Akshay Bankar
2019201011

Assignment 1 : Camera Caliberation

Input and Output images :

https://iiitaphyd-my.sharepoint.com/:f:/g/personal/akshay_bankar_students_iiit_ac_in/Ervuw-ETcJBGgALyMG5s1IwBRGzFtKlG8-4YztVEiCMhTg

Distortion in camera :

Distortion :

Pinhole cameras introduce significant distortion to images. Two major kinds of distortion are radial distortion and tangential distortion.

1. Radial distortion causes straight lines to appear curved. Radial distortion becomes larger the farther points are from the center of the image.

Radial distortion can be represented as follows:

```
xdistorted=x(1+k1r^2+k^2r^4+k3r^6)
ydistorted=y(1+k1r^2+k2r^4+k3r^6)
```

2. Tangential distortion occurs because the image-taking lense is not aligned perfectly parallel to the imaging plane. So, some areas in the image may look nearer than expected. The amount of tangential distortion can be represented as below:

```
xdistorted=x+(2*p1*x*y+p2(r^2+2*x^2))
ydistorted=y+ p1(r^2+2*y^2)+2*p2*xy)
```

We need to find five parameters, known as distortion coefficients given by:

```
Distortioncoefficients=(k1,k2,p1,p2,k3)
```

We need to some other information, like the intrinsic and extrinsic parameters of the camera. Intrinsic parameters are specific to a camera. They include information like focal length (fx,fy) and optical centers (cx,cy).

```
cameramatrix=[[fx,0,cx],[0,fy,cy],[0,0,1]]
```

Extrinsic parameters corresponds to rotation and translation vectors which translates a coordinates of a 3D point to a coordinate system.

Akshay Bankar
2019201011

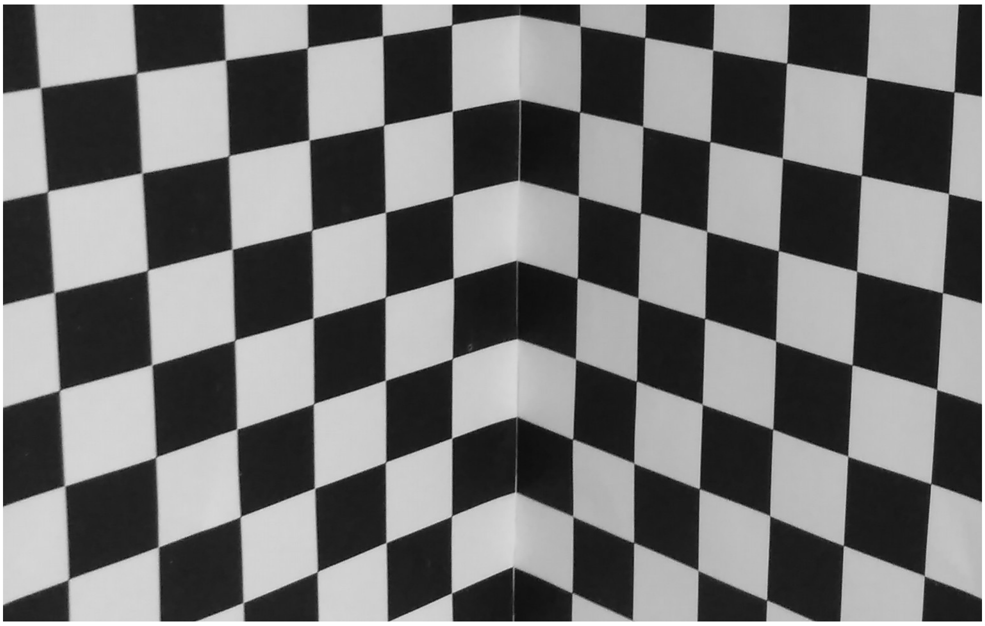
1. Camera Caliberation using Direct Linear Transform (DLT)

1. For the given image, find the corners of the chessboard squares in the image. These correspond to 2D image points. These points should be on different planes of the 3D object.
2. Normalize these image points using L2-norm.
3. Define world points corresponding to the 2D image points defined in step-1.
4. For part-1, get 6 image points randomly and the corresponding world points.
5. For part-2, find 6 points from the defined image and world points such that they minimize the projection error.

```
Minimum error :
 2308.8102858326415

Projection matrix :
 [[-2.72067705e-02 -7.57862780e-03  7.76554901e-02 -6.89944949e-01]
 [ 1.91237521e-02 -8.31631660e-02  1.73892155e-02 -7.13837188e-01]
 [ 1.62361154e-05 -4.86507771e-06  1.48200532e-05 -4.48015184e-04]]

Camera matrix :
 [[3.36210511e+03 1.68710761e+01 1.47164263e+03]
 [0.00000000e+00 3.35838256e+03 1.91906130e+03]
 [0.00000000e+00 0.00000000e+00 1.00000000e+00]]

Rotation matrix for 1st image:
 [[-6.74268232e-01 -6.35400319e-04  7.38486254e-01]
 [-1.59156767e-01 -9.76374586e-01 -1.46156736e-01]
 [ 7.21132078e-01 -2.16083929e-01  6.58237238e-01]]

Translation matrix for 1st image :
 [ -0.41428073   1.92997331 -19.8987327 ]
```

Finding distortion parameters:

- Define image points corresponding to ixel positions and corresponding world points.
- Using OpenCV's caliberateCamera() find camera matrix, rotation matrix, translation matrix,
- Undistort the image using OpenCV's undistort().

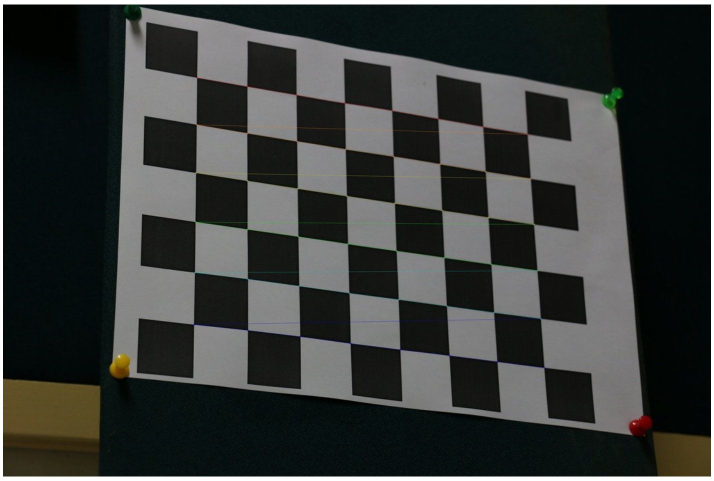


Akshay Bankar
2019201011

2. Camera Caliberation using Zhangs' method

Steps :

1. Read 12 chessboard pattern images each has chessboard of 9x7 boxes.
2. Create 3-D image points with z=0 plane, which would correspond to 2-D image corners on the chessboard.
3. Find the corners of the chessboard which would correspond to coordinates of image plae.
4. Draw these corners on the images (using opencv function).



5. Find the camera matrix, distortion coefficients, rotation and translation vectors for each image provided.

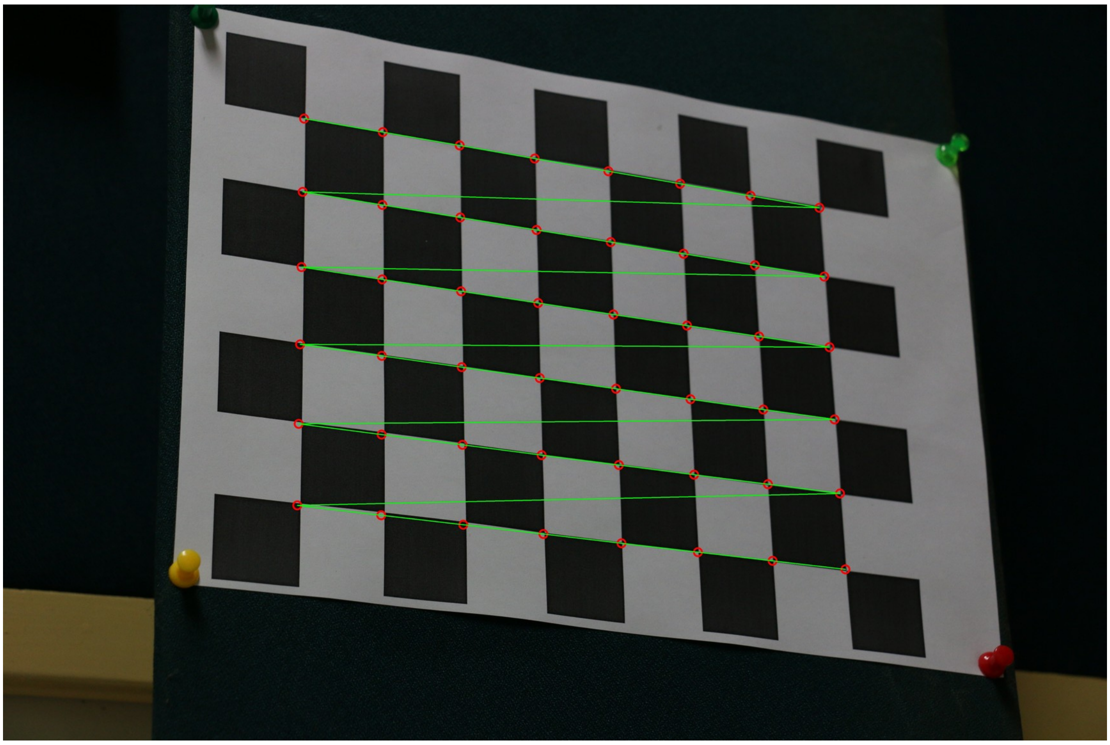
```
Camera matrix :
 [[1.36415094e+04 0.00000000e+00 3.31635881e+03]
 [0.00000000e+00 1.36632517e+04 1.50037396e+03]
 [0.00000000e+00 0.00000000e+00 1.00000000e+00]]
Distortion matrix :
 [[ 9.79057908e-02  9.45876425e+00 -1.53012034e-02  2.72096493e-02
  -1.48434591e+02]]
Rotation matrix for 1st image:
 [[-0.01040574]
 [-0.04719484]
 [-0.00305521]]
Translation matrix for 1st image :
 [[-5.09954028]
 [-2.0599507 ]
 [30.85011107]]
```

The image of world origin is is the 4th column of the 3x4 caliberation matrix

6. Using the camera matrix, distortion coefficients, rotation and translation vector found above, project the 3-D world points on the image to get the chessboard corners.

7. Draw the wireframe on the image to show the projected points.

Akshay Bankar
2019201011



Akshay Bankar
2019201011

3. Caliberating on my phone's camera

1. Using DLT :

```
Projection matrix :
 [[ 3.52602650e-02 -5.25156018e-03 -2.64884039e-03  5.74329051e-01]
 [ 1.74773590e-02  1.44533358e-02 -2.97485750e-02  8.16987643e-01]
 [ 9.54843720e-06  8.03218437e-06 -1.58616133e-06  3.40695793e-04]]

Camera matrix :
 [[ 2.12425397e+03  1.25564994e+01  1.88806467e+03]
 [-0.00000000e+00 -2.11893506e+03  2.08691450e+03]
 [ 0.00000000e+00  0.00000000e+00  1.00000000e+00]]
Rotation matrix for 1st image:
 [[-0.64440484  0.7646517  -0.0070836 ]
 [-0.0919041  -0.08664117 -0.9919914 ]
 [-0.75914164 -0.63859305  0.12610662]]
Translation matrix for 1st image :
 [  2.55620581   3.9766361  -27.08677442]
```

2. Using Zhangs method :

```
Camera matrix :
 [[3.27054542e+03 0.00000000e+00 1.63429944e+03]
 [0.00000000e+00 3.30205443e+03 2.06217865e+03]
 [0.00000000e+00 0.00000000e+00 1.00000000e+00]]
Distortion matrix :
 [[ 1.53016399e-01 -6.77183950e-01 -9.99885889e-04  2.44330084e-02
   3.74743657e+00]]
Rotation matrix for 1st image:
 [[ 0.0655073 ]
 [-0.03135368]
 [ 0.01567144]]
Translation matrix for 1st image :
 [[-3.80320166]
 [-3.66367848]
 [16.29104669]]
```