Johns Hopkins Engineering

Computer Vision

Camera Calibration and Photogrammetry

JOHNS HOPKINS
WHITING SCHOOL
of ENGINEERING

Camera Calibration and Photogrammetry

Method to find a camera's parameters and a method to estimate 3D structure using two cameras.

Topics:

(1) Linear Camera Model

(2) Camera Calibration

(3) Photogrammetry and Stereo

Camera Calibration and Photogrammetry

Method to find a camera's parameters and a method to estimate 3D structure using two cameras.

Topics:

(1) Linear Camera Model

Forward Imaging Model: 3D to 2D

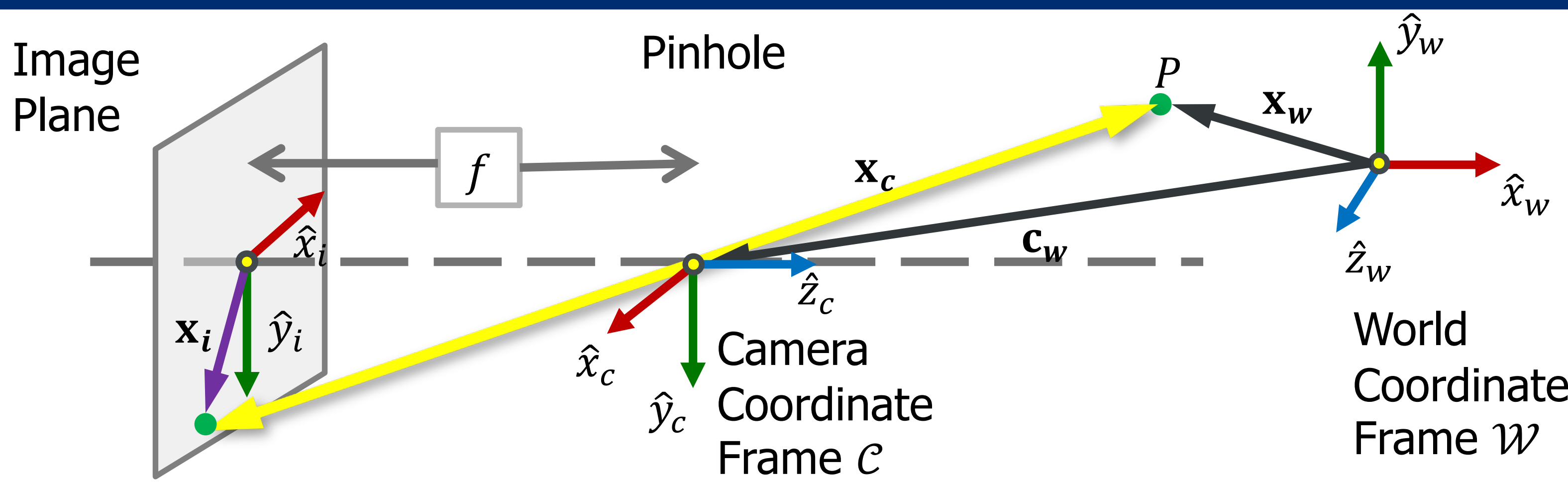


Image Plane Coordinates		Camera Coordinates		World Coordinates
$\mathbf{x}_i = \begin{bmatrix} x_i \\ y_i \end{bmatrix}$	← Perspective Projection	$\mathbf{x}_c = \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix}$	← Coordinate Transformation	$\mathbf{x}_w = \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix}$

Forward Imaging Model: 3D to 2D

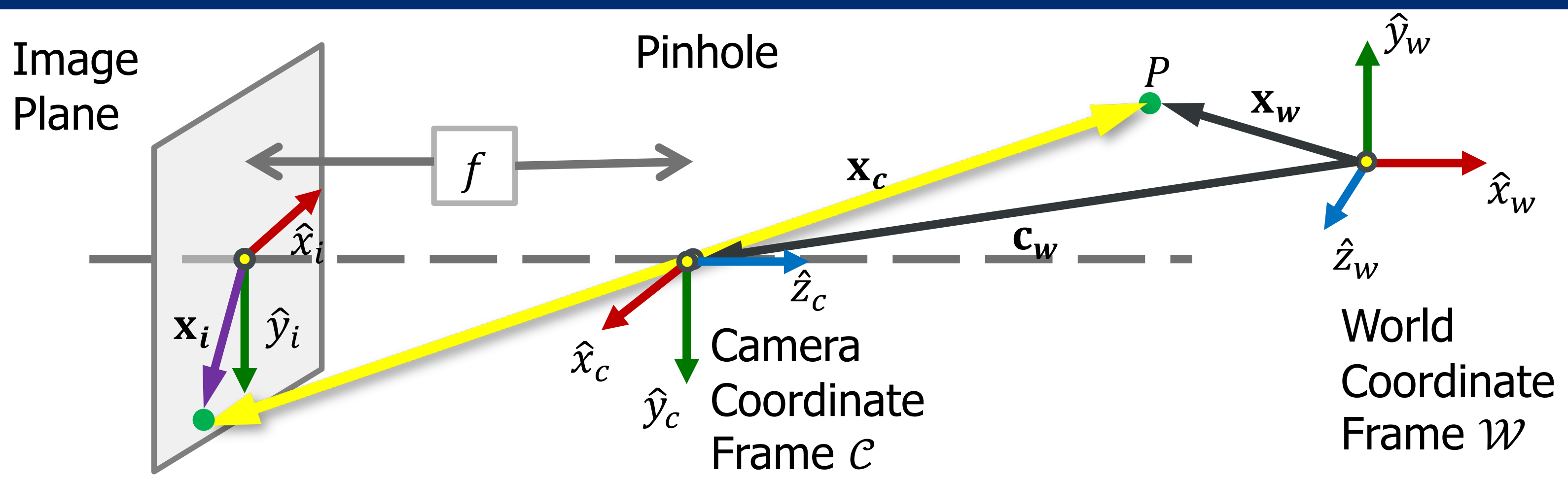


Image Plane Coordinates

$$\mathbf{x}_i = \begin{bmatrix} x_i \\ y_i \end{bmatrix}$$

Perspective Projection

Camera Coordinates

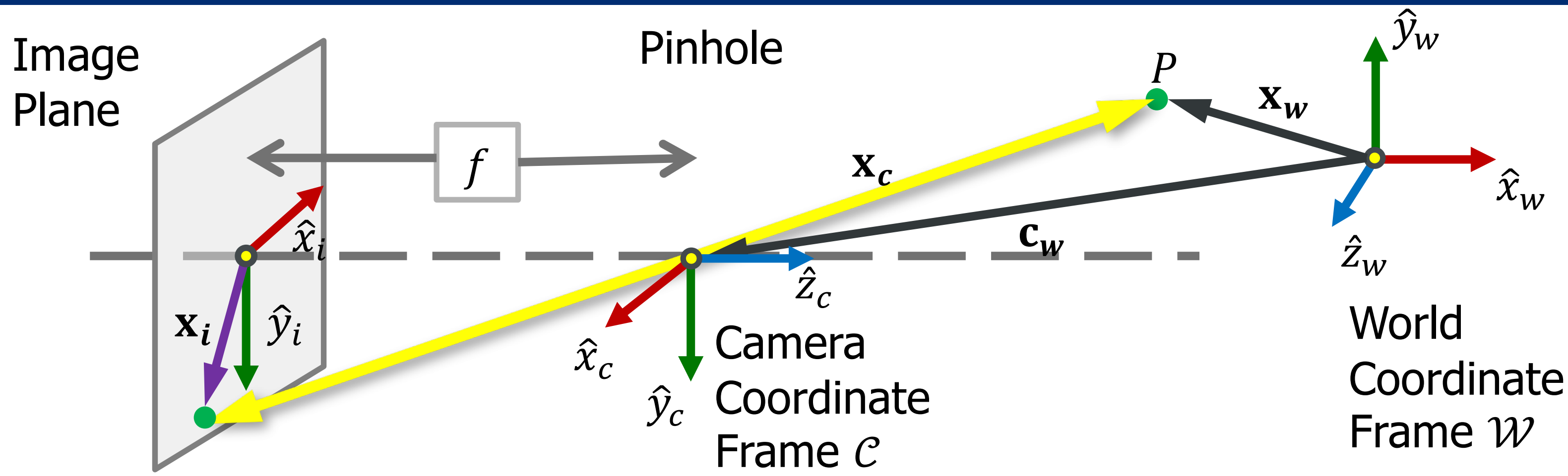
$$\mathbf{x}_c = \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix}$$

Coordinate Transformation

World Coordinates

$$\mathbf{x}_w = \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix}$$

Perspective Projection



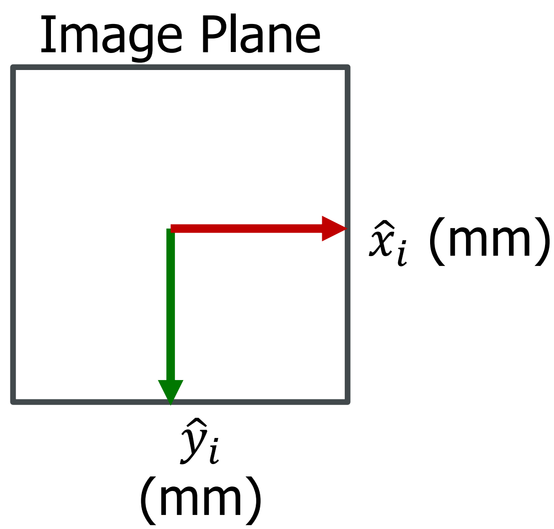
We know that: $\frac{x_i}{f} = \frac{x_c}{z_c}$ and $\frac{y_i}{f} = \frac{y_c}{z_c}$

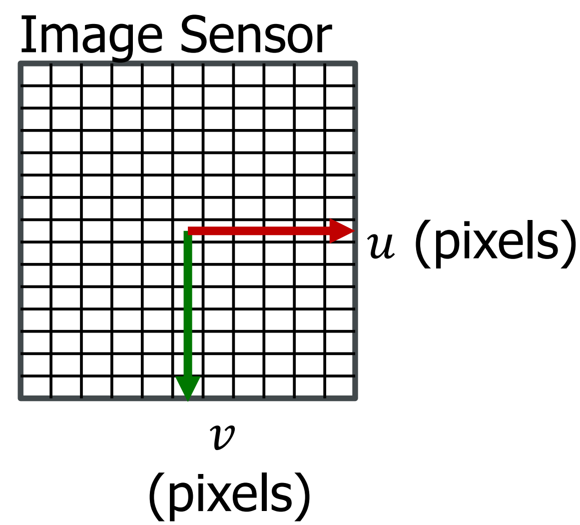
Therefore: $x_i = f\frac{x_c}{z_c}$ and $y_i = f\frac{y_c}{z_c}$

What are the units of x_i?

Image Plane to Image Sensor Mapping

Pixels may be rectangular.



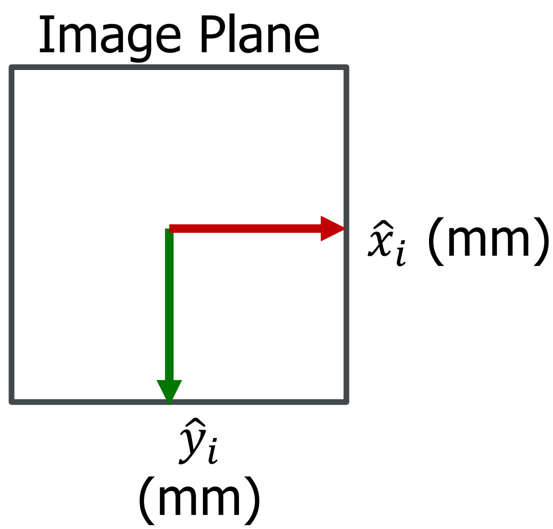


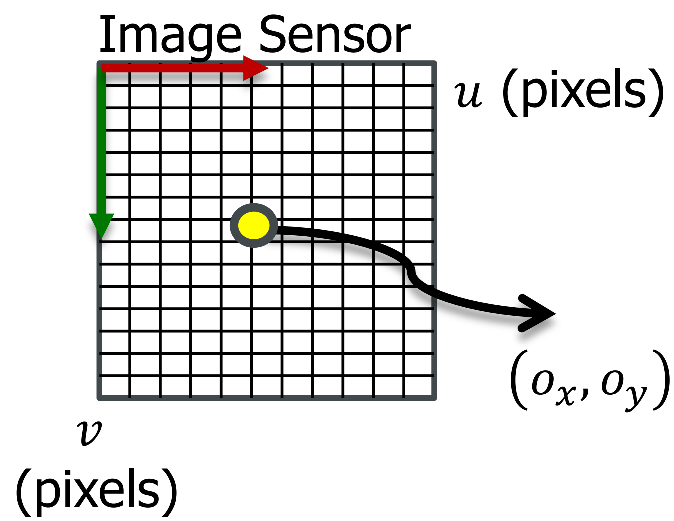
If m_x and m_y are the pixel densities (ex: pixels/mm) in x and y directions respectively, then pixel coordinates are:

$$u = m_x x_i = m_x f \frac{x_c}{z_c} \qquad v = m_y y_i = m_y f \frac{y_c}{z_c}$$

Image Plane to Image Sensor Mapping (cont.)

Pixels may be rectangular.





We usually treat the top-left corner of the image sensor as its origin (easier for indexing). If the optical axis passes through (o_x, o_y) (Principle Point) on the sensor, then:

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

Perspective Projection

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

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

where: $(f_x, f_y) = (m_x f, m_y f)$ are the focal lengths in pixels in x and y directions, respectively.

$(\boldsymbol{f_x}, \boldsymbol{f_y}, \boldsymbol{o_x}, \boldsymbol{o_y})$**: Intrinsic parameters** of the camera.
They represent the **camera's internal geometry**.

Perspective Projection (cont.)

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

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

Equations for Perspective projection are Non-Linear.

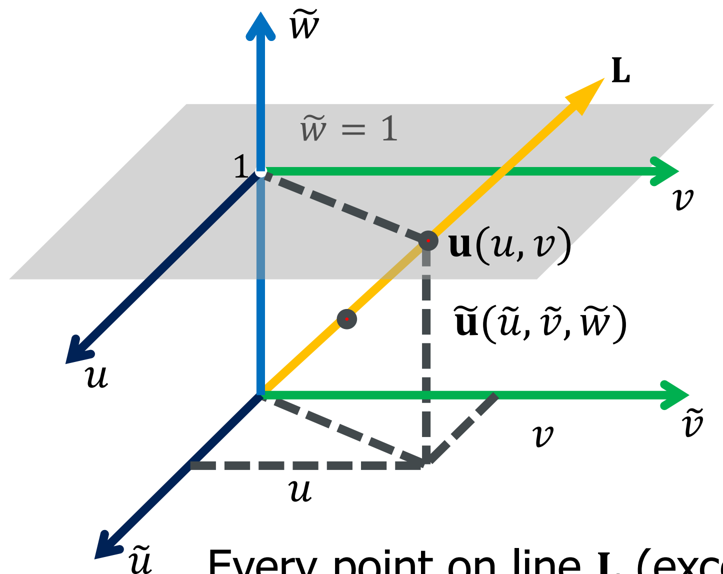
It is often convenient to express them as linear equations.

Homogenous Coordinates

The **homogenous** representation of a 2D point $\mathbf{u} = (u, v)$ is a 3D point $\tilde{\mathbf{u}} = (\tilde{u}, \tilde{v}, \tilde{w})$. The third coordinate $\tilde{w} \neq 0$ is fictitious such that:

$$u = \frac{\tilde{u}}{\tilde{w}} \qquad v = \frac{\tilde{v}}{\tilde{w}}$$

$$\boxed{\mathbf{u} \equiv \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \equiv \begin{bmatrix} \tilde{w}u \\ \tilde{w}v \\ \tilde{w} \end{bmatrix} \equiv \begin{bmatrix} \tilde{u} \\ \tilde{v} \\ \tilde{w} \end{bmatrix} = \tilde{\mathbf{u}}}$$



Every point on line $\mathbf{L}$ (except origin) represents the homogenous coordinate of $\mathbf{u}(u, v)$

Homogenous Coordinates (cont.)

- The homogenous representation of a 3D point $\mathbf{x} = (x, y, z)$ is a 4D point $\tilde{\mathbf{x}} = (\tilde{x}, \tilde{y}, \tilde{z}, \tilde{w})$. The fourth coordinate $\tilde{w} \neq 0$ is fictitious such that:

$$x = \frac{\tilde{x}}{\tilde{w}} \qquad y = \frac{\tilde{y}}{\tilde{w}} \qquad z = \frac{\tilde{z}}{\tilde{w}}$$

$$\boxed{\mathbf{x} \equiv \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \equiv \begin{bmatrix} \tilde{w}x \\ \tilde{w}y \\ \tilde{w}z \\ \tilde{w} \end{bmatrix} \equiv \begin{bmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{z} \\ \tilde{w} \end{bmatrix} = \tilde{\mathbf{x}}}$$

Perspective Projection in Homogenous Coordinates

Perspective projection equations:

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

Homogenous coordinates of (u, v):

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

where: $(u, v) = \left({}^{\tilde{u}}/_{\tilde{w}}, {}^{\tilde{v}}/_{\tilde{w}} \right)$

Intrinsic Matrix

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

Calibration Matrix:

$$K = \begin{bmatrix} f_x & 0 & o_x \\ 0 & f_y & o_y \\ 0 & 0 & 1 \end{bmatrix}$$

Upper Right Triangular Matrix

Intrinsic Matrix:

$$M_{int} = [K|0] = \begin{bmatrix} f_x & 0 & o_x & 0 \\ 0 & f_y & o_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$\boxed{\tilde{\mathbf{u}} = [K|0]\, \tilde{\mathbf{x}}_c = M_{int}\, \tilde{\mathbf{x}}_c}$$

Forward Imaging Model: 3D to 2D

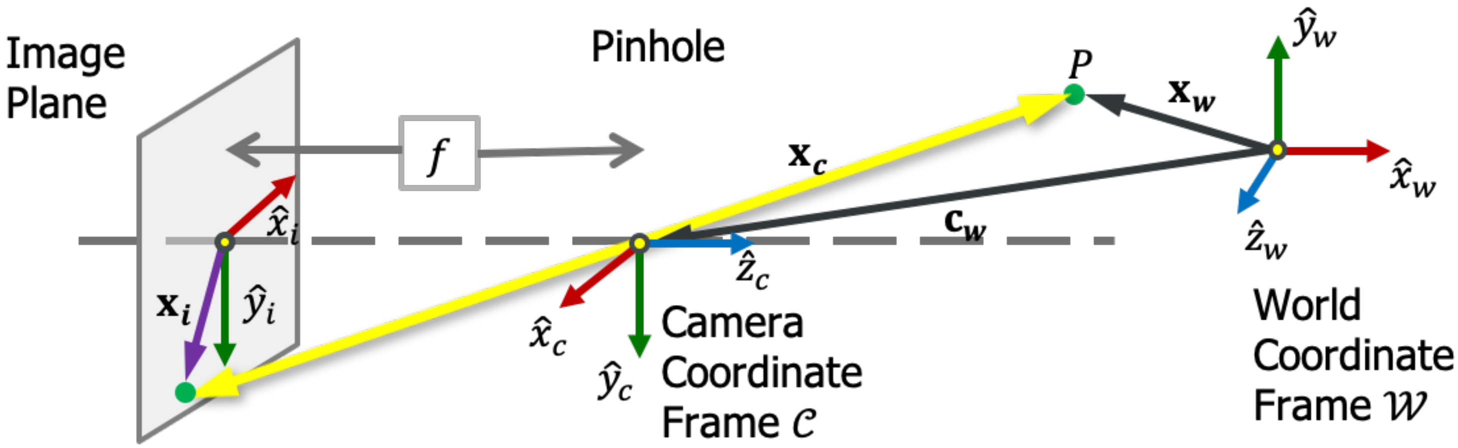
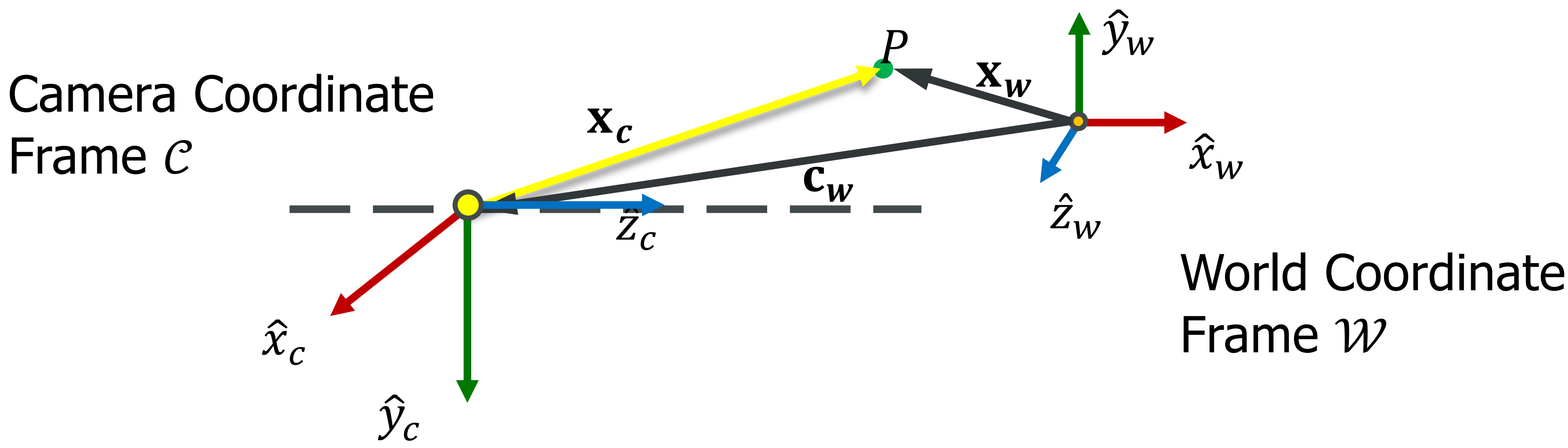


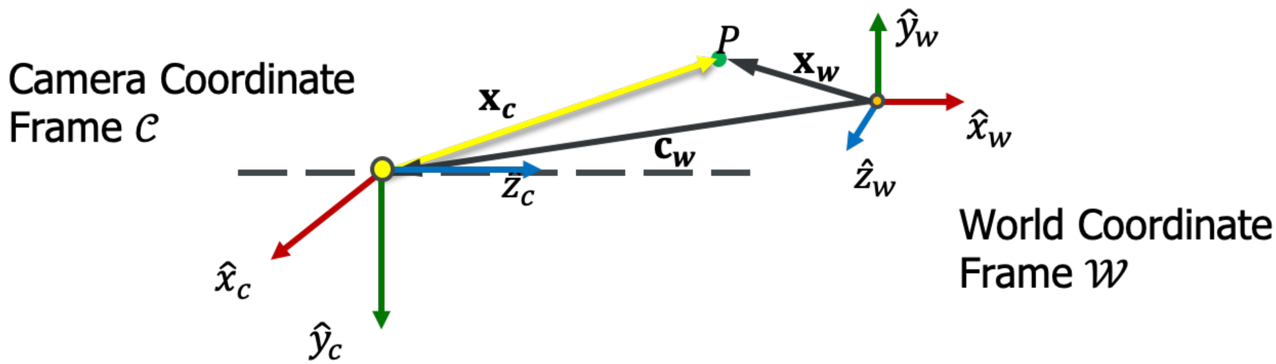
Image Coordinates	Camera Coordinates	World Coordinates
$\mathbf{u} = \begin{bmatrix} u \\ v \end{bmatrix}$ $\xleftarrow{M_{int}}$ Perspective Projection	$\mathbf{x}_c = \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix}$	$\xleftarrow{?}$ Coordinate Transformation $\mathbf{x}_w = \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix}$

Extrinsic Parameters



Position $\mathbf{c}_w$ and Orientation R of the camera in the world coordinate frame $\mathcal{W}$ are the camera's **Extrinsic Parameters.**

World-to-Camera Transformation



Given the extrinsic parameters $(R, \mathbf{c}_w)$ of the camera, the camera-centric location of any point $\mathbf{x}_w$ in the world coordinate frame is:

$$\mathbf{x}_c = R(\mathbf{x}_w - \mathbf{c}_w) = R\mathbf{x}_w - R\mathbf{c}_w = R\mathbf{x}_w + \mathbf{t} \qquad (\mathbf{t} = -R\mathbf{c}_w)$$

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

World-to-Camera Transformation (cont.)

Rewriting using homogenous coordinates:

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

Extrinsic Matrix: $$M_{ext} = \begin{bmatrix} R_{3\times3} & \mathbf{t}_{3\times1} \\ \mathbf{0}_{1\times3} & 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

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

Forward Imaging Model: 3D to 2D (cont.)

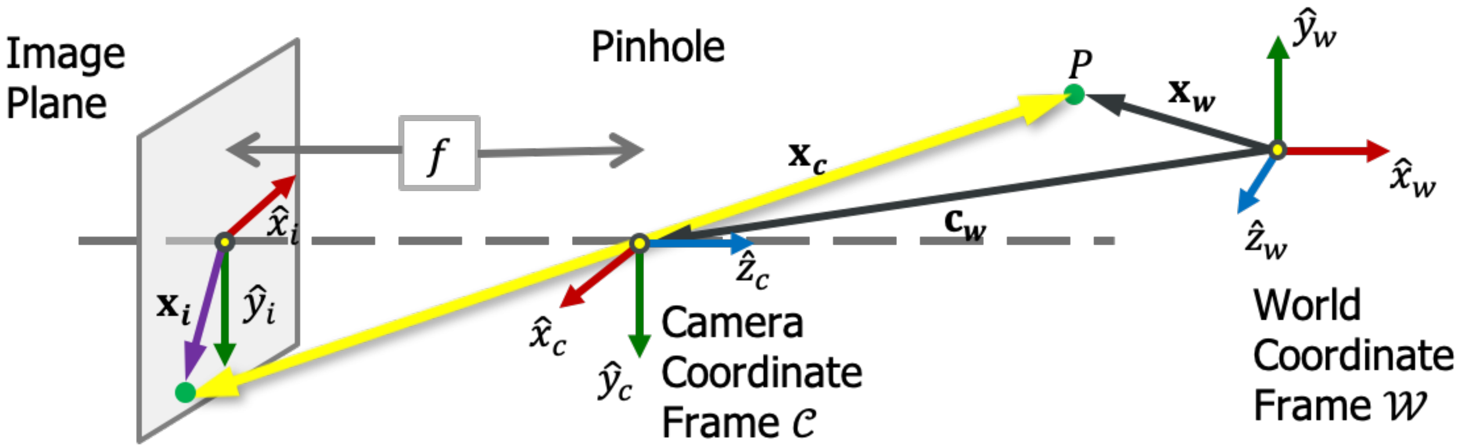


Image Coordinates		Camera Coordinates		World Coordinates
$\mathbf{u} = \begin{bmatrix} u \\ v \end{bmatrix}$	$\xleftarrow{M_{int}}$ Perspective Projection	$\mathbf{x}_c = \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix}$	$\xleftarrow{M_{ext}}$ Coordinate Transformation	$\mathbf{x}_w = \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix}$

Linear Camera Model

Camera to Pixel

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

$$\tilde{\mathbf{u}} = M_{int}\, \tilde{\mathbf{x}}_{\boldsymbol{c}}$$

World to Camera

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

$$\tilde{\mathbf{x}}_{\boldsymbol{c}} = M_{ext}\tilde{\mathbf{x}}_{\boldsymbol{w}}$$

Combining the above two equations, we get the Projection Matrix P:

$$\tilde{\mathbf{u}} = M_{int}M_{ext}\, \tilde{\mathbf{x}}_{\boldsymbol{w}} = P\, \tilde{\mathbf{x}}_{\boldsymbol{w}}$$

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

Scale of Projection Matrix

Projection matrix acts on homogenous coordinates.

We know that: $\begin{bmatrix} \tilde{u} \\ \tilde{v} \\ \tilde{w} \end{bmatrix} \equiv k \begin{bmatrix} \tilde{u} \\ \tilde{v} \\ \tilde{w} \end{bmatrix}$ ($k \neq 0$ is any constant)

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

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

Projection Matrix P needs to be determined only up to a scale factor.

Summary

Camera Calibration and Photogrammetry: Method to find a camera's parameters and estimate 3D structure using two cameras.

- Essential concepts in this lecture:
  - 3D homogeneous transforms
  - Internal camera parameters
  - Full 3D linear projection model

Johns Hopkins Engineering

Computer Vision

Camera Calibration and Photogrammetry

JOHNS HOPKINS
WHITING SCHOOL
of ENGINEERING

Camera Calibration and Photogrammetry

Method to find a camera's parameters and a method to estimate 3D structure using two cameras.

Topics:

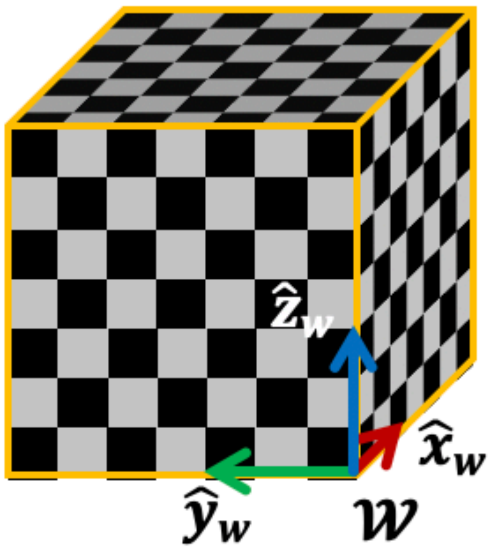
(1) Camera Calibration

Camera Calibration

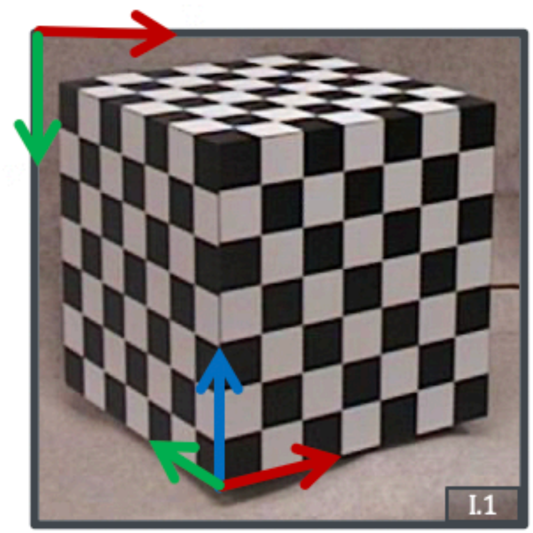
Most vision applications require the knowledge of intrinsic (f_x, f_y, o_x, o_y) and extrinsic $(R, \mathbf{t})$ parameters of the cameras being used.

We "**Calibrate**" the cameras to determine these.

Camera Calibration Procedure (1)



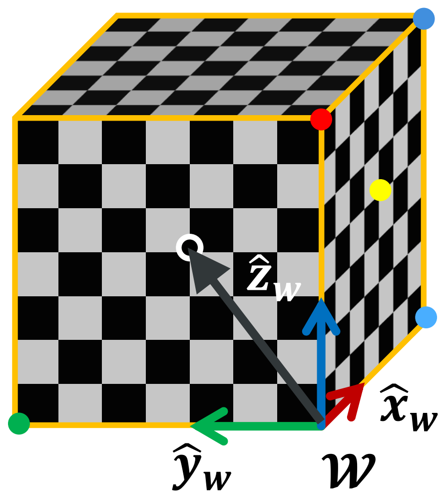
Object whose precise geometry is known

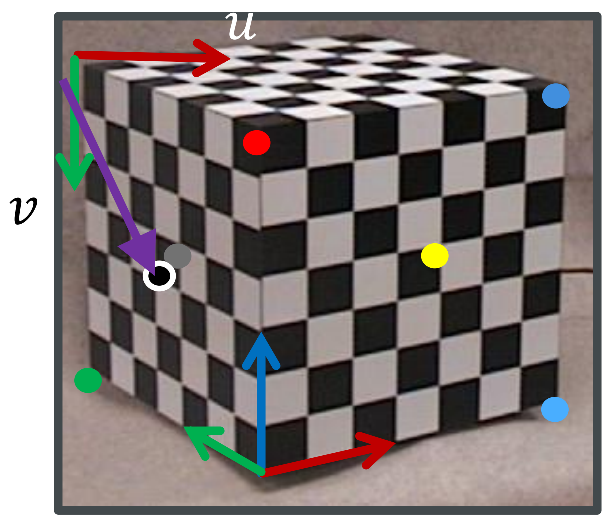


Captured Image

Camera Calibration Procedure (2)

Object whose precise geometry is known





Captured Image

- $\mathbf{x}_w = \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} = \begin{bmatrix} 0 \\ 3 \\ 4 \end{bmatrix}$

- $\mathbf{u} = \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} 56 \\ 115 \end{bmatrix}$

Camera Calibration Procedure (3)

Step 3: For each corresponding point i in scene and image:

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

Expanding the matrix as linear equations:

$$u^{(i)} = \frac{p_{11}x_w^{(i)} + p_{12}y_w^{(i)} + p_{13}z_w^{(i)} + p_{14}}{p_{31}x_w^{(i)} + p_{32}y_w^{(i)} + p_{33}z_w^{(i)} + p_{34}}$$

$$v^{(i)} = \frac{p_{21}x_w^{(i)} + p_{22}y_w^{(i)} + p_{23}z_w^{(i)} + p_{24}}{p_{31}x_w^{(i)} + p_{32}y_w^{(i)} + p_{33}z_w^{(i)} + p_{34}}$$

Camera Calibration Procedure (4)

Step 4: Rearranging the terms:

$$\underbrace{\begin{bmatrix} x_w^{(1)} & y_w^{(1)} & z_w^{(1)} & 1 & 0 & 0 & 0 & 0 & -u_1 x_w^{(1)} & -u_1 y_w^{(1)} & -u_1 z_w^{(1)} & -u_1 \\ 0 & 0 & 0 & 0 & x_w^{(1)} & y_w^{(1)} & z_w^{(1)} & 1 & -v_1 x_w^{(1)} & -v_1 y_w^{(1)} & -v_1 z_w^{(1)} & -v_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_w^{(i)} & y_w^{(i)} & z_w^{(i)} & 1 & 0 & 0 & 0 & 0 & -u_i x_w^{(i)} & -u_i y_w^{(i)} & -u_i z_w^{(i)} & -u_i \\ 0 & 0 & 0 & 0 & x_w^{(i)} & y_w^{(i)} & z_w^{(i)} & 1 & -v_i x_w^{(i)} & -v_i y_w^{(i)} & -v_i z_w^{(i)} & -v_i \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_w^{(n)} & y_w^{(n)} & z_w^{(n)} & 1 & 0 & 0 & 0 & 0 & -u_n x_w^{(n)} & -u_n y_w^{(n)} & -u_n z_w^{(n)} & -u_n \\ 0 & 0 & 0 & 0 & x_w^{(n)} & y_w^{(n)} & z_w^{(n)} & 1 & -v_n x_w^{(n)} & -v_n y_w^{(n)} & -v_n z_w^{(n)} & -v_n \end{bmatrix}}_{\substack{A \\ \text{(Known)}}} \underbrace{\begin{bmatrix} p_{11} \\ p_{12} \\ p_{13} \\ p_{14} \\ p_{21} \\ p_{22} \\ p_{23} \\ p_{24} \\ p_{31} \\ p_{32} \\ p_{33} \\ p_{34} \end{bmatrix}}_{\substack{\mathbf{p} \\ \text{(Unknown)}}} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Step 5: Solve for $\mathbf{p}$: $\boxed{A\,\mathbf{p} = \mathbf{0}}$

Least Squares Solution for P

$$A\,\mathbf{p} = \mathbf{0}$$

If $\overline{\mathbf{p}}$ is a solution, so is $k\overline{\mathbf{p}}$ for any constant k.

But, Projection Matrix P needs to be determined only up to a scale factor. We can assume any scale for $\mathbf{p}$.

Set scale so that: $\|\mathbf{p}\|^2 = 1$

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

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

See Appendix A

Find the eigenvector of A^tA with zero eigenvalue (or "SVD trick" to solve this)

Rearrange solution $\mathbf{p}$ to form the projection matrix P.

Extracting Intrinsic and Rotation Parameters

We know that:

$$P = \boxed{\begin{bmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \\ p_{31} & p_{32} & p_{33} \end{bmatrix}} \begin{matrix} p_{14} \\ p_{24} \\ p_{34} \end{matrix} = \underbrace{\begin{bmatrix} f_x & s & o_x & 0 \\ 0 & f_y & o_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{\text{(Intrinsic)}} \underbrace{\begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{\text{(Extrinsic)}}$$

That is:

$$\begin{bmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \\ p_{31} & p_{32} & p_{33} \end{bmatrix} = \begin{bmatrix} f_x & 0 & o_x \\ 0 & f_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} = KR$$

Given that K is an Upper Right Triangle matrix and R is an Orthonormal matrix, it is possible to "decouple" K and R from their product using RQ factorization.

(See Appendix B)

Extracting Translation Parameters

We know that: $$P = \begin{bmatrix} p_{11} & p_{12} & p_{13} & \boxed{p_{14}} \\ p_{21} & p_{22} & p_{23} & \boxed{p_{24}} \\ p_{31} & p_{32} & p_{33} & \boxed{p_{34}} \end{bmatrix} = \underbrace{\begin{bmatrix} f_x & s & o_x & 0 \\ 0 & f_y & o_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{\text{(Intrinsic)}} \underbrace{\begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{\text{(Extrinsic)}}$$

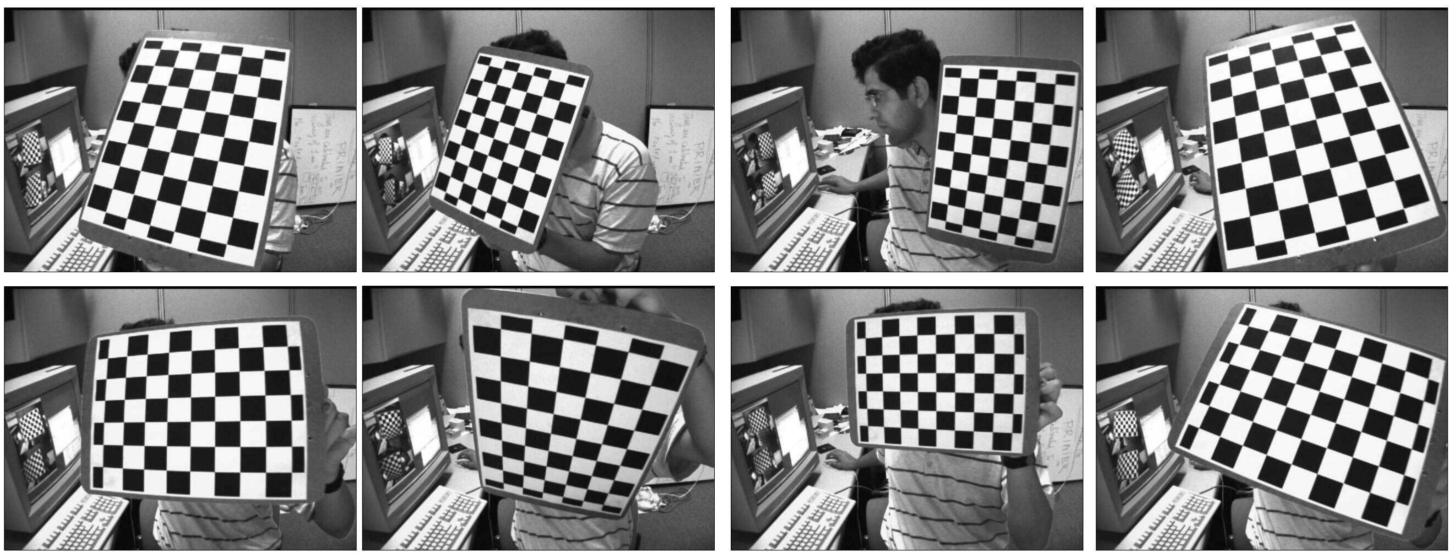
That is: $$\begin{bmatrix} p_{14} \\ p_{24} \\ p_{34} \end{bmatrix} = \begin{bmatrix} f_x & s & o_x \\ 0 & f_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} = K\mathbf{t} \quad = -KR\mathbf{c}_w \qquad (\mathbf{t} = -R\mathbf{c}_w)$$

Therefore: $$\boxed{\mathbf{t} = K^{-1} \begin{bmatrix} p_{14} \\ p_{24} \\ p_{34} \end{bmatrix}} \qquad \boxed{\mathbf{c}_w = -R^T K^{-1} \begin{bmatrix} p_{14} \\ p_{24} \\ p_{34} \end{bmatrix}}$$

Camera Calibration (1)

So what's a practical procedure to calibrate a camera?
Turns out, we don't need a full 3D object – just planar ones(!)

First, we generally collect "checkerboard" images:



Camera Calibration (2)

$$\begin{bmatrix}u\\v\\1\end{bmatrix} = \begin{bmatrix}f_x & s & o_x\\0 & f_y & o_y\\0 & 0 & 1\end{bmatrix}\begin{bmatrix}r_{11} & r_{12} & r_{13} & t_x\\r_{21} & r_{22} & r_{23} & t_y\\r_{31} & r_{32} & r_{33} & t_z\end{bmatrix}\begin{bmatrix}x_w\\y_w\\z_w\\1\end{bmatrix}$$

Since all points lie in a plane: $z_w = 0$

$$\begin{bmatrix}u\\v\\1\end{bmatrix} = \begin{bmatrix}f_x & s & o_x\\0 & f_y & o_y\\0 & 0 & 1\end{bmatrix}\begin{bmatrix}r_{11} & r_{12} & \cancel{r_{13}} & t_x\\r_{21} & r_{22} & \cancel{r_{23}} & t_y\\r_{31} & r_{32} & \cancel{r_{33}} & t_z\end{bmatrix}\begin{bmatrix}x_w\\y_w\\\boxed{0}\\1\end{bmatrix}$$

Thus, we can delete the 3rd column of the Extrinsic parameter matrix

$$\begin{bmatrix}u\\v\\1\end{bmatrix} = \underbrace{\overset{K}{\begin{bmatrix}f_x & s & o_x\\0 & f_y & o_y\\0 & 0 & 1\end{bmatrix}}\overset{\begin{matrix}\mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_3\end{matrix}}{\begin{bmatrix}r_{11} & r_{12} & t_x\\r_{21} & r_{22} & t_y\\r_{31} & r_{32} & t_z\end{bmatrix}}}_{\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2, \mathbf{h}_3]}\begin{bmatrix}x_w\\y_w\\1\end{bmatrix}$$

Camera Calibration (3)

$$\mathbf{h_1} = K\mathbf{r_1}, \mathbf{h_2} = K\mathbf{r_2} \quad \Rightarrow \quad \mathbf{r_1} = K^{-\mathbf{1}}\mathbf{h_1}, \mathbf{r_2} = K^{-\mathbf{1}}\mathbf{h_2}$$

$$\mathbf{r}_1^T\mathbf{r_1} = \mathbf{r}_2^T\mathbf{r_2} = \mathbf{1}, \mathbf{r}_1^T\mathbf{r_2} = \mathbf{0}$$

$$\begin{cases} \mathbf{h}_1^T K^{-\boldsymbol{T}} K^{-\mathbf{1}}\mathbf{h_1} - \mathbf{h}_2^T K^{-\boldsymbol{T}} K^{-\mathbf{1}}\mathbf{h_2} = \mathbf{0} \\ \mathbf{h}_1^T K^{-\boldsymbol{T}} K^{-\mathbf{1}}\mathbf{h_2} = \mathbf{0} \end{cases}$$

Define $B = K^{-\boldsymbol{T}} K^{-\mathbf{1}}$

Note that B is symmetric and positive definite

K can be calculated from B using **Cholesky factorization**

We now have a form that let's us solve **linearly** for B using the two equations from the homography that relates the images of the left and right cameras!

Camera Calibration (4)

$$\mathbf{h_1} = K\mathbf{r_1}, \mathbf{h_2} = K\mathbf{r_2} \quad \Longrightarrow \quad \mathbf{r_1} = K^{-1}\mathbf{h_1}, \mathbf{r_2} = K^{-1}\mathbf{h_2}$$

$$\mathbf{r}_1^T\mathbf{r_1} = \mathbf{r}_2^T\mathbf{r_2} = \mathbf{1}, \mathbf{r}_1^T\mathbf{r_2} = \mathbf{0}$$

$$\begin{cases} \mathbf{h}_1^T K^{-\boldsymbol{T}} K^{-\mathbf{1}}\mathbf{h_1} - \mathbf{h}_2^T K^{-\boldsymbol{T}} K^{-\mathbf{1}}\mathbf{h_2} = \mathbf{0} \\ \mathbf{h}_1^T K^{-\boldsymbol{T}} K^{-\mathbf{1}}\mathbf{h_2} = \mathbf{0} \end{cases}$$

$B = K^{-\boldsymbol{T}} K^{-\mathbf{1}}$ is symmetric and positive definite

Each plane gives us two equations for B

Since B has 6 degrees of freedom (why?), we need at least 3 different homographies (i.e. 3 image pairs)

We need at least 4 points per plane to compute the homography

Camera Calibration (4)

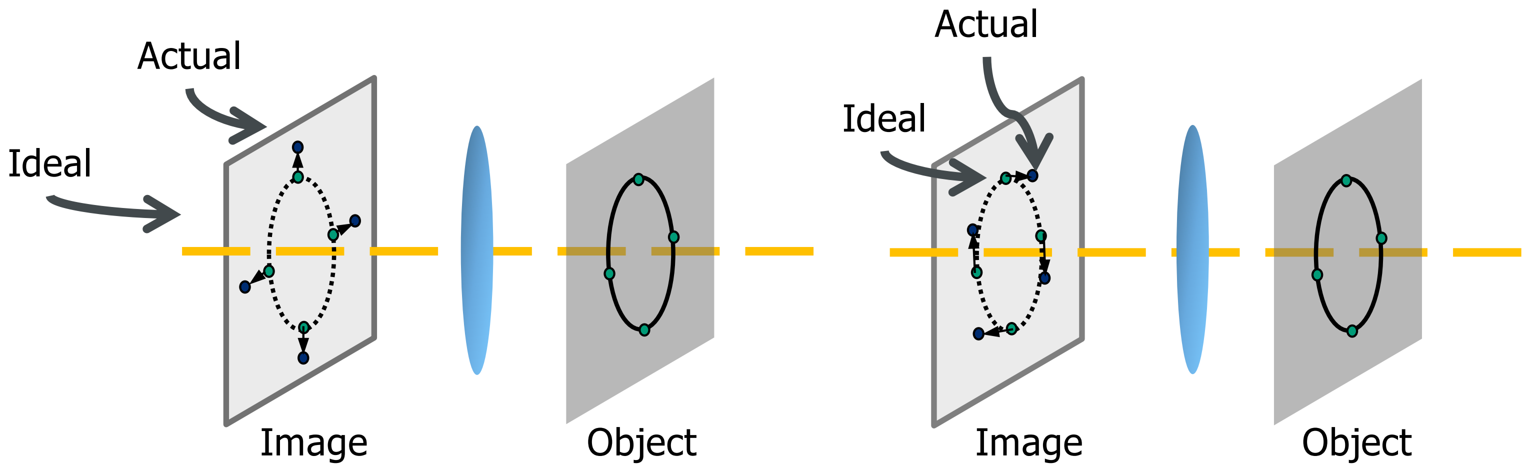
So what's the procedure to calibrate a camera?

Then, we:

- Extract corners in each checkerboard
- Associate each corner to a 3D location ON the checkerboard
- Collect across different views
- Solve for camera intrinsics: focal length (in pixels), principal point (in pixels), distortion parameters

Other Intrinsic Parameters: Distortion

Pinholes do not exhibit image distortions. Lenses do.



Radial distortion | **Tangential** distortion

The mathematical model of the camera will need to incorporate the distortion coefficients.

Distortion Parameters

How do we use the distortion parameters? Given radial distortion coefficients k_1, k_2, k_3 and tangential distortion coefficients p_1 and p_2:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = R \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + t$$

$$x' = \frac{x}{z} \qquad y' = \frac{y}{z}$$

where:

$$x'' = \mathrm{x}'(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + 2p_1 x'y' + p_2(r^2 + 2x'^2)$$
$$y'' = \mathrm{y}'(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + 2p_2 x'y' + p_1(r^2 + 2y'^2)$$
$$r^2 = x'^2 + y'^2$$

finally: $u = f_x x'' + c_x$ v$= f_y y'' + c_y$

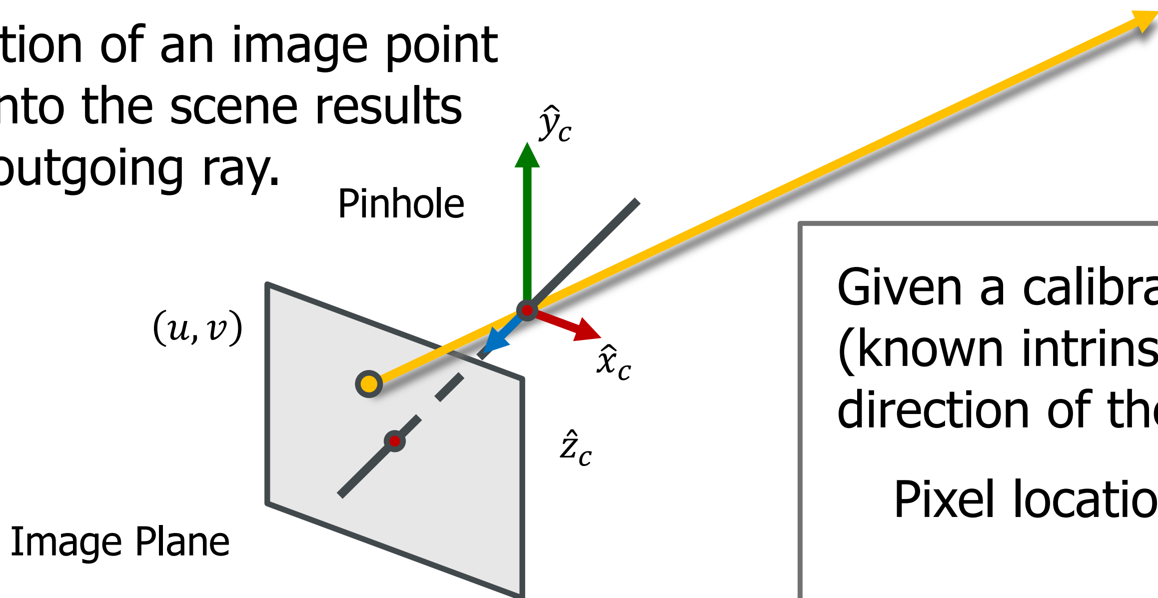
Non-linear Refinement

- Closed-form solution minimized algebraic distance.
- Since full-perspective is a non-linear model
  - Can include distortion parameters (radial, tangential)
  - Use maximum likelihood inference for our estimated parameters.

$$\sum_{i=1}^{n} \sum_{j=1}^{m} ||m_{ij} - \hat{m}(A, R_k, T_k, M_j)||^2$$

Backward Projection: From 2D to 3D

Projection of an image point back into the scene results in an outgoing ray.



Given a calibrated camera (known intrinsics), what is the direction of the ray?

Pixel location: $$u = m_x x_i + o_x$$ $$v = m_y y_i + o_y$$

Image point: $\mathbf{x}_i = (x_i, y_i, f)$

Direction of ray: $\dfrac{-\mathbf{x}_i}{\|\mathbf{x}_i\|}$

Summary

Camera Calibration and Photogrammetry: Method to find a camera's parameters and estimate 3D structure using two cameras.

- Essential concepts in this lecture:
  - Setting up calibration as a series of homographies
  - Extracting camera parameters
  - Including distortion and nonlinear refinement

Johns Hopkins Engineering

Computer Vision

Camera Calibration and Photogrammetry

JOHNS HOPKINS
WHITING SCHOOL
of ENGINEERING