ROBxTASK ArUco detection

1. Implementation:

Camera - Logitech Business Brio Ultra HD Camera (**link**)

OS and ROS version : Ubuntu 16.04 (Xenial Xerus) and ROS1 (Kinetic)

Specification of camera frame opencv :

camera frame - 1980 * 1080 @ 30fps

Video conversion – opencv video writer fourcc [**MJPG]** format (to get better resolution and higher fps)

Frame conversion – BGR to gray scale to detect ArUco marker more accurately.

ArUco marker used is 5x5 and size of the marker 11.5 cm

opencv version '3.3.1-dev'

published rostopic of detected Aruco marker as list

- Id of the ArUco
- center_x pixel in camera frame of the detected ArUco marker
- center_y pixel in camera frame of the detected ArUco marker
- theta orientation or Yaw of the detected ArUco marker

How to run the script:

1. open new terminal enter the command “roscore”

2. In another terminal enter “rosrun aruco_recognition aruco_detection.py”

3. do “rostopic echo /aruco_data: to listen to the published data.

Things to remember:

1. Calibrate the camera before using. Check calibration folder for more insight which is located in “scripts/calibration”

2. The script is made for all vision system but right now it specifically build for Logitech BRIO to use other camera just change “cv2.VideoCapture(frame_cam)” to “cv2.VideoCapture(0)”.

2. Camera frame of recognized ArUco marker:

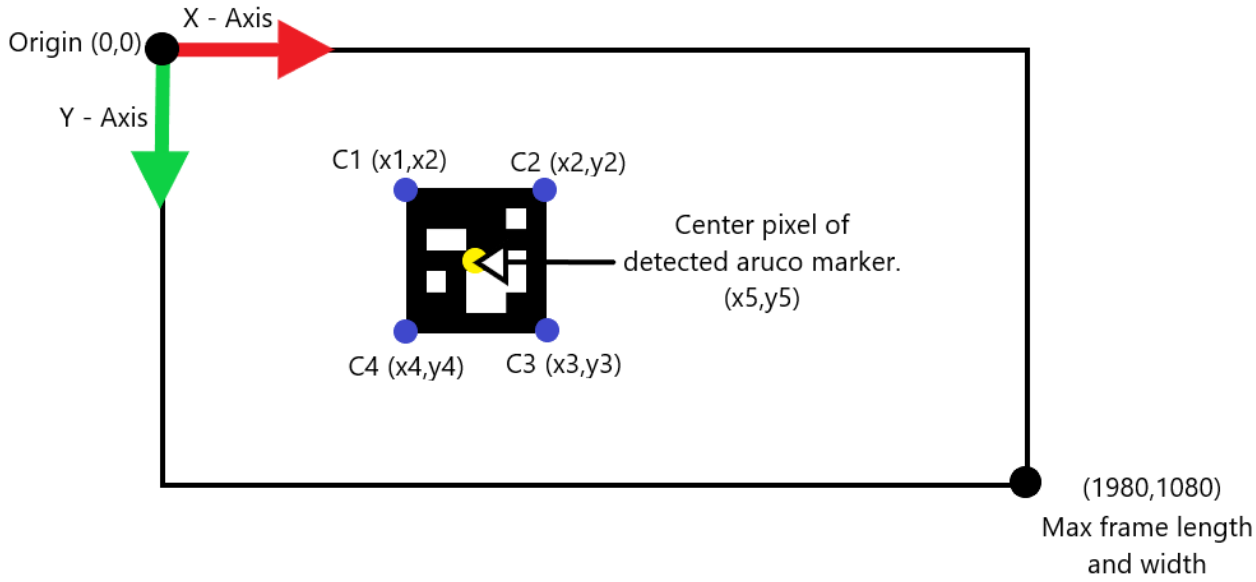


Figure 1 – Recognized ArUco marker

After detecting the ArUco marker we can get the 4 corners of the marker in the camera frame but we actually need the center pixel of the detected ArUco marker which is (x5,y5).

It can be found by;

$X5 = (x1+x2+x3+x4) * 0.25$

$Y5 = (y1+y2+y3+y4) * 0.25$

By above method we can easily get the detected center of ArUco marker. But we can't use the values straight away because the origin is in top right corner so we need to get the x5,y5 values according to the actual center of the camera frame.

We need:

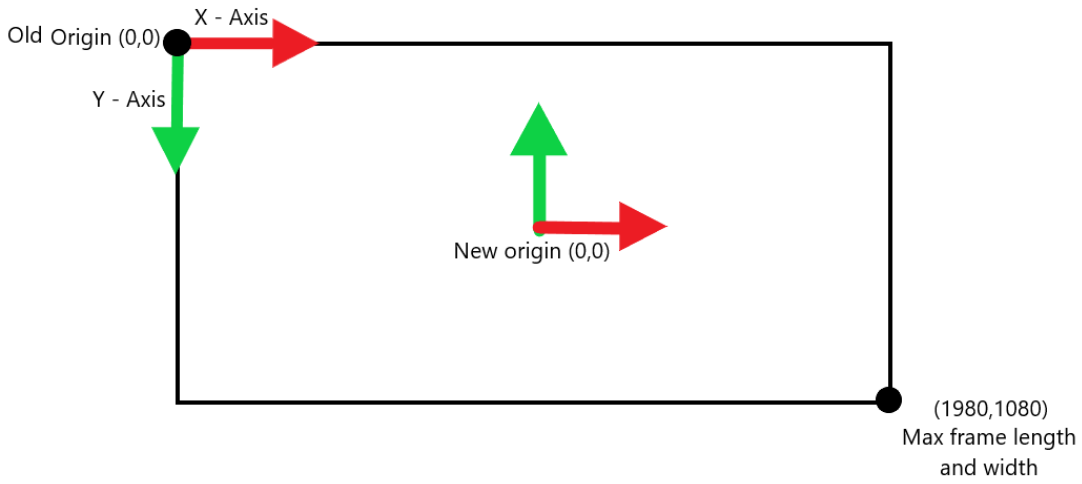


Figure 2: Expectation change of origin in camera frame

We know that new origin is in center of the camera frame so the center can of the frame can be found by:

New origin x axis = 1980/2 (which is max x pixel divided by 2 so we get center of the rectangle)

New origin y axis = 1080/2 * -1 (which is max x pixel divided by 2 so we get center of the rectangle)

Ref - For y axis we multiply with -1 because to invert. As we see in figure 1 Y-axis of the origin is increasing in downwards but we need to have the +ve along upwards ref-figure2

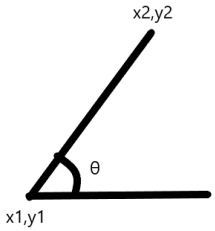
As we found the origin we need the detected ArUco center pixel according to the new origin:

New x5 = x5 – (new origin x)

New y5 = y5 – (new origin y)

3. To find the orientation of the detected ArUco marker:

Formula used to find the orientation / angle between 2 points:



$$\tan\theta = \frac{(y2-y1)}{(x2-x1)}$$

$$\theta = tan^{-1}\frac{(y2-y1)}{(x2-x1)}$$

Let **x1, y1 be the center pixel of detected ArUco marker which is [New x5,y5]** (ref figure – 1). Were x2,y2 is the midpoint of the C2 and C3.

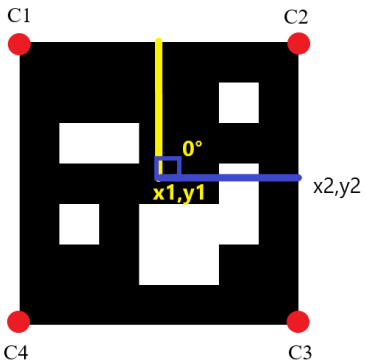


Figure 3 – Reference for 2 point required for orientation.

ROBxTASK ArUco detection

Lets keep corner 2 which is C2 (X2,Y2) and corner 3 is (X3,Y3). We need to find x2,y2 which can be done by formula:

x2 = ((X2 – X3)/2) + X3

y2 = ((Y2 – Y3)/2) + Y3

Though we the formula its not enough because we have huge complication which the above formula will only work if the ArUco is exactly by 90° facing up.

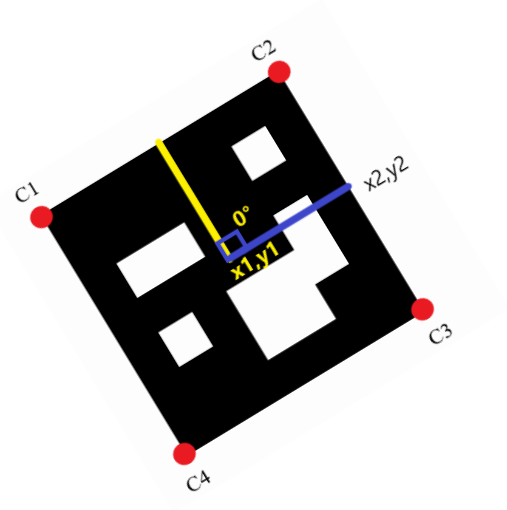


Figure 4 – Formula can't be applied with this orientation.

The above mentioned orientation won't be possible for the formula because X2 is smaller than X3 and same applies to y axis.

Solution:

The algorithm will be:

List of x in 2 corners = [X2,X3]

List of y in 2 corners = [Y2,Y3]

x2 = ((max(list of x) – min(list of x))/2) + min(list of x)

y2 = ((max(list of y) – min(list of y))/2) + min(list of y)

Then converting the points to the new origin:

New x2 = x2 – (1980/2)

New y2 = y2 – (1080/2) * -1

Finally applying:

Ref - **x1, y1 be the center pixel of detected ArUco marker which is [New x5,y5]**

$\theta = tan^{-1} \frac{(New\ y2 - y1)}{(\text{New x2} - \text{x1})}$