

Smart System in Automation’s Report

Topic: Identify and Measure precisely Object distance

Members

Huỳnh Du Kiệt

Dương Trọng Anh

Cao Minh Ngọc Anh

1. INTRODUCTION

1.1) General information about the Project

* In this project, we will use the depth camera D435 to measure the distance of an object from the camera and also integrate object detection with artificial intelligence into this project.

1.2) General information about depth camera D435

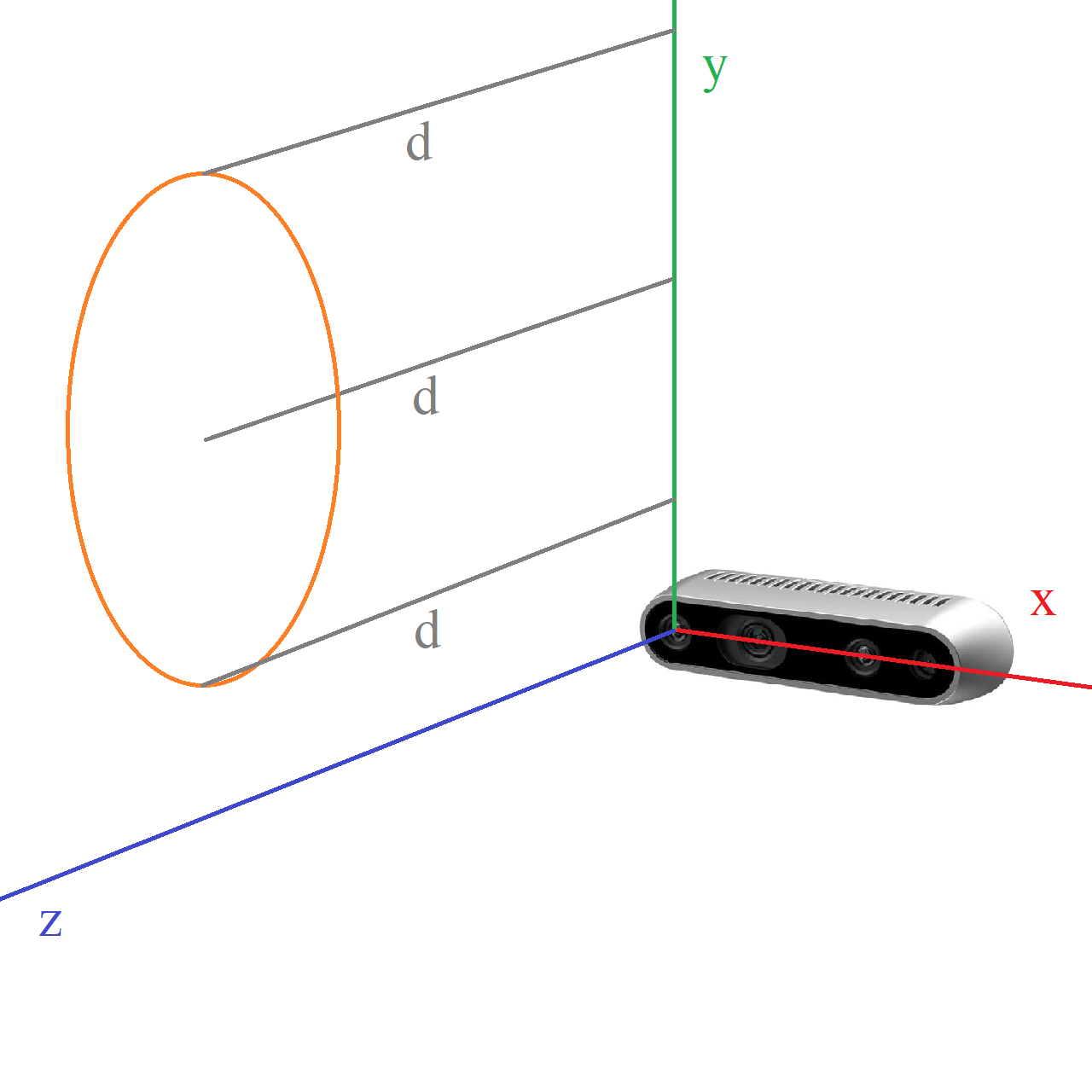
| **FEATURES** | **Use Environment** | Indoor/Outdoor |
| --- | --- | --- |
| **Depth Technology** | Active IR Stereo |
| **Image Sensor Technology** | Global Shutter, 3μm x3μm pixel size |
| **Depth FOV (H x V)** | 85.2o x 58o (±3o) for HD 16:9 |
| **Depth Output Resolution** | Up to 1280 x 720 |
| **Frame Rate** | Up to 90fps |
| **MAJOR COMPONENTS** | **Camera Module** | Intel® RealSense™ Module D430 + RGB Camera |
| **Vision Processor Board** | Intel® RealSense™ Vision Processor D4 |
| **Minimum Depth Distance** | 0.105m |
| **Maximum Range** | 10+ m |
| **RGB Resolution** | Up to 1920 x 1080 |
| **RGB FOV (H x V x D)** | 69.4 x 42.5 x 77 (±3o) |
| **PHYSICAL** | **Form Factor** | Camera Peripheral |
| **Connectors** | USB 3 Type-C |
| **L x D x H** | 90mm x 25mm x 25mm |

1. METHODOLOGY

2.1) Measure Object Distance

2.1.1) General information

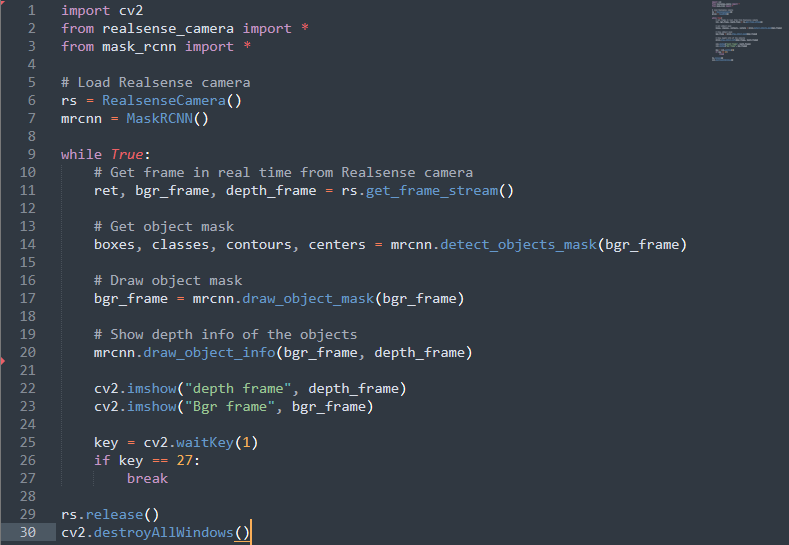
* The principle of measuring the distance of objects is to display the value of “depth” from the camera lens to the objects. To put it simply, if we put the Intel RealSense D435 on a 3D graph, the camera lens will be on the xy-plane, and the distance will be projection **d** along the z-axis as shown in the analogy picture here:



* **d** will be the value that the camera detects.
* However, the surface of an object can have a lot of points which have variable distances to the vertical plane where the lens is. To solve this problem, we only need one singular reference point to estimate the overall distance of the target. More details are explained in the next section.

2.1.2) System construction

* This section will explain the mechanism of distance measuring through the code as shown here:



* Before coding, OpenCV, and the Mask R-CNN (explained in the next section) are needed.
* First, we initialize the camera by assigning it to a variable, as shown in the 6th line. The Mask R-CNN algorithm is added as well
* When the initialization is completed without any issues, the next part of the code starts running. It starts with the 11th line, which gets the real-time, the current frame captured by the camera, and the depth of everything in the frame respectively.
* In the 14th line, the machine will receive the objects detected, what they are and where each of them are located. Then the target will be covered in a colored area.
* For each target, a reference point is selected in order for the machine to receive the estimated distance of the target from the camera len.
* To visualize the data, line 20 to line 23 put the text telling the distance value returned from the camera.

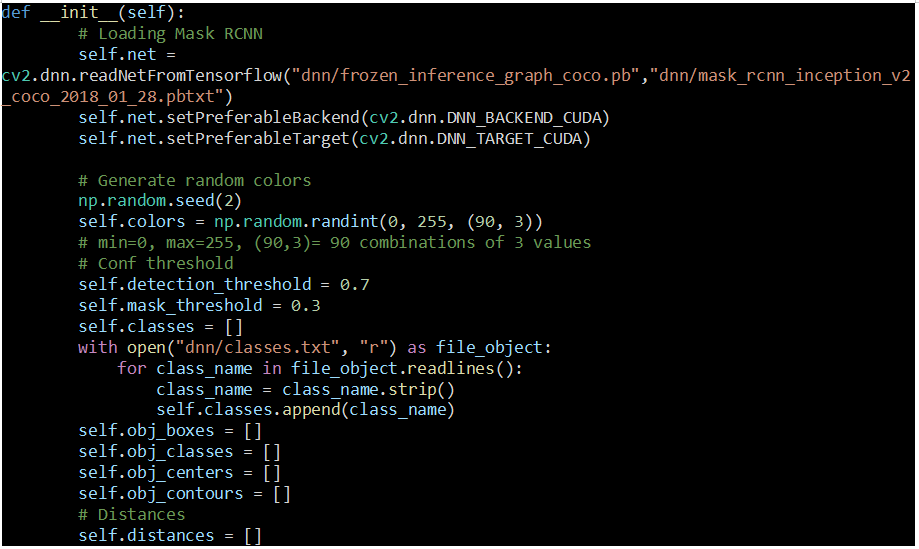
2.2) Identify the Object

2.2.1) General information

* In this project, we will use the Mask R-CNN algorithm. Mask R-CNN is an instant segmentation algorithm, which means it can recognize objects in images while concurrently applying a mask to each item. This implies that for a person, you have not just the box but also the coordinates that surround the person.
* Mask R-CNN was invented in 2017, thus it is rather ancient; yet, in the world of computer vision software development, it is still a great method that can be utilized in commercial applications.

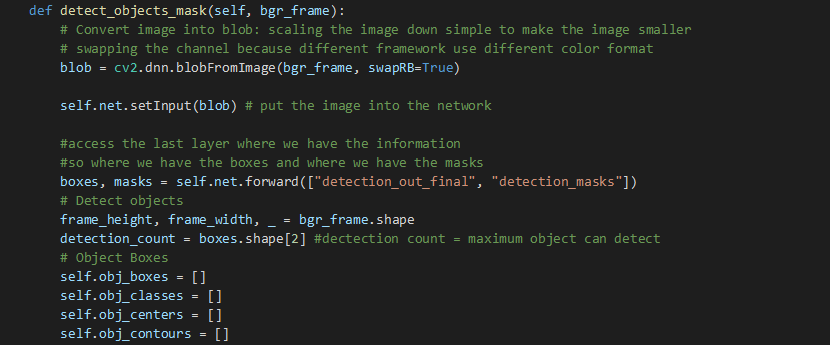
2.2.2) System construction

1. Initialize function

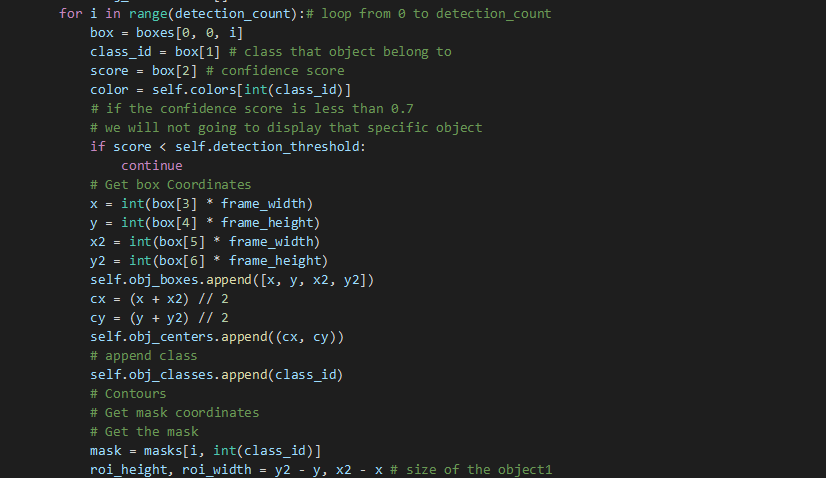


* The first step is to load Mask RCNN to our network. Then, based on the device that the code is running from, either using CUDA cores from the Nvidia GPU or using the CPU entirely, it continues to the next step.
* The second step begins by generating random colors for the mask, also the detection threshold and mask threshold. Next, it appends all the classes in “classes.txt” into the array classes.
* The third step of initialization is creating necessary arrays for the object, its box, class, center, and its contours. Finally, it also covers the distance of the object.

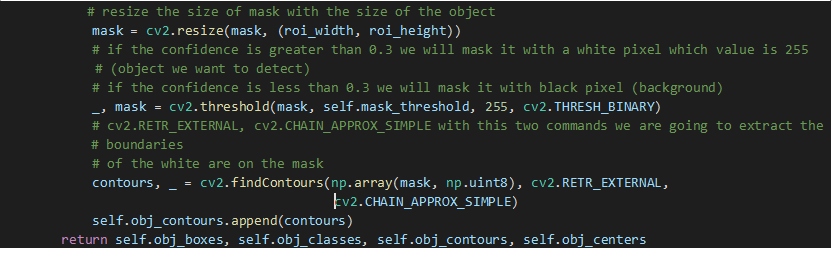
1. detect\_objects\_mask function



* This function, firstly, converts the image into blob, which stands for a binary large object, which is a collection of binary data stored as a single entity. Blobs are typically images, audio, or other multimedia objects[1]. Secondly, put the blob into the network and utilize its last layers, where the important information is stored, to make the box moreover the object mask.



* Loop of detection for each detected object. Append its confident score, box, center, and class it belongs to.
* Next is to detect the object contours, which is getting mask coordinates, to the object mask.



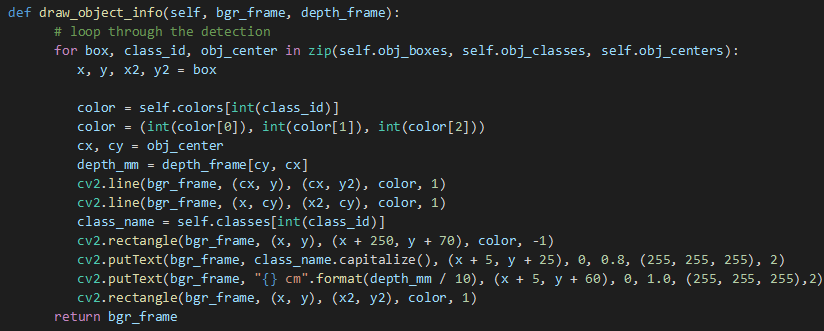
* Configure the size of the object, then resize the mask according to the object’s size. If the confidence is greater than 0.3, a white pixel, a value of 255, is the indicator of the object we would like to detect. In Contrast, if the confidence is less than 0.3, we will mask it as a black pixel, it is the background we would have said.
* With these two commands, cv2.RETR\_EXTERNAL, cv2.CHAIN\_APPROX\_SIMPLE, the extraction of the boundaries of the white pixels are on the mask.

1. draw\_object\_mask function



* After the detection of an object's mask, its mask needs to be drawn on the network which will be called a background frame. A loop function is used to go through the detection, their frame is set with a color based on their classes and the contours will be drawn as well as a polygon on the region of interest (ROI).

1. draw\_object\_info function



* The final procedure to finish Mark RCNN is to display the object information to the network. In the loop function for each detection, the object's name, and distance are displayed on the camera screen.

1. EXPERIMENT (Weakness and Strength of the Project)
2. Weakness

* When we tested the project, there are still some vibrations in the distance measurement (± 2%)
* It takes a few seconds (10-15 seconds) for the object detection function to determine the object correctly.

1. Strength

* Although there is a weakness in distance measurement, it measures the distance quite fast and is trustable.
* The object detection function works well since it can detect correctly almost any object that includes in the object’s classes

1. CONCLUSION AND FUTURE WORK

* This is a promising project, in the future, we can improve and develop it so that it can be used to identify VGU’s students by measuring the matching percentage between card photos on the student card and their face.