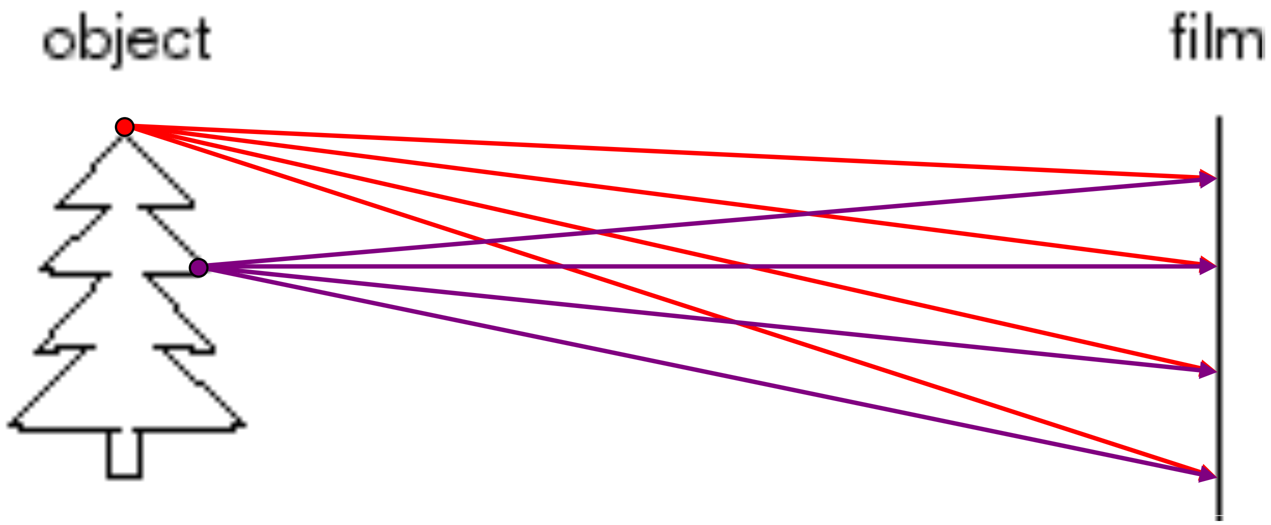
Image Formation

(카메라 캘리브레이션)

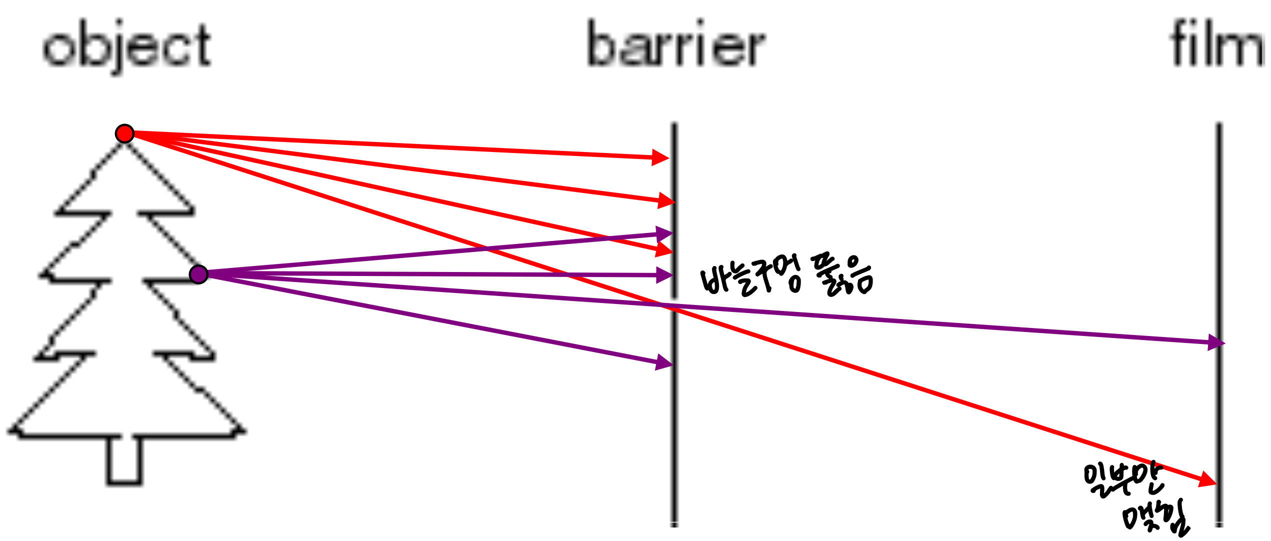
- Camera design
  - Idea 1: put a piece of film in front of an object



  - Do we get a reasonable image?

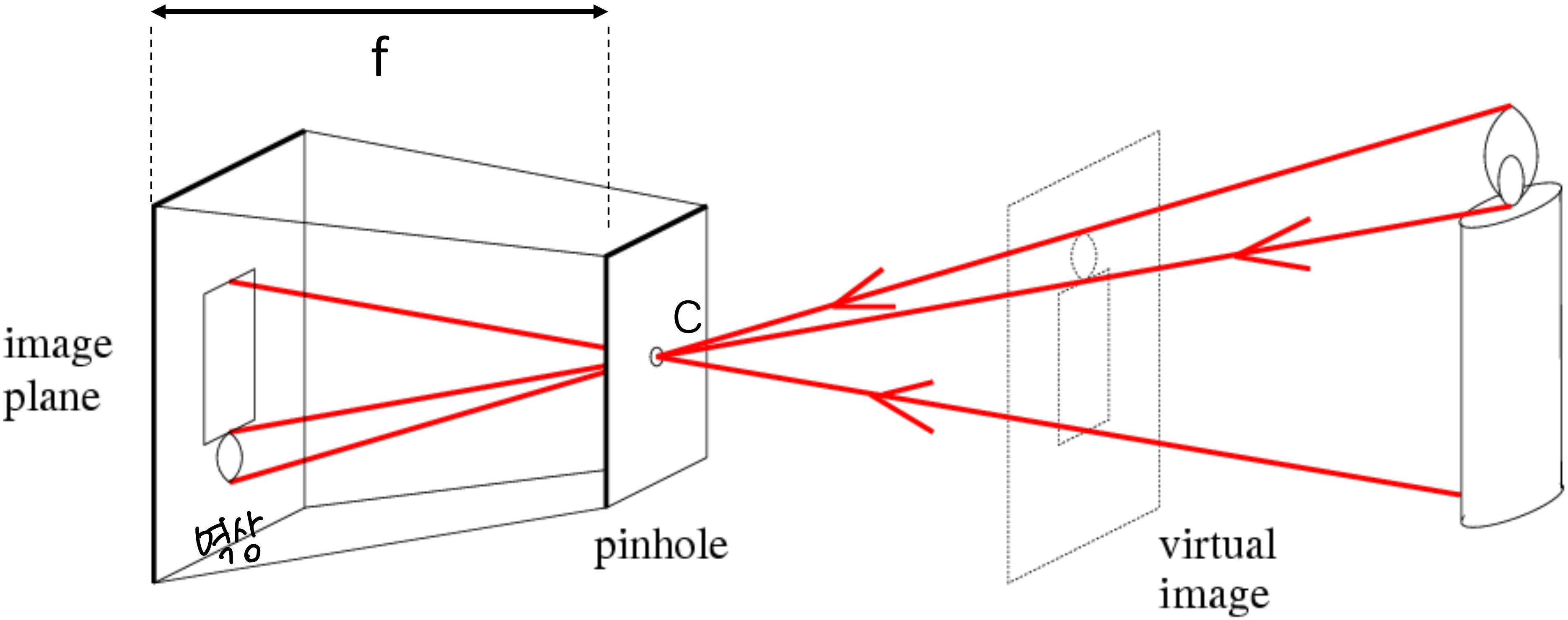
Image Formation

- Pinhole camera
  - Idea 2: add a barrier to block off most of the rays
    - This reduces blurring
    - The opening known as the **aperture**



Pinhole Camera

- Abstract camera model : a box with a small hole

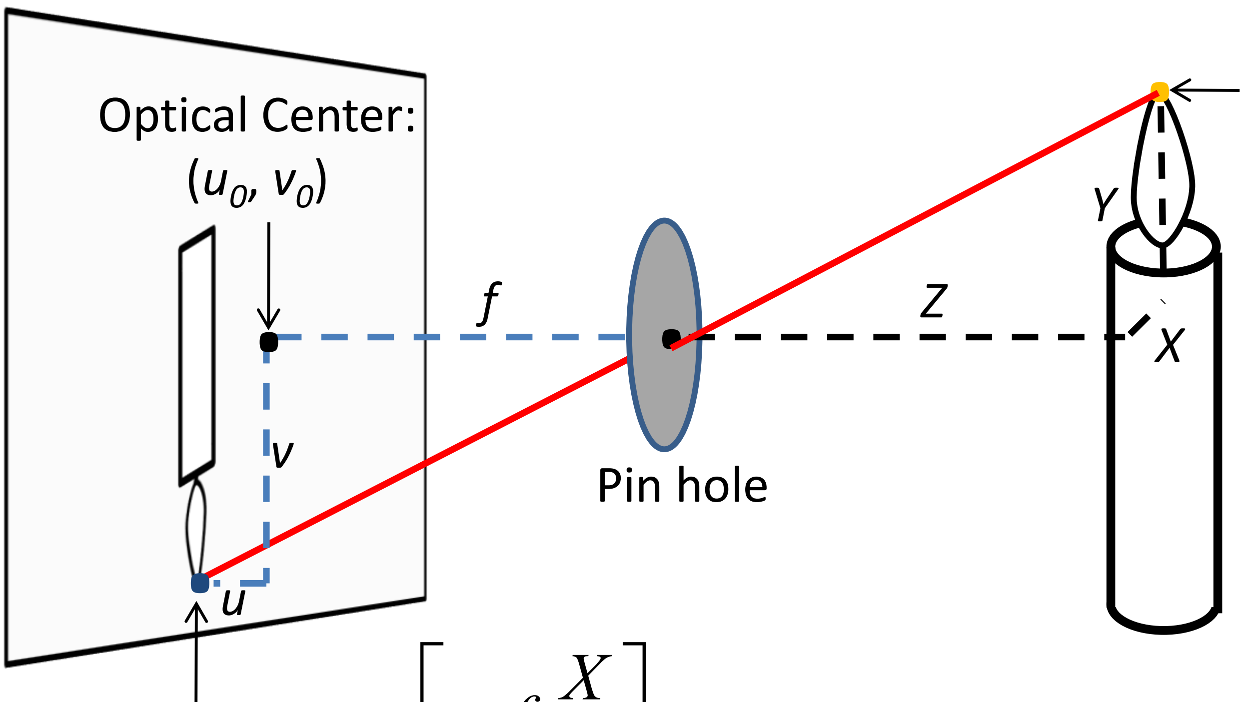


f = focal length (image plane ↔ hole)

c = center of the camera

What is (perspective) Projection?

- 3D world coordinates → 2D image coordinates



$$\mathbf{P} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

$$\mathbf{p} = \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} -f\dfrac{X}{Z} \\ -f\dfrac{Y}{Z} \end{bmatrix}$$

Revisit: Homogeneous Coordinates

- Converting to homogeneous coordinates

$$(x, y) \Rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

homogeneous image coordinates

$$(x, y, z) \Rightarrow \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

homogeneous scene (3d) coordinates

- Invariant to scaling

$$k \begin{bmatrix} x \\ y \\ w \end{bmatrix} = \begin{bmatrix} x \\ y \\ w \end{bmatrix}$$

Homogeneous Coordinates

Revisit: Homogeneous Coordinates

- Converting from homogeneous coordinates to Cartesian.

직교 좌표계

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} \Rightarrow (x/w, y/w)$$

$$= \begin{bmatrix} x/w \\ y/w \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} \Rightarrow (x/w, y/w, z/w)$$

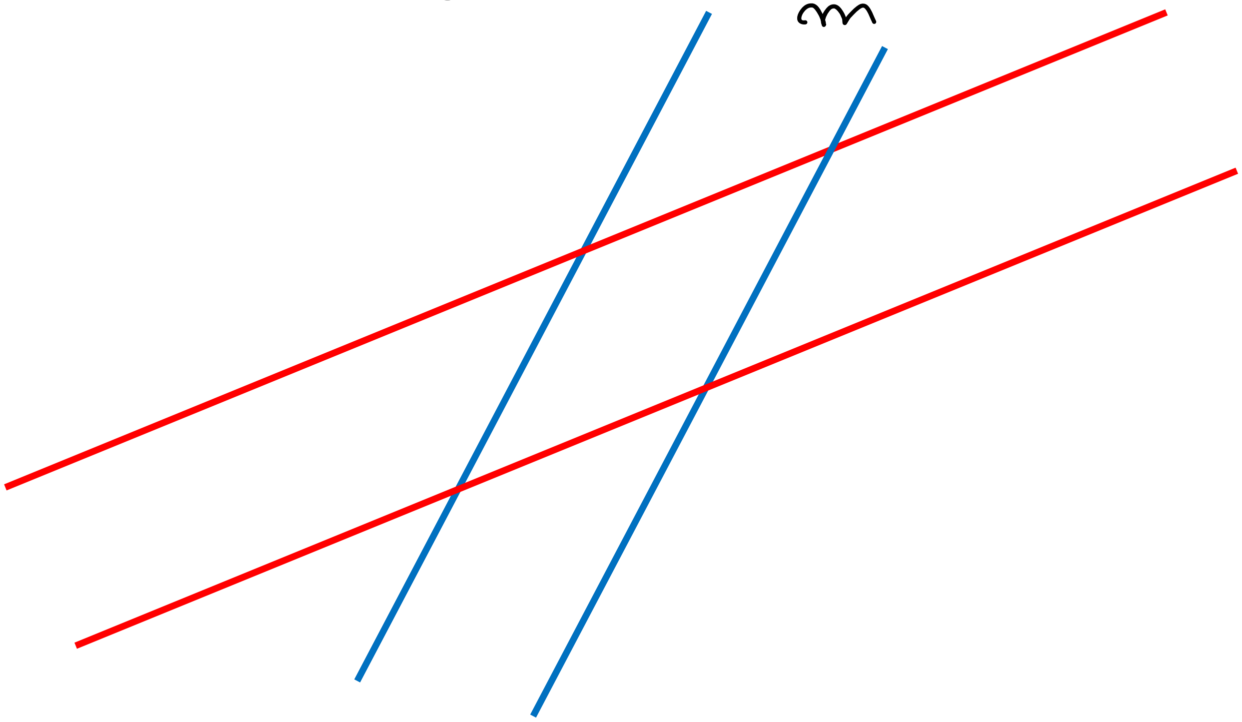
Homogeneous coordinates

- Intersection of parallel lines

이미지 도메인에서 ∞ 위치에 있는 점 표현시

Cartesian: (Inf, Inf) $\frac{1}{0} \to \infty$
Homogeneous: (1, 1, 0)

Cartesian: (Inf, Inf)
Homogeneous: (1, 2, 0)



Perspective Projection Matrix

- Projection is a matrix multiplication using homogeneous coordinates

$PX = x$

가 정의되는 원점 ⟷ Pinhole 원점

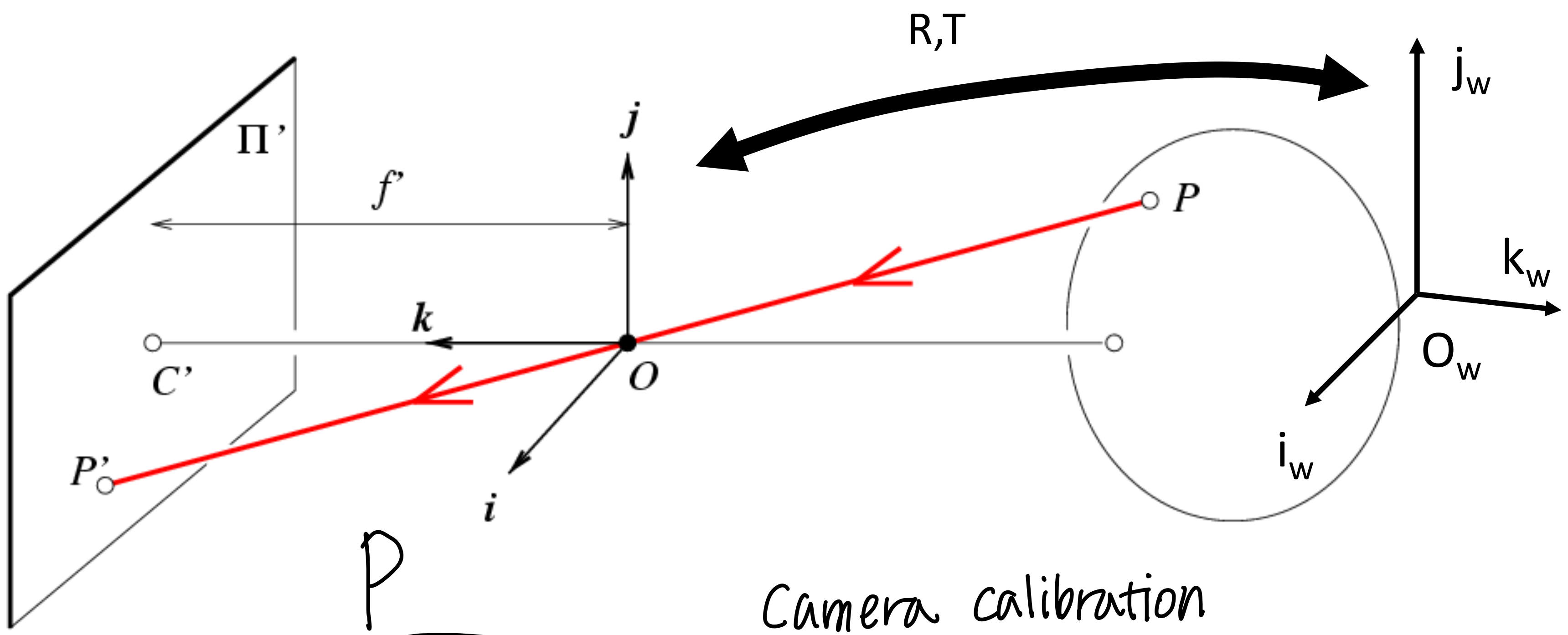
$$\begin{bmatrix} & & & \\ & & & \\ & & & \end{bmatrix}\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} \\ \\ \\ \end{bmatrix} \Rightarrow \left(-f\frac{x}{z}, -f\frac{y}{z}\right)$$

divide by the third coordinate

- In practice: lots of ***coordinate*** transformations…

$$\begin{pmatrix} \text{2D} \\ \text{point} \\ \text{(3x1)} \end{pmatrix} = \begin{pmatrix} \text{Camera to} \\ \text{pixel coord.} \\ \text{trans. matrix} \\ \text{(3x3)} \end{pmatrix} \begin{pmatrix} \text{Perspective} \\ \text{projection matrix} \\ \text{(3x4)} \end{pmatrix} \begin{pmatrix} \text{World to} \\ \text{camera coord.} \\ \text{trans. matrix} \\ \text{(4x4)} \end{pmatrix} \begin{pmatrix} \text{3D} \\ \text{point} \\ \text{(4x1)} \end{pmatrix}$$

Projection matrix



$$\mathbf{x} = \overbrace{\mathbf{K}[\mathbf{R} \quad \mathbf{t}]}^{P}\mathbf{X}$$

카메라의
하드웨어 스펙
(u_0, v_0, f)

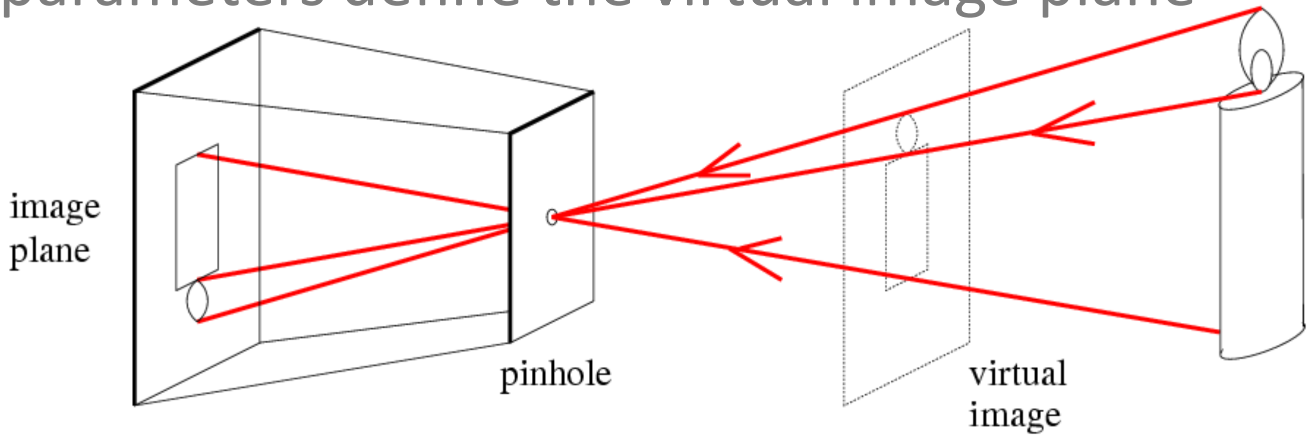
카메라 핀홀과
실제 X가 기준으로 하는 원점의 차이

Camera calibration

x: Image Coordinates: (u,v,1)
K: Intrinsic Matrix (3x3)
R: Rotation (3x3)
t: Translation (3x1)
Extrinsic Matrix
X: World Coordinates: (X,Y,Z,1)

Camera Intrinsic Matrix (K matrix)

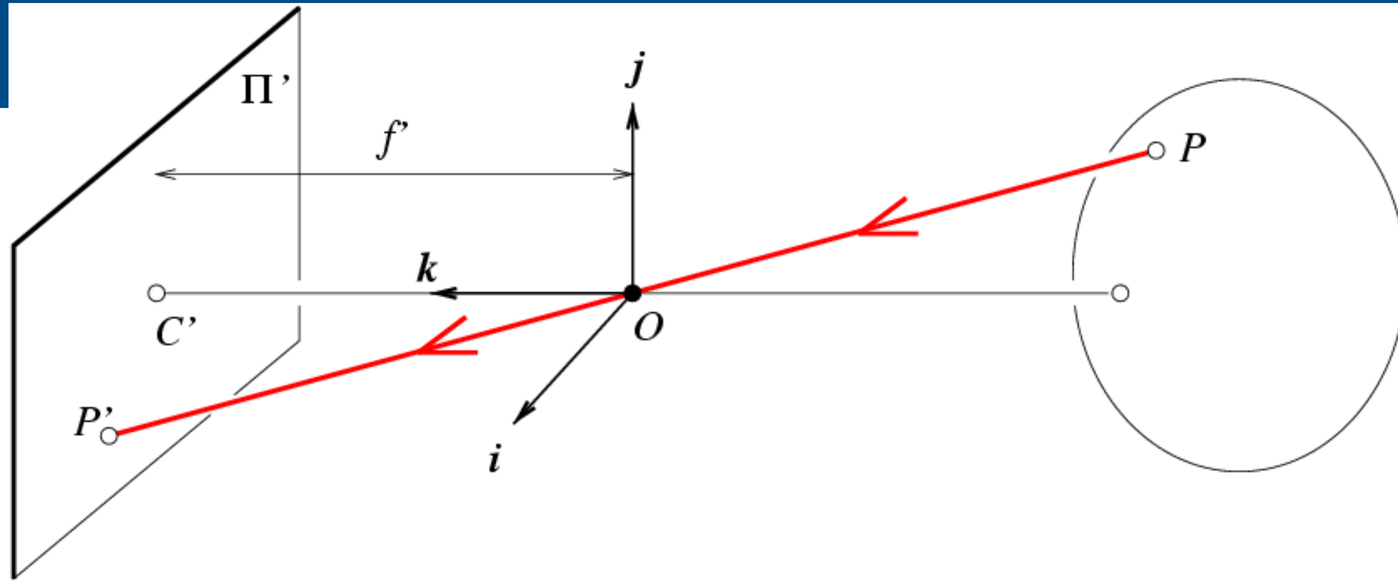
- Intrinsic parameters define the virtual image plane



  - Focal length = f
  - Image center = (u_0, v_0)
  - $K = \begin{bmatrix} -f & 0 & u_0 \\ 0 & -f & v_0 \\ 0 & 0 & 1 \end{bmatrix}$, for simplilcity $K = \begin{bmatrix} f & 0 & u_0 \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{bmatrix}$

Image center

Projection matrix



Intrinsic Assumptions

- Focal length = f
- Image center = (u_0, v_0)
- Unit aspect ratio
- No skew

Extrinsic Assumptions

- No rotation
- Camera at (0,0,0)

$$\mathbf{x} = \mathbf{K}\begin{bmatrix}\mathbf{I} & \mathbf{0}\end{bmatrix}\mathbf{X} \Rightarrow w\begin{bmatrix}u\\v\\1\end{bmatrix} = \begin{bmatrix}\boxed{\begin{matrix}f & 0 & u_0\\0 & f & v_0\\0 & 0 & 1\end{matrix}} & \begin{matrix}0\\0\\0\end{matrix}\end{bmatrix}\begin{bmatrix}x\\y\\z\\1\end{bmatrix}$$

K

Projection matrix

- Remove assumption: square pixels

Intrinsic Assumptions
- Focal length = f
- Image center = (u_0, v_0)
- Aspect ratio: α, β
- No Skew

Extrinsic Assumptions
- No rotation
- Camera at (0,0,0)

$$\mathbf{x} = \mathbf{K}\begin{bmatrix}\mathbf{I} & \mathbf{0}\end{bmatrix}\mathbf{X} \Rightarrow w\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha f & 0 & u_0 & 0 \\ 0 & \beta f & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

K

Projection matrix

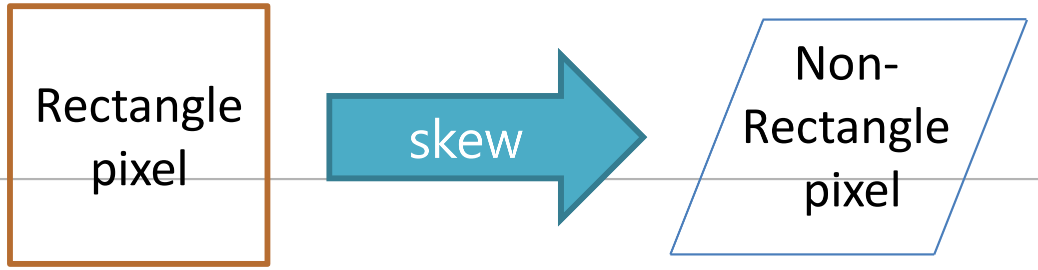
- Remove assumption: non-skewed pixels

Intrinsic Assumptions
- Focal length = f
- Image center = (u_0, v_0)
- Aspect ratio: α/β
- Skew = s

Extrinsic Assumptions
- No rotation
- Camera at (0,0,0)

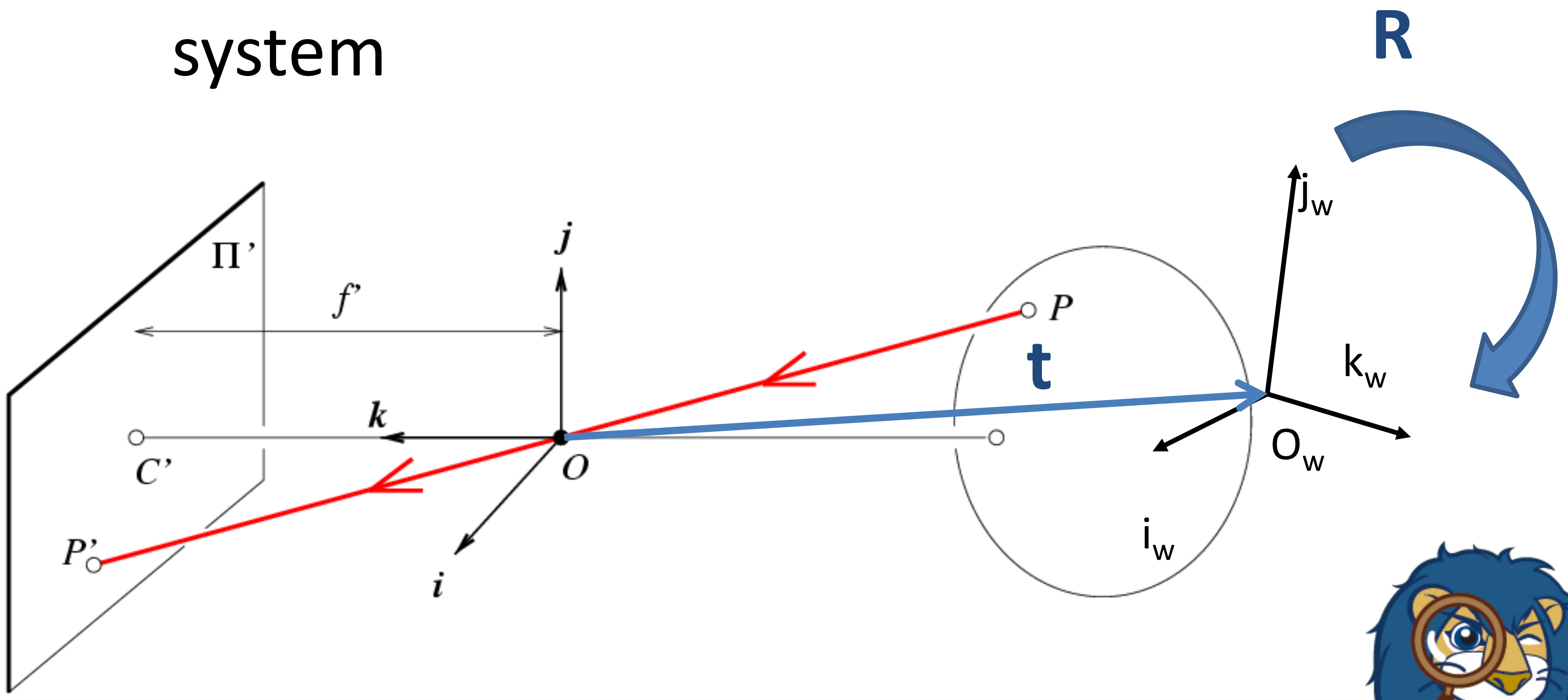
$$\mathbf{x} = \mathbf{K}[\mathbf{I} \quad \mathbf{0}]\mathbf{X} \Rightarrow w\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha f & s & u_0 & 0 \\ 0 & \beta f & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$\boldsymbol{K}$



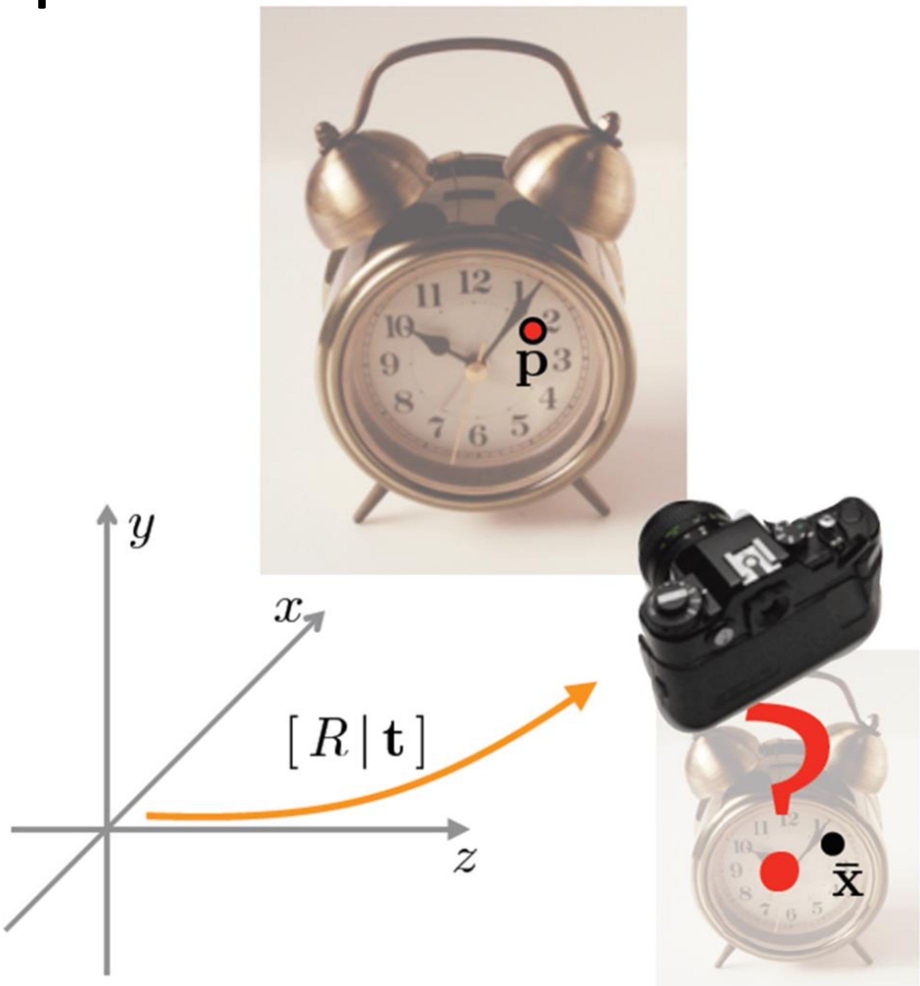
Projection matrix

- Oriented and Translated Camera
  - camera's pose in the reference coordinate system



Projection matrix

- Oriented and Translated Camera
  - camera's pose in the reference coordinate system



Projection matrix

- Allow camera translation

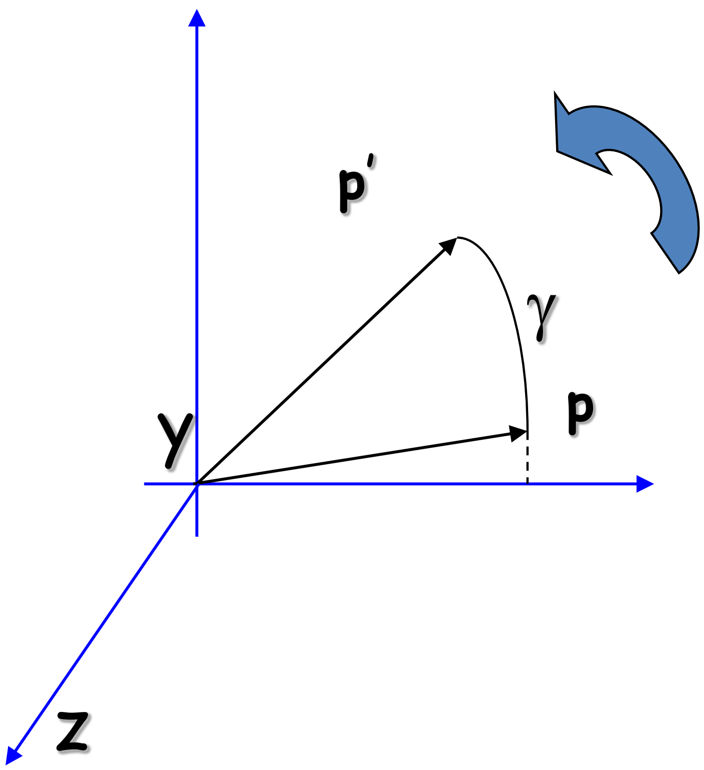
Intrinsic Assumptions
- Focal length = f
- Image center = (u_0, v_0)
- Aspect ratio: α, β
- Skew = s

Extrinsic Assumptions
- No rotation
- Camera at (t_x, t_y, t_z)

$$\mathbf{x} = \mathbf{K}\begin{bmatrix}\mathbf{I} & \mathrm{t}\end{bmatrix}\mathbf{X} \quad \Rightarrow \quad w\begin{bmatrix}u\\v\\1\end{bmatrix} = K\begin{bmatrix}\boldsymbol{I} & \begin{matrix}t_X\\t_y\\t_Z\end{matrix}\end{bmatrix}\begin{bmatrix}x\\y\\z\\1\end{bmatrix}$$

Projection matrix

- Rotation around the coordinate axes, counter-clockwise:



$$R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix}$$

$$R_y(\beta) = \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix}$$

$$R_z(\gamma) = \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Projection matrix

- Full model: allow camera rotation

$$\mathbf{x} = \mathbf{K}\begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix}\mathbf{X}$$

$$w\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha f & s & u_0 \\ 0 & \beta f & v_0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix}\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Projection matrix

- Degree of freedom

$$\mathbf{x} = \mathbf{K}\begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix}\mathbf{X}$$

$$w\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} f_x & s & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix}}_{5}\underbrace{\begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix}}_{6}\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Projection matrix

- Degree of freedom

$$\mathbf{x} = \mathbf{K}\begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix}\mathbf{X}$$

$$w\begin{bmatrix}u\\v\\1\end{bmatrix} = \begin{bmatrix}f & s & u_0\\0 & f & v_0\\0 & 0 & 1\end{bmatrix}\begin{bmatrix}r_{11} & r_{12} & r_{13} & t_x\\r_{21} & r_{22} & r_{23} & t_y\\r_{31} & r_{32} & r_{33} & t_z\end{bmatrix}\begin{bmatrix}x\\y\\z\\1\end{bmatrix}$$

4

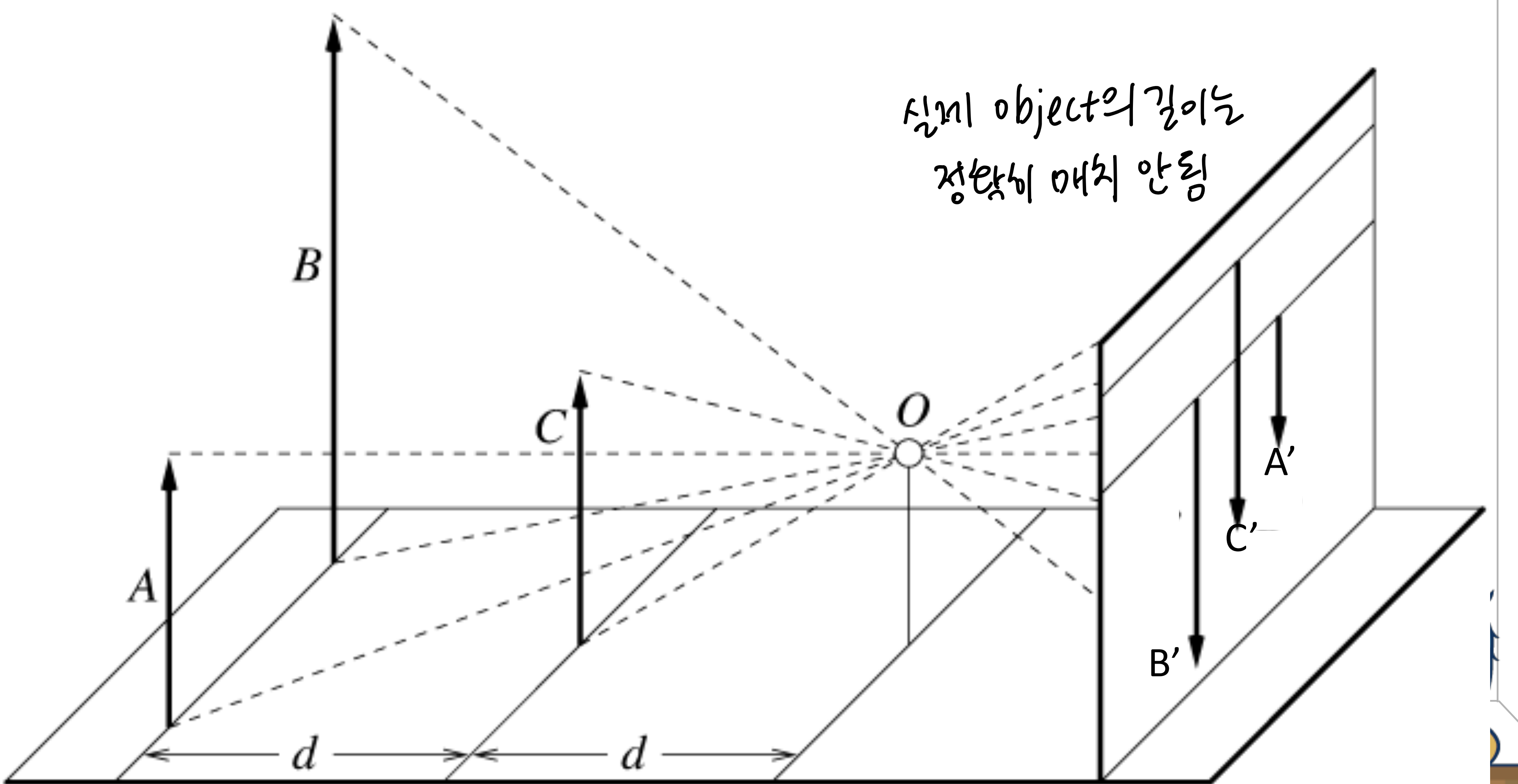
6

Projective Geometry

- What is lost
  - Length

길이의 우열은 보존X

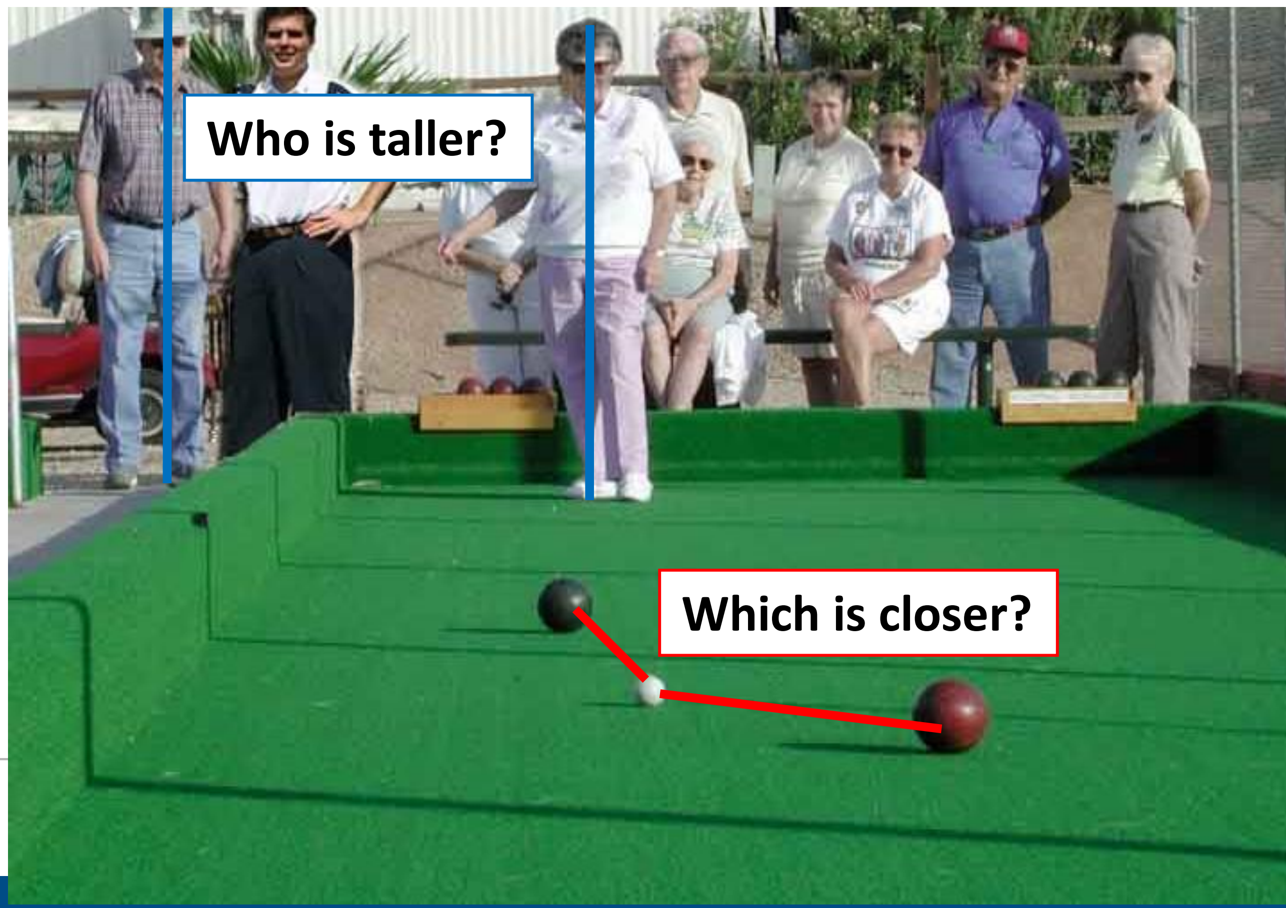
실제 object의 길이는
정확히 매치 안됨



Projective Geometry

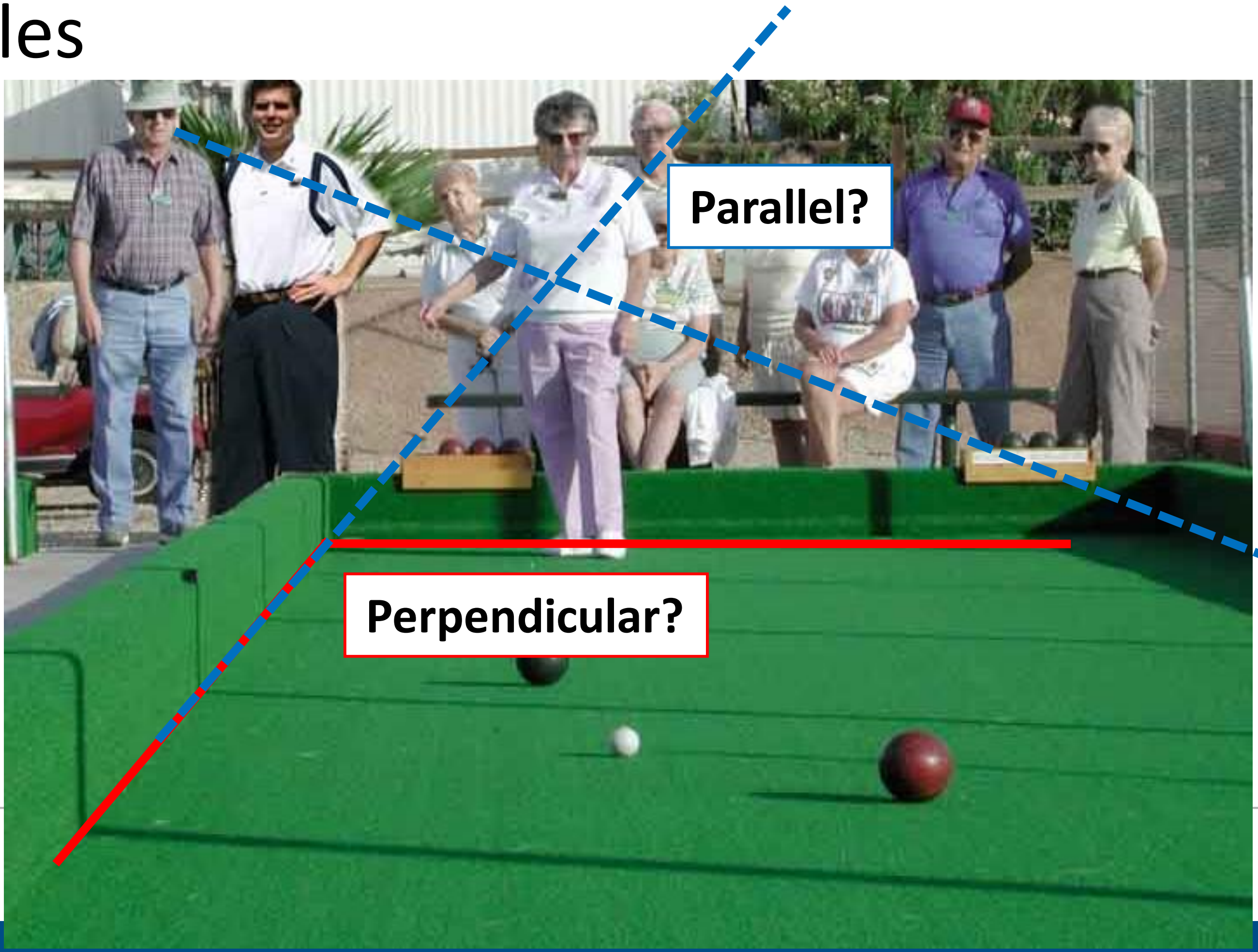
- What is lost
  - Length

가까운게 크게보임



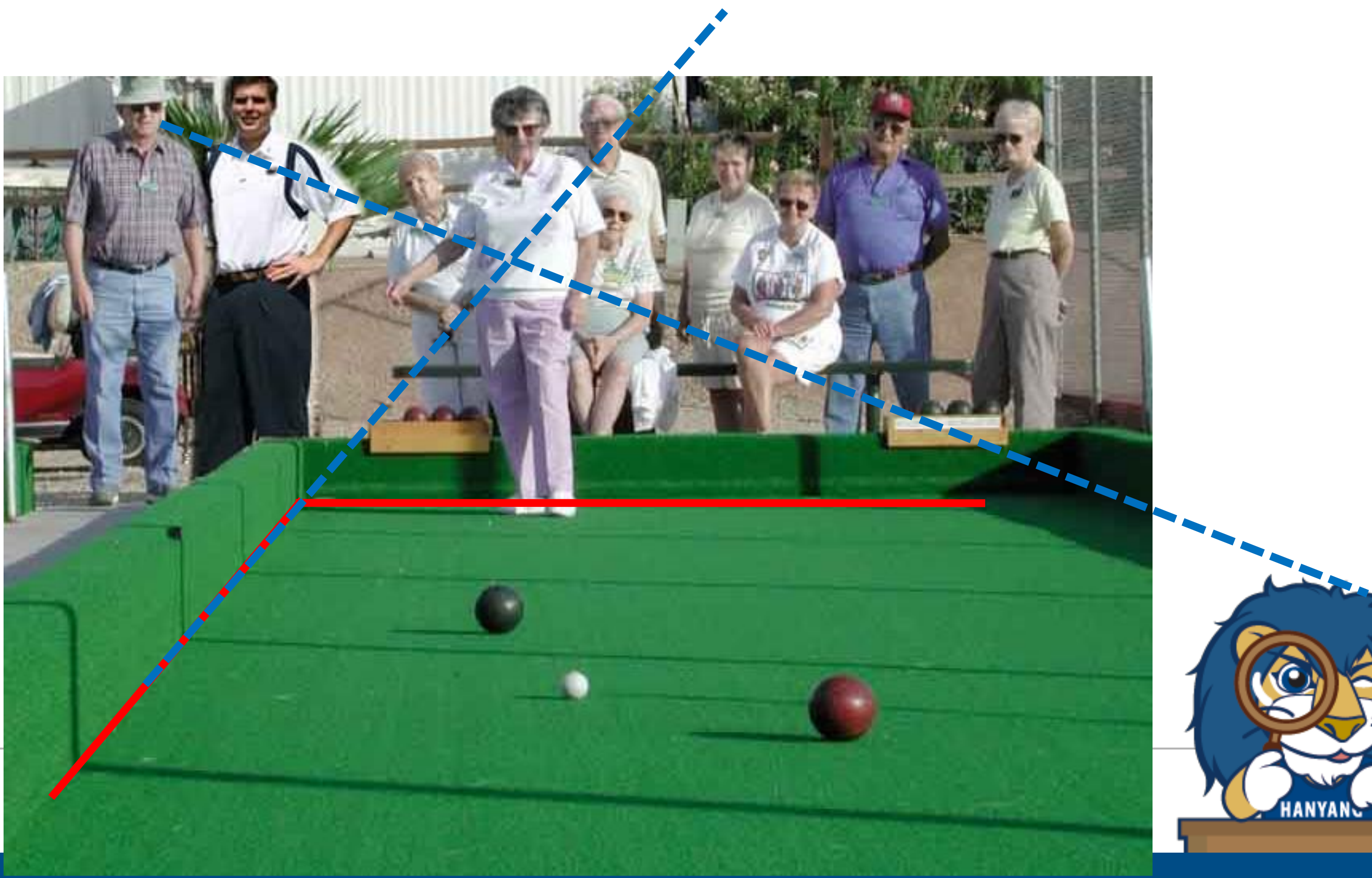
Projective Geometry

- What is lost
  - Length
  - Angles



Projective Geometry

- What is preserved
  – Straight lines are still straight



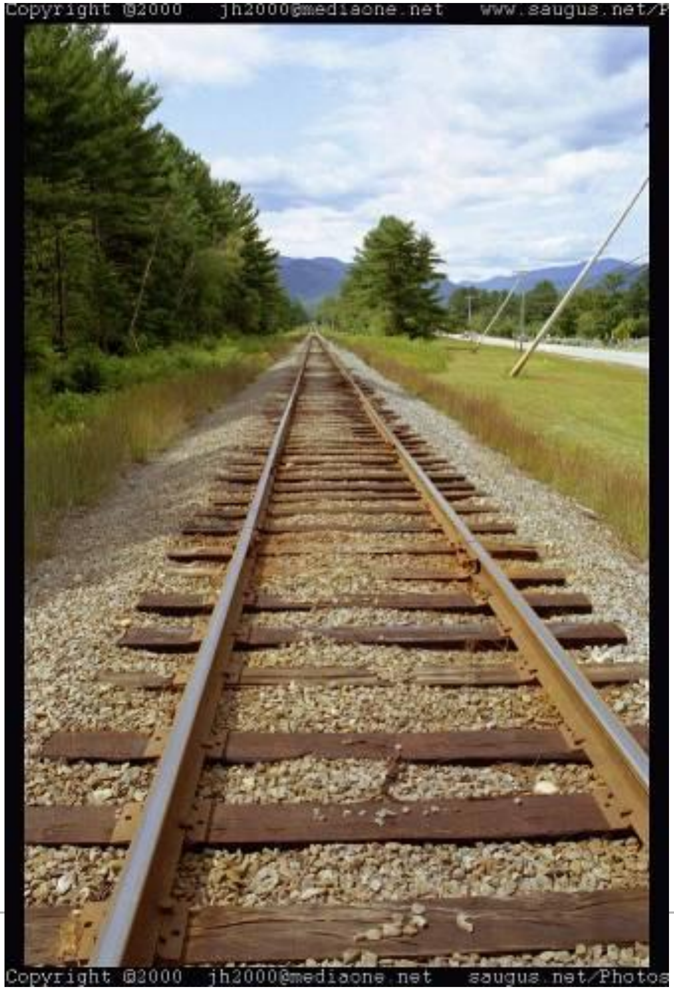
Vanishing points and lines

- Parallel lines in the world intersect in the image at a "vanishing point"

소실점

↓

3D 위치상 ∞에 있는 점



Vanishing points

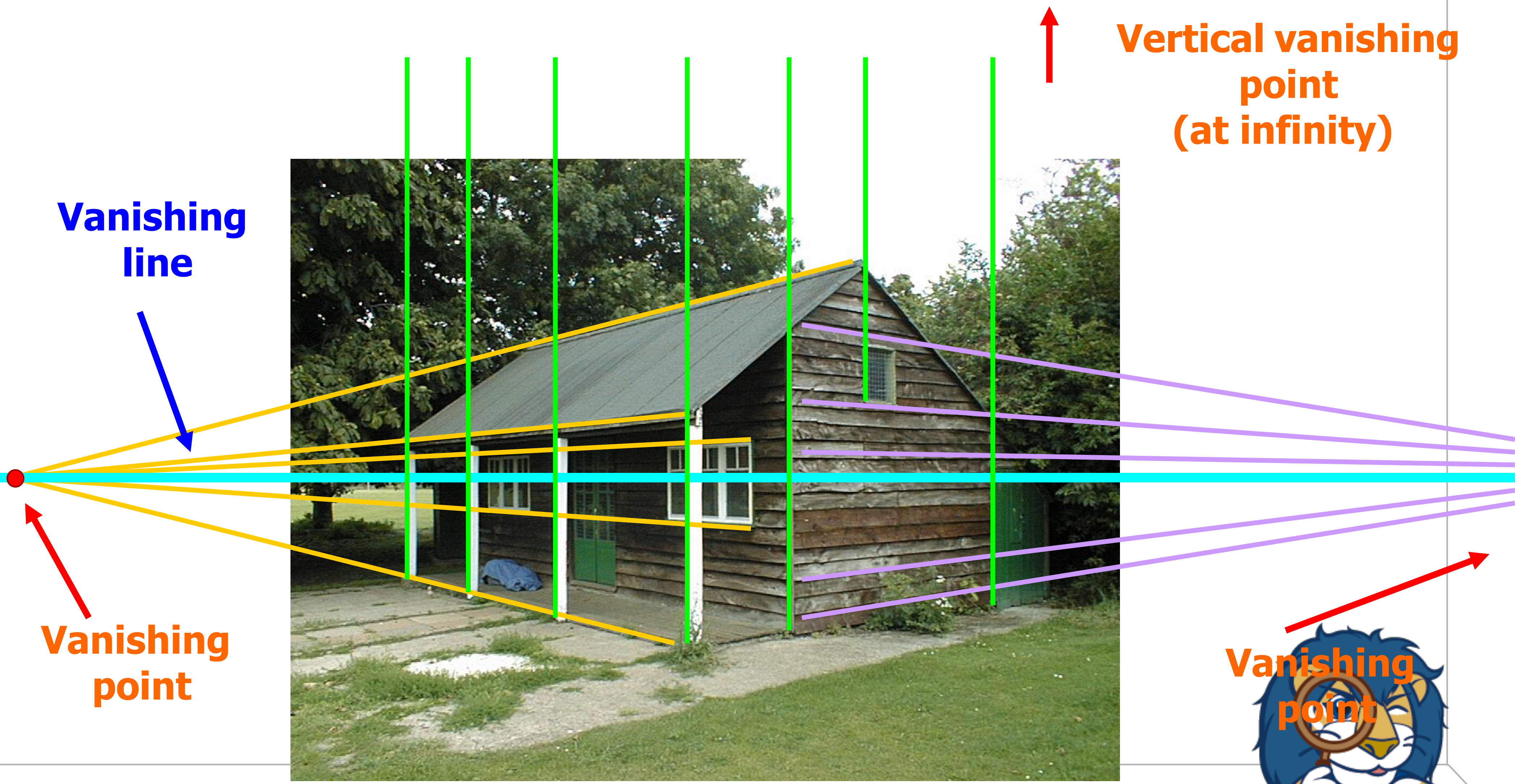
- Vanishing Point = Projection from Infinity

$$\mathbf{p} = \mathbf{K}\begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix}\begin{bmatrix} x \\ y \\ z \\ 0 \end{bmatrix} \Rightarrow \mathbf{p} = \mathbf{KR}\begin{bmatrix} x \\ y \\ z \end{bmatrix} \Rightarrow \mathbf{p} = \mathbf{K}\begin{bmatrix} x_R \\ y_R \\ z_R \end{bmatrix}$$

3D상 ∞ 위치에 있음

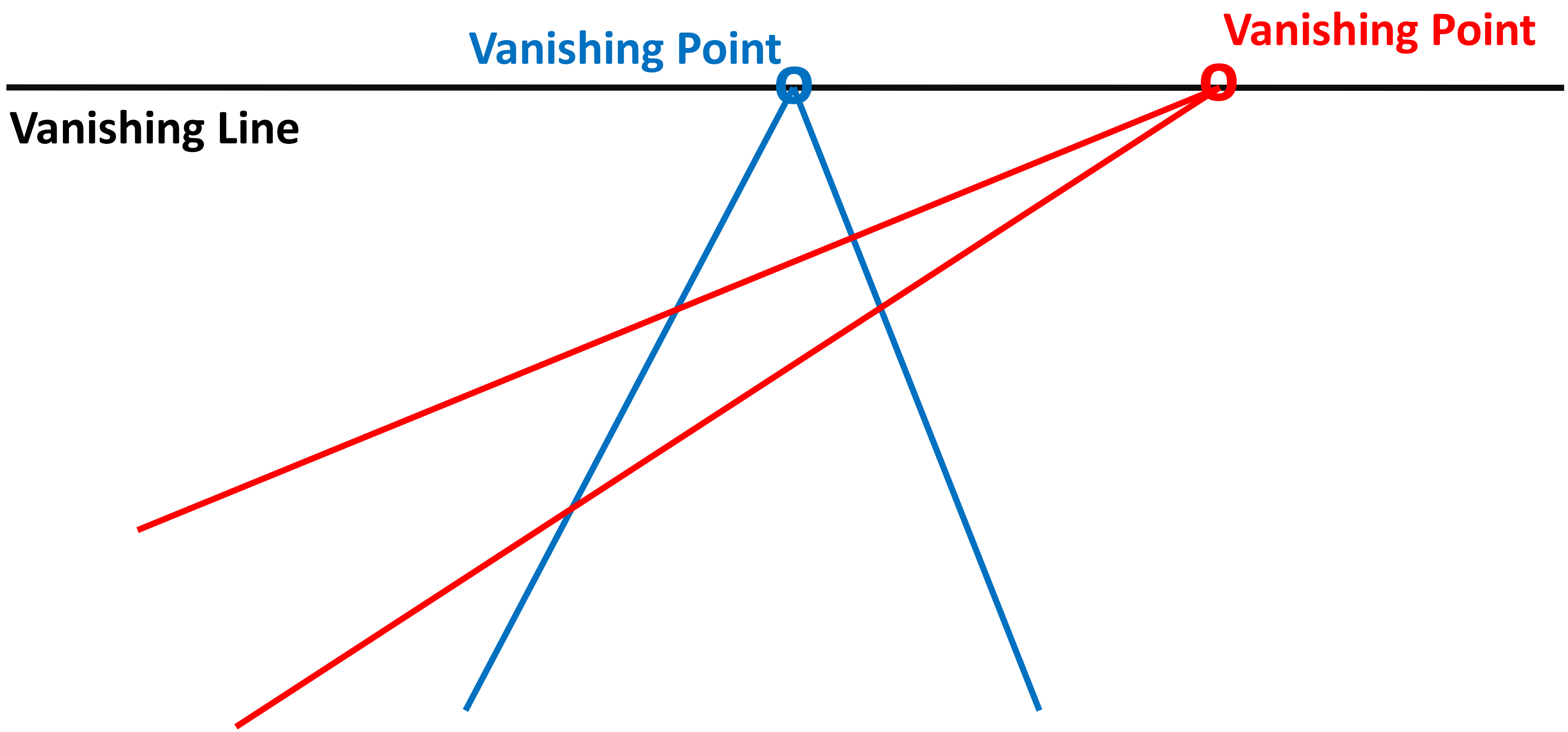
$$w\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & u_0 \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} x_R \\ y_R \\ z_R \end{bmatrix} \Rightarrow \begin{array}{l} u = \dfrac{f x_R}{z_R} + u_0 \\ \\ v = \dfrac{f y_R}{z_R} + v_0 \end{array}$$

Vanishing points and lines



Slide from Efros, Photo from Criminisi

Vanishing points and lines



- The **projections** of parallel 3D lines intersect at a **vanishing point**
- The **projection** of parallel 3D planes intersect at a **vanishing line**
- If a set of parallel 3D lines are also parallel to a particular plane, their vanishing point will lie on the vanishing line of the plane
- Not all lines that intersect are parallel

Calibration by orthogonal vanishing points

- Intrinsic camera matrix
  - Use orthogonality as a constraint
  - Model **K** with only f, u_0, v_0

2 orthogonal vanishing points

$$\mathbf{p}_i = \underset{3\times3}{\mathbf{K}}\mathbf{R}\mathbf{X}_i \qquad \mathbf{X}_i^T \mathbf{X}_j = 0$$

$$\mathbf{X_i} = \mathbf{R}^{-1}\mathbf{K}^{-1}\mathbf{p_i}$$

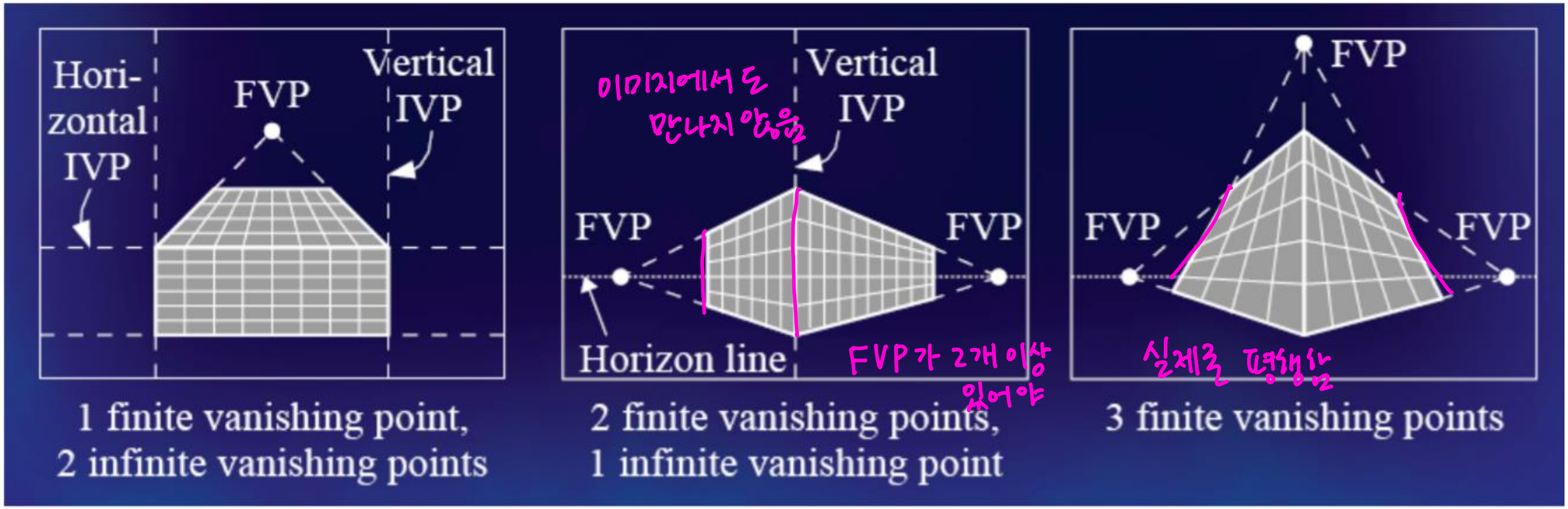
$$\mathbf{p_i}^{\mathrm{T}}\left(\mathbf{K}^{-1}\right)^{\mathrm{T}}(\mathbf{R})(\mathbf{R}^{-1})(\mathbf{K}^{-1})\mathbf{p}_j = \mathbf{0}$$

$$\Rightarrow \mathbf{p_i}^{\mathrm{T}}\left(\mathbf{K}^{-1}\right)^{\mathrm{T}}(\mathbf{K}^{-1})\mathbf{p}_j = \mathbf{0}$$

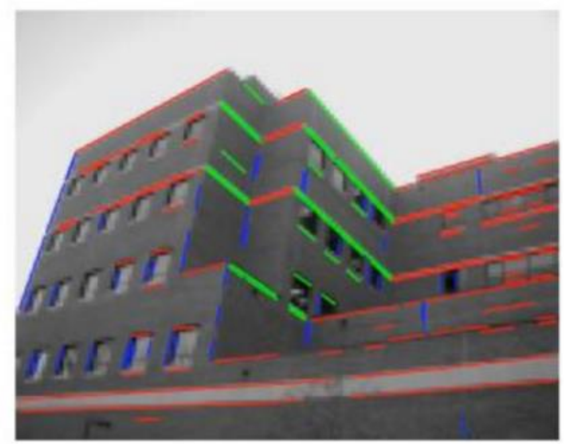
이미지에서
vanishing point의 위치

- What if you don't have three finite vanishing points?
  - Two finite VP: solve f, get valid u_0, v_0 closest to image center
  - One finite VP: u_0, v_0 is at vanishing point;
    - can't solve for f

Intrinsic calibration from vanishing points



Cannot recover focal length, image center is the finite vanishing point



Can solve for focal length, image center

Rotation from vanishing points

- Rotation matrix R
  - Set directions of vanishing points
    - e.g., X_1 = [1, 0, 0, 0], X_2 = [0, 1, 0, 0], X_2 = [0, 0, 1, 0]
  - Each VP provides one column of **R**: $\mathbf{p_i} = \mathbf{K}\mathbf{r_i}$
  - Special properties (constraints) of **R**
    - inv(**R**)=**R**^{T}
    - Each row and column of **R** has unit length
      - $R = [r_1\ r_2\ r_3]: where\ \|r_i\|^2 = 1$

t는 못찾음

FVP 3개 → K → r 찾음

Calibration from vanishing points: Summary

- Solve for K using three orthogonal vanishing points
  - Focal length, principal point (f, u_0, v_0)
- Get rotation directly from vanishing points once K is known

- Advantages
  - No need for calibration chart (2D-3D correspondences)
  - Could be completely automatic
- Disadvantages
  - Only applies to certain kinds of scenes
  - Inaccuracies in computation of vanishing points (supplement class)
  - Problems due to infinite vanishing points

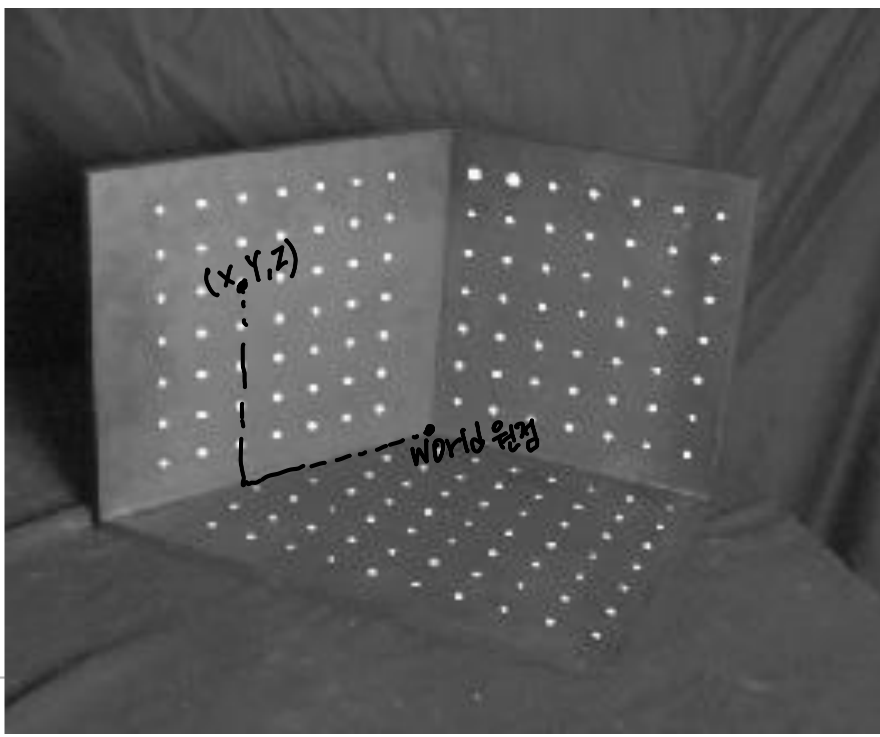
How to calibrate the camera?

$$\mathbf{x} = \overbrace{\underbrace{\mathbf{K}}_{\text{하드웨어 정보}} \left[\underbrace{\mathbf{R} \quad \mathbf{t}}_{\text{카메라의 좌표계}} \right]}^{3\times 4 \text{ matrix}} \mathbf{X}$$

$$w \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Camera Calibration #2

- Use an object (calibration grid) with known geometry
  - Corresponding image points to 3d points
  - Get least squares solution (or non-linear solution)



Known 2d image coordinates

Known 3d points

$P = K[R\ t]$

$$\begin{bmatrix} wu \\ wv \\ w \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Unknown Camera Parameters

Camera Calibration #2

- one corresponding point → two equations

Unknown Camera Parameters

Known 2d image points

$$\begin{bmatrix} su \\ sv \\ s \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Known 3d points

$$(m_{31}X + m_{32}Y + m_{33}Z + m_{34})u = m_{11}X + m_{12}Y + m_{13}Z + m_{14}$$

$$(m_{31}X + m_{32}Y + m_{33}Z + m_{34})v = m_{21}X + m_{22}Y + m_{23}Z + m_{24}$$

$$0 = m_{11}X + m_{12}Y + m_{13}Z + m_{14} - m_{31}uX - m_{32}uY - m_{33}uZ - m_{34}u$$

$$0 = m_{21}X + m_{22}Y + m_{23}Z + m_{24} - m_{31}vX - m_{32}vY - m_{33}vZ - m_{34}v$$

Unknown Camera Parameters

Known 2d image points

$$\begin{bmatrix} su \\ sv \\ s \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Known 3d points

- **Method 1** – nonhomogeneous linear system. Solve for m's entries using linear least squares (**Ax=b** form)

$$\begin{bmatrix} X_1 & Y_1 & Z_1 & 1 & 0 & 0 & 0 & 0 & -u_1X_1 & -u_1Y_1 & -u_1Z_1 \\ 0 & 0 & 0 & 0 & X_1 & Y_1 & Z_1 & 1 & -v_1X_1 & -v_1Y_1 & -v_1Z_1 \\ & & & & & & \vdots & & & & \\ X_n & Y_n & Z_n & 1 & 0 & 0 & 0 & 0 & -u_nX_n & -u_nY_n & -u_nZ_n \\ 0 & 0 & 0 & 0 & X_n & Y_n & Z_n & 1 & -v_nX_n & -v_nY_n & -v_nZ_n \end{bmatrix} \begin{bmatrix} m_{11} \\ m_{12} \\ m_{13} \\ m_{14} \\ m_{21} \\ m_{22} \\ m_{23} \\ m_{24} \\ m_{31} \\ m_{32} \\ m_{33} \end{bmatrix} = \begin{bmatrix} u_1 \\ v_1 \\ \vdots \\ u_n \\ v_n \end{bmatrix}$$

```
M = A\Y;
M = [M;1];
M = reshape(M,[],3)';
```

Unknown Camera Parameters

Known 2d image points

$$\begin{bmatrix} su \\ sv \\ s \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Known 3d points

$$0 = m_{11}X + m_{12}Y + m_{13}Z + m_{14} - m_{31}uX - m_{32}uY - m_{33}uZ - m_{34}u$$

$$0 = m_{21}X + m_{22}Y + m_{23}Z + m_{24} - m_{31}vX - m_{32}vY - m_{33}vZ - m_{34}v$$

- **Method 2** – homogeneous linear system. Solve for m's entries using linear least squ ares (**Ax=0** form)

$$\begin{bmatrix} X_1 & Y_1 & Z_1 & 1 & 0 & 0 & 0 & 0 & -u_1X_1 & -u_1Y_1 & -u_1Z_1 & -u_1 \\ 0 & 0 & 0 & 0 & X_1 & Y_1 & Z_1 & 1 & -v_1X_1 & -v_1Y_1 & -v_1Z_1 & -v_1 \\ & & & & & & \vdots & & & & & \\ X_n & Y_n & Z_n & 1 & 0 & 0 & 0 & 0 & -u_nX_n & -u_nY_n & -u_nZ_n & -u_n \\ 0 & 0 & 0 & 0 & X_n & Y_n & Z_n & 1 & -v_nX_n & -v_nY_n & -v_nZ_n & -v_n \end{bmatrix} \begin{bmatrix} m_{11} \\ m_{12} \\ m_{13} \\ m_{14} \\ m_{21} \\ m_{22} \\ m_{23} \\ m_{24} \\ m_{31} \\ m_{32} \\ m_{33} \\ m_{34} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}$$

```
[U, S, V] = svd(A);
M = V(:,end);
M = reshape(M,[],3)';
```

Calibration with linear method

- Advantages
  - Easy to formulate and solve
  - Provides good initialization for non-linear methods
- Disadvantages
  - Do not give you the camera parameters
  - Can’t impose constraints
    - focal length
    - lens distortion
  - Do not minimize projection error
- Non-linear methods are preferred
  - Define error as difference between projected points and measured points
  - Minimize error using Newton’s method or other non-linear optimizers

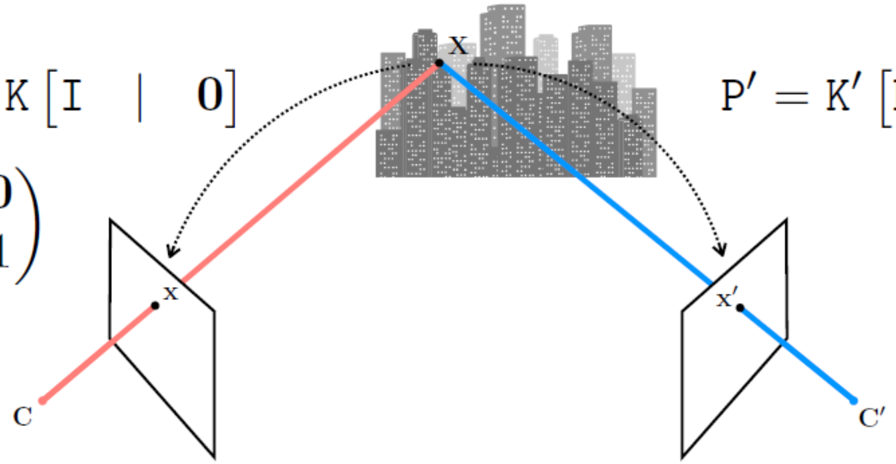
Extrinsic Camera Calibration #3

- The Essential Matrix is the Calibrated Analog to the Fundamental Matrix

$$P = K\,[I \mid \mathbf{0}] \qquad P' = K'\,[R \mid \mathbf{t}]$$

$$\mathbf{C} = \begin{pmatrix} \mathbf{0} \\ 1 \end{pmatrix} \qquad \mathbf{C}' = \begin{pmatrix} -R^T\mathbf{t} \\ 1 \end{pmatrix}$$

$$P^+ = \begin{bmatrix} K^{-1} \\ \mathbf{0}^T \end{bmatrix}$$



– $l' = e' \times x' = [P'C] \times [P'P^+]x = Fx$

- $F = [P'C]_x[P'P^+] = [K'[R|t]C]_x[K'[R|t]\begin{bmatrix} K^{-1} \\ 0 \end{bmatrix}]$
- $= [K't]_x K'RK^{-1} = K'^{-T}\underbrace{[t]_x R}_{E}K^{-1} \rightarrow E = K'^T F K$

$[\mathbf{t}]_\times M = M^{-T}[M^{-1}\mathbf{t}]_\times$

The essential matrix **E** has five degrees of freedom
(3 from rotation, 3 from translation, one less due to homogeneity)

Thank you!