Project 1: Detecting 3D objects
Wenxin Zu U75249292

The purpose of the object detection task is to find the regions of interest in an image, and then determine the location and category of these regions. Because the detected targets have different appearances and shapes, as well as the interference of other factors in the imaging process, such as light, occlusion, etc. Therefore, in the field of computer vision, object detection is very challenging.

3D object detection contains information such as the length, width, height, and rotation angle of the object. In practical applications, object detection in 3D space has far-reaching significance. In the fields of AR/VR, unmanned driving, and biomedical detection, 3D target detection technology can well reflect its own advantages and complete related tasks. Specifically, when an autonomous vehicle drives on the road, it needs to perceive the surrounding 3D scene, and 3D object detection is used to obtain the position and category information of the object in the 3D space, which is the basis of the autonomous driving perception system. Planning, motion prediction, and collision avoidance play an important guiding role.

I am going to apply ARKit to detect 3d objects. ARKit provides functions such as position tracking, plane modeling, ambient lighting rendering, scale estimation, XCode application templates, and real-time rendering, allowing developers to use computer vision technology to build virtual content in real-world scenes, develop interactive games, immerse shopping experience, industrial design, etc. The most basic requirement of an augmented reality experience is to create and track the correspondence between the real world and the virtual world where the user is located, which is also the core feature of ARKit[1].

ARKit brings lots of benefits to people’s life. To developers, ARKit gives app designers and developers the freedom to add AR reality to their apps, giving them the freedom to focus on creating great experiences and content. Object mapping and spatial tracking are no longer performed using computer vision and physics algorithms. For businesses, the potential use cases for AR are significant, ranging from retail or virtual shopping to training, to customer engagement, to gaming and more. For the public, ARKit can make our lives more convenient, such as indoor maps of shopping malls and airports in map applications, and the ability to check whether clothes or shoes fit you on your phone at home, and more.

There are two main methods detect 3D objects, 2D Reconstruction and Point Cloud detection. In computer vision, 2D images are stored as a matrix. In the 2D reconstruction method, a neural network can be used to learn many pictures of a certain type, and objects can be compared with these data. Since the data of 3D objects requires depth information, there are many methods for depth estimation based on pictures. T. Hassner and R. Basri presented a new solution to the problem of depth reconstruction from a single image. Single-view 3D reconstruction is an anachronistic problem. They solve this problem using an example-based synthesis approach[2]. Since some information is lost after the depth calculation, this will cause the gradient to disappear, resulting in inaccurate detection. Wenbo Lan improved the network structure of YOLO algorithm, and a new network structure YOLO-R is proposed to improve the ability of the network to extract the feature information of shallow items[3].

Since 2D based method cannot provide accurate 3D position information, which is a crucial issue for 3D detecting application, like robotics, or autonomous driving. So, point cloud detection became popular since it preserves all the original information of a 3D object like coordinate triplets. The output data of LiDAR is a 3D point cloud. A point cloud is a digital 3D representation of a physical object or space. It consists of millions of individual measurement points, each with x, y, and z coordinates. Each point can also include RGB color data, and even reflection intensity information. There are two main tools that can be used to capture point clouds: laser scanners and photogrammetry. Accurately detecting objects in 3D point clouds is a core problem for many applications, such as autonomous navigation, housekeeping robotics, and virtual reality. Yin Zhou eliminates the need for manual feature engineering of 3D point clouds and proposes VoxelNet[4], which significantly outperforms state-of-the-art lidar-based 3D detection methods. Weijing Shi proposed a graph neural network for object detection from lidar point clouds. The results demonstrate the potential of using graph neural networks as a new method for 3D object detection[5].

For 2D reconstruction methods, convolutional neural networks in deep learning can extract feature expressions from a large amount of data, and the extracted features can summarize object performance attributes. Therefore, convolutional neural networks are very suitable for processing large-scale image data. R-CNN first uses selective search to obtain the target candidate area in the image, but requires the input image size to be fixed, and the image needs to be scaled in the early stage, resulting in image deformation and distortion. Fast R-CNN uses a multi-task model to integrate classification and localization tasks into one network, which simplifies the network structure and speeds up the detection rate. FasterR-CNN adds anchors on its basis, but it is not suitable for detecting small targets due to multiple down sampling and the use of anchor boxes to generate reference candidates. YOLO regards target detection as a regression problem and uses an independent network to directly predict the position and category of the target bounding box through a convolution, and the speed is further improved. 3D object detection based on point cloud is divided into two steps: generating candidate regions and classifying objects. Among them, PointNet is a deep network with a unified structure, which directly takes the point cloud as the input and outputs the target category label or the segmentation result of the point cloud. The classification accuracy is high, but the local information in the point cloud is not utilized, resulting in poor recognition of complex scenes. PointNet++ can extract local features in the data, and the network efficiency is also greatly improved.

In general, each method has its own advantages and disadvantages. Getting a computer to process a complex 3D point cloud is a challenging task. During the research process this semester, I will try to use one of these methods to identify a 3D object in a life scene, or to innovate based on the data information of the identified object, letting ARKit be closer to our lives.

Citation:

[1] *Apple Developer Documentation*. developer.apple.com/documentation/arkit.

Accessed 16 Sept. 2022.

[2] T. Hassner and R. Basri, "Example Based 3D Reconstruction from Single 2D Images," 2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'06), 2006, pp. 15-15, doi: 10.1109/CVPRW.2006.76.

[3] W. Lan, J. Dang, Y. Wang and S. Wang, "Pedestrian Detection Based on YOLO Network Model," 2018 IEEE International Conference on Mechatronics and Automation (ICMA), 2018, pp. 1547-1551, doi: 10.1109/ICMA.2018.8484698.

[4] Y. Zhou and O. Tuzel, "VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection," in 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 2018 pp. 4490-4499.

[5] W. Shi and R. Rajkumar, "Point-GNN: Graph Neural Network for 3D Object Detection in a Point Cloud," in 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 2020 pp. 1708-1716.

Open Source:

https://developer.apple.com/augmented-reality/arkit/
https://ieeexplore.ieee.org/abstract/document/9204243
https://github.com/jigs611989/ARKitDemo
https://github.com/shu223/ARKit-Sampler