Angle between 2 ArUco markers planes

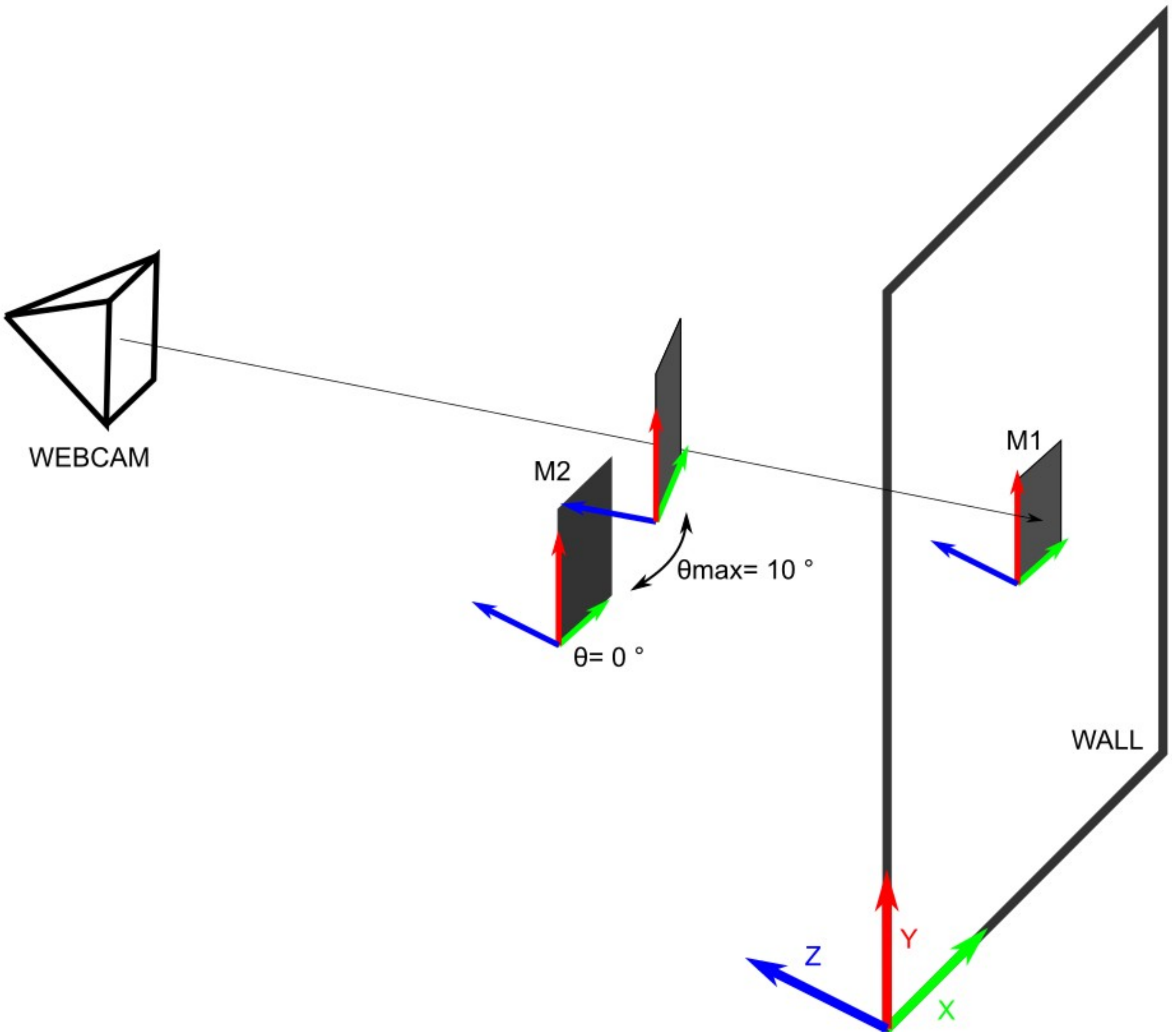
Asked 1 year, 5 months ago Modified 1 year, 5 months ago Viewed 2k times

2

I want to measure the deviation of the angle of an ArUco marker to a plane defined by a second reference ArUco marker.

A reference ArUco marker (M1) is fixed against a flat wall and a second ArUco marker (M2) is a few centimeters in front of that same wall. I want to know when the marker M2 is deviating more than 10 degrees from the xy plane of M1.

Here is an illustration of the configuration:

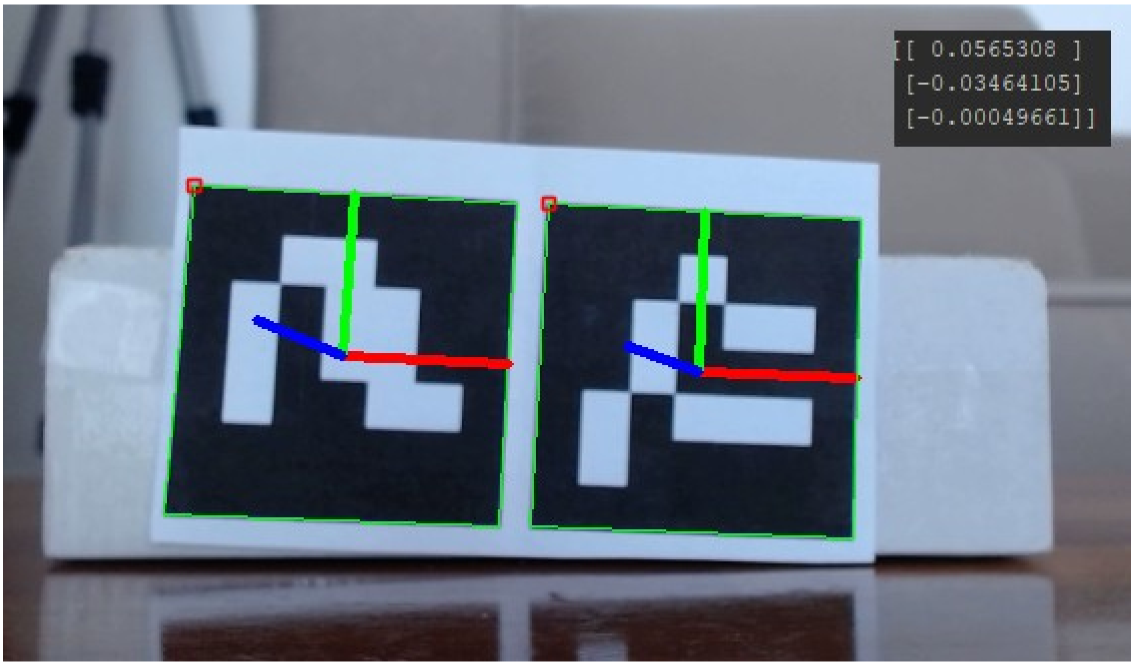


To do so, I thaught I should calculate the relative rotation between the pose rvec as explained

```
def inversePerspective(rvec, tvec):
""" Applies perspective transform for given rvec and tvec. """
    R, _ = cv2.Rodrigues(rvec)
    R = np.matrix(R).T
    invTvec = np.dot(R, np.matrix(-tvec))
    invRvec, _ = cv2.Rodrigues(R)
    return invRvec, invTvec



def relativePosition(rvec1, tvec1, rvec2, tvec2):
""" Get relative position for rvec2 & tvec2. Compose the returned rvec & tvec to
use composeRT
with rvec2 & tvec2 """
    rvec1, tvec1 = rvec1.reshape((3, 1)), tvec1.reshape((3, 1))
    rvec2, tvec2 = rvec2.reshape((3, 1)), tvec2.reshape((3, 1))

    # Inverse the second marker, the right one in the image
    invRvec, invTvec = inversePerspective(rvec2, tvec2)

    info = cv2.composeRT(rvec1, tvec1, invRvec, invTvec)
    composedRvec, composedTvec = info[0], info[1]

    composedRvec = composedRvec.reshape((3, 1))
    composedTvec = composedTvec.reshape((3, 1))
    return composedRvec, composedTvec
```

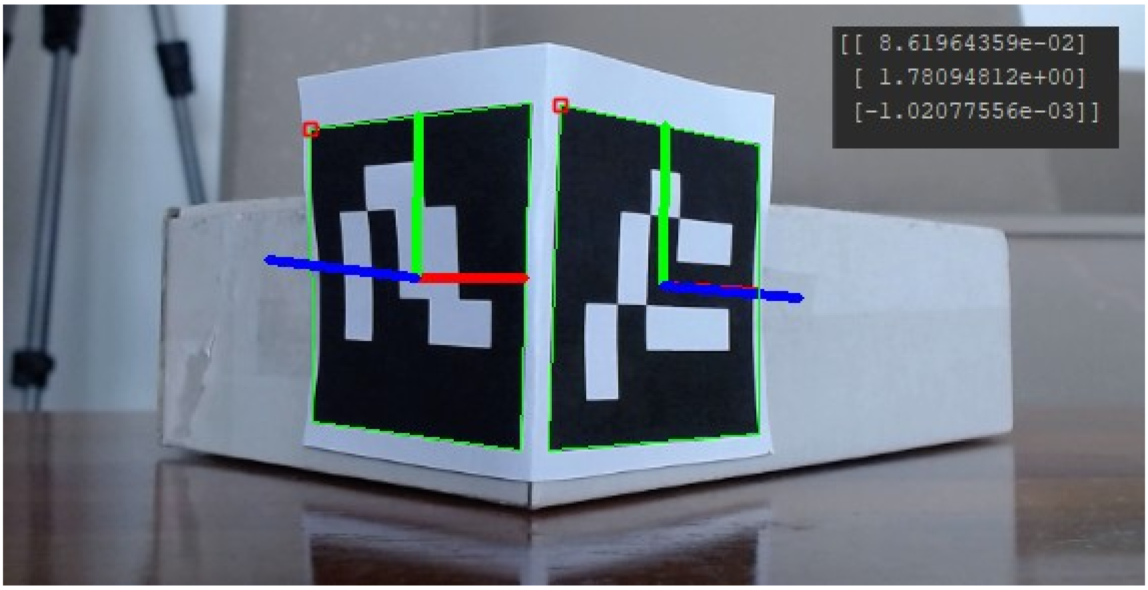
Computing the composedRvec, I get the following results :

With both ArUco markers in the same plane (composedRvec values in the top right corner) :



```
[[ 0.0565308 ]
 [-0.03464105]
 [-0.00049661]]
```

With both ArUco markers at a 90 degrees angle:



```
[[ 8.61964359e-02]
 [ 1.78094812e+00]
 [-1.02077556e-03]]
```

I do not really understand the results:

Ok for with the 0,0,0 composedRvec when markers in the same plane.

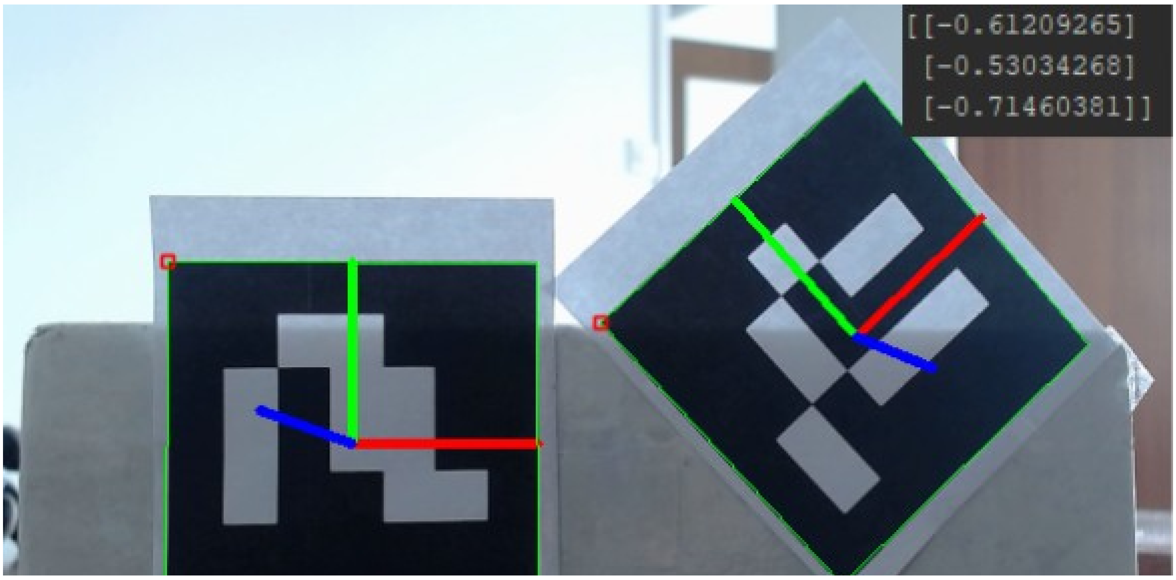
But why 0,1.78,0 in the second case?

What general condition should I have on the resulting composedRvec to tell me when the angle between the 2 markers is above 10 degrees?

Am I even following the right strategy with the composedRvec?

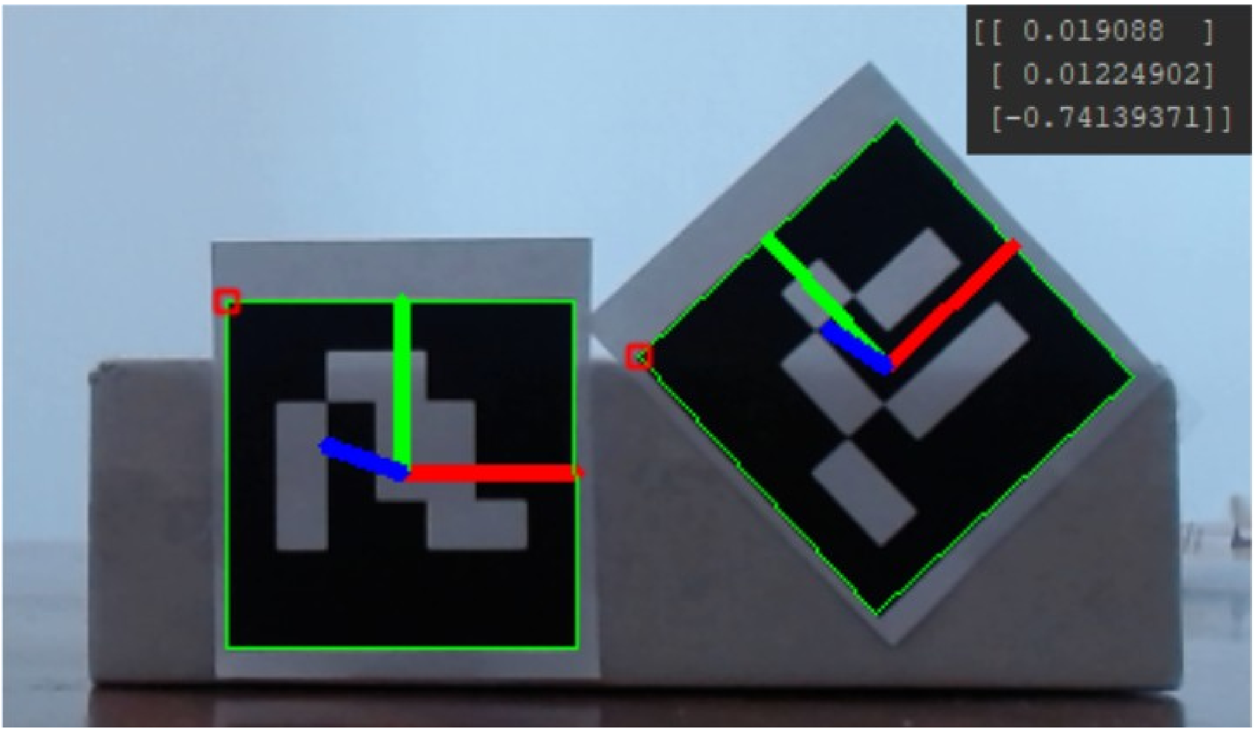
**** EDIT ***

Results of the 2 markers in the same xy plane with a 40° angle:



```
[[-0.61209265]
 [-0.53034268]
 [-0.71460381]]
```

By retaking measurements in the 40° angle configuration, I found out that the values are quite fluctuating even without modifying the set up or lightening. From time to time, I fall on the correct values:



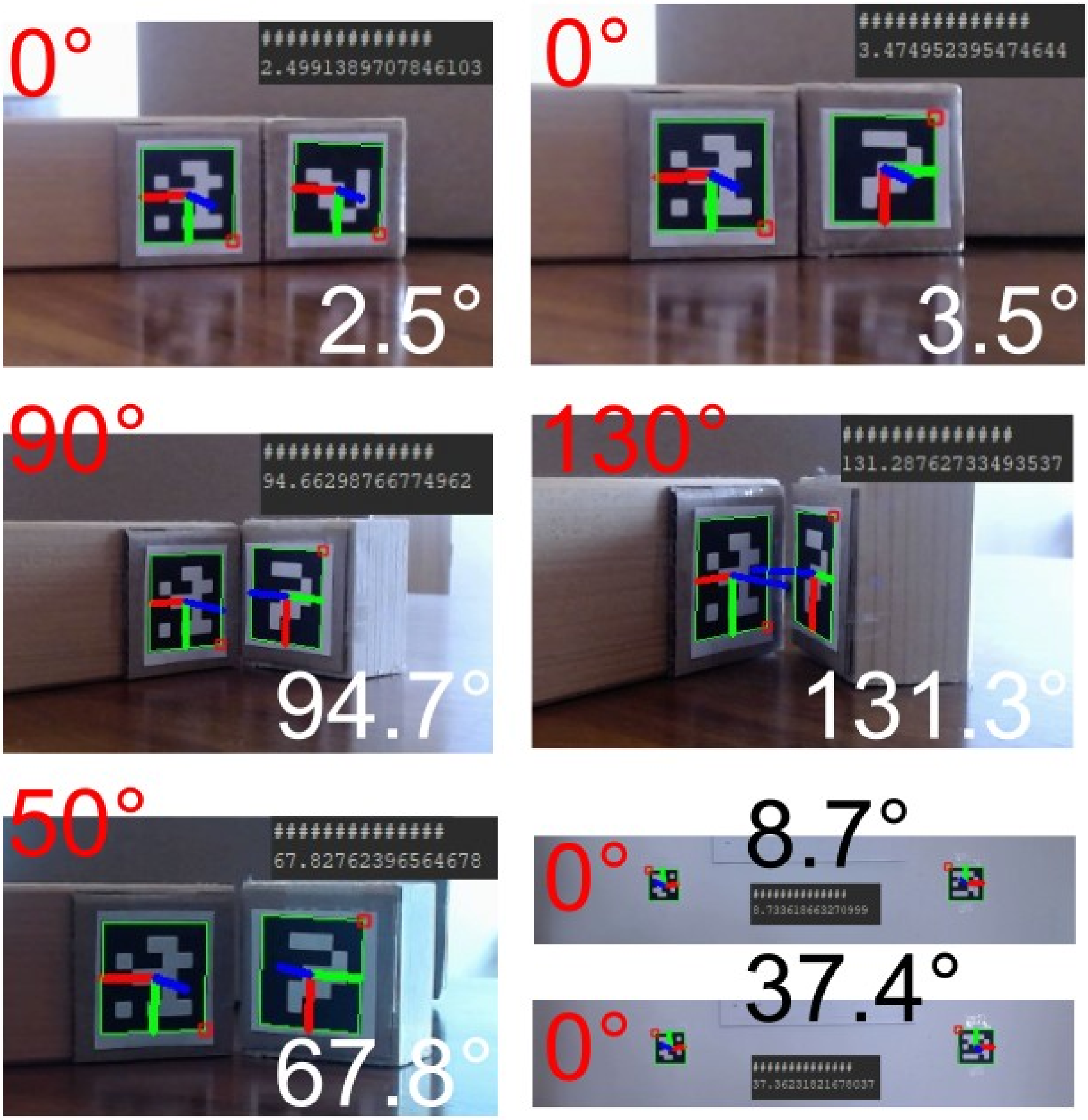
```
[[ 0.019088  ]
 [ 0.01224902]
 [-0.74139371]]
```

||composedRvec||= sqrt(0.019^2+0.012^2+0.74^2)=0.74 rad = 42.4° which is quite accurate.

**** EDIT 3 ***

So here is my final code based on @Gilles-Philippe Paillé's edited answer:

```
import numpy as np
import cv2
import cv2.aruco as aruco


cap = cv2.VideoCapture(0, cv2.CAP_DSHOW)  # Get the camera source
img_path='D:/your_path/'
# FILE_STORAGE_READ
cv_file = cv2.FileStorage(img_path+"camera.yml", cv2.FILE_STORAGE_READ)
matrix_coefficients = cv_file.getNode("K").mat()
distortion_coefficients = cv_file.getNode("D").mat()

nb_markers=2

def track(matrix_coefficients, distortion_coefficients):
    while True:

        ret, frame = cap.read()
        # operations on the frame come here
    # store rz1 and rz2
    R_list=[]

    if np.all(ids is not None):  # If there are markers found by detector
        for i in range(0, len(ids)):  # Iterate in markers

        # Estimate pose of each marker and return the values rvec and tvec---
different from camera coefficients
            rvec, tvec, markerPoints =
aruco.estimatePoseSingleMarkers(corners[i], 0.02, matrix_coefficients,

distortion_coefficients)
            (rvec - tvec).any()  # get rid of that nasty numpy value array error



            aruco.drawDetectedMarkers(frame, corners)  # Draw A square around the
markers

            aruco.drawAxis(frame, matrix_coefficients, distortion_coefficients,
rvec, tvec, 0.01)  # Draw Axis


            R, _ = cv2.Rodrigues(rvec)
            # convert (np.matrix(R).T) matrix to array using
np.squeeze(np.asarray()) to get rid off the ValueError: shapes (1,3) and (1,3)
not aligned
            R = np.squeeze(np.asarray(np.matrix(R).T))
            R_list.append(R[2])


        # Display the resulting frame


    if len(R_list) == 2:


        print('##############')
        angle_radians = np.arccos(np.dot(R_list[0], R_list[1]))
        angle_degrees=angle_radians*180/np.pi
        print(angle_degrees)


    cv2.imshow('frame', frame)
#   Wait 3 milisecoonds for an interaction. Check the key and do the
corresponding job.
    key = cv2.waitKey(3000) & 0xFF
    if key == ord('q'):
     break



track(matrix_coefficients, distortion_coefficients)
```



This is out of the scope of the question but I find that the pose estimation is quite fluctuating. For example when the 2 markers are against the wall, the values easely jump from 9° to 37° without touching the system.

python opencv geometry aruco

Share Improve this question Follow

edited Aug 31, 2021 at 13:06

asked Aug 27, 2021 at 8:22

michltm

1,379 2 13 33

by the way, how did you do the first picture, the illustration? – Christoph Rackwitz Aug 27, 2021 at 18:25

1 @ChristophRackwitz : got it, thanks. the illustration is made with inkscape: inkscape.org – michltm Aug 28, 2021 at 10:01

1 Answer

Sorted by: Highest score (default)

2

The result uses the Angle-axis representation, i.e., the norm of the vector is the angle of rotation (what you want), and the direction of the vector is the axis of rotation.

You are looking for `θ = ||composedRvec||` . Note that the result is in radians. The condition would be `||composedRvec|| > 10*π/180` .

Edit: To only consider the angle between the Z-axis of both planes, convert the two rotation vectors `rvec1` and `rvec2` into matrices and extract the 3rd columns. The angle is then `angle_radians = np.arccos(np.dot(rz1, rz2))`

Share Improve this answer Follow edited Aug 28, 2021 at 13:53 answered Aug 27, 2021 at 18:24

Gilles-Philippe Paillé
3,078 1 7 20

I guess I did not completely get. If I rotate te 2 markers in the same xy plane as in the Edit section, I would want to read 0 since they are in the same plane and the ||composedRvec|| gives me 61.87 degrees. – michltm Aug 28, 2021 at 11:04

@michltm If the rotation is only in the Z-axis, it doesn't make sense that the rotation vector has non-zero components for the X and Y coordinates. What do you provide to `relativePosition` ? However, I agree that simply taking the norm doesn't handle this case. I'll edit my answer. – Gilles-Philippe Paillé

I am not sure on how to get rz1 . is it R1, _ = cv2.Rodrigues(rvec1) and then rz1=R1[2] ? – michltm

Aug 28, 2021 at 17:22

Just as a precision, I get rvec with the following code: rvec, tvec, markerPoints = aruco.estimatePoseSingleMarkers(corners[i], 0.02, matrix_coefficients, distortion_coefficients)
– michltm Aug 29, 2021 at 7:24