LOCATION ESTIMATOR USING ARUCO MARKERS

INTRODUCTION

The Location of a person in a room is estimated with the help of Aruco Markers.
After experimenting with a lot of tools available online, I found ArucoMarkers to work more efficiently that Conventional Trackers.

Flow:

1) Camera is Initially Caliberated using CameraCaliberate.py
2) Instructions on how to run CameraCaliberate.py is available on the readme.md
3) The CameraCaliberate writes on to “Focal.txt”. Focal Length and the the Marker Length are written.
4) These two values are then read while running the main. If Camera is not caliberated atleast once, the Main.py will not run.
5) Steps to Run the Main.py is also explained in the readme.md with video links

IMPORTANT CONCEPTS

Finding the Focal Length
The Focal lengthn helps in estimating the depth of an object.
To do this, we need
1) ArucoMarker Length in cms (Dist_real) - passed as argument to the program with “--AWidth”
2) ArucoMarker Length in pixels(Dist_pix) - computed during the run time
3) ArucoMarker Distance from camera(Known_Depth) - passed as an agrument with “--ADist”

Theory:

Formula for depth
Dist_pix = width of the box in pixels
Dist_real = distance of the box in real world estimates
Depth = Known depth

Computing the focal length of the camera

F = (Dist_pix * Known depth) / Dist_real

Caliberation is complete, you can use this F to Compute every other depths
For example,
New Depth = (Dist_real * Focal)/ Dist_pix

Now To Caliberate the X axis.
Camera has 1280 pixels in my case. (i.e) 640 in the left and 640 in the right.

Depending upon the Camera position, The pixels are mapped to real_world units
For Example:
if Camera is placed at 2 ft and the wall's length is 10 ft.

The first 61 pixels (2 ft is 61 cms) is mapped to the first 640 pixels of the camera
The rest 305-61 pixels(10ft - 2ft) is mapped to the other 640 pixels

Assuming that the Camera angle is perpendicular and the camera's Y axis is 0,
We Caliberate the X pixels to that of cms in the Screen

LOCATION ESTIMATOR USING ARUCO MARKERS

RoadBlocks faced

1) Tracking using the Boxes produced by Gender Recognition/ Haar Classifier xml models, was expensive and computer crashed often.
2) Reducing the Screen size without changing the aspect ratio worked faster, but details were lost.
3) Computing the camera parameters (intrinsic, Extrinsic) and lens distorsion was relatively difficult to implement. Reference plane was hard to find.
4) Haar Classifiers had Pedestrain detector xml models which gave good results outdoors, Inside the room the accuracy was poor and caliberatiing the boxes from the classifiers to real world metrics were difficult.
5) This led the way to using ArucoMarkers on people.

Advantages of using ArucoMarkers

1) Computationally less expensive
2) Tracks really fast
3) Can be used with objects and hence, even if the detection is not present, the marker stays in the last position for 30 frames and disappears.
4) Can be used with a tracking algorithm like CSRT, KCF, MIL, where the detection of the ArucoMarker can be processed every (say) 5 frames which will improve the speed.
5) A Huge Number of Markers can be detected at once without any errors
6) Many other functionalities can be done using them. (Pose Estimation of a person etc)