Exercise: Camera Model/Camera Calibration

Calibrate your smartphone camera.

1. Estimate intrinsic parameters by using Zhang's method.

2. Estimate extrinsic parameters assuming the world coordinate is defined by the chess board used for the calibration of Zhang's method.

Then, draw some **3D objects (e.g. XYZ axes, boxes, etc.)** onto each input image.

https://docs.opencv.org/4.x/dc/dbb/tutorial_py_calibration.html

Discussion

- If the spec sheet of your smartphone's camera is available, please compare the focal length of the spec sheet with the focal length obtained by the calibration.

- Note that the focal length of smartphone's camera is defined by actual length (mm, etc), but the focal length given by the calibration is defined by pixel length.

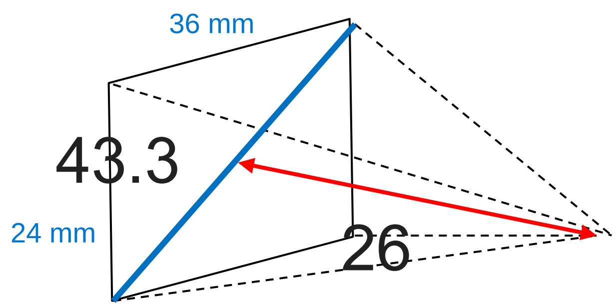
- The wide-angle lens on an iPhone Xs or Xs Max has a focal length of 4.25 mm, which is equivalent to 26 mm on a full-frame camera. (https://improvephotography.com/55460/what-is-the-focal-length-of-an-iphone-camera-and-why-should-i-care/)

-

Discussion

Full frame camera (35mm film)

- A standard 35 mm film image is 36 mm wide by 24 mm tall (35 mm refers to the height of the film including the perforations for film transport), and the diagonal is 43.3 mm



iPhone (standard camera) 26mm

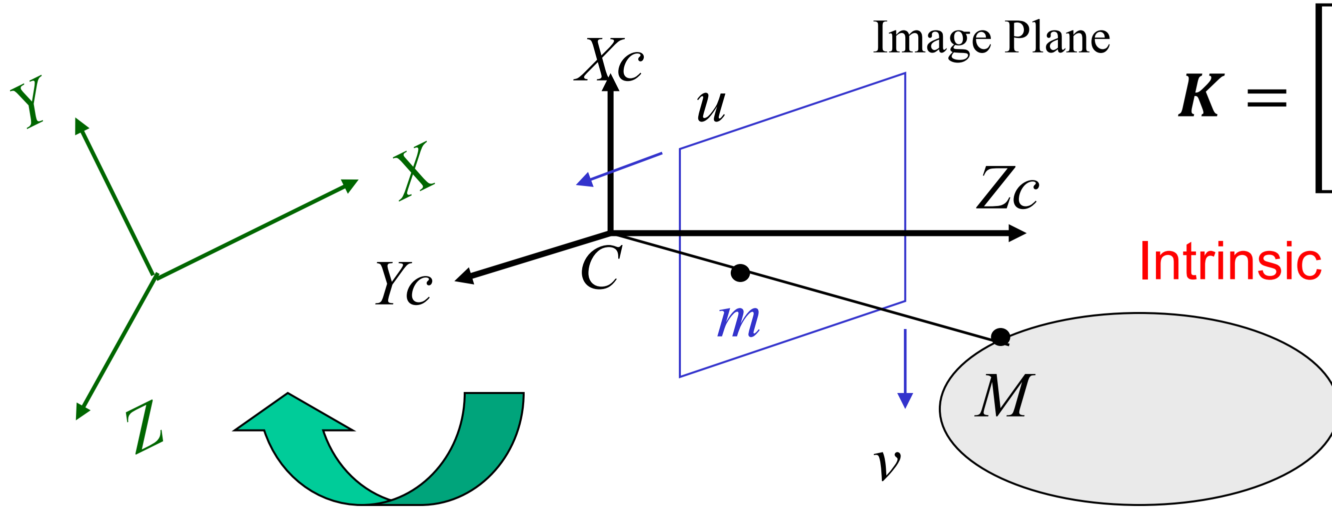
focal length

Summary of Projection Matrix

- Projection **P:** from World Coord. **M** to Image Coord. **m**

$$s\tilde{\mathbf{m}} = \boldsymbol{P}\tilde{\mathbf{M}}$$

$$s\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \boldsymbol{P}\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \boldsymbol{K}[\boldsymbol{R}|\boldsymbol{t}]\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & u_0 \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} R_{11} & R_{21} & R_{31} & t_X \\ R_{12} & R_{22} & R_{32} & t_Y \\ R_{13} & R_{23} & R_{33} & t_Z \end{bmatrix}\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$



$$\boldsymbol{K} = \begin{bmatrix} f & 0 & u_0 \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

Intrinsic Params

(R, t):Extrinsic Params(6DOFs)

Camera Calibration

Input Data

X1,Y1,Z1 → u1,v1

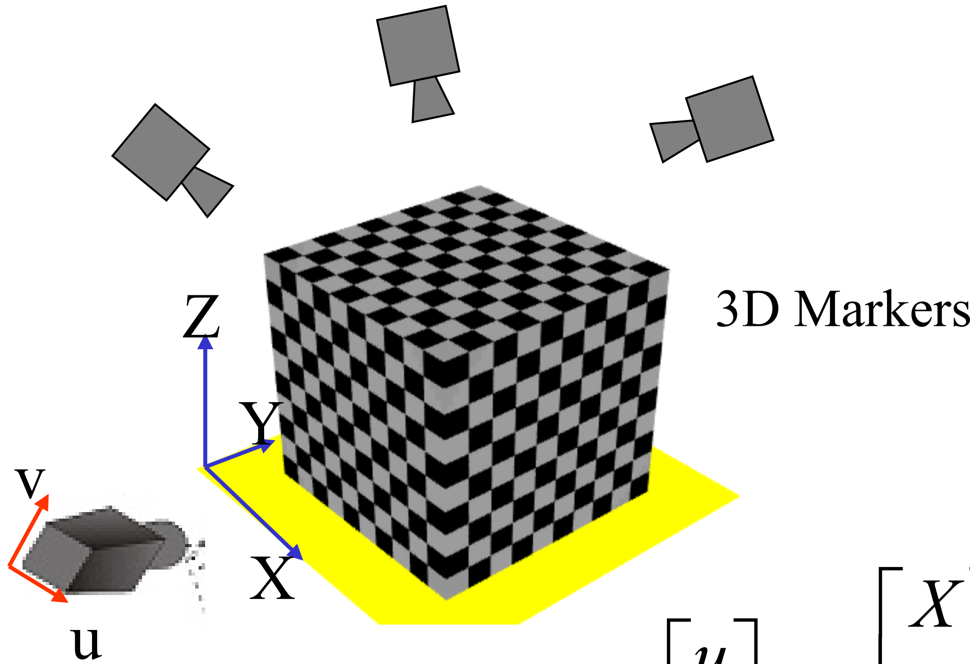
X2,Y2,Z2 → u2,v2

X3,Y3,Z3 → u3,v3

X4,Y4,Z4 → u5,v5

.

.



$$s\tilde{\mathbf{m}} = \mathbf{P}\tilde{\mathbf{M}}$$

$$s\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{P}\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Projection Matrix $\boldsymbol{P}$

Basic Method for Camera Calibration:

Distribute markers with known 3D positions (X,Y,Z) in objective space

Find image positions (u,v) onto which the markers are projected

Linear Solution for Estimating **P**

$$s\tilde{\mathbf{m}} = \mathbf{P}\tilde{\mathbf{M}} \qquad s\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} P_{11} & P_{12} & P_{13} & P_{14} \\ P_{21} & P_{22} & P_{23} & P_{24} \\ P_{31} & P_{32} & P_{33} & P_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

(X_n, Y_n, Z_n) --- (u_n, v_n)

$$P_{11}X_n + P_{12}Y_n + P_{13}Z_n + P_{14} - P_{31}X_n u_n - P_{32}Y_n u_n - P_{33}Z_n u_n - P_{34}u_n = 0$$

$$P_{21}X_n + P_{22}Y_n + P_{23}Z_n + P_{24} - P_{31}X_n v_n - P_{32}Y_n v_n - P_{33}Z_n v_n - P_{34}v_n = 0$$

$$\begin{array}{cc} & 11 \\ 2n & \begin{bmatrix} X_1 & Y_1 & Z_1 & 1 & 0 & 0 & 0 & 0 & -X_1u_1 & -Y_1u_1 & Z_1u_1 \\ 0 & 0 & 0 & 0 & X_1 & Y_1 & Z_1 & 1 & -X_1v_1 & -Y_1v_1 & Z_1v_1 \\ & & & & & & & \cdots & & & \\ X_n & Y_n & Z_n & 1 & 0 & 0 & 0 & 0 & -X_nu_n & -Y_nu_n & Z_nu_n \\ 0 & 0 & 0 & 0 & X_n & Y_n & Z_n & 1 & -X_nv_n & -Y_nv_n & Z_nv_n \end{bmatrix} \end{array} \begin{bmatrix} P_{11} \\ P_{12} \\ \vdots \\ P_{32} \\ P_{33} \end{bmatrix} = \begin{bmatrix} P_{34}u_1 \\ P_{34}v_1 \\ \vdots \\ P_{34}u_n \\ P_{34}v_n \end{bmatrix}$$

11 11

n>11/2 markers are required for estimating **P**

Estimation of Intrinsic and Extrinsic Parameters

Projection Matrix $\boldsymbol{P}$ Intrinsic Matrix $\boldsymbol{K}$ Extrinsic Matrix $[\,\mathbf{R}\,|\boldsymbol{t}]$

$$\boldsymbol{P} = \boldsymbol{K}[\,\mathbf{R}\,|\boldsymbol{t}] = \begin{bmatrix} f & 0 & c_u \\ 0 & f & c_v \\ 0 & 0 & 1 \end{bmatrix} \left[\begin{array}{ccc|c} r_{11} & r_{12} & r_{13} & t_X \\ r_{21} & r_{22} & r_{23} & t_Y \\ r_{31} & r_{32} & r_{33} & t_Z \end{array}\right]$$

$$= [\boldsymbol{K}\mathbf{R}|\boldsymbol{K}\boldsymbol{t}]$$

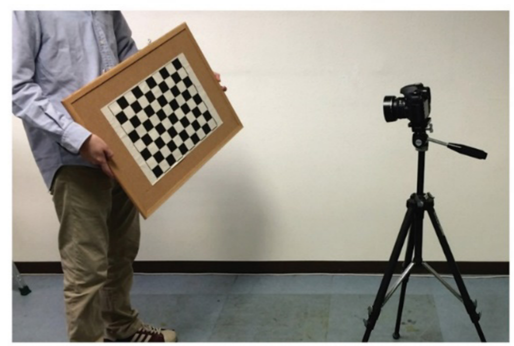
Zhang's Method：A Flexible New Technique for Camera Calibration

http://research.microsoft.com/en-us/um/people/zhang/calib/

Standard Method for Camera Calibration

http://labs.eecs.tottori-u.ac.jp/sd/Member/oyamada/OpenCV/html/py_tutorials/py_calib3d/py ion.html

■図14.7——平面状のキャリブレーションターゲットの利用



「ディジタル画像処理　改訂新版」2015年 ／ 公益財団法人 画像情報教育振興協会(CG-ARTS協会)

Camera Calibration

https://docs.opencv.org/4.x/dc/dbb/tutorial_py_calibration.html

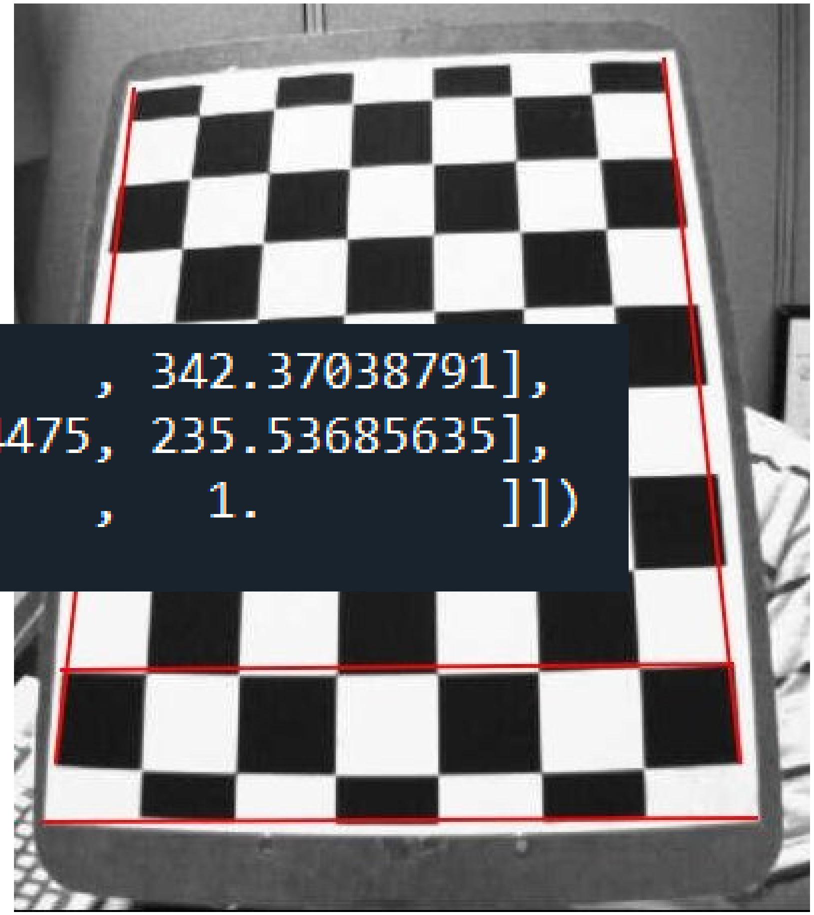
Radial distortion

- causes straight lines to appear curved.
- increases with distance

```
array([[536.07343019,   0.        , 342.37038791],
       [  0.        , 536.01634475, 235.53685635],
       [  0.        ,   0.        ,   1.        ]])
```

Di

=



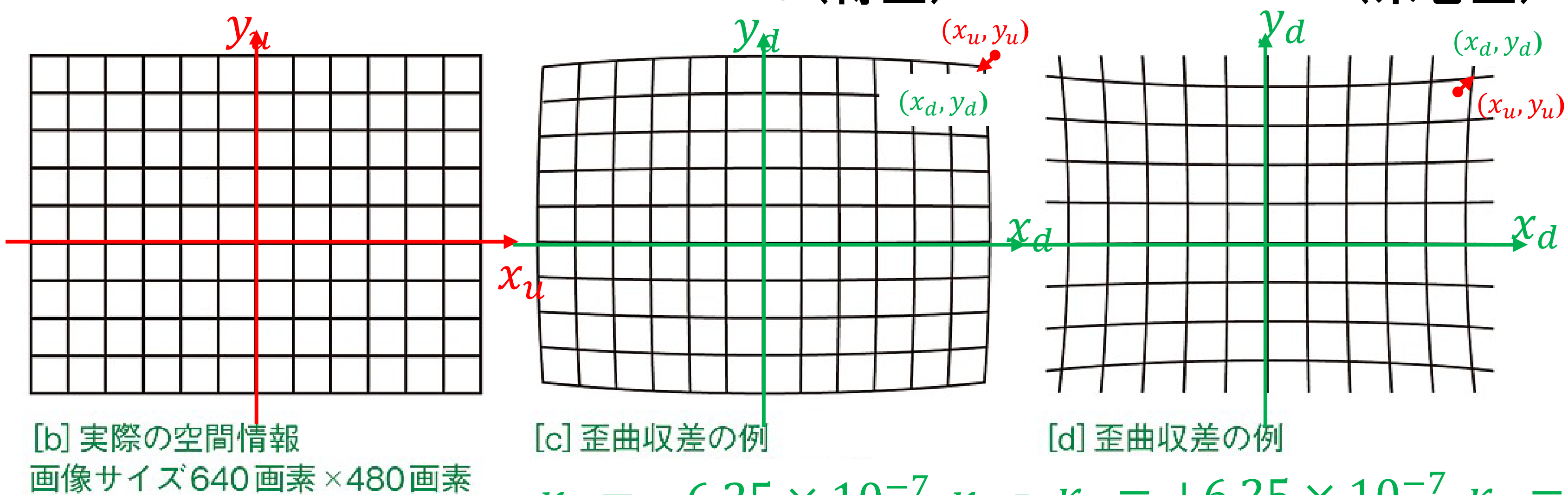
Intrinsic Matrix

$$\boldsymbol{K} = \begin{bmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

Distortion Model

Radial distortion

No distortion **Barrel(樽型)** **Pin cushion(糸巻型)**



$\kappa_1 = -6.25 \times 10^{-7}, \kappa_2 =$ $\kappa_1 = +6.25 \times 10^{-7}, \kappa_2 = 0$

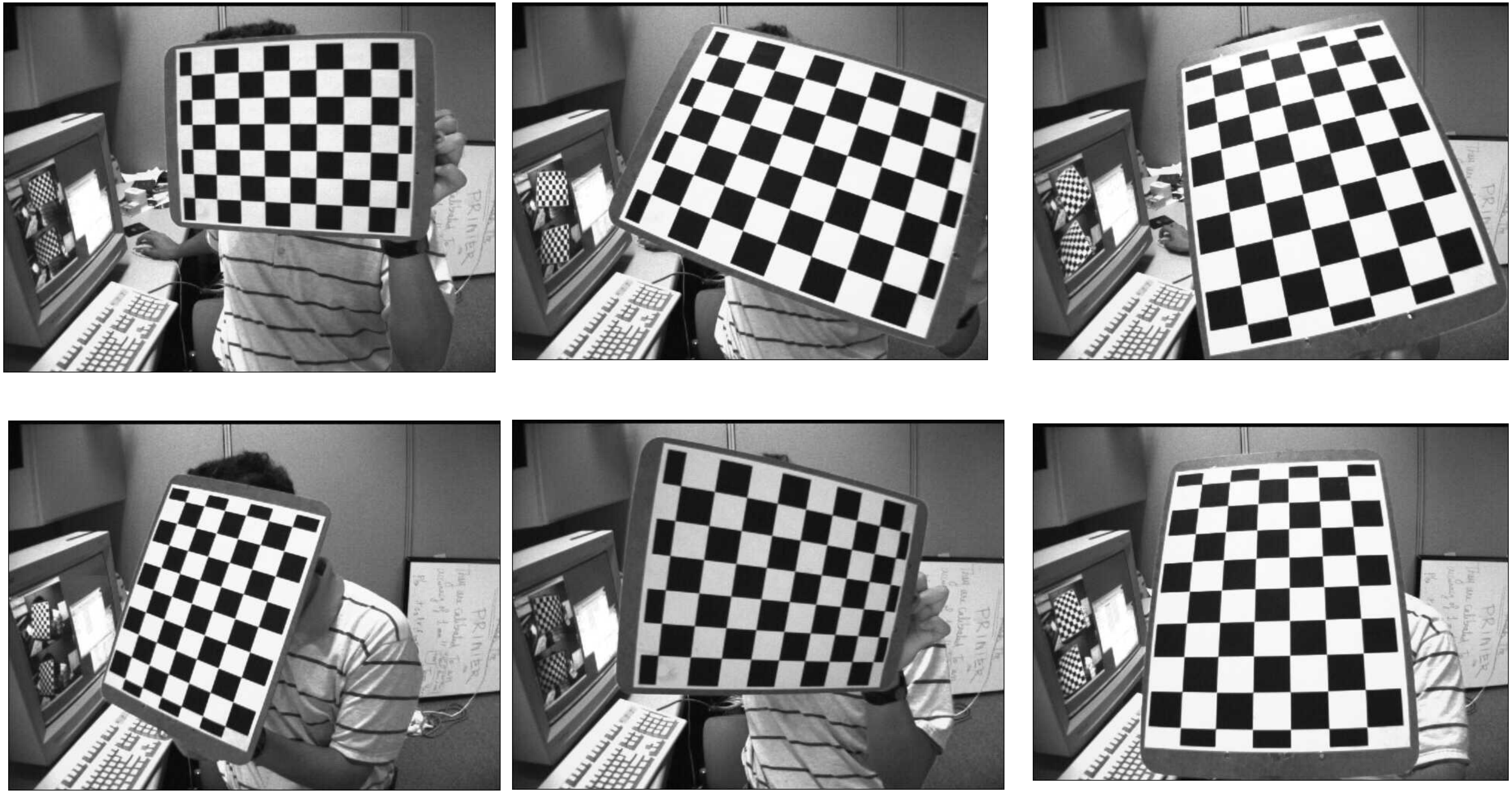
$$x_d = (1 + \kappa_1 r^2 + \kappa_2 r^4) x_u, \quad y_d = (1 + \kappa_1 r^2 + \kappa_2 r^4)\, y_u, \quad r^2 = x_u^2 + y_u^2$$

tangential distortion

$$x_d = x_u + [2p_1 xy + p_2(r^2 + 2x_u^2)], \quad y_d = y_u + [p_1(r^2 + 2y_u^2) + 2p_2 xy],$$

$$r^2 = x_u^2 + y_u^2$$

Input data: Images of chess board

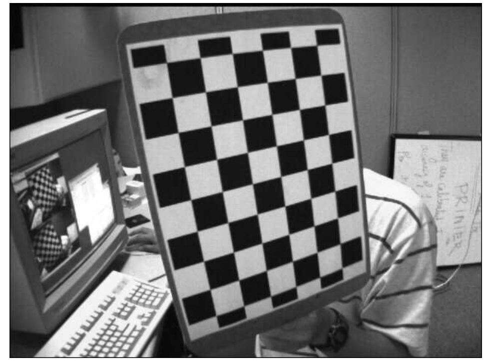


Output

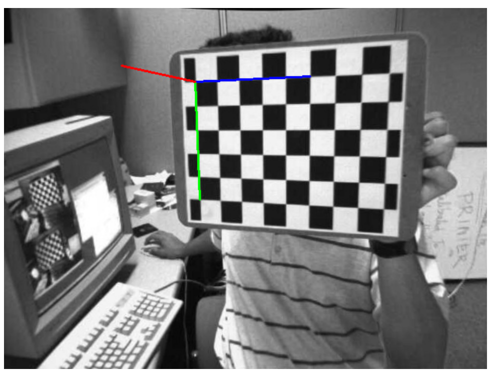
- Intrinsic Parameters

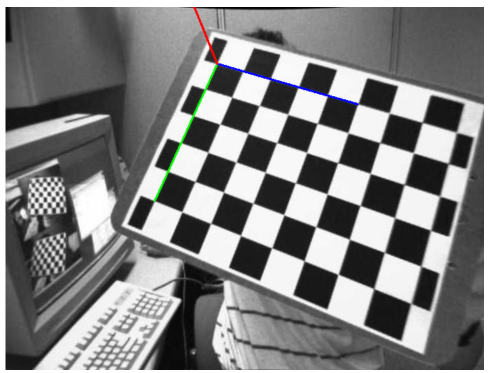
```
In [2]: mtx
Out[2]:
array([[536.07343019,   0.        , 342.37038791],
       [  0.        , 536.01634475, 235.53685635],
       [  0.        ,   0.        ,   1.        ]])
```

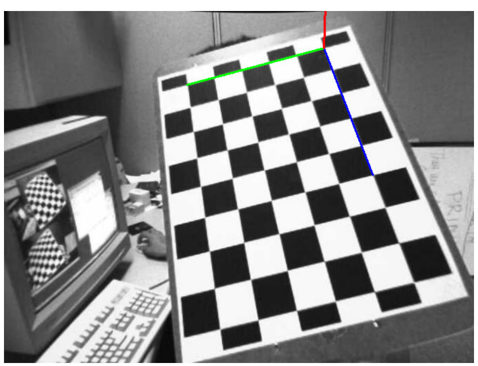
```
In [3]: dist
Out[3]: array([[-0.26509006, -0.0467439 ,  0.00183301, -0.00031471,  0.25231586]])
```

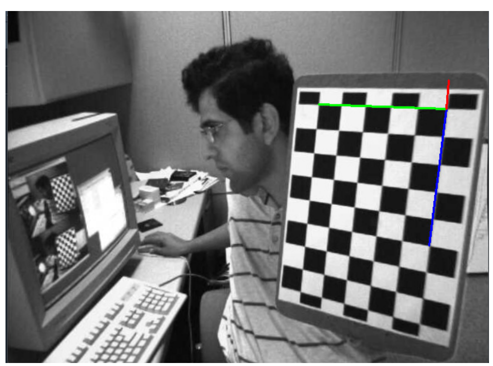


- Extrinsic Parameters for Each Chess Pattern









```
In [5]: rvecs
Out[5]:
(array([[0.16853568],
        [0.27575314],
        [0.01346807]]),
 array([[ 0.41306755],
        [ 0.64934521],
        [-1.3371948 ]]),
 array([[-0.27697519],
        [ 0.186891  ],
        [ 0.35483188]]),
 array([[-0.11082302],
        [ 0.23974769],
        [-0.0021351 ]]),
 array([[-0.29188239],
        [ 0.42829931],
        [ 1.31269863]]),
 array([[0.40772934],
        [0.30384783],
        [1.64906546]]),
 array([[0.1794729 ],
        [0.34574756],
        [1.86847037]]),
 array([[-0.09096653],
        [ 0.47965894],
        [ 1.75338411]]),
```

```
In [6]: tvecs
Out[6]:
(array([[-3.01118544],
        [-4.35756677],
        [15.99287288]]),
 array([[-2.3455136 ],
        [ 3.31931519],
        [14.15396033]]),
 array([[-1.59581799],
        [-4.0160139 ],
        [12.72969818]]),
 array([[-3.93839211],
        [-2.69241843],
        [13.23774819]]),
 array([[ 2.33766726],
        [-4.61207374],
        [12.69075982]]),
 array([[ 6.68813485],
        [-2.62204557],
        [13.46297328]]),
 array([[ 0.77879779],
        [-2.87200426],
        [15.58024528]]),
```

Report submission

- Deadline: May 1
- Language: English or Japanese
- Content: Goal, Method, Results, Discussion, Conclusion