Computer Vision Assignment I - Report

AadilMehdi Sanchawala
24-01-2020
Assignment 1
20171043 `aadilmehdi.s@students.iiit.ac.in - 20171043`

1 Introduction

In the following report we demonstrate Camera Calibration techniques using two popular methods namely the Direct Linear Transform method and the Zhang's Calibration Method. We use RANSAC with DLT estimation in order to refine our estimate of the Projection Matrix. We estimate the Projection matrix P and decompose the same into Rotation Matrix R, Translation Vector t, and Camera Intrinsic Matrix K. We calibrate the camera for the images provided in the question and also for the images taken from a phone camera.

2 Direct Linear Transform with RANSAC

We use DLT to compute the orientation of an uncalibrated camera using ≥ 6 known points (≥ 6 world coordinates to image coordinates correspondences are known).

$$x_i = P * X_i \tag{1}$$

where $i = 1, ..., n$

$$\begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} * X_i = \begin{bmatrix} A^T \\ B^T \\ C^T \end{bmatrix} * X_i \tag{2}$$

2.1 DLT

We construct a matrix M of size 2*N $\times$ 12 using the following,
$a_{xi}^T = (-X_i, -Y_i, -Z_i, -1, 0, 0, 0, 0,$
$x_iX_i, x_iY_i, X_iZ_i, x_i)$
$a_{yi}^T = (0, 0, 0, 0, -X_i, -Y_i, -Z_i, -1,$
$x_iX_i, x_iY_i, X_iZ_i, x_i)$

We reshape our projection matrix into a 12$\times$1 vector p. Thus rewriting equation (1), for each point we get, $a_{xi}^T p = 0$
$a_{yi}^T p = 0$

We use SVD of M to solve the algebraic minimisation problem.

```
U, s, V_T = svd(M)
```

We take the 12^{th} (last) column of V^T and reshape it into 3 $\times$ 4 matrix and this is our estimate of the projection matrix P.

2.2 Refining our estimate of P using RANSAC

We use RANSAC to better our estimate of the projection matrix. For this we randomly select 6 pairs of corresponding world and image corrdinates and use them to calculate the projection matrix. We then minimize the projection error, $\epsilon = \frac{1}{N}$ $\sum_{i=1}^{N}(x_i - x_i^{'})^2$

2.3 Decomposing P into Rotation, Translation, and Intrinsic Matrix

$$P = KR[I| - X_0] \tag{3}$$

where K is Intrinsic Matrix, R is Rotation Matrix and X_0 is the Camera position in the world.

We define $H_\infty = KR$ and $h = -KRX_0$
Therefore, $t = -H_\infty^{-1}h$ and R and T can be obtained from RQ decomposition of H_∞.

2.4 Experimentation

We take 50 corresponding points and estimate the Camera parameters. RANSAC is run for 50,000 iterations.

2.5 Results

The following are the Camera Parameters obtained from DLT without RANSAC.

$$P = \begin{bmatrix} 2.355e+00 & 5.987e-01 & -6.160e+00 & 1541 \\ -1.384e+00 & 6.687e+00 & -1.259e+00 & 1595 \\ -1.197e-03 & 3.832e-04 & -1.102e-03 & 1.000 \end{bmatrix}$$

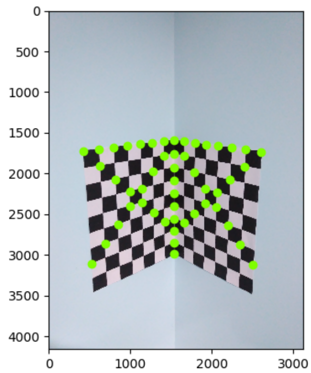


Figure 1: Projected Points

$$R = \begin{bmatrix} -0.6778 & -0.0018 & 0.7352 \\ -0.1673 & -0.9733 & -0.1567 \\ 0.7159 & -0.2292 & 0.6594 \end{bmatrix}$$

$$K = \begin{bmatrix} 3.664e+03 & 6.857e+00 & 1.503e+03 \\ 0.000e+00 & 3.636e+03 & 2.006e+03 \\ 0.000e+00 & 0.000e+00 & 1.000e+00 \end{bmatrix}$$

$$t = \begin{bmatrix} 435.26848437 \\ -71.28428592 \\ 409.66228975 \end{bmatrix}$$

The mean-squared projection error is 9.978
The following are the Camera Parameters obtained from DLT using RANSAC for 50,000 iterations.

$$P = \begin{bmatrix} 2.345e+00 & 6.051e-01 & -6.208e+00 & \\ -1.366e+00 & 6.710e+00 & -1.336e+00 & \\ -1.195e-03 & 3.891e-04 & -1.150e-03 & 1.000 \end{bmatrix}$$

$$R = \begin{bmatrix} -0.693 & 0.002 & 0.720 \\ -0.166 & -0.973 & -0.156 \\ 0.701 & -0.228 & 0.675 \end{bmatrix}$$

$$K = \begin{bmatrix} 3.578e+03 & 4.047e+00 & 1576.018 \\ 0.000e+00 & 3.577e+03 & 1990.917 \\ 0.000e+00 & 0.000e+00 & 1.000 \end{bmatrix}$$

$$t = \begin{bmatrix} 426.27640483 \\ -70.45749168 \\ 402.38057958 \end{bmatrix}$$

The mean-squared projection error is 21.045

2.6 Correcting for Radial Distortion

The following OPENCV functions were used after getting an initial estimate of the Intrinsic matrix in order to obtain the distortion parameters and correct for radial distortion.

```
ret, mtx, dist, rvecs, tvecs =
cv2.calibrateCamera(
[world_coords[:,:3].astype('float32')],
[pixel_coords[:,:2].astype('float32')],
(img.shape[1],img.shape[0]), K, None, None,
flags=(cv2.CALIB_USE_INTRINSIC_GUESS))

K_optim, ROI =
cv2.getOptimalNewCameraMatrix(K, dist,
(img.shape[1],img.shape[0]), 1,
(img.shape[1],img.shape[0]))

img_undistort = cv2.undistort(img,
mtx, dist, None, K_optim)
```

Following are the results for correcting for radial distortion

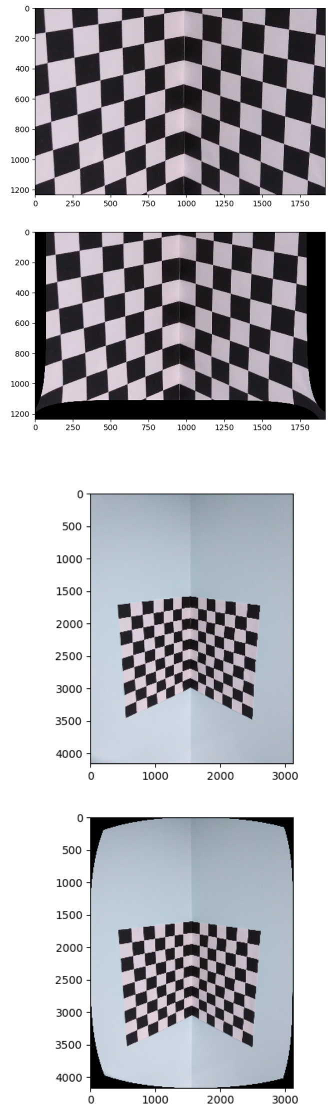
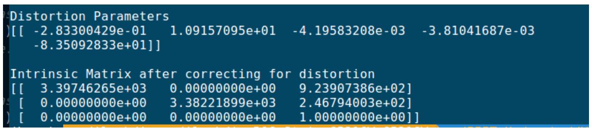


Figure 2: Corrected For Radial/Barrel Distortion

As can be seen the 1^{st} image has pin-cushion distortion and 2^{nd} image has barrel distortion.



```
Distortion Parameters
[[ -2.83300429e-01   1.09157095e+01  -4.19583208e-03  -3.81041687e-03
   -8.35092833e+01]]

Intrinsic Matrix after correcting for distortion
[[  3.39746265e+03   0.00000000e+00   9.23907386e+02]
 [  0.00000000e+00   3.38221899e+03   2.46794003e+02]
 [  0.00000000e+00   0.00000000e+00   1.00000000e+00]]
```

Figure 3: Distortion Parameters and Optimal Intrinsic Matrix

3 Zhang's Calibration

To perform Zhang's Calibration, we require at least three images of some known 2D pattern on a flat surface.
To perform a full calibration by the Zhang method at least three different images of the calibration target/gauge are required, either by moving the gauge or the camera itself.
In the first step, an approximation of the estimated projection matrix H between the calibration target and the image plane is determined using DLT method.
We need at least 4 points per plane to compute the matrix H.
Each plane gives us two equations
Using Cholesky decomposition we estimate the camera parameters.

3.1 Experimentation

The camera matrix was estimated using the Zhang's method by feeding the input checkerboard images of size 8×6. A total of 15 checkerboard images.

3.2 Observations

In all the images we can see that the wire frame fits over the image pattern, however, the world origin is almost always correctly mapped to the correct pixel point.

3.3 Results

The following are the Camera and Distortion parameters obtained using Zhang's calibration.

$$\text{K} = \begin{bmatrix} 1.36e+04 & 0.00e+00 & 3.33e+03 \\ 0.00e+00 & 1.36e+04 & 1.49e+03 \\ 0.00e+00 & 0.00e+00 & 1.00e+00 \end{bmatrix}$$

$$\text{Distortion Parameters} = \begin{bmatrix} 9.51e-02 \\ 1.01e+01 \\ -1.52e-02 \\ 2.87e-02 \\ -1.60e+02 \end{bmatrix}$$

$$\text{Rotation} = \begin{bmatrix} -0.013 & -0.051 & -0.003 \\ -0.007 & 0.226 & 0.035 \\ -0.009 & -0.416 & -0.035 \\ 0.016 & -0.618 & -0.077 \\ 0.004 & 0.307 & 0.065 \\ -0.325 & -0.222 & -0.017 \\ -0.192 & -0.385 & -0.003 \\ 0.244 & -0.322 & 0.024 \\ 0.311 & 0.066 & 0.070 \\ 0.349 & -0.563 & -0.060 \\ 0.224 & -0.405 & -0.056 \\ -0.113 & -0.298 & 0.004 \\ -0.380 & 0.007 & 0.069 \\ -0.374 & -0.486 & 0.028 \\ -0.441 & -0.425 & 0.052 \end{bmatrix}$$

$$\text{Translation} = \begin{bmatrix} -149.18 & -59.48 & 895.43 \\ -147.25 & -56.89 & 938.47 \\ -131.40 & -51.27 & 841.83 \\ -120.50 & -50.32 & 822.66 \\ -139.55 & -61.59 & 948.23 \\ -149.469 & -53.864 & 1043.80 \\ -104.13 & -58.89 & 926.99 \\ -163.859 & -35.034 & 1015.47 \\ -154.09 & -48.97 & 885.29 \\ -62.84 & -38.20 & 851.56 \\ -136.116 & -25.798 & 1123.35 \\ -121.432 & -55.995 & 1119.77 \\ -123.537 & -65.487 & 1017.74 \\ -150.201 & -59.100 & 1007.72 \\ -130.65 & -65.91 & 966.88 \end{bmatrix}$$

3.4 Overlaying Wire frames

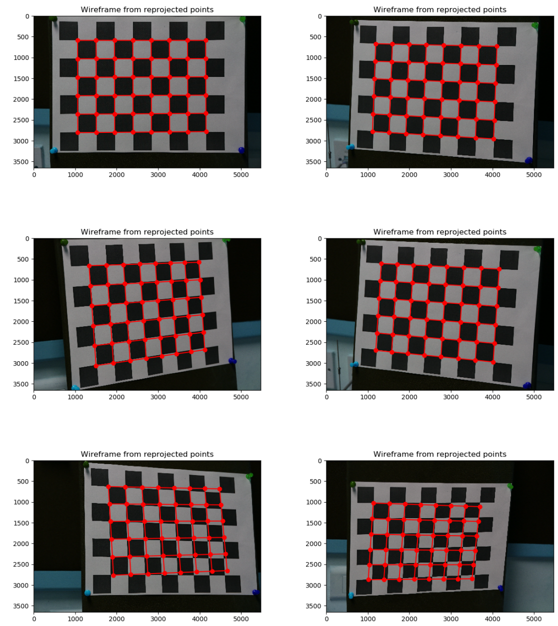


Figure 4: Caption

3.5 Image of the world origin given the calibration matrix

Given the projection/calibration matrix, the image of the world origin is given by the last column that is p_4 of the P matrix.

```
[1.06006339e+03 5.87720391e+02 1.00000000e+00]
[1060.06338922  587.72039138]
[1.19255328e+03 6.67148483e+02 1.00000000e+00]
[1192.55327986  667.14848312]
[1.20371209e+03 6.63214560e+02 1.00000000e+00]
[1203.71208577  663.21456038]
[1.33498761e+03 6.59579528e+02 1.00000000e+00]
[1334.98760867  659.5795278 ]
[1.32567303e+03 6.07838364e+02 1.00000000e+00]
[1325.67303198  607.83836382]
[1.37995095e+03 7.90574399e+02 1.00000000e+00]
[1379.95094944  790.57439859]
[1.80158652e+03 6.27431950e+02 1.00000000e+00]
[1801.58652393  627.43194993]
[1.13173999e+03 1.02455771e+03 1.00000000e+00]
[1131.73998908 1024.55771104]
[958.21626764 739.76570926   1.        ]
[958.21626764 739.76570926]
[2.32809920e+03 8.82821067e+02 1.00000000e+00]
[2328.09919502  882.82106672]
[1.68090957e+03 1.18237352e+03 1.00000000e+00]
[1680.90957121 1182.37352307]
[1.85480192e+03 8.12428347e+02 1.00000000e+00]
[1854.80191579  812.42834722]
[1.67799355e+03 6.16237386e+02 1.00000000e+00]
[1677.99355492  616.23738566]
[1.29996937e+03 6.94207641e+02 1.00000000e+00]
[1299.96937118  694.2076413 ]
[1.49020821e+03 5.63870112e+02 1.00000000e+00]
[1490.20820732  563.87011159]
```

Figure 5: Image of the World origins

As seen in Figure 5, the all the odd lines are the rightmost columns of the P matrix and the even lines are the projected coordinates of the world origin.

It can be observed that the p_4 and the projected points are nearly the same.

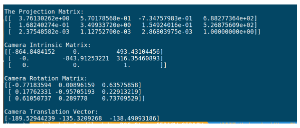
4 Hands On - Calibration of Phone Camera

4.1 Changing the focus

To change the focus of the image taken by phone, since we assume the camera to be an ideal pinhole camera, all the rays coincide with the projection plane in order to create an image in which the whole world is in focus.
This means that the camera constant c ($c = f_x = f_y$), which is the shortest distance between the principal plane and the projection plane, will change only if either of the plane shifts.
Now, for the ideal pinhole camera case since all the objects in the image are in focus, it doesn't make sense when we talk about bringing one object in focus with respect to others. Therefore in an ideal pinhole camera, the camera constant would not change and hence the intrinsic camera matrix would not change.
However, in real camera, we can bring certain objects in focus in comparison to other objects in the image. This means that the camera lens moves with respect to the projection place (CCD), in order to achieve the focus effect. Therefore, the distance between the principal plane and the projection plane changes and hence changes the camera constant, which would in turn change the calibration matrix.
In our case, since we assume ideal pinhole camera model, the pixel values we select will remain the same regardless if the object is in focus or not, which makes our pixel coordinates to be the same. Therefore, ideally our intrinsic matrix should not change (provided rotation and translation are fixed).
In our case, even when the projection plane and the principal plane change in distance, since the movement is very small (in orders of micrometers), not much change will be observed in the camera constant parameters of the intrinsic matrix.
Note: Due to manual errors while selecting points the intrinsic matrix may not turn out to be the same.

4.2 DLT Calibration

Below are the results for the camera calibration for a phone camera with different objects in focus, while maintaining the rotation and translation constant.



```
The Projection Matrix:
[[  3.76130262e+00   5.70178568e-01  -7.34757983e-01   6.88277364e+02]
 [  1.68240274e-01   3.49933720e+00   1.54924016e-01   5.26875609e+02]
 [  2.37548582e-03   1.12752700e-03   2.86803975e-03   1.00000000e+00]]

Camera Intrinsic Matrix:
[[-864.8484152     0.          493.43104456]
 [  -0.         -843.91253221  316.35460893]
 [   0.            0.            1.        ]]

Camera Rotation Matrix:
[[-0.77183594  0.00896159  0.63575858]
 [ 0.17762331 -0.95705193  0.22913219]
 [ 0.61050737  0.289778    0.73709529]]

Camera Translation Vector:
[-189.52944239 -135.3209268  -138.49093186]
```

Figure 6: Camera Parameters

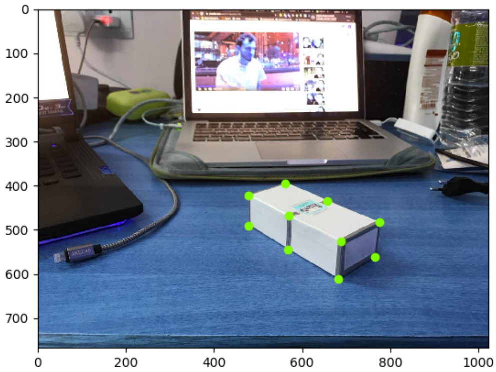


Figure 7: Box in Focus

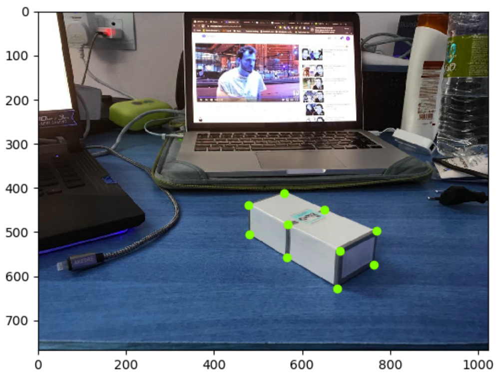


Figure 8: Keyboard in Focus

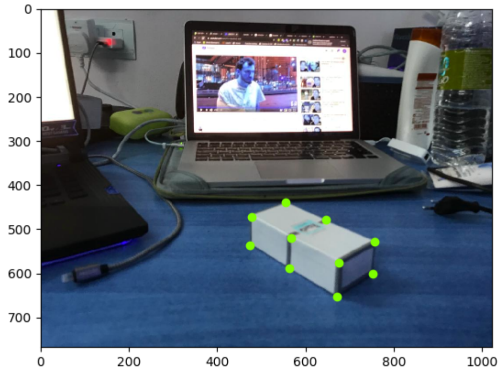


Figure 9: Bottle in Focus

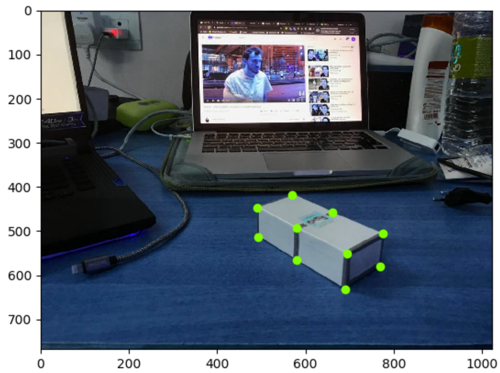


Figure 10: Plug in Focus

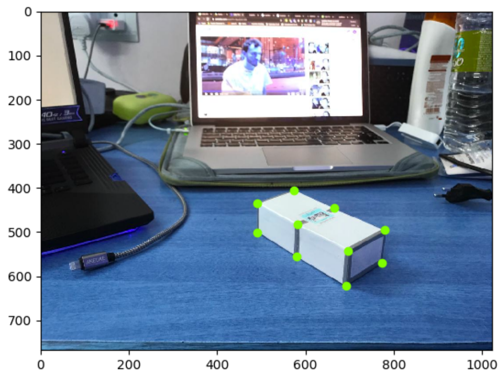


Figure 11: Box in Focus

4.3 Zhang's Calibration

Below are the results and wire frames of Zhangs calibration of a phone camera by taking pictures of a 8×6 checkerboard displayed on a laptop screen.

```
Distortion Vectors
[[  2.71588989e-01  -1.78753062e+00  -7.50954248e-05   3.03568568e-03
    3.30554822e+00]]

Rotation Vectors
[array([[ 0.09819039],
       [-0.31683617],
       [-0.03054376]]), array([[ 0.12308988],
       [-0.06587285],
       [ 0.00742345]]), array([[ 0.34817958],
       [-0.06784019],
       [ 0.0146122 ]]), array([[ 0.17896793],
       [ 0.22243543],
       [ 0.03349005]]), array([[-0.02247103],
       [-0.03143109],
       [-0.02217949]]), array([[ 0.10776335],
       [-0.08636057],
       [-0.02141936]]), array([[ 0.07221297],
       [-0.03871922],
       [-0.00248393]]), array([[-0.05398003],
       [-0.31737848],
       [-0.00439985]]), array([[-0.08799598],
       [ 0.1137621 ],
       [ 0.07142219]]), array([[ 0.236616  ],
       [ 0.04773605],
       [-0.24937777]]), array([[ 0.2825155 ],
       [-0.04448822],
       [ 0.30565493]])]

Translation Vectors
[array([[ -57.07202917],
       [ -52.60991494],
       [ 336.59580108]]), array([[ -93.52518703],
       [ -65.10525006],
       [ 360.34522442]]), array([[ -98.68274159],
       [ -49.05035146],
       [ 379.25188029]]), array([[-126.00224204],
       [ -85.52184463],
       [ 438.51189022]]), array([[ -98.79308851],
       [-118.29041543],
       [ 428.97499144]]), array([[-124.28920614],
       [ -95.60735983],
       [ 585.97090863]]), array([[-109.22461591],
       [ -69.41192748],
       [ 355.51414591]]), array([[ -69.42295205],
       [ -78.69253823],
       [ 331.95808282]]), array([[-100.31693605],
       [-104.40285006],
       [ 428.75403816]]), array([[-135.09571939],
       [ -19.27066136],
       [ 409.28491719]]), array([[ -56.87968096],
       [ -73.30154227],
       [ 362.96519133]])]
```

Figure 12: Camera Parameters

```
Intrinsix Matrix
[[ 840.07872007    0.          509.67650337]
 [   0.          842.29960902  372.62916456]
 [   0.            0.            1.        ]]
```

Figure 13: Camera Parameters

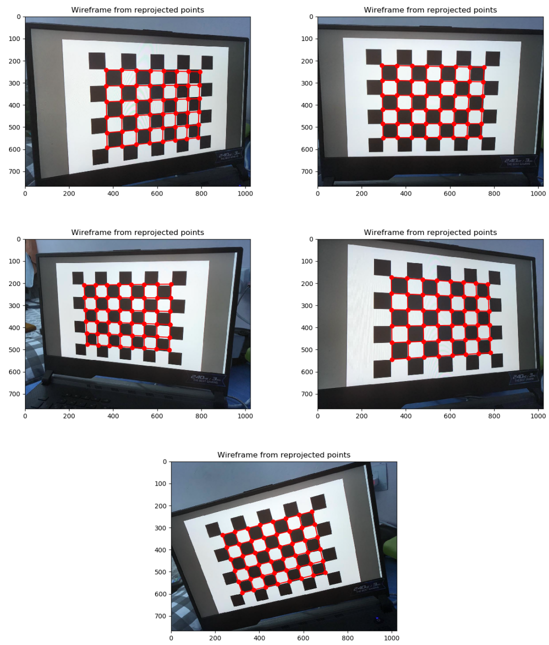


Figure 14: Zhangs Calibration using the Phone Camera