ECE 484: Principles of Safe Autonomy (Fall 2025)
Lectures 5
Perception: Reconstructing 3D world from images
(Part I)

Professor: Huan Zhang

https://publish.illinois.edu/safe-autonomy/

https://huan-zhang.com

huanz@illinois.edu

Announcement

As requested by the TA:

For preparation for the **MP0 demo**, have a screenshot of R3 (third initial set) plot ready. It should be using whatever function proves safety or unsafely for this scenario.

Check Campuswire for more details

Role of Perception in Autonomy

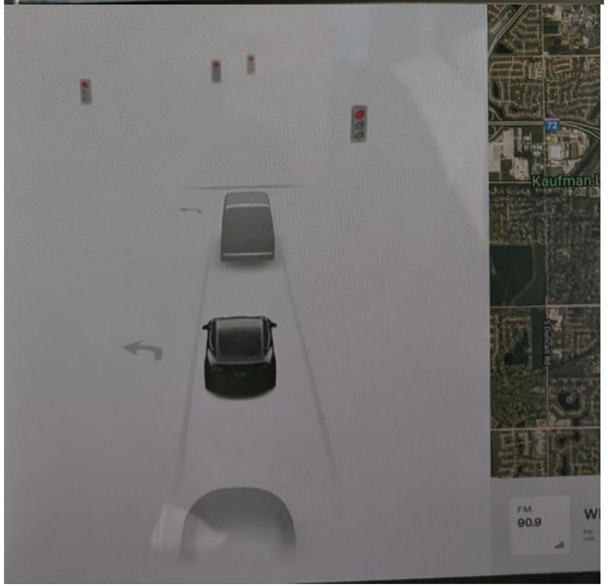
Perception module converts signals from the environment **state estimates** for the autonomous agent and its environment

Examples of state estimates:

- Type of lead vehicle, traffic sign
- Position of ego on the map, relative to the lane, distance to the leading vehicle
- Position of lead vehicle, speed, intention of the pedestrian

Types of estimates:

- Semantic: E.g., type of vehicle, sign
- Geometric: E.g., position, speed



Problem

Reconstructing the 3D structure of the scene from images

Input: image with points in pixels

Output: position of objects in millimeters in world camera frame

We will develop a method to find camera's internal and external parameters

Outline:

Linear Camera Model (Projection matrix)

Camera calibration

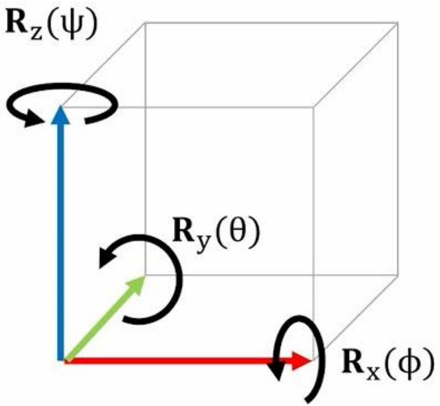
Simple stereo

Background: rotation in 3D

$$\mathbf{R}_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) & \cos(\phi) \end{bmatrix}$$

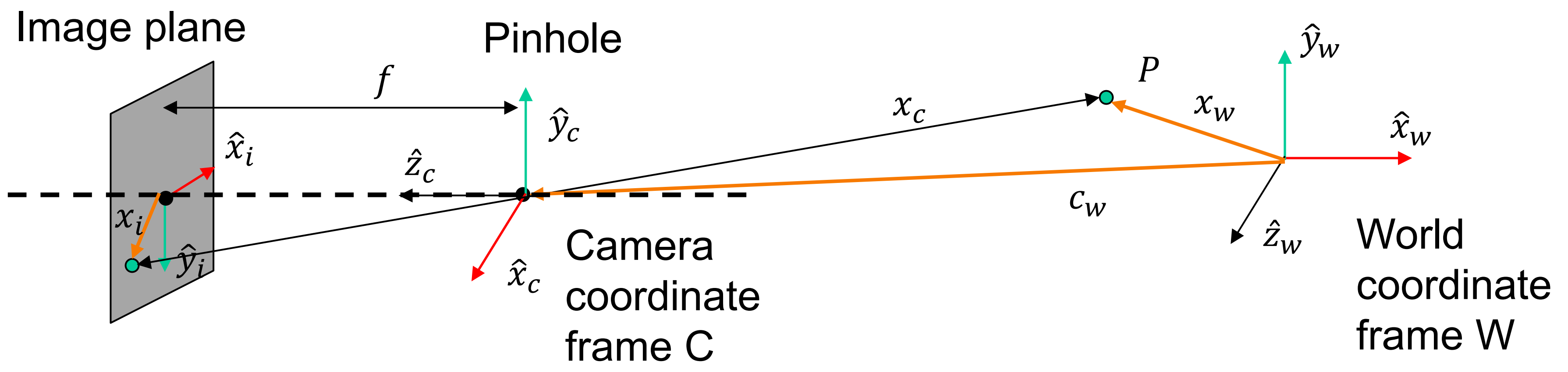
$$\mathbf{R}_y(\theta) = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix}$$

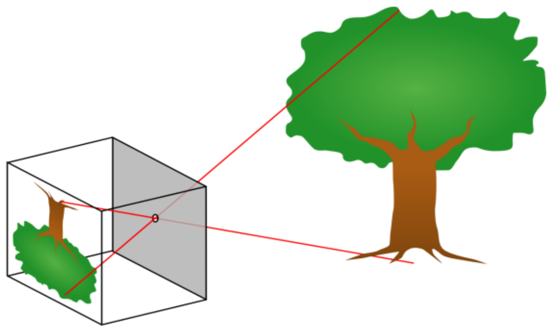
$$\mathbf{R}_z(\psi) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



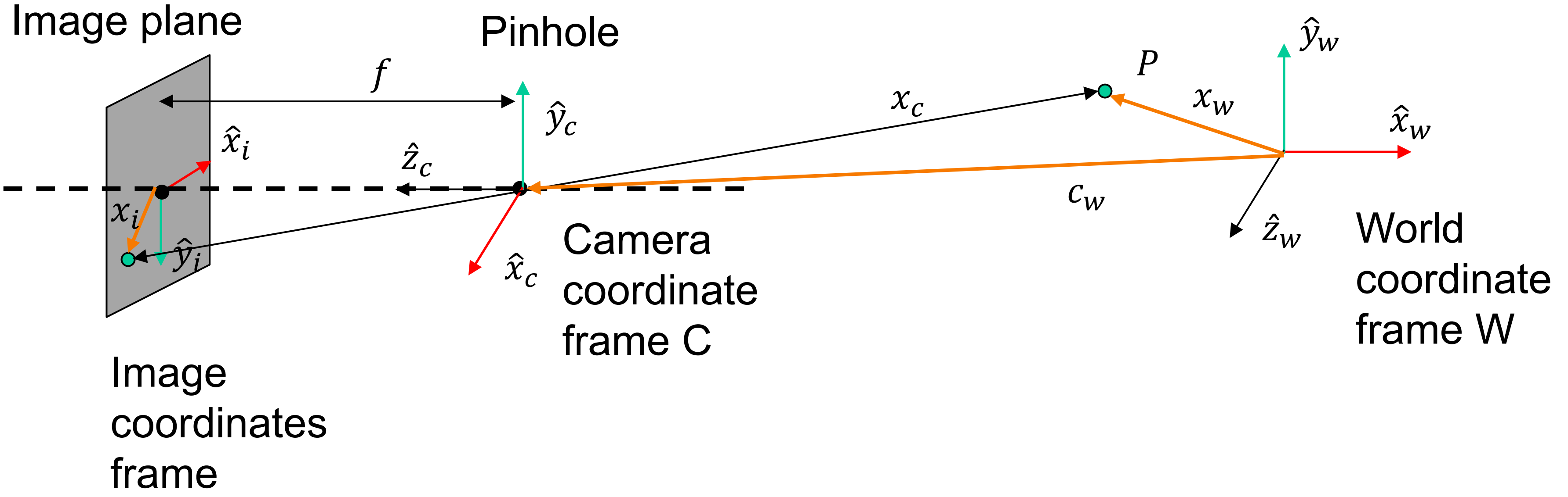
https://www.youtube.com/watch?v=wg9bI8-Qx2Q

Forward Imaging Model: 3D to 2D



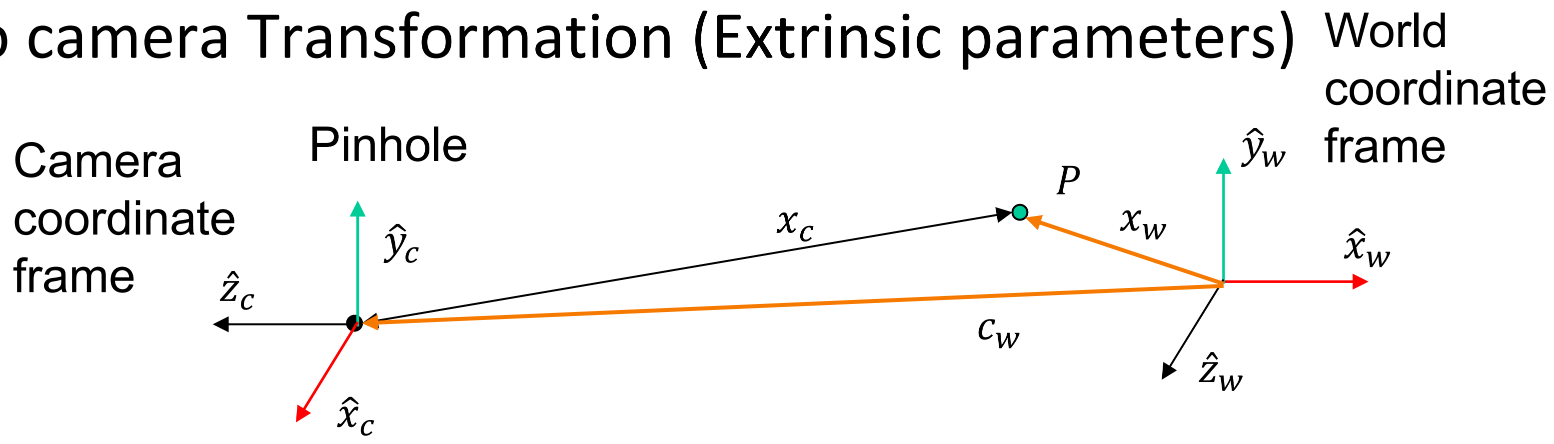


Forward Imaging Model: 3D to 2D



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

World to camera Transformation (Extrinsic parameters)

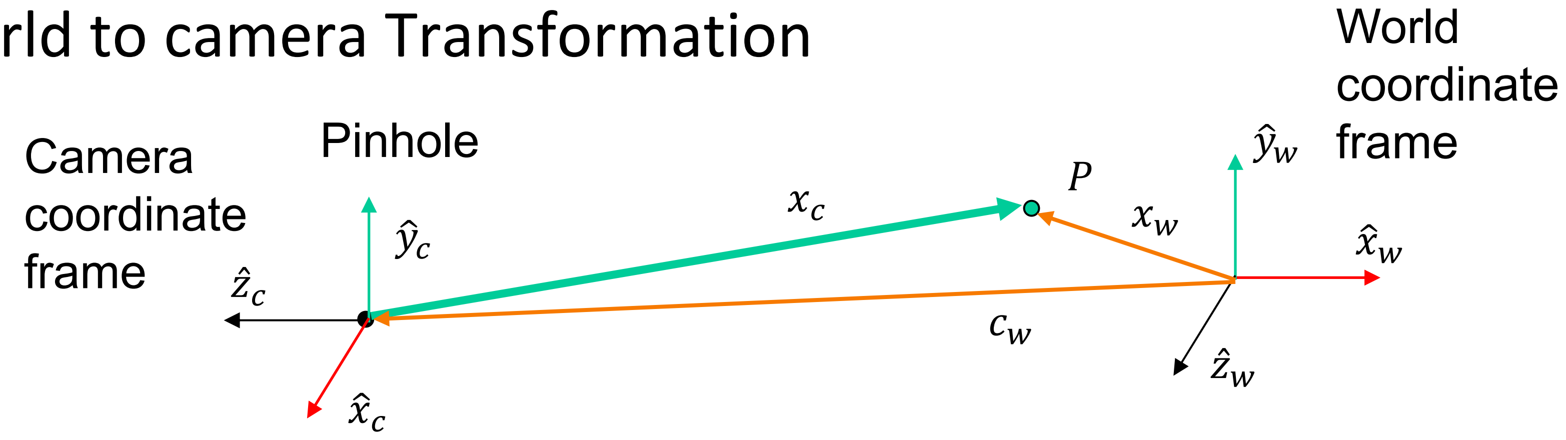


Position c_w and the orientation R of the camera in the world coordinate frame (W) are the camera's **Extrinsic Parameters**

Rotation matrix $R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$ → row 1 is the direction of $\hat{x}_c$ in world coordinates, 2 for $\hat{y}_c$, …

This is an **orthonormal matrix**, i.e., the row vectors or the column vectors are orthonormal
$R^{-1} = R^T$ i.e., $R^T R = R R^T = I$
Proof: $(Rx)^T (Rx) = x^T R^T R x = x^T x$ (since rotation perseves length). So $R^T R = I$

World to camera Transformation



Position c_w and the orientation R of the camera in the world coordinate frame (W) are the camera's **Extrinsic Parameters**

Given the extrinsic parameters (R, c_w) of the camera, the camera-centric location of the point P in the world coordinate (w) is simply $(x_c)_w = x_w - c_w$

In the camera coordinate (c) $x_c = R(x_w - c_w) = Rx_w - Rc_w = Rx_w + t$

$$t = -Rc_w$$

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

Extrinsic Matrix

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

We have an affine transformation: $x_c = Rx_w + t$

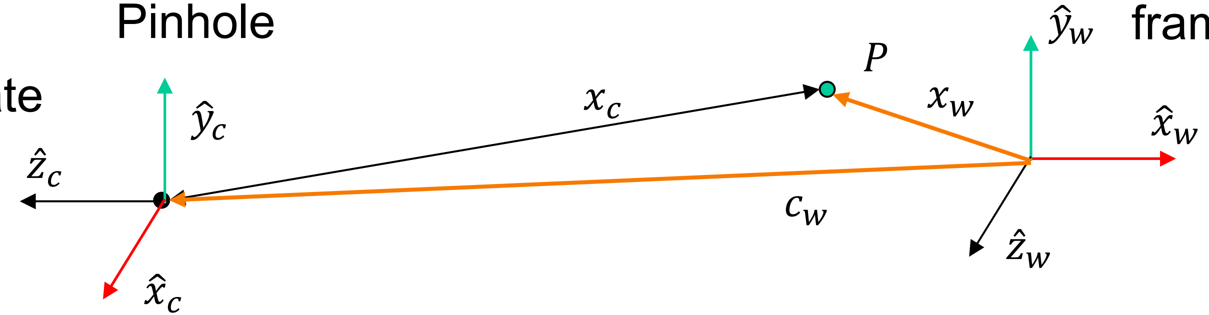
Can we represent it as $x_c = Mx_w$? No

We can introduce a new coordinate $\tilde{x}_c = [\tilde{x}, \tilde{y}, \tilde{z}, 1]^T$

Now can we represent this as a matrix multiplication $\tilde{x}_c = M\tilde{x}_w$

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

World to camera Transformation (Extrinsic matrix)



Given the extrinsic parameters (R, c_w) of the camera, the camera-centric location of the point P in the world coordinate is

$$x_c = R(x_w - c_w) = Rx_w - Rc_w = Rx_w + t \qquad t = -Rc_w$$

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

Using homogeneous coordinates

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

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

Geometry of Homogeneous coordinates (for 2D)

$$x_c = \begin{bmatrix} x_c \\ y_c \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} \\ r_{21} & r_{22} \end{bmatrix} \begin{bmatrix} x_w \\ y_w \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

Affine transformation: $\mathrm{x_c} = \mathrm{R}\mathrm{x_w} + \mathrm{t}$

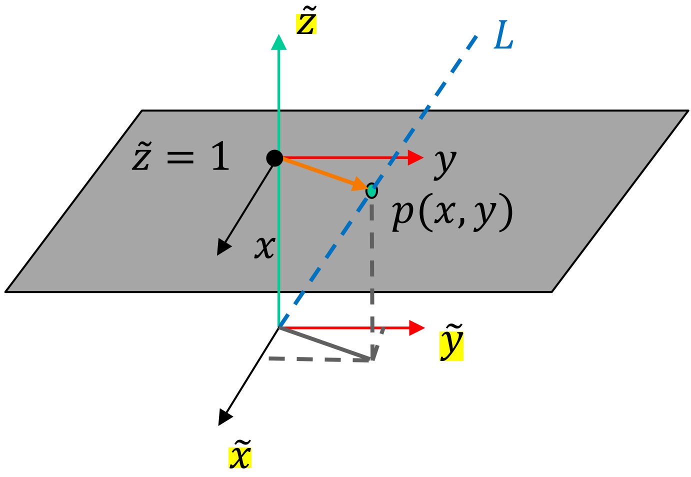
How to represent this as $\tilde{\mathrm{x}}_\mathrm{c} = \mathrm{M}\tilde{\mathrm{x}}_\mathrm{w}$

The homogeneous representation of a 2D point $p = (x, y)$ is a 3D point $\tilde{p} = (\tilde{x}, \tilde{y}, \tilde{z})$.

The third coordinate $\tilde{z} \neq 0$ is fictitious such that:

$$p = (x, y) \quad x = \frac{\tilde{x}}{\tilde{z}} \; y = \frac{\tilde{y}}{\tilde{z}}$$

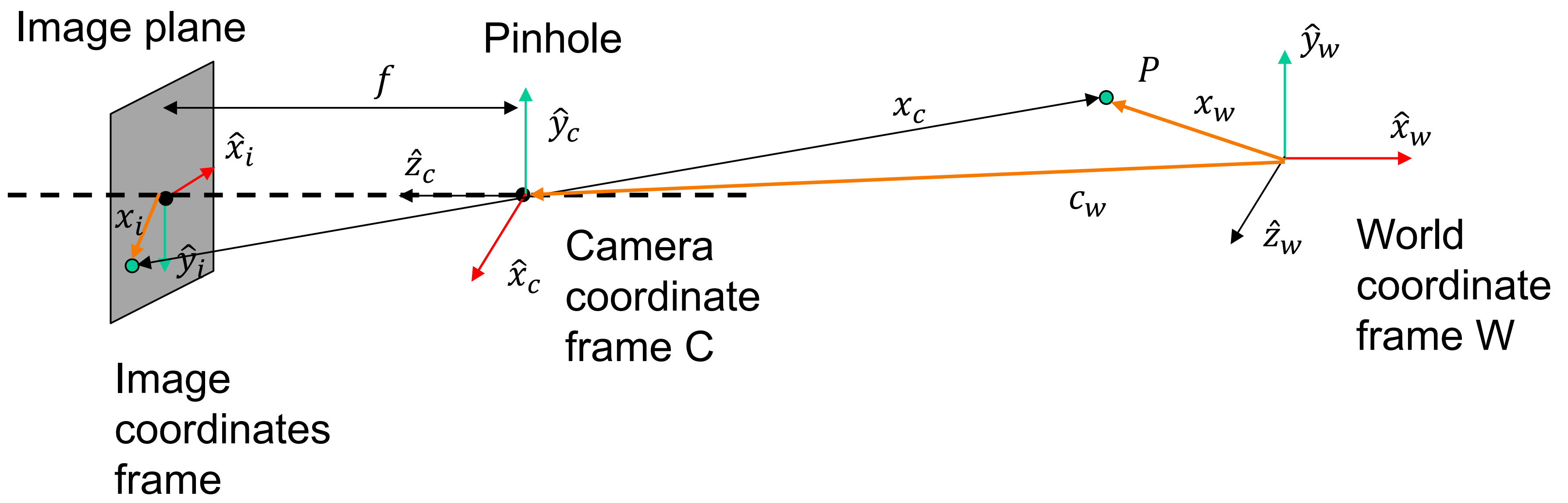
$$p \equiv \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \equiv \begin{bmatrix} \tilde{z}x \\ \tilde{z}y \\ \tilde{z} \end{bmatrix} \equiv \begin{bmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{z} \end{bmatrix} = \tilde{p}$$



Geometric interpretation: all points on the line L (except origin) represent homogeneous coordinate $p(x, y)$

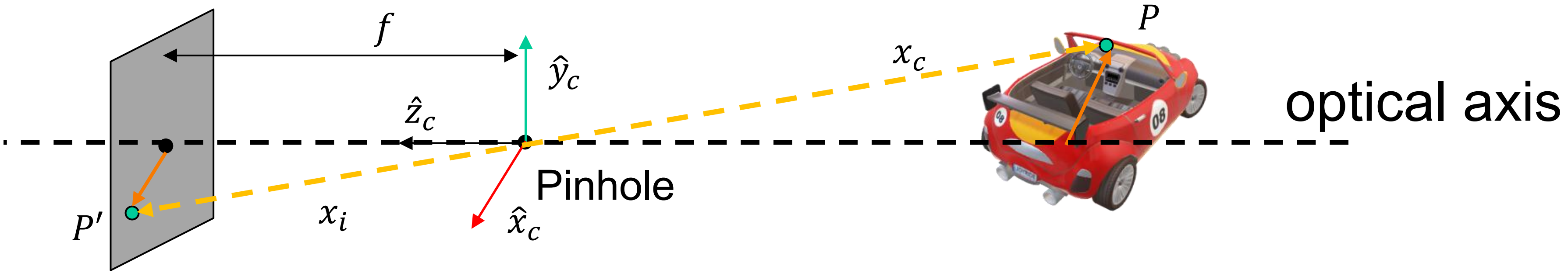
For 3D, homogeneous representation is 4D: $p \equiv \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \equiv \begin{bmatrix} wx \\ \tilde{w}y \\ \tilde{w}z \\ \tilde{w} \end{bmatrix} \equiv \begin{bmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{z} \\ \tilde{w} \end{bmatrix} = \tilde{p}$

Forward Imaging Model: 3D to 2D



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

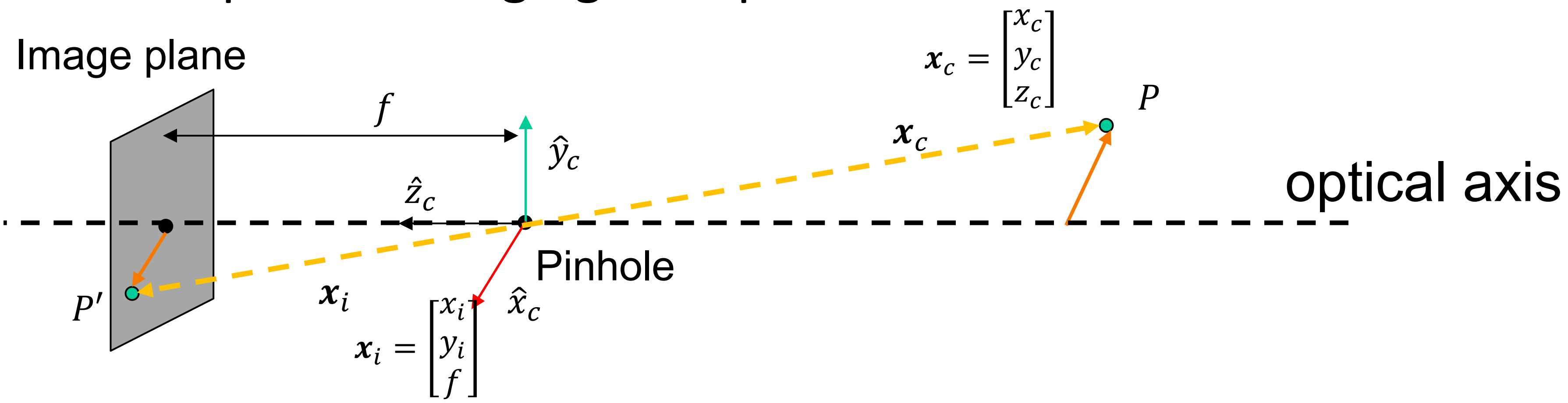
Perspective imaging with pinhole



f: Effective focal length

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

Perspective imaging with pinhole

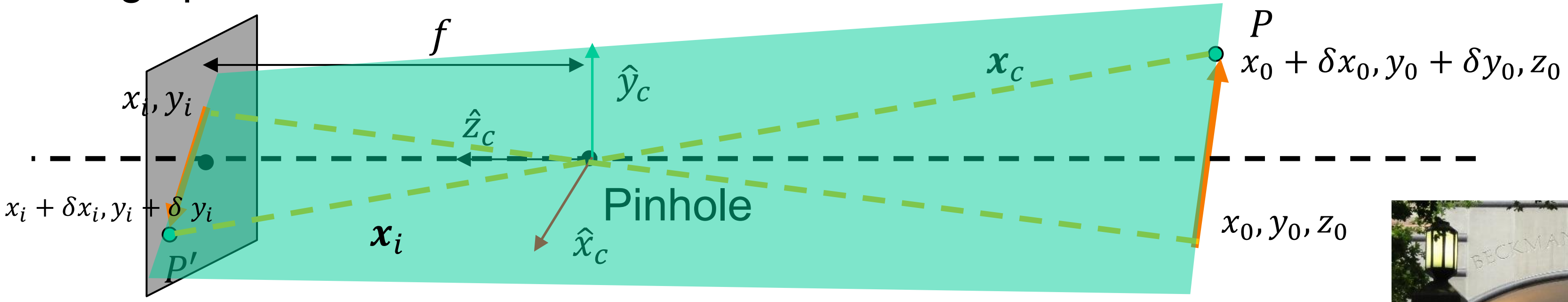


f: Effective focal length

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

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

Perspective projection of a line and magnification

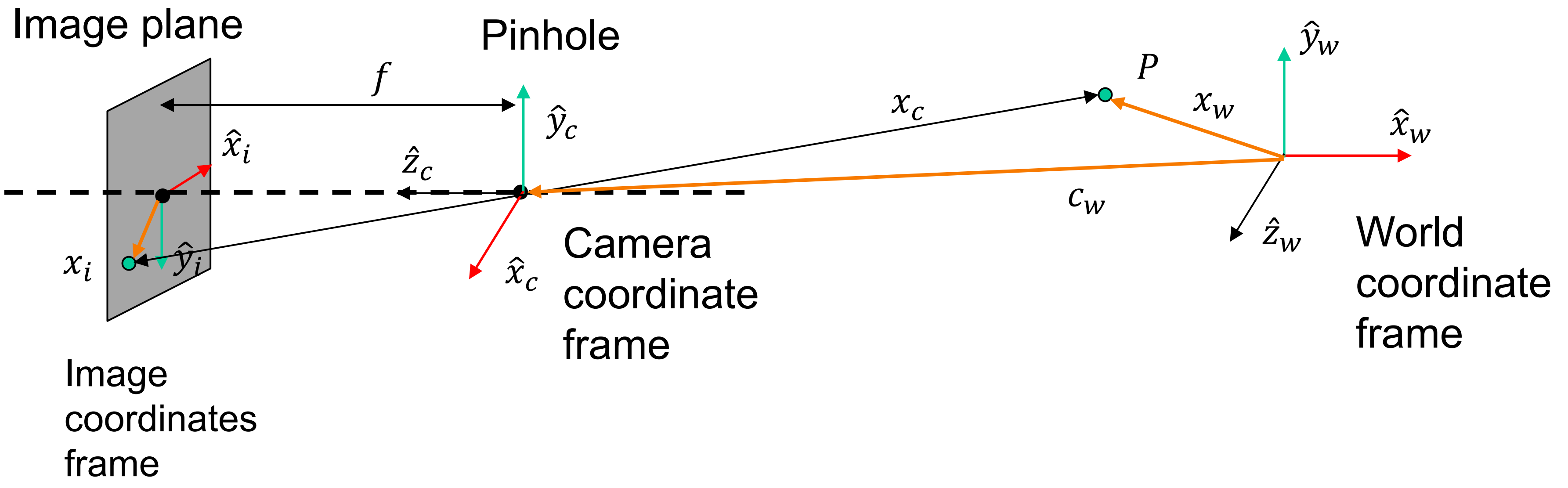


A line in 3D gets mapped to a line in the image plane

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

Exercise: Show that magnification $|\mathrm{m}| = \frac{image\ length}{actual\ length} = \frac{\sqrt{\delta x_i^2 + \delta y_i^2}}{\sqrt{\delta x_o^2 + \delta y_o^2}} = \left|\frac{f}{z_0}\right|$

Camera coordinates to image plane coordinates



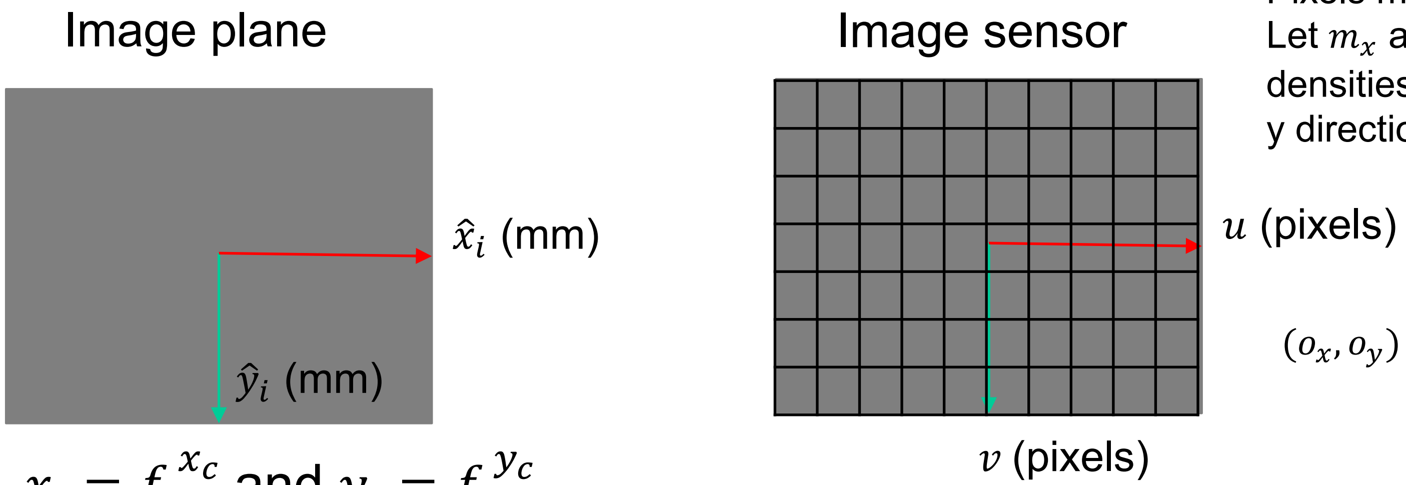
Perspective projection

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

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

Image plane to image sensor mapping

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) Principle point

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

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

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

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

Intrinsic parameters: f_x, f_y, o_x, o_y

Nonlinear to linear model using homogeneous coordinates

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

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

$uz_c = f_x x_c + o_x z_c$ and $vz_c = f_y\, y_c + o_y z_c$

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

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

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

Intrinsic matrix (M_{int})

Calibration matrix K (upper right triangular)

Forward Camera Model

Camera to pixel

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

World to camera

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

$$\tilde{u} = M_{int} \ \tilde{x}_w$$

$$\tilde{x}_c = M_{ext} \ \tilde{x}_w$$

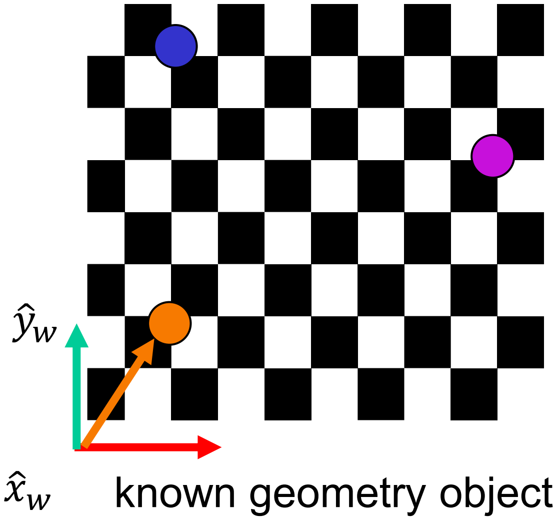
$$\tilde{u} = M_{int} M_{ext} \tilde{x}_w = P \ \tilde{x}_w$$

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

P: **Projection matrix**

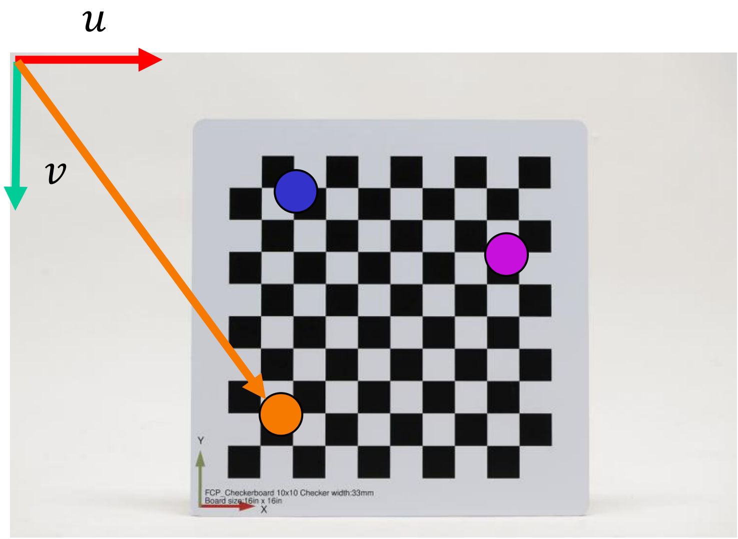
Camera Calibration Procedure

Step 1. Capture image of object with known geometry



known geometry object
(world coordinate)

$$\boldsymbol{x}_W = \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix}$$



captured image
(pixel coordinate)

$$\boldsymbol{u} = \begin{bmatrix} u \\ v \end{bmatrix}$$

Camera Calibration

Step 3. For each point i in the scene and the image we get a linear equation

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

Step 4. Collecting many $u^{(i)} = \frac{p_{11}x_w^{(i)} + p_{12}y_w^{(i)} + p_{13}z_w^{(i)} + p_{14}}{p_{31}x_w^{(i)} + p_{32}y_w^{(i)} + p_{33}z_w^{(i)} + p_{34}}$ points (same for $v^{(i)}$) **and rearranging p as a vector** we get $A\boldsymbol{p} = 0$

e.g., $p_{11}x_w^{(i)} + p_{12}\,y_w^{(i)} + p_{13}\,z_w^{(i)} + p_{14} - p_{31}x_w^{(i)}u^{(i)} - p_{32}\,u^{(i)}y_w^{(i)} - p_{33}\,u^{(i)}z_w^{(i)} - p_{34}u^{(i)} = 0$ is one row of this equality

Step 5. Solve for $\boldsymbol{p}$

Projection matrix scale

Since projection matrix works on homogeneous coordinates

$$\begin{bmatrix} \tilde{u} \\ \tilde{v} \\ \widetilde{w} \end{bmatrix} \equiv k \begin{bmatrix} \tilde{u} \\ \tilde{v} \\ \widetilde{w} \end{bmatrix}$$

Therefore

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

Therefore, Projection Matrices P and kP produce the same homogenous pixel coordinates

Projection matrix is defined only upto a scale factor

Scaling the world and the camera will produce indistinguishable images

That is , we can only find the projection matrix up to scale; we choose $||\boldsymbol{p}|| = \mathbf{1}$

Least Squares Solution for Projection Matrix

We want $A\boldsymbol{p}$ as close to 0 as possible and $||\boldsymbol{p}||^2 = 1$

$\min_{\boldsymbol{p}} ||A\boldsymbol{p}||^{\mathbf{2}}$ such that $||\boldsymbol{p}||^2 = 1$

$\min_{\boldsymbol{p}} ||\boldsymbol{p}^{\boldsymbol{T}} A^T A\boldsymbol{p}||^{\mathbf{2}}$ such that $\boldsymbol{p}^{\boldsymbol{T}}\boldsymbol{p} = 1$

$$L(\boldsymbol{p}, \lambda) = \boldsymbol{p}^{\boldsymbol{T}} A^T A\boldsymbol{p} - \lambda(\boldsymbol{p}^{\boldsymbol{T}}\boldsymbol{p} - 1)$$

Taking derivative $\frac{\partial L}{\partial \boldsymbol{p}} = 0$ gives $2A^T A\boldsymbol{p} - 2\lambda\boldsymbol{p} = \mathbf{0}$

$$\boxed{A^T A\boldsymbol{p} = \lambda\boldsymbol{p}}$$

$\boldsymbol{p}$ is the Eigenvector corresponding to the smallest eigenvalue of $A^T A$

Rearrange $\boldsymbol{p}$ to get the projection matrix **P**

Homography

Image transformations

2x2 transformations

3x3 transformations

Computing homography

Dealing with outliers RANSAC

Image manipulation

Image filtering: Change range (e.g., brightness)

$g(x,y) = Tr(f(x,y))$

Tr

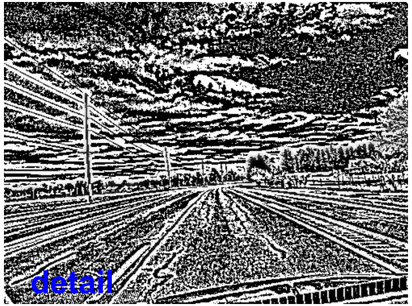


Image warping: Change domain (e.g., rotation)

$g(x,y) = f(Td(x,y))$

2x2 Linear Transformations

$p_1 = (x_1, y_1)$

$p_2 = (x_2, y_2)$

$p_2 = T p_1$

T can be represented by a matrix

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}$$

Scaling (stretching or squishing)

Forward

$$x_2 = a x_1 \quad y_2 = b y_1$$

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = S \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} a & 0 \\ 0 & b \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}$$

Inverse

$$x_2 = \frac{1}{a} x_1 \quad y_2 = \frac{1}{b} y_1$$

$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = S^{-1} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} \frac{1}{a} & 0 \\ 0 & \frac{1}{b} \end{bmatrix} \begin{bmatrix} x_2 \\ y_2 \end{bmatrix}$$

2D Rotation

$$x_1 = r\,cos(\psi)$$
$$y_1 = r\,sin(\psi)$$

Forward

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = R \begin{bmatrix} x \\ y_1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}$$

Inverse

$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = R^{-1} \begin{bmatrix} x \\ y_1 \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x_2 \\ y_2 \end{bmatrix}$$

2x2 Matrix Transformations

Any transformation of the form:

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}$$

Examples: Scaling, rotation, skew, mirror

Properties

- Origin maps to origin
- Lines map to lines
- Parallel lines remain parallel
- Closed under composition

If $p_2 = T_{21} p_1,\ p_3 = T_{32} p_2$ then $p_3 = T_{31} p_1$ where $T_{31} = T_{32} T_{21}$

Translation

Forward

$x_2 = x_1 + t_x$

$y_2 = y_1 + t_y$

Can a 2x2 matrix express this transformation? No

Homogeneous coordinates

The homogeneous representation of a 2D point $p = (x, y)$ is a 3D point $\tilde{p} = (\tilde{x}, \tilde{y}, \tilde{z})$. The third coordinate $\tilde{z} \neq 0$ is fictitious such that:

$$x = \frac{\tilde{x}}{\tilde{z}} \; y = \frac{\tilde{y}}{\tilde{z}}$$

$$p \equiv \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \equiv \begin{bmatrix} \tilde{z}x \\ \tilde{z}y \\ \tilde{z} \end{bmatrix} \equiv \begin{bmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{z} \end{bmatrix} = \tilde{p}$$

Translation

Forward

$$x_2 = x_1 + t_x$$

$$y_2 = y_1 + t_y$$

$$\begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} \equiv \begin{bmatrix} \tilde{x}_2 \\ \tilde{y}_2 \\ \tilde{z}_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix}$$

3x3 **Affine** Transformations

Scaling

$$\begin{bmatrix}\tilde{x}_2\\ \tilde{y}_2\\ \tilde{z}_2\end{bmatrix} = \begin{bmatrix}s_x & 0 & 0\\ 0 & s_y & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}x_1\\ y_1\\ 1\end{bmatrix}$$

Skew

$$\begin{bmatrix}\tilde{x}_2\\ \tilde{y}_2\\ \tilde{z}_2\end{bmatrix} = \begin{bmatrix}1 & m_x & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}x_1\\ y_1\\ 1\end{bmatrix}$$

Translation

$$\begin{bmatrix}\tilde{x}_2\\ \tilde{y}_2\\ \tilde{z}_2\end{bmatrix} = \begin{bmatrix}1 & 0 & t_x\\ 0 & 1 & t_y\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}x_1\\ y_1\\ 1\end{bmatrix}$$

Rotation

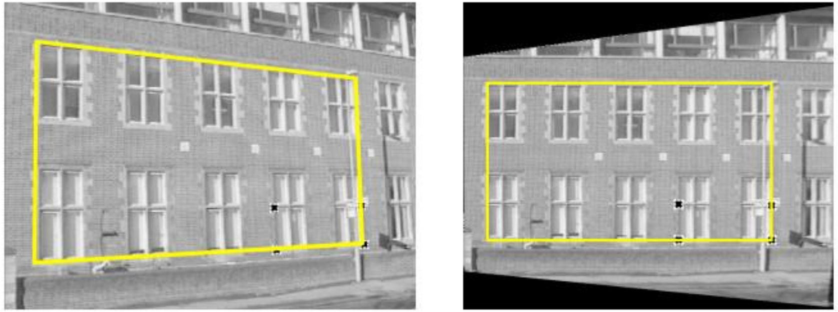
$$\begin{bmatrix}\tilde{x}_2\\ \tilde{y}_2\\ \tilde{z}_2\end{bmatrix} = \begin{bmatrix}\cos\theta & -\sin\theta & 0\\ \sin\theta & \cos\theta & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}x_1\\ y_1\\ 1\end{bmatrix}$$

Composition of Transformations

General form $$\begin{bmatrix}\tilde{x}_2\\ \tilde{y}_2\\ \tilde{z}_2\end{bmatrix} = \begin{bmatrix}a_{11} & a_{12} & a_{13}\\ a_{21} & a_{22} & a_{23}\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}x_1\\ y_1\\ 1\end{bmatrix}$$

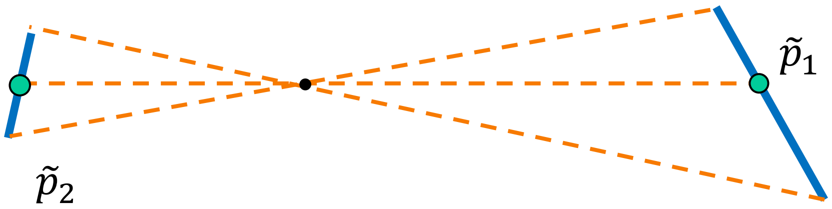
Projective Transformations

General form $\begin{bmatrix} \tilde{x}_2 \\ \tilde{y}_2 \\ \tilde{z}_2 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} \tilde{x}_1 \\ \tilde{x}_2 \\ \tilde{x}_3 \end{bmatrix}$



from Hartley & Zisserman

H is called a homography



Origin does not necessarily map to origin

Lines map to lines

Parallel lines do not necessarily remain parallel

Closed under composition

Homographies are defined up to scaling

When $h_{31} = h_{31} = 0$, $h_{33} = 1$, it becomes **affine** homography