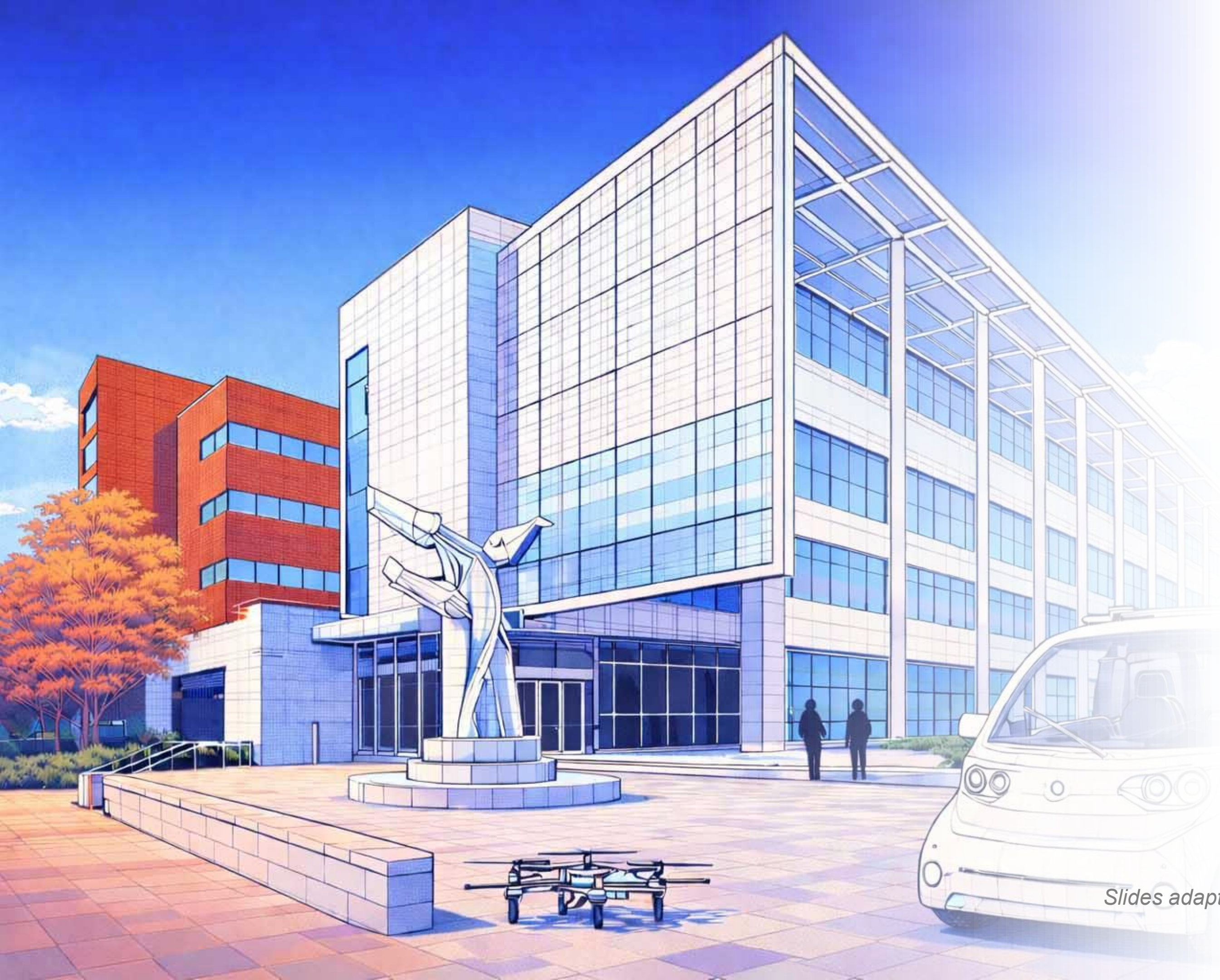


ECE 484
Lecture 6
3D Vision

Prof. Huan Zhang

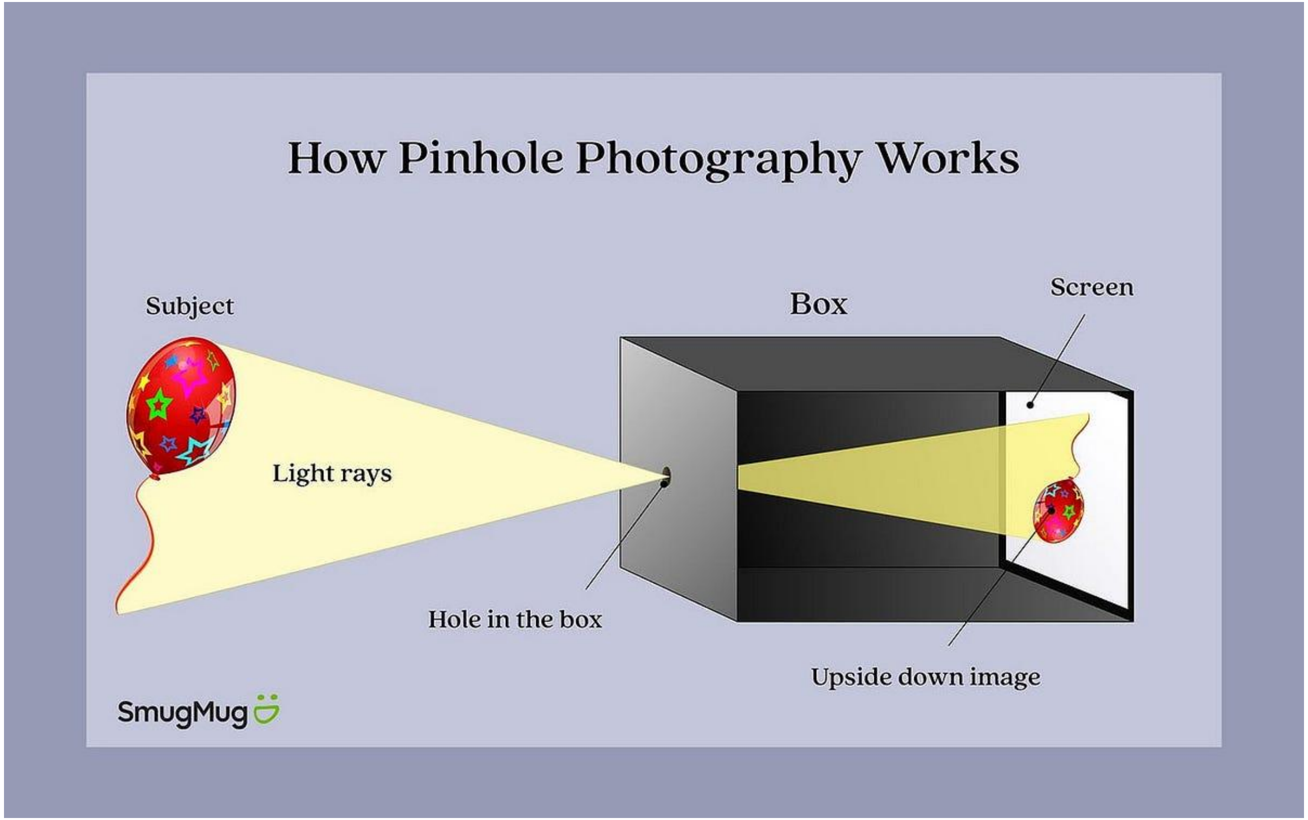
Fall 2026

Slides adapted in part from materials by Prof. Sayan Mitra.

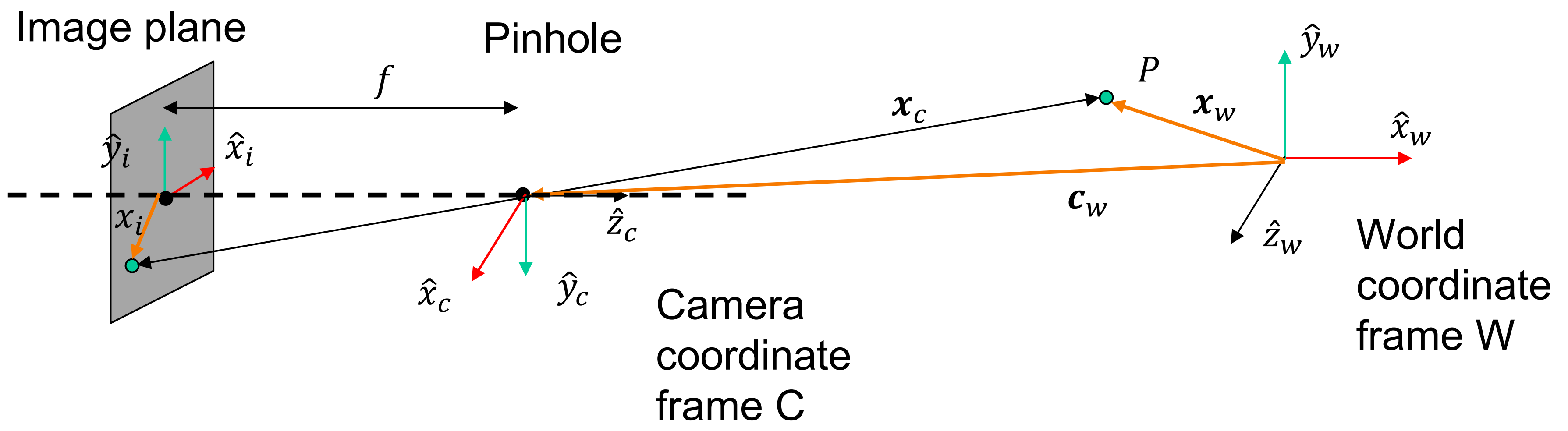
Cameras now span 6+ orders of magnitude in scale and 9+ orders in time:

microns (on-chip) to meters (space optics), microseconds to hours (Earth-observing revisits)

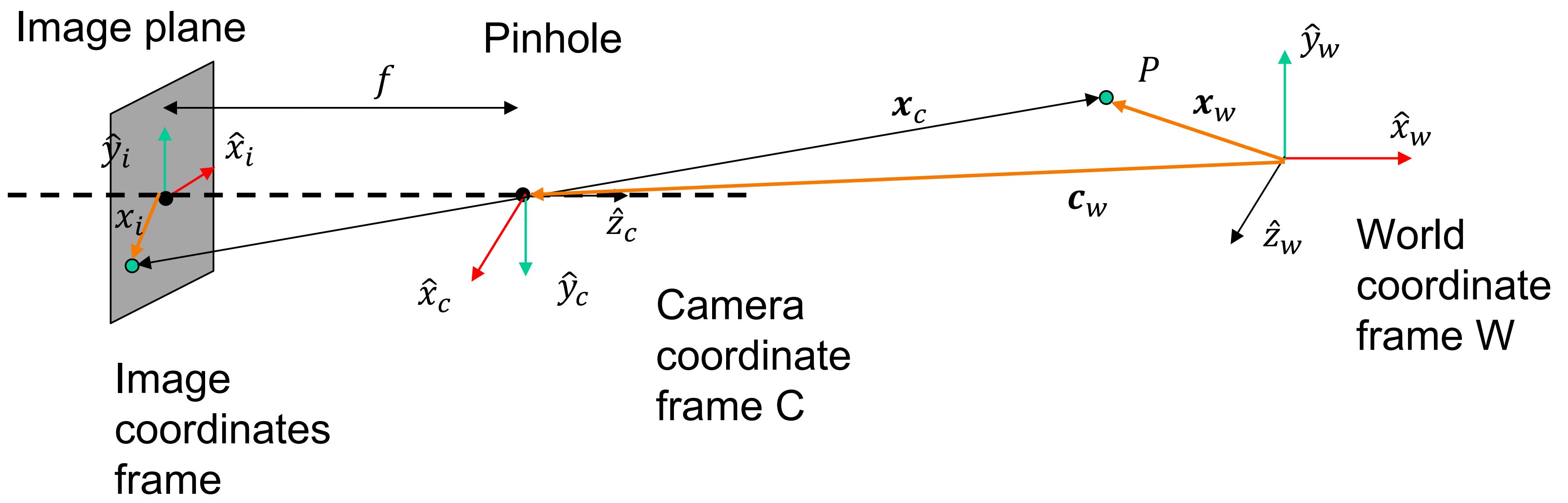
Mathematical Model of Camera: the Pinhole Camera



The forward imaging model: from a 3-D point to a 2-D image point



The forward imaging model has two stages



$$\boldsymbol{x}_i = \begin{bmatrix} x_i \\ y_i \end{bmatrix} \xleftarrow{\text{3D-2D}} \boldsymbol{x}_c = \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} \xleftarrow{\text{3D-3D}} \boldsymbol{x}_w = \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix}$$

Slide ideas from First Principles of Computer Vision by Shree Nayar

Coordinate frames: the same point, different numbers

A **frame** A = an origin O_A + three orthonormal, right-handed axes $\hat{x}_A, \hat{y}_A, \hat{z}_A$

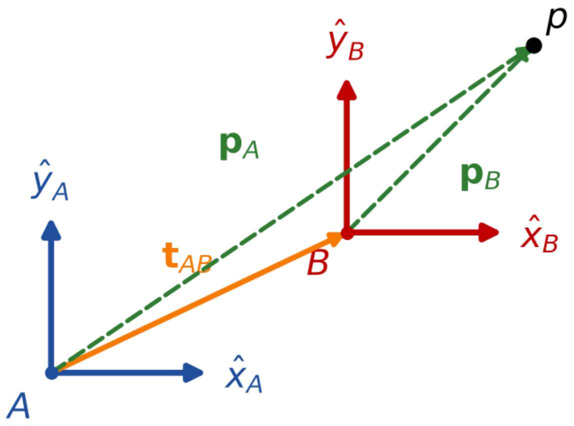
A point p has coordinates $\boldsymbol{p}_A = (x_A, y_A, z_A)^T$ in frame A: $p = O_A + x_A\hat{x}_A + y_A\hat{y}_A + z_A\hat{z}_A$

The same point has different coordinates $\boldsymbol{p}_B$ in another frame B — the vehicle, the camera, the map all have their own frame

Translation only (axes of A and B parallel): let $\boldsymbol{t}_{AB}$ = position of O_B expressed in A. Then

- $\boldsymbol{p}_A = \boldsymbol{p}_B + \boldsymbol{t}_{AB}$, inverse $\boldsymbol{p}_B = \boldsymbol{p}_A - \boldsymbol{t}_{AB}$ (= $\boldsymbol{p}_A + \boldsymbol{t}_{BA}$, with $\boldsymbol{t}_{BA} = -\boldsymbol{t}_{AB}$)

Subscript convention: $\boldsymbol{p}_A$ = "coordinates in A"; $\boldsymbol{t}_{AB}, R_{AB}, T_{AB}$ = "takes B-coordinates to A-coordinates"



Rotation in 2-D: build the matrix from the axes of the rotated frame

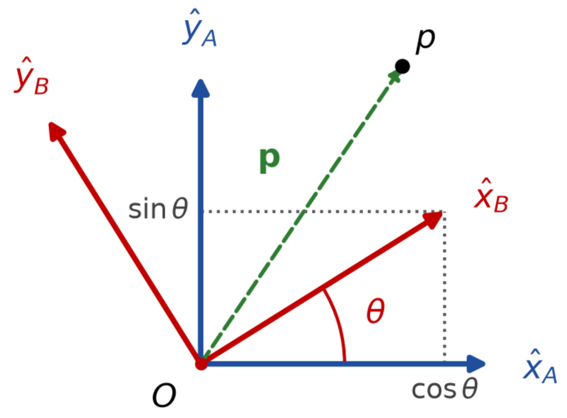
Frame B has the same origin as A, rotated counter-clockwise by θ

Axes of B written in A-coordinates: $\hat{x}_B = \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix}$, $\hat{y}_B = \begin{bmatrix} -\sin\theta \\ \cos\theta \end{bmatrix}$

A point with B-coordinates $\boldsymbol{p}_B = (x_B, y_B)^T$ is $p = O + x_B\hat{x}_B + y_B\hat{y}_B$, so in A-coordinates

- $\boldsymbol{p}_A = x_B\,\hat{x}_B + y_B\,\hat{y}_B = [\hat{x}_B \quad \hat{y}_B]\boldsymbol{p}_B = R_{AB}(\theta)\,\boldsymbol{p}_B$
- $R_{AB}(\theta) = [\hat{x}_B \quad \hat{y}_B] = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$

Rotation matrix R_{AB}: its **columns are the axes of** B **expressed in** A; it maps B-coordinates to A-coordinates



Q: What is R_{BA}, the matrix that takes A-coordinates to B-coordinates?

Rotate back by $-\theta$: $R_{BA} = R_{AB}(-\theta) = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} = R_{AB}(\theta)^T$ — the inverse of a rotation is its transpose (next slide: why, in any dimension).

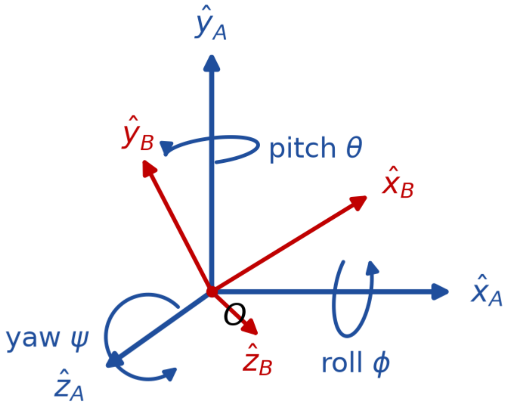
Rotation in 3-D: yaw, pitch and roll

$R_{AB} = [\hat{x}_B \quad \hat{y}_B \quad \hat{z}_B] = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$: column j = axis j of B in A-coordinates

$\boldsymbol{p}_A = R_{AB}\,\boldsymbol{p}_B$; row i = axis i of A in B-coordinates (since $R_{BA} = R_{AB}^T$)

Any rotation = **yaw** ψ about $\hat{z}$, **pitch** θ about $\hat{y}$, **roll** ϕ about $\hat{x}$ (figure) — each a 2-D rotation in one coordinate plane:

- $R = R_z(\psi)\,R_y(\theta)\,R_x(\phi)$ (acting on $\boldsymbol{p}_B$ right to left: roll, then pitch, then yaw)



$$R_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix}, \quad R_y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}, \quad R_z(\psi) = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R = R_z(\psi)\,R_y(\theta)\,R_x(\phi) = \begin{bmatrix} \cos\psi\cos\theta & \cos\psi\sin\theta\sin\phi - \sin\psi\cos\phi & \cos\psi\sin\theta\cos\phi + \sin\psi\sin\phi \\ \sin\psi\cos\theta & \sin\psi\sin\theta\sin\phi + \cos\psi\cos\phi & \sin\psi\sin\theta\cos\phi - \cos\psi\sin\phi \\ -\sin\theta & \cos\theta\sin\phi & \cos\theta\cos\phi \end{bmatrix}$$

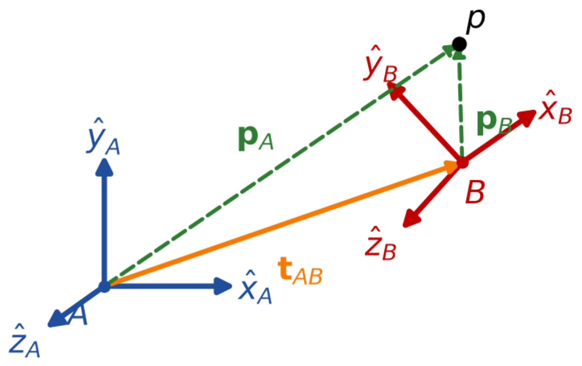
Checks: $\theta = \psi = 0$ leaves $R = R_x(\phi)$; each factor has $\det = +1$, hence $\det R = +1$; three angles for the 3 degrees of freedom of $SO(3)$.
Order matters: $R_x(\phi)\,R_z(\psi) \neq R_z(\psi)\,R_x(\phi)$.

Rotation + translation: the rigid transform

Frame B is rotated by R_{AB} and its origin sits at $\boldsymbol{t}_{AB}$ (in A-coordinates). Then

- $\boldsymbol{p}_A = R_{AB}\,\boldsymbol{p}_B + \boldsymbol{t}_{AB}$ — first rotate B-coordinates into the axes of A, then add the origin offset

Inverse: $\boldsymbol{p}_B = R_{AB}^T(\boldsymbol{p}_A - \boldsymbol{t}_{AB})$, i.e. $R_{BA} = R_{AB}^T$ and $\boldsymbol{t}_{BA} = -R_{AB}^T\,\boldsymbol{t}_{AB}$



A rigid transform preserves distances and angles (6 degrees of freedom: 3 rotation + 3 translation) — exactly what "moving a camera" can do

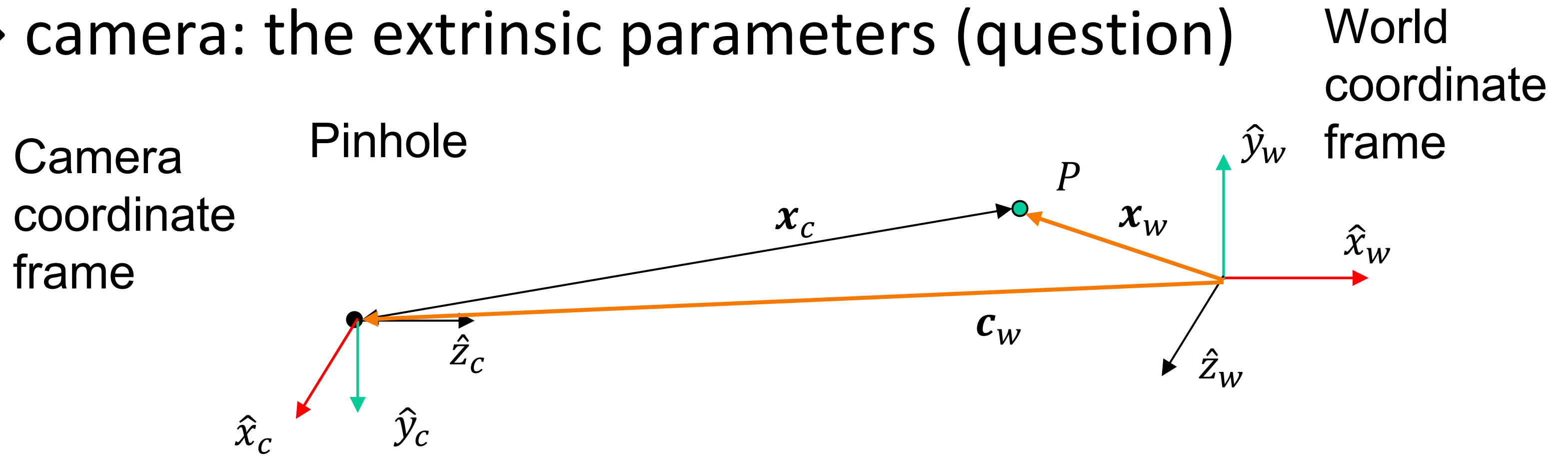
Homogeneous coordinates: one 4×4 matrix

Q: Can $\boldsymbol{p}_A = R_{AB}\,\boldsymbol{p}_B + \boldsymbol{t}_{AB}$ be written as a single matrix product $M\boldsymbol{p}_B$?

No. A linear map sends 0 to 0, but $\boldsymbol{p}_B = 0$ (the origin of B) must map to $\boldsymbol{t}_{AB} \neq 0$. The map is **affine**, not linear.

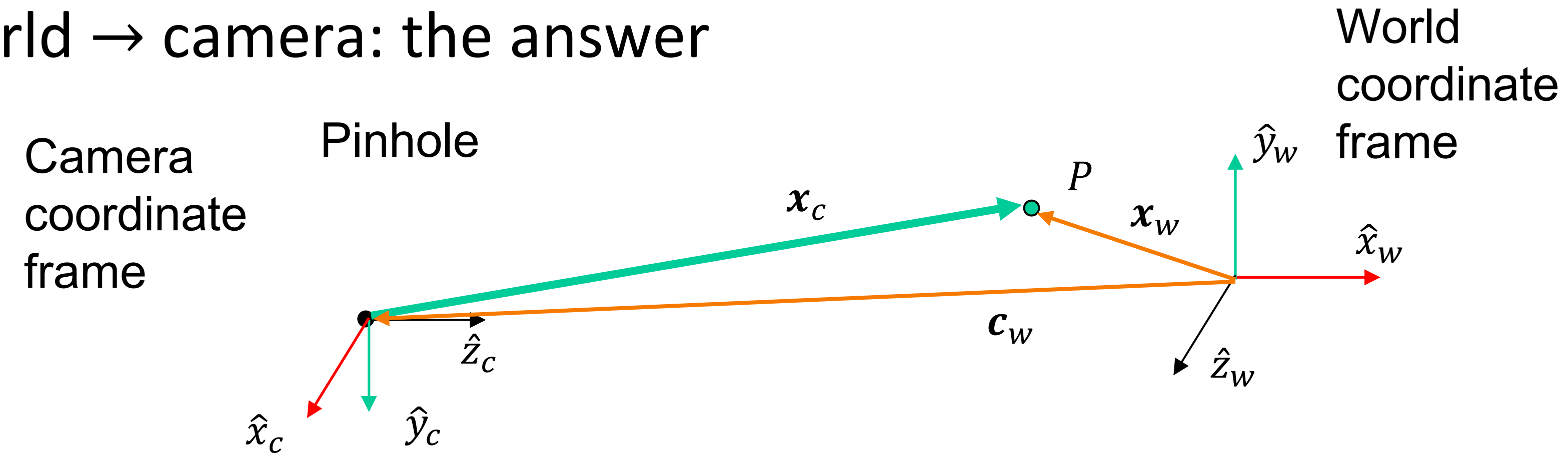
- The fix: append a constant 1 — the **homogeneous coordinates** $\tilde{\boldsymbol{p}} = \begin{bmatrix} \boldsymbol{p} \\ 1 \end{bmatrix} \in \mathbb{R}^4$. Rotation and translation become one 4×4 matrix:
  - $\tilde{\boldsymbol{p}}_A = T_{AB}\,\tilde{\boldsymbol{p}}_B, \quad T_{AB} = \begin{bmatrix} R_{AB} & \boldsymbol{t}_{AB} \\ 0^T & 1 \end{bmatrix} \in \mathrm{SE}(3)$
- Check the last row: $0 \cdot x_B + 0 \cdot y_B + 0 \cdot z_B + 1 \cdot 1 = 1$; rows 1–3 give $R_{AB}\boldsymbol{p}_B + \boldsymbol{t}_{AB}$
- Inverse and chaining are now matrix algebra: $T_{BA} = T_{AB}^{-1} = \begin{bmatrix} R_{AB}^T & -R_{AB}^T\boldsymbol{t}_{AB} \\ 0^T & 1 \end{bmatrix}, \quad T_{AC} = T_{AB}\,T_{BC}$
- Example, map → vehicle → camera: $\tilde{\boldsymbol{p}}_{map} = T_{map,veh}\,T_{veh,cam}\,\tilde{\boldsymbol{p}}_{cam}$

World → camera: the extrinsic parameters (question)



Q: We know the camera's pose in the world: its centre $\boldsymbol{c}_w$ and its orientation R_{wc} (columns = the camera axes $\hat{x}_c, \hat{y}_c, \hat{z}_c$ written in world coordinates), and the world coordinates $\boldsymbol{x}_w$ of a point P. What are the camera coordinates $\boldsymbol{x}_c$ of P?

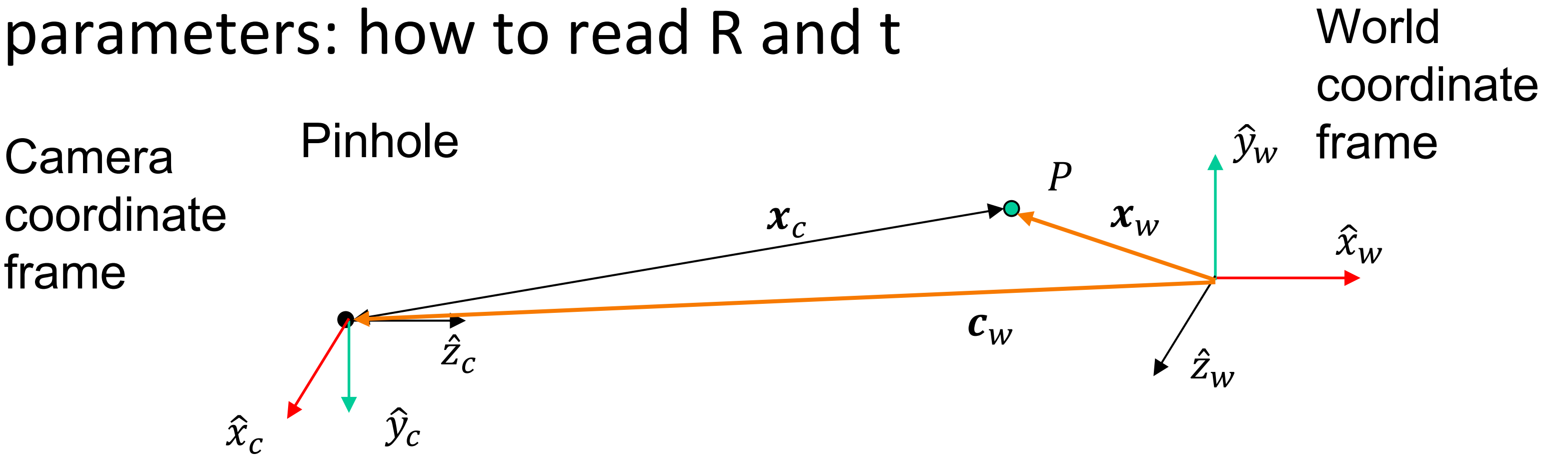
World → camera: the answer



Camera → world is the rigid transform: $\boldsymbol{x}_w = R_{wc}\,\boldsymbol{x}_c + \boldsymbol{c}_w$
Invert it ($R_{wc}^{-1} = R_{wc}^T$): $\boldsymbol{x}_c = R_{wc}^T(\boldsymbol{x}_w - \boldsymbol{c}_w) = R\,\boldsymbol{x}_w + \boldsymbol{t}$
Camera convention: $R := R_{wc}^T = R_{cw}$ (its **rows** are the camera axes in world coordinates), $\boldsymbol{t} := -R\,\boldsymbol{c}_w$.
$(R, \boldsymbol{t})$ = the camera's **extrinsic parameters**

$$\boldsymbol{x}_c = \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} \qquad \boldsymbol{x}_c = R\boldsymbol{x}_w + \boldsymbol{t}$$

Extrinsic parameters: how to read R and t



$$\boldsymbol{x}_c = R\,\boldsymbol{x}_w + \boldsymbol{t}, \quad R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} = R_{wc}^T:$$

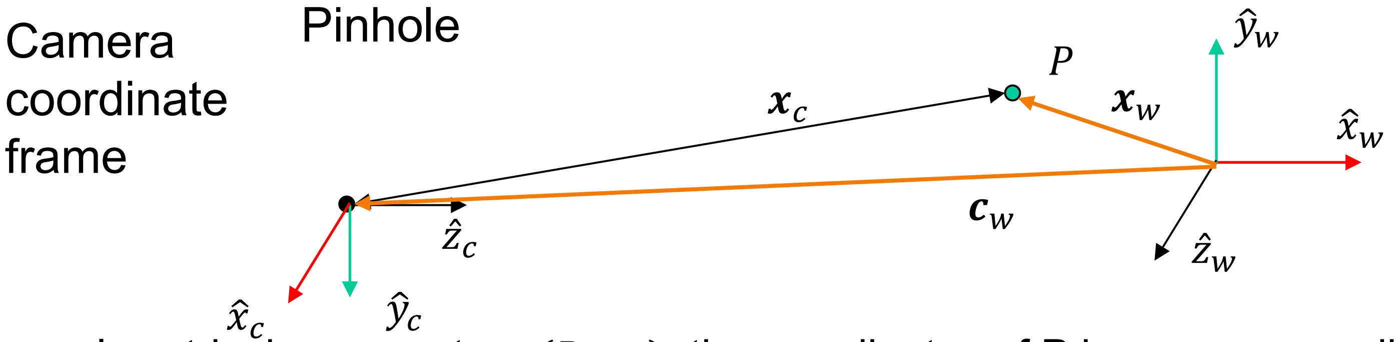
row 1 = direction of $\hat{x}_c$ in world coordinates, row 2 of $\hat{y}_c$, row 3 of $\hat{z}_c$

R is a rotation matrix: $R^T R = RR^T = I$, $\det R = +1$ ($\det R = -1$ would be a reflection) — so $R^{-1} = R^T$

Q: Is $\boldsymbol{t}$ the position of the camera?

No. $\boldsymbol{t} = -R\,\boldsymbol{c}_w$ is the **world origin expressed in the camera frame** (the displacement camera → world origin, measured along the camera axes). The camera position is $\boldsymbol{c}_w = -R^T\boldsymbol{t}$. Together $(R, \boldsymbol{t})$ hold 3 + 3 = 6 numbers: the 6-DoF pose.

Summary: the extrinsics map the world frame W to the camera frame C



Given camera's extrinsic parameters $(R, \boldsymbol{c}_w)$, the coordinates of P in camera coordinates

$\boldsymbol{x}_c = R(\boldsymbol{x}_w - \boldsymbol{c}_w) = R\boldsymbol{x}_w - R\boldsymbol{c}_w = R\boldsymbol{x}_w + \boldsymbol{t}$ with $\boldsymbol{t} = -R\boldsymbol{c}_w$

$$\boldsymbol{x}_c = \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}$$

Using homogeneous coordinates

$$\tilde{\boldsymbol{x}}_c = \begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}$$

Extrinsic matrix M_{ext} $\quad \tilde{\boldsymbol{x}}_c = M_{ext}\, \tilde{\boldsymbol{x}}_w$

$M_{ext} = T_{cw}$ of the previous slides $(A = C,\ B = W)$; $M_{ext}^{-1} = T_{wc}$

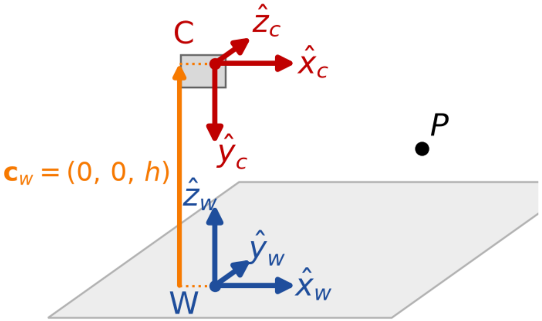
Slide ideas from First Principles of Computer Vision by Shree Nayar

Put numbers in: a camera on the vehicle

Camera 1.5 m above the road, looking forward. World frame: $\hat{x}_w$ right, $\hat{y}_w$ forward, $\hat{z}_w$ up, origin on the road directly below the camera, so $\boldsymbol{c}_w = (0,0,1.5)^T$ (metres)

Camera frame: $\hat{x}_c$ right, $\hat{y}_c$ down, $\hat{z}_c$ forward. Columns of R_{wc} = the camera axes in world coordinates: $(1,0,0)^T$, $(0,0,-1)^T$, $(0,1,0)^T$, so $R = R_{wc}^T$ has rows $(1,0,0)$, $(0,0,-1)$, $(0,1,0)$; $\boldsymbol{t} = -R\,\boldsymbol{c}_w = (0,1.5,0)^T$

Q: The bumper of the lead car is at $\boldsymbol{x}_w = (0.5,20,0.4)^T$. What is $\boldsymbol{x}_c$? Interpret each coordinate.

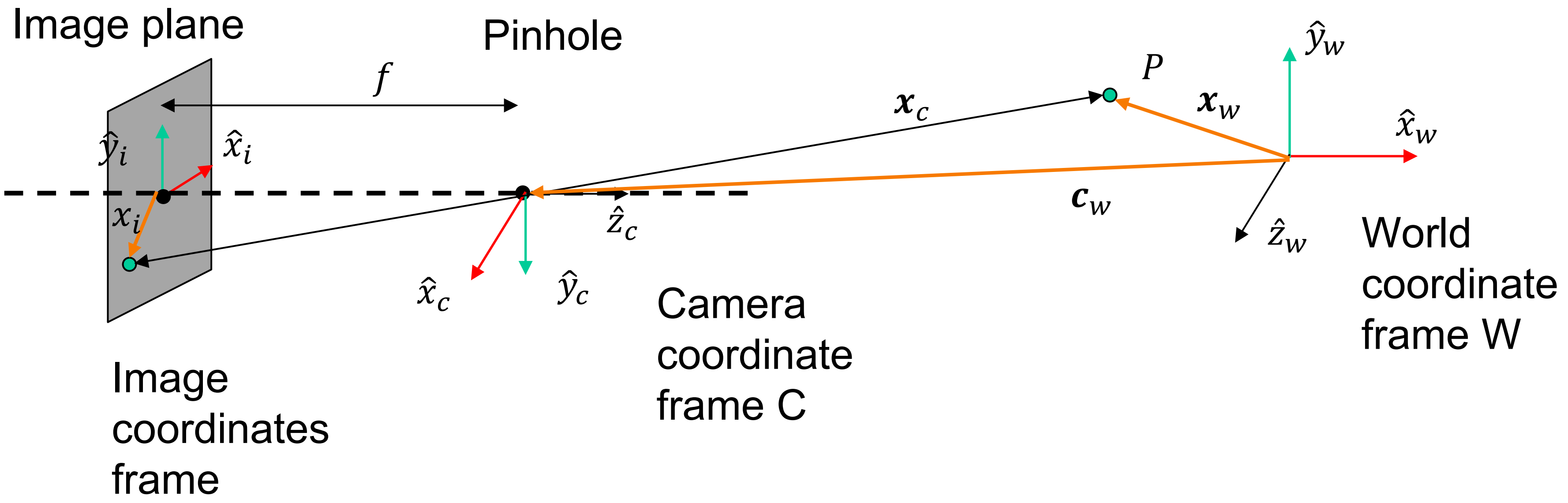


$R\,\boldsymbol{x}_w = (0.5,-0.4,20)^T$, $\boldsymbol{x}_c = R\,\boldsymbol{x}_w + \boldsymbol{t} = (0.5,\ 1.1,\ 20)^T$

$z_c = 20$: 20 m ahead of the camera (depth); $x_c = 0.5$: half a metre to the right; $y_c = 1.1$: 1.1 m below the camera ($\hat{y}_c$ points down, and the bumper at 0.4 m is 1.1 m under the camera at 1.5 m).

Check with $\boldsymbol{x}_c = R(\boldsymbol{x}_w - \boldsymbol{c}_w) = R\,(0.5,20,-1.1)^T = (0.5,1.1,20)^T$ ✓. Next lecture: this point → pixel (u,v).

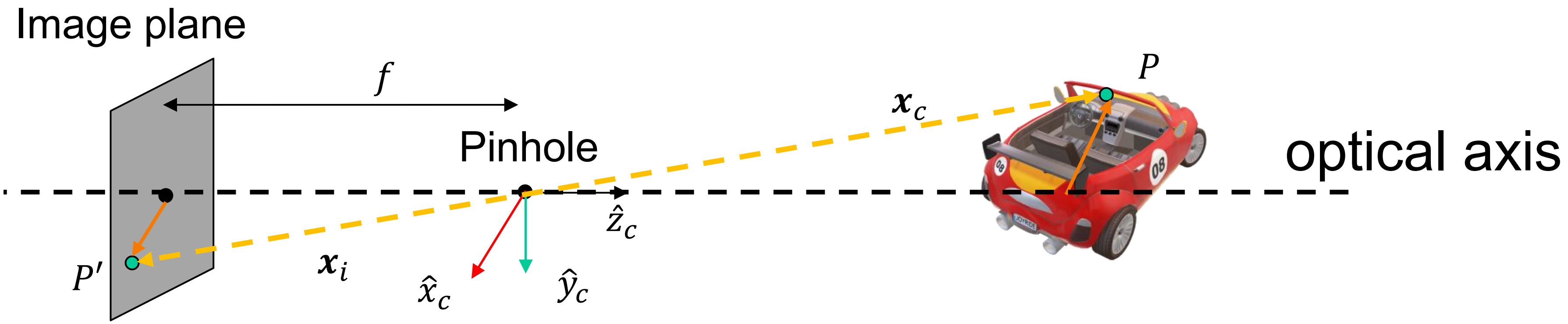
Forward Imaging Model: 3D to 2D



$$\boxed{\boldsymbol{x}_i = \begin{bmatrix} x_i \\ y_i \end{bmatrix} \quad \xleftarrow{\text{3D-2D}} \quad \boldsymbol{x}_c = \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix}} \quad \xleftarrow{\text{3D-3D}} \quad \boldsymbol{x}_w = \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix}$$

Slide ideas from First Principles of Computer Vision by Shree Nayar

Perspective imaging with pinhole

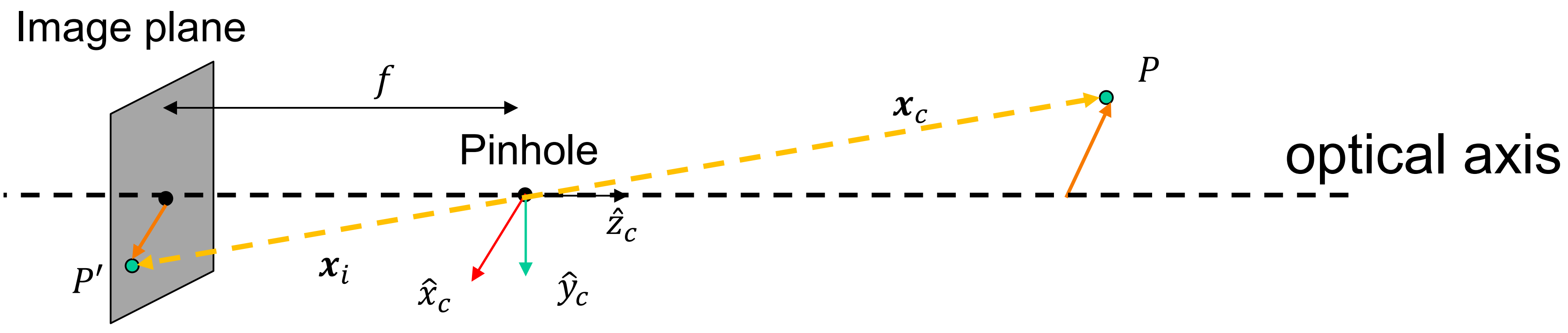


f: Effective focal length

$$\boldsymbol{x}_c = \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} \quad \boldsymbol{x}_i = \begin{bmatrix} x_i \\ y_i \\ f \end{bmatrix}$$

Slide ideas from First Principles of Computer Vision by Shree Nayar

Perspective imaging with pinhole



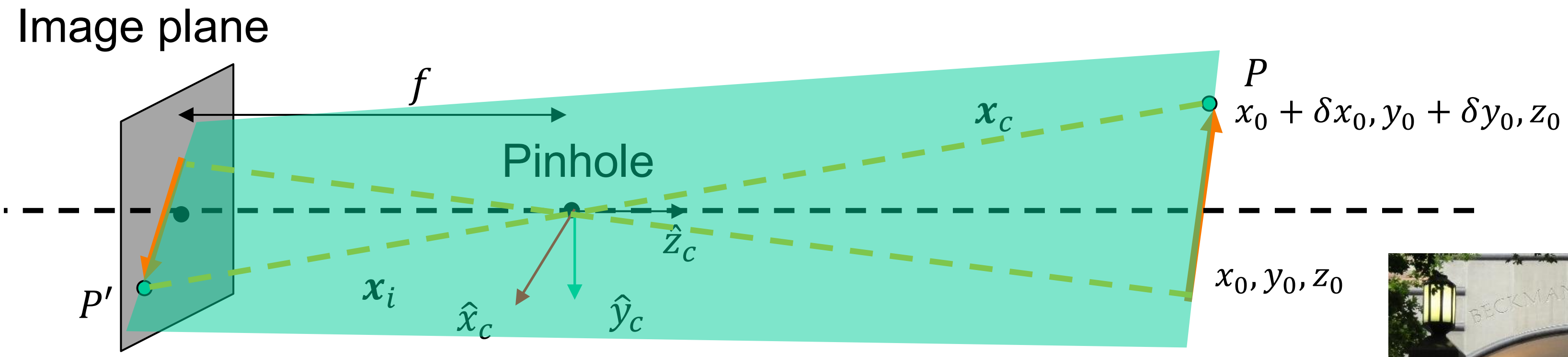
f: Effective focal length

$$\boldsymbol{x}_c = \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} \quad \boldsymbol{x}_i = \begin{bmatrix} x_i \\ y_i \\ f \end{bmatrix} \quad \boxed{\frac{\boldsymbol{x}_i}{f} = \frac{\boldsymbol{x}_c}{z_c}} \Rightarrow \boxed{\frac{x_i}{f} = \frac{x_c}{z_c}, \frac{y_i}{f} = \frac{y_c}{z_c}}$$

Not a linear transformation because division by z_c

Slide ideas from First Principles of Computer Vision by Shree Nayar

Perspective projection of a line and magnification



A line in 3D (not through the pinhole) is imaged as a line

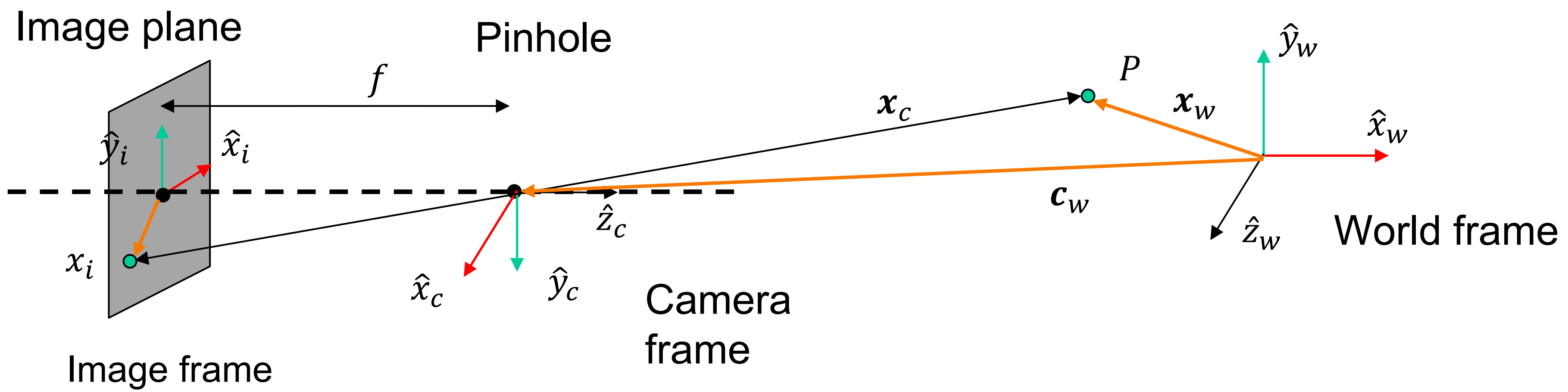
$$\frac{\boldsymbol{x}_i}{f} = \frac{\boldsymbol{x}_c}{z_c} \qquad \Rightarrow \frac{x_i}{f} = \frac{x_c}{z_c}, \frac{y_i}{f} = \frac{y_c}{z_c}$$

Exercise: show that for a small segment parallel to the image plane at depth z_0

the magnification is $|m| = \frac{\text{image length}}{\text{object length}} = \frac{\sqrt{\delta x_i^2 + \delta y_i^2}}{\sqrt{\delta x_0^2 + \delta y_0^2}} = \left|\frac{f}{z_0}\right|$

Slide ideas from First Principles of Computer Vision by Shree Nayar

Camera coordinates to image plane coordinates



Perspective projection

$$\frac{x_i}{f} = \frac{x_c}{z_c} \text{ and } \frac{y_i}{f} = \frac{y_c}{z_c}$$

$$\boxed{x_i = f\frac{x_c}{z_c} \text{ and } y_i = f\frac{y_c}{z_c}}$$

Slide ideas from First Principles of Computer Vision by Shree Nayar

Image plane to image sensor mapping

Image plane

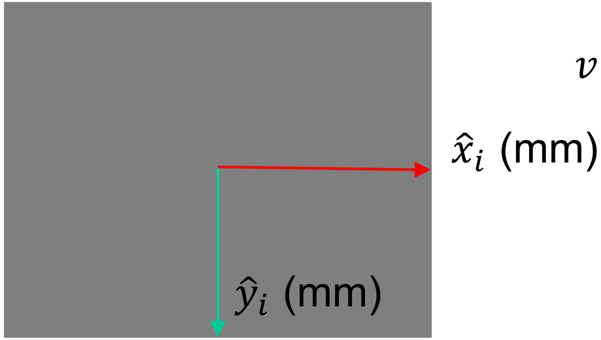
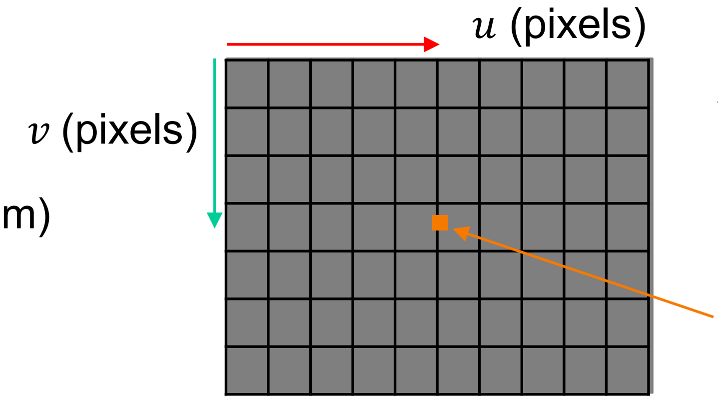


Image sensor



Pixels may be rectangular
Let m_x and m_y be the pixel densities (pixels/mm) in x and y directions

(o_x, o_y): principal point (pixel coordinates of the optical axis)

$$x_i = f\frac{x_c}{z_c} \text{ and } y_i = f\frac{y_c}{z_c}$$

$$u = m_x f\frac{x_c}{z_c} \text{ and } v = m_y f\frac{y_c}{z_c}$$

$$u = m_x f\frac{x_c}{z_c} + o_x \text{ and } v = m_y f\frac{y_c}{z_c} + o_y$$

$$\boxed{u = f_x\frac{x_c}{z_c} + o_x \text{ and } v = f_y\frac{y_c}{z_c} + o_y}$$

Intrinsic parameters: f_x, f_y, o_x, o_y
$f_x = m_x f$, $f_y = m_y f$ (pixels): f in mm, m_x, m_y in pixels/mm

Slide ideas from First Principles of Computer Vision by Shree Nayar

Nonlinear to linear model using homogeneous coordinates

$u = f_x \frac{x_c}{z_c} + o_x$ and $v = f_y \frac{y_c}{z_c} + o_y$

$u z_c = f_x x_c + o_x z_c$ and $v z_c = f_y\, y_c + o_y z_c$

Adding $z_c = 0 \cdot x_c + 0 \cdot y_c + 1 \cdot z_c$

Stacking them

$$\boxed{\tilde{\boldsymbol{u}} \equiv \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \equiv \begin{bmatrix} z_c u \\ z_c v \\ z_c \end{bmatrix} = \begin{bmatrix} f_x x_c + z_c o_x \\ f_y y_c + z_c o_y \\ z_c \end{bmatrix} = \begin{bmatrix} f_x & 0 & o_x & 0 \\ 0 & f_y & o_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix}}$$

Homogeneous representation of (u, v) as a 3D point $\tilde{\boldsymbol{u}} = (\tilde{u}, \tilde{v}, \tilde{w})$

$(u z_c, v z_c, z_c) \equiv (u, v, 1)$

Linear model of perspective projection $\tilde{\boldsymbol{u}} = [K|0]\tilde{\boldsymbol{x}}_c = M_{int}\tilde{\boldsymbol{x}}_c$

Intrinsic matrix (M_{int}) **Calibration matrix** K

Forward Camera Model

Camera to pixel

$$\boxed{\begin{bmatrix} \tilde{u} \\ \tilde{v} \\ \tilde{w} \end{bmatrix} = \begin{bmatrix} f_x & 0 & o_x & 0 \\ 0 & f_y & o_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix}}$$

World to camera

$$\boxed{\begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}}$$

$$\widetilde{\boldsymbol{u}} = M_{int}\ \widetilde{\boldsymbol{x}}_c = [K|0]\widetilde{\boldsymbol{x}}_c$$

$$\widetilde{\boldsymbol{x}}_c = M_{ext}\ \widetilde{\boldsymbol{x}}_w$$

$$\boxed{\widetilde{\boldsymbol{u}} = M_{\text{int}}\, M_{\text{ext}}\, \widetilde{\boldsymbol{x}}_w = P\, \widetilde{\boldsymbol{x}}_w = K[R\,|\,\boldsymbol{t}]\, \widetilde{\boldsymbol{x}}_w}$$

$$\begin{bmatrix} \tilde{u} \\ \tilde{v} \\ \tilde{w} \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}$$

P: **Projection matrix**

The left 3x3 matrix in P is a product of an upper triangular and an orthonormal matrix

P is 3×4 and defined only up to scale
⇒ 11 degrees of freedom

Worked example: a 3-D point to a pixel

- Camera 1.5 m above the road, looking forward. World frame: $\hat{x}_w$ right, $\hat{y}_w$ forward, $\hat{z}_w$ up, origin on the road below the camera, so $\boldsymbol{c}_w = (0,\ 0,\ 1.5)$ m
- Rows of R = camera axes in world coordinates: $\hat{x}_c = (1{,}0{,}0)$, $\hat{y}_c = (0{,}0{,}-1)$ (down), $\hat{z}_c = (0{,}1{,}0)$ (forward), so $R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}$, $\det R = +1$, $\boldsymbol{t} = -R\boldsymbol{c}_w = (0,\ 1.5,\ 0)$
- Bumper of the lead car at $\boldsymbol{x}_w = (0.5,\ 20,\ 0.4)$ m: $\boldsymbol{x}_c = R\boldsymbol{x}_w + \boldsymbol{t} = (0.5,\ -0.4,\ 20) + (0,\ 1.5,\ 0) = (0.5,\ 1.1,\ 20)$ — depth $z_c = 20$ m > 0
- With $f_x = f_y = 800$ px and $(o_x, o_y) = (320{,}240)$: $u = 800 \cdot 0.5/20 + 320 = 340$, $v = 800 \cdot 1.1/20 + 240 = 284$ (below the centre, as it must be with $\hat{y}_c$ down)