CV ASSIGNMENT 1

Siya Puttagunta

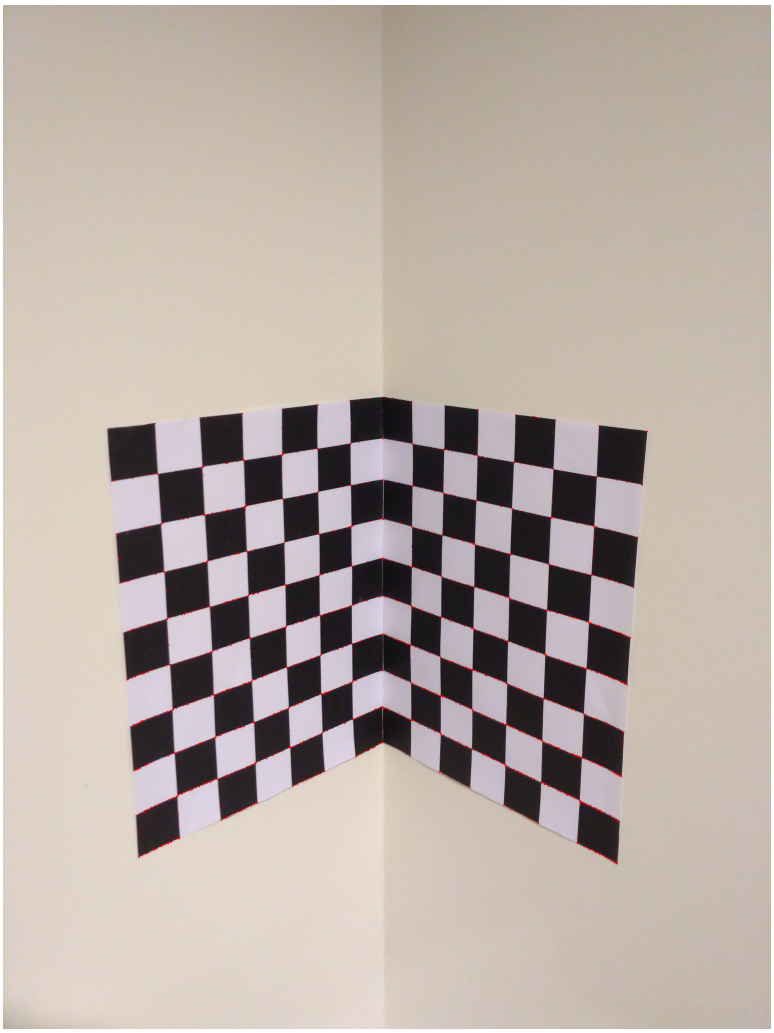
2021101062

Q1

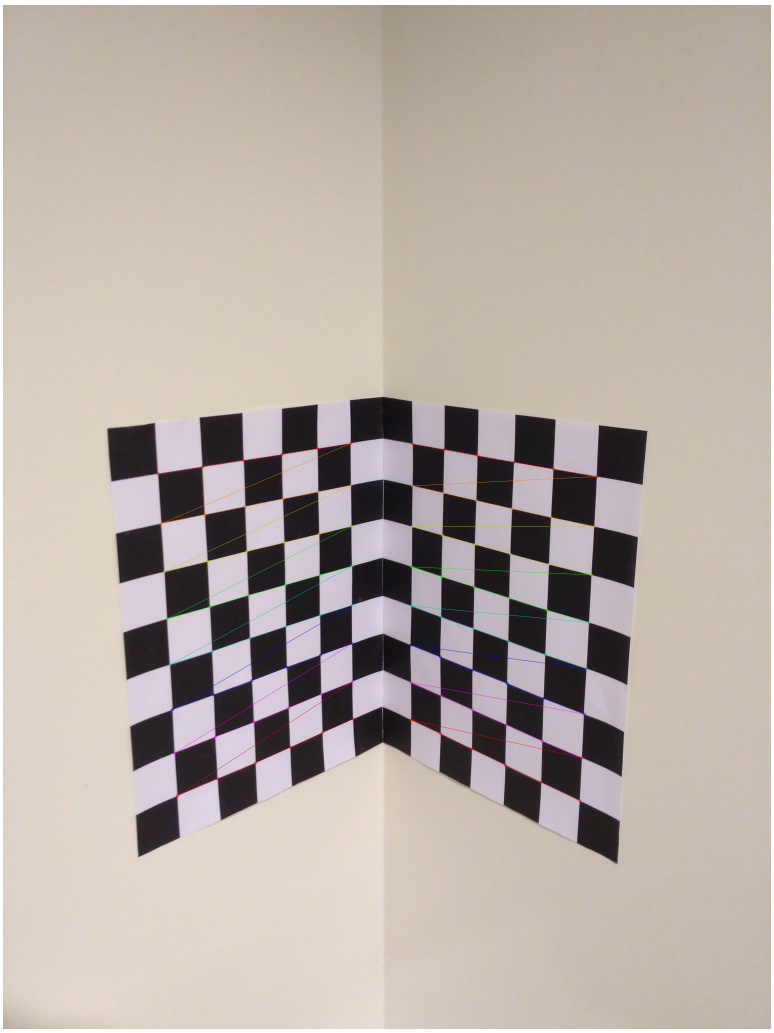
1

Detection of internal corners is done using 2 methods: Harris Corners and findChessboardCorners().

Harris Corners



findChessboardCorners()



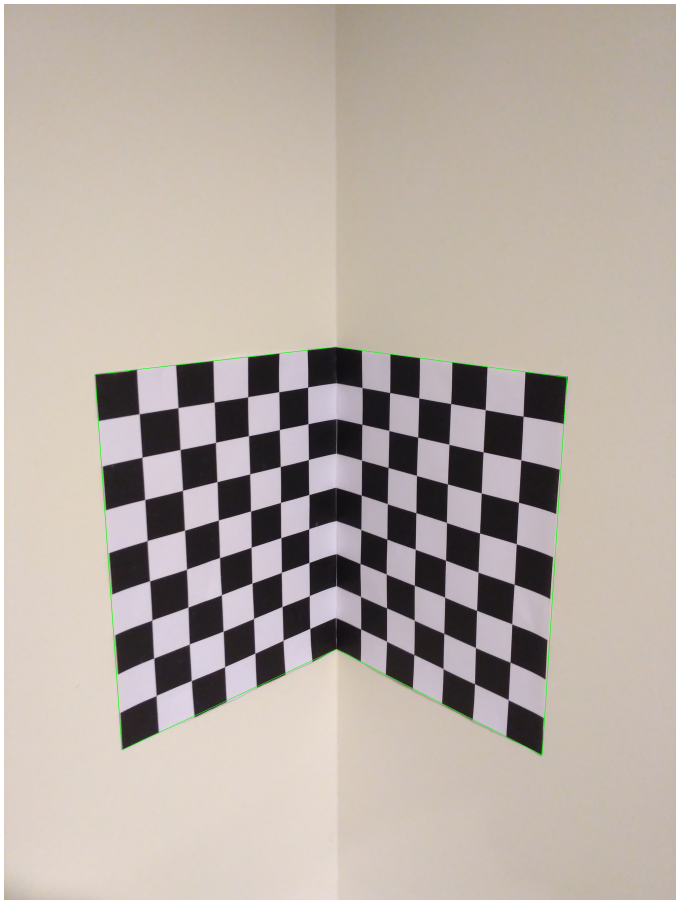
2

Take a set of world coordinates and their corresponding image coordinates. Here, the world origin is assumed to be the bottom most point (internal corner) on the edge at which both the walls meet. Use the Direct Linear Transform method to find the Projection matrix. It is the last column vector of V obtained by performing SVD on M.

K (Intrinsics Matrix) and R (Rotation Matrix) can be found out by performing QR decomposition on M. The translation vector can be found out by performing an inverse operation.

3

Image coordinates can be found out by multiplying projection matrix with the world coordinates. Image coordinates of the 6 external corners can be found out using this method. Then these 6 corners can be used to draw a wireframe over the actual image which acts like a boundary of the image. The overlay fits perfectly over the image.



4

The rotation angles are the angles of rotation about the x, y and z axes. The rotation about the x axis is tilt, the rotation about the y axis is pan and the rotation about the z axis is roll.

Pan(rotation about x-axis), tilt(rotation about y-axis) and roll(rotation about z-axis) can be found out from the rotation matrix using their trigonometric relations. All of these

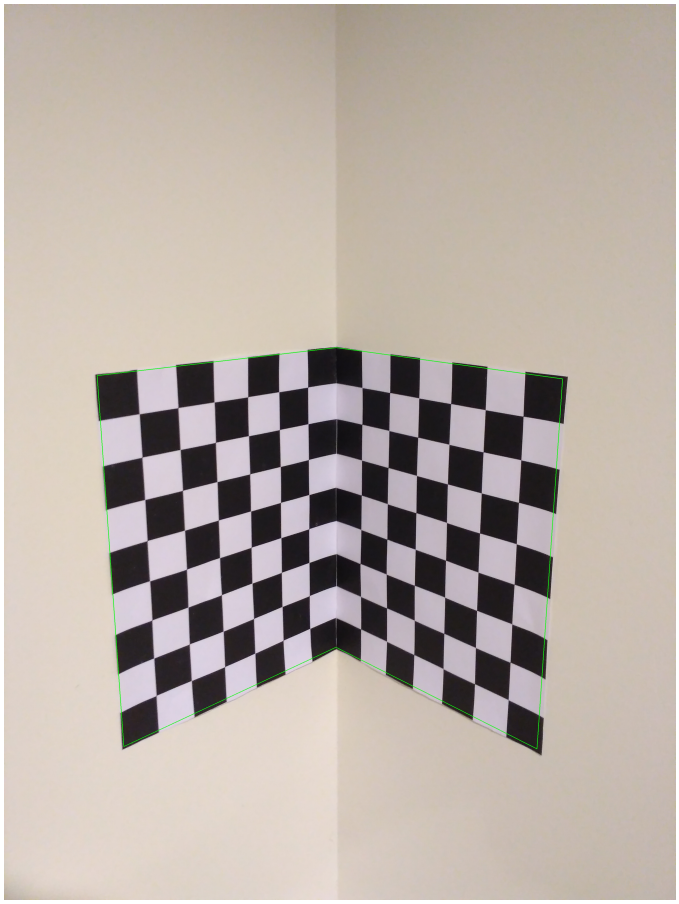
angles are found using the ZYX Euler angle convention. Pan, tilt and roll tell us about the orientation of camera with respect to the world origin.

Q2

1

Since the given chessboard is non-planar, an initial estimate of the intrinsics matrix has to be given to calibrateCamera() in order to get the actual K matrix. This estimate is found out by assuming that the world coordinates are planar. Once the values of K, R and t are known then, projection matrix P can be found out using the formula: P = K[R|t].

Now taking the world coordinates of external corners, a wireframe can be overlayed on the image using the image coordinates. And these image coordinates are calculated using the formula x = PX (x = image coordinate and X = world coordinate).



It is not as accurate as the result obtained in Q1 but it almost overlays the image. There is a slight distortion.

2

Here the given chessboard is planar. Image coordinates are found out using findChessboardCorners() and this is given as input to calibrateCamera(). From this, we get the values of K, R and t. Now, the above mentioned method is used to draw the wireframe on this image. It overlays perfectly and doesn't show any distortion.

Solution will not be unique in case of co-planar coordinates and this will lead to numerical instability (such as division by zero) in the projection matrix. This is equivalent to losing some degrees of freedom.

3

Here, the world origin is assumed to be the top-left internal corner. The image of world origin is the last column vector of projection matrix P (p4). This can be found out using the formula P * [0, 0, 0, 1].T. The coordinate found should be homogenised since it is on the image plane.

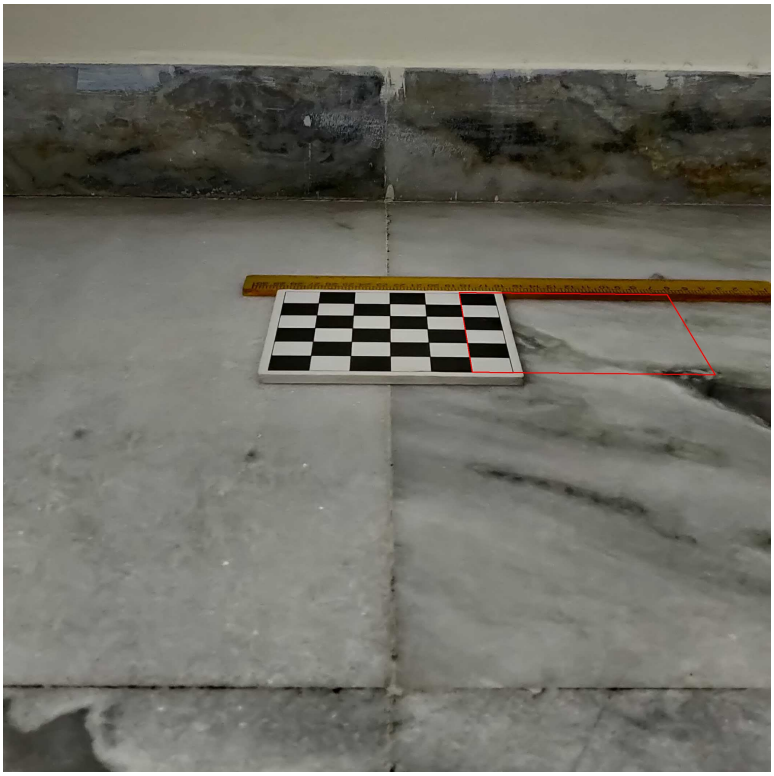
Yes, the image of world origin perfectly agrees with the observations.

Q3

1

Now, we have to image that the chessboard is shifted by 10cm to the right. Accordingly, the wireframe has to be drawn. In order to achieve this, just shift the world coordinates

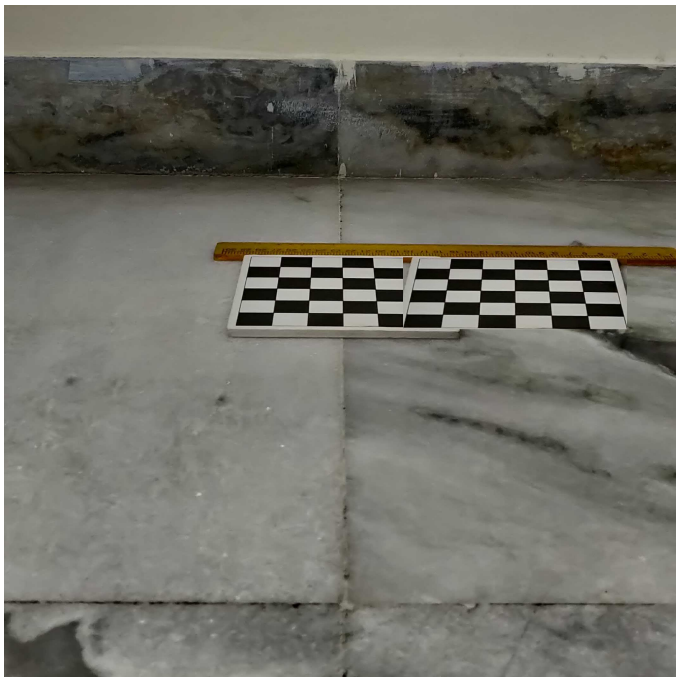
of the external corners by 10 cm along the X-axis.



The wireframe looks good and is consistent with our expectations.

2

Now, we have to shift the whole chessboard pattern such that it fits into the new shifted wireframe as shown above (10 cm shifted to the right). The final image contains both the old chessboard position and the new chessboard position.



Yes, the overlay is consistent with our expectations. There is a slight distortion.