EEE-598 Camera Calibration

Hemanth Balaji Dandi

1 Part 1- Intrinsic parameter computation

Given Information : 3D to 2D Correspondences;

$$2DPoints- >' pts2D' \tag{1}$$

$$Corresponding3DPoints- > `campts3D' \tag{2}$$

To Find :

$$Intrinsic - Parameters, throughMatrixK \tag{3}$$

Procedure and Conclusion:
Given the 2D points and corresponding 3D points,

$$x' = KX \tag{4}$$

where x' = (x,y,1)', which are the 2D points, K is the Intrinsic Parameter Matrix X = (X,Y,Z,1) is the corresponding 3D Coordinate points

1.

Since we are given with X and x', hence in order to find the K matrix, we have to perform matrix inverse of the D points and multiply that with the 2D point matrix, ie,

$$K = X^{-1}x' \tag{5}$$

1.1 The K Matrix, the Intrinsic Parameter Matrix

The K Matrix is defined as :

$$K = \begin{bmatrix} \alpha x & s & px \\ 0 & \alpha y & py \\ 0 & 0 & 1 \end{bmatrix}$$

where αx is the focal length along x-direction and αy is the focal length along y-direction, hence their ratio, αx/αy represents the aspect ratio. x and y is the principal point location.

1.2 The K Computed K-Matrix

The following image represents the K-Matrix calculated by using the equations above.

```
K= [[ 3.38625072e+03  1.27159260e+01  8.52916937e+02]
 [ 1.85319235e+00  3.42886718e+03  6.27551184e+02]
 [-1.11022302e-15  2.99760217e-15  1.00000000e+00]]
```

Figure 1: Intrinsic parameter matrix (K)

2 Part 2- Intrinsic and Extrinsic Parameter Computation

In this section, both the camera intrinsics and extrinsics are estimated given the 3D points and 2-D points of the rubic cube. There are 28 3D points given to us. **??**

2.1 DLT Algorithm

Let X be the 3D points and x be the 2-D coordinates, then the 2D maps to 3D via the below equation,

$$x' \ PX' \tag{6}$$

where X' and x' are the homogeneous coordinates of the 3D and 2D points respectively. and P is the Projection Matrix which is of dimensions 3x4 . To remove the similarity, we use DLT.

First, we perform a cross product as follows,

$$x'xPX' = 0 \tag{7}$$

Solving the above equation, we get

$$\begin{bmatrix} 0^T & -X_i^T & y_i X_i^T \\ X_i^T & 0^T & -x_i X_i^T \\ -y_i X_i^T & X_i^T & 0^T \end{bmatrix} . \begin{bmatrix} P_1 \\ P_2 \\ P_3 \end{bmatrix} = 0 \tag{8}$$

where 0^T is a 1x4 dimension array of 0s; X is a 1x4 matrix containing $[X_i, Y_i, Z_i, 1]$, and it's multiplied with -1 because the third coordinate of the 2D coordinate vector $[x_i, y_i, 1]$ is set as 1.

Since in the Left Hand Side matrix , the first two equations are linearly independent, while the 3rd equation isn't, so we just use the first two equations to from our correspondences. Since we need minimum 6 points to solve for P. and since we are given with 28 points, so we form a 28-point correspondence set matrix,$'A'$, by stacking the 2x12 Matrix 28 times, to form a 56x12 matrix.

2.2 SVD of Matrix A

The equation after obtaining the A matrix is

$$A.P = 0 \tag{9}$$

In, order to solve for A, we decompose A into 3 matrices by SVD, by the following equation

$$A = U\Sigma V^T \tag{10}$$

where U, Σ, V are the left singular, Diagonal containing only the eigen values, and the right singular matrix respectively. The left and right singular matrices are orthogonal to each other and contain eigen vectors corresponding to AA^T and A^TA.

We are interested in the right singular matrix, V, the last column of which gives the P matrix, which is your calibration matrix, after reshaping into 3x4.

2.3 Minimum number of point correspondences needed for calculating P ?

The minimum number of point correspondences needed is 6.

2.4 K, R and T Parameters

From the obtained P, ie, projection matrix, it is decomposed into K, and $[Rt]$ by a variation of QR decomposition as KR is orthogonal and K is upper diagonal , as in the below equation,

$$P = K[Rt] = [KR][Kt] \tag{11}$$

where K is the calibration matrix, R is the Rotation Matrix, and [K R] has dimensions 3x3, and t is the Translation vector of dimensions 3x1.

```
P =  [[ 2.24468524e-01  2.82252106e-02 -4.46581005e-02  6.47418343e-01]
 [ 3.13049122e-03 -1.77345292e-01 -1.43567369e-01  6.89636308e-01]
 [-8.96065658e-07  2.74079257e-05 -5.24236468e-05  1.17447045e-03]]
K =  [[ 3.80781877e+03 -8.16178109e+00  8.32399977e+02]
 [ 0.00000000e+00  3.78129094e+03  7.60763925e+02]
 [ 0.00000000e+00  0.00000000e+00  1.00000000e+00]]
R =  [[ 0.99974008  0.02211926 -0.00552403]
 [ 0.01704062 -0.88594483 -0.46347748]
 [-0.01514576  0.46326288 -0.88609148]]
t =  [-1.46772505 -0.91125501 19.85150442]
```

Figure 2: P,K,R,T Matrix

2.5 Final Results

After obtaining P,K,R,t matrices, then the re-projected 2D points are calculated by

$$x_reprojected = PX \tag{12}$$

Then, in order to view both mathematically and through vision, the RMS value of error between the re-projected points and the original 2-D points is calculated, to check the intrinsic and extrensic camera parameter matrices.

As we can see, the original(green) is very close to the re-projected 2D points(blue).

```
('Error', 0.4413167951853338)
```

Figure 3: 2D re-projection error

2.6 The RubiK Cube

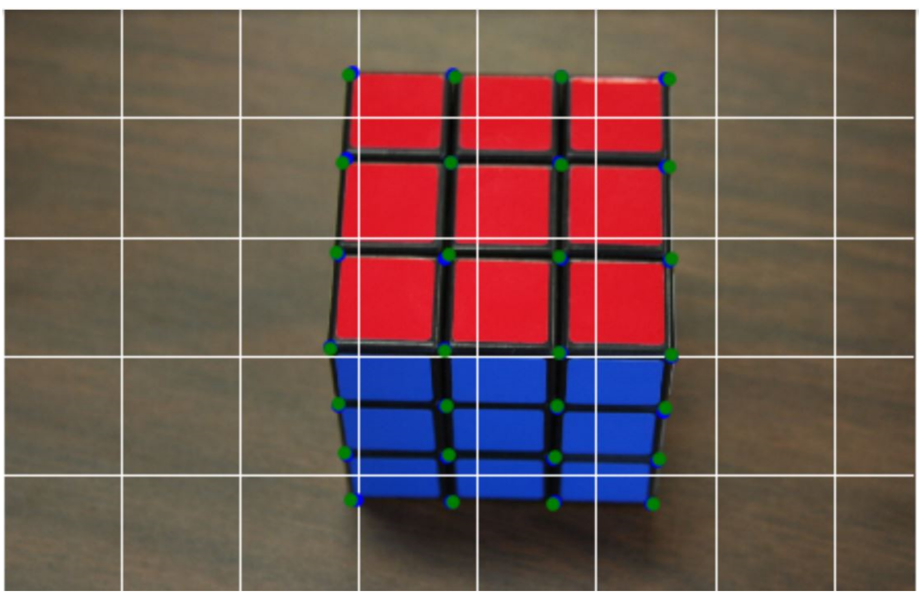

Figure 4: Rubik cube with given and re-projected 2D points on it