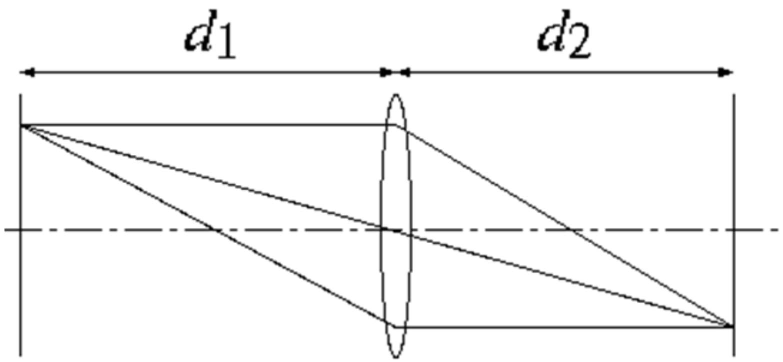
2. Camera models & calibration

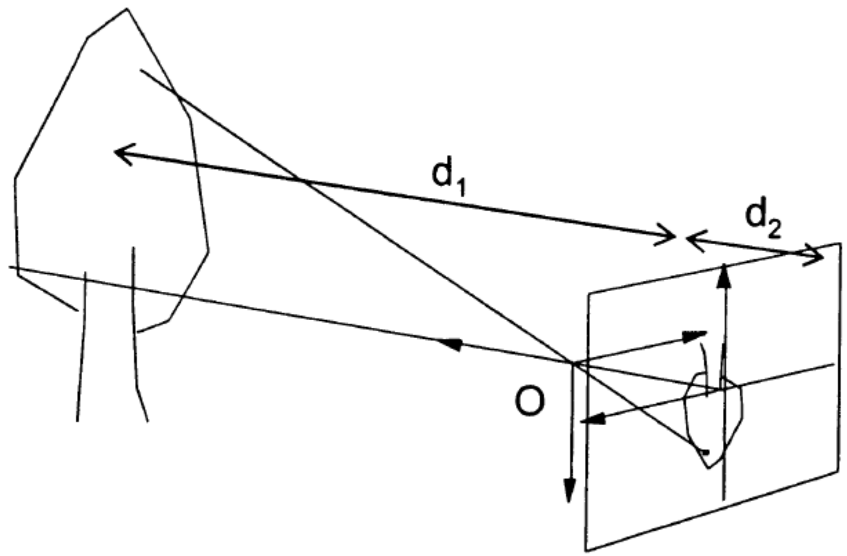
- Idealized model of lens
- Internal parameters of a camera
- External parameters of a camera
- Camera matrix
- Camera calibration
- Image of absolute conic (IAC)

What is an optical lens?

- A device that makes (some of) the rays emitting from scene points focus on points on the image plane
  - There is a relation between the distances from/to the scene and image points, which is known as “Gauss’s law”
  - An “ideal lens” performs central projection

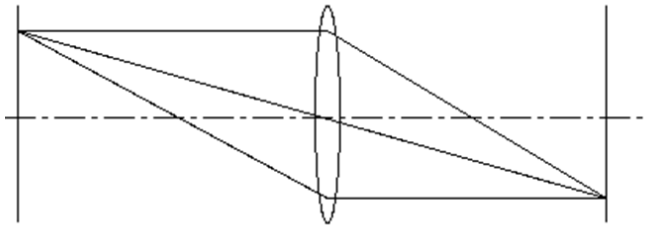


$$\frac{1}{f} = \frac{1}{d_1} + \frac{1}{d_2}$$

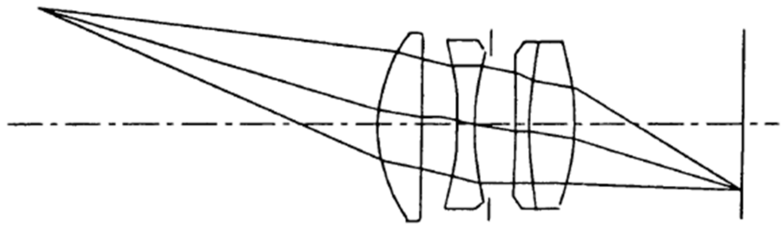


Construction of lenses

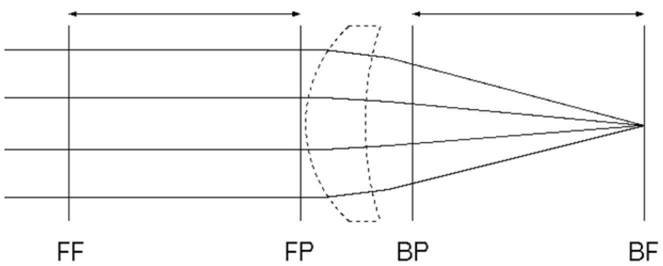
- Although real lenses have to have thickness, we may consider them to be ideal lenses by neglecting the space between the two “principal points”



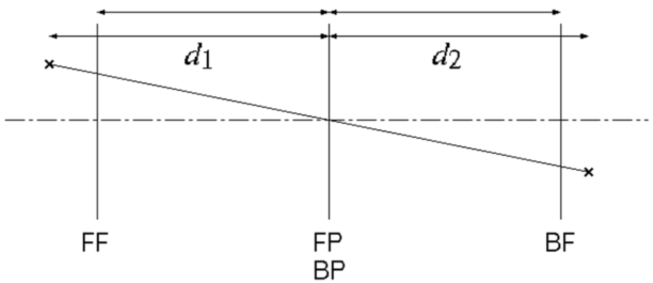
Ideal lens (thin lens)



Real lens



FF: Front focal point
FP: Front principal point
BF: Back focal point
BP: Back principal point



$$\frac{1}{f} = \frac{1}{d_1} + \frac{1}{d_2}$$

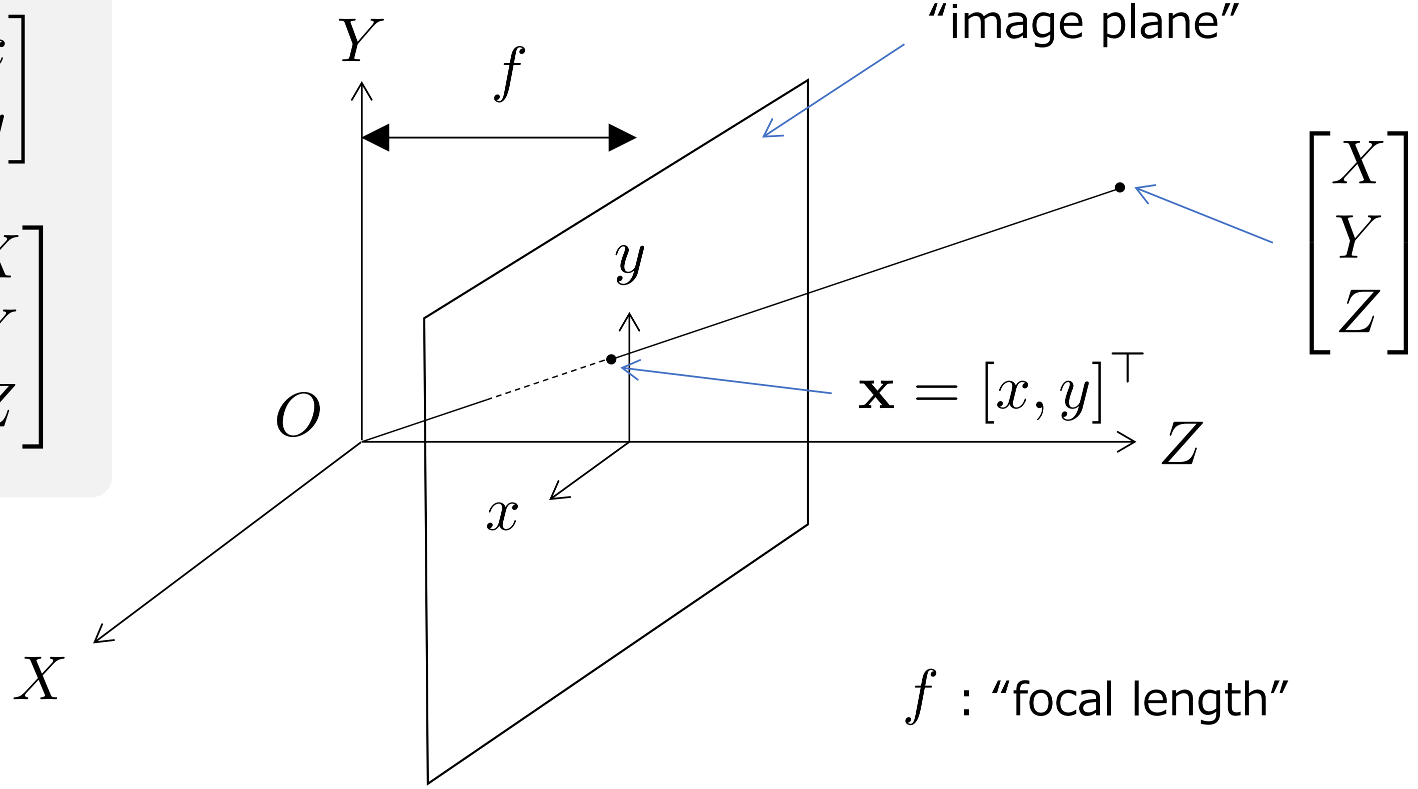
Basic model of a camera

- Perspective camera (pinhole camera model/central projection)

$$x = f\frac{X}{Z} \quad y = f\frac{Y}{Z} \Leftrightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \propto \begin{bmatrix} fX \\ fY \\ Z \end{bmatrix} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

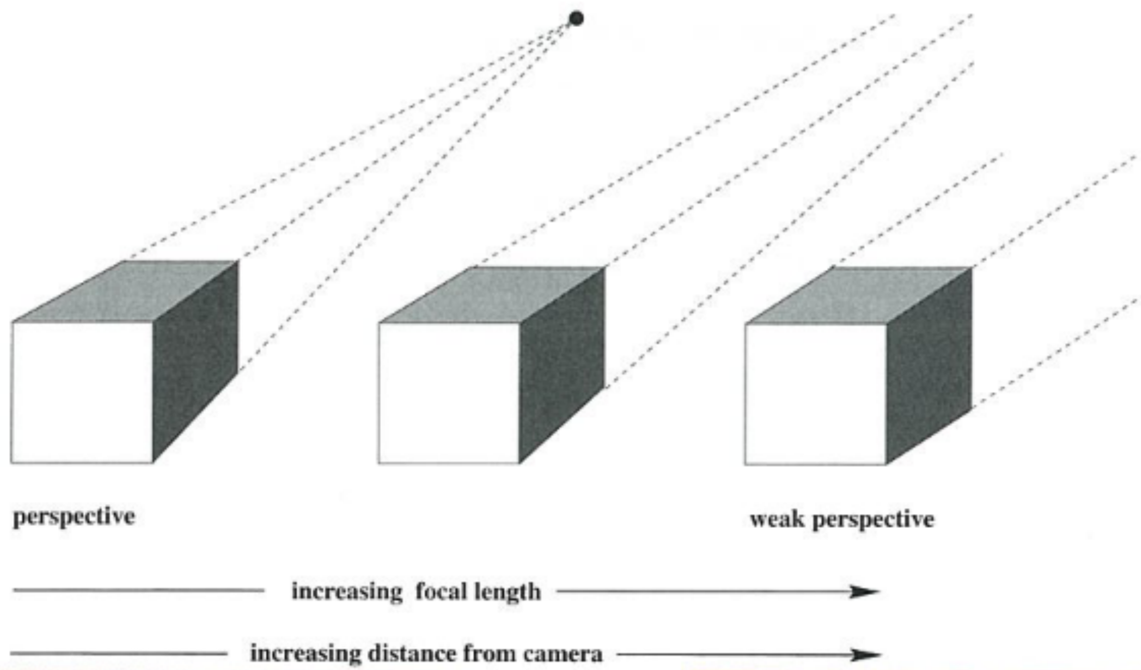
image coordinates $\begin{bmatrix} x \\ y \end{bmatrix}$

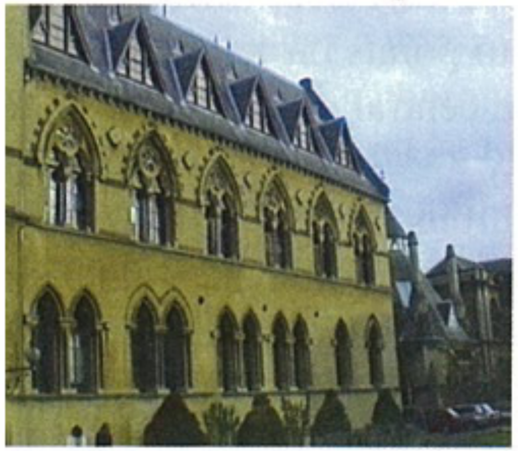
camera coordinates $\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$



f : “focal length”

Focal length and field of view

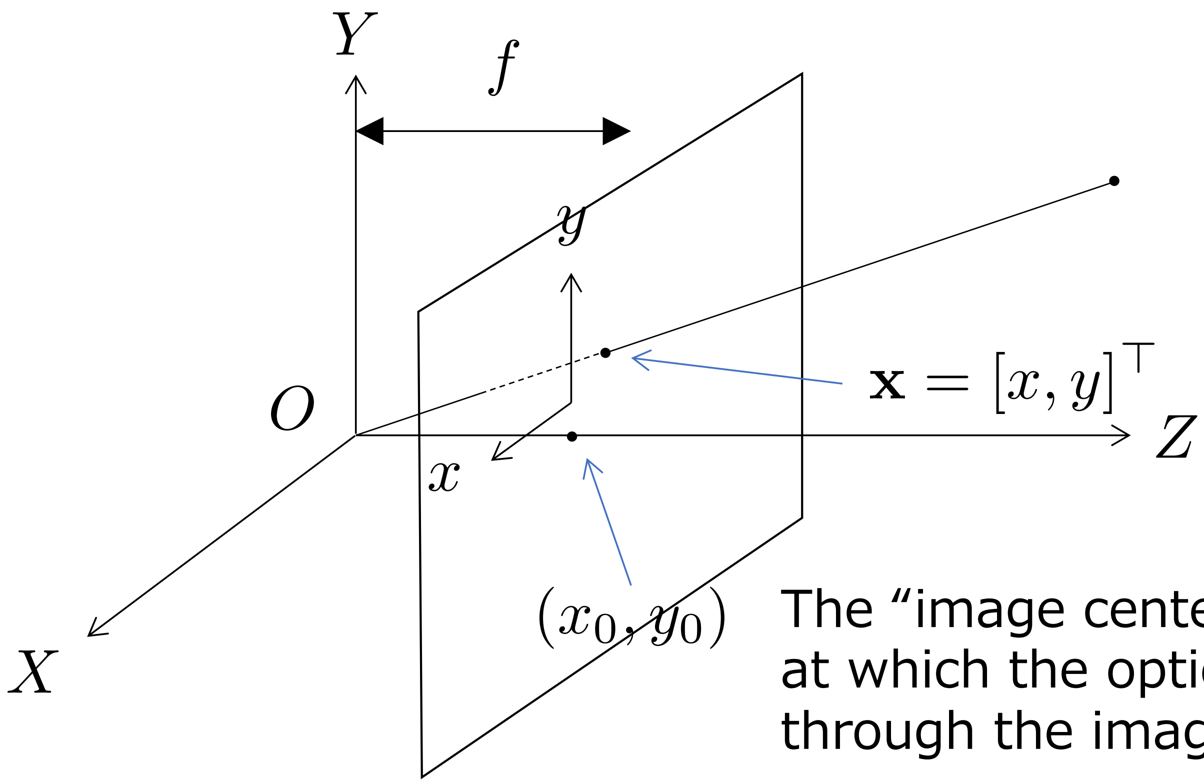




A slightly generalized model

- Optical axis (Z) does not pass through the origin of the xy coord.

$$x = f\frac{X}{Z} + x_0 \qquad y = f\frac{Y}{Z} + y_0 \quad \Leftrightarrow \quad \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \propto \begin{bmatrix} fX + x_0 Z \\ fY + y_0 Z \\ Z \end{bmatrix} = \begin{bmatrix} f & 0 & x_0 \\ 0 & f & y_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$



The “image center”: the point at which the optical axis passes through the image plane

Further generalization

- The most general model

$$\mathbf{x} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \propto \begin{bmatrix} f & sf & x_0 \\ 0 & \alpha f & y_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \mathtt{K}\mathbf{X}$$

$$s : \text{skew}$$

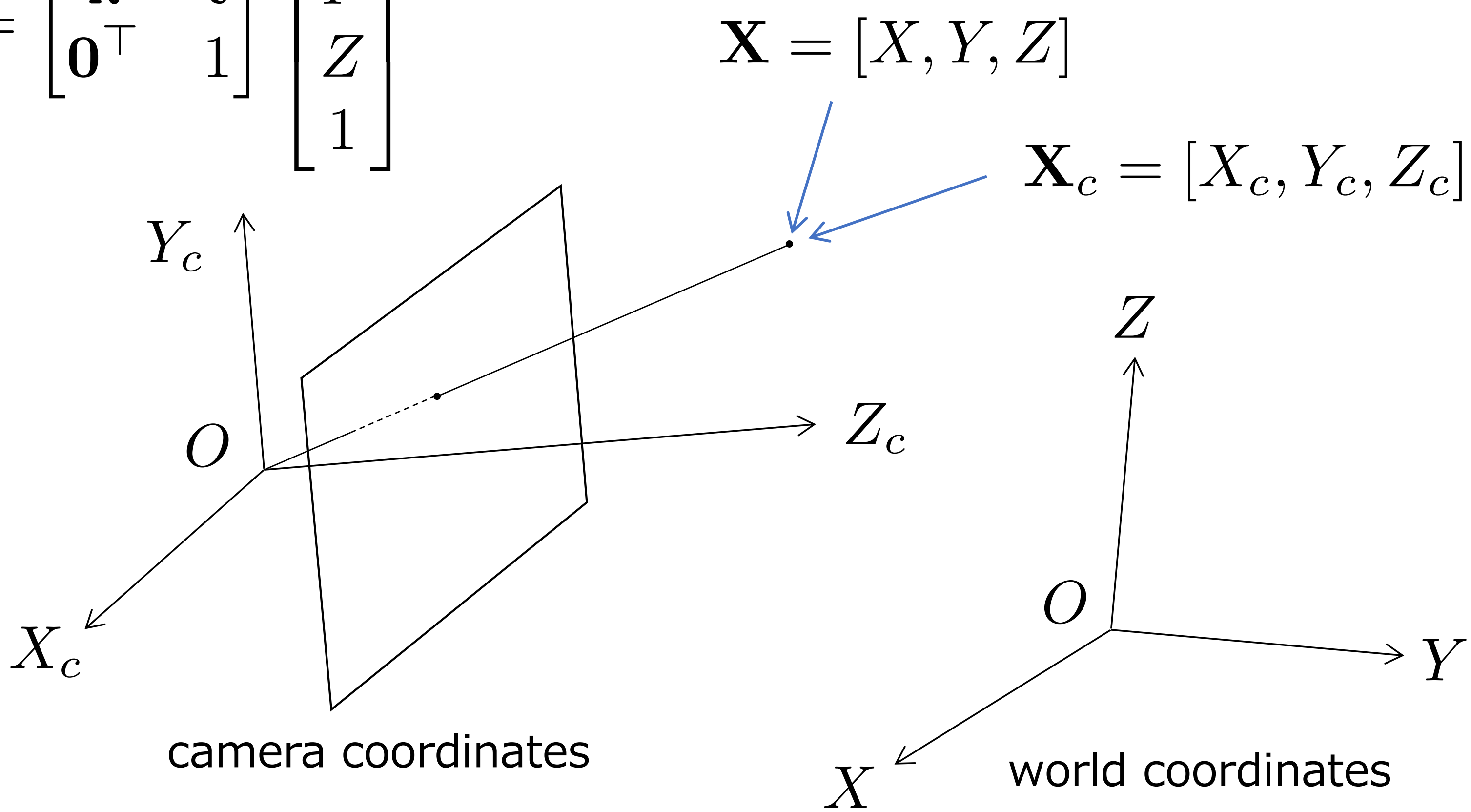
$$\alpha : \text{aspect ratio}$$

- K contains all the optical parameters $\mathtt{K} = \begin{bmatrix} f & sf & x_0 \\ 0 & \alpha f & y_0 \\ 0 & 0 & 1 \end{bmatrix}$
  - K is called a (camera) internal matrix
  - Note the shape of K; K is a upper triangular matrix

Representation of poses of a camera

- Camera pose = A coordinate transformation from the world coordinate system to the camera coordinate system
  - An identical point has two different representation

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} \mathtt{R} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$



Camera matrix

- How is a point having the world coordinates [X,Y,Z] projected onto the image plane?
  - An integrated transformation cascading the two transformations

$$\mathbf{x} \propto \mathtt{K}\mathbf{X}_c = \underbrace{\mathtt{K}}_{\text{Internal parameters}} \underbrace{\begin{bmatrix}\mathtt{R} & \mathbf{t}\end{bmatrix}\begin{bmatrix}X\\Y\\Z\\1\end{bmatrix}}_{\text{External parameters}} = \mathtt{P}\mathbf{X}$$

camera matrix:

$$\mathtt{P} = \mathtt{K}\begin{bmatrix}\mathtt{R} & \mathbf{t}\end{bmatrix}$$

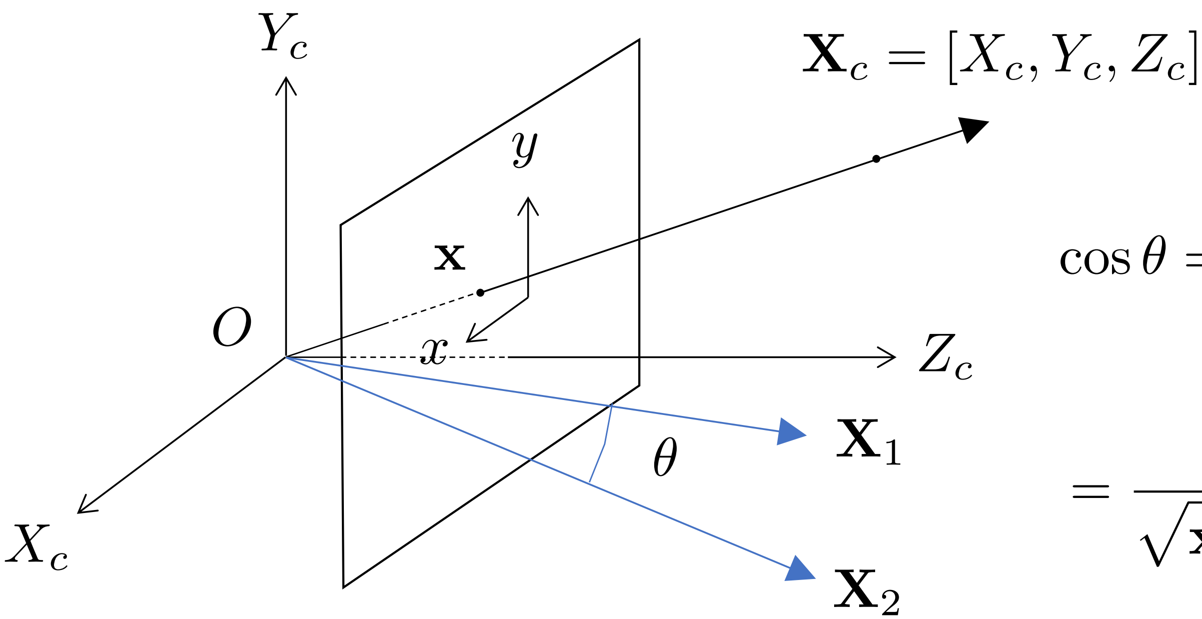
$$\mathbf{x} =\propto \begin{bmatrix} f & sf & x_0 \\ 0 & \alpha f & y_0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix}X_c\\Y_c\\Z_c\end{bmatrix} = \mathtt{K}\mathbf{X}_c$$

$$\begin{bmatrix}X_c\\Y_c\\Z_c\end{bmatrix} = \begin{bmatrix}\mathtt{R} & \mathbf{t}\end{bmatrix}\begin{bmatrix}X\\Y\\Z\\1\end{bmatrix}$$

Camera calibration

- Calibration of a camera is to know the matrix K of the camera
- If K (and camera pose) is known, the ray of an image point can be obtained

$$\mathbf{x} \propto \mathtt{K}\mathbf{X}_c \quad \Leftrightarrow \quad \mathbf{X}_c \propto \mathtt{K}^{-1}\mathbf{x}$$



$$\cos\theta = \frac{\mathbf{X}_1^\top \mathbf{X}_2}{\sqrt{\mathbf{X}_1\mathbf{X}_1}\sqrt{\mathbf{X}_2\mathbf{X}_2}}$$

$$= \frac{\mathbf{x}_1\mathtt{K}^{-\top}\mathtt{K}^{-1}\mathbf{x}_2}{\sqrt{\mathbf{x}_1\mathtt{K}^{-\top}\mathtt{K}^{-1}\mathbf{x}_1}\sqrt{\mathbf{x}_2\mathtt{K}^{-\top}\mathtt{K}^{-1}\mathbf{x}_2}}$$

A simple (but cumbersome) calibration method

- Using a reference object with known 3D shape
  - Camera matrix P is computed from point correspondences
  - A single point gives two equations constraining P
  - P has 12 elements (11 DoFs) → 6 pairs or more to determine P

$$\mathbf{x}_i \propto \mathtt{P}\mathtt{X}_i \ (i = 1, 2, \ldots) \qquad \mathtt{P} = \mathtt{K}\begin{bmatrix}\mathtt{R} & \mathbf{t}\end{bmatrix}$$

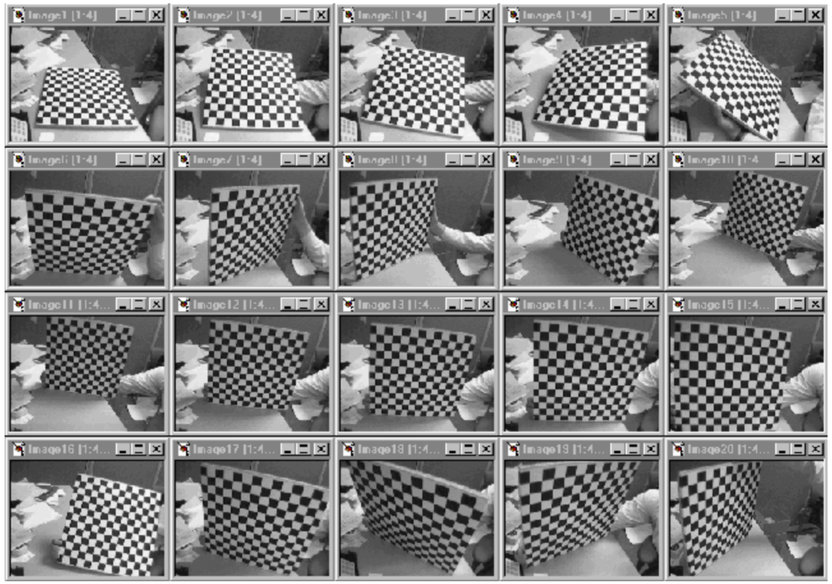
  - Decompose (the first 3x3 submatrix) of P into K and R using QR decomposition (actually 'RQ' decomposition)
  - QR decomposition: Any square matrix can be uniquely decomposed into a product of an orthogonal mat. Q (Q') and upper-right triangular mat. R (R')

$$\mathtt{A} = \mathtt{QR} = \mathtt{R}'\mathtt{Q}'$$

http://www.fibus.org/fibustrk.htm

More methods for calibration

- Standard: "Easy camera calibration" [Zhang98]
  - Three or more images of a plane with known pattern
  - Regarded as a standard procedure for years
- Single image calibration without a reference object
  - Is it possible to calibrate a camera from a single image without an object of known shape?
  - Yes; instead, some info about the scene is necessary
  - Formulated as identification of image of the absolute conic (IAC)



http://www.vision.caltech.edu/bouguetj/calib_doc/htmls/example.html

Easy camera calibration --- OpenCV python code

```
import numpy, cv2

cap = cv2.VideoCapture(1)
cap.set(cv2.cv.CV_CAP_PROP_FRAME_WIDTH, 640)
cap.set(cv2.cv.CV_CAP_PROP_FRAME_HEIGHT, 480)

patw, path = 7, 6
objp = numpy.zeros((patw*path, 3))
for i in range(patw*path):
    objp[i,:2] = numpy.array([i % patw, i / patw], numpy.float32)

objp_list, imgp_list = [], []

while 1:
    stat, image = cap.read(0)

    ret, centers = cv2.findCirclesGrid(image, (patw, path))
    cv2.drawChessboardCorners(image, (patw, path), centers, ret)

    cv2.imshow('Camera', image)
    key = cv2.waitKey(10)

    if key == 0x1b: # ESC
        break
    elif key == 0x20 and ret == True:
        print 'Saved!'
        objp_list.append(objp.astype(numpy.float32))
        imgp_list.append(centers)

if len(objp_list) >= 3:
    K = numpy.zeros((3,3), float)
    dist = numpy.zeros((5,1), float)
    cv2.calibrateCamera(objp_list, imgp_list, (image.shape[1], image.shape[2]), K,
dist)
    print 'K = ¥n', K
    numpy.savetxt('K.txt', K)
    print 'Dist coeff = ¥n', dist
    numpy.savetxt('distCoef.txt', dist)

cap.release()
cv2.destroyAllWindows()
```

Points in 3D space

- A point [X,Y,Z] in a 3D space is represented as

$$\mathbf{X} = \begin{bmatrix} X & Y & Z & 1 \end{bmatrix}^\top$$

  - Scaled vectors represent the same point

$$\begin{bmatrix} X & Y & Z & 1 \end{bmatrix}^\top \propto \begin{bmatrix} kX & kY & kZ & k \end{bmatrix}^\top$$

  - From homogeneous to inhomogeneous coordinates

$$\begin{bmatrix} X_1 & X_2 & X_3 & X_4 \end{bmatrix}^\top \to X = \frac{X_1}{X_4}, Y = \frac{X_2}{X_4}, Z = \frac{X_3}{X_4}$$

  - Points at infinity $\mathbf{X} = \begin{bmatrix} X_1 & X_2 & X_3 & 0 \end{bmatrix}^\top$

  - Finite points $\mathbf{X} = \begin{bmatrix} X_1 & X_2 & X_3 & X_4 \end{bmatrix}^\top \ (X_4 \neq 0)$

Planes

- A plane in a 3D space is given by

$$\pi_1 X + \pi_2 Y + \pi_3 Z + \pi_4 = 0$$

  - Rewritten as $\pi^\top \mathbf{X} = 0 \Leftrightarrow \begin{bmatrix} \pi_1 & \pi_2 & \pi_3 & \pi_4 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = 0$

  - Corresponds to the line in a 2D space
  - Normal n to a plane is represented as

$$\mathbf{n}^\top \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + d = 0$$

  - Distance from a plane to the origin is given by

$$d/\|\mathbf{n}\|$$

The plane at infinity

- A plane $\pi_\infty \equiv \begin{bmatrix}0 & 0 & 0 & 1\end{bmatrix}^\top$ is called the plane at infinity
  - Recall that $\mathbf{l}_\infty \equiv \begin{bmatrix}0 & 0 & 1\end{bmatrix}^\top$ is called the line at infinity
  - Any point at infinity lies on this plane
- Any affine trans. preserves π_∞, and inverse is true

$$\pi'_\infty \propto \mathtt{H}_A^{-\top} \pi_\infty = \begin{bmatrix} \mathtt{A}^{-\top} & \mathbf{0} \\ \mathbf{t}^\top \mathtt{A}^{-\top} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix}$$

$$\Leftrightarrow \quad \mathbf{X}' \propto \mathtt{H}_A \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ 0 \end{bmatrix} = \begin{bmatrix} X'_1 \\ X'_2 \\ X'_3 \\ 0 \end{bmatrix}$$

The absolute conic (AC)

- The absolute conic (AC) = points $\mathbf{X} = \begin{bmatrix} X_1 & X_2 & X_3 & X_4 \end{bmatrix}^\top$ satisfying the following equations:

$$\begin{cases} X_1^2 + X_2^2 + X_3^2 = 0 \\ X_4 = 0 \end{cases}$$

  - Denoted by Ω_∞

  - $X_4 = 0$ → AC lies on π_∞
  - $X_1^2 + X_2^2 + X_3^2 = 0$ → AC is a conic $\mathtt{C} = \mathtt{I}$ on π_∞

$$\begin{bmatrix} X_1 & X_2 & X3 \end{bmatrix} \mathtt{I} \begin{bmatrix} X_1 \\ X_2 \\ X3 \end{bmatrix} = 0$$

Image of the absolute conic (IAC)

- IAC (denoted by ω) = Projection of AC Ω_∞ onto the image
- ω is given by $\mathtt{K}^{-\top}\mathtt{K}^{-1}$
  - Projective trans. $\mathtt{H}$ from π_∞ to the image is given by $\mathtt{H} = \mathtt{KR}$
  - Because any point on π_∞ is represented by $\mathbf{X} = [\mathbf{d}^\top \ 0]^\top$
  - Thus,

$$\mathbf{x} \propto \mathtt{K}\begin{bmatrix}\mathtt{R} & \mathbf{t}\end{bmatrix}\begin{bmatrix}\mathbf{d} \\ 0\end{bmatrix} = \mathtt{KR}\mathbf{d}$$

  - Recall Ω_∞ is a conic $\mathtt{C} = \mathtt{I}$ on π_∞
  - Hence,

$$\omega = \mathtt{H}^{-\top}\mathtt{I}\mathtt{H}^{-1} = (\mathtt{KR})^{-\top}(\mathtt{KR})^{-1}$$

$$= \mathtt{K}^{-\top}\mathtt{R}^{-\top}\mathtt{R}^{-1}\mathtt{K}^{-1} = \mathtt{K}^{-\top}\mathtt{K}^{-1}$$

Image of the absolute conic (IAC)

- Summary: IAC ω is given by $\mathtt{K}^{-\top}\mathtt{K}^{-1}$
  - It depends only on internal parameters $\mathtt{K}$
  - Dual of IAC (DIAC) is given as follows:

$$\omega^* = \mathtt{K}\mathtt{K}^\top$$

  - Once ω or ω^* is identified, we obtain $\mathtt{K}$ by decompose it using the Cholesky decomposition

The Cholesky decomposition: Any symmetric matrix can be decomposed into a product of a upper triangular matrix and its transpose

- Thus, calibrating a camera is equivalent to knowing IAC

How can we obtain IAC ω ?

- Methods using circular points
  - For any plane, its circular points are on the AC, as will be shown later
  - Then, the images of circular points should always on IAC
  - If we have them for different multiple planes, we may compute IAC, because IAC should pass through them on the image
- Methods using vanishing points
  - Knowing IAC (and K) is closely related to knowing orientation of lines in space
  - The image of the point at infinity of a line gives information about its orientation in space

The circular points in plane (2D proj. space)

- The following imaginary points are called the circular points

$$\mathbf{I} = \begin{bmatrix} 1 \\ i \\ 0 \end{bmatrix} \qquad \mathbf{J} = \begin{bmatrix} 1 \\ -i \\ 0 \end{bmatrix}$$

Imaginary unit

$$i^2 = -1$$

- This name comes from the fact that any circle has intersections with $\mathbf{l}_\infty$ at these points

A circle: $x_1^2 + x_2^2 + dx_1x_3 + ex_2x_3 + fx_3^2 = 0$

$\mathbf{l}_\infty$ (set of points at infinity): $x_3 = 0$

Their intersection: $x_1^2 + x_2^2 = 0$

→ $x_1 = ix_2, \text{ or } x_1 = -ix_2$

AC & circular points

- Proposition: Any plane π intersects with the AC Ω_∞ at two points. They are the circular points of π

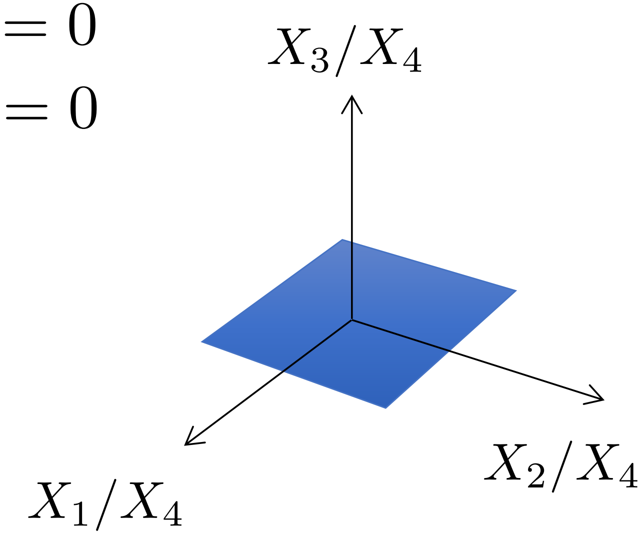
Proof)
We may set $\pi = [0,0,1,0]^\top$ without loss of generality, since any π can be transformed to it by a similarity trans. and AC Ω_∞ is invariant to any similarity trans.

$$\pi = [0,0,1,0]^\top \Leftrightarrow X_3 = 0$$

Substitute $X_3 = 0$ into AC $\left\{\begin{array}{r} X_1^2 + X_2^2 + X_3^2 = 0 \\ X_4 = 0 \end{array}\right.$

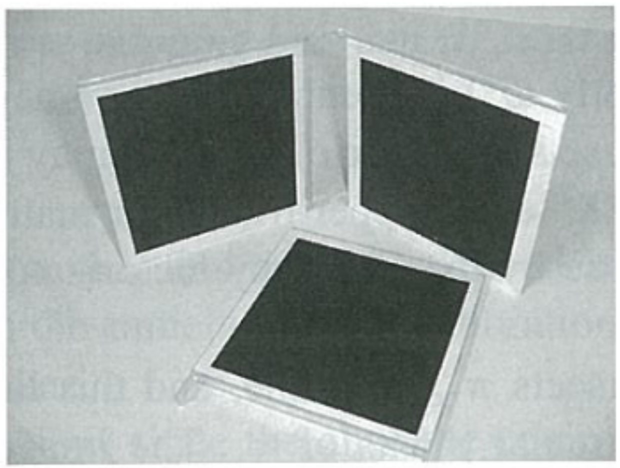
Then we have

$$\left\{\begin{array}{r} X_1^2 + X_2^2 = 0 \\ X_4 = 0 \end{array}\right. \Leftrightarrow \begin{bmatrix} X_1 \\ X_2 \\ X_4 \end{bmatrix} = \begin{bmatrix} 1 \\ \pm i \\ 0 \end{bmatrix}$$



A method for identifying images of circular points

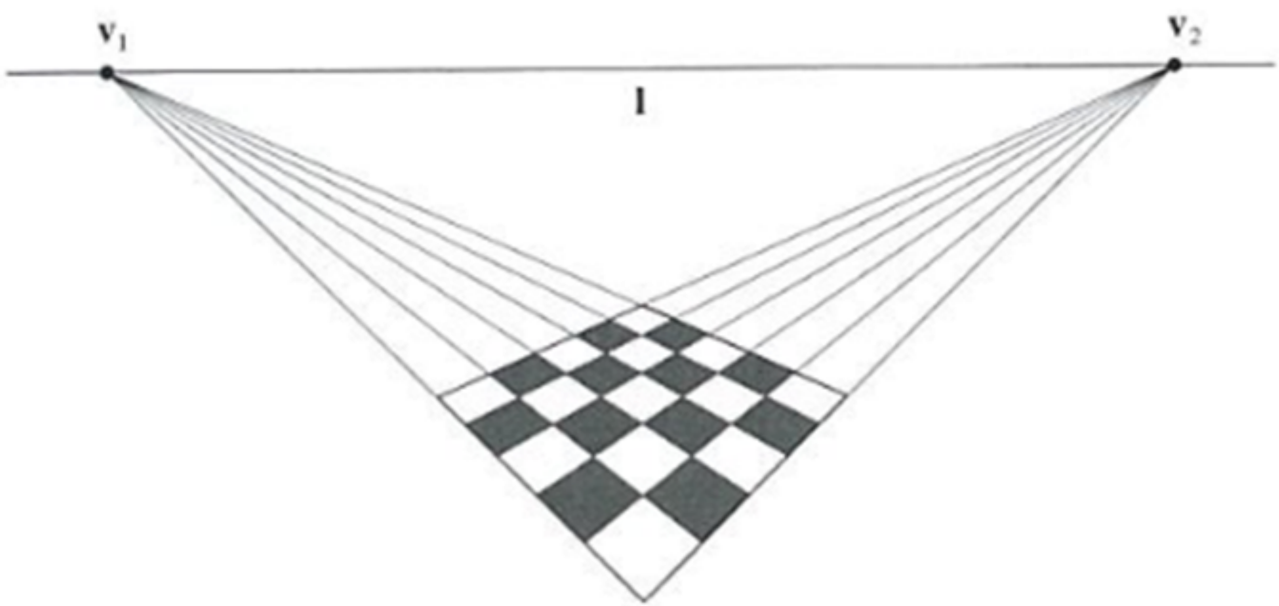
- A plane intersects with the AC Ω_∞ at its circular points
- Consider the images of the circular points; IAC passes through them
- Assume that there are N surfaces in the scene, for each of which the images of its circular points can be obtained → 2N points
- IAC ω should pass through these 2N points
  - IAC has 5 dofs; N must be equal to or greater than 3
- Procedure:
  - For each plane, find H that transforms (0,0), (0,1), (1,0), (1,1) to the corresponding corners
  - Compute the images of the circular points of the plane by $\mathtt{H}\begin{bmatrix}1 & \pm i & 0\end{bmatrix}^\top$
  - Find a conic passing through them, which gives ω!



[Hartley-Zisserman03]

Vanishing points and vanishing lines

- Vanishing point of a line = the image of a point at infinity lying on the line
- Vanishing line of a plane = the image of the plane's line at infinity



IAC and vanishing points

- Suppose a line whose orientation in space is $\mathbf{d} = [d_1, d_2, d_3]^\top$
- The point at infinity lying on the line is given by

$$\mathbf{X}_\infty = [\mathbf{d}^\top \ \ 0]^\top$$

- Its image (vanishing point) $\mathbf{v}$ is represented by

$$\mathbf{v} \propto \mathtt{P}\mathbf{X}_\infty = \mathtt{K}\begin{bmatrix}\mathtt{I} & \mathbf{0}\end{bmatrix}\mathbf{X}_\infty = \mathtt{K}\mathbf{d}$$

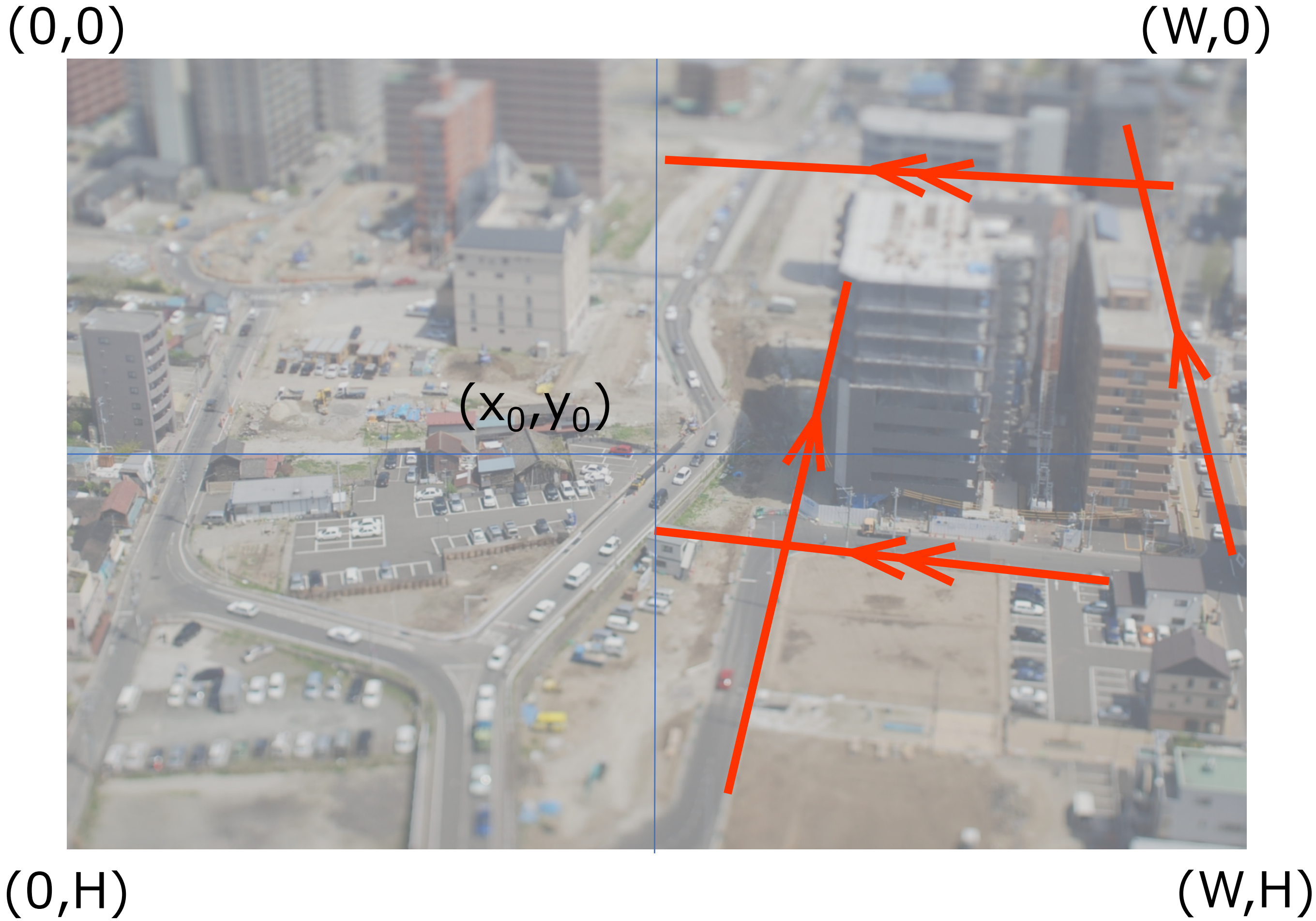
  - The camera coordinate system may be arbitrarily chosen

- Suppose two lines whose vanishing points are $\mathbf{v}_1$ and $\mathbf{v}_2$
- Their angle is given by

$$\cos\theta = \hat{\mathbf{d}_1}^\top \hat{\mathbf{d}_2} = \frac{\mathbf{v}_1^\top \mathtt{K}^{-\top}\mathtt{K}^{-1}\mathbf{v}_2}{\sqrt{\mathbf{v}_1^\top \mathtt{K}^{-\top}\mathtt{K}^{-1}\mathbf{v}_1}\sqrt{\mathbf{v}_2^\top \mathtt{K}^{-\top}\mathtt{K}^{-1}\mathbf{v}_2}} = \frac{\mathbf{v}_1^\top \omega \mathbf{v}_2}{\sqrt{\mathbf{v}_1^\top \omega \mathbf{v}_1}\sqrt{\mathbf{v}_2^\top \omega \mathbf{v}_2}}$$

Using vanishing points to determine IAC ω

- Assume that there are two pair of parallel lines which make the right angle
- Also assume that we know other parameters than focal length



$$K = \begin{bmatrix} f & 0 & x_0 \\ 0 & f & y_0 \\ 0 & 0 & 1 \end{bmatrix}$$

$(x_0, y_0) = (W/2, H/2)$

Using vanishing points to determine IAC ω

- Procedure
  - Identify the vanishing point of the paired parallel lines
  - As we have two pairs, we have two vanishing points $\mathbf{p}_1, \mathbf{p}_2$
  - Solve the following equation to determine f

$$\mathbf{p}_1 \mathtt{K}^{-\top} \mathtt{K}^{-1} \mathbf{p}_2 = 0$$

- 3rd Assignment
  - Use your smart phone to capture an image (of a plane)
  - Compute the focal length of its camera in pixels