Distance Estimation using Intel RealSense Camera – Report

---

1. About Intel RealSense Depth Camera D455

Intel's RealSense technology leverages stereo image sensing techniques using cameras to calculate depth, enabling devices to see, understand and interact with the 3D world in real time. The latest Depth Camera D455 extends the distance between the depth sensors to 95 mm which improves the depth error to less than 2% at 4 m. It comes with an RGB sensor with a global shutter and is matches to the depth FOV of 86 °. The camera integrates an Inertial Measurement Unit (IMU) to allow applications to refine its depth awareness in any situation where the camera moves.

2. Objective for Testing the camera

To estimate the distance the distance between the camera and 3D objects in real time. For this purpose, we'll get the frames from Depth Camera and integrate with OpenCV – Python to display the measured distance between a specific point of frame and camera. For testing, we have estimated the distance for different use-cases, objects and environments.

3. Packages Required

opencv-python – An open-source library for computer vision, machine learning and image processing.

pyrealsense2 – Python library for accessing Intel RealSense cameras.

numpy – For different numerical operations on the array data**.**

4. Results

The screenshots below are obtained from different testing conditions w.r.t the lighting, angle of incidence and the use-case. The camera is mounted on a tripod and placed at a fixed distance of around 30 cm measured using a linear scale.

Errors in measurement are seen with increasing distance above 100cm observed by flickering values. Distances below 30cm are also not detected properly, in most cases estimated as zero since it's much closer to the sensor. It is therefore recommended that the object be placed at a

distance of 30-100 cms from the camera for a real application. The estimated distance is also subjected to a small amount of error due to the position of camera, thickness of objects and imprecise sensing.

4.1. Distance between the camera and the wall (Indoor lighting)

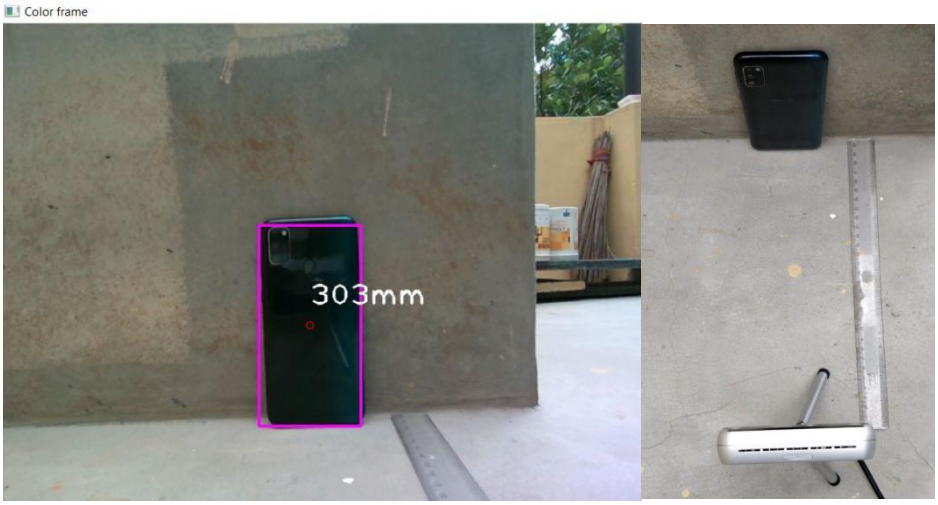
This exercise was performed inside a square 15ft x 15ft room.

Here it estimates the distance of the wall from the camera to be 423.7 cm which is around 14ft. Since the camera is not exactly positioned at origin, the exact distance is lesser than 15ft.

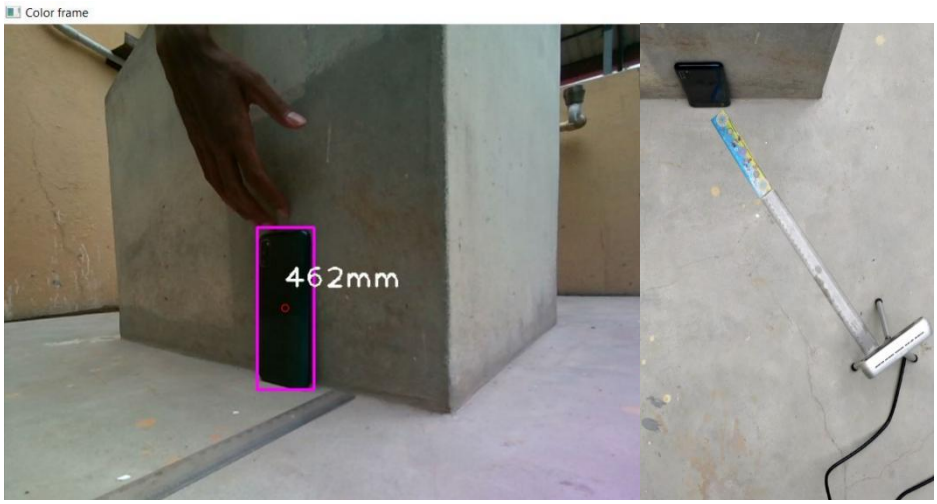
4.2. Distance between camera and mobile phone supported on a wall (Outdoor - Daylight)

This exercise was performed in outdoor conditions in daylight. Here, the frames are captured at two different angles and the distance is measured for a specific object of interest, mobile phone inclined on a wall. The input image frames are retrieved from the depth camera and the object of interest is detected using YOLOv3 object detection model using pertained weights and classes. Once the detection is seen, a bounding box is drawn around the mobile phone and the centroid is then taken as frame of reference for calculating distance from the origin point, where the camera is positioned. The above use-case is tested in two different angles, perpendicular distance and angled distance and in both the cases, estimation is seen and are subjected to small errors of around ±2 cm.

i. Perpendicular distance



ii. Angled distance

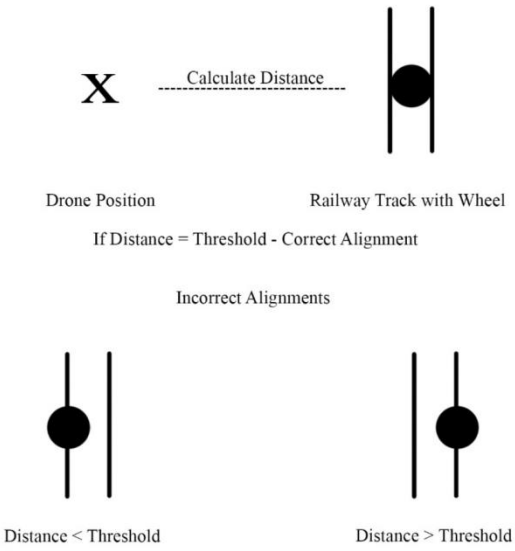


5. Future Work

The above implementation can be extended to a more realistic use-case where we check for misalignments and improper positions by estimating the gap distance between the desired objects. Any distance above a set threshold is then labelled as a warning and the alignment can be fixed, right away.

Wheel Alignment Use-Case:

A misalignment in the train's wheel position is caused due to certain crane operations on containers placed on goods train. The wheel is either shifted to the left/right of the track which causes problems in the train's operation. This usecase deals with this particular issue by using the Intel RealSense Depth Camera to calculate the distance of the wheel from the drone.



Algorithm:

1) Precalculate the distance between an aligned wheel and the position of the drone. (Assume X centimetres)

2) In actual deployment, position of drone is kept fixed and distance between the wheel and drone is calculated.

3) If Distance D is within range of X ± Error, wheel is aligned.

4) If Distance D is outside range of X ± Error, wheel is misaligned, and error is raised.