos. Traditional methods to the job is features extracting including SIFT, HOG and SURF.

Then when the deep learning came into the public, people thought they can do the detection job by using deep learning, the basic idea is using artificial neural network to learn the training data, and to do the detection job, mainstream methods include CNN, R-CNN, YOLO.

3D objects detection

Since 2D detection can not provides depth, size and other real world information, the 3D detection came to people's eyes. In real cases, for one hand we need accuracy and for the other we want to reduce the cost.

The survey[1] divides the 3D detection methods into 4 types depending on different kind of inputs.

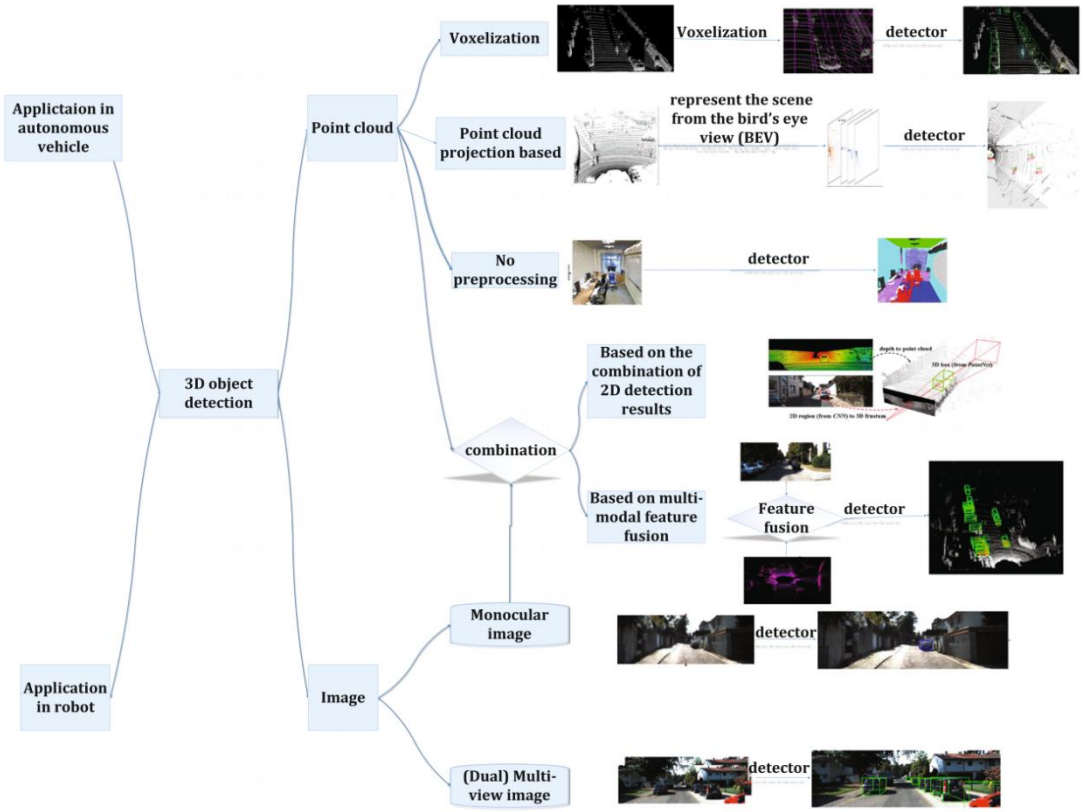

Figure1. From paper[1]

See Figure1 which is from[1], there are four kinds of input: original point cloud, monocular image, multi-view image, fusion of point cloud and image.

The paper also list the advantages and disadvantages of four methods, see Figure2.

Table 1 Comparison of frameworks based on different input data types

Methods	Advantage	Disadvantage
Point cloud	Richer spatial structure information; higher accuracy than image-based 3D object detection	The cost of acquiring data is expensive; the original point cloud cannot provide texture information of object; computationally expensive
Fusion of point cloud and image	Utilize point cloud and image at the same time to get more accurate output	Computationally expensive
Multi-view image	Fuse image features from different angles in images; has higher accuracy than Monocular-based method	Need to calculate depth information
Monocular image	The data is easy to get; most of them are improved on the basis of 2D object detection method	Lack of accurate depth information; lack of features of spatial structure; need prior information; need to calculate depth information

Figure2. From paper[1]

From[2] we can see the sensors are very import for 3D detection, base on the different kind of detection methods, the sensors include monocular camera, stereo camera, Lidar,

Solid-State lidar. And we need to find a suitable way to make balance between cost and accuracy.

Related Work:

Original point cloud:
Lots of work are done for the classification on the original point cloud data segmentation, the survey in this field[3] shows five methods are used to in 3D point cloud segmentation includes edge based, region based, attributes based, model based, graph based.

Fusion of point cloud and image:
For the fusion of image and point cloud, the survey[4] concluded the main works in autonomous cars field includes Early fusion, late fusion and deep fusion. And this paper concludes advantages and disadvantages of each methods.

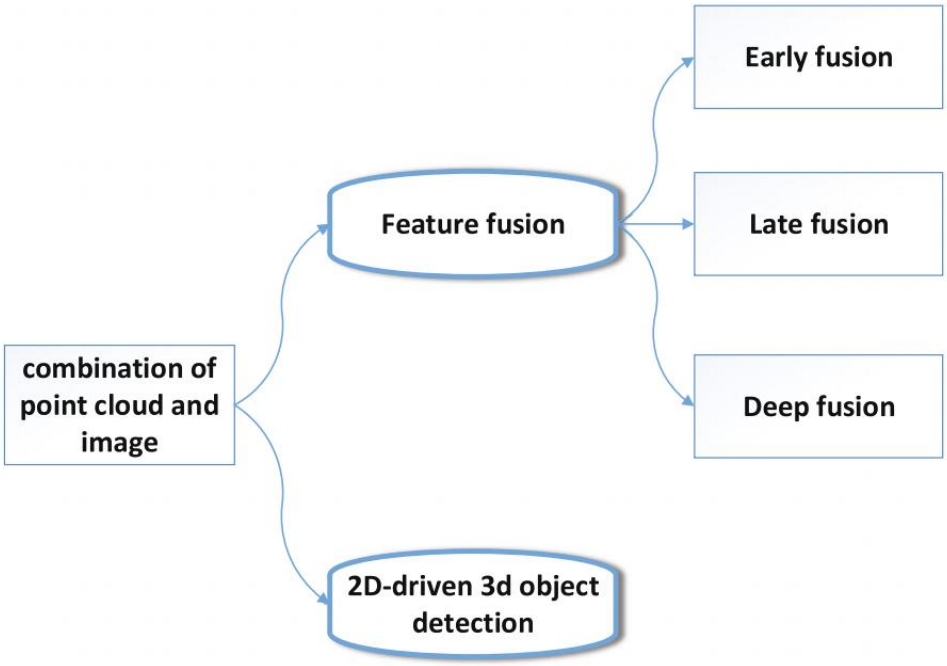

Figure3. From paper[1]

Multi-view image:
There are several ides are proposed in this area. When some people take advantages of the depth information using RGB-D cameras, while some people did not believe the depth estimation can make a big difference. The basic method to use multi-view images is take multi-view images as input and project it into 3D point cloud data.

Monocular image:
The last one is take monocular image as input, the key is doing the depth estimation. People used deep learning and other models to estimate the distance. This method is lack of accuracy.

Applications:

Measurement: Since we already know the 3D detection can provides more objects' information like size, position. We can use this to measure the length and width of the real world objects.
Design: This includes many fields like shoes virtual try-on, furniture virtual try to see if it fit the size of the room. These are all base on the object had been detected.
Game: Another important application fields is AR games, which also can be done by detecting objects.
Self-driving: As I mentioned before, the self-driving cars need to know more objects information to make the right decisions, so the 3D object detection in more important.

Here is a example from Apple, which is a 3D object detection app developed by Apple using ARKit.
(https://developer.apple.com/documentation/arkit/content_anchors/scanning_and_detecting_3d_objects)

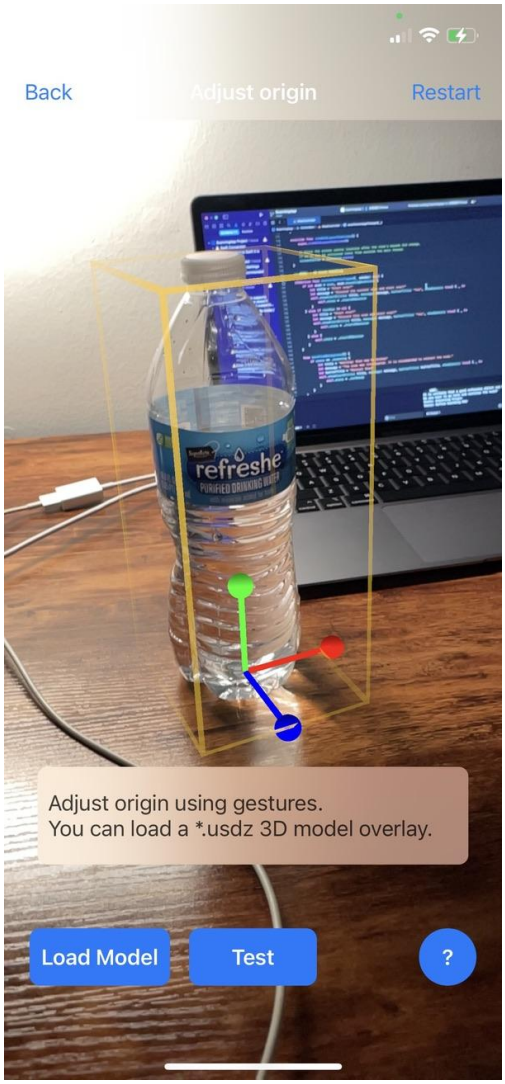

Figure3. The 3D detection app provided by Apple

Open Sources:

Data sets:

●*Original point cloud data sets:*

1. Cornell RGBD datasets 2.VMR-Oakland dataset 3.KITTI dataset 4.Robotic 3D Scan Repository

●*Indoor datasets:*

1. Princeton modelnet https://modelnet.cs.princeton.edu/
2. NYU Depth Dataset V2 https://cs.nyu.edu/~silberman/datasets/nyu_depth_v2.html

●*Outdoor datasets:*

1. KITTI https://www.cvlibs.net/datasets/kitti/
2. PandaSet https://scale.com/open-av-datasets/pandaset

Official documentation:

[1] https://developer.apple.com/documentation/arkit/content_anchors/scanning_and_detecting_3d_objects

[2] https://developer.apple.com/cn/documentation/

Other open sources:

[3] https://google.github.io/mediapipe/solutions/objectron#overview

[4] https://github.com/brianadvent/3DObjectScanningAndDetection

[5] https://github.com/jmousseau/RoomObjectReplicator

[6] https://github.com/jeeliz/jeelizAR

References:

[1] Liang, W., Xu, P., Guo, L., Bai, H., Zhou, Y. and Chen, F., 2021. A survey of 3D object detection. Multimedia Tools and Applications, 80(19), pp.29617-29641.

[2] Wang, Y. and Ye, J., 2020. An overview of 3d object detection. arXiv preprint arXiv:2010.15614.

[3] Nguyen, A. and Le, B., 2013, November. 3D point cloud segmentation: A survey. In 2013 6th IEEE conference on robotics, automation and mechatronics (RAM) (pp. 225-230). IEEE.

[4] Cui, Y., Chen, R., Chu, W., Chen, L., Tian, D., Li, Y. and Cao, D., 2021. Deep learning for image and point cloud fusion in autonomous driving: A review. IEEE Transactions on Intelligent Transportation Systems, 23(2), pp.722-739.

[5] Mandikal, P., Navaneet, K.L., Agarwal, M. and Babu, R.V., 2018. 3D-LMNet: Latent embedding matching for accurate and diverse 3D point cloud reconstruction from a single image. arXiv preprint arXiv:1807.07796.

[6] Ghasemi, Y., Jeong, H., Choi, S.H., Park, K.B. and Lee, J.Y., 2022. Deep learning-based object detection in augmented reality: A systematic review. Computers in Industry, 139, p.103661.