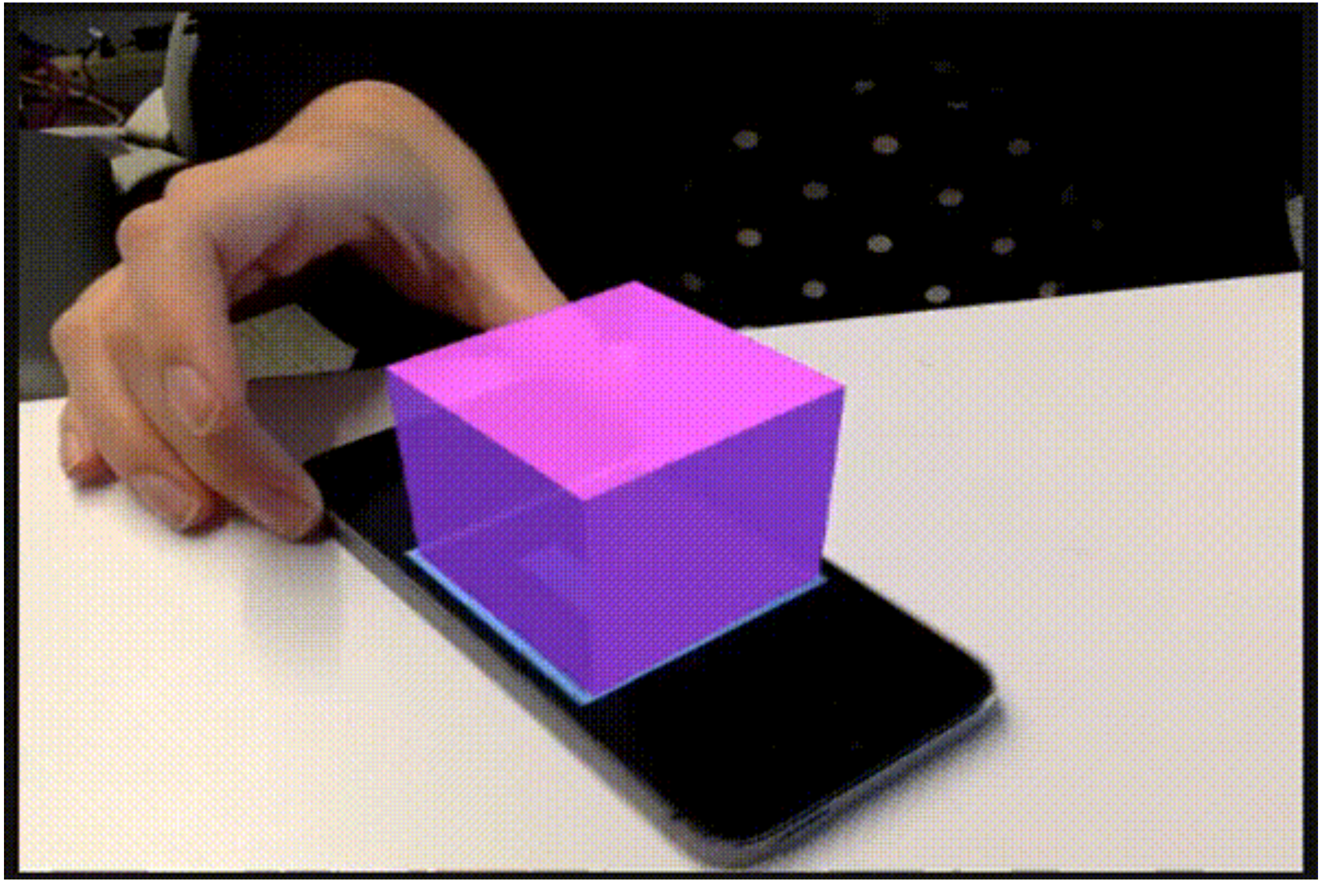
CE4003/CZ4003 Computer Vision

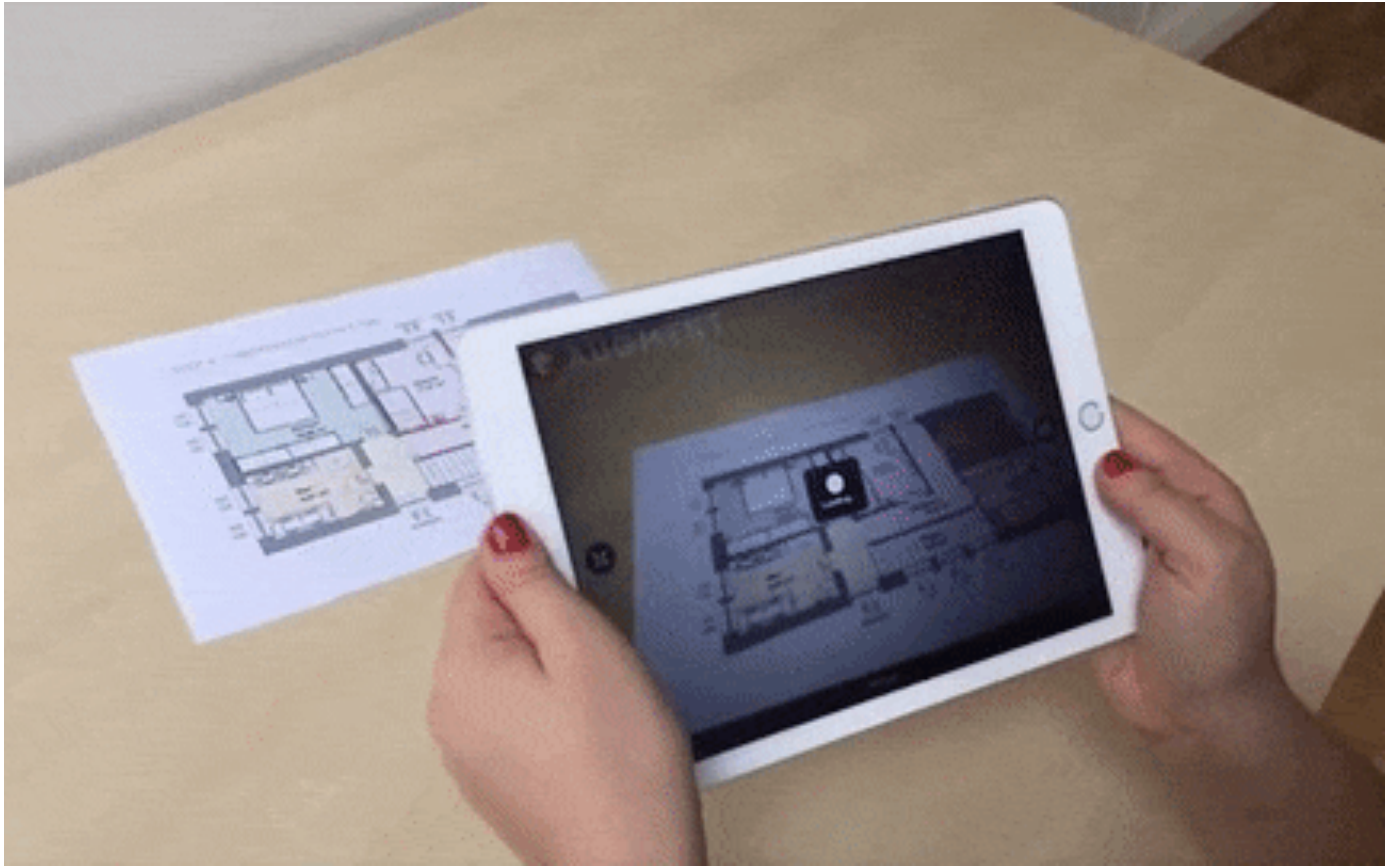
6. Imaging Geometry

camera → capture 2D → add model → display in 3D.

Overview – Imaging Geometry

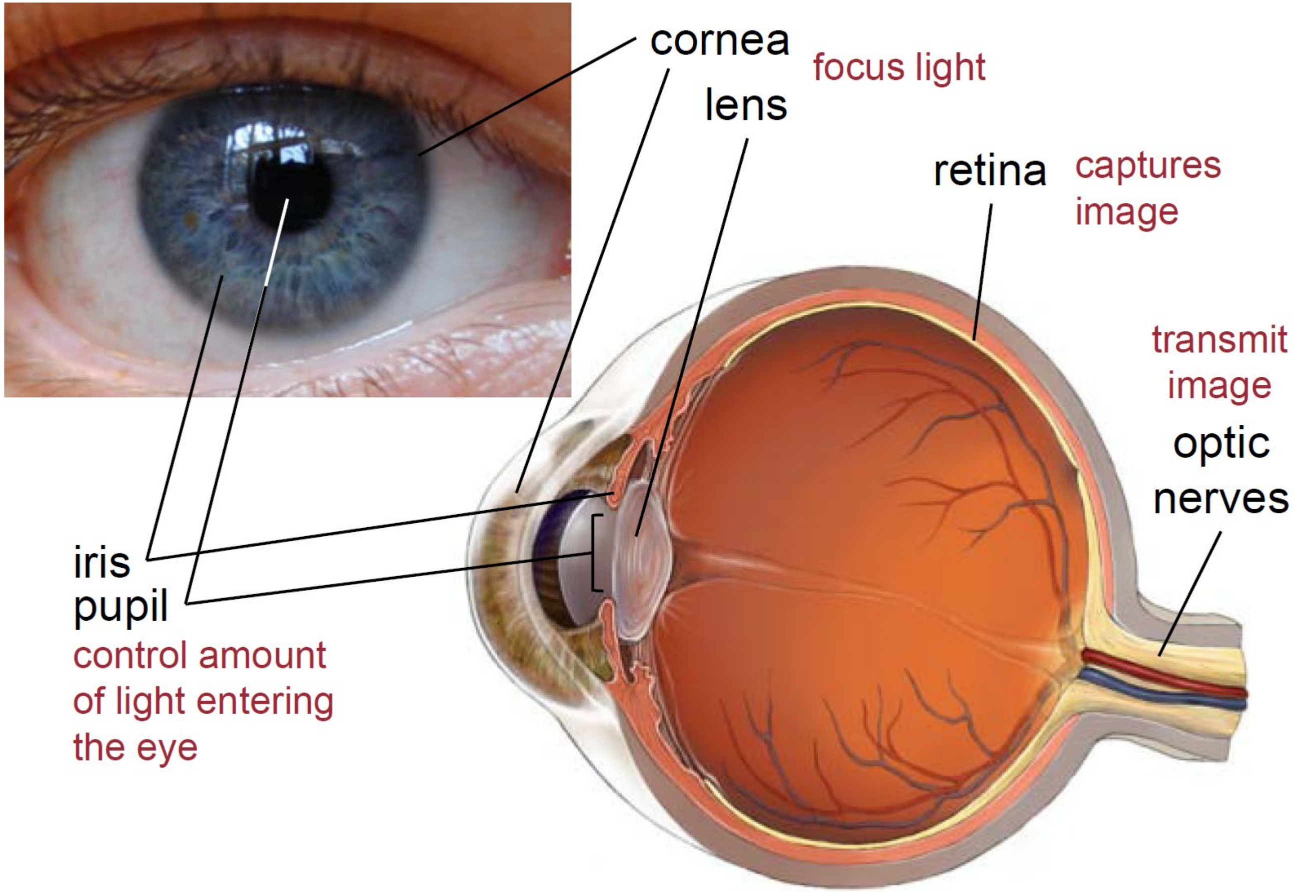


Overview – Imaging Geometry

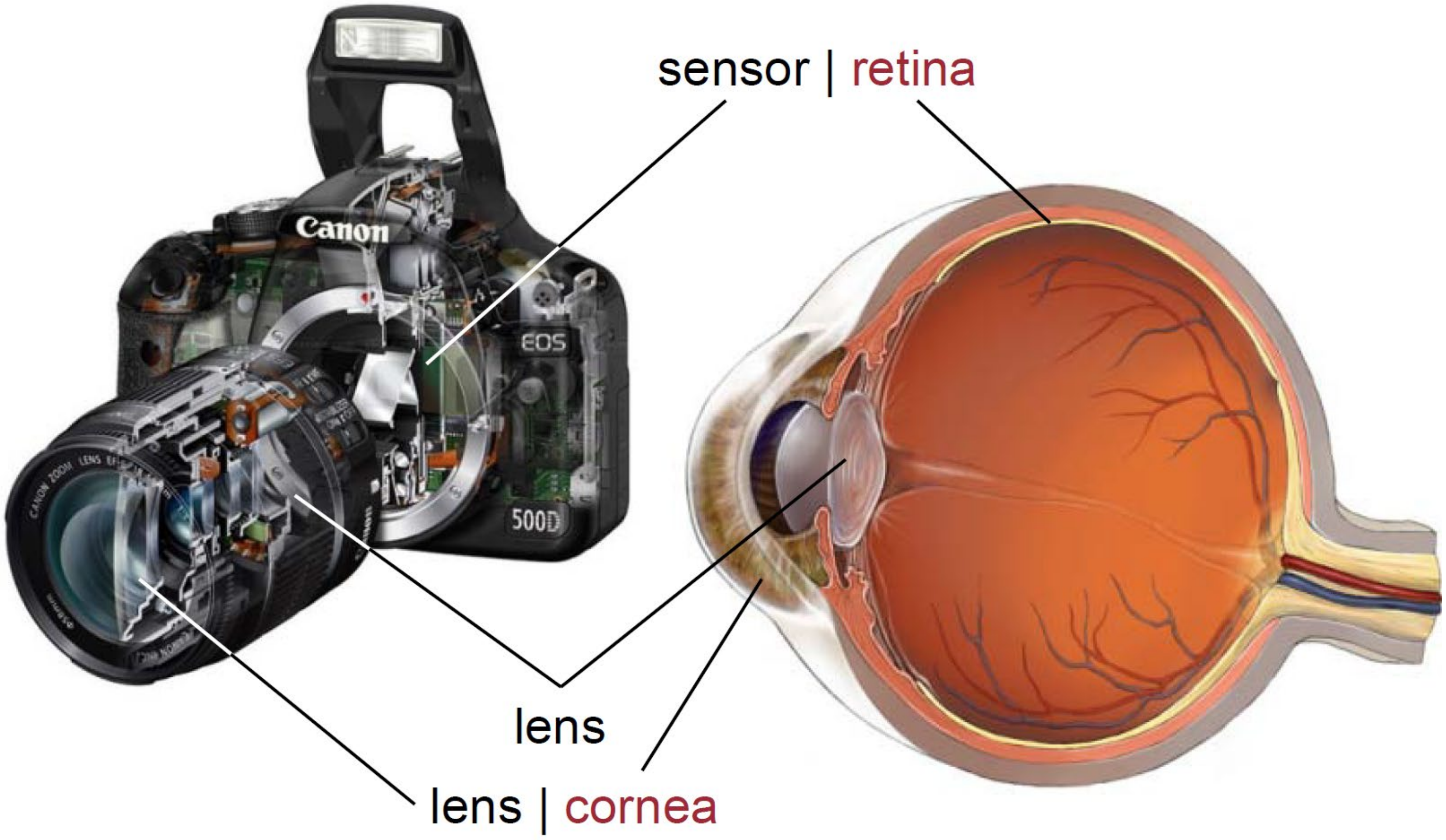


Contents

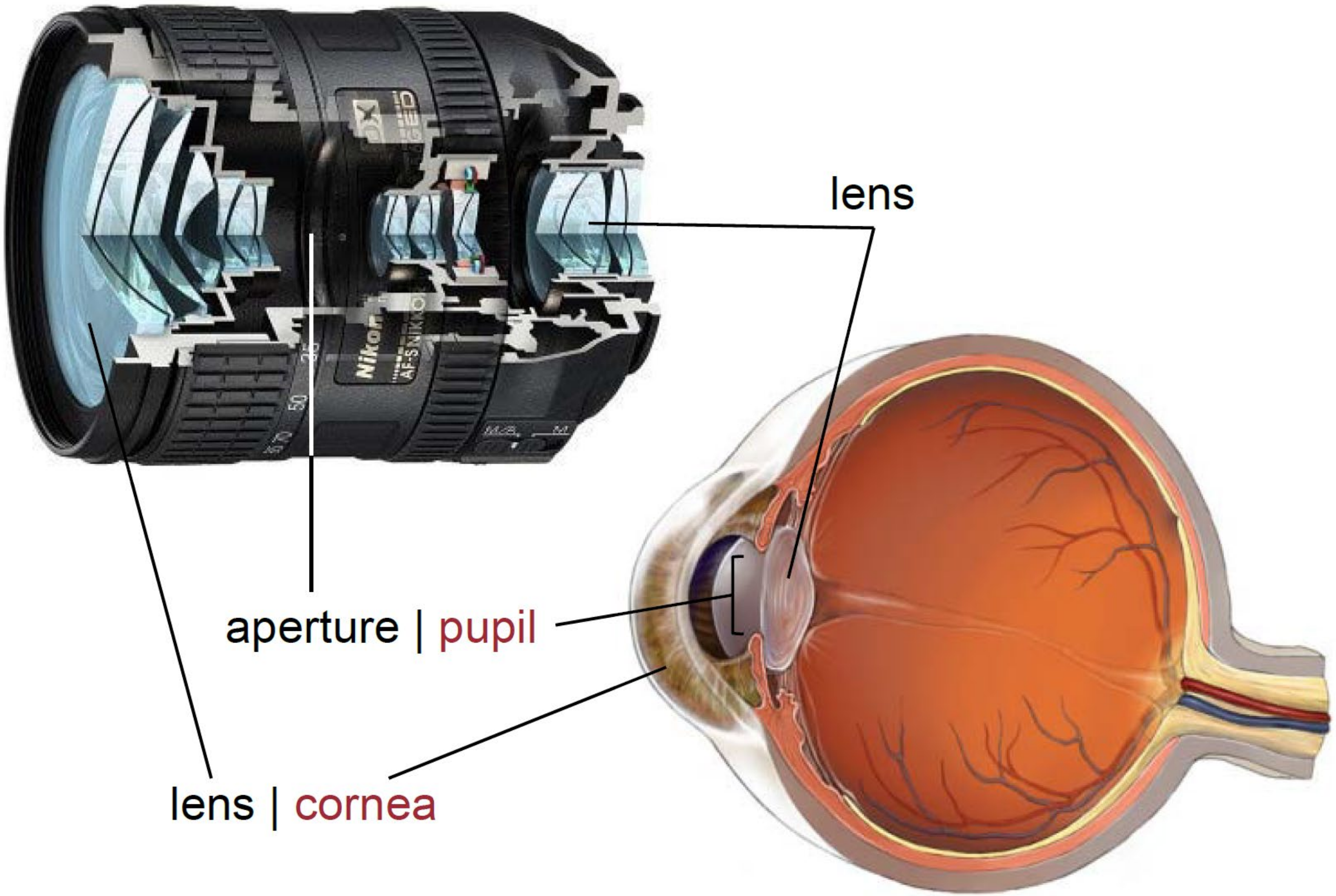
1. Revisiting Imaging Systems



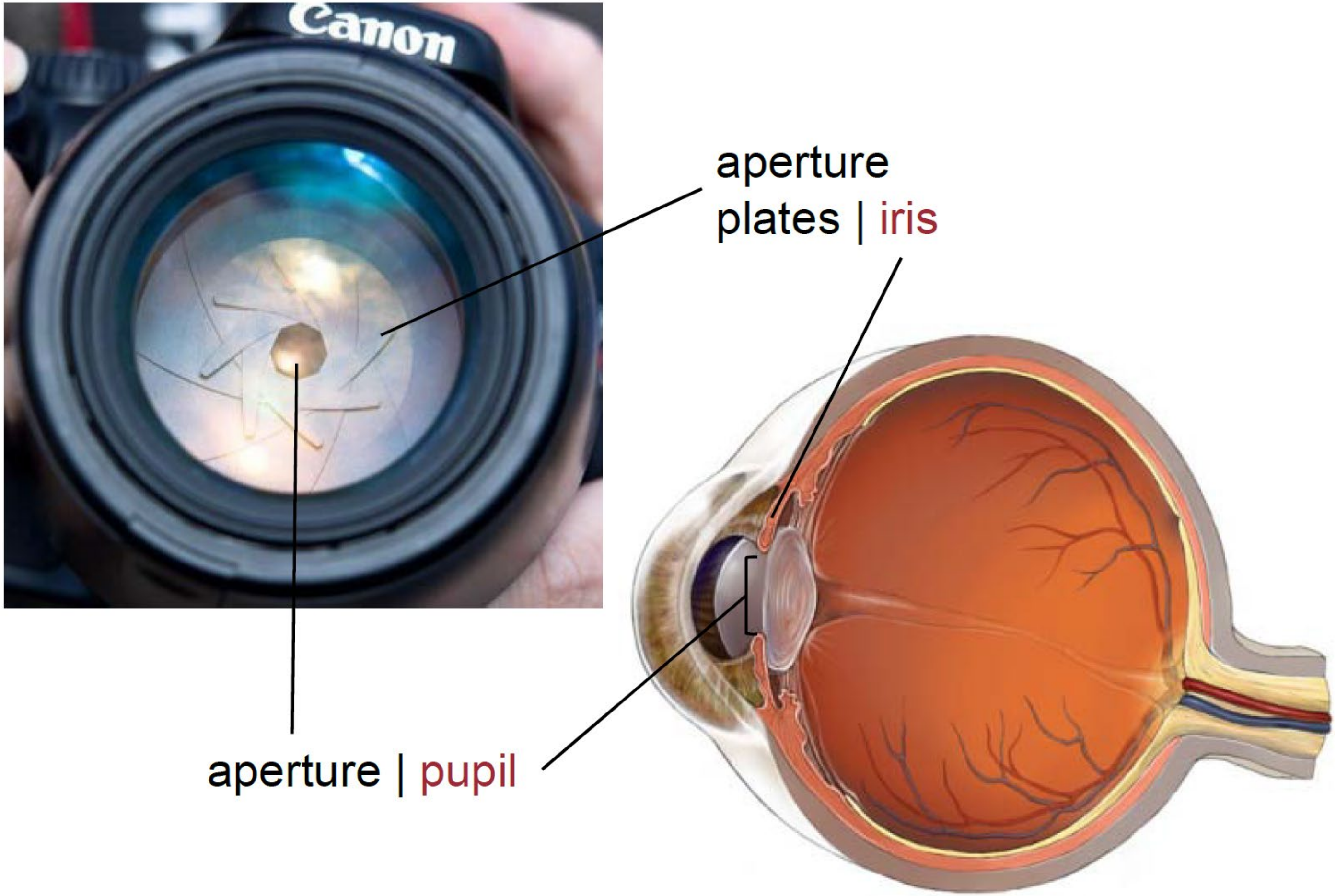
1. Revisiting Imaging Systems



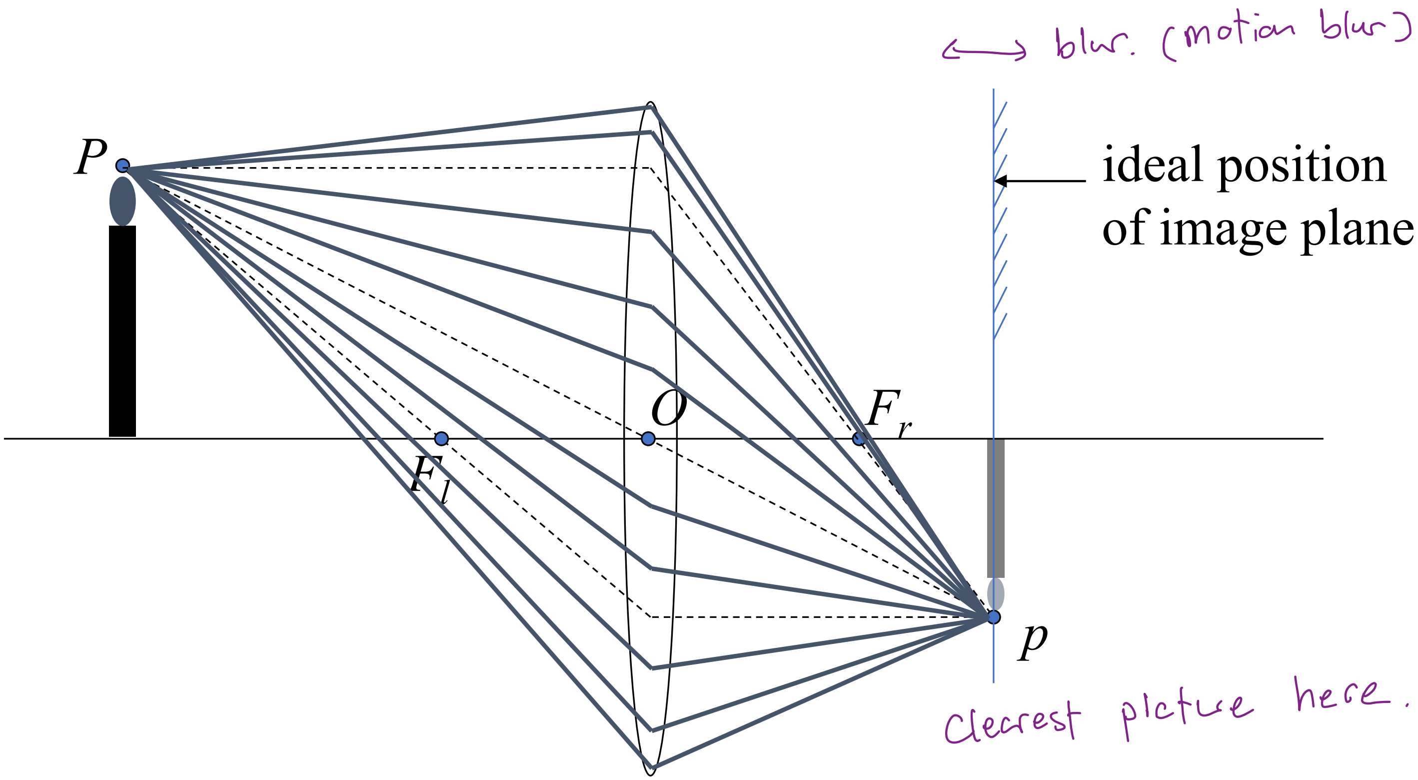
1. Revisiting Imaging Systems



1. Revisiting Imaging Systems



1. Revisiting Imaging Systems



1. Revisiting Imaging Systems

The 3D-to-2D imaging is sufficient and useful to many real-world applications.

to recognize plate number.

https://clouard.users.greyc.fr/Pantheon/experiments/licenseplate-detection/index-en.html

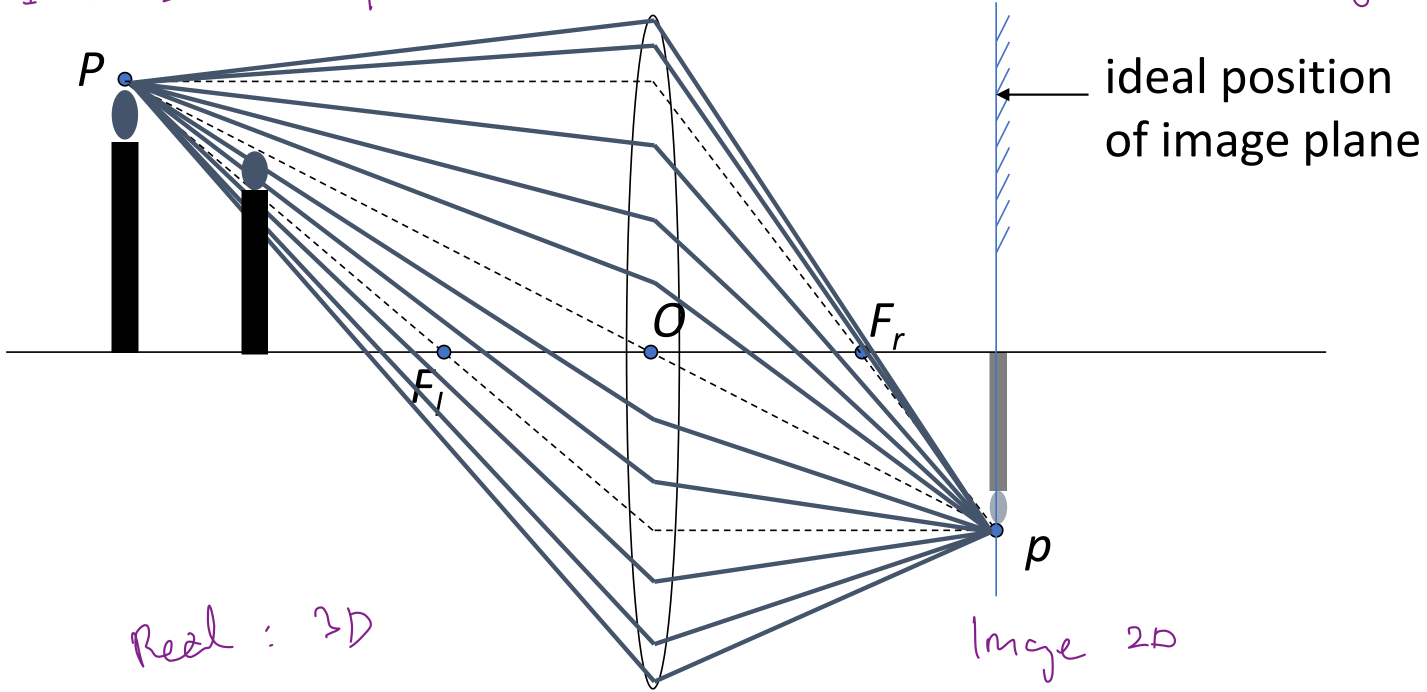
1. Revisiting Imaging Systems

But it might not be sufficient for many applications that requires to get 3D information from 2D images.

- Can we recover the 3D world from 2D images?
- How to build up the relation between scene points and their image points?
- Different systems have different relations. How?

many-to-1 projection

not 1-to-1 correspondence



Real : 3D

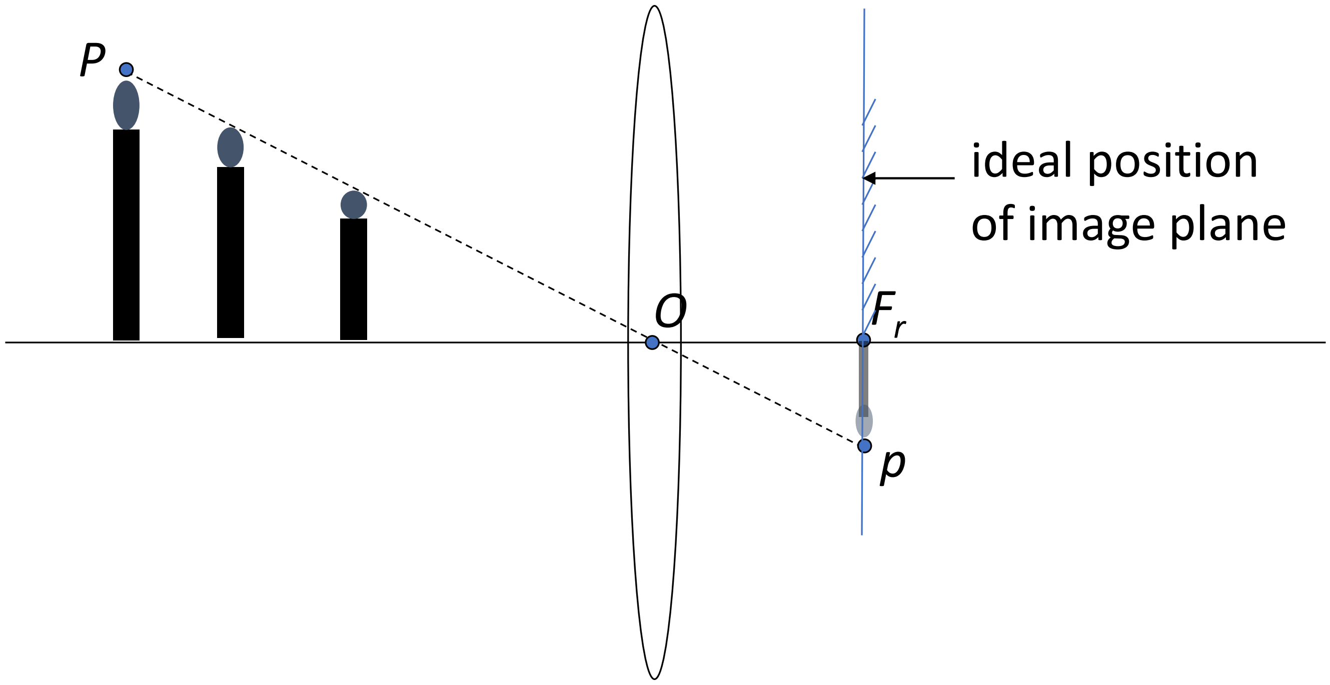
Image 2D

2.1 Imaging Geometry

Big question: how to construct 3D from 2D.

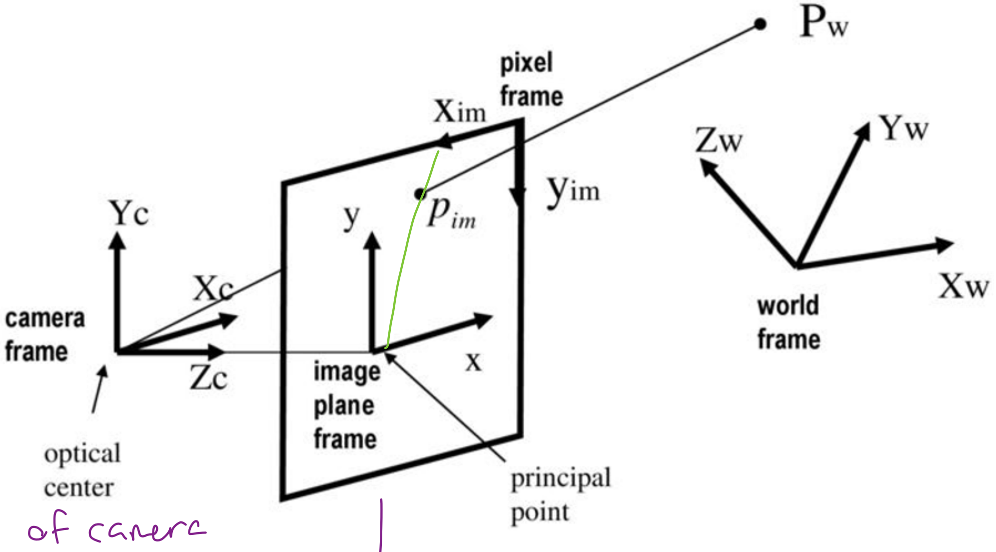
2.1 Imaging Geometry: the Pinhole Model

We start to build up the imaging geometry with a simplified pinhole camera.



2.1 Imaging Geometry: the Pinhole Model

There are **four** coordinate systems in the pinhole model.



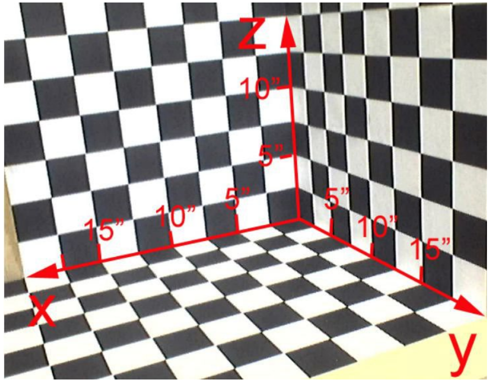


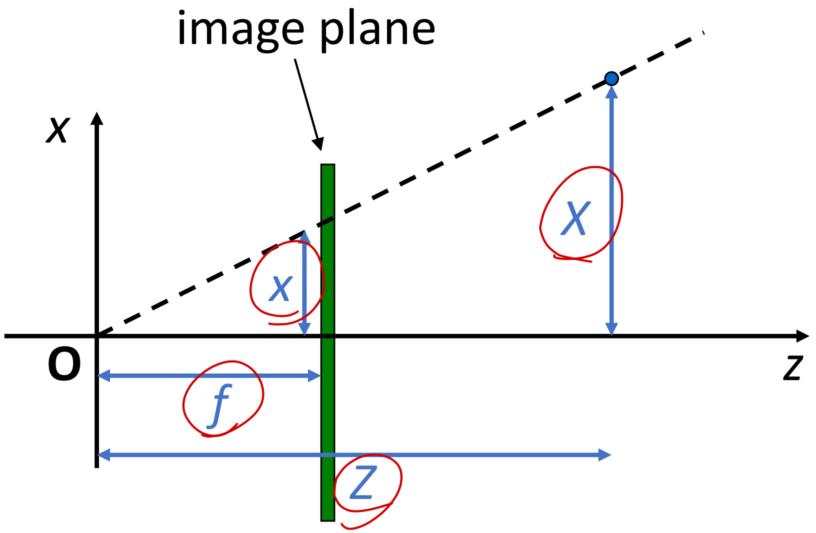
image plane.

Camera model:

$$p_{im} = \overset{3\times 4}{\begin{bmatrix} transformation \\ matrix \end{bmatrix}} P_w$$

World Coords		Camera Coords		Image Coords		Pixel Coords
X_W, Y_W, Z_W	→	X_C, Y_C, Z_C	→	x, y	→	x_{im}, y_{im}

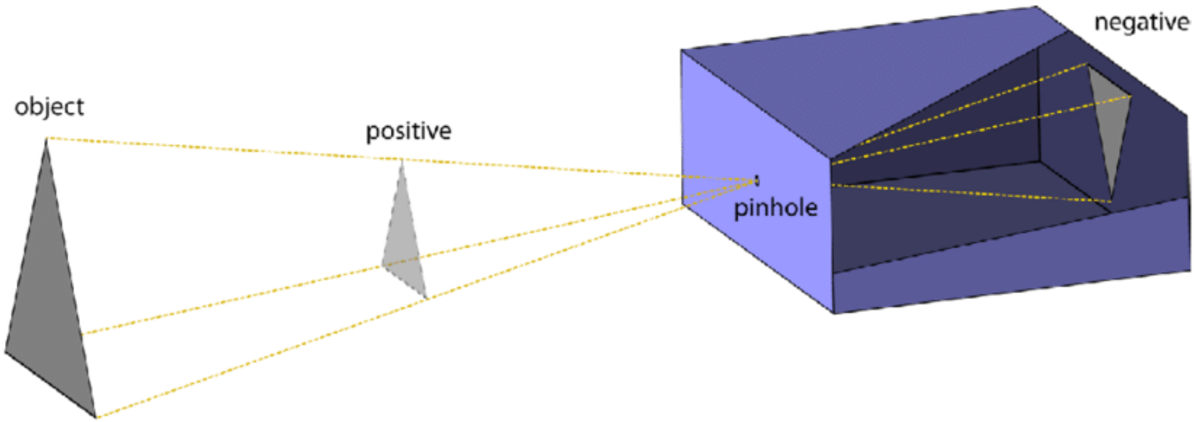
2.1 Between the Camera and Image Frames

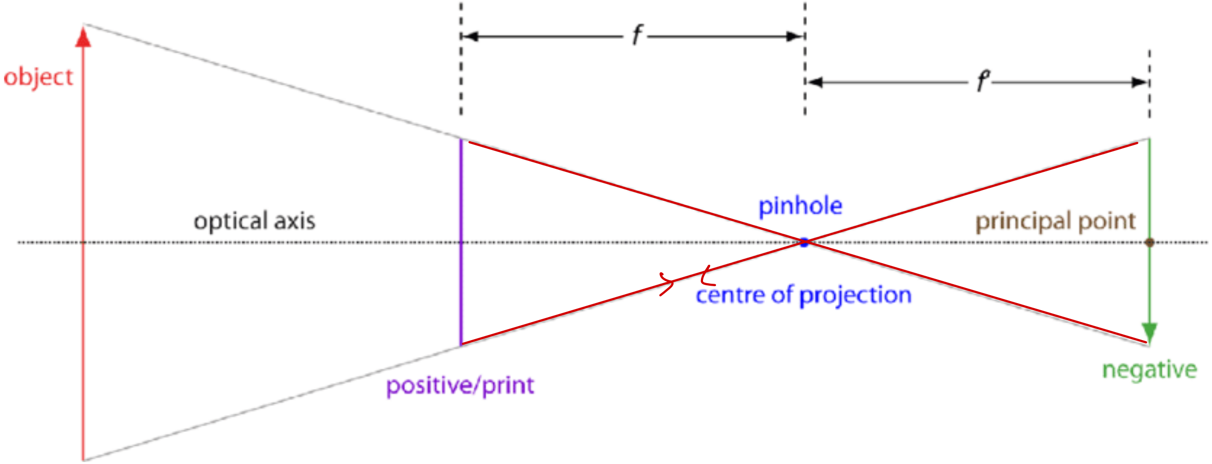


Based on similar triangles, in the camera and image frames:

$$x = f\frac{X}{Z} \qquad y = f\frac{Y}{Z}$$

2.1 Between the Camera and Image Frames





$$kx = fX$$

$$\frac{x}{f} = \frac{X}{k}$$

Similar Δ

2.1 Between the Camera and Image Frames

~ scaling factor.

many 3D point possible.

$$\begin{bmatrix} kx \\ ky \\ k \end{bmatrix} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

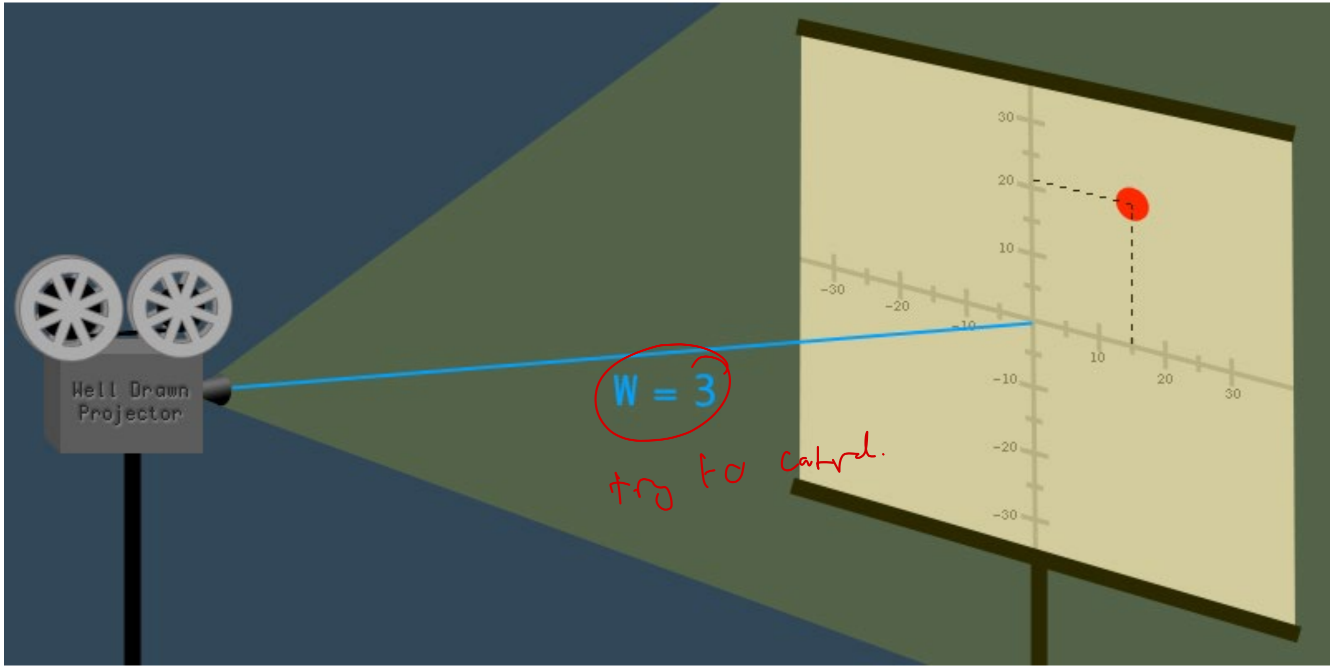
2D point in the camera frame

Perspective projection

3D point in the camera frame

2.1 Between the Camera and Image Frames

Homogenous coordinate is widely used in 3D vision. It includes one more dimension beyond the Euclidean coordinate, acting as a scaling factor. A point (5, 7) in Euclidean coordinate can thus be represented by (5, 7, 1) with w=1 or (15, 21, 3) with w=3 in the homogenous coordinate.

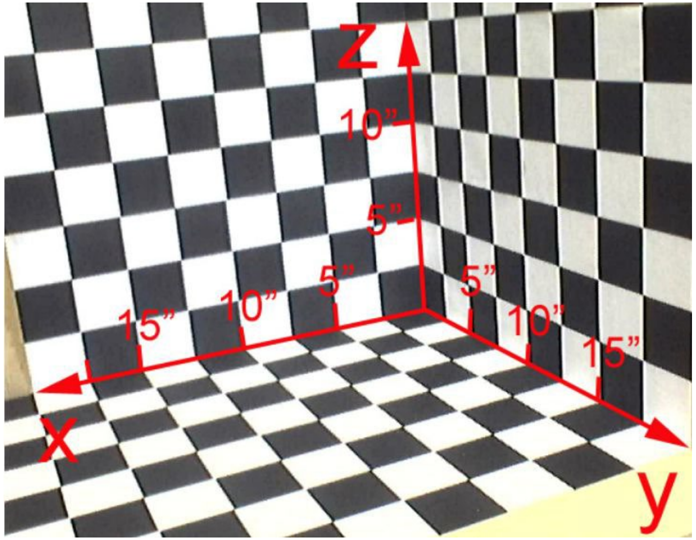


2.1 In the Camera Frame

→ 3D but intangible; no convenience to work with.

- The camera frame is intangible
- Object can be measured in the World Frame
- Image can be measured in the Image frame

World frame



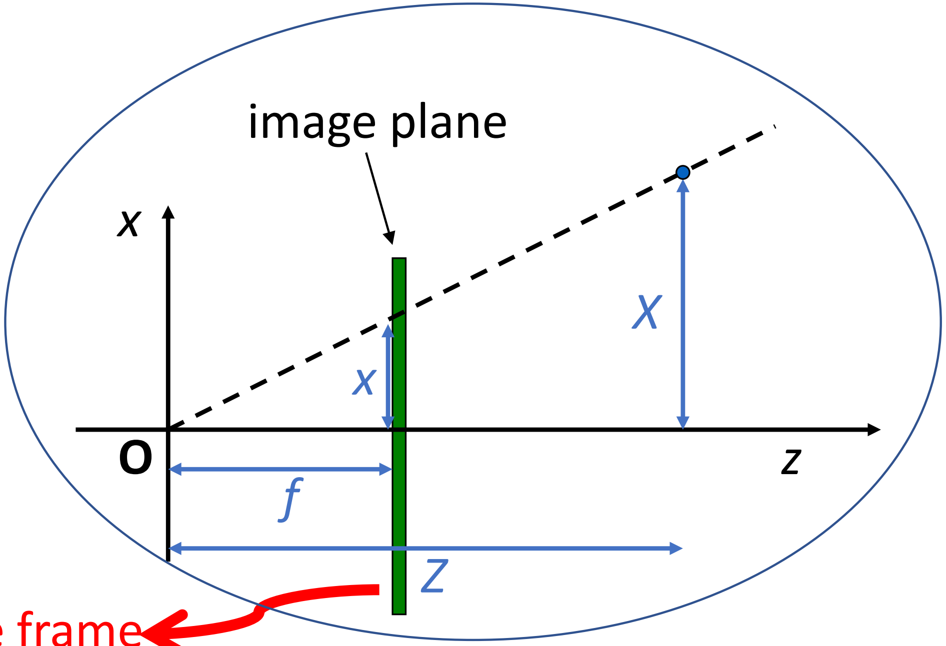


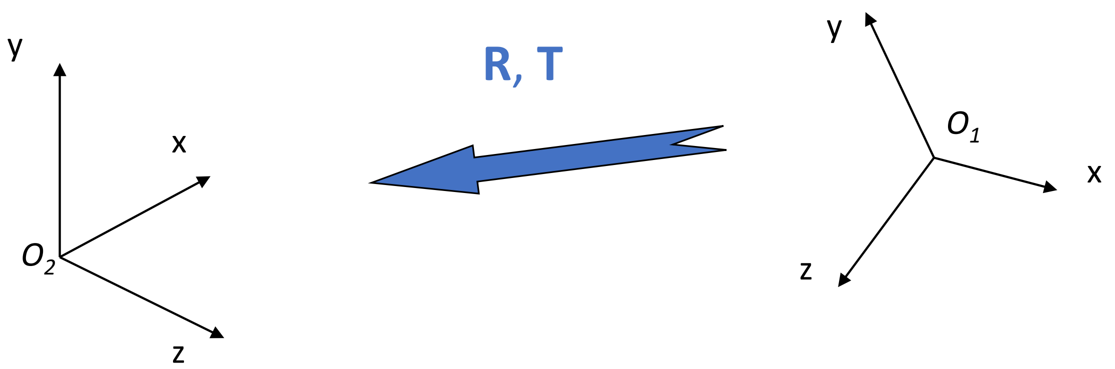
Image frame

2.2 Between the Camera and World Frames

The mapping between the two frames is defined by 3D rotation and translation

- **R** matrix - Rotation about some 3D axis through O_1
- **T** vector – Translation of origin from O_1 to O_2

rot + translation.



2.2 Between the Camera and World Frames

Translation

3 parameters.

$$\mathbf{P}' = \mathbf{P} + \mathbf{T} \qquad \begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \begin{bmatrix} T_X \\ T_Y \\ T_Z \end{bmatrix}$$

Rotation

$$\mathbf{P}' = \mathbf{R}\mathbf{P} \qquad \begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = \overbrace{\begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}}^{3\times3 \text{ symmetric}} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

2.2 Between the Camera and World Frames

Combined: rotation about origin of first frame, followed by translation

$$\mathbf{P} = \mathbf{R}\mathbf{P}_w + \mathbf{T}$$

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} + \begin{bmatrix} T_X \\ T_Y \\ T_Z \end{bmatrix}$$

2.2 Between the Camera and World Frames

A neater form

homogeneous coordinate.

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & T_X \\ r_{21} & r_{22} & r_{23} & T_Y \\ r_{31} & r_{32} & r_{33} & T_Z \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

3D point in the camera frame

Rigid transform

3D point in the world frame

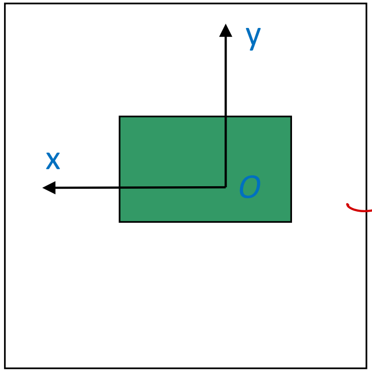
2.3 Between the Image and Pixel Frames

(0,0) → x+

Pixel sys

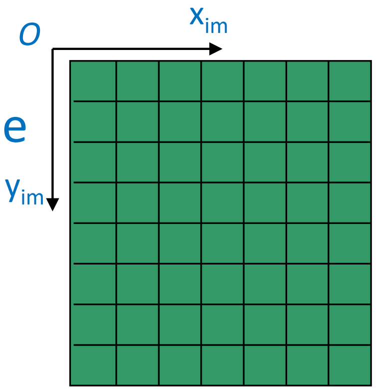
Image frame

y+



→ Image

Pixel frame



check res with camera

$$\begin{cases} x = -(x_{im} - o_x)s_x \\ y = -(y_{im} - o_y)s_y \end{cases}$$

(o_x,o_y): the coordinates in pixel of the image center

(s_x,s_y): the effective physical size of the pixel

$$\begin{cases} x_{im} = -x/s_x + o_x \\ y_{im} = -y/s_y + o_y \end{cases}$$

An equivalent form

2.3 Between the Image and Pixel Frames

A neater form

offset

$$\begin{bmatrix} x_{im} \\ y_{im} \\ 1 \end{bmatrix} = \begin{bmatrix} -1/s_x & 0 & o_x \\ 0 & -1/s_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

2D point in the pixel frame

discretization

2D point in the image frame

2.4 Between the World and Pixel Frames

$$\begin{bmatrix} kx \\ ky \\ k \end{bmatrix} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

$$\begin{bmatrix} x_{im} \\ y_{im} \\ 1 \end{bmatrix} = \begin{bmatrix} -1/s_x & 0 & o_x \\ 0 & -1/s_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$\times k$ $\qquad$ $\times k$

$$\begin{bmatrix} kx_{im} \\ ky_{im} \\ k \end{bmatrix} = \begin{bmatrix} -1/s_x & 0 & o_x \\ 0 & -1/s_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} kx \\ ky \\ k \end{bmatrix}$$

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & T_X \\ r_{21} & r_{22} & r_{23} & T_Y \\ r_{31} & r_{32} & r_{33} & T_Z \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

2.4 Between the World and Pixel Frames

$$\begin{bmatrix} kx_{im} \\ ky_{im} \\ k \end{bmatrix} = \begin{bmatrix} -1/s_x & 0 & o_x \\ 0 & -1/s_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & T_X \\ r_{21} & r_{22} & r_{23} & T_Y \\ r_{31} & r_{32} & r_{33} & T_Z \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

From the image frame
to the pixel frame

From the world frame
to the camera frame

image point
in the pixel frame

From the camera frame to
the image frame
(perspective projection)

3D point in
world frame

2.4 Between the World and Pixel Frames

$$\begin{bmatrix} kx_{im} \\ ky_{im} \\ k \end{bmatrix} = \underbrace{\begin{bmatrix} -f/s_x & 0 & o_x \\ 0 & -f/s_y & o_y \\ 0 & 0 & 1 \end{bmatrix}}_{\text{internal camera}} \underbrace{\begin{bmatrix} r_{11} & r_{12} & r_{13} & T_X \\ r_{21} & r_{22} & r_{23} & T_Y \\ r_{31} & r_{32} & r_{33} & T_Z \end{bmatrix}}_{\text{external camera -}} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

M_{int} $\qquad$ M_{ext}

Involves only intrinsic parameters

from the camera (CCD

Involves only extrinsic parameters

from env.

2.4 Between the World and Pixel Frames

one-to-many: many k → multiple depth ~ scaling factor.

$$\begin{bmatrix} kx_{im} \\ ky_{im} \\ k \end{bmatrix} = \begin{bmatrix} -fr_{11}/s_x + o_x r_{31} & -fr_{12}/s_x + o_x r_{32} & -fr_{13}/s_x + o_x r_{33} & -fT_X/s_x + o_x T_Z \\ -fr_{21}/s_y + o_y r_{31} & -fr_{22}/s_y + o_y r_{32} & -fr_{23}/s_y + o_y r_{33} & -fT_Y/s_x + o_y T_Z \\ r_{31} & r_{32} & r_{33} & T_Z \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

$$= \underbrace{\begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix}}_{M} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

→ normalize ⇒ $\underline{\underline{m_{34} = 1}}$

M Projection matrix

What is the degree of freedom of M?

11 unknowns.
→ 5 internal parameters.
→ 6 external. parameters

2.5 Summary

- The methodology
  - Build the basic model that link the camera frame and the image frame
  - Link the camera and world frames
  - Link the image and pixel frames
  - Build the final model between the world and pixel frames
- A scene point and its image point can be linked by a 3*4 projection matrix M which is useful and important for 3D reconstruction.
- But M is unknown yet. To make it known, we need camera calibration.

3. Camera Calibration

The model we have obtained

$$\begin{bmatrix} kx_{im} \\ ky_{im} \\ k \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

In the vision problem, what we know is

$$\begin{bmatrix} x_{im} \\ y_{im} \end{bmatrix}$$

What we want to know is

$$\begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix}$$

The difficulty is that we do not know

$$\begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & 1 \end{bmatrix}$$

The good news is, if the two vectors are known, *M* can be made known. With M, we can get the intrinsic matrix to achieve **camera calibration**.

3. A Mathematical Solution

The above relationship can be written as

$$\begin{cases} kx_{im} = m_{11}X_w + m_{12}Y_w + m_{13}Z_w + m_{14} \\ ky_{im} = m_{21}X_w + m_{22}Y_w + m_{23}Z_w + m_{24} \\ k = m_{31}X_w + m_{32}Y_w + m_{33}Z_w + 1 \end{cases}$$

And further rearranged as

$$\begin{cases} X_w m_{11} + Y_w m_{12} + Z_w m_{13} + m_{14} - x_{im}X_w m_{31} - x_{im}Y_w m_{32} - x_{im}Z_w m_{33} = x_{im} \\ X_w m_{21} + Y_w m_{22} + Z_w m_{23} + m_{24} - y_{im}X_w m_{31} - y_{im}Y_w m_{32} - y_{im}Z_w m_{33} = y_{im} \end{cases}$$

These are two equations but 11 unknowns m_{ij}

3. A Mathematical Solution

If we have *n* point correspondences, we have 2*n* equations

$$X_w^1 m_{11} + Y_w^1 m_{12} + Z_w^1 m_{13} + m_{14} - x_{im}^1 X_w^1 m_{31} - x_{im}^1 Y_w^1 m_{32} - x_{im}^1 Z_w^1 m_{33} = x_{im}^1$$

$$X_w^1 m_{21} + Y_w^1 m_{22} + Z_w^1 m_{23} + m_{24} - y_{im}^1 X_w^1 m_{31} - y_{im}^1 Y_w^1 m_{32} - y_{im}^1 Z_w^1 m_{33} = y_{im}^1$$

$$X_w^2 m_{11} + Y_w^2 m_{12} + Z_w^2 m_{13} + m_{14} - x_{im}^2 X_w^2 m_{31} - x_{im}^2 Y_w^2 m_{32} - x_{im}^2 Z_w^2 m_{33} = x_{im}^2$$

$$X_w^2 m_{21} + Y_w^2 m_{22} + Z_w^2 m_{23} + m_{24} - y_{im}^2 X_w^2 m_{31} - y_{im}^2 Y_w^2 m_{32} - y_{im}^2 Z_w^2 m_{33} = y_{im}^2$$

$$\ldots$$

$$X_w^n m_{11} + Y_w^n m_{12} + Z_w^n m_{13} + m_{14} - x_{im}^n X_w^n m_{31} - x_{im}^n Y_w^n m_{32} - x_{im} Z_w^n m_{33} = x_{im}^n$$

$$X_w^n m_{21} + Y_w^n m_{22} + Z_w^n m_{23} + m_{24} - y_{im}^n X_w^n m_{31} - y_{im}^n Y_w^n m_{32} - y_{im}^n Z_w^n m_{33} = y_{im}^n$$

3. A Mathematical Solution

We form a linear equation system Au = v

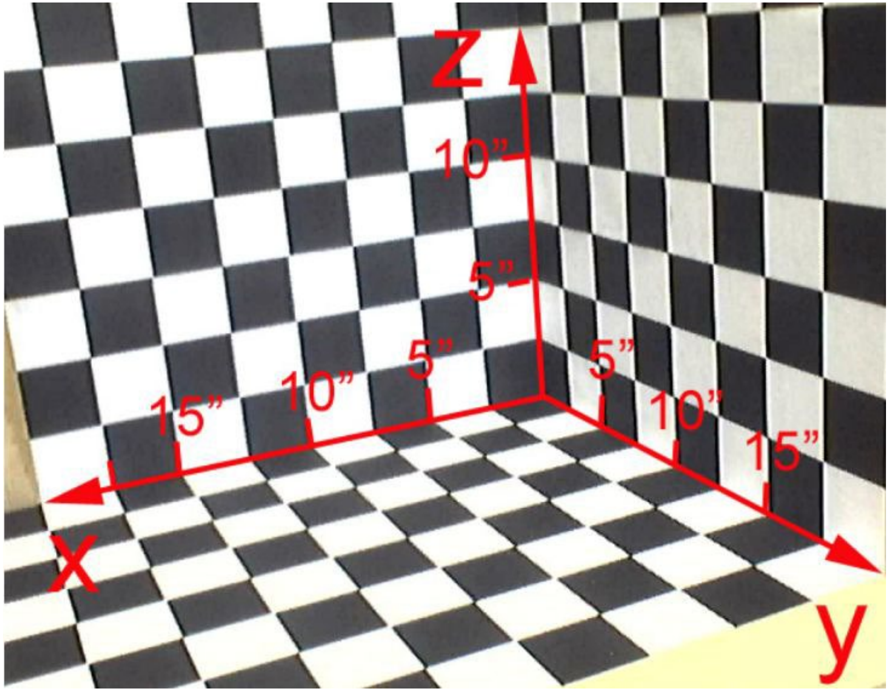
$$
\begin{bmatrix}
X_w^1 & Y_w^1 & Z_w^1 & 1 & 0 & 0 & 0 & 0 & -x_{im}^1 X_w^1 & -x_{im}^1 Y_w^1 & -x_{im}^1 Z_w^1 \\
0 & 0 & 0 & 0 & X_w^1 & Y_w^1 & Z_w^1 & 1 & -y_{im}^1 X_w^1 & -y_{im}^1 Y_w^1 & -y_{im}^1 Z_w^1 \\
X_w^2 & Y_w^2 & Z_w^2 & 1 & 0 & 0 & 0 & 0 & -x_{im}^2 X_w^2 & -x_{im}^2 Y_w^2 & -x_{im}^2 Z_w^2 \\
0 & 0 & 0 & 0 & X_w^2 & Y_w^2 & Z_w^2 & 1 & -y_{im}^2 X_w^2 & -x_{im}^2 Y_w^2 & -y_{im}^2 Z_w^2 \\
. & . & . & . & . & . & . & . & . & . & . \\
. & . & . & . & . & . & . & . & . & . & . \\
X_w^n & Y_w^n & Z_w^n & 1 & 0 & 0 & 0 & 0 & -x_{im}^n X_w^n & -x_{im}^n Y_w^n & -x_{im}^n Z_w^n \\
0 & 0 & 0 & 0 & X_w^n & Y_w^n & Z_w^n & 1 & -y_{im}^n X_w^n & -x_{im}^n Y_w^n & -y_{im}^n Z_w^n
\end{bmatrix}
\begin{bmatrix}
m_{11} \\ m_{12} \\ m_{13} \\ m_{14} \\ m_{21} \\ m_{22} \\ m_{23} \\ m_{24} \\ m_{31} \\ m_{32} \\ m_{33}
\end{bmatrix}
=
\begin{bmatrix}
x_{im}^1 \\ y_{im}^1 \\ x_{im}^2 \\ y_{im}^2 \\ . \\ . \\ x_{im}^n \\ y_{im}^n
\end{bmatrix}
$$

3. A Mathematical Solution

- The projection matrix can be determined by matrix inverse $\mathbf{u} = \mathbf{A}^{-1}\mathbf{v}$ (when the matrix is square). The MATLAB function is inv(A).
- Or by matrix psudo-inverse $\mathbf{u} = \mathbf{A}^{+}\mathbf{v}$ (when the matrix is non-square). The MATLAB function is pinv(A).
- 6 point pairs are sufficient, but we often use more points to make the solution more robust.
- From the projection matrix *M*, we can further get all intrinsic and extrinsic parameters.
- Now we need to find scene and image point pairs

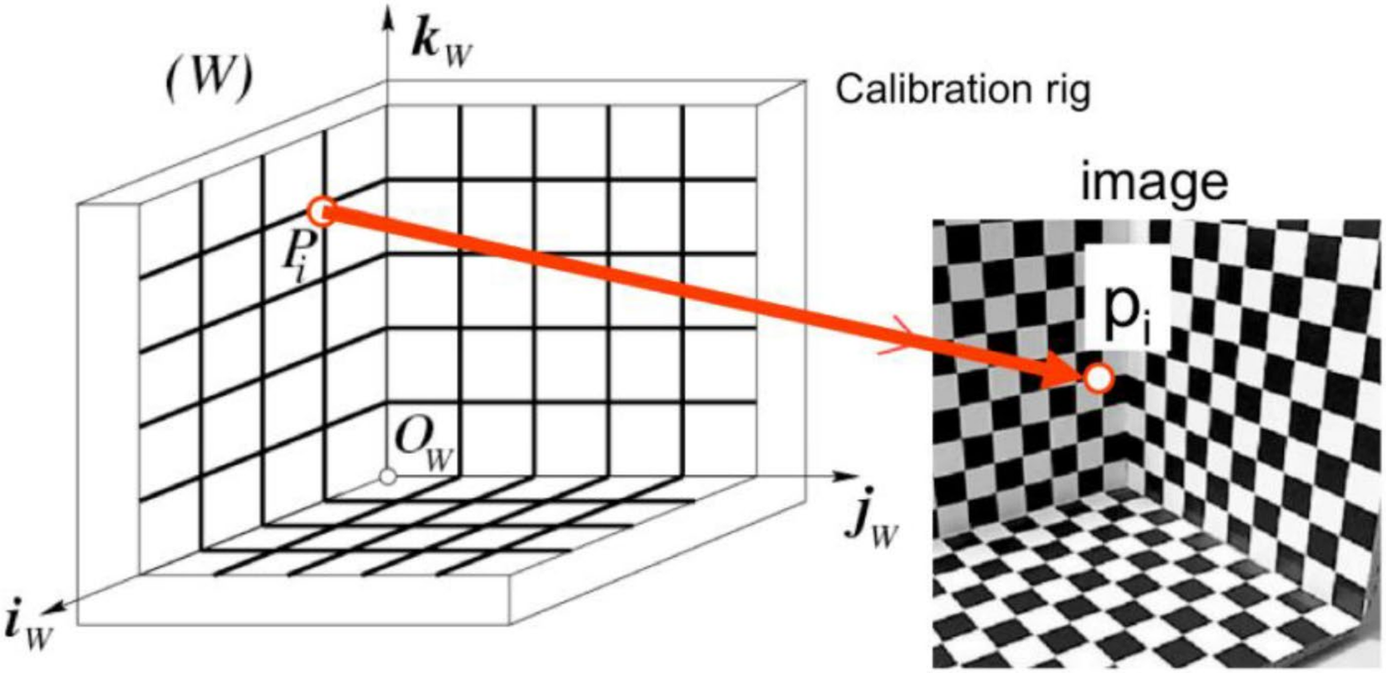
3. Practical Way to Find Point Pairs

Calibration Chart or cube consists of a known structure fabricated with high accuracy, so the coordinates of corners in the world frame are known.



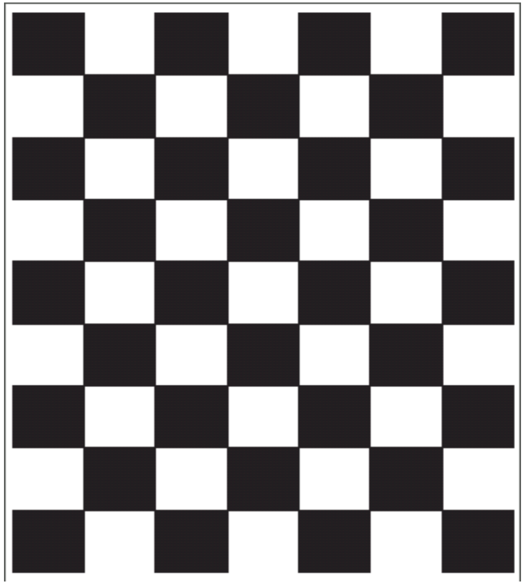
3. Practical Way to Find Point Pairs

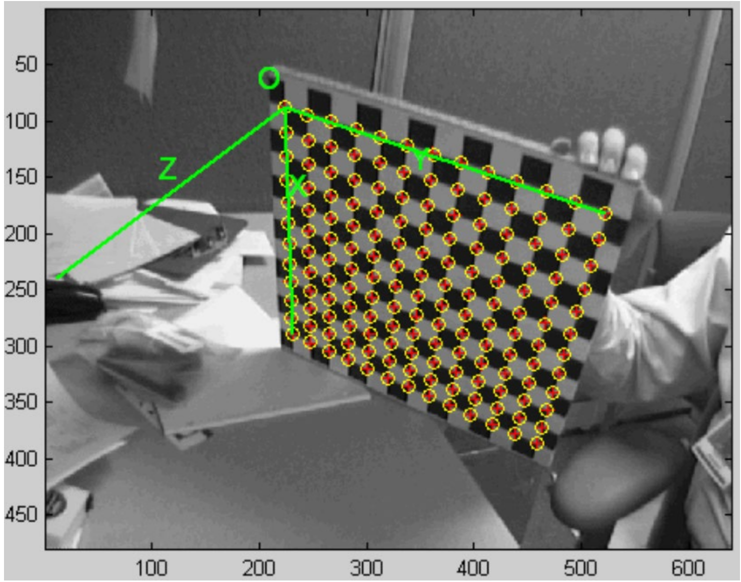
The corners in the image are detected by edge intersection or corner detection and thus also known.



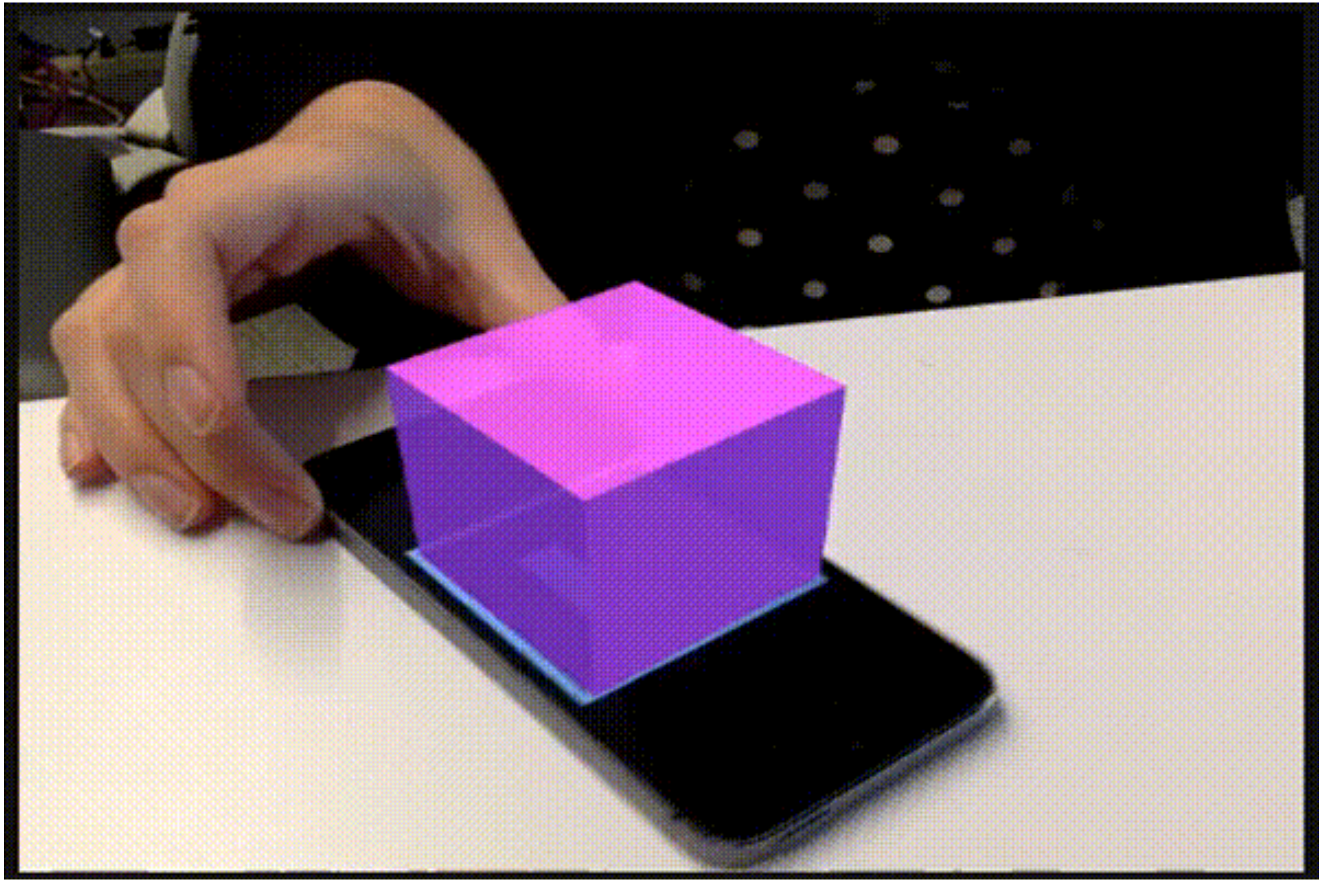
3. Suggested Practical Way to Start

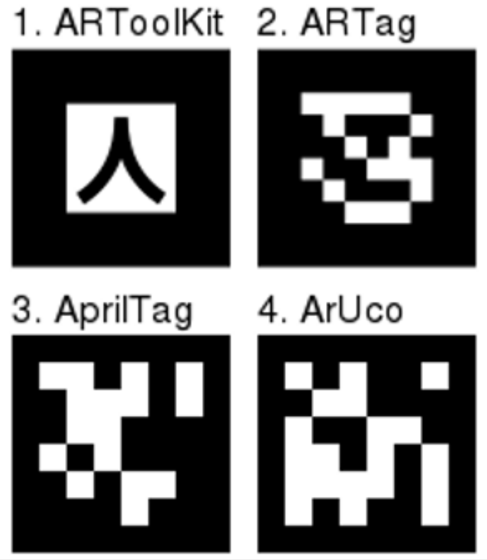
- Download and try the codes available from the internet.
- http://www.vision.caltech.edu/bouguetj/calib_doc/





What about calibration with moving cameras? Need realtime speed.





How is 3D information determined in above example?

4. Special Case I: From Plane to Plane

When the scene points are in a plane

Planar Eq : represent Zw

with Xw, Yw.

$$\begin{bmatrix} kx_{im} \\ ky_{im} \\ k \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ aX_w + \beta Y_w + \gamma \\ 1 \end{bmatrix}$$

By reducing the redundancy, we get

$$\begin{bmatrix} kx_{im} \\ ky_{im} \\ k \end{bmatrix} = \begin{bmatrix} m'_{11} & m'_{12} & m'_{13} \\ m'_{21} & m'_{22} & m'_{23} \\ m'_{31} & m'_{32} & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ 1 \end{bmatrix}$$

→ Panorama in mobile phone.

Q & A

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Copyright Notice