

e-Yantra Robotics Competition (eYRC-2017)

Task1.1 – Collector Bot

ArUco Detection

ArUco markers have a distinct id and the ArUco library (refer **Tutorials**) gives us the pixel position of the four corners and the center position of a marker. So this information helps you in finding the position and orientation of a marker. However, there must be a relative scale to measure the orientation of the marker. Orientation of the marker can be measured with respect to horizontal or vertical position. Let's fix the horizontal position as 0° and increasing upto angle 359° in anti-clockwise direction as shown in Figure 1:

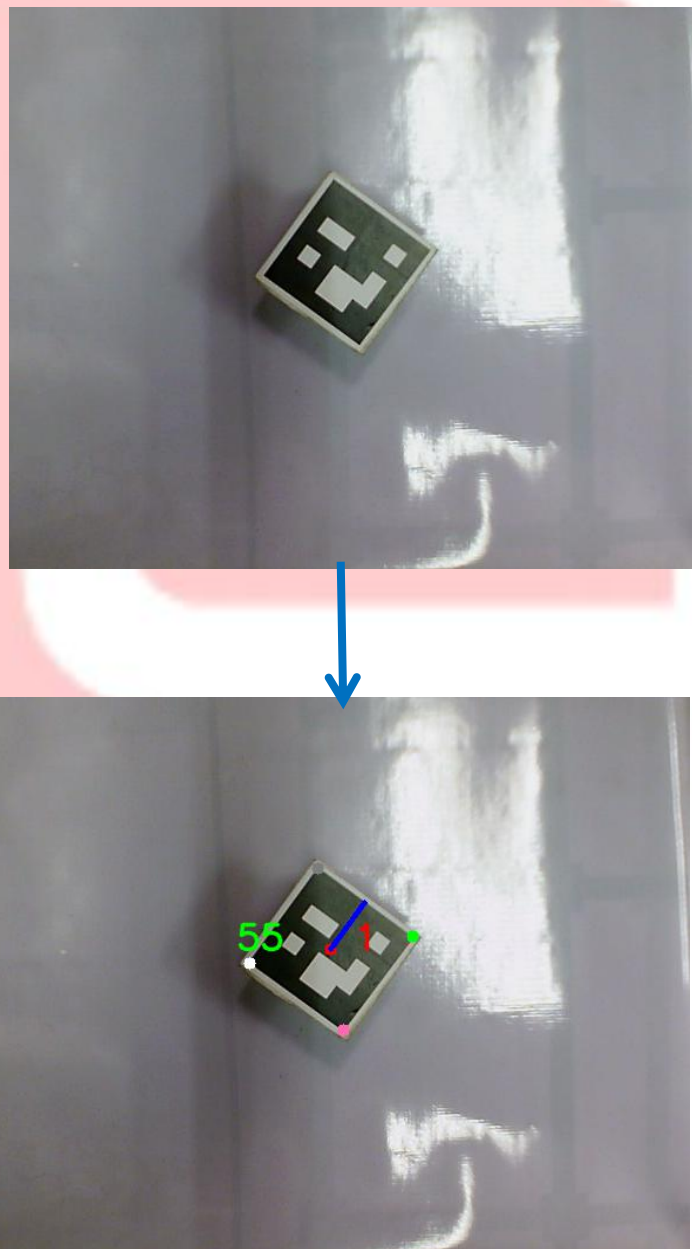


Figure 1: ArUco marker Detection

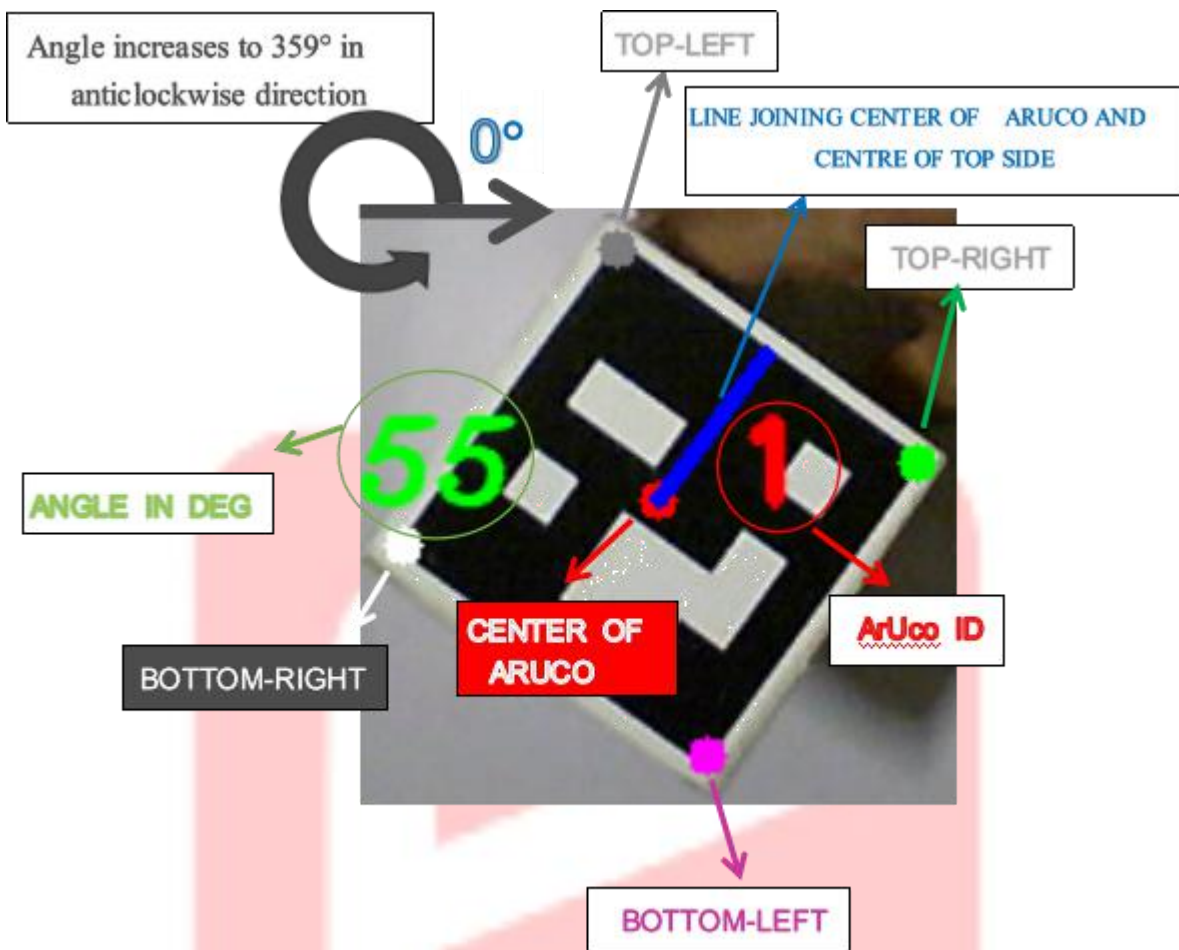


Figure 2: Magnified version of detected ArUco marker

Features in the resulting detected image as shown in Figure 2 are:

- Gray dot(125,125,125)(B,G,R) corresponds to top-left of the ArUco Marker
- Green dot(0,255,0) corresponds to the top-right of the ArUco Marker
- Pink dot(180,105,255) corresponds to the bottom-left of the ArUco Marker
- White dot(255,255,255) corresponds to the bottom-right of the ArUco Marker
- Red dot(0,0,255) corresponds to the center of the ArUco Marker
- Blue line (255,0,0) corresponds to the line joining between the centre of ArUco marker and the mid point of the line joining top-left and top-right. This line gives us the indication of the orientation of the ArUco. The angle between this line with the horizontal axis gives us the required angle.
- The upper thick black line indicates the horizontal line. Angle increases till 359° in anticlockwise direction. 55° is the angle between the blue line with this black line.
- Number in Red colour corresponds to the ArUco id
- Number in Green colour corresponds to the Angle of orientation of the ArUco marker
- Watch this [video](#) of ArUco detection for understanding the orientation scale better.

Problem Statement

Write Python script for Detecting ArUco markers . The Resulting image must have ArUco markers marked as shown in Figure1. Hence the resulting image will have ArUco markers with:

- Gray dot indicating top-left
- Green dot indicating top-right
- Pink dot indicating bottom-left
- White dot indicating bottom-right
- Red dot indicating center of the ArUco
- Blue line joining center of ArUco marker and the mid-point between top-left and top-right
- ArUco id number in RED colour
- Orientation of the ArUco in degrees in GREEN colour

Preparations

- Find Python script template *ArUco_library.py* and *ArUco_detection.py* in the folder **ProblemStatement**
- Find *Test_image_1.png* and *Test_image_2.png* in the folder **Test images**. These are the test images whose ArUco marker has to be detected and marked as shown in Figure1

Procedure

Step 1: Open *ArUco_library.py* present in **ProblemStatement** folder.

Step 2: Complete the following functions:

- ***detect_ArUco(img)***
 - Functionality
 - ◆ Detect id and corners of all the ArUco markers in the test image
 - Arguments:
 - ◆ *img*: the test image which needs to be tested
 - Return:
 - ◆ *Detected_ArUco_markers*: Dictionary in which each id (keys) corresponds to its corners(values).
- ***Calculate_orientation_in_degrees(Detected_ArUco_markers)***
 - Functionality
 - ◆ Calculate the orientation of all the ArUco markers (degrees) in the test image relative to the scale as mentioned in Figure 1.
 - Arguments:
 - ◆ *Detected_ArUco_markers*: the Dictionary returned by *detect_ArUco(img)* function
 - Return:
 - ◆ *ArUco_markers_angles*: Dictionary in which keys are the ArUco ids and the values are the corresponding angle of ArUco's in degree
- ***Mark_ArUco(img,Detected_ArUco_markers,ArUco_markers_angles)***

- Functionality
 - ◆ Mark ArUco markers in the test image as per the instructions given in Figure1.
- Arguments:
 - ◆ img: Test image
 - ◆ Detected_ArUco_markers: Dictionary returned by `detect_ArUco(img)` function
 - ◆ ArUco_markers_angles: Dictionary returned by `Calculate_orientation_in_degrees(Detected_ArUco_markers)`
- Return:
 - ◆ img: the resultant image after marking the ArUco's.

Step 3: After completing *ArUco_library.py*, run the Python script *Aruco_detection.py* which is present in the same folder as that of *ArUco_library.py*. Don't change anything in *ArUco_detection.py*. Once you successfully run *ArUco_detection.py* you will find two files namely *Result_image_1.png* and *Result_image_2.png* in the folder Test_images.

Step 4: Open the Result images to ensure that all the attributes that mentioned in Figure1 is marked on all the ArUco markers in the test images.

Note: For debugging purposes, download these videos of *Undetected ArUco markers.avi* and *Detected ArUco markers.avi* from this [google drive](#). Run your code to detect ArUco markers in the *Undetected ArUco markers.avi* and save it. Compare it with *Detected ArUco markers.avi* for debugging. However, make sure that you edit *ArUco_detection.py* to read the frame from video. Make sure you undo all the changes made in *ArUco_detection.py* after debugging to run the test images.

Submission Instructions

- Copy *ArUco_library.py*, *Result_image_1.png* and *Result_image_2.png* to a folder named <team_id>Task1_1.
- Instructions for uploading the task will be provided on the portal.