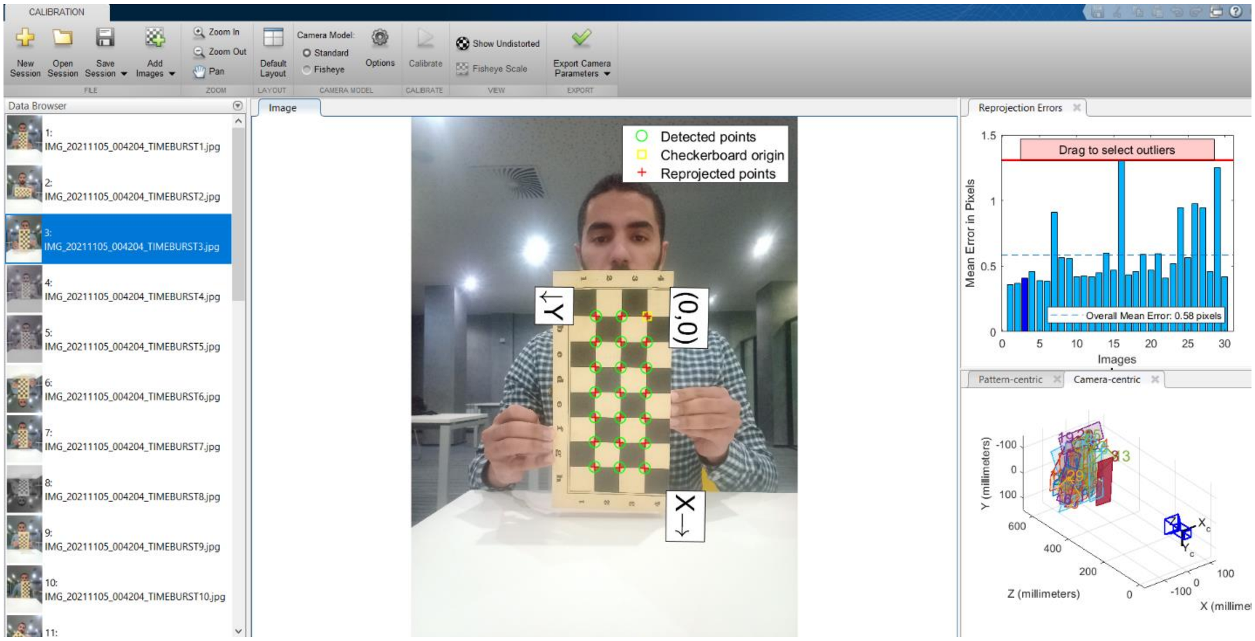
SPA

HW2 -Task 2

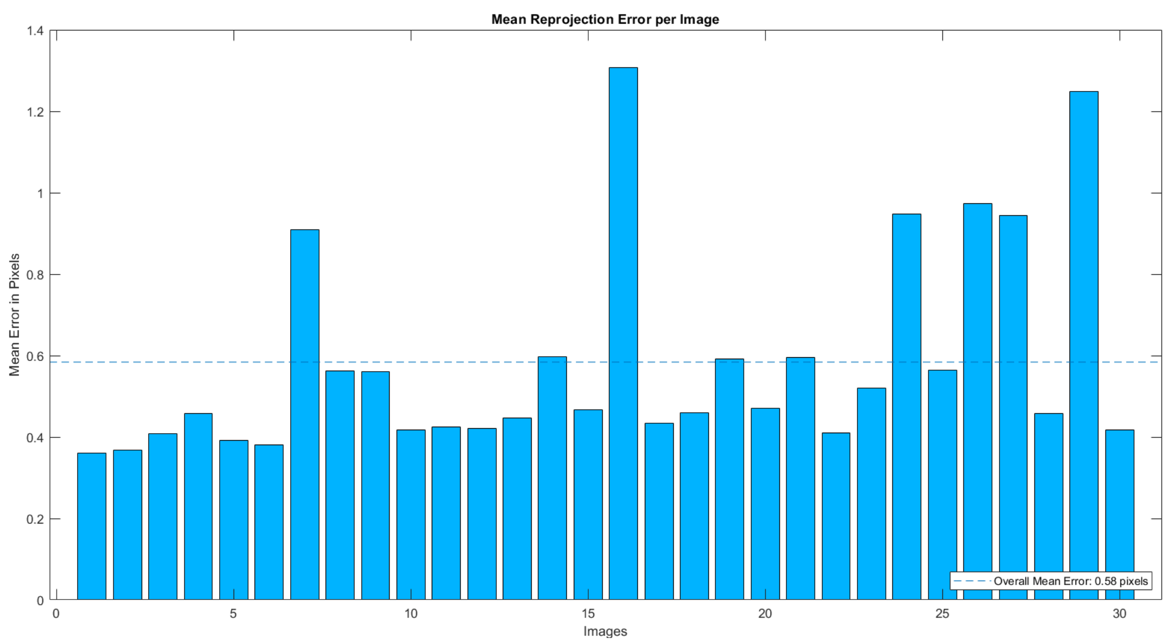
Walid Shaker

Mobile camera is calibrated using MATLAB Camera Calibrator App as shown below.

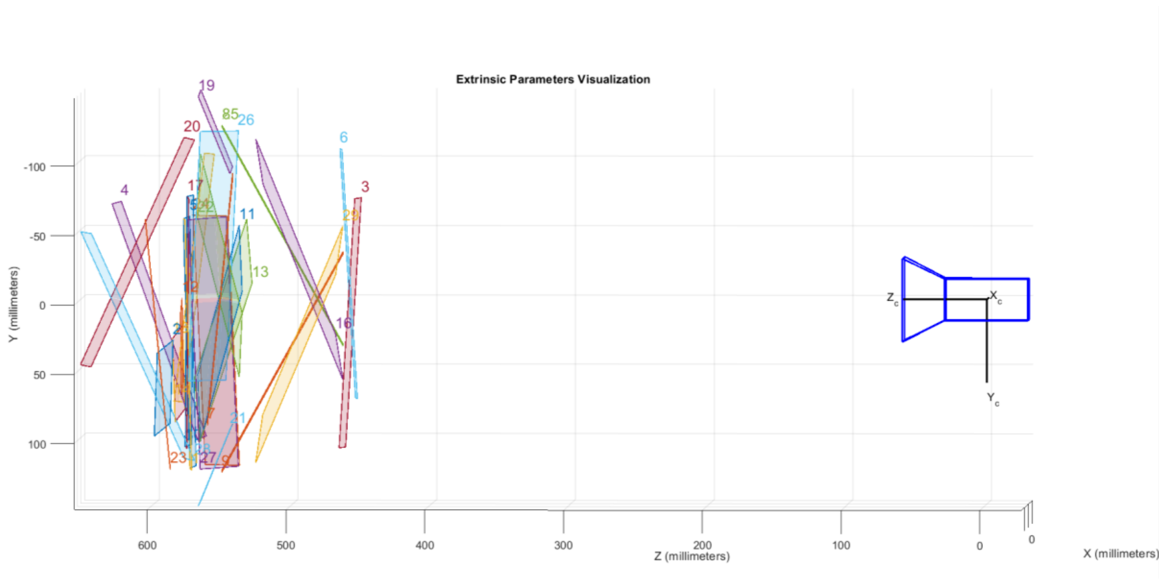


30 photos of different poses for a chessboard are uploaded to camera calibration session in order to get camera parameters, hence, to calculate the intrinsic parameters. Attached below

Attached below the mean reprojection error for each image.



In the following graph, visualization of the extrinsic parameters for each image/frame, which include the rotation matrix and translation vector for each image.



The result for the calibration is depicted below:

```
Intrinsics
----------
Focal length (pixels):   [ 3120.9998 +/- 11.7763    3117.4276 +/- 11.9710 ]
Principal point (pixels):[ 1566.7033 +/- 5.1698     2107.6689 +/- 3.5008  ]
Skew:                    [    0.1644 +/- 0.8082  ]
Radial distortion:       [    0.0812 +/- 0.0068      -0.2790 +/- 0.0425  ]
```

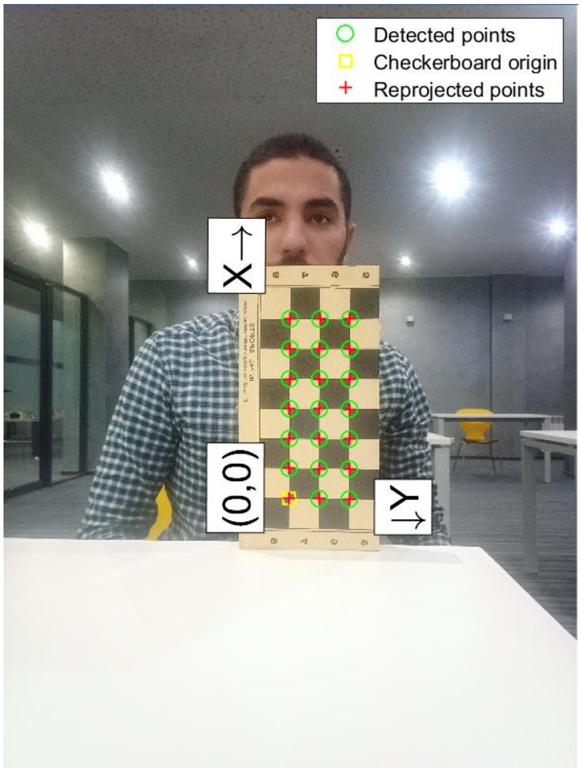
Using these parameters to calculate K matrix as follow:

$$\begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

```
K matrix for intrinsic parameters:
   1.0e+03 *

    3.1210    0.0002    1.5667
         0    3.1174    2.1077
         0         0    0.0010
```

Now, it is extrinsic parameters turn. Selecting one image from the dataset to define a world frame for the object to be measured, and to get the rotation and translation matrix for this frame.




```
>> R

R =

    0.0048   -1.0000   -0.0075
    0.9988    0.0052   -0.0490
    0.0490   -0.0073    0.9988

>> T

T =

   -2.0869
  106.2524
  577.7027
```

Combine them together, we would obtain the following:

```
RT =

    0.0048   -1.0000   -0.0075   -2.0869
    0.9988    0.0052   -0.0490  106.2524
    0.0490   -0.0073    0.9988  577.7027
```

Then, we could calculate P matrix which contain both intrinsic parameters P = KRT

```
P =

   1.0e+06 *

    0.0001   -0.0031    0.0015    0.8986
    0.0032    0.0000    0.0020    1.5488
    0.0000   -0.0000    0.0000    0.0006
```

Before moving into next step, below are the MATLAB script used to calculate all previous steps:

```
fx = cameraParams.FocalLength(1);
fy = cameraParams.FocalLength(2);
cx = cameraParams.PrincipalPoint(1);
cy = cameraParams.PrincipalPoint(2);
s = cameraParams.Skew;
K = [fx s cx; 0 fy cy; 0 0 1];
fprintf("K matrix for intrinsic parameters:\n")
disp(K)
R = cameraParams.RotationMatrices(:,:,30);
T = transpose(cameraParams.TranslationVectors(30,:));
RT = [R T];
fprintf("RT matrix of extrinsic parameters for image 30:\n")
disp(RT)
P = K*RT;
fprintf("P matrix for image 30:\n")
disp(P)
```

Recall the relationship between point in the camera and world frame that can be given as follows:

$$\lambda x = K[Rt]X = P\,X$$

$$\lambda_i\,\mathrm{x}_i = P\,\mathrm{X}_i$$

For measuring points on the same xy-plane, $\lambda_i = \mathrm{z}_c$ which is the distance from the camera center to the plane of the measured object.

The previous equation could be written as follows:

$$z_c\begin{bmatrix}u\\v\\1\end{bmatrix} = \begin{bmatrix}P1 & P2 & P3 & P4\\P6 & P6 & P7 & P8\\P9 & P10 & P11 & P12\end{bmatrix}\begin{bmatrix}X\\Y\\Z\\1\end{bmatrix}$$

Where $u = \frac{x_c}{z_c}$ and $v = \frac{y_c}{z_c}$, so we need to calculate u, v, z_c for 2 points of the object on image plane, in order to get corresponding X, Y, Z for the object in world frame and compare them to the real dimensions of our object.

z_c is the z component of the translation vector obtained using Camera Calibrator App for the image selected, this leads to $z_c = T(1,3) = 577.7027$

2 points are selected on the object (Cup) in order to get its size and compare it to the actual dimensions of the cup.

To get u, v MATLAB script is created to measure u, v for each point with respect to the image plane coordinates frame in the top left corner.

The output for this script is shown below as:

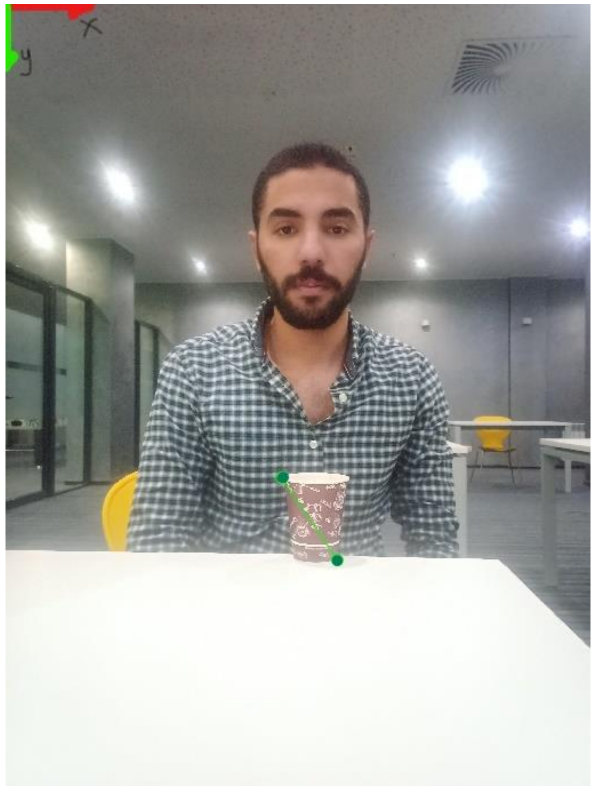
```
1.0e+03 *

 1.4435    1.8215
 2.5245    2.9325
```

The script is called readPoints.m and it will be attached.

So, now we have only 3 unknowns for each u, v there are X, Y, Z in the world frame.

It is expected that to get $Z1 = Z2 \approx 0$



That is because the cup is pictured from the same the location of the image selected; the cup is located on that image xy-plane.

Rearranging the previous equation:

$$P1 * X + P2 * Y + P3 * Z = -u * z_c - P4$$

$$P5 * X + P6 * Y + P7 * Z = -v * z_c - P8$$

$$P9 * X + P10 * Y + P11 * Z = z_c - P12$$

Now we have 3 unknowns in 3 equations, solve this system for $u1, v1$ and $u2, v2$, the output is obtained through a MATLAB script titled main.m

```
>> main
X1 , X2 : -29.072 mm , 46.780 mm
Y1 , Y2 : 20.572 mm , -49.000 mm
Z1 , Z2 : 1.576 mm , -2.652 mm
since Y axis is H and X axis is V in world frame of image 30
Cup Width : 69.572 mm
Cup Height : 75.852 mm
```

Results Comment:

	Actual	Camera Measured
Cup Width (mm)	70	69.572
Cup Hieght (mm)	75	75.852

As expected Z1 and Z2 have values near to zeros because the cup was pictured from the same distance of the selected image.

Appendix: main.m

```
leftpointC = pts(:,1);
x1 = leftpointC(1);
y1 = leftpointC(2);
rightpointC = pts(:,2);
x2 = rightpointC(1);
y2 = rightpointC(2);
S = T(3); % 577.7027
syms X Y Z
eqn1 = P(1,1)*X + P(1,2)*Y + P(1,3)*Z == x1*S-P(1,4);
eqn2 = P(2,1)*X + P(2,2)*Y + P(2,3)*Z == y1*S-P(2,4);
eqn3 = P(3,1)*X + P(3,2)*Y + P(3,3)*Z == 1*S-P(3,4);
sol = solve([eqn1, eqn2, eqn3], [X, Y, Z]);
X1 = double(sol.X);
Y1 = double(sol.Y);
Z1 = double(sol.Z);
eqn1 = P(1,1)*X + P(1,2)*Y + P(1,3)*Z == x2*S-P(1,4);
eqn2 = P(2,1)*X + P(2,2)*Y + P(2,3)*Z == y2*S-P(2,4);
eqn3 = P(3,1)*X + P(3,2)*Y + P(3,3)*Z == 1*S-P(3,4);
sol = solve([eqn1, eqn2, eqn3], [X, Y, Z]);
X2 = double(sol.X);
Y2 = double(sol.Y);
Z2 = double(sol.Z);

fprintf("X1 , X2 : %0.3f mm , %0.3f mm\n",X1,X2)
fprintf("Y1 , Y2 : %0.3f mm , %0.3f mm\n",Y1,Y2)
fprintf("Z1 , Z2 : %0.3f mm , %0.3f mm\n",Z1,Z2)
fprintf("since Y axis is H and X axis is V in world frame of image 30\n")
fprintf("Cup Width : %0.3f mm\n",abs(Y2-Y1))
fprintf("Cup Height : %0.3f mm\n",abs(X2-X1))
```