اصول علم ربات – اسلاید دوازدهم

Fundamentals of Robotics – Slide 11

Camera model, calibration, depth estimation

دکتر مهدی جوانمردی

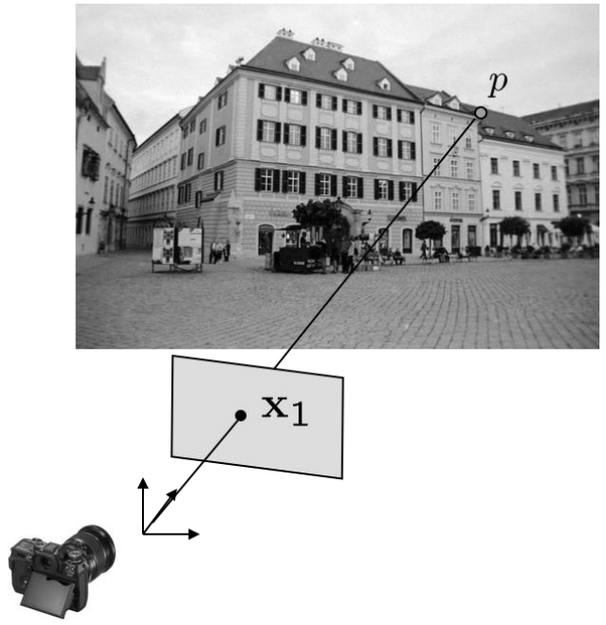
زمستان–بهار ۱۴۰۱

Today's agenda

- Images
- Pinhole cameras
- Cameras & lenses
- The geometry of pinhole cameras
  - Intrinsic
  - Extrinsic

What is an image?

- Projection of the scene on the image plane
- Digital (discrete) image
  - A matrix of integer values



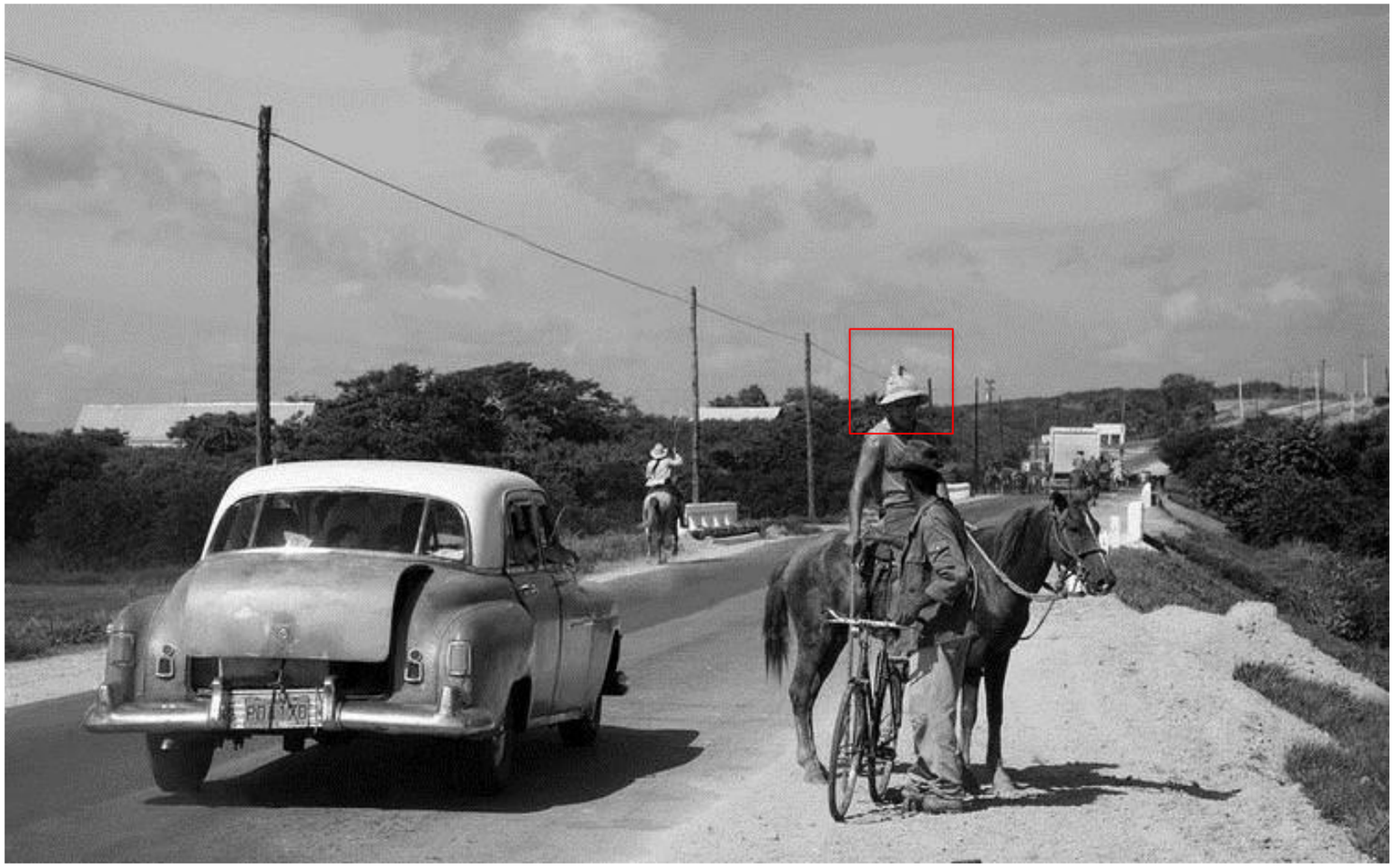
j →

i ↓

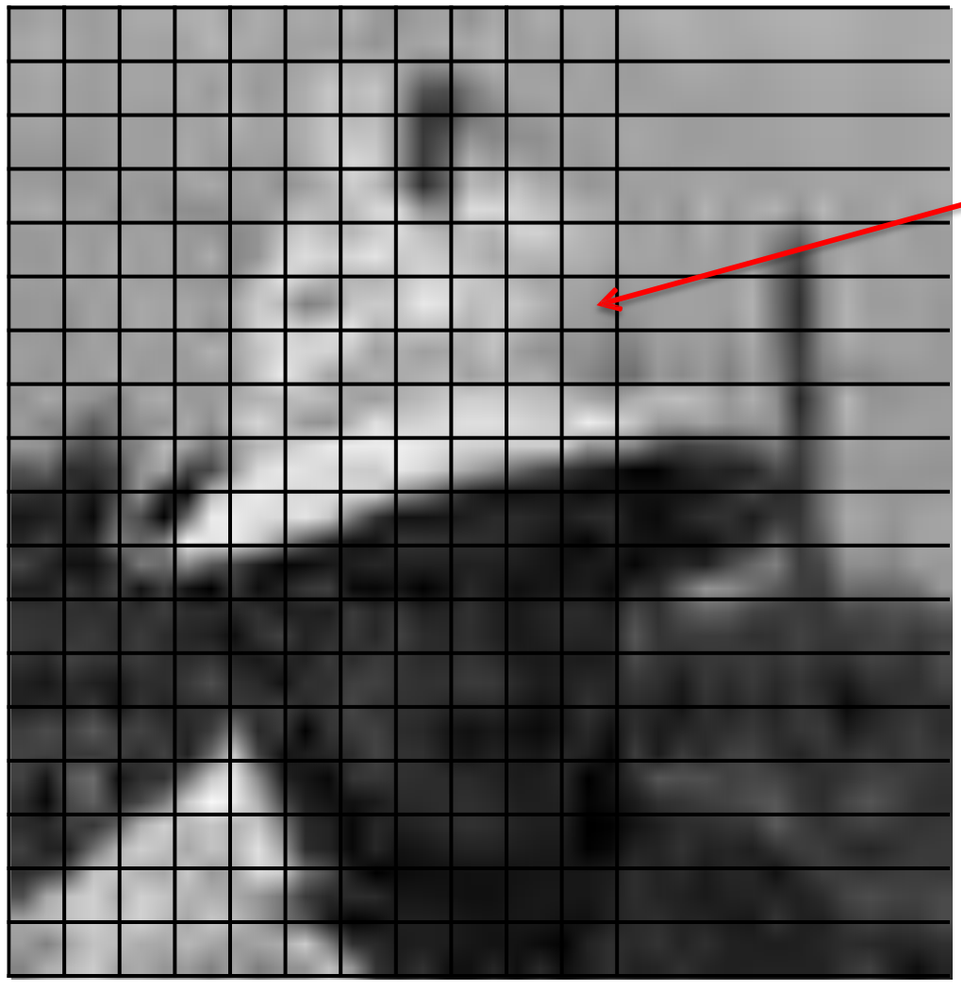
62	79	23	119	120	105	4	0
10	10	9	62	12	78	34	0
10	58	197	46	46	0	0	48
176	135	5	188	191	68	0	49
2	1	1	29	26	37	0	77
0	89	144	147	187	102	62	208
255	252	0	166	123	62	0	31
166	63	127	17	1	0	99	30

$$P = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \rightarrow P' = \begin{bmatrix} x' \\ y' \end{bmatrix} \qquad \mathbb{R}^3 \rightarrow \mathbb{R}^2$$

What is an image?



What is an image?

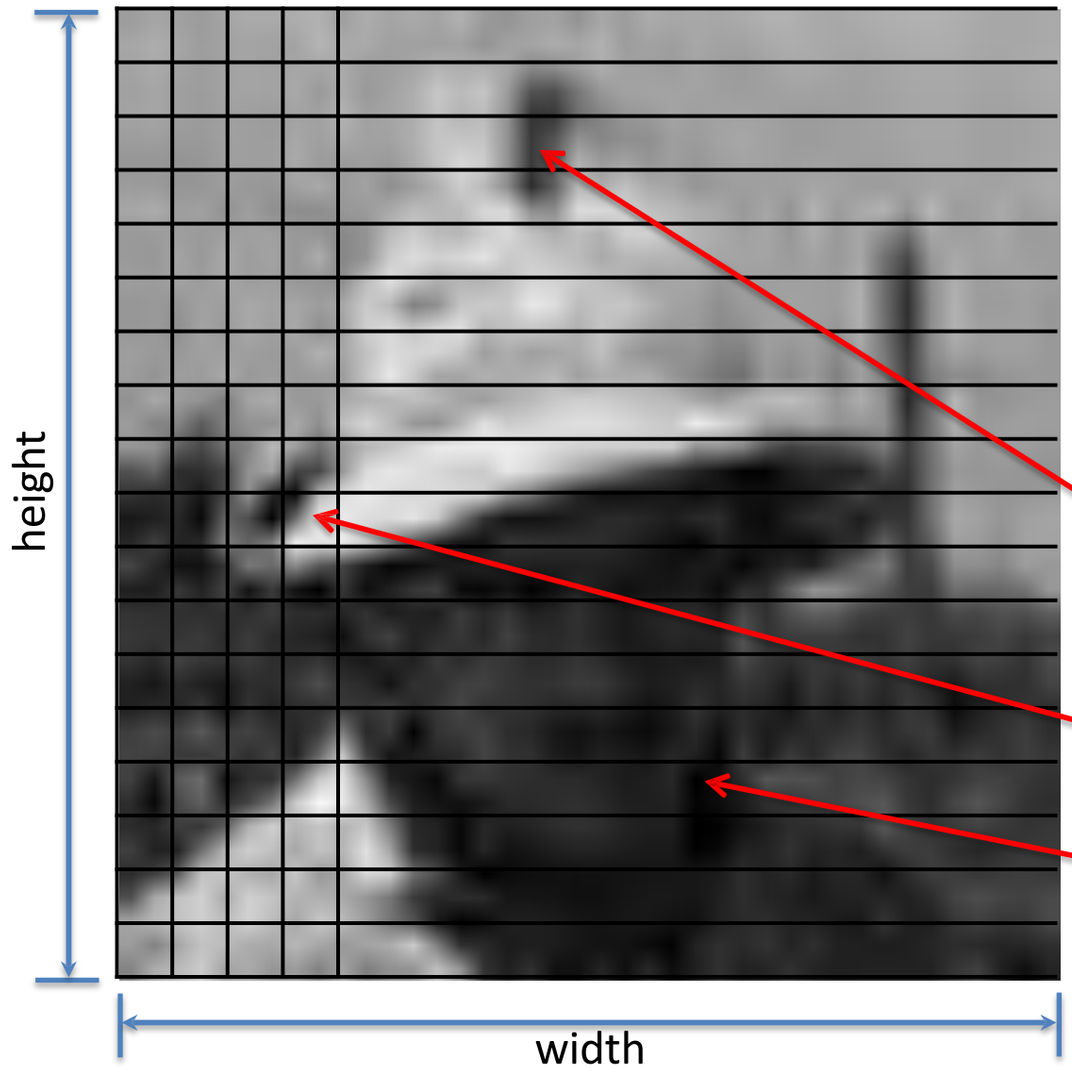


$$P = f(x, y)$$

$$f : R^2 \Rightarrow R$$

Pixel

What is an image?



- Defined over a rectangle
- Intensity values

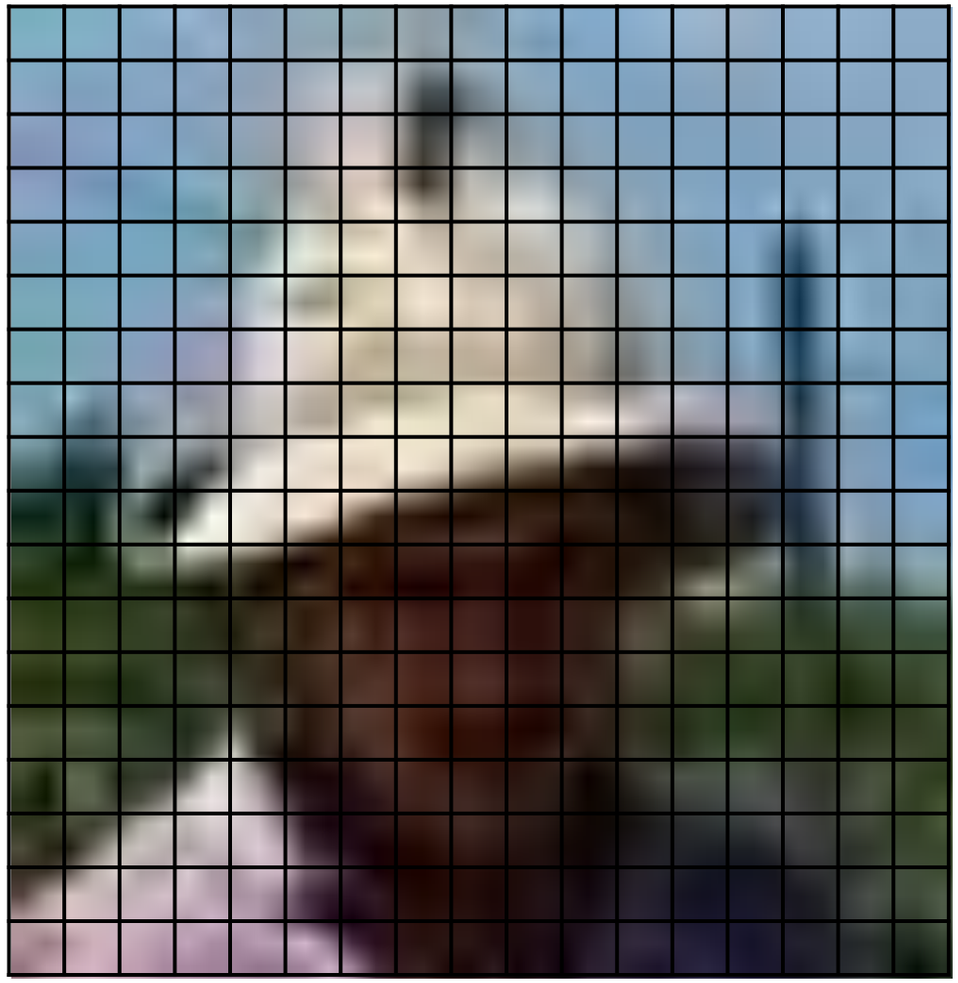
$f(x, y) \in [0, 255]$

$f(x, y) = 97$

$f(x, y) = 213$

$f(x, y) = 0$

What is an image?



A color image: R, G, B channels

$$f(x,y) = \begin{bmatrix} r(x,y) \\ g(x,y) \\ b(x,y) \end{bmatrix}$$

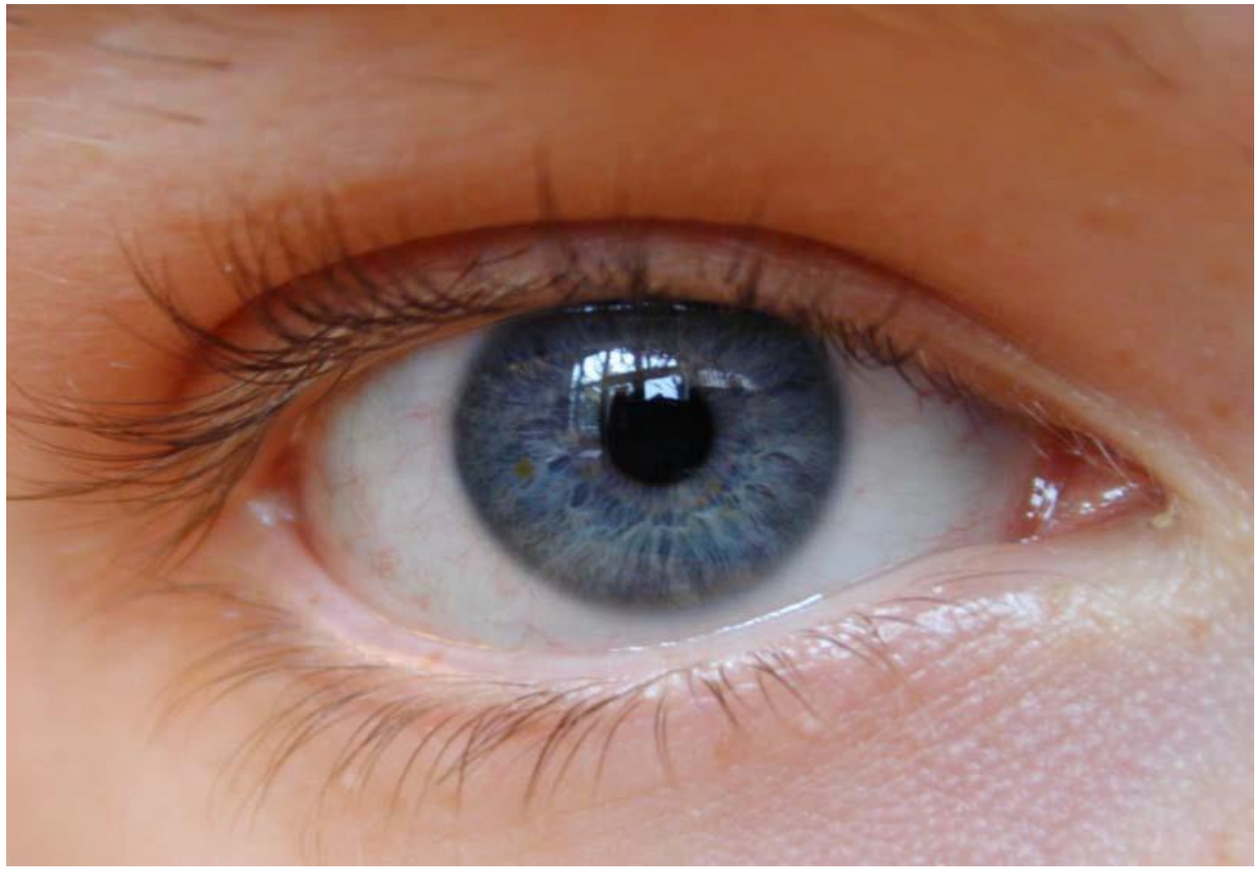
“vector-valued” function

Imaging…

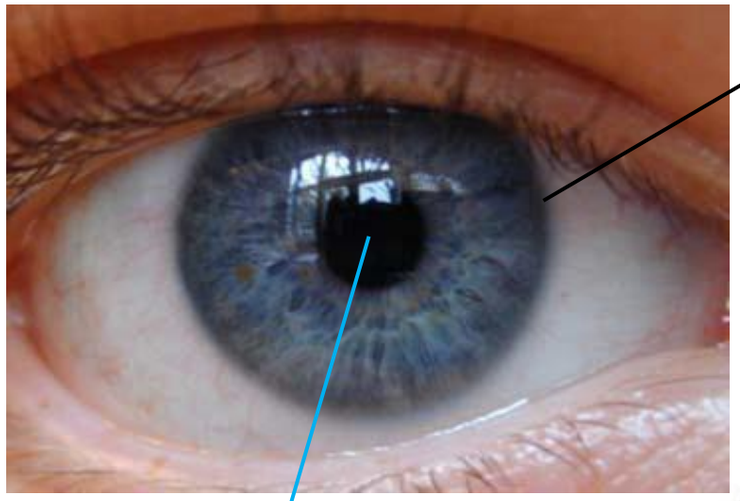
- Images are 2D projections of real-world scenes
- Images capture two kinds of information:
  - Geometric: points, lines, curves, etc.
  - Photometric: intensity, color.
- Complex 3D-2D relationships
  - Camera models approximate relationships.

Through our eyes…

- We see the world



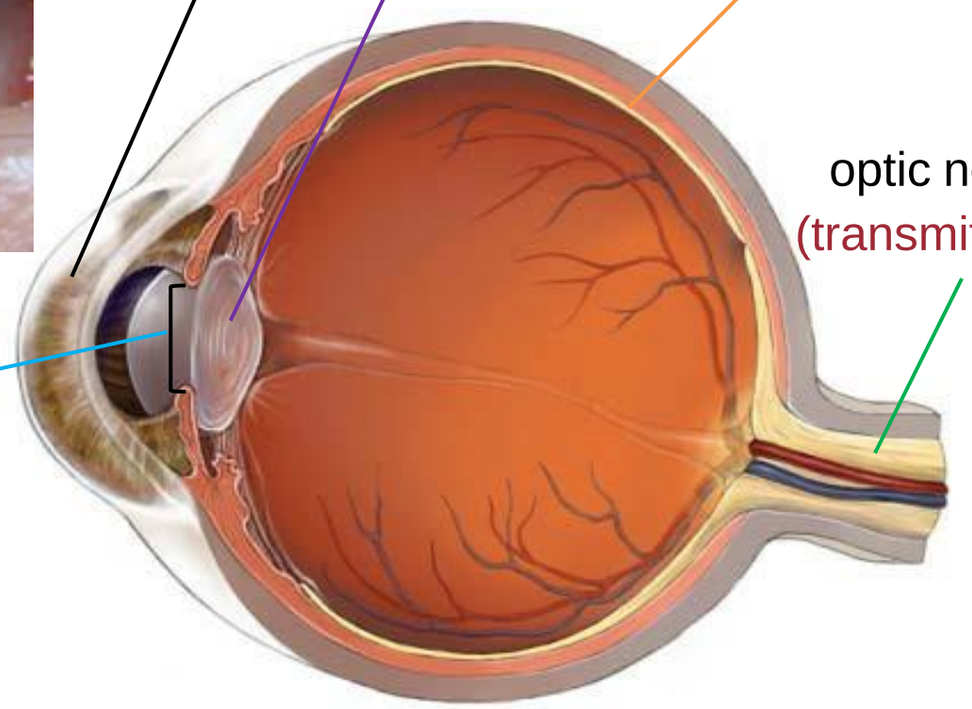
Through our eyes…



cornea (قرنیه)
(focus light)

lens
(لیز)

retina (شبکیه)
(captures image)

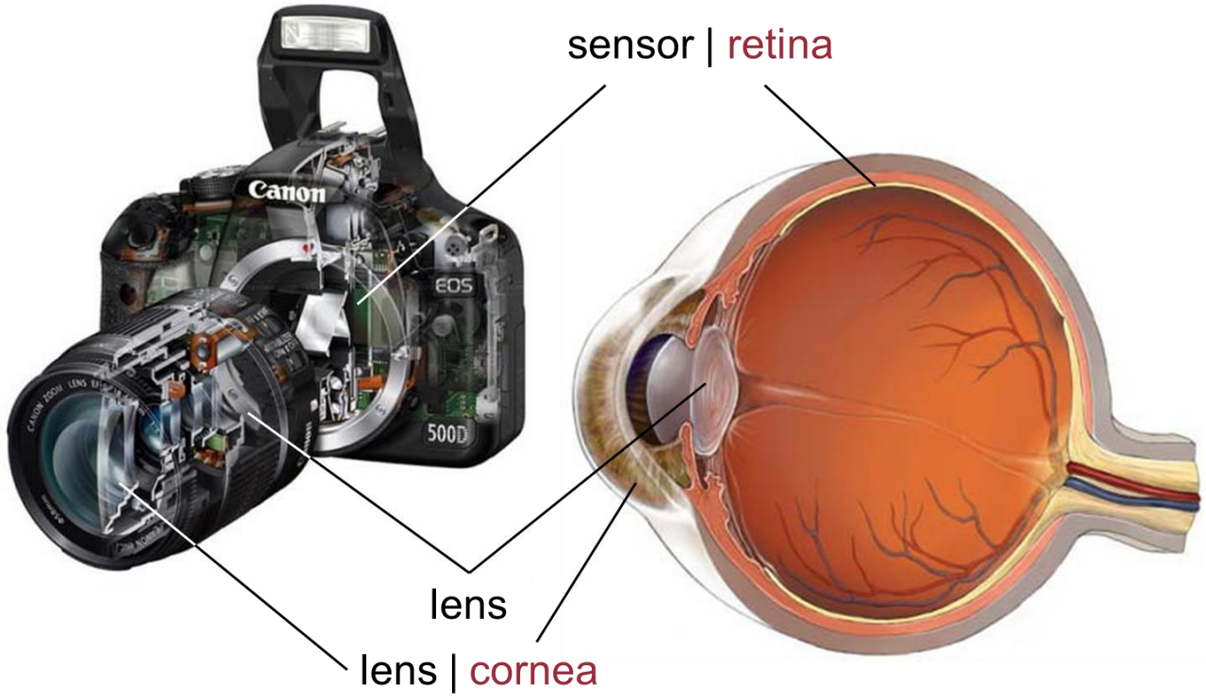


optic nerves (اعصاب بینایی)
(transmit image)

pupil (مردمک)
(control amount of light entering the eye)

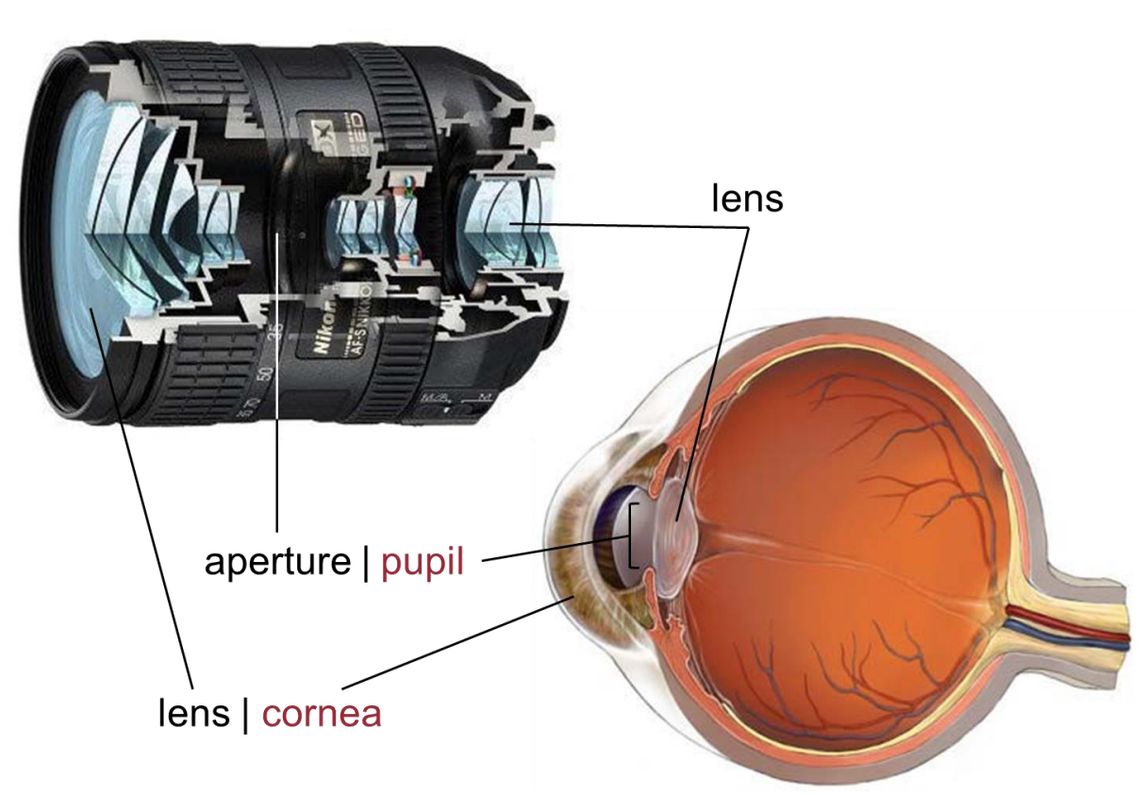
Through their eyes…

- Camera is structurally the same as the eye
  - Lens does similarly to our lens and cornea
  - Sensor receives the light signals to form images



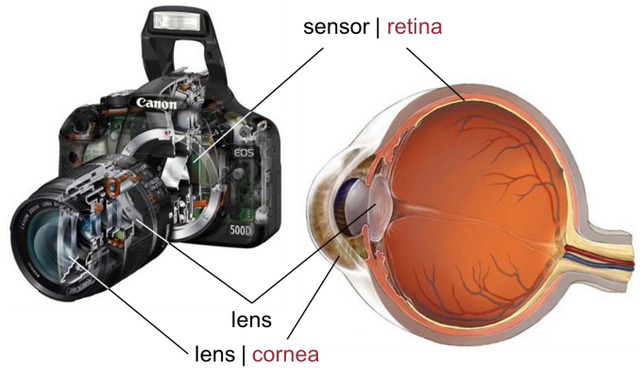
Through their eyes…

- Camera is structurally the same as the eye
  - Aperture controls the amount of light



Camera vs. eye

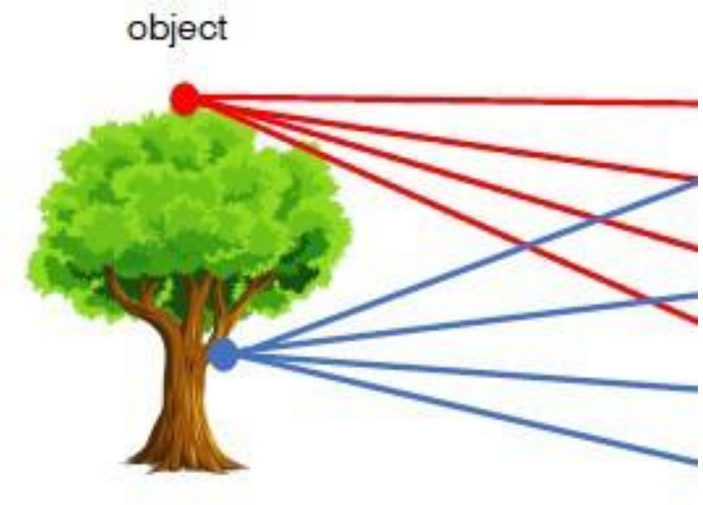
- Similarities
  - Image focusing
  - Light adjustment
- Differences (to name a few)
  - Lens focus
    - Camera: lens moves closer/further from the film
    - Eye: lens changes shape to focus
  - Sensitivity to light
    - Camera: A film is uniformly sensitive to light
    - Eye: retina is not; has greater sensitivity in dark



Camera Models

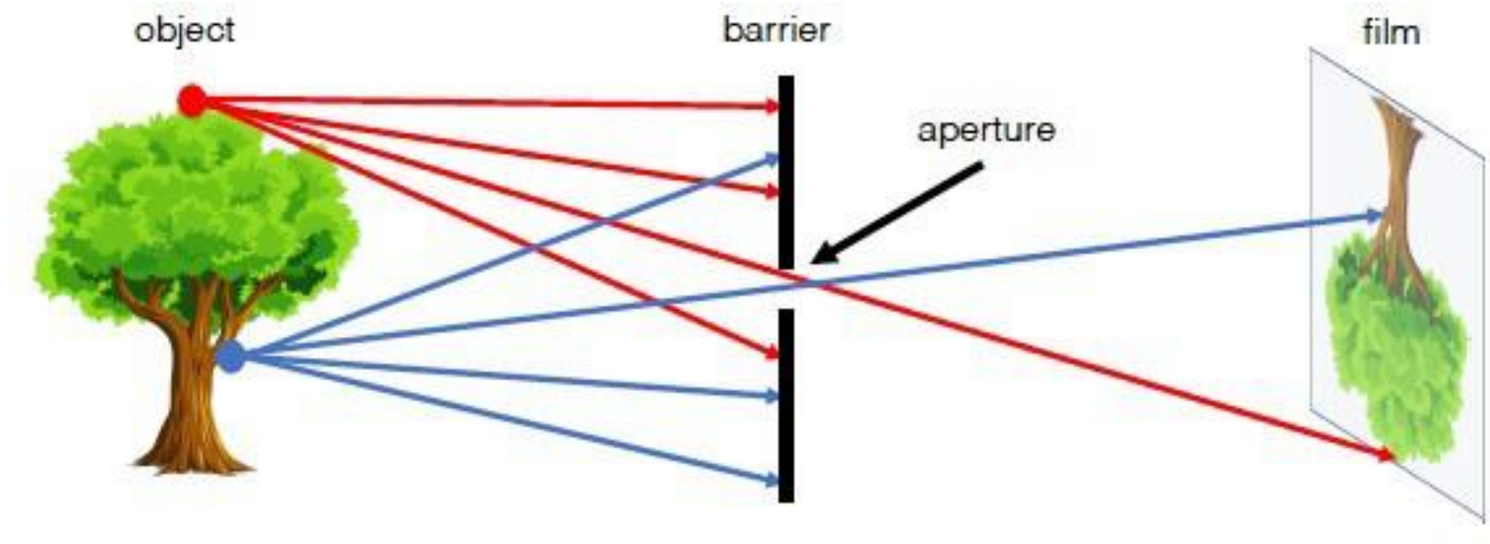
- Images
- Pinhole cameras
- Cameras & lenses
- The geometry of pinhole cameras
  - Intrinsic
  - Extrinsic

How do we see the world?

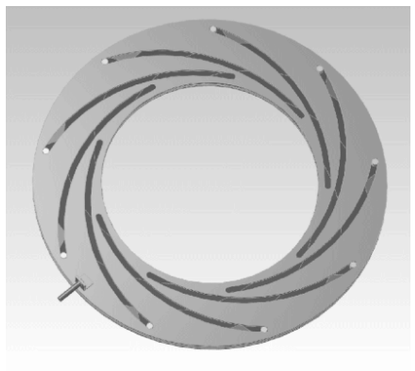


- Let’s design a camera
  - Idea 1: put a piece of film in front of an object
  - Do we get a reasonable image?

Pinhole camera



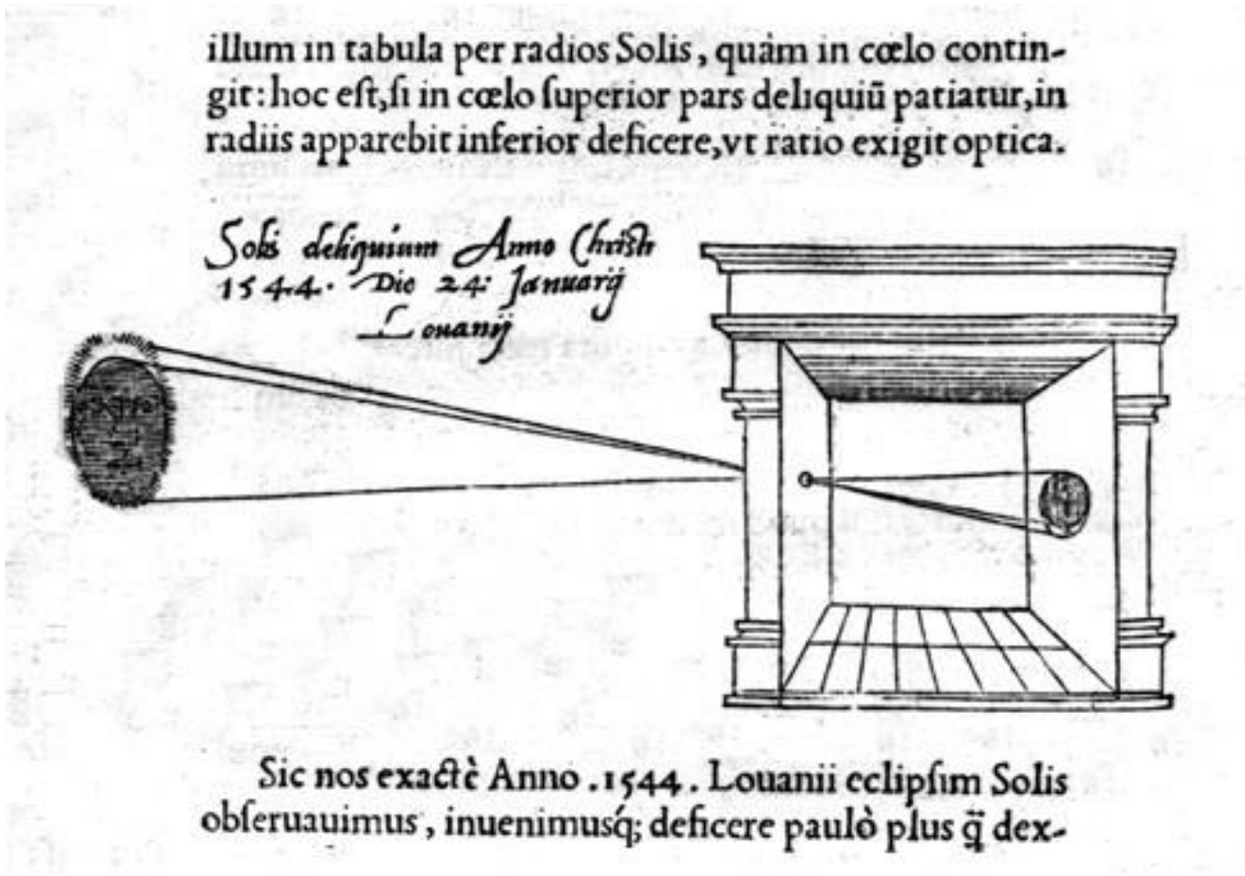
- Idea 2: Add a barrier to block off most of the rays
  - This reduces blurring
  - The opening is known as the **aperture**



Some history…

Milestones:

- Leonardo da Vinci (1452-1519):
  first record of camera *obscura (1502)*


illum in tabula per radios Solis, quàm in cœlo contingit: hoc eſt, ſi in cœlo ſuperior pars deliquiũ patiatur, in radiis apparebit inferior deficere, vt ratio exigit optica.

Solis deliquium Anno Christi 1544. Die 24. Januarij Louanij

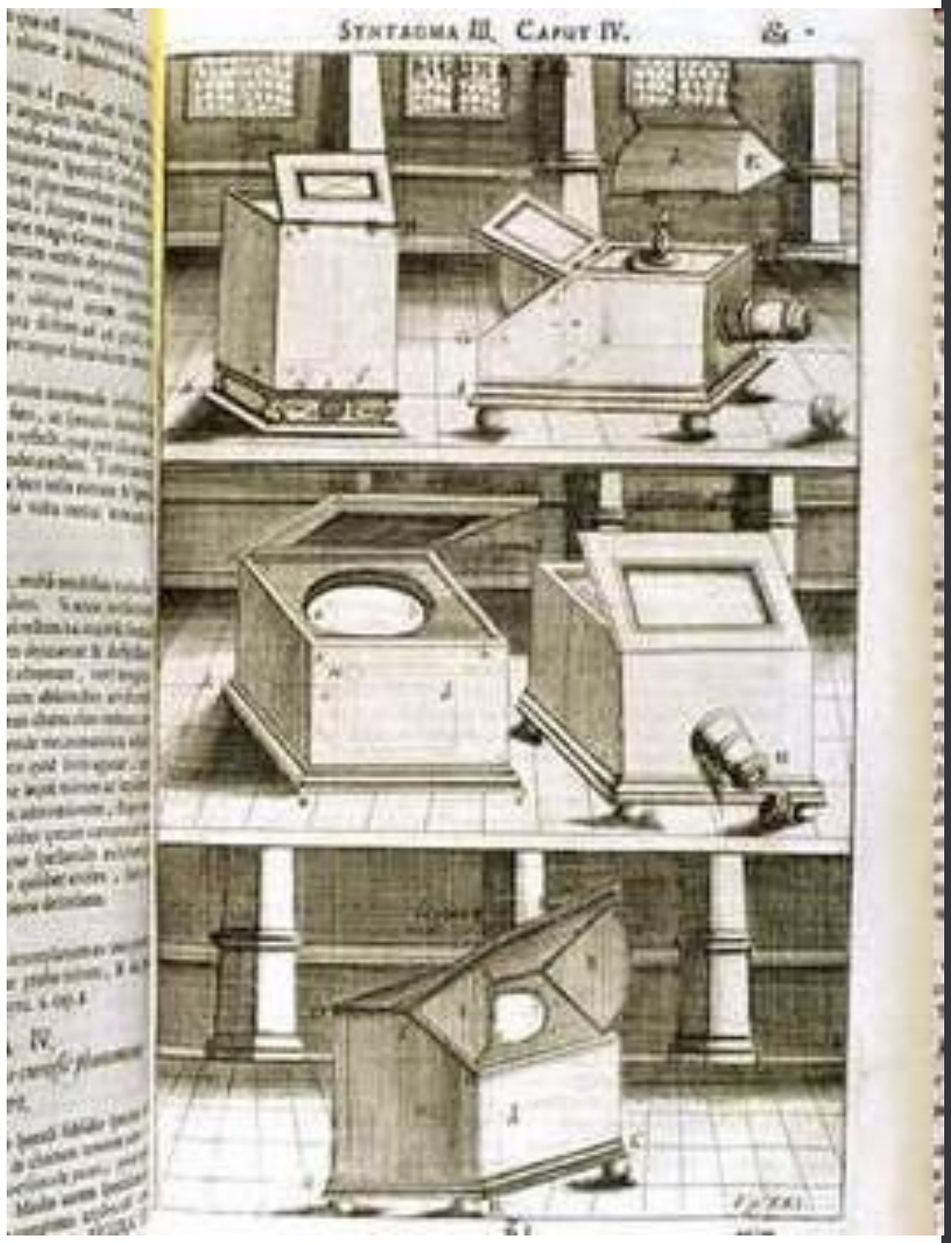
Sic nos exactè Anno .1544. Louanii eclipſim Solis obſeruauimus, inuenimusq; deficere paulò plus q̃ dex-

Image credit: Brendan Barry

Some history…

Milestones:

- Leonardo da Vinci (1452-1519):
  First record of camera *obscura*
- Johann Zahn (1685):
  First portable camera



Some history…

Milestones:

- Leonardo da Vinci (1452-1519):
  First record of camera *obscura*
- Johann Zahn (1685):
  First portable camera
- Joseph Nicéphore Niépce (1822):
  First photo - birth of photography

Photography (Niépce, “La Table Servie,” 1822)

Some history…

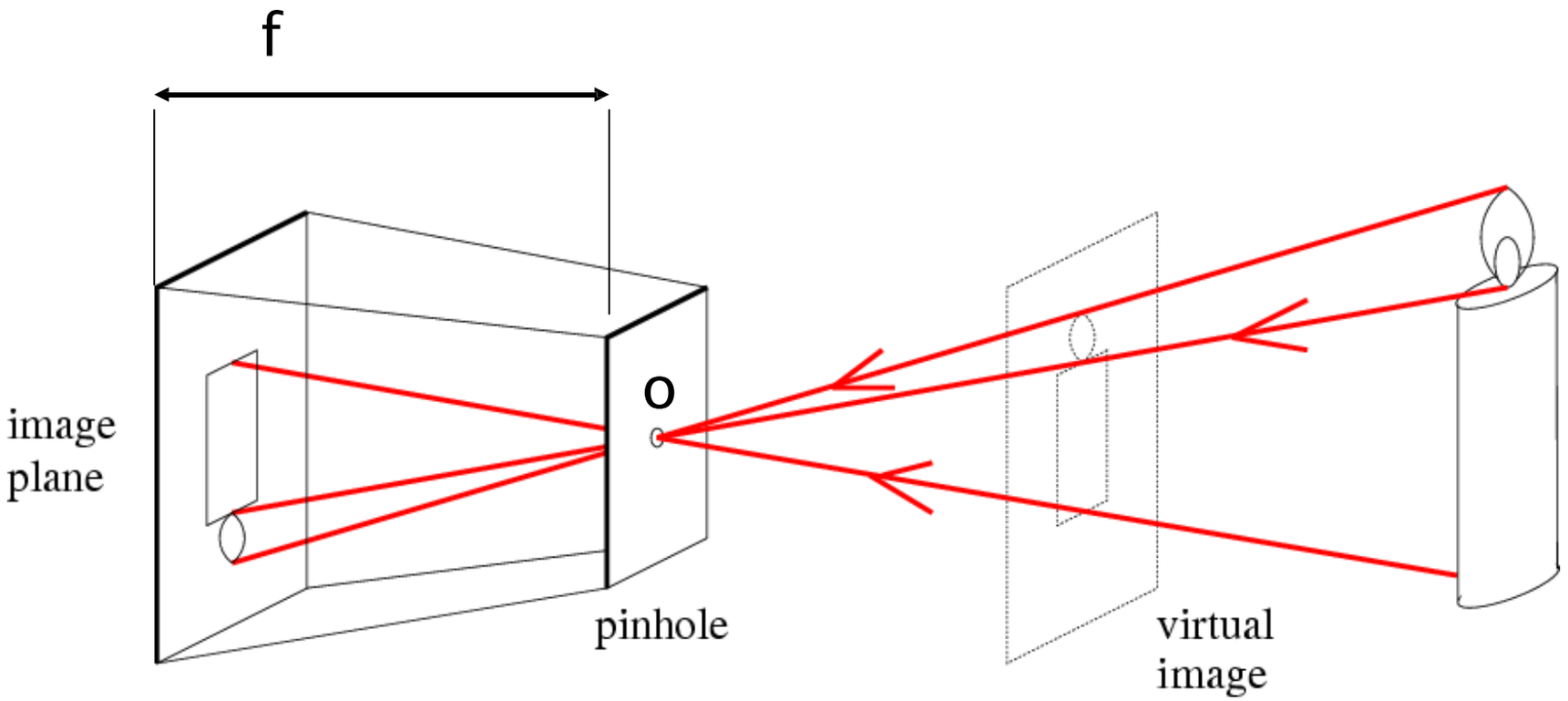
Milestones:

- Leonardo da Vinci (1452-1519):
  First record of camera *obscura*
- Johann Zahn (1685):
  First portable camera
- Joseph Nicéphore Niépce (1822):
  First photo - birth of photography

Photography (Niépce, “La Table Servie,” 1822)

- Daguerréotypes (1839)
- Photographic Film (Eastman, 1889)
- Cinema (Lumière Brothers, 1895)
- Color Photography (Lumière Brothers, 1908)

Pinhole camera

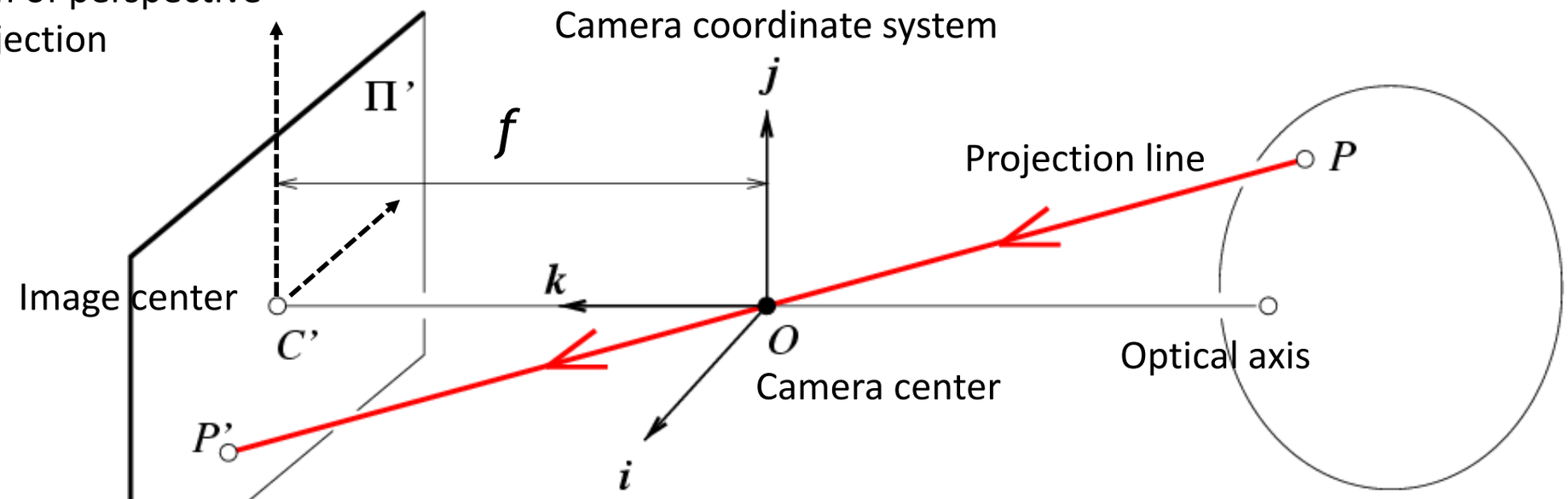


f = focal length

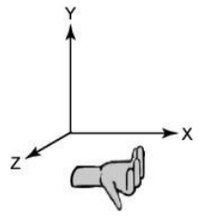
o = aperture = pinhole = center of the camera

Pinhole camera

Simplest form of perspective projection



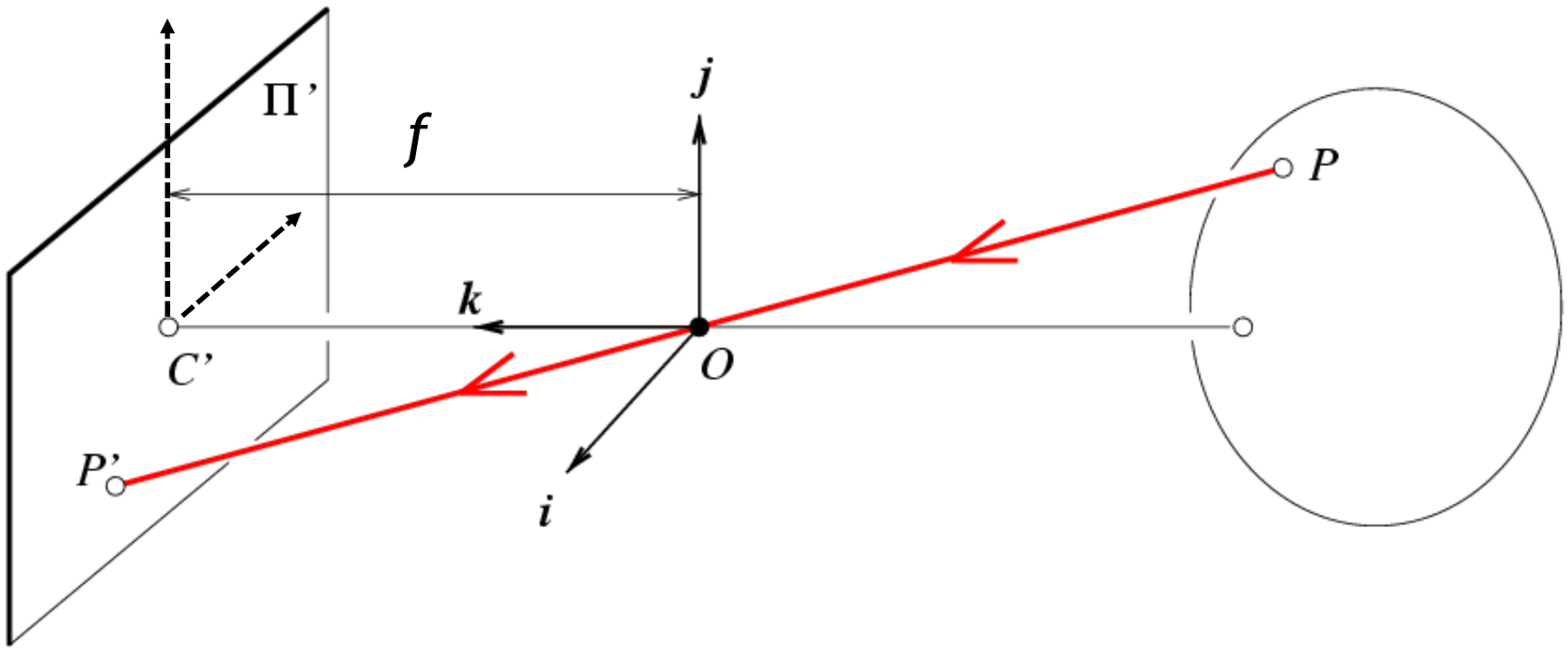
$$P = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \rightarrow P' = \begin{bmatrix} x' \\ y' \end{bmatrix} \quad \begin{cases} x' = f\,\dfrac{x}{z} \\ y' = f\,\dfrac{y}{z} \end{cases} \qquad \text{[Eq. 1]}$$

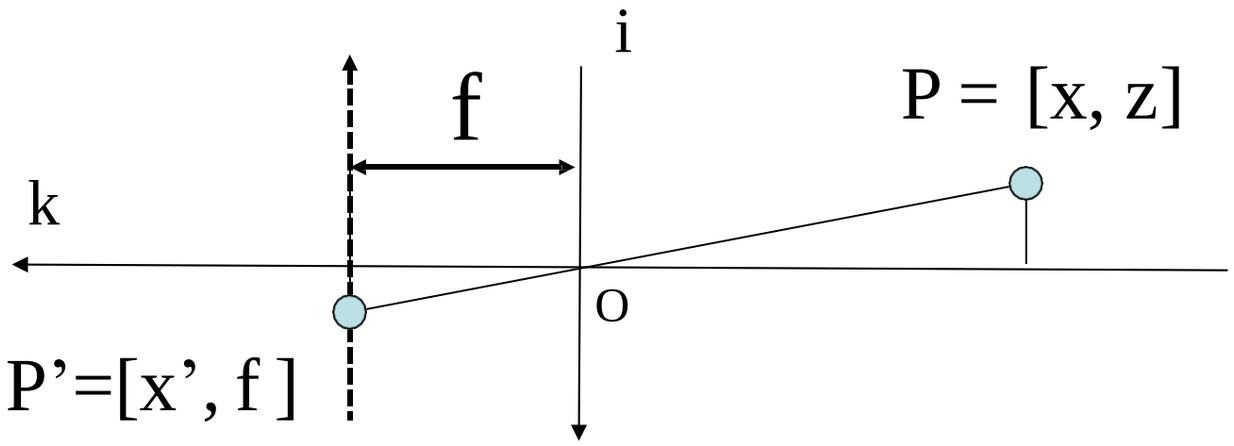


Right-handed coordinate system

Derived using similar triangles

Pinhole camera



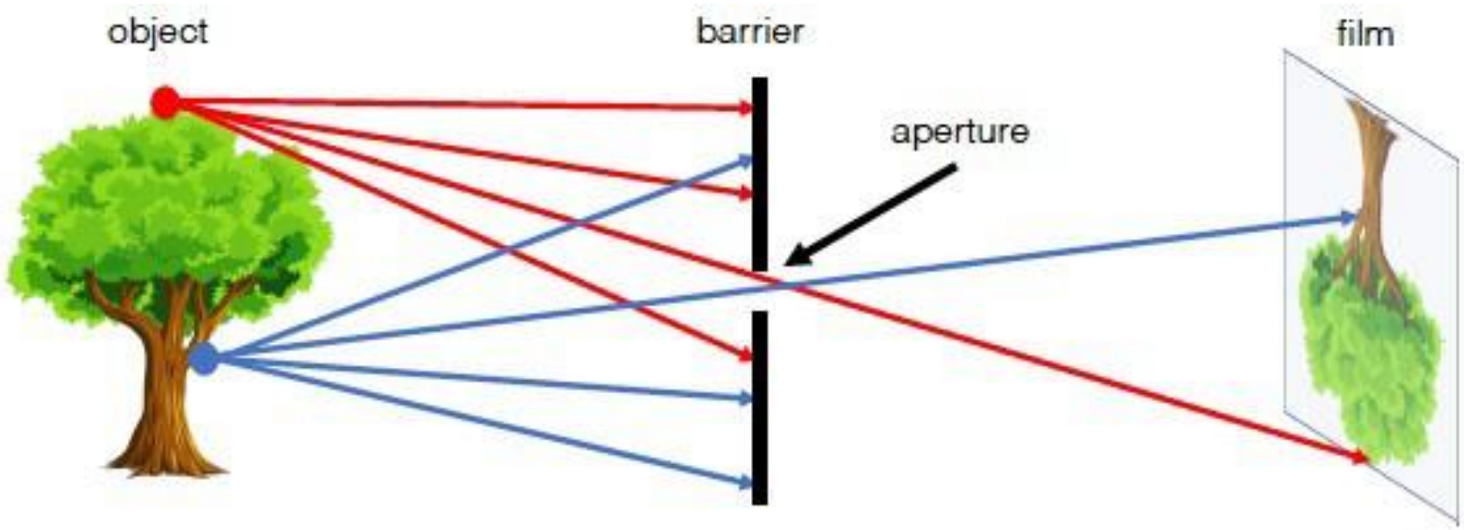


[Eq. 2]

$$\frac{x'}{f} = \frac{x}{z}$$

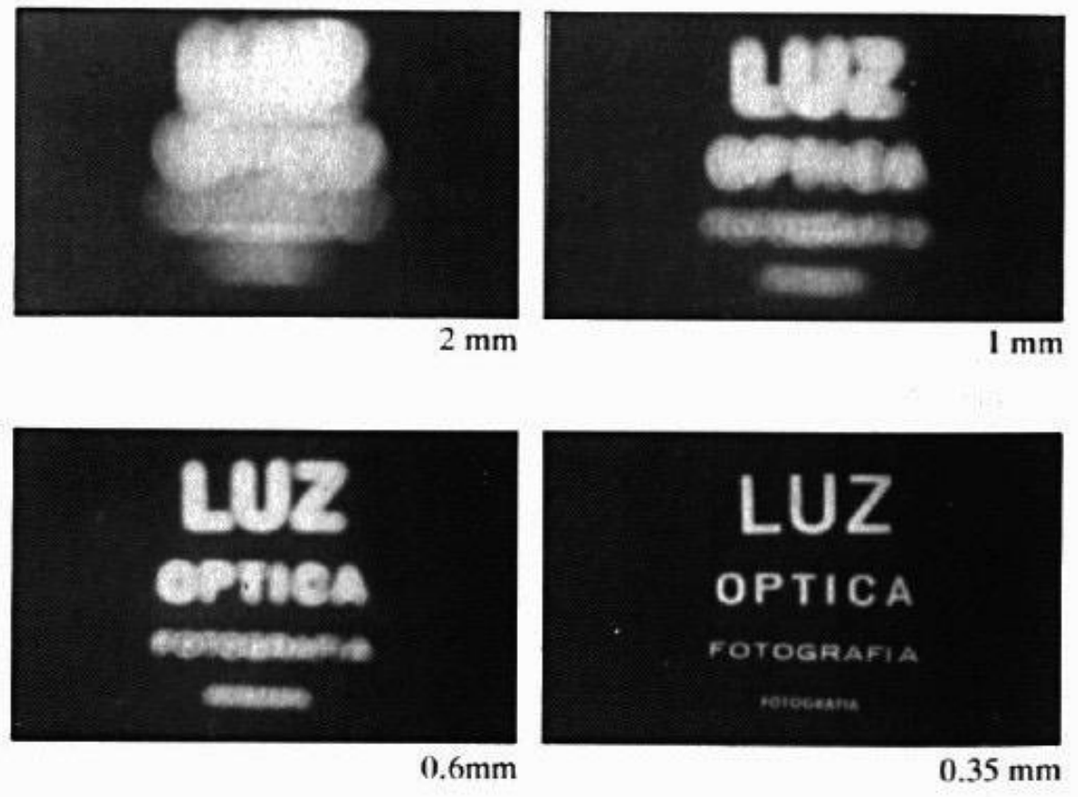
Pinhole camera

Is the size of the aperture important?



Assumption: aperture is a single point.

Shrinking aperture size

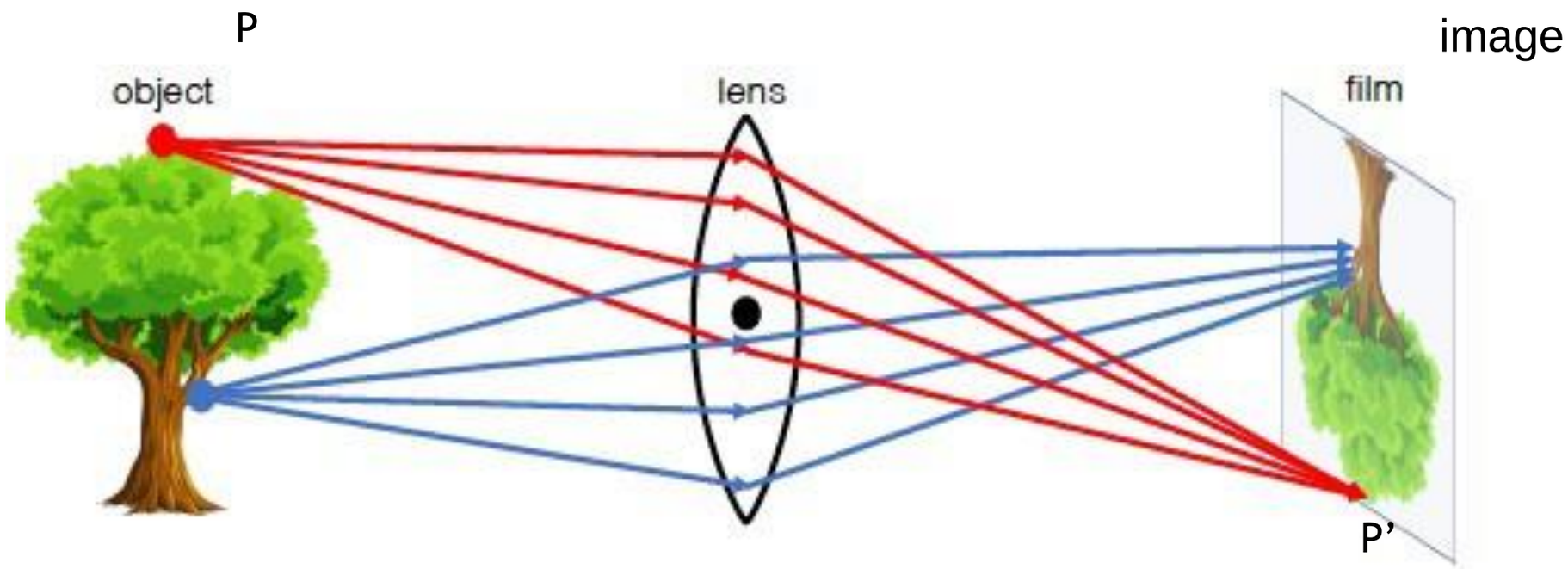


-What happens if the aperture is too small?

-Less light passes through

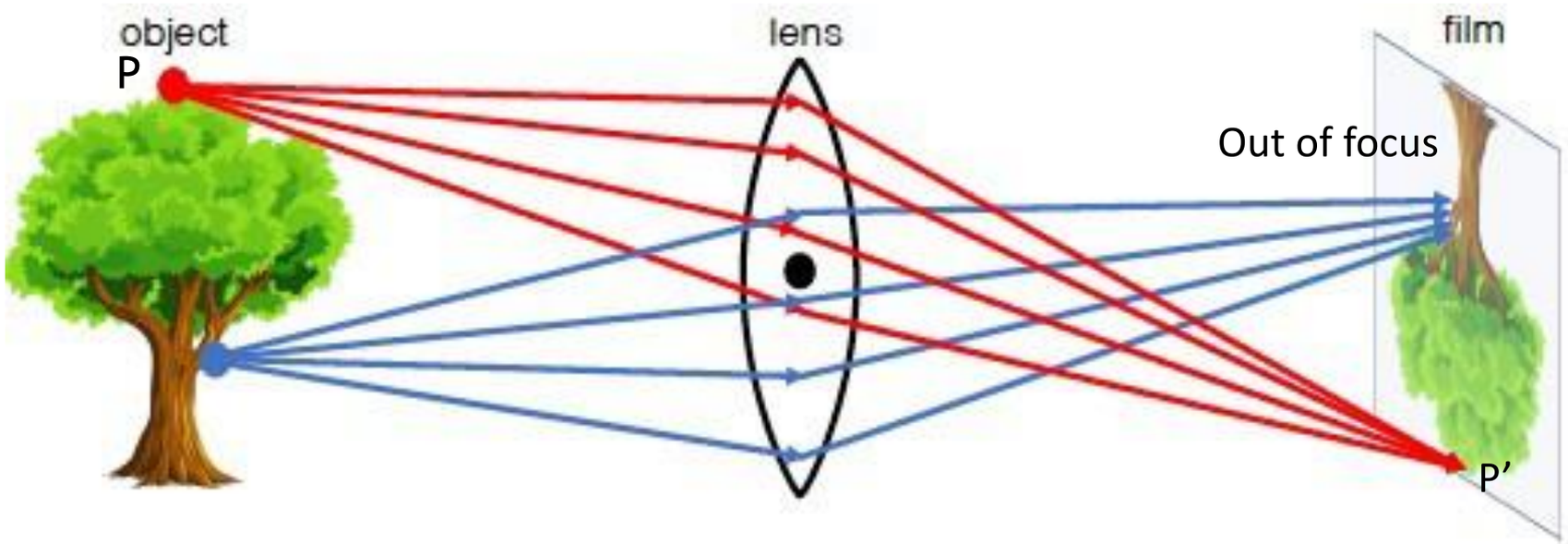
Adding lenses!

Cameras & Lenses



- A lens focuses light onto the film

Cameras & Lenses

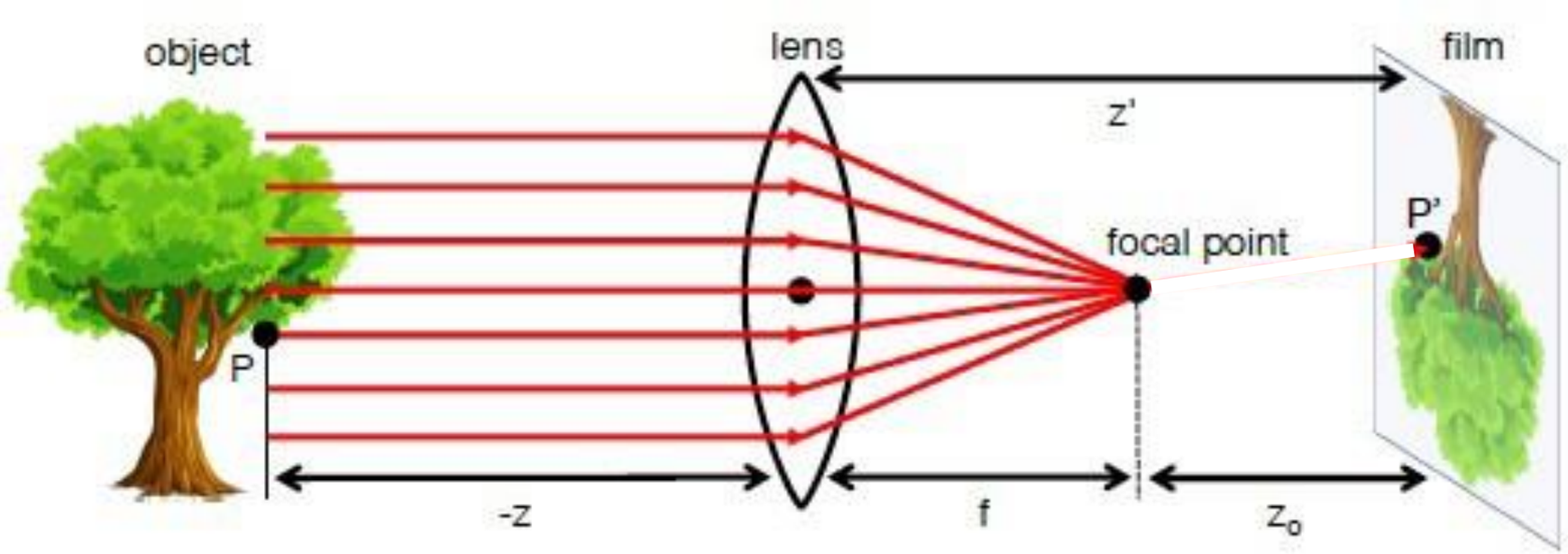


- A lens focuses light onto the film
  - There is a specific distance at which objects are “in focus”
  - Related to the concept of depth of field

Cameras & Lenses

- A lens focuses light onto the film
  - There is a specific distance at which objects are “in focus”
  - Related to the concept of depth of field

Cameras & Lenses

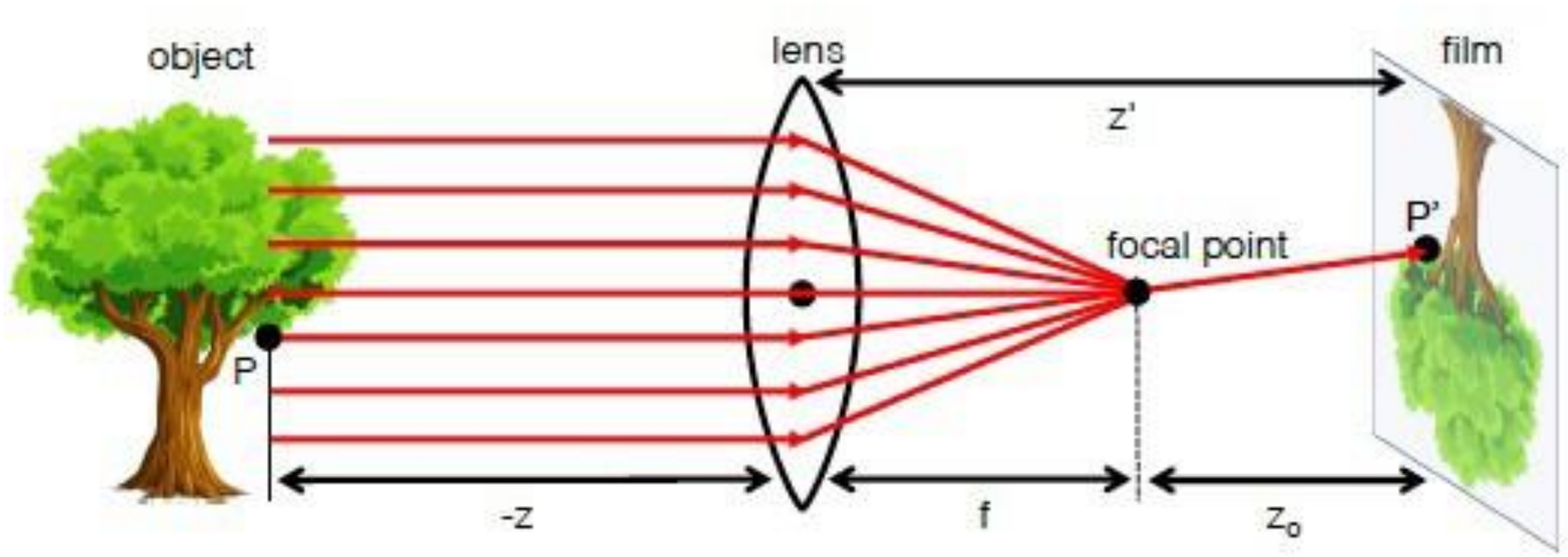


– A lens focuses light onto the film

- All rays parallel to the optical (or principal) axis converge to one point (the *focal point*) on a plane located at the *focal length f* from the center of the lens.
- Rays passing through the center are not deviated

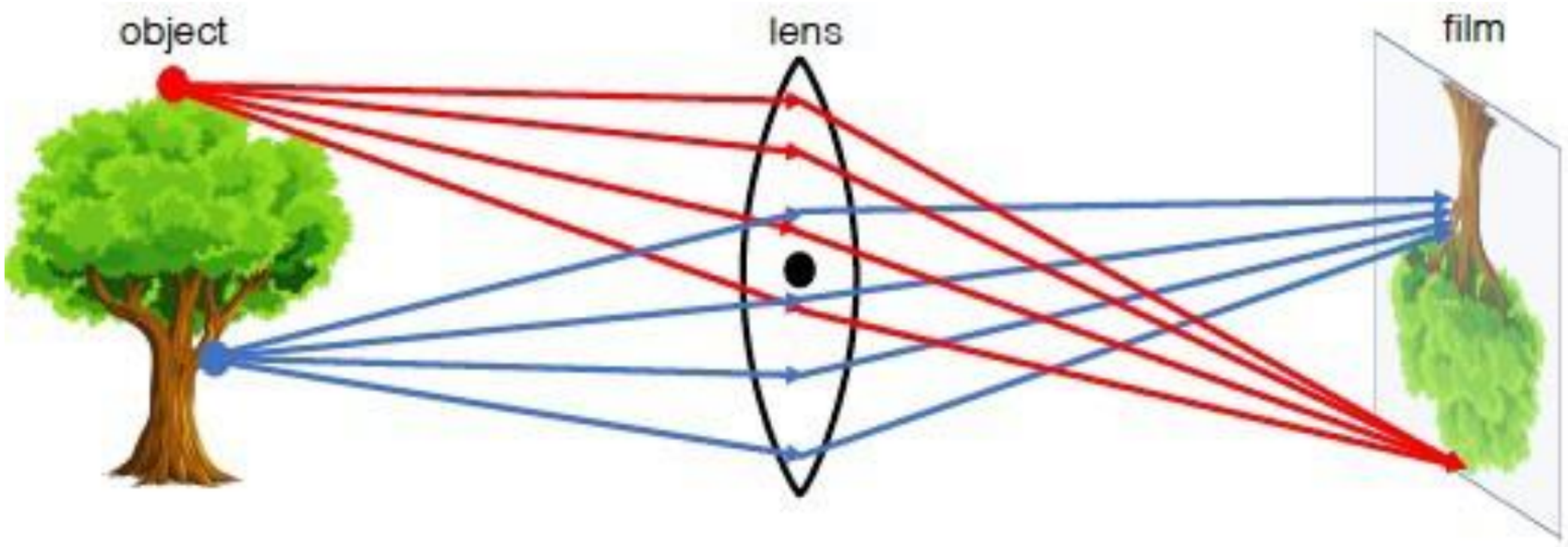
Paraxial refraction model

- Sharpness vs. brightness?
- Lens
  - Focus parallel light rays to the focal point



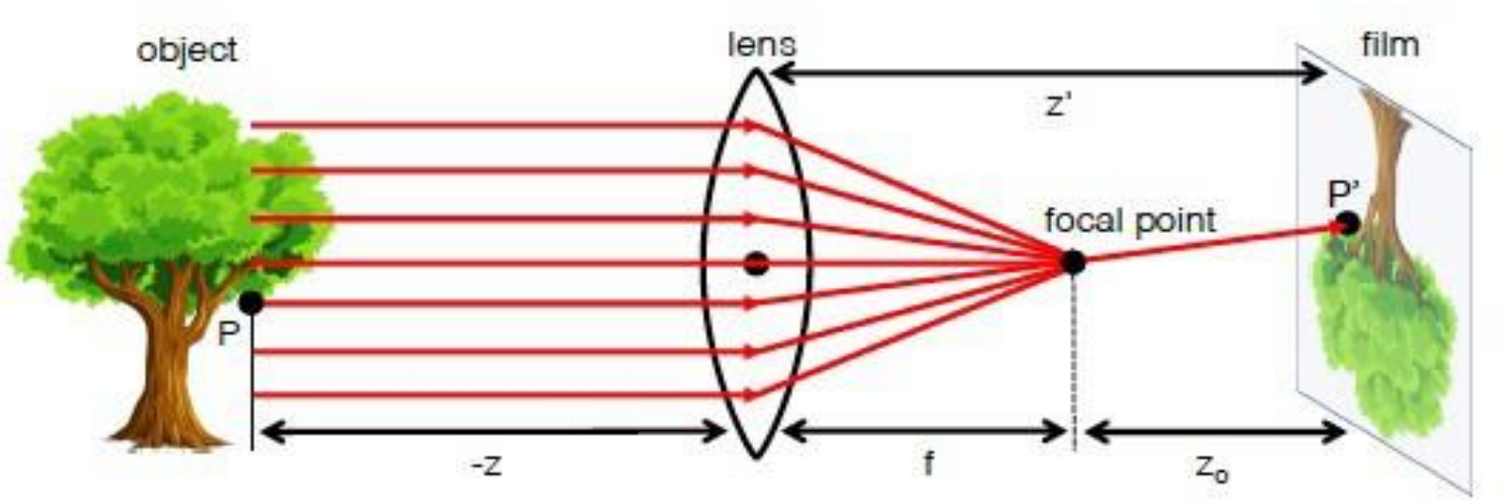
Paraxial refraction model

- Sharpness vs. brightness?
- Lens
  - Refract light and converge to a singe point



Paraxial refraction model

- Sharpness vs. brightness?
- Lens
  - Model takes advantage of the paraxial or “thin lens”



$$P' = \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} z' \frac{x}{z} \\ z' \frac{y}{z} \end{bmatrix}$$

Pinhole model

$$z' = f$$

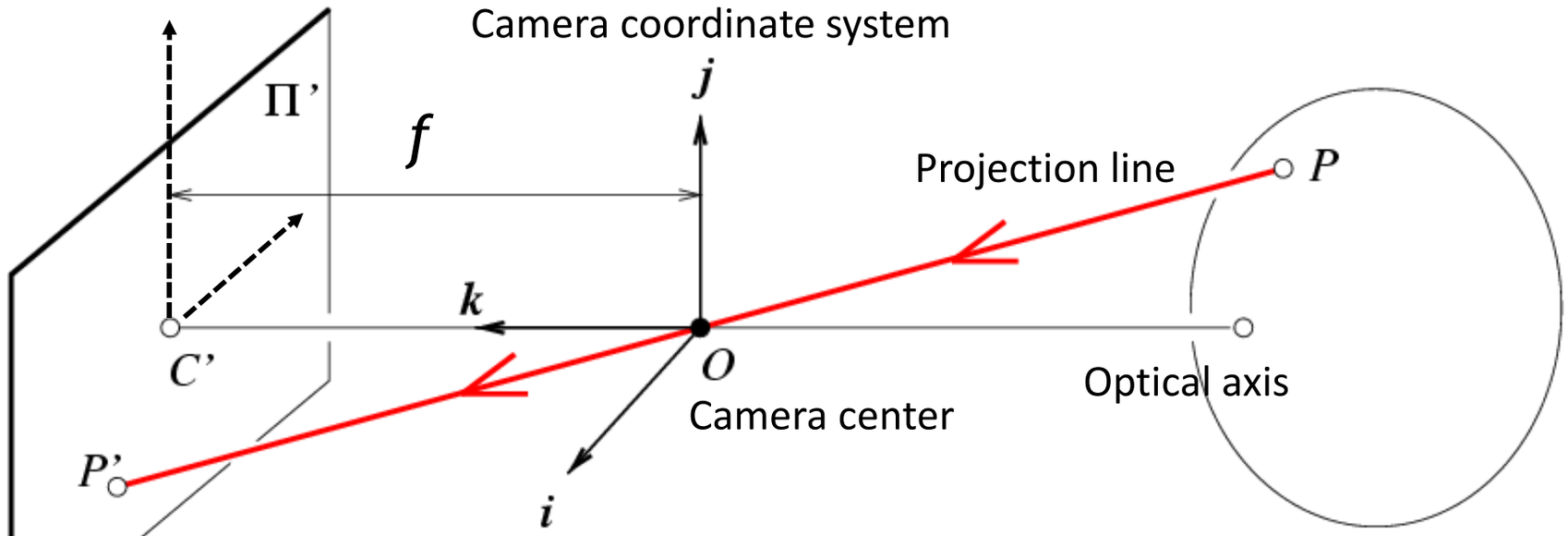
Lens-based model

$$z' = f + z_0$$

Geometry of camera

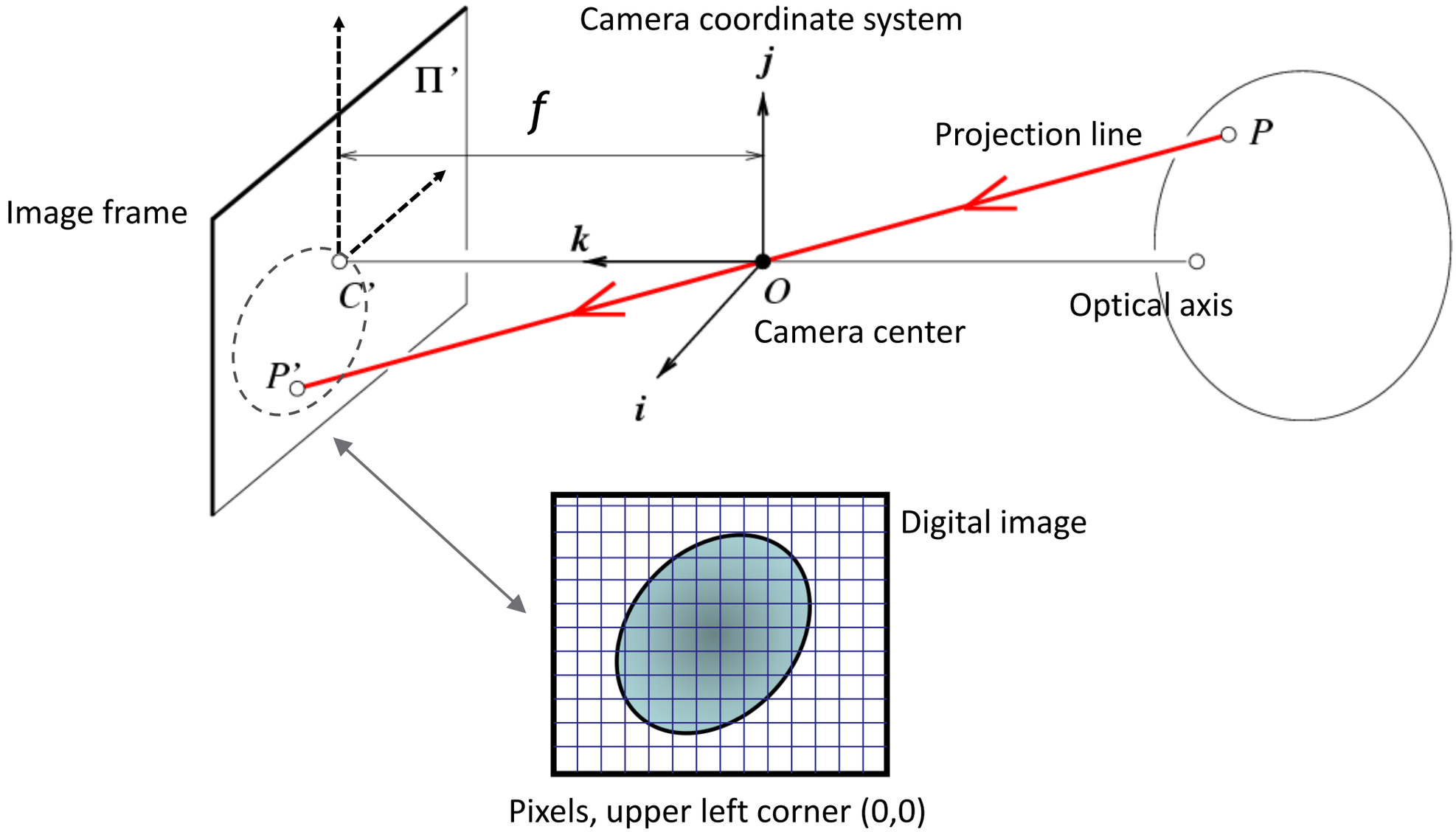
- Images
- Pinhole cameras
- Cameras & lenses
- The geometry of pinhole cameras
  - Intrinsic
  - Extrinsic

Pinhole camera

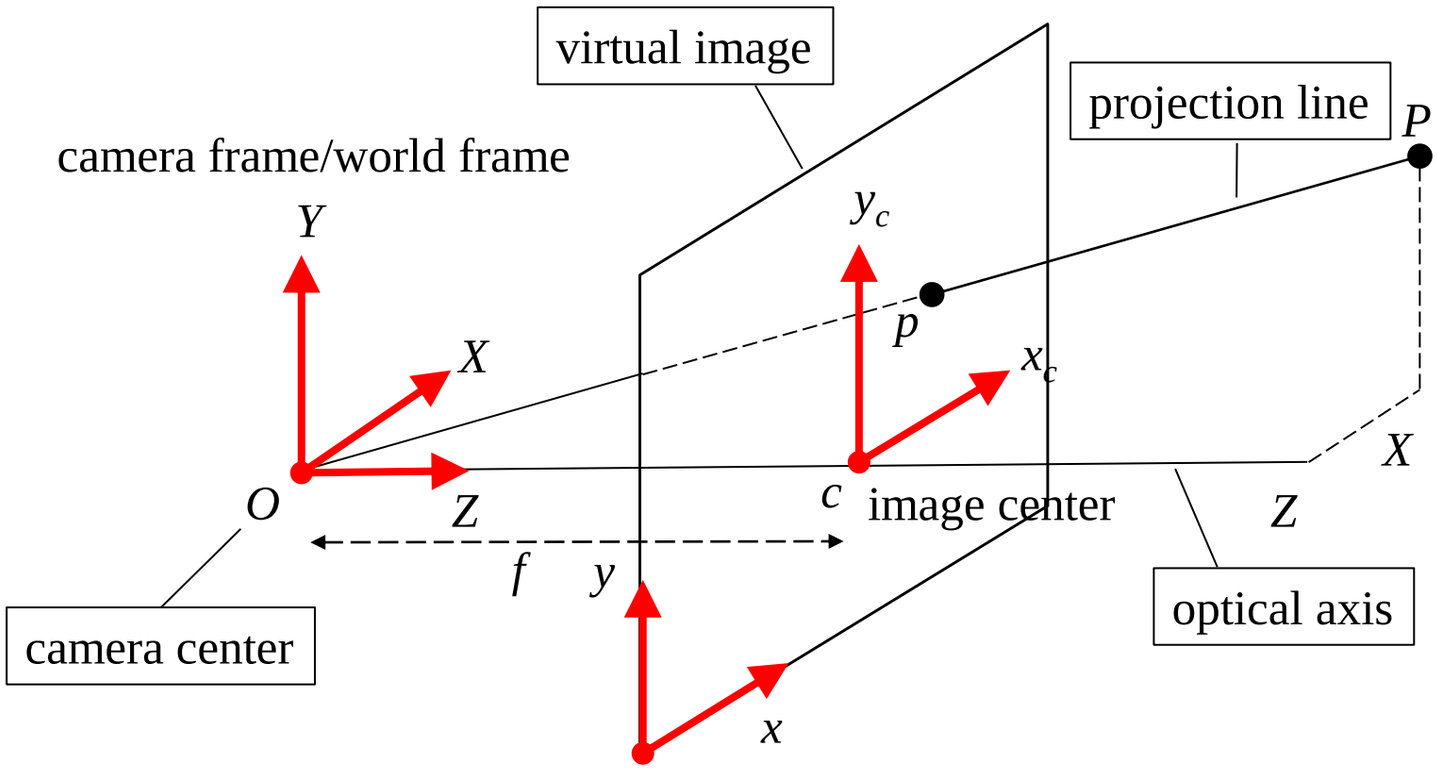


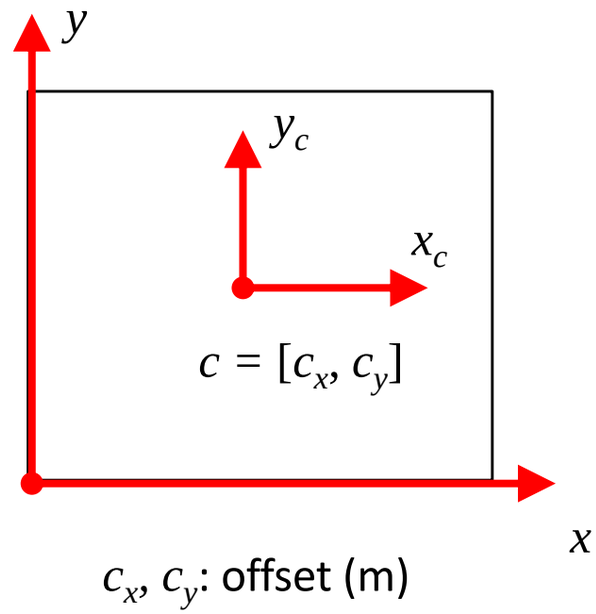
$$P = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \rightarrow P' = \begin{bmatrix} x' \\ y' \end{bmatrix} \quad \begin{cases} x' = f\,\dfrac{x}{z} \\ y' = f\,\dfrac{y}{z} \end{cases} \qquad \mathbb{R}^3 \rightarrow \mathbb{R}^2$$

From retina plane to images



Camera intrinsic: offset



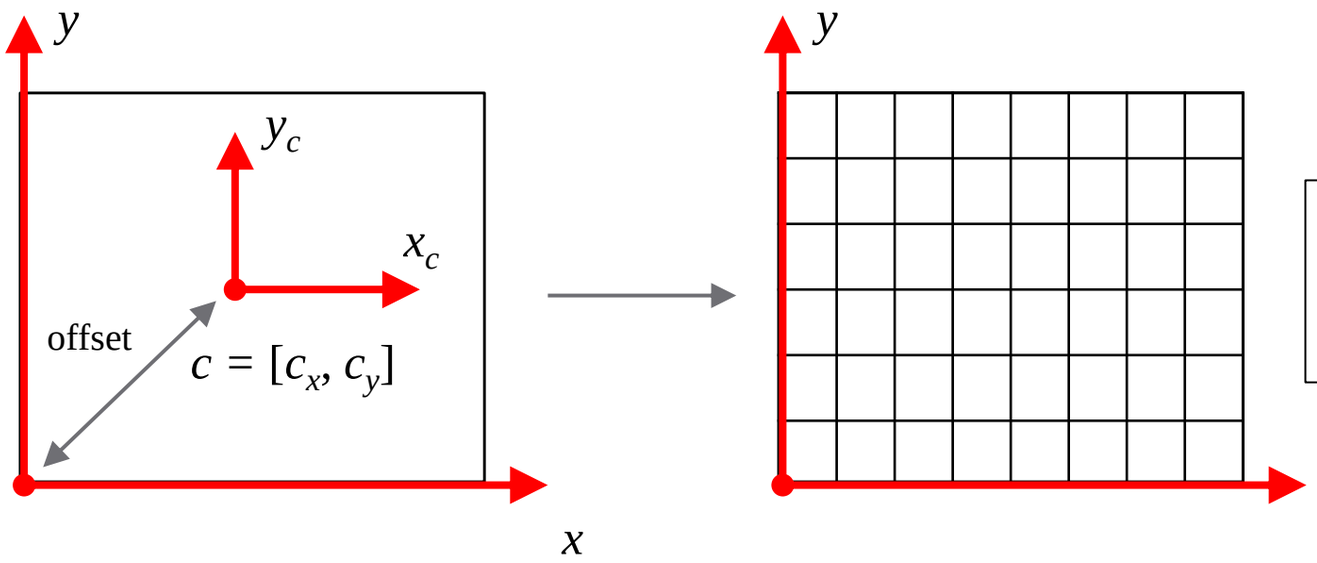


c_x, c_y: offset (m)

$$(x, y, z) \rightarrow \left(f\frac{x}{z} + c_x, f\frac{y}{z} + c_y\right)$$

We must consider: an unknown translation between the origin of the digital image coordinate system and the image center

Camera intrinsic: from metric to pixel



c_x, c_y: offset (pixel)

$$x = f_x \frac{X}{Z} + c_x, \quad y = f_y \frac{Y}{Z} + c_y$$

$$x = kf\frac{X}{Z}, \quad y = lf\frac{Y}{Z}$$

$$f_x = kf \qquad f_y = lf \text{ (pixel)}$$

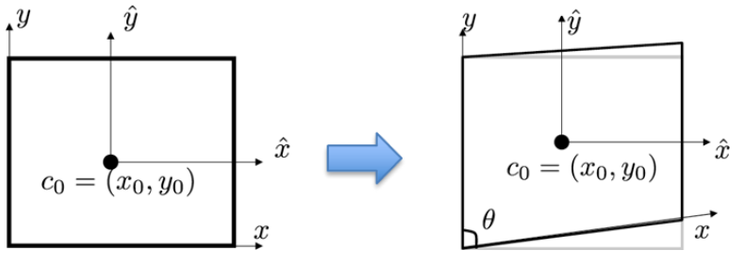
$$x = f_x \frac{X}{Z}, \quad y = f_y \frac{Y}{Z}$$

- x, y : coordinates (m)
- k, l : scale parameters (pixel/m)
- f : focal length (m)

Camera intrinsic: skew

$$x = f_x \frac{X}{Z} + c_x, \quad y = f_y \frac{Y}{Z} + c_y$$

- Image frame may not be exactly rectangular
  - Let ϑ denote skew angle between x- and y-axis



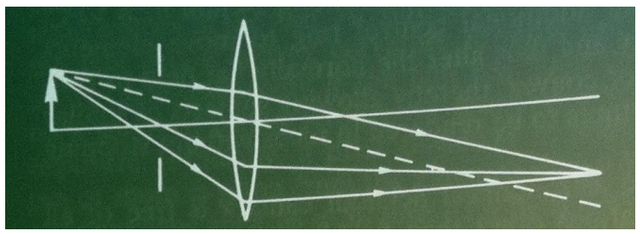
$$x = f_x \frac{X}{Z} - f_x \cot\theta \frac{Y}{Z} + c_x, \quad y = \frac{f_y}{\sin\theta} \frac{Y}{Z} + c_y$$

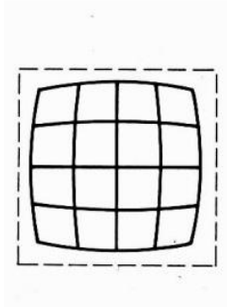
- Most cameras have zero-skew, but some degree of skewness may occur because of sensor manufacturing errors

Issues with lenses: Radial Distortion

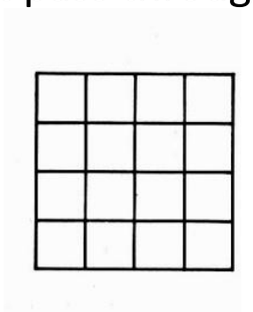
- Image frame may not be exactly rectangular

Common aberration: referred to as **radial distortion**
the **image magnification** to decrease or increase as a **function of the distance to the optical axis**
Reason: different portions of the lens have differing focal lengths.
Deviations are most noticeable for rays that pass through the edge of the lens

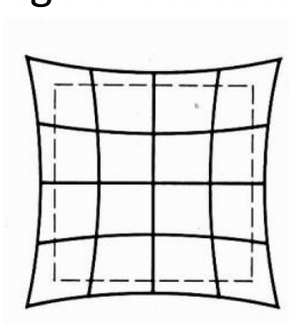




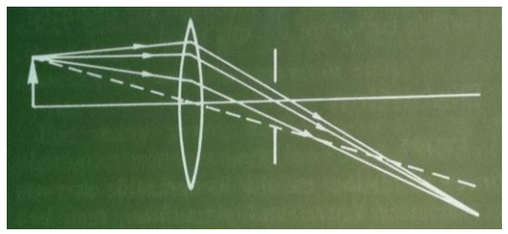
Barrel



Normal



Pincushion



Camera matrix model

$$x = f_x \frac{X}{Z} - f_x \cot\theta \frac{Y}{Z} + c_x, \quad y = \frac{f_y}{\sin\theta} \frac{Y}{Z} + c_y$$

- Combine all the parameters

$$P' = \begin{bmatrix} x' \\ y' \\ z \end{bmatrix} = K \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = KP \qquad K = \text{?}$$

Homogeneous coordinates

- Internal characteristics: focal length, skew, distortion, and image center

Homogeneous Coordinate System

One way to solve this problem is to change the coordinate systems. For example, we introduce a new coordinate, such that any point $P' = (x', y')$ becomes $(x', y', 1)$. Similarly, any point $P = (x, y, z)$ becomes $(x, y, z, 1)$. This augmented space is referred to as the **homogeneous coordinate system**. As demonstrated previously, to convert a Euclidean vector $(v_1, ..., v_n)$ to homogeneous coordinates, we simply append a 1 in a new dimension to get $(v_1, ..., v_n, 1)$. Note that the equality between a vector and its homogeneous coordinates only occurs when the final coordinate equals one. Therefore, when converting back from arbitrary homogeneous coordinates $(v_1, ..., v_n, w)$, we get Euclidean coordinates $(\frac{v_1}{w}, ..., \frac{v_n}{w})$. Using homogeneous coordinates, we can formulate

$$P'_h = \begin{bmatrix} \alpha x + c_x z \\ \beta y + c_y z \\ z \end{bmatrix} = \begin{bmatrix} \alpha & 0 & c_x & 0 \\ 0 & \beta & c_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha & 0 & c_x & 0 \\ 0 & \beta & c_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} P_h \qquad \boldsymbol{P_i'} = \begin{bmatrix} \frac{x}{z} \\ \frac{y}{z} \end{bmatrix}$$

Camera matrix model

$$P_h' = \begin{bmatrix} x' \\ y' \\ z \end{bmatrix} = \begin{bmatrix} f_x & -f_x \cot\theta & c_x & 0 \\ 0 & \frac{f_y}{\sin\theta} & c_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & -f_x \cot\theta & c_x & 0 \\ 0 & \frac{f_y}{\sin\theta} & c_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} P = MP_h$$

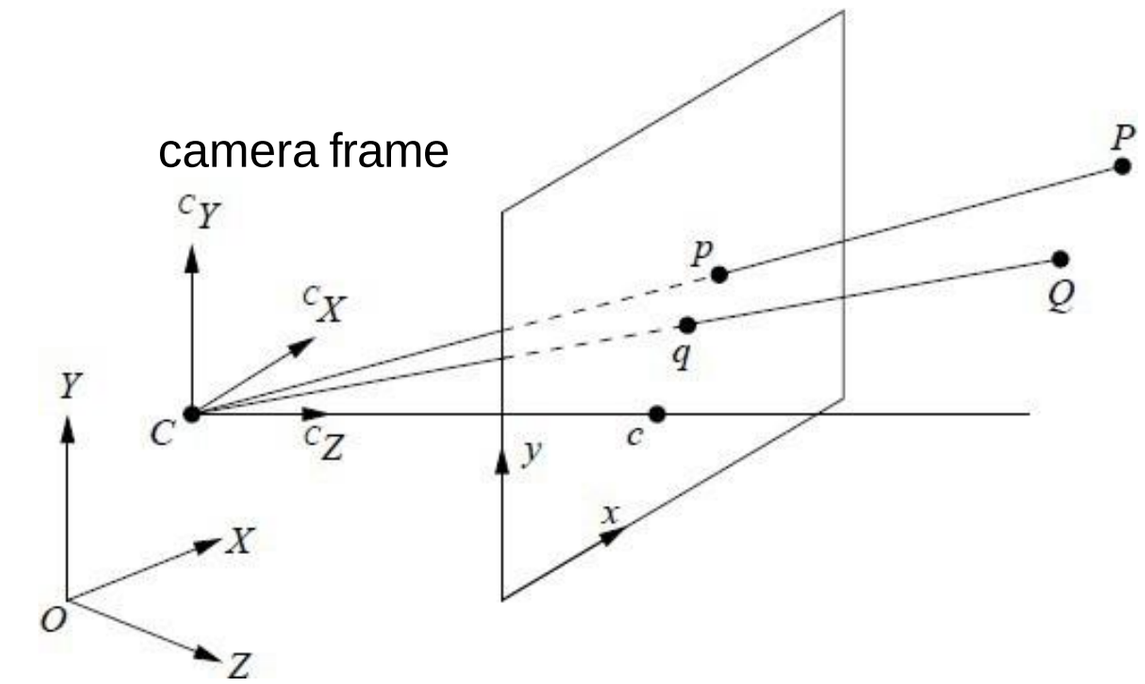
$$P_h' = MP_h = \begin{bmatrix} f_x & -f_x \cot\theta & c_x \\ 0 & \frac{f_y}{\sin\theta} & c_y \\ 0 & 0 & 1 \end{bmatrix} [I \quad 0] P_h = K[I \quad 0]P_h$$

$$P_i' = \frac{1}{Z} M P_i$$

K: camera matrix $[I \quad 0]$: 4×3

Extrinsic Parameters

- Camera frame is not aligned with world frame
- Camera can move and rotate



Extrinsic Parameters

- Camera frame is not aligned with world frame
- Camera can move and rotate
- Rigid transformation between them

$$P = \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} P_w$$

$$R = R_z(\alpha)\, R_y(\beta)\, R_x(\gamma) = \underbrace{\begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\text{yaw}} \underbrace{\begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix}}_{\text{pitch}} \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\gamma & -\sin\gamma \\ 0 & \sin\gamma & \cos\gamma \end{bmatrix}}_{\text{roll}}$$

$$= \begin{bmatrix} \cos\alpha\cos\beta & \cos\alpha\sin\beta\sin\gamma - \sin\alpha\cos\gamma & \cos\alpha\sin\beta\cos\gamma + \sin\alpha\sin\gamma \\ \sin\alpha\cos\beta & \sin\alpha\sin\beta\sin\gamma + \cos\alpha\cos\gamma & \sin\alpha\sin\beta\cos\gamma - \cos\alpha\sin\gamma \\ -\sin\beta & \cos\beta\sin\gamma & \cos\beta\cos\gamma \end{bmatrix}$$

$$T = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

$$P'_h = K[I \quad 0]P_h = K[I \quad 0]\begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} P_{w_h} = K[R \quad T]P_{w_h} = MP_{w_h}$$

From 3D points to pixels

- Combine intrinsic and extrinsic parameters

$$P_h = K[I \quad 0]P_h \qquad\qquad P_h = \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} P_{w_h}$$

- Use a simpler notation

$$P'_{h\,3\times1} = M_{3\times4}P_{w_h} = K_{3\times3}[R \quad T]_{3\times4}P_{w_h 4\times1} \qquad \mathbb{R}^4 \rightarrow \mathbb{R}^3$$

$$\boxed{P_i' = \frac{1}{Z}MP_i} \qquad \mathbb{R}^3 \rightarrow \mathbb{R}^2$$

Summary Camera Models

- Simplest camera model: pinhole model.
- Most commonly used model: perspective model
- Total 11 degrees of freedom for M
- Intrinsic parameters:
  - Focal length, principal point (image center), skew factor
- Extrinsic parameters:
  - Camera rotation and translation

Further reading :

R. Szeliski. *Computer Vision: Algorithms and Applications*. Springer, 2010.

- Camera models: Section 2.1.5
- Lens distortion: Section 2.1.6

Camera Calibration: General Idea

- Why is camera calibration necessary?
  - Given 3D scene, knowing the precise 3D to 2D projection requires
    - Intrinsic and extrinsic parameters
  - Reconstructing 3D geometry from images also requires these parameters

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

Internal (intrinsic) parameters

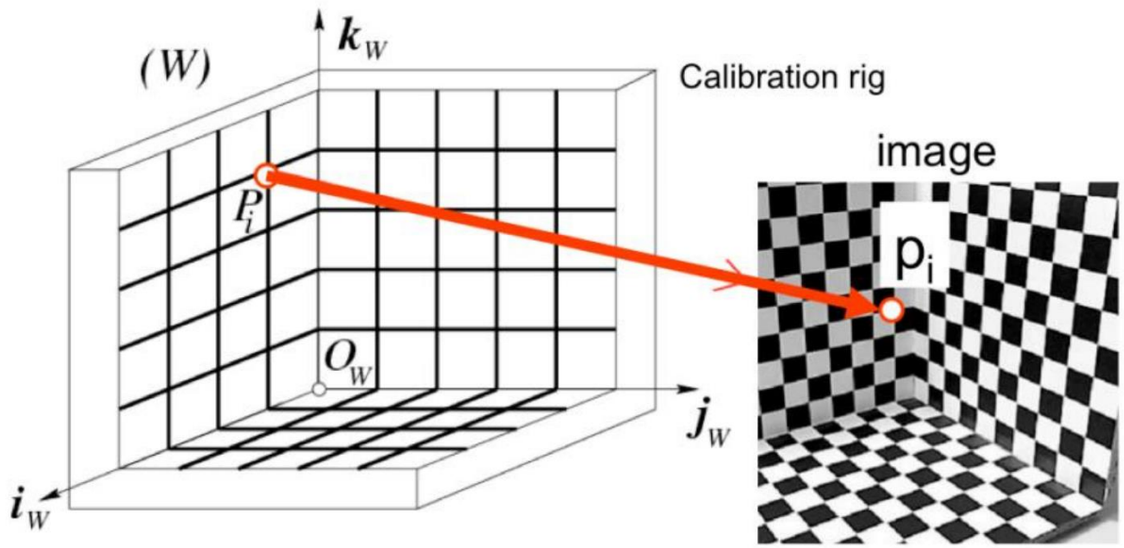
External (extrinsic) parameters

Camera Calibration: General Idea

- Why is camera calibration necessary?
- What information do we have?
  - Images only

Camera Calibration: General Idea

- Why is camera calibration necessary?
- What information do we have?
  - Images only
  - Calibration setup



setup of an example calibration rig

Camera Calibration: General Idea

- Why is camera calibration necessary?
- What information do we have?
- Camera calibration
  - Recovering K
  - Recovering R and $\mathbf{t}$

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

Internal (intrinsic) parameters

External (extrinsic) parameters

Camera Calibration: General Idea

- How many parameters to recover?

$$\mathbf{p} = M\mathbf{P}$$

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

Internal (intrinsic) parameters

External (extrinsic) parameters

Camera Calibration: General Idea

- How many parameters to recover?
  - How many intrinsic parameters?

$$K = \begin{bmatrix} f_x & -f_x \cot\theta & c_x \\ 0 & \dfrac{f_y}{\sin\theta} & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{p} = M\mathbf{P}$$
$$= \boxed{K}\begin{bmatrix} R & \mathbf{t} \end{bmatrix}\mathbf{P}$$

Internal (intrinsic) parameters

Camera Calibration: General Idea

- How many parameters to recover?
  - How many intrinsic parameters?
  - How many extrinsic parameters?

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

External (extrinsic) parameters

$$R = \begin{bmatrix} \mathbf{r}_1^T \\ \mathbf{r}_2^T \\ \mathbf{r}_3^T \end{bmatrix}, \quad \mathbf{t} = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}$$

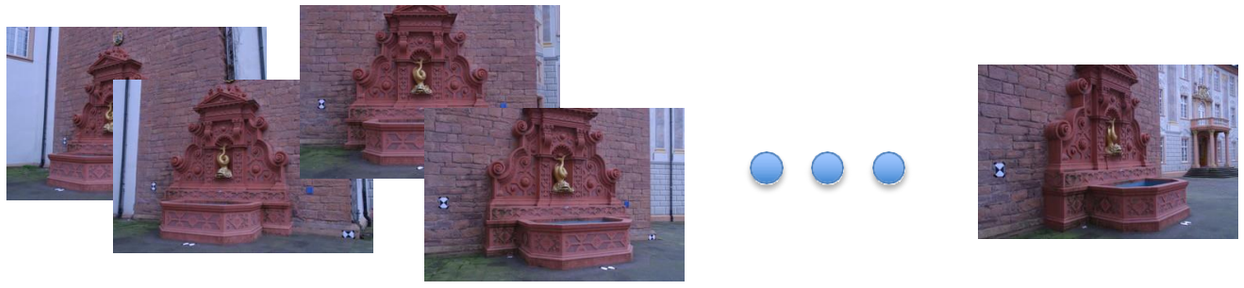
Camera Calibration: General Idea

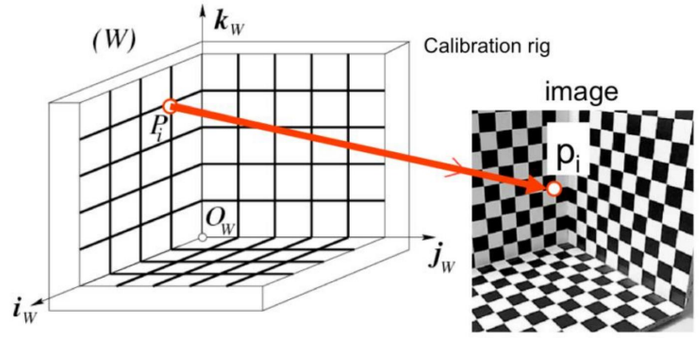
- How many parameters to recover: 11
  - 5 intrinsic parameters
    - 2 for focal lengths
    - 2 for offset (image center, or principle point)
    - 1 for skewness
  - 6 extrinsic parameters
    - 3 for rotation
    - 3 for translation

$$K = \begin{bmatrix} f_x & -f_x \cot\theta & c_x \\ 0 & \frac{f_y}{\sin\theta} & c_y \\ 0 & 0 & 1 \end{bmatrix}, \quad R = \begin{bmatrix} \mathbf{r}_1^T \\ \mathbf{r}_2^T \\ \mathbf{r}_3^T \end{bmatrix}, \quad \mathbf{t} = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}$$

Camera Calibration: General Idea

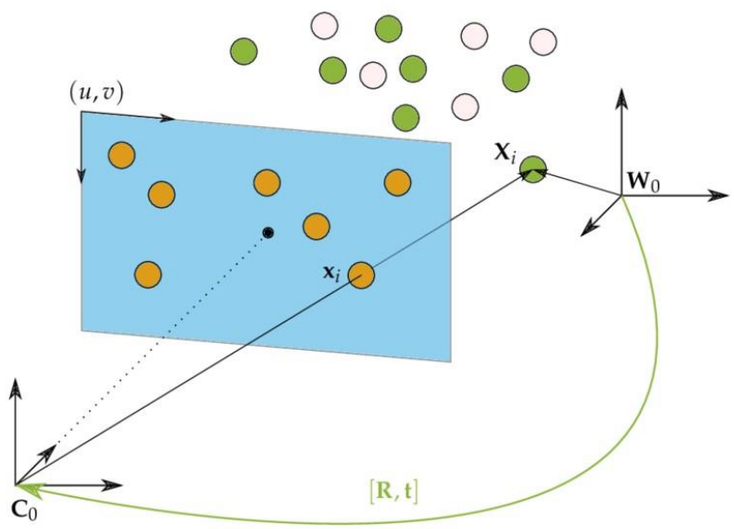
- What information to use?





– Corresponding 3D-2D point pairs

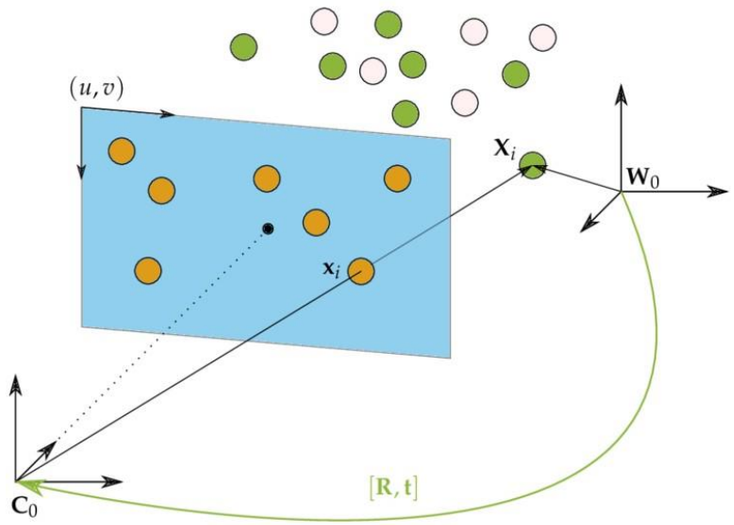
$$\mathbf{p} = M\mathbf{P}$$
$$= K \begin{bmatrix} R & \mathbf{t} \end{bmatrix} \mathbf{P}$$



Camera Calibration: General Idea

- What information to use?
  - Corresponding 3D-2D point pairs
    - How many pairs do we need?

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



Camera Calibration: General Idea

- What information to use?
  - Corresponding 3D-2D point pairs
    - How many pairs do we need?
      - How much information each pair of corresponding point can provide?

$$M = \begin{bmatrix} \mathbf{m}_1^T \\ \mathbf{m}_2^T \\ \mathbf{m}_3^T \end{bmatrix}$$

$$\mathbf{p} = M\mathbf{P} \Rightarrow \mathbf{p}_i = \begin{bmatrix} u_i \\ v_i \end{bmatrix} = M\mathbf{P}_i = \begin{bmatrix} \frac{\mathbf{P}_i^T \mathbf{m}_1}{\mathbf{P}_i^T \mathbf{m}_3} \\ \\ \frac{\mathbf{P}_i^T \mathbf{m}_2}{\mathbf{P}_i^T \mathbf{m}_3} \end{bmatrix} \Rightarrow \begin{aligned} \mathbf{P}_i^T \mathbf{m}_1 - u_i(\mathbf{P}_i^T \mathbf{m}_3) = 0 \\ \mathbf{P}_i^T \mathbf{m}_2 - v_i(\mathbf{P}_i^T \mathbf{m}_3) = 0 \end{aligned}$$

Camera Calibration: General Idea

- What information to use?
  - Corresponding 3D-2D point pairs
    - How many pairs do we need?
      - Each 3D-2D point pair -> 2 equations
      - 11 unknown -> 6 point correspondence
      - Use more to handle noisy data

$$\mathbf{p} = M\mathbf{P} \Rightarrow \mathbf{p}_i = \begin{bmatrix} u_i \\ v_i \end{bmatrix} = M\mathbf{P}_i = \begin{bmatrix} \frac{\mathbf{P}_i^T\mathbf{m}_1}{\mathbf{P}_i^T\mathbf{m}_3} \\ \\ \frac{\mathbf{P}_i^T\mathbf{m}_2}{\mathbf{P}_i^T\mathbf{m}_3} \end{bmatrix} \Rightarrow \begin{aligned} \mathbf{P}_i^T\mathbf{m}_1 - u_i(\mathbf{P}_i^T\mathbf{m}_3) = 0 \\ \mathbf{P}_i^T\mathbf{m}_2 - v_i(\mathbf{P}_i^T\mathbf{m}_3) = 0 \end{aligned}$$

$\mathbf{m}_1$, $\mathbf{m}_2$, $\mathbf{m}_3$: the three rows of the projection matrix *M*

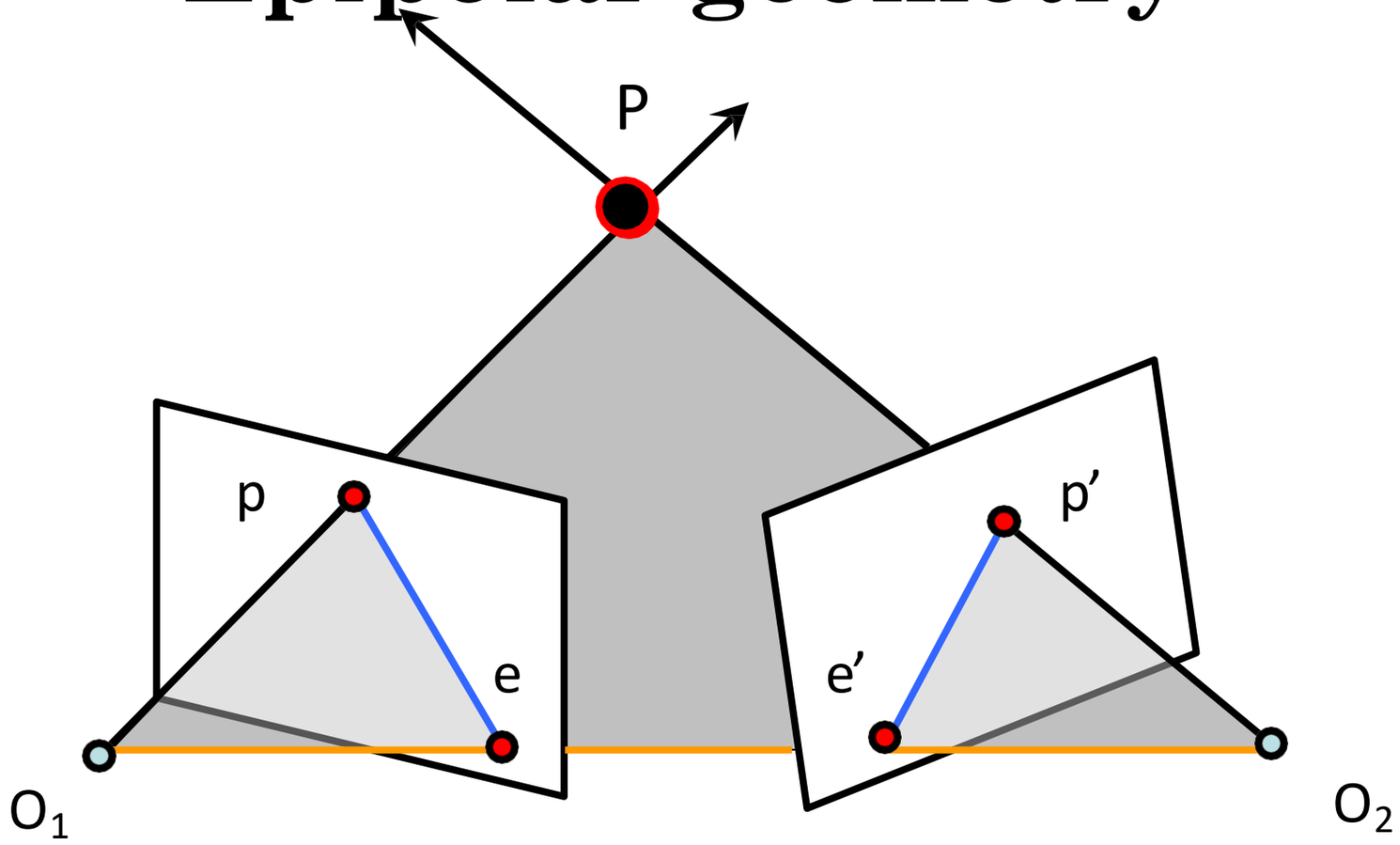
Range estimation using camera?

- Consider we have access to a camera image
- Can we estimate distant objects/pixels distances?
- What if we have access to several images of a scene?
- Concepts: Monocular depth estimation
- Concepts: Stereo depth estimation, Multiview, SfM

Stereo Principle

- If you know
  - ❖ Intrinsic parameters of each camera
  - ❖ The relative pose between the cameras
- If you measure
  - ❖ An image point in the left camera
  - ❖ The corresponding point in the right camera
- Each image point corresponds to a ray emanating from that camera
- You can intersect the rays (triangulate) to fine absolute point position

Epipolar geometry

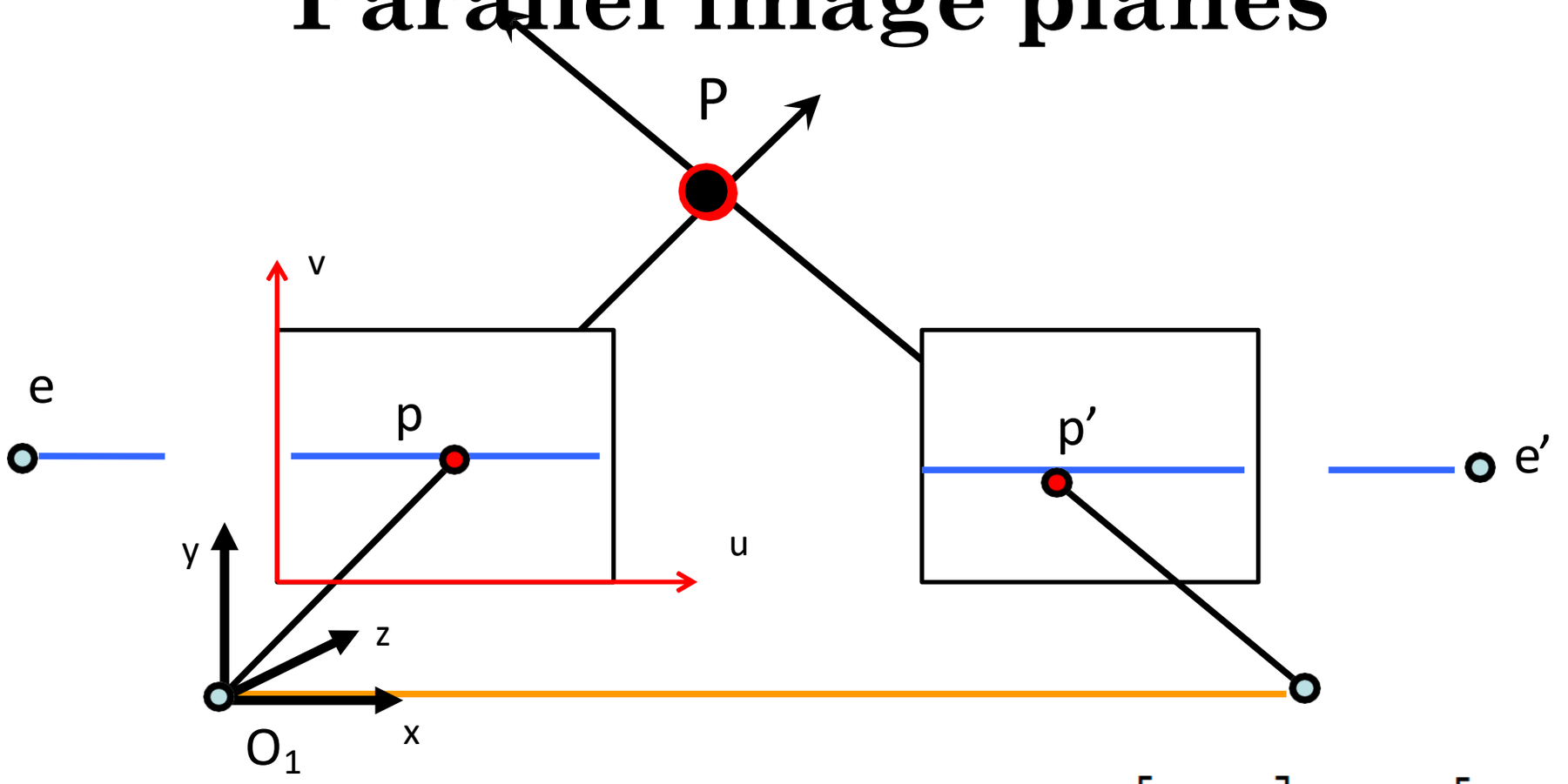


- Epipolar Plane
- Baseline
- Epipolar Lines

- Epipoles e, e’

= intersections of baseline with image planes
= projections of the other camera center

Parallel image planes



- Epipolar lines are horizontal
- Epipoles go to infinity
- v-coordinates are equal

$$p = \begin{bmatrix} p_u \\ p_v \\ 1 \end{bmatrix} \quad p' = \begin{bmatrix} p'_u \\ p_v \\ 1 \end{bmatrix}$$

Why are parallel images useful?

- Makes triangulation easy
- Makes the correspondence problem easier

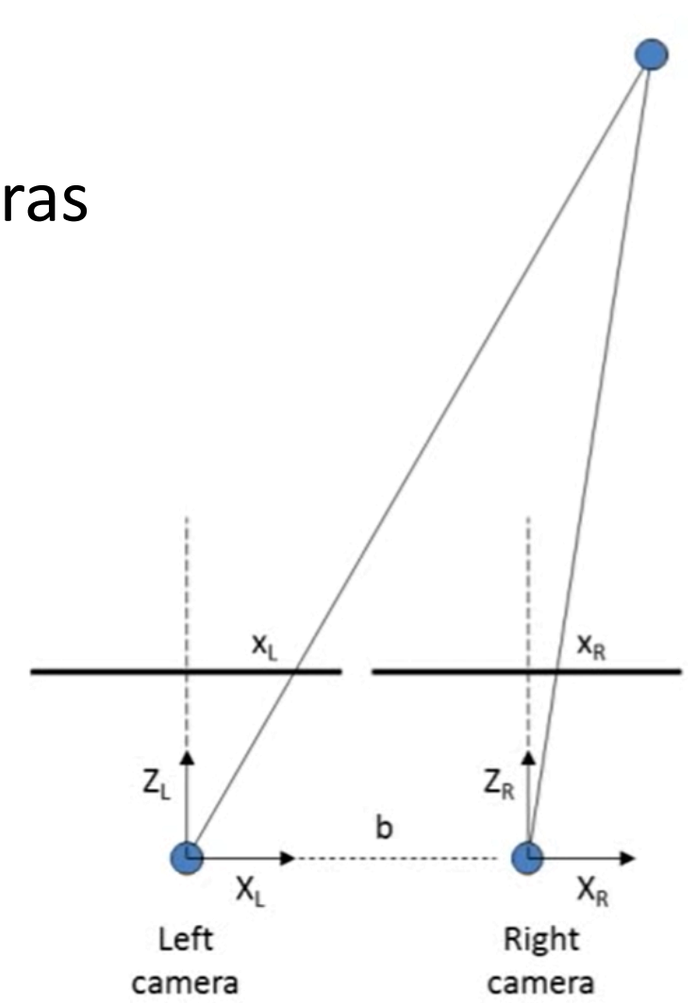
Stereo Principle

- Assume image planes are coplanar
- There is only a translation in the *X* direction between the two coordinates frames
- *b* is the baseline distance between the cameras

$$Disparity\ d = x_R - x_L$$

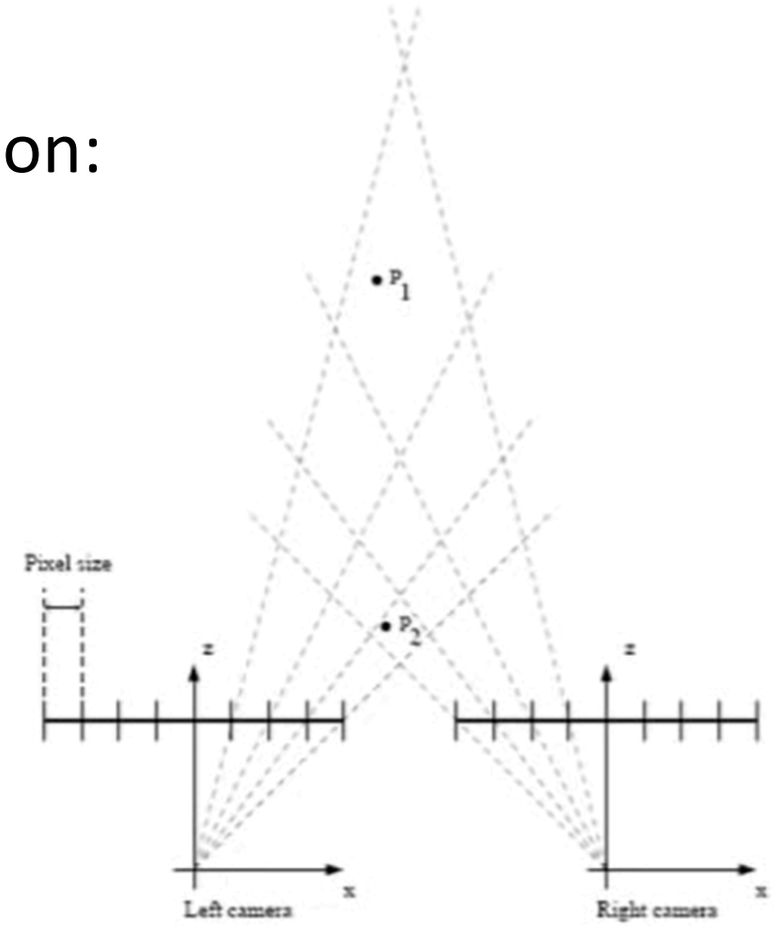
$$x_L = f\frac{X_L}{Z_L} \qquad x_R = f\frac{X_R}{Z_R} \qquad \text{Cameras are aligned} \begin{cases} Z_L = Z_R = Z \\ X_R = X_L + b \end{cases}$$

$$x_R = f\frac{(X_L+b)}{Z} \qquad d = f\frac{(X_L+b) - X_L}{Z} = f\frac{b}{Z} \qquad Z = f\frac{b}{d}$$



Reconstruction Error

- Given the uncertainty in pixel projection of the point, what is the error in depth?
- Obviously, the error in depth (ΔZ) will depend on:
  - Z, b, f
  - $\Delta x_L, \Delta x_R$
- Let's find the expected value of the error, and the variance of the error



From http://www.danet.dk/sensor_fusion

Reconstruction Error

- First, find the error in disparity Δd, from the error of locating the feature in each image, Δx_L and Δx_R
  - Taking the total derivative of each side

$$Disparity\ d = x_R - x_L$$

  - Assuming Δx_L and Δx_R are independent and zero mean

$$d(d) = d(x_R) - d(x_L) \longrightarrow \Delta d = \Delta x_R - \Delta x_L$$

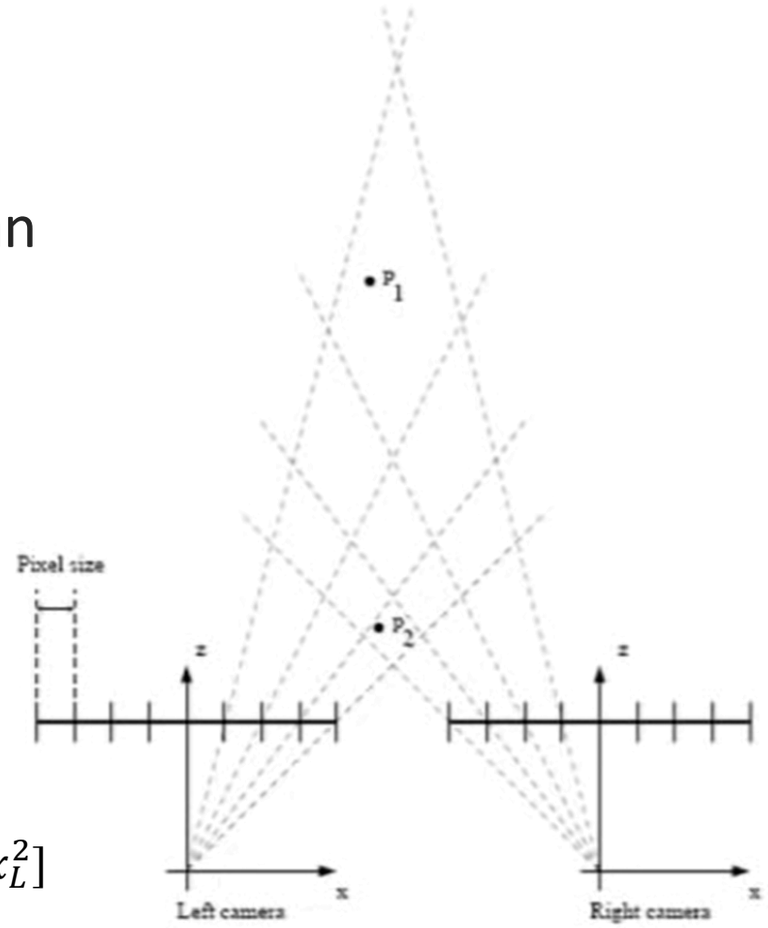
$$\mu = E[\Delta d] = E[\Delta x_R] - E[\Delta x_L] = 0$$

$$Var[\Delta d] = E[(\Delta d - \mu)^2] = E[(\Delta d)^2]$$

$$Var[\Delta d] = E[(\Delta x_R - \Delta x_L)^2] = E[\Delta x_R^2 - 2\Delta x_R \Delta x_L + \Delta x_L^2]$$

$$= E[\Delta x_R^2] - 2E[\Delta x_R \Delta x_L] + E[\Delta x_L^2] \qquad = E[\Delta x_R^2] + E[\Delta x_L^2]$$

$$\boxed{\sigma_d{}^2 = \sigma_R{}^2 + \sigma_L{}^2}$$



Reconstruction Error

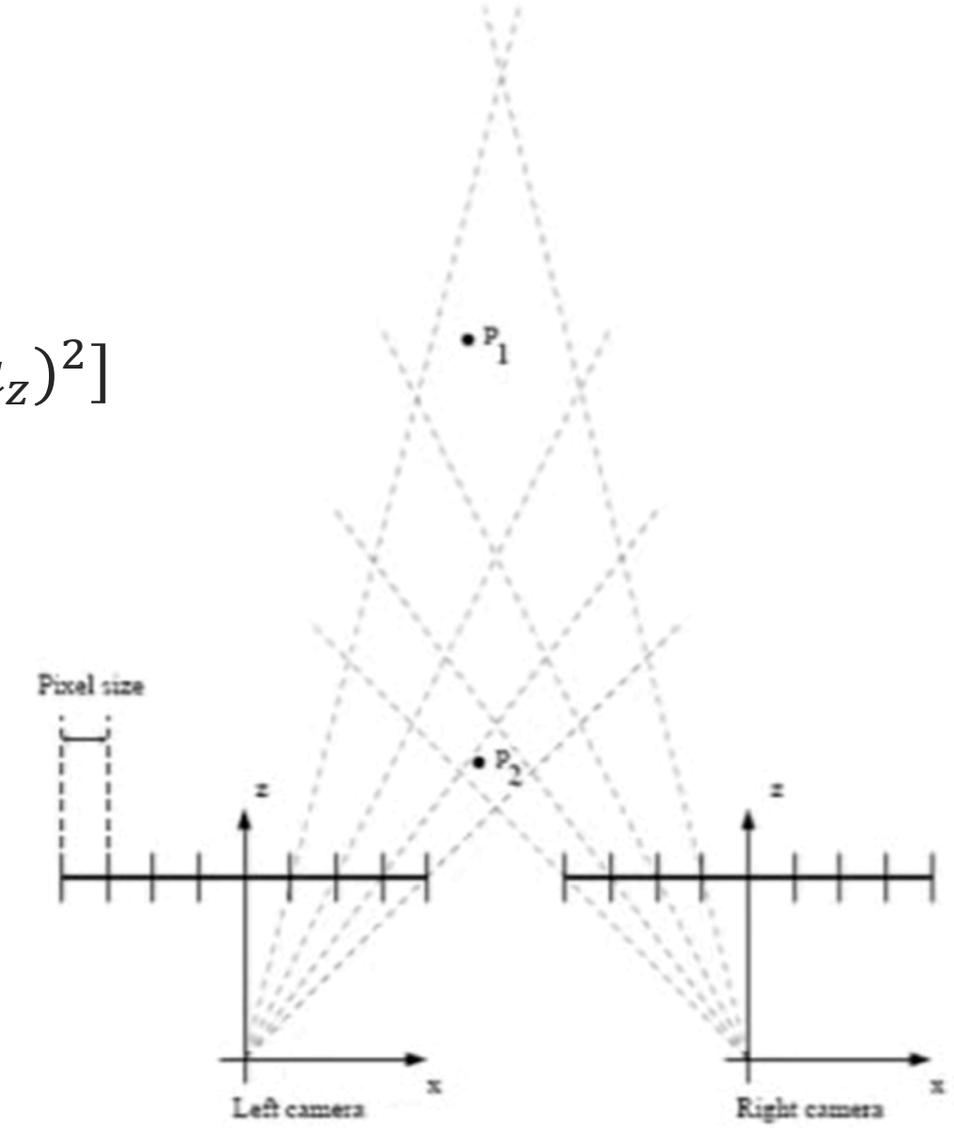
- Next, we take the total derivative of $Z = fb/d$
  - If the only uncertainty is in the disparity *d*
  - The mean error is $\mu_z = E[\Delta Z] = 0$

$$\Delta Z = f\frac{b}{d^2}(-\Delta d)$$

  - The variance of the error is ${\sigma_z}^2 = E[(\Delta Z - \mu_z)^2]$

$${\sigma_z}^2 = E[(\Delta Z)^2] = (f\frac{b}{d^2})^2 E[(\Delta d)^2] = (f\frac{b}{d^2})^2 {\sigma_d}^2$$

$$\sigma_z = f\frac{b}{d^2}\sigma_d = Z\frac{\sigma_d}{d} \qquad standard\ deviation$$



Example

- A stereo vision system estimated the disparity of a point as d=10 pixels
  - ❖ What is the depth (Z) of the point, if f = 500 pixels and b = 10 cm?

$$Z = f\frac{b}{d} = 500\ px\ \frac{10\ cm}{10\ px} = 500cm$$

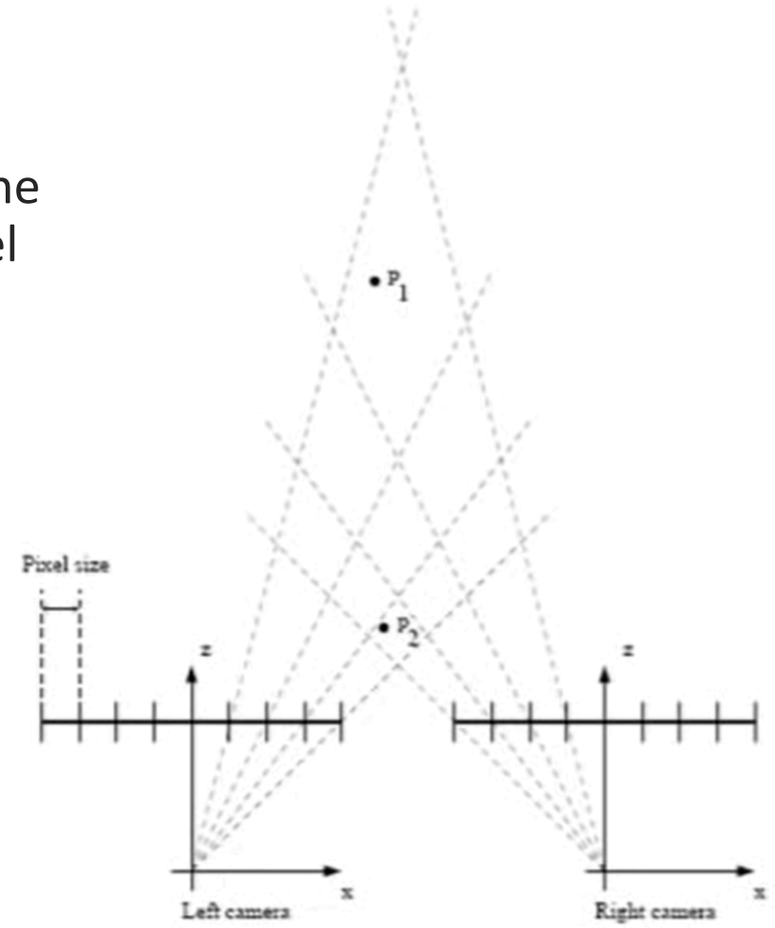
  - ❖ What is the uncertainty (standard deviation) of the depth, if the standard deviation of locating a feature in each image = 1 pixel

$$\sigma_z = Z\frac{\sigma_d}{d} \qquad \sigma_d{}^2 = \sigma_R{}^2 + \sigma_L{}^2 = 2 \qquad \sigma_d = \sqrt{2}$$

$$\sigma_z = 500cm\frac{\sqrt{2}\ px}{10\ px} \cong 70cm$$

❖ What if the disparity was d=1 pixel?

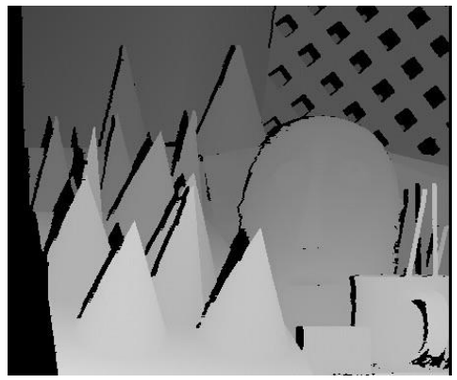
$$Z = 50m \qquad \sigma_z = 50m\frac{\sqrt{2}\ px}{1\ px} \cong 70\ m$$



Stereo Process

- Extract features from the left and right images
- Match the left and right images features, to get their disparity
  in position (the “correspondence problem”)
- Use stereo disparity to compute depth (the reconstruction problem)

http://vision.middlebury.edu/stereo/



Disparity map / depth map

- The correspondence problem is the most difficult

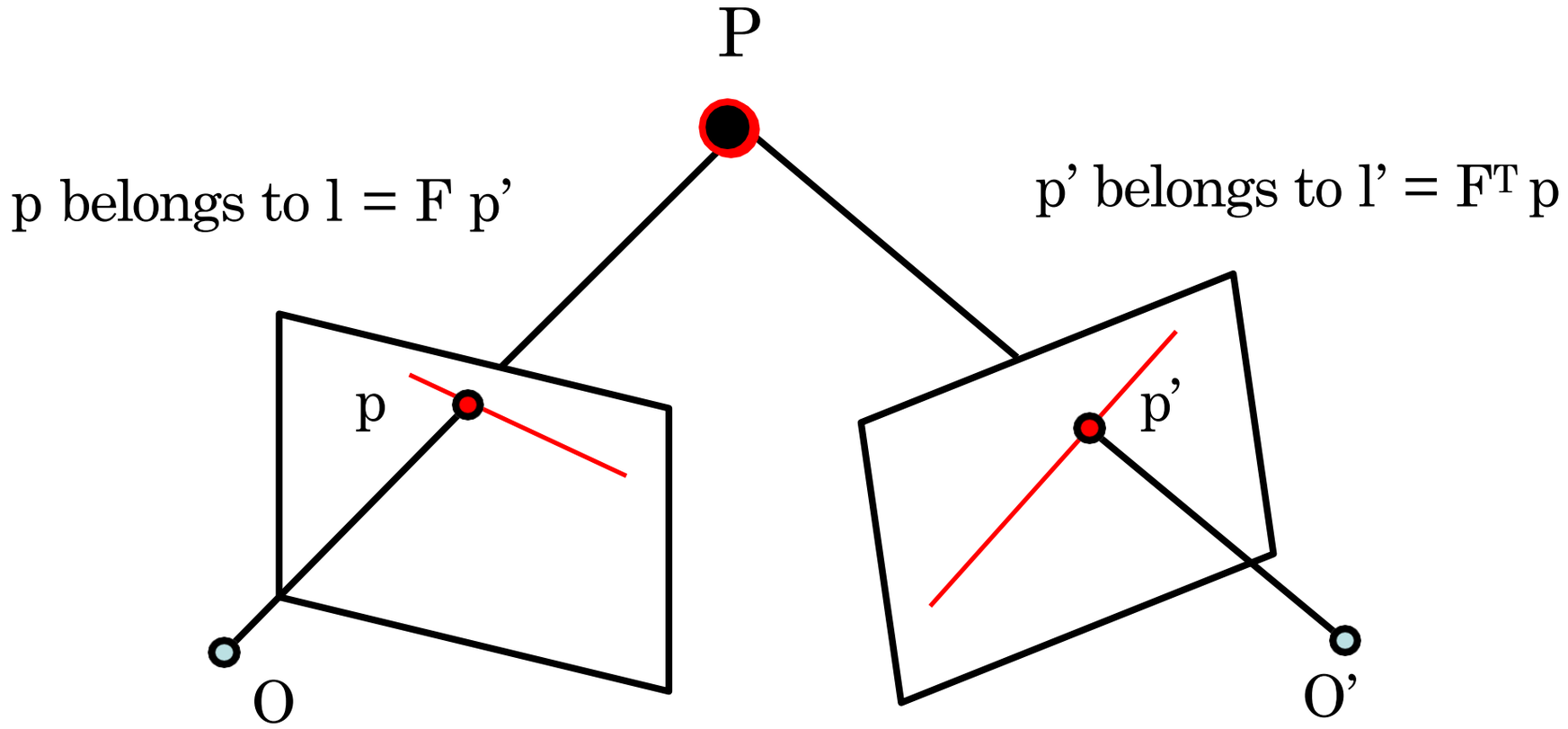
Why are parallel images useful?

- Makes triangulation easy
- Makes the correspondence problem easier

Correspondence problem

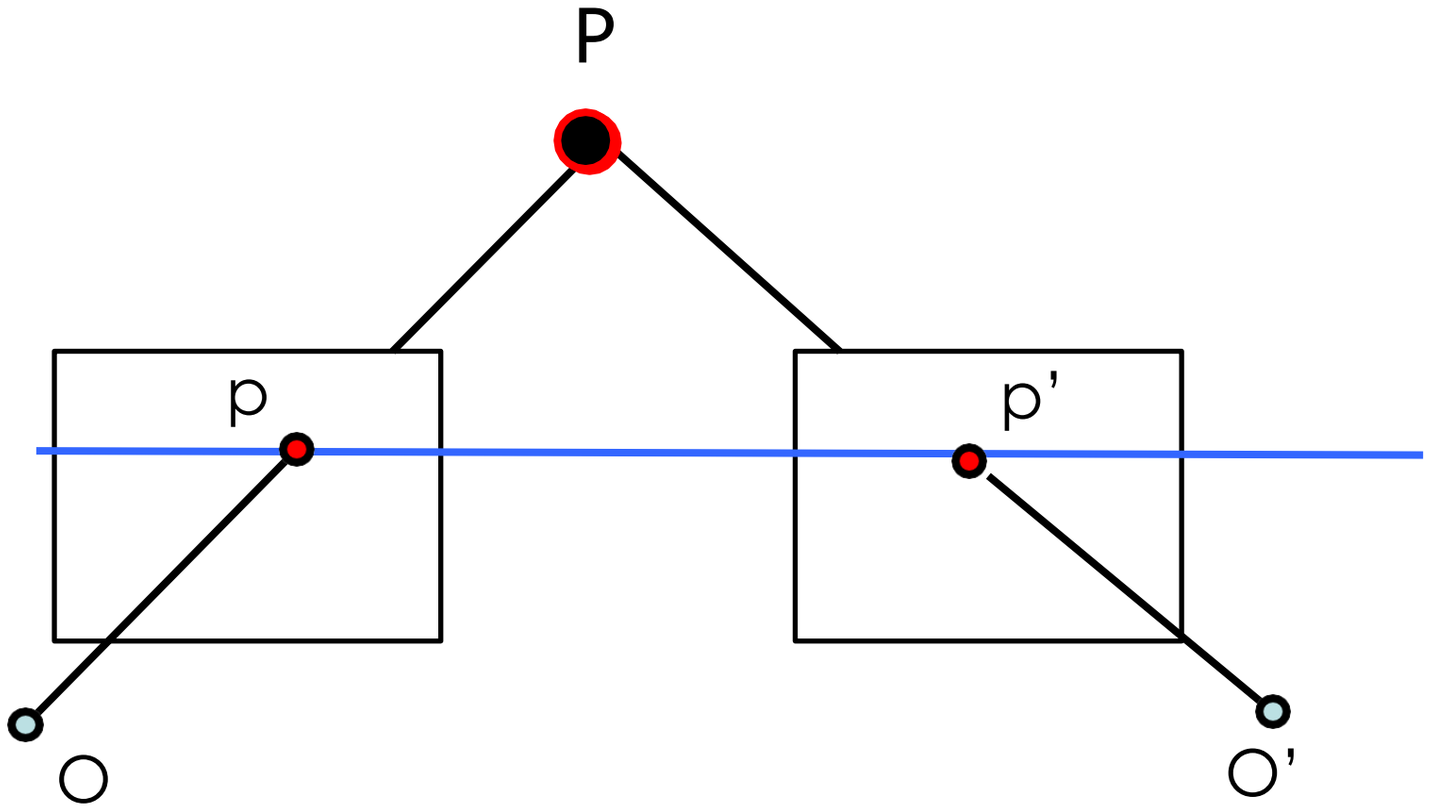
- For every point in the left image, there are many possible matches in the right image
- Locally, many points look similar -> matches are ambiguous
- We can use the (known) geometry of the cameras to help limit the search for matches
- The most important constraint is epipolar constraint
  - We can limit the search for a match to be along a certain line in the other image

Correspondence problem



Given a point in 3D, discover corresponding observations
in left and right images [also called binocular fusion problem]

Correspondence problem



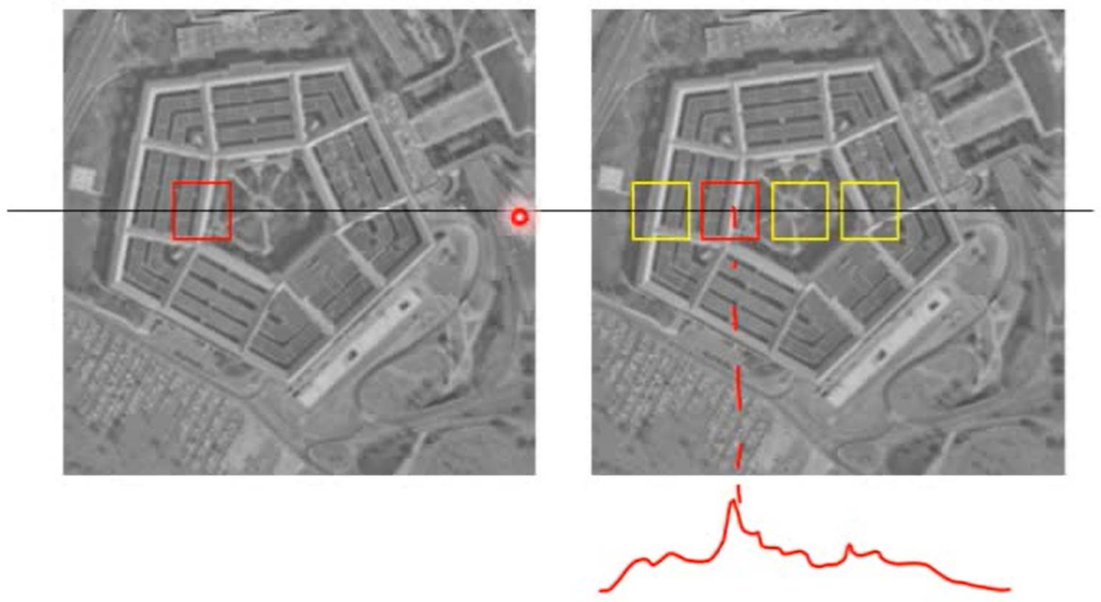
When images are rectified, this problem is much easier!

Correspondence problem

- Match points on local similarity between images
- Two general approaches
- Correlation-based approaches
  - Matches image patches using correlation
  - Assumes only a translational difference between the two local patches (no rotation, or differences in appearance due to perspective)
  - A good assumption if patch covers a single surface, and surface is far away compared to baseline between cameras
  - Works well for scenes with lots of texture
- Feature-based approaches
  - Matches edges, lines, or corners
  - Gives a sparse reconstruction
  - May be better for scenes with little texture

Correlation approach

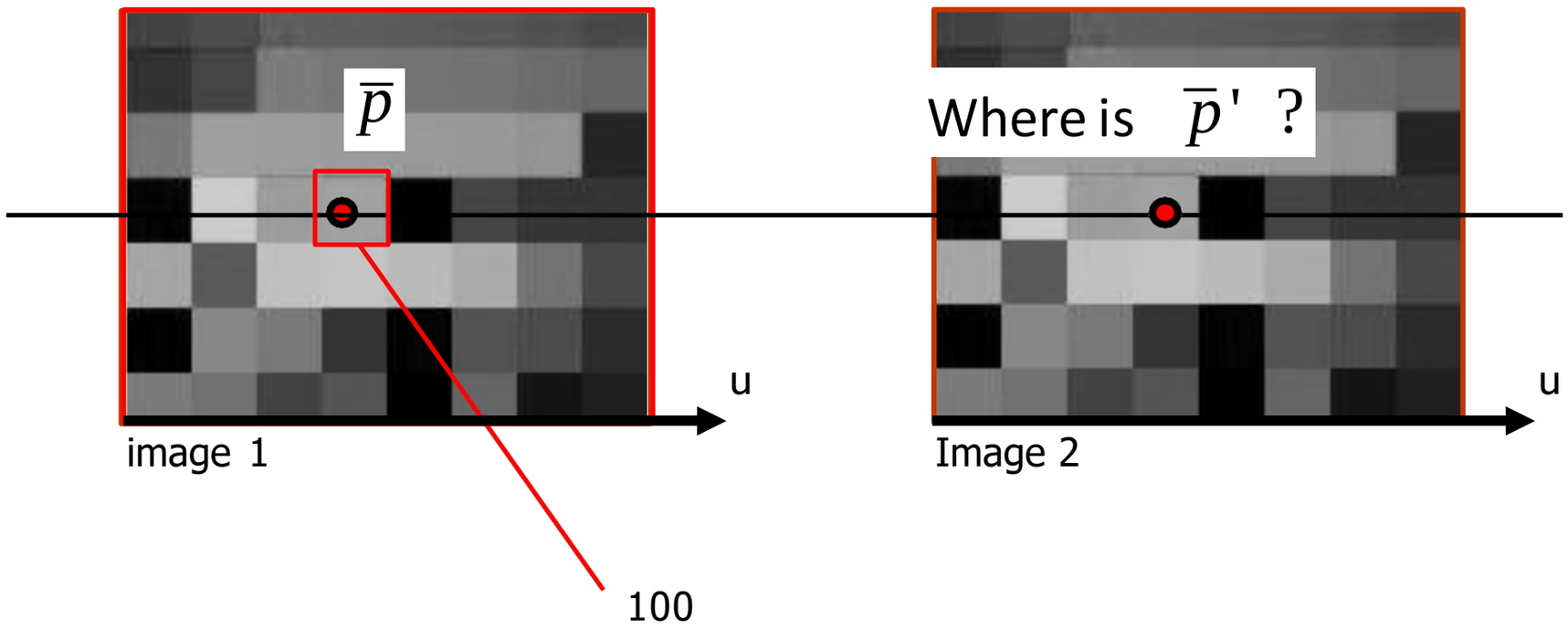
- Select a range of disparities to search
- For each patch in the left image, compute cross correlation score for every point along the epipolar line
- Find maximum correlation score along that line



Correlation Methods (1970--)

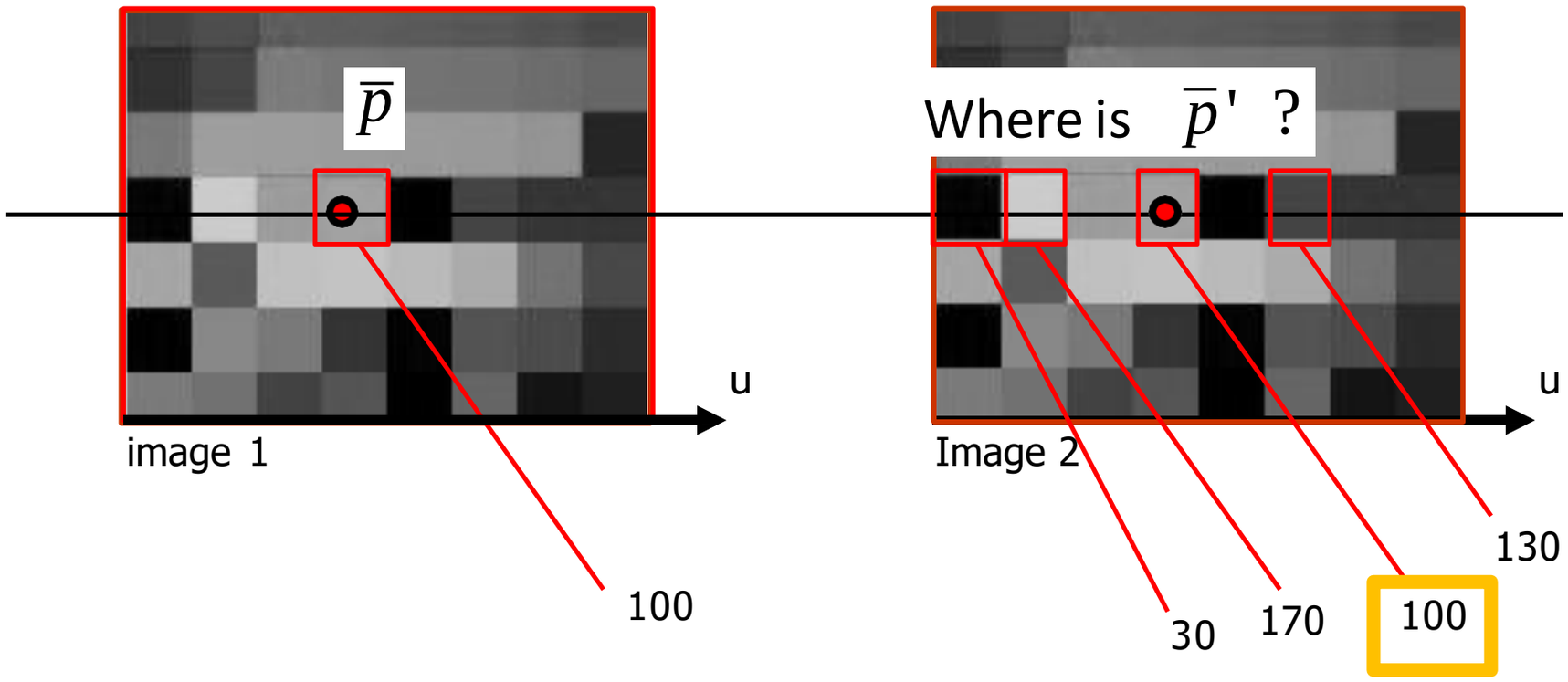
$$\overline{p} = \begin{bmatrix} \overline{u} \\ \overline{v} \\ 1 \end{bmatrix} \qquad \overline{p}' = \begin{bmatrix} \overline{u}' \\ \overline{v} \\ 1 \end{bmatrix}$$

Correlation Methods (1970--)



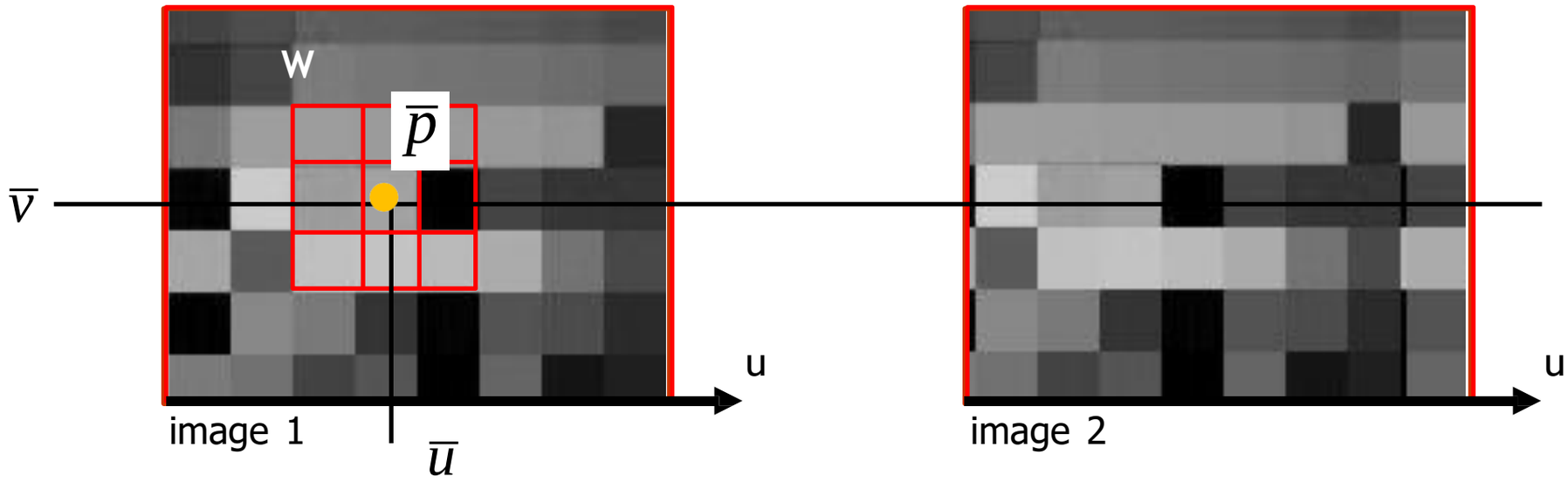
$$\overline{p} = \begin{bmatrix} \overline{u} \\ \overline{v} \\ 1 \end{bmatrix} \qquad \overline{p}' = \begin{bmatrix} \overline{u}' \\ \overline{v} \\ 1 \end{bmatrix}$$

Correlation Methods (1970--)



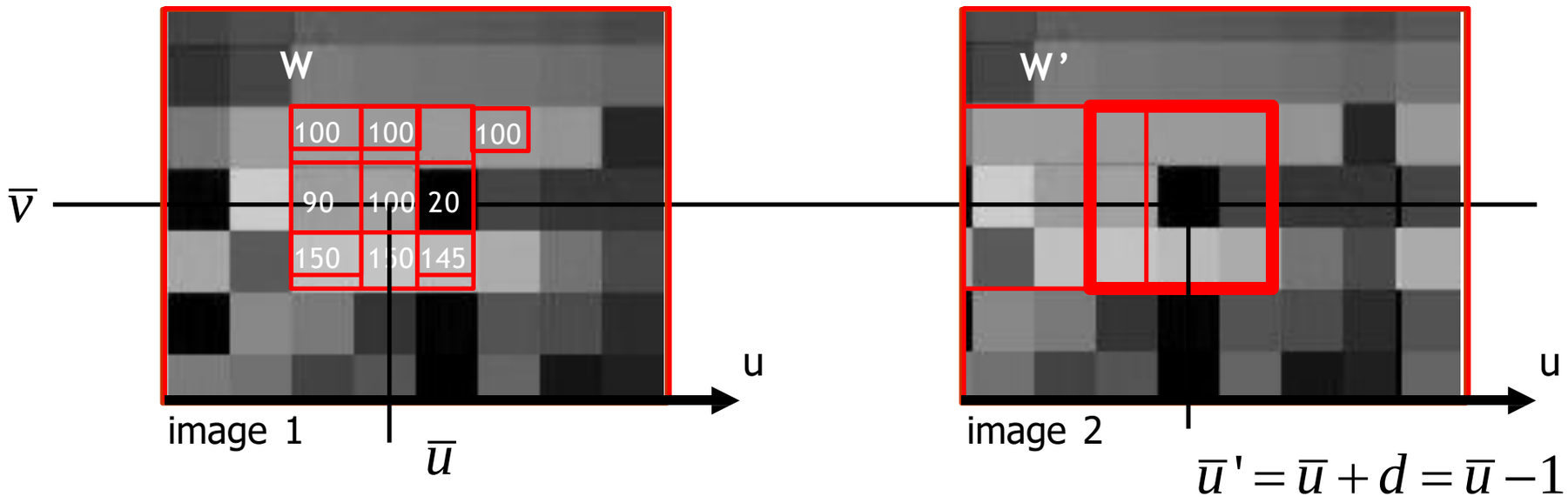
What's the problem with this?

Window-based correlation



- Pick up a window **W** around $\overline{p} = (\overline{u}, \overline{v})$

Window-based correlation



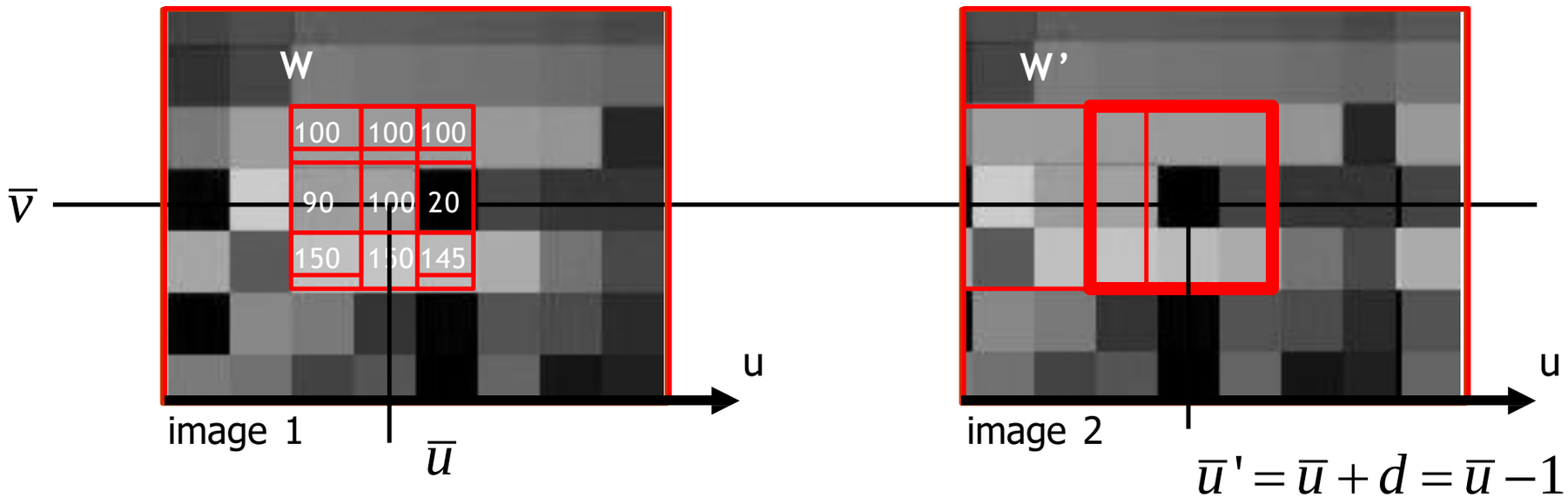
Example: **W** is a 3x3 window in red

w is a 9x1 vector

w = [100, 100, 100, 90, 100, 20, 150, 150, 145]^T

- Pick up a window **W** around $p = (u, \overline{v})$
- Build vector **w**
- Slide the window **W** along v = $\overline{V}$ in image 2 and compute **w'** (u) for each u
- Compute the dot product $\mathbf{w}^T\ \mathbf{w'}$(u) for each u and retain the max value

Window-based correlation



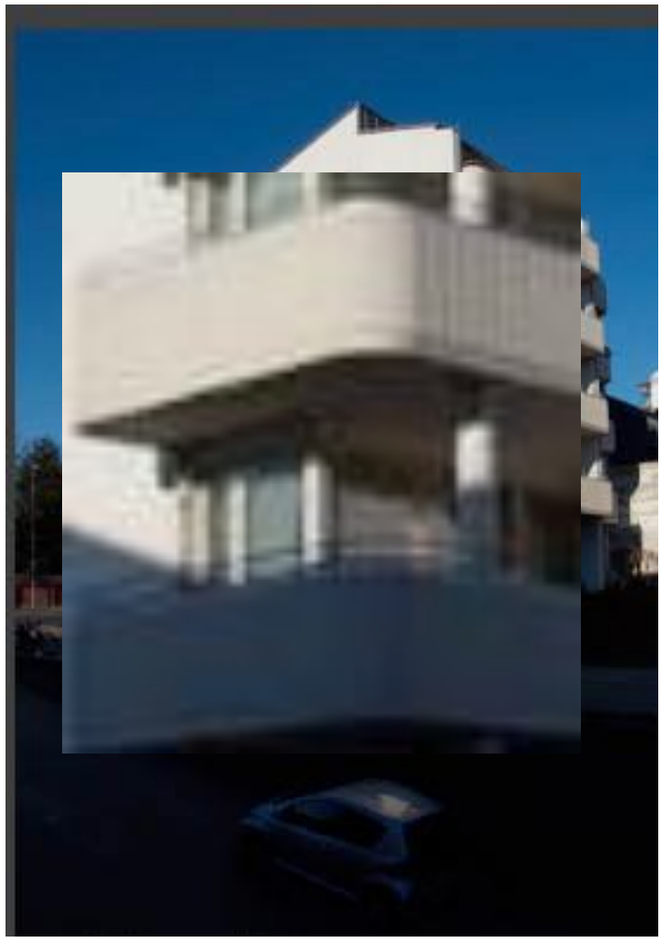
Example: **W** is a 3x3 window in red

w is a 9x1 vector

w = [100, 100, 100, 90, 100, 20, 150, 150, 145]^T

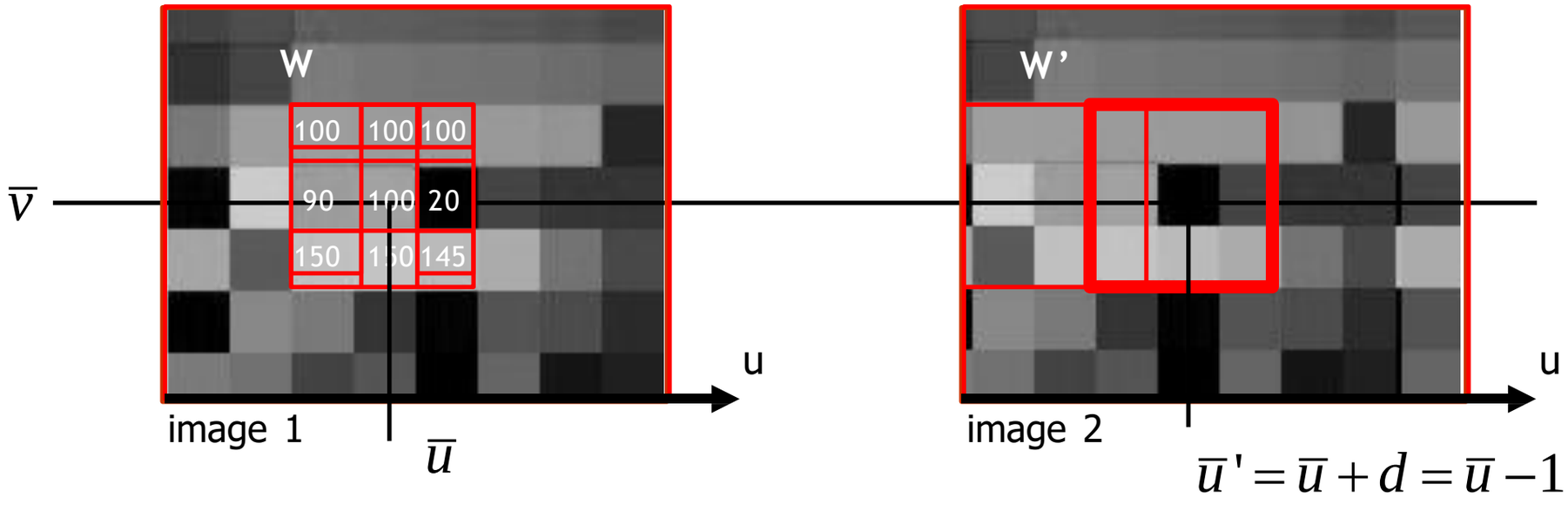
What's the problem with this?

Changes of brightness/exposure



Changes in the mean and the variance of intensity values in corresponding windows!

Normalized cross-correlation



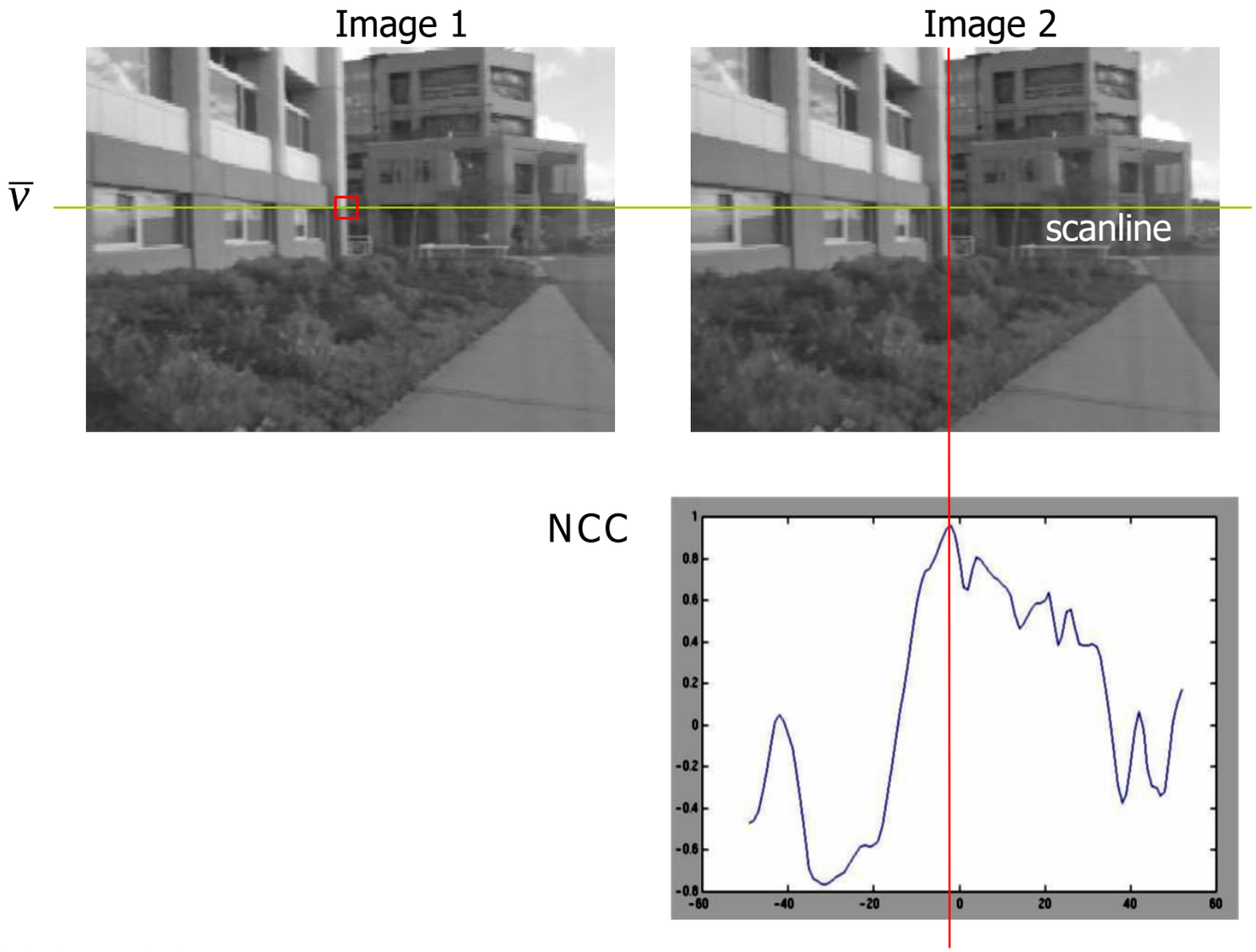
Find u that maximizes:

$$\frac{(w-\overline{w})^T (w'(u)-\overline{w}')}{\left\|(w-\overline{w})\right\| \left\|(w'(u)-\overline{w}')\right\|} \quad \text{[Eq. 2]}$$

$\overline{w}$ = mean value within **W** located at u^{bar} in image 1

$\overline{w}'(u)$ = mean value within **W** located at u in image 2

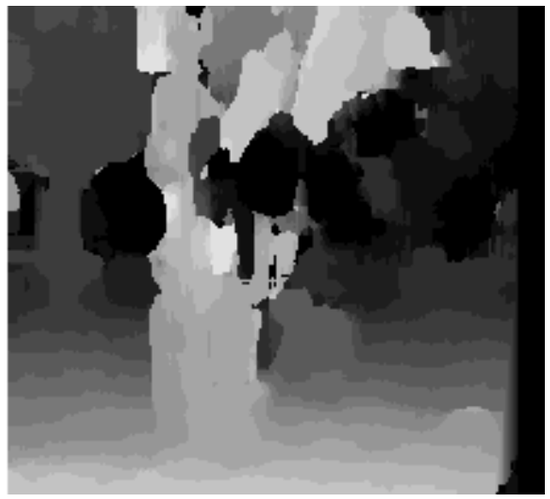
Example



Credit slide S. Lazebnik

Effect of the window's size

Window size = 3



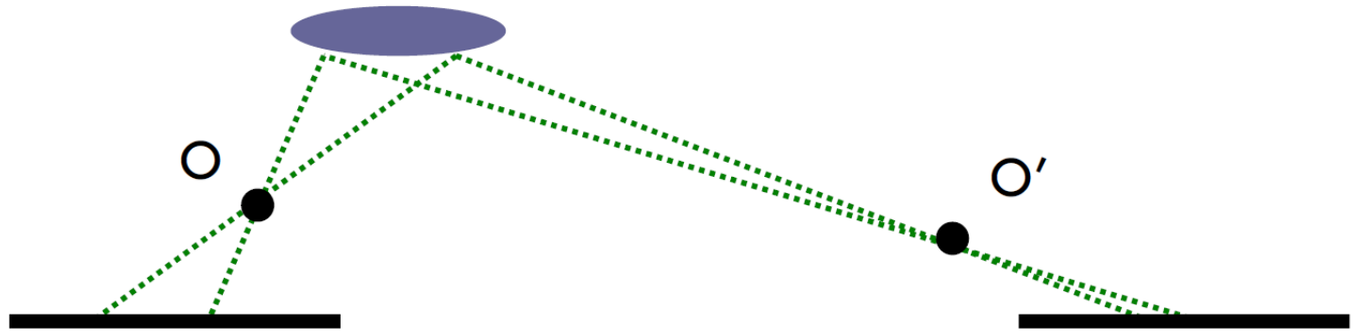
Window size = 20

- Smaller window
  - More detail
  - More noise
- Larger window
  - Smoother disparity maps
  - Less prone to noise

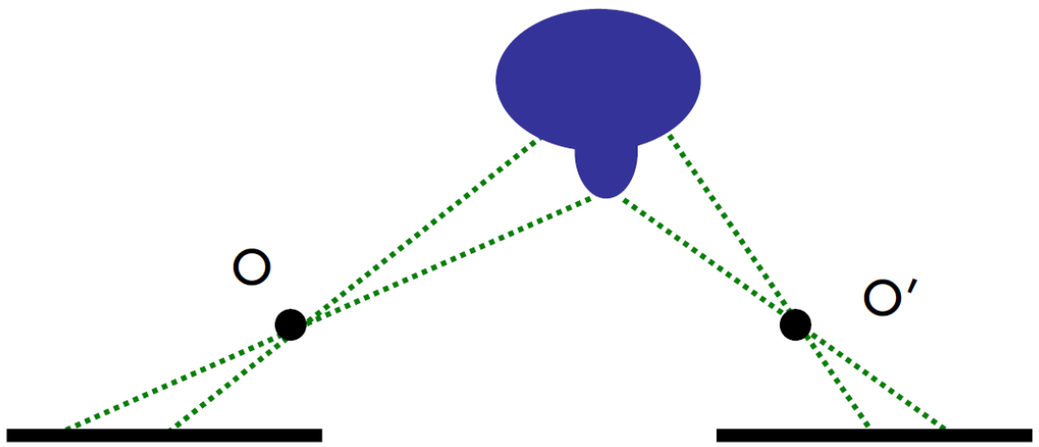
Credit slide S. Lazebnik

Issues

- Fore shortening effect



- Occlusions

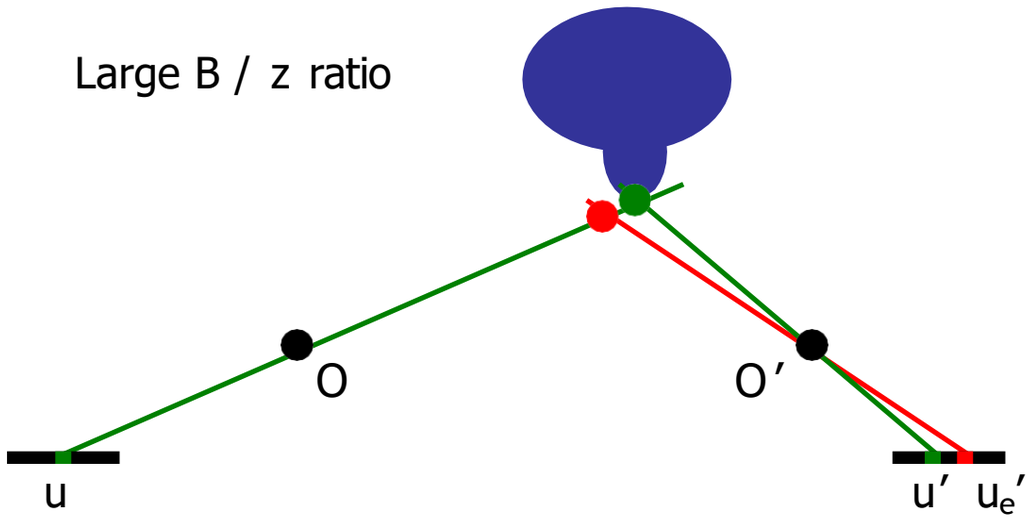


Base line trade-off

- To reduce the effect of foreshortening and occlusions, it is desirable to have a small B / z ratio!
- However, when B/z is small, small errors in measurements imply large errors in estimating depth

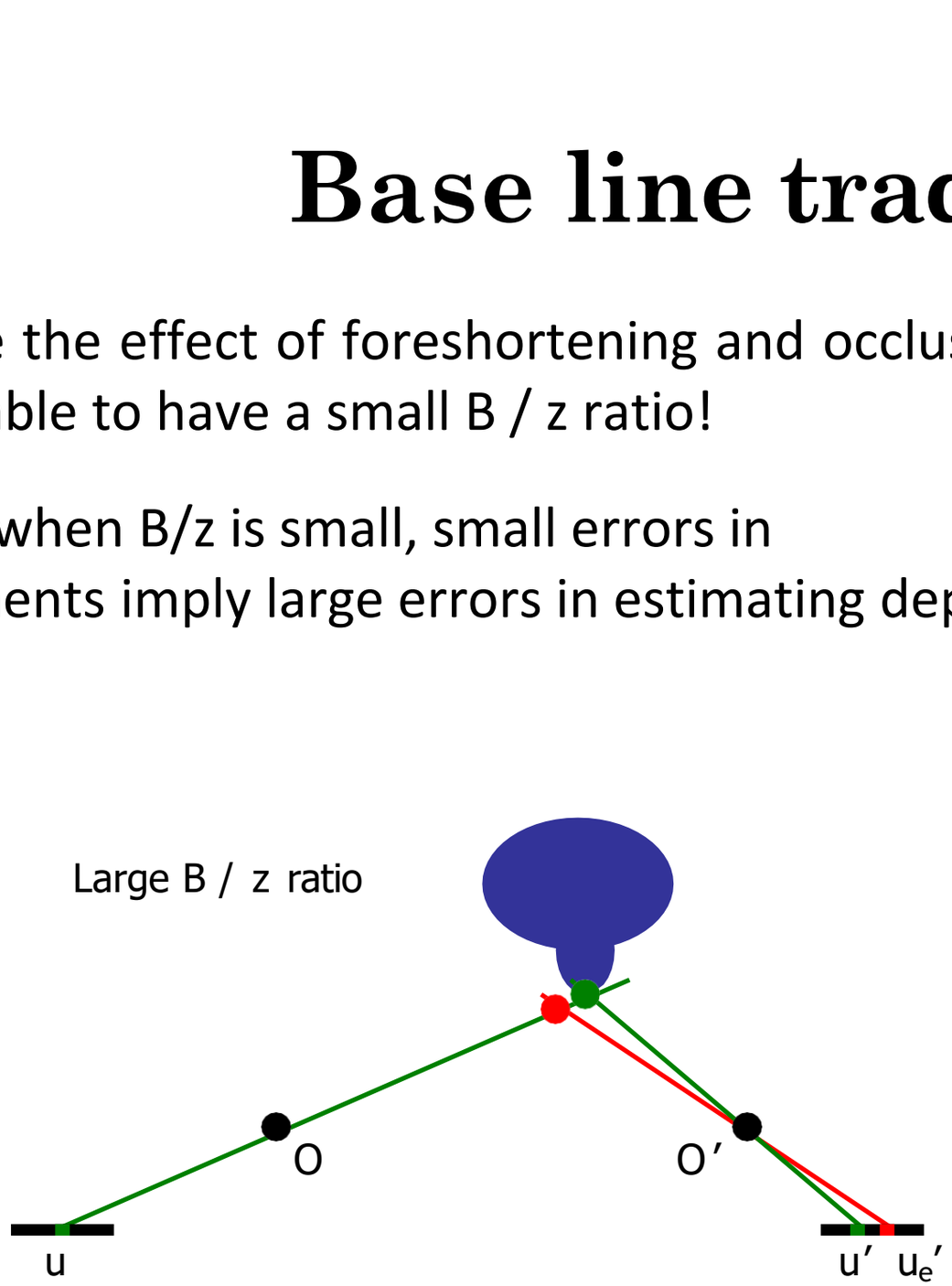
Base line trade-off

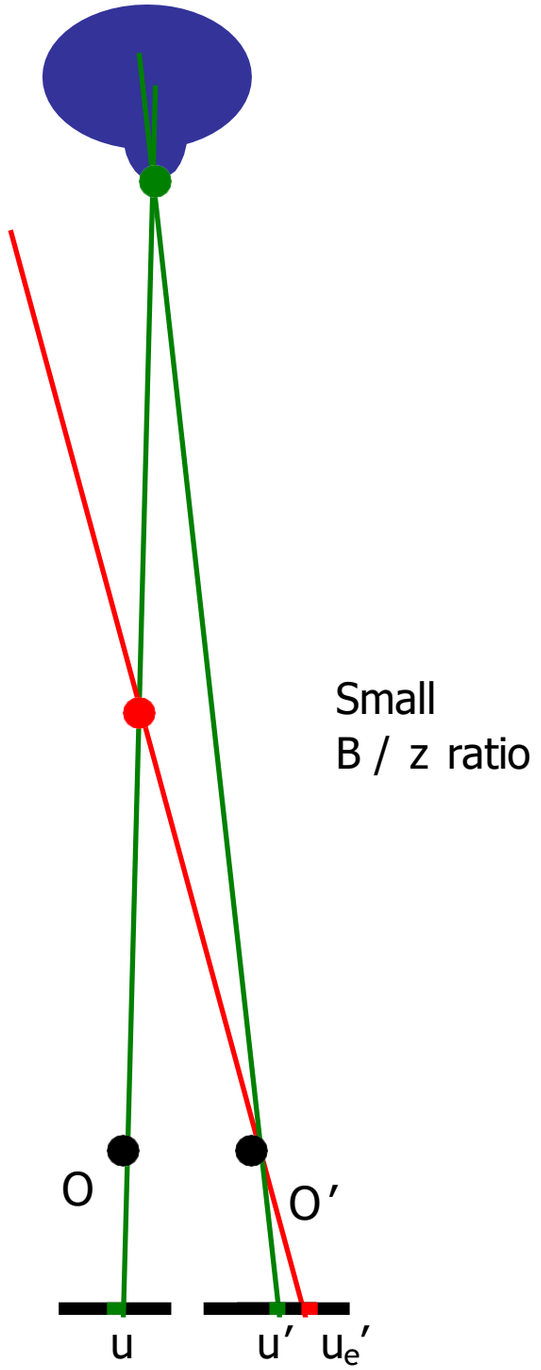
- To reduce the effect of foreshortening and occlusions, it is desirable to have a small B / z ratio!
- However, when B/z is small, small errors in measurements imply large errors in estimating depth



Base line trade-off

- To reduce the effect of foreshortening and occlusions, it is desirable to have a small B / z ratio!
- However, when B/z is small, small errors in measurements imply large errors in estimating depth





More issues!

- Homogeneous regions

More issues!

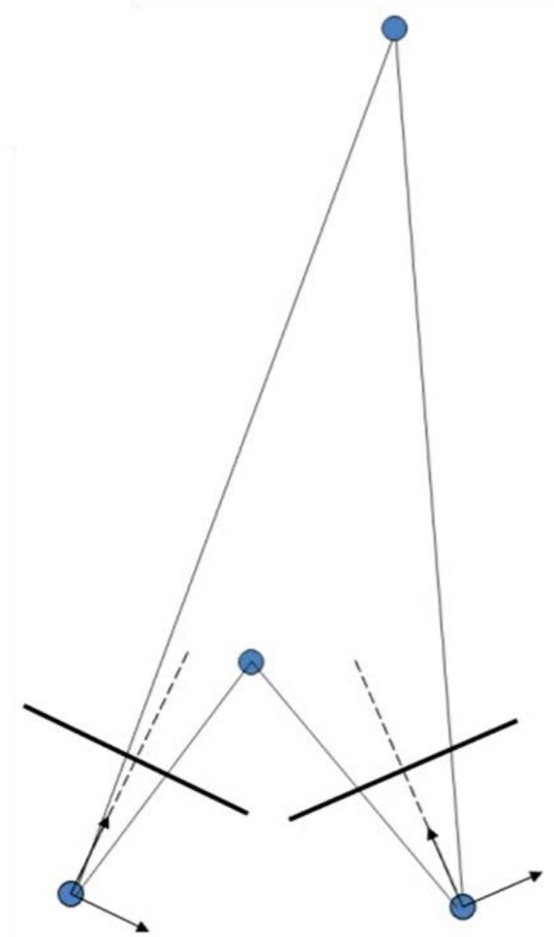
- Repetitive patterns

Correspondence problem

- Even using the epipolar constraint, there are many possible matches
- Worst case scenarios
  - A whiteboard (no feature)
  - A checkered wallpaper (ambiguous matches)
- The problem is under-constrained
- To solve, we need to impose assumptions about the real world
  - Disparity limits
  - Appearance
  - Uniqueness
  - Ordering
  - Smoothness

Disparity limit

- Assume that valid disparities are within certain limits
  - Constrains search
- Why usually true?



Appearance

- Assume features should have similar appearance in the left and right images

http://vision.middlebury.edu/stereo/data/scenes2003/

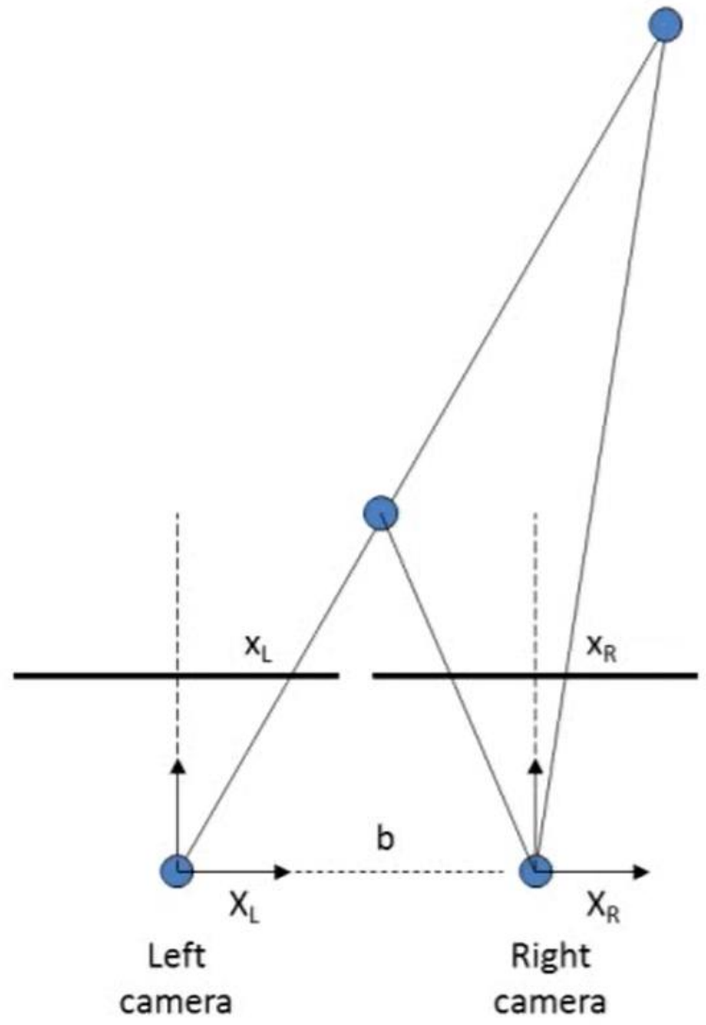
- Why usually true?

- When is it violated?

Uniqueness

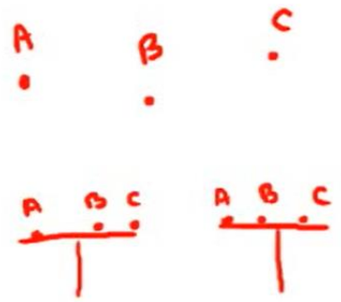
- Assume features should have similar appearance in the left and right images

- Why usually true?



Ordering

- Assume features should be in the same left to right order in each image



- Why usually true?

- When is it violated?

Smoothness

- Assume objects have mostly smooth surfaces, meaning that disparities should vary smoothly (e.g., have a low second derivative)

- Why usually true?

- When is it violated?