Camera Calibration and Photogrammetry

Introduction to Computational Photography:

EECS 395/495

Northwestern University

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

Forward Imaging Model: 3D to 2D

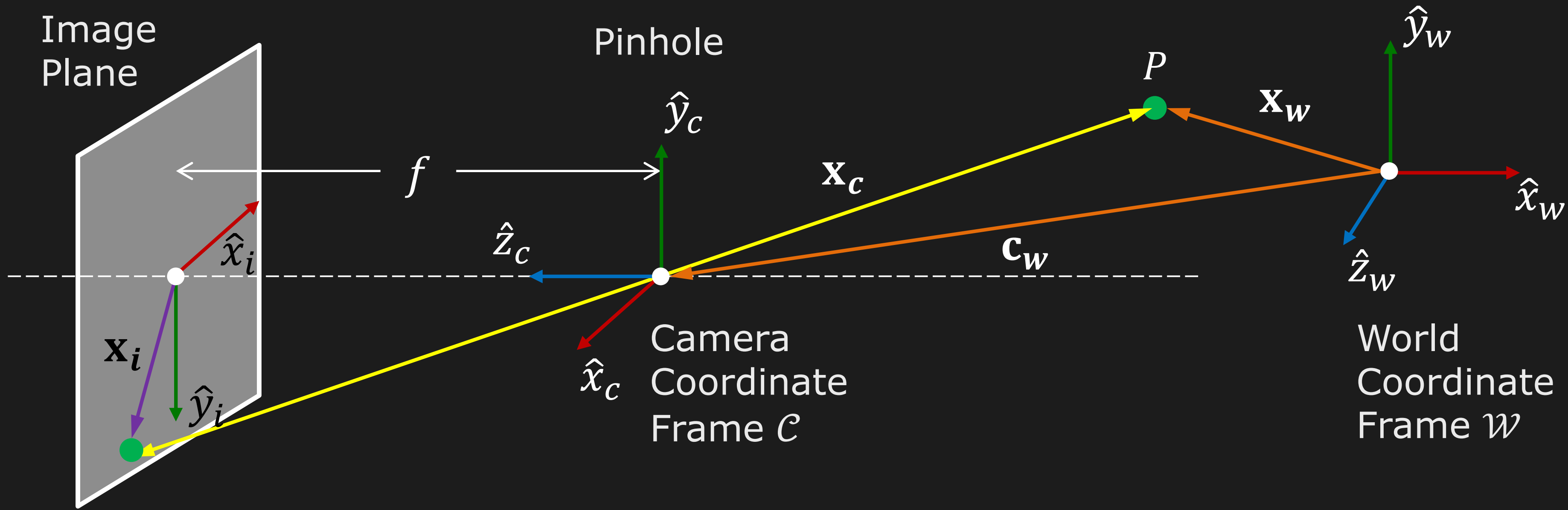


Image Plane Coordinates

Camera Coordinates

World Coordinates

$$\mathbf{x}_i = \begin{bmatrix} x_i \\ y_i \end{bmatrix} \quad \mathbf{x}_c = \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} \quad \mathbf{x}_w = \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix}$$

Perspective Projection

Coordinate Transformation

Forward Imaging Model: 3D to 2D

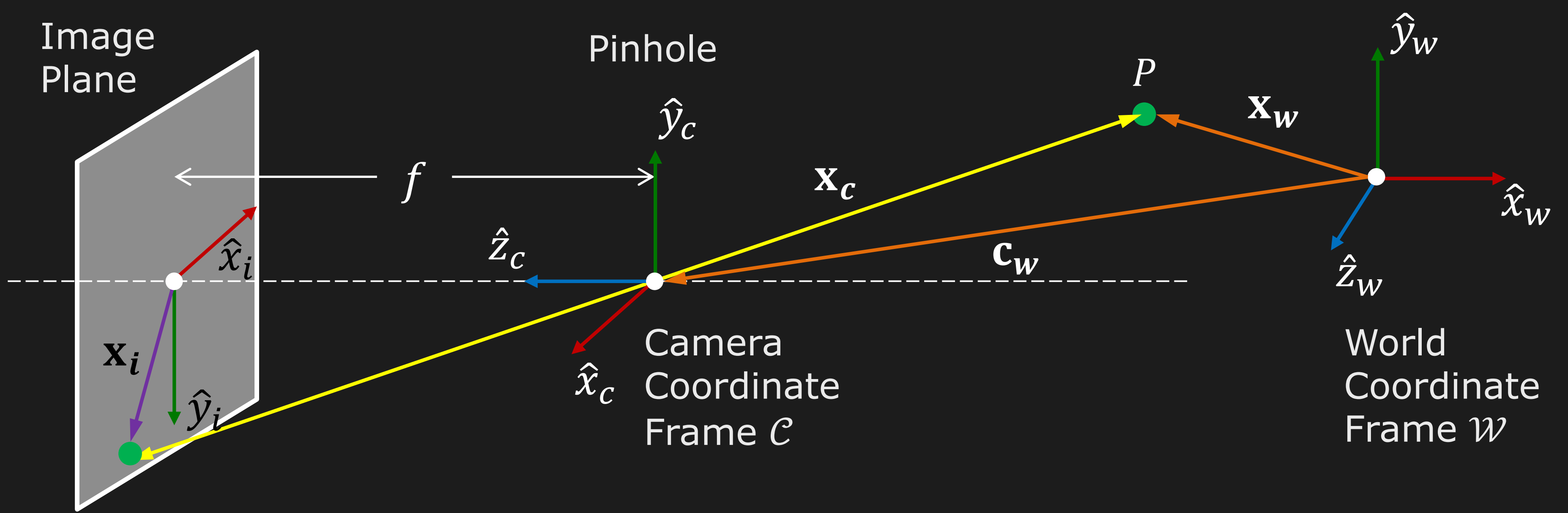


Image Plane Coordinates

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

Camera Coordinates

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

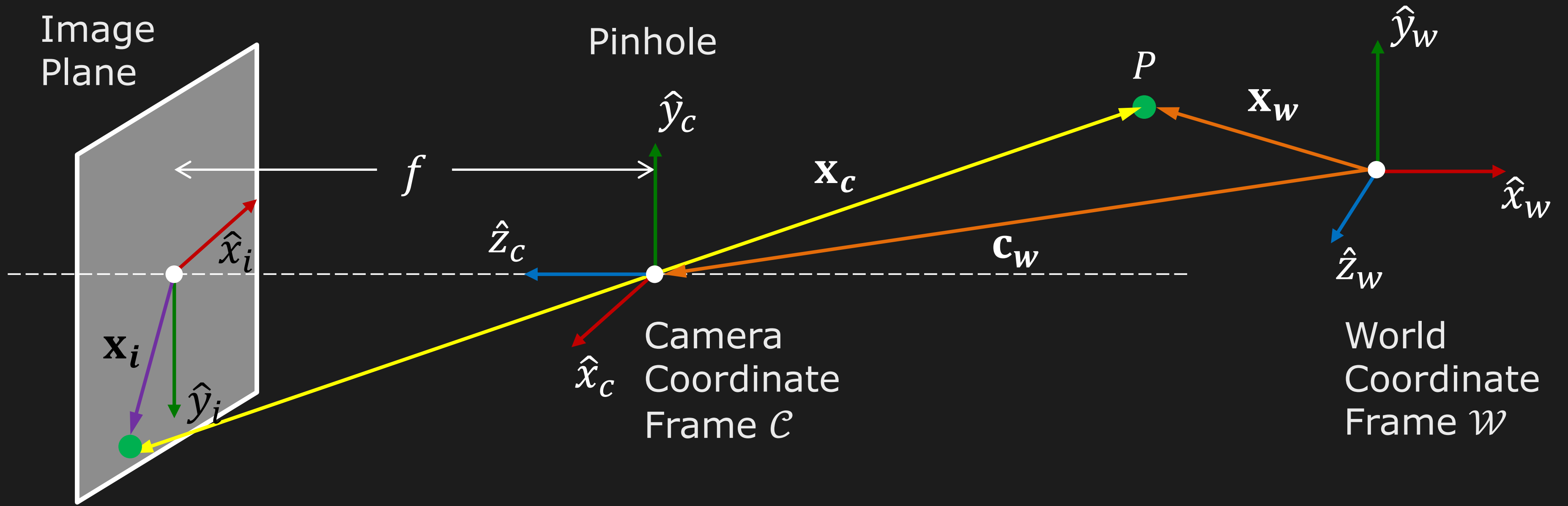
World Coordinates

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

Perspective Projection

Coordinate Transformation

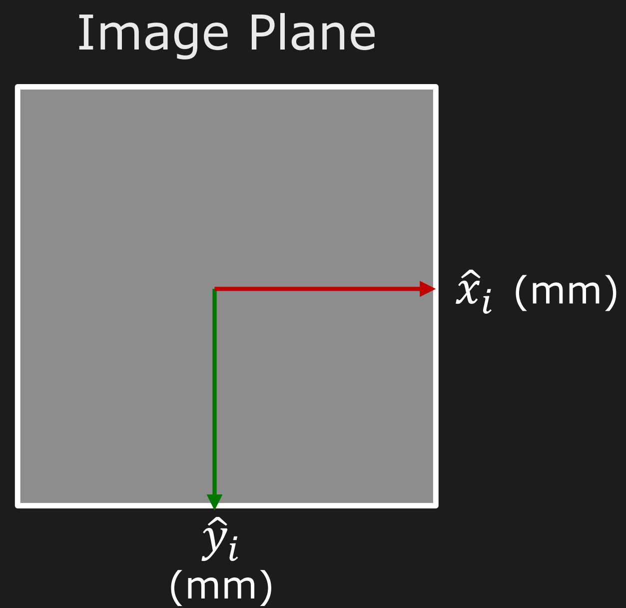
Perspective Projection

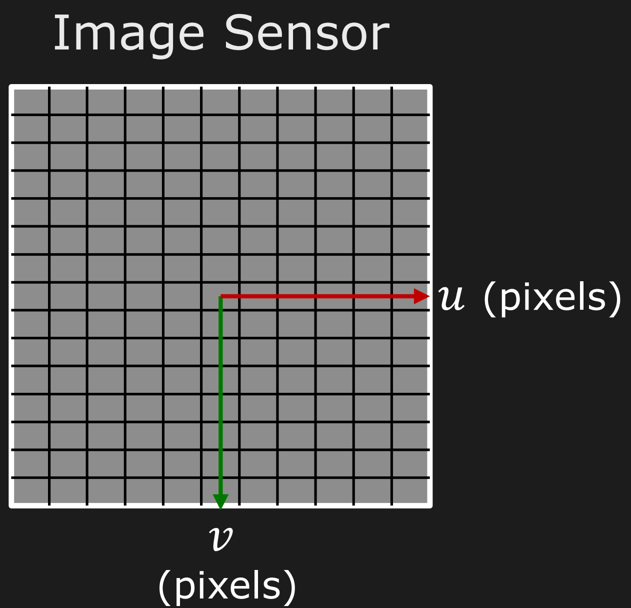


We know that: $\dfrac{x_i}{f} = \dfrac{x_c}{z_c}$ and $\dfrac{y_i}{f} = \dfrac{y_c}{z_c}$

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

Image Plane to Image Sensor Mapping





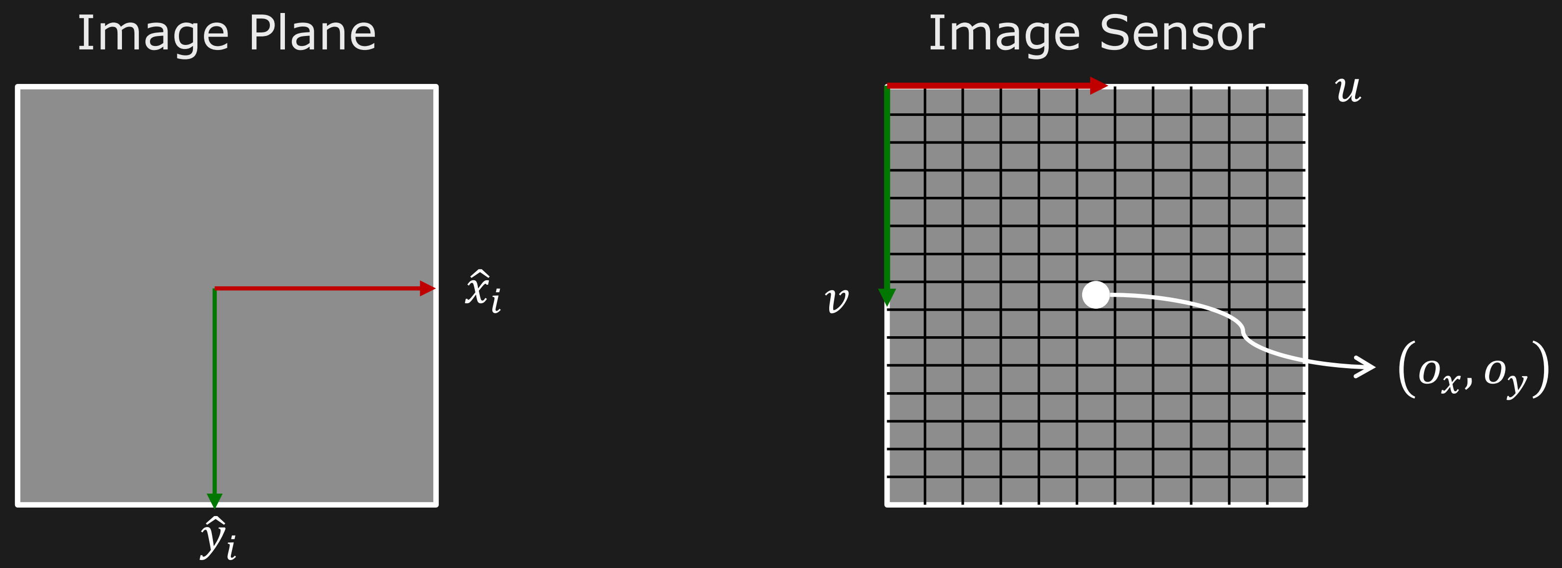
Pixels may be rectangular.

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

$$u = m_x x_i = m_x f \frac{x_c}{z_c}$$

$$v = m_y y_i = m_y f \frac{y_c}{z_c}$$

Image Plane to Image Sensor Mapping



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

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

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

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

(f_x, f_y, o_x, o_y): Intrinsic parameters of the camera.
They represent the camera's internal geometry.

Perspective Projection

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

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

Equations for Perspective projection are Non-Linear.

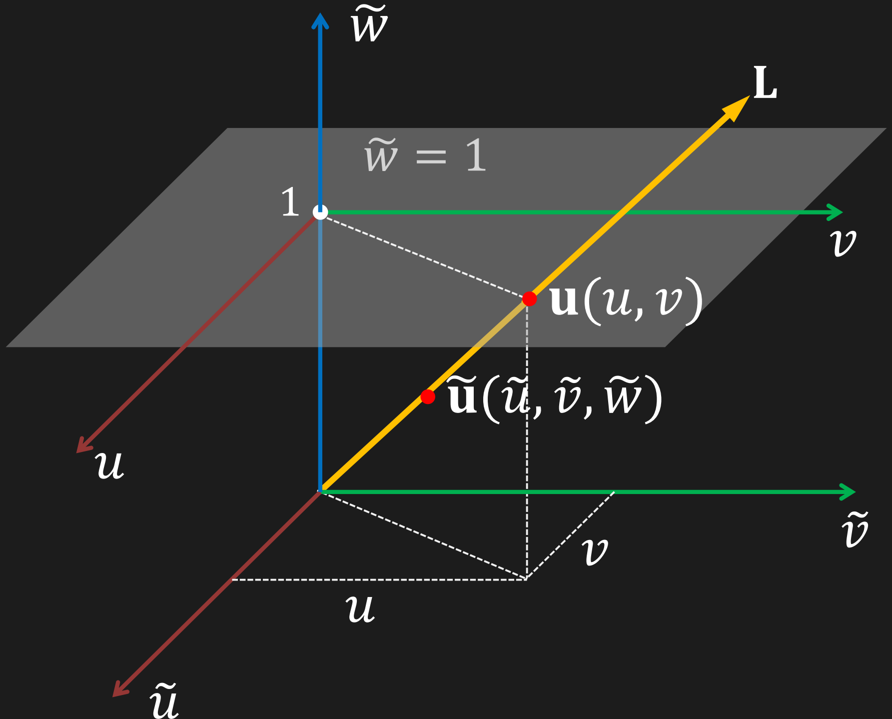
It is often convenient to express them as linear equations.

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

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):

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

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 Triangle Matrix

Intrinsic Matrix:

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

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

Forward Imaging Model: 3D to 2D

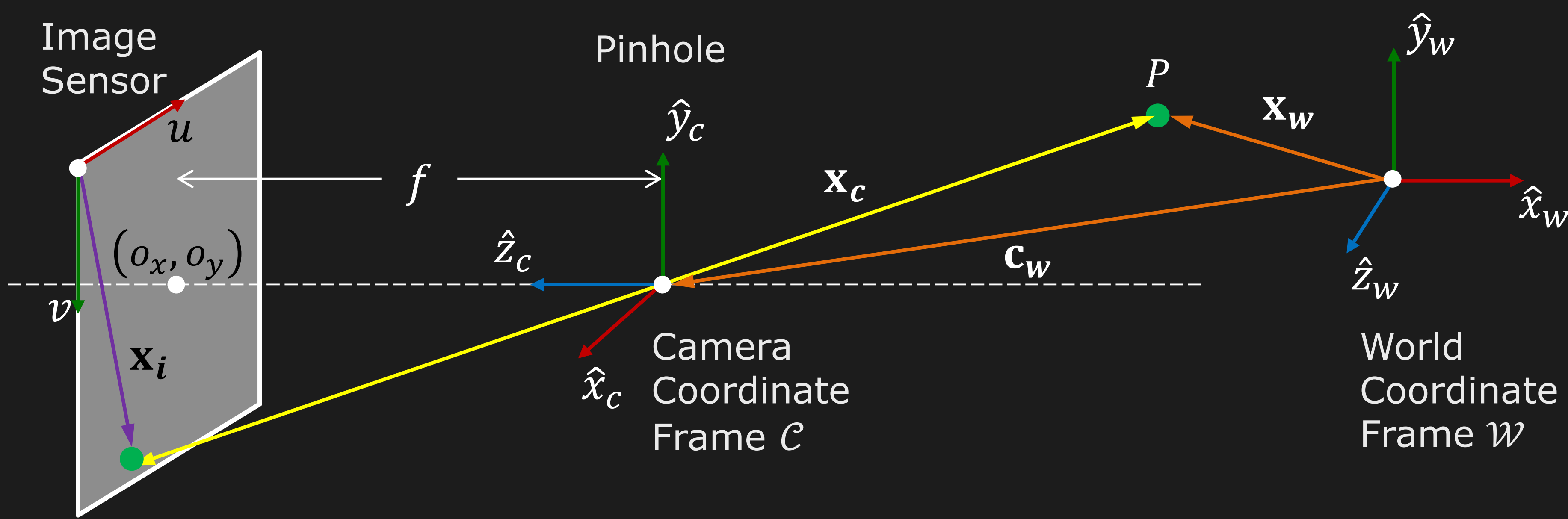
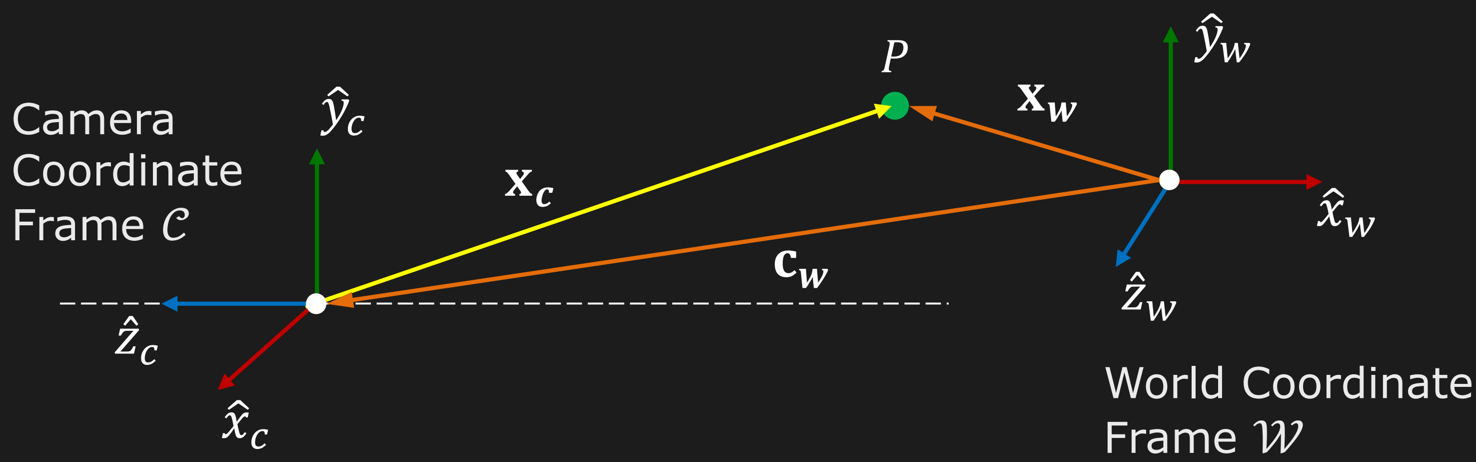


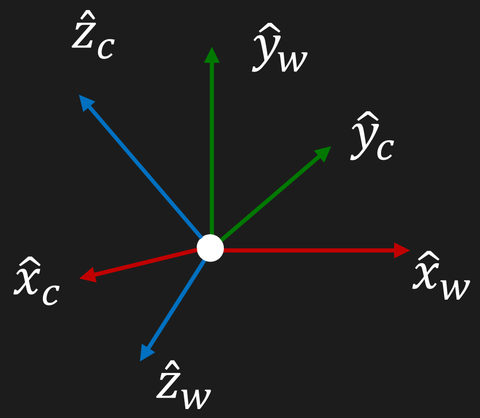
Image Coordinates		Camera Coordinates		World Coordinates
$\mathbf{u} = \begin{bmatrix} u \\ v \end{bmatrix}$	$\xleftarrow{M_{int}}$	$\mathbf{x}_c = \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix}$	$\xleftarrow{?}$	$\mathbf{x}_w = \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix}$
	Perspective Projection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

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.

Rotation Matrix R



$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$$

→ Row 1: Direction of $\hat{x}_c$ in world coordinate frame

→ Row 2: Direction of $\hat{y}_c$ in world coordinate frame

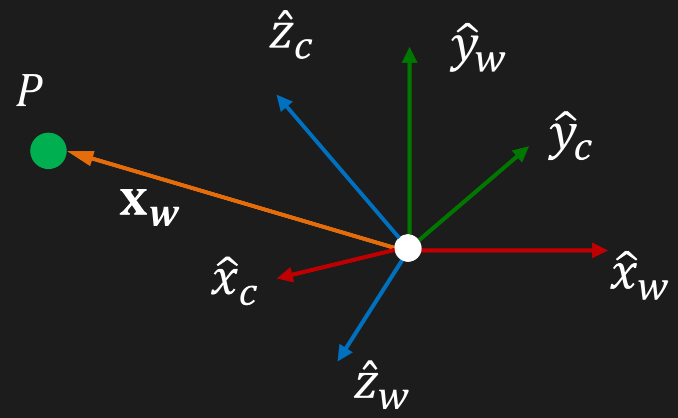
→ Row 3: Direction of $\hat{z}_c$ in world coordinate frame

Orientation/Rotation Matrix R is Orthonormal: The rows and columns of R are unit vectors and are orthogonal (perpendicular) to each other.

$$\boxed{R^{-1} = R^T}$$

Inverse of R = Transpose of R

Rotation Matrix R

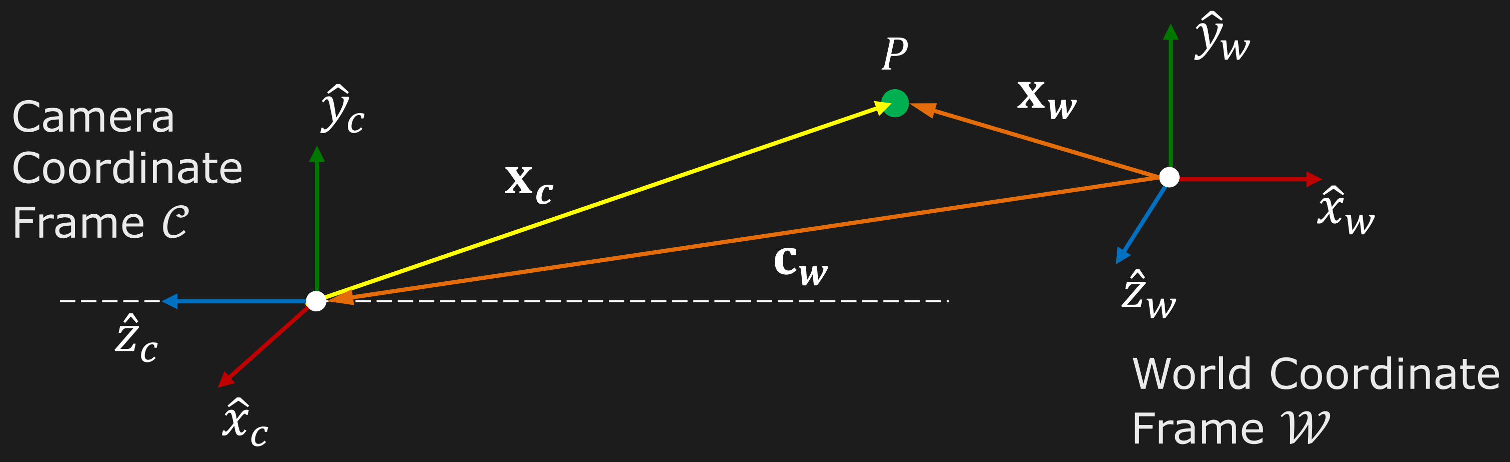


Let point P be at $\mathbf{x}_w$ in the world coordinate frame $\mathcal{W}$.

If the origin of the camera coordinate frame $\mathcal{C}$ were to coincide with $\mathcal{W}$, then the position of point P in the camera coordinate frame is given by:

$$\mathbf{x}_c = R\mathbf{x}_w$$

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

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

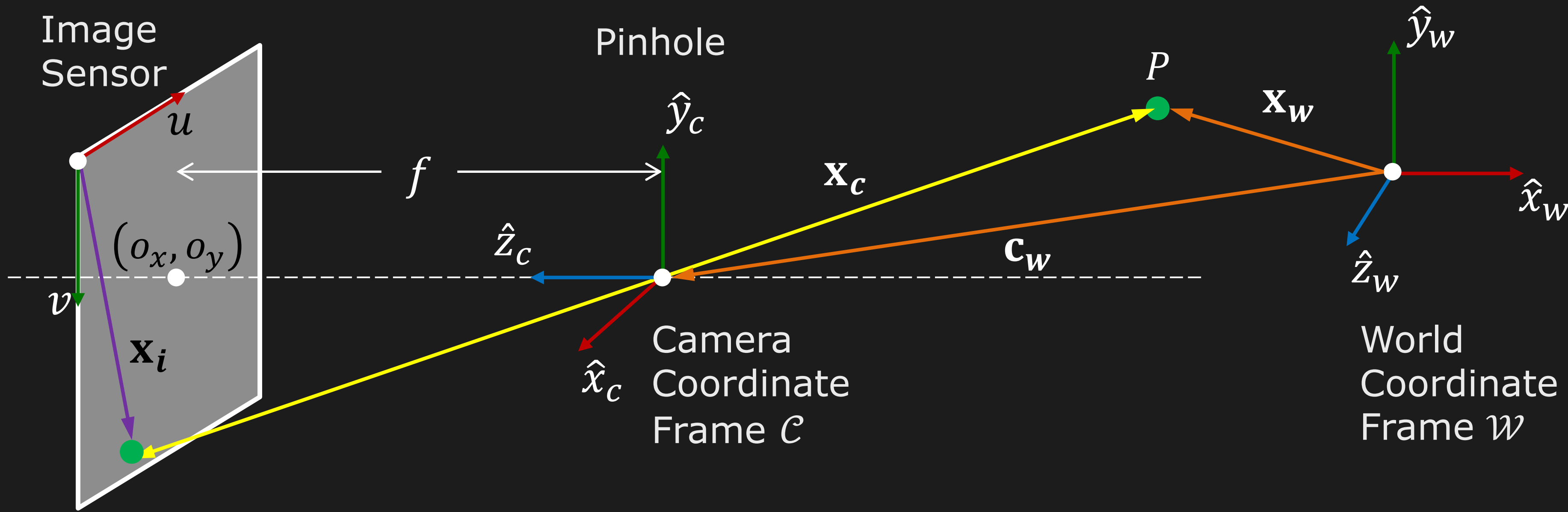


Image Coordinates		Camera Coordinates		World Coordinates
$\mathbf{u} = \begin{bmatrix} u \\ v \end{bmatrix}$	$\xleftarrow{M_{int}}$	$\mathbf{x}_c = \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix}$	$\xleftarrow{M_{ext}}$	$\mathbf{x}_w = \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix}$
	Perspective Projection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

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:

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

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

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.

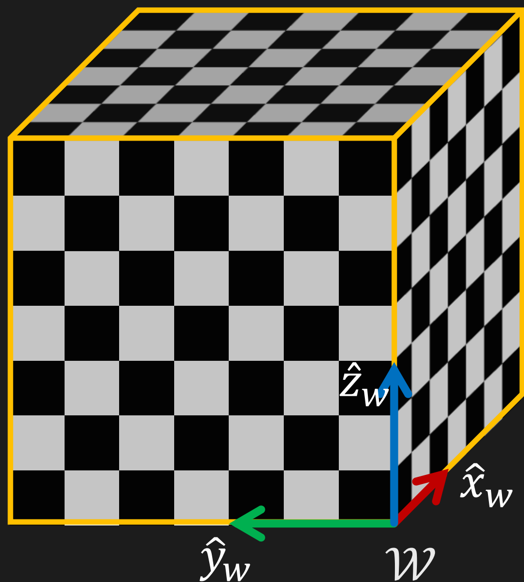
Camera Calibration

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.

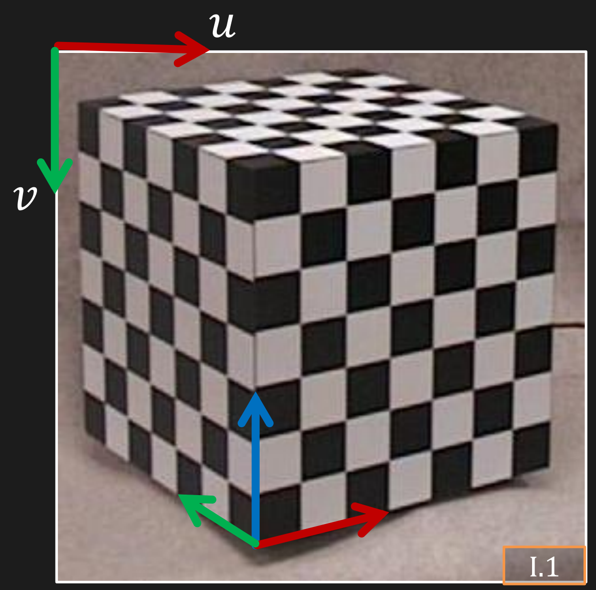
We "Calibrate" the cameras to determine these.

Camera Calibration Procedure

Step 1: Capture an image of an object with known geometry.



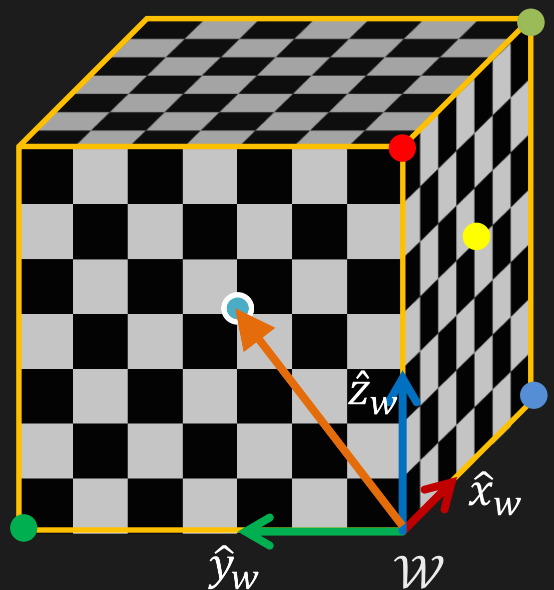
Object whose precise geometry is known



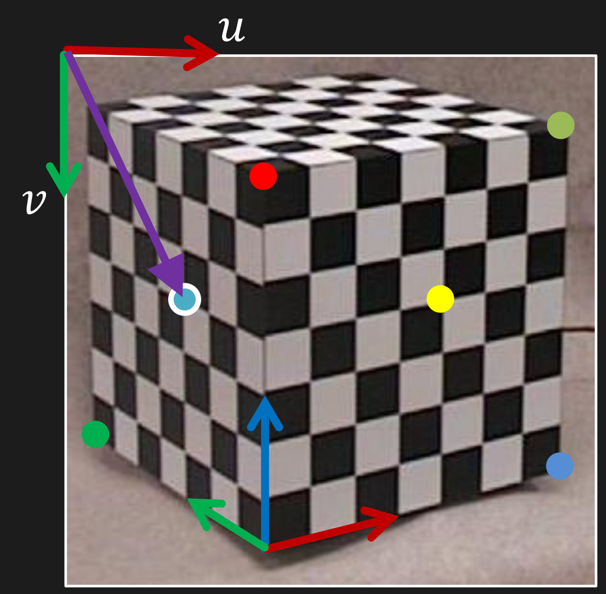
Captured Image

Camera Calibration Procedure

Step 2: Identify correspondence between 3D scene points and captured image.



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

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

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 $\bar{\mathbf{p}}$ is a solution, so is $k\bar{\mathbf{p}}$ 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 for method to solve this constrained linear least squares problem)

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

Extracting Intrinsic and Rotation Parameters

We know that: (Intrinsic) (Extrinsic)

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

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:

(Intrinsic) (Extrinsic)

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

That is:

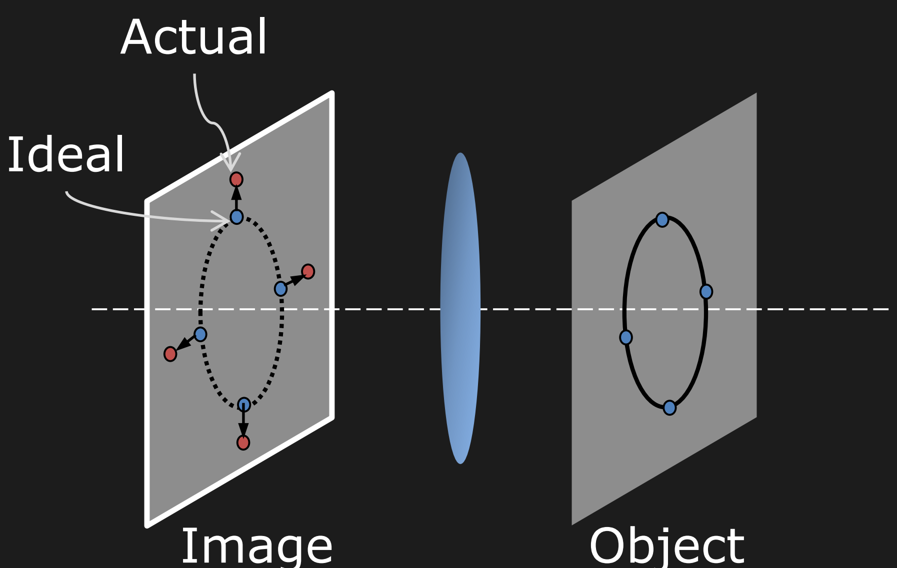
$$\begin{bmatrix} p_{14} \\ p_{24} \\ p_{34} \end{bmatrix} = \begin{bmatrix} f_x & s & o_x \\ 0 & f_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} = K\mathbf{t} = -KR\mathbf{c}_w \qquad (\mathbf{t} = -R\mathbf{c}_w)$$

Therefore:

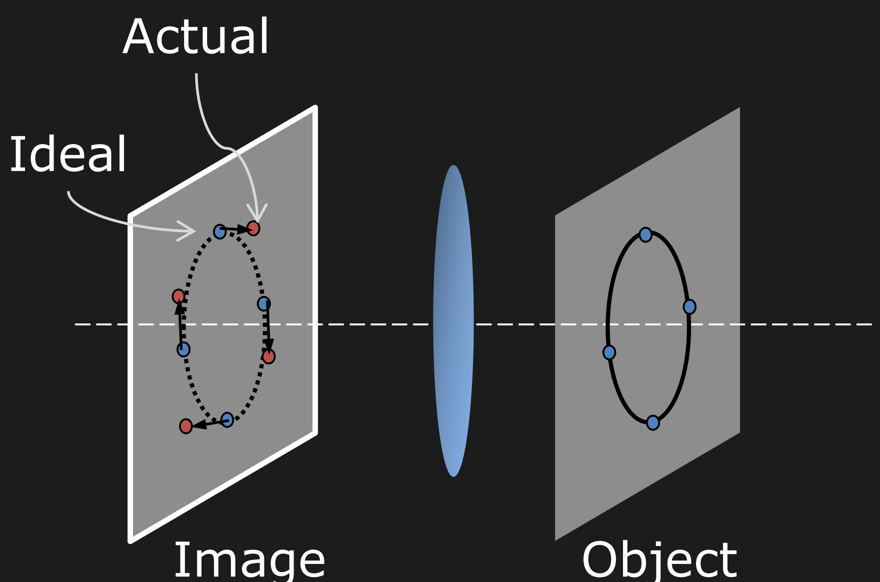
$$\boxed{\mathbf{t} = K^{-1} \begin{bmatrix} p_{14} \\ p_{24} \\ p_{34} \end{bmatrix}} \qquad \boxed{\mathbf{c}_w = -R^T K^{-1} \begin{bmatrix} p_{14} \\ p_{24} \\ p_{34} \end{bmatrix}}$$

Other Intrinsic Parameters: Distortion

Pinholes do not exhibit image distortions. Lenses do.



Radial distortion

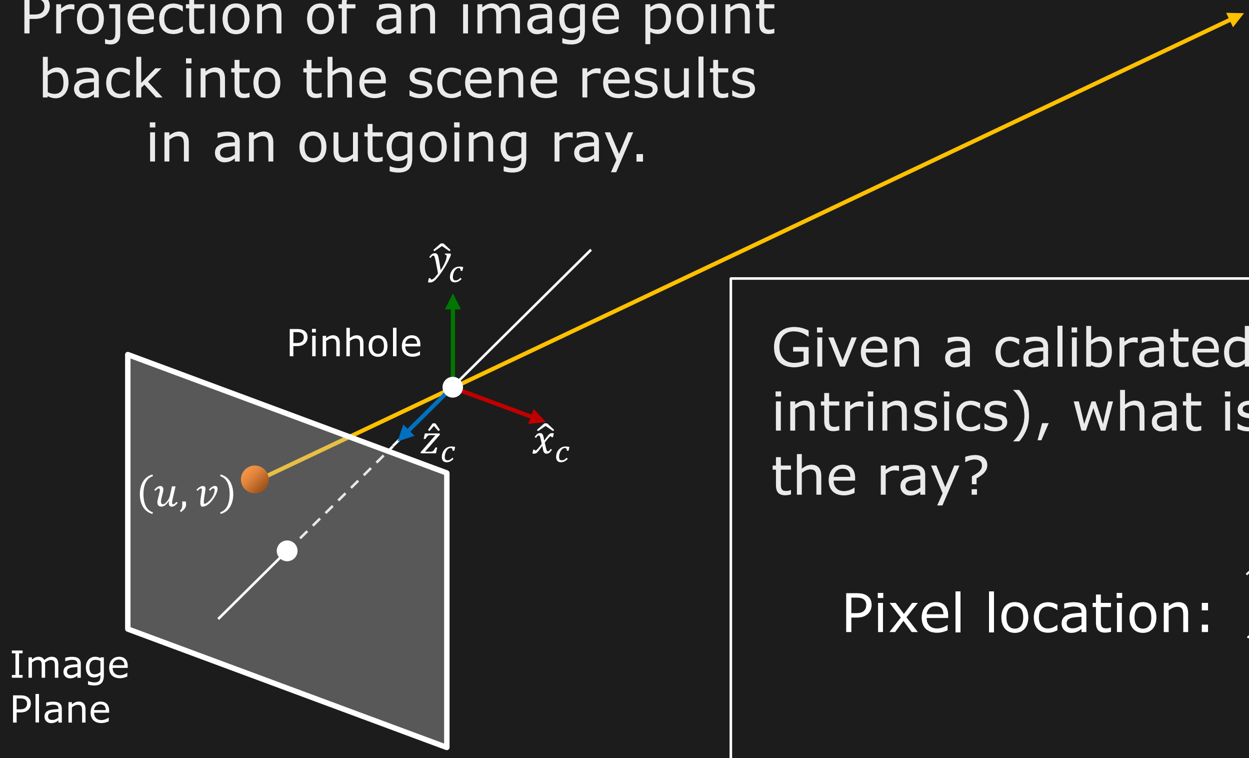


Tangential distortion

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

Backward Projection: From 2D to 3D

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



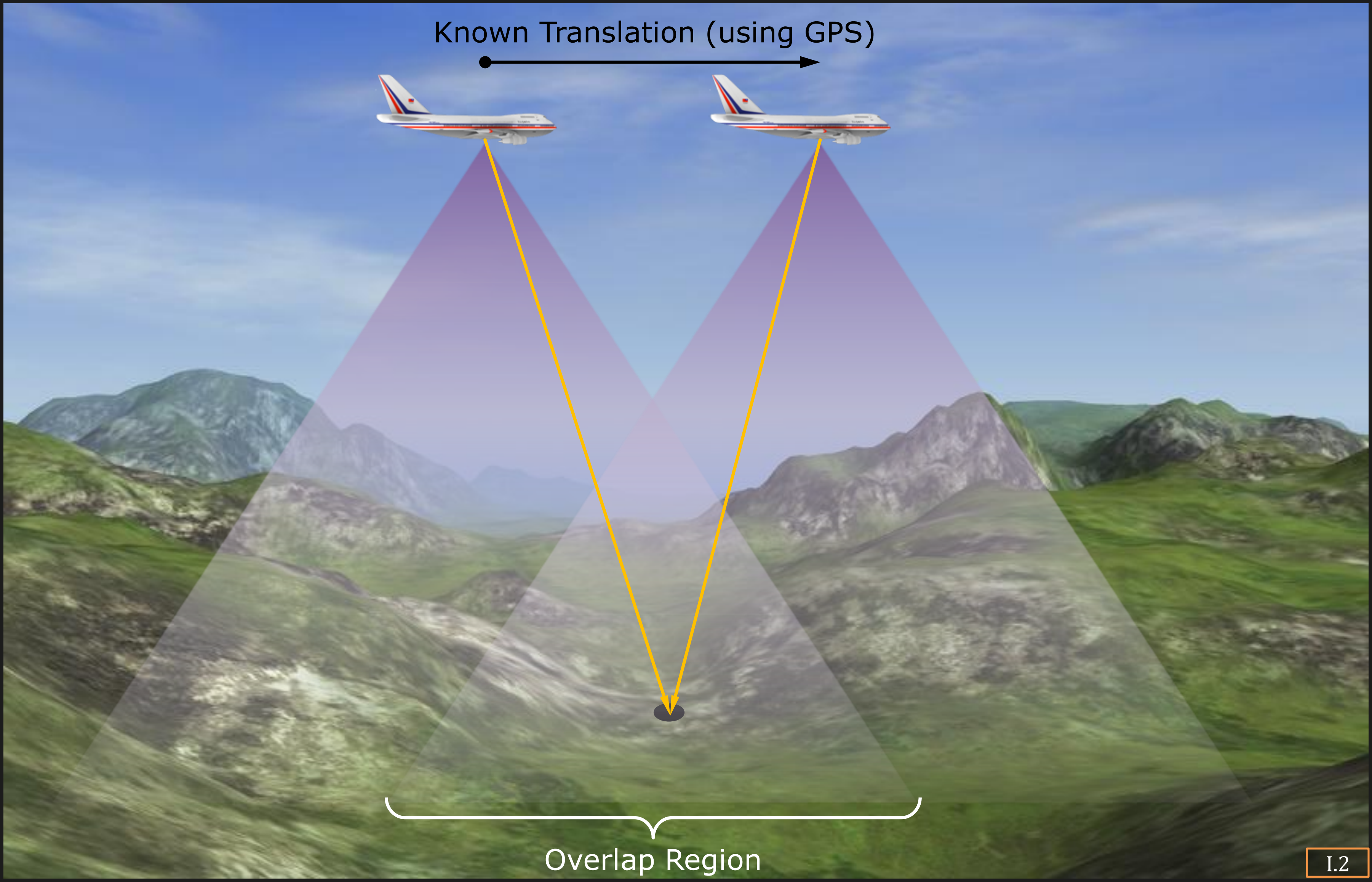
Given a calibrated camera (known intrinsics), what is the direction of the ray?

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

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

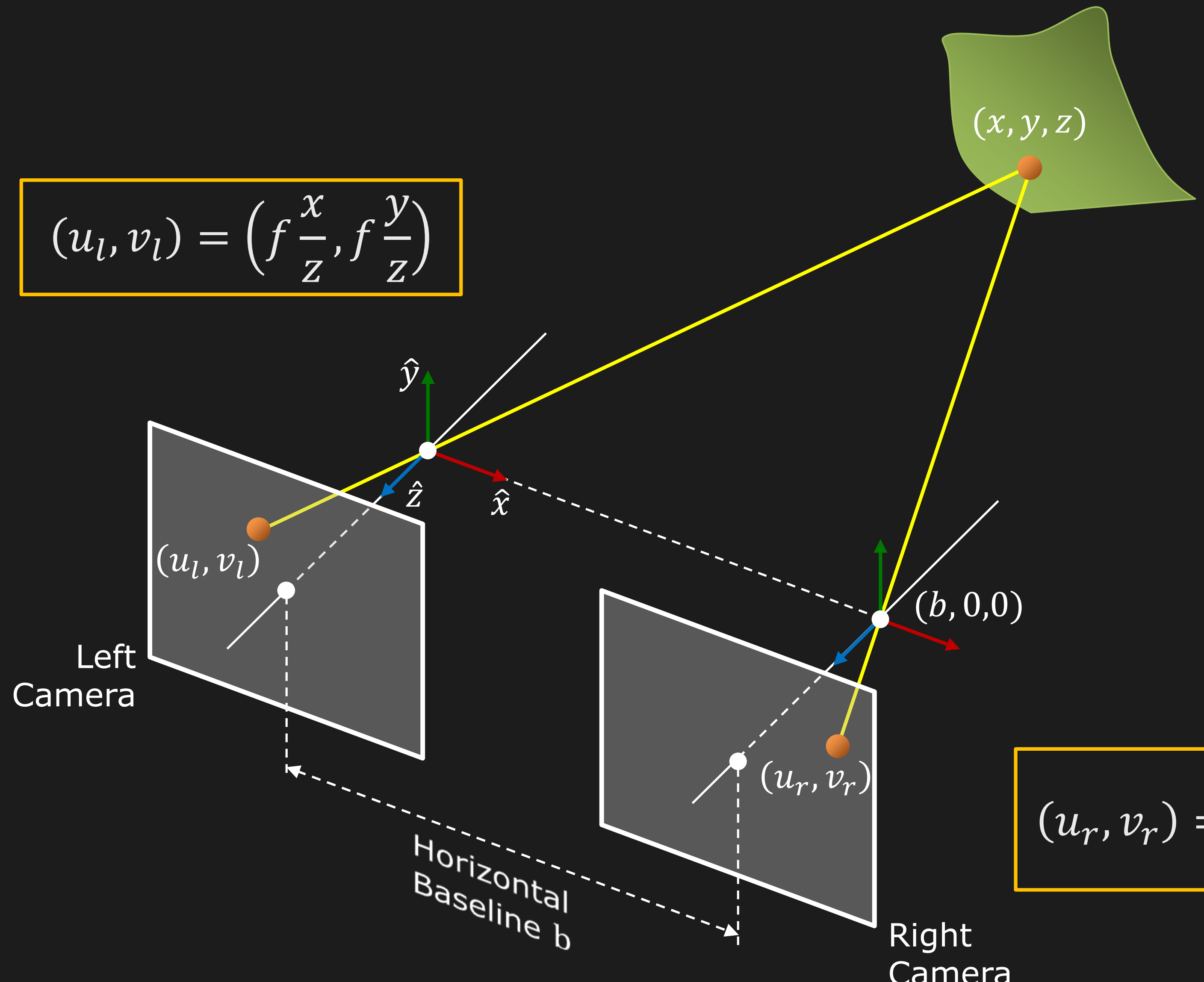
Direction of ray: $\dfrac{-\mathbf{x}_i}{\|\mathbf{x}_i\|}$

Aerial Photogrammetry



Simple Stereo

$$(u_l, v_l) = \left(f\frac{x}{z}, f\frac{y}{z}\right)$$



$$(u_r, v_r) = \left(f\frac{x-b}{z}, f\frac{y}{z}\right)$$

Simple Stereo: Depth and Disparity

For two pixels corresponding to the same scene point in the left and right images:

$$(u_l, v_l) = \left(f\frac{x}{z}, f\frac{y}{z}\right) \qquad (u_r, v_r) = \left(f\frac{x-b}{z}, f\frac{y}{z}\right)$$

Unknown:

Scene point position (x, y, z)

Known:

Pixel positions (u_l, v_l) and (u_r, v_r)

Baseline b from construction (or other calibration methods)

Focal length f from calibration

Simple Stereo: Depth and Disparity

For two pixels corresponding to the same scene point in the left and right images:

$$(u_l, v_l) = \left(f\frac{x}{z}, f\frac{y}{z}\right) \qquad (u_r, v_r) = \left(f\frac{x-b}{z}, f\frac{y}{z}\right)$$

Solving for (x, y, z):

$$x = \frac{b(u_l + u_r)}{2(u_l - u_r)} \qquad y = \frac{b(v_l + v_r)}{2(u_l - u_r)} \qquad \boxed{z = \frac{bf}{(u_l - u_r)}}$$

where $(u_l - u_r)$ is called the Disparity.

Given baseline b and focal length f, the scene depth for each pixel can be found by determining the disparity.

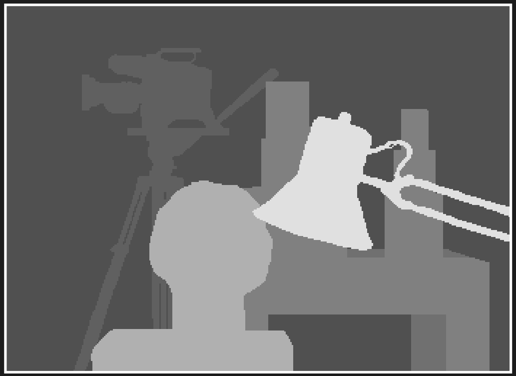
Depth is inversely proportional to Disparity.

Finding Correspondences

Goal: Find the disparity between left and right stereo pairs.

Left Image

Right Image



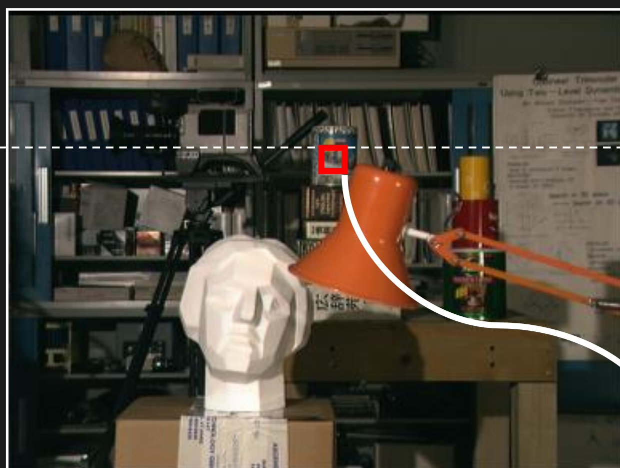
Disparity

From perspective projection: $v_l = v_r = f\frac{y}{z}$

Corresponding scene points lie on the same horizontal scan line.

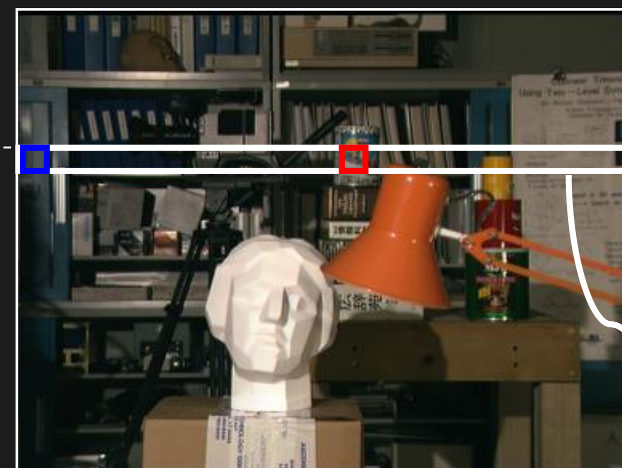
Window based Methods

Determine Disparity using Template Matching



Template Window T

Left Camera Image E_l



Search Scan Line L

Right Camera Image E_r

Disparity, $d = u_l - u_r$

Similarity Metrics for Template Matching

Find pixel $(k, l) \in L$ with Minimum Sum of Absolute Differences:

$$SAD(k,l) = \sum_{(i,j)\in T} |E_l(i,j) - E_r(i+k, j+l)|$$

Find pixel $(k, l) \in L$ with Minimum Sum of Squared Differences:

$$SSD(k,l) = \sum_{(i,j)\in T} |E_l(i,j) - E_r(i+k, j+l)|^2$$

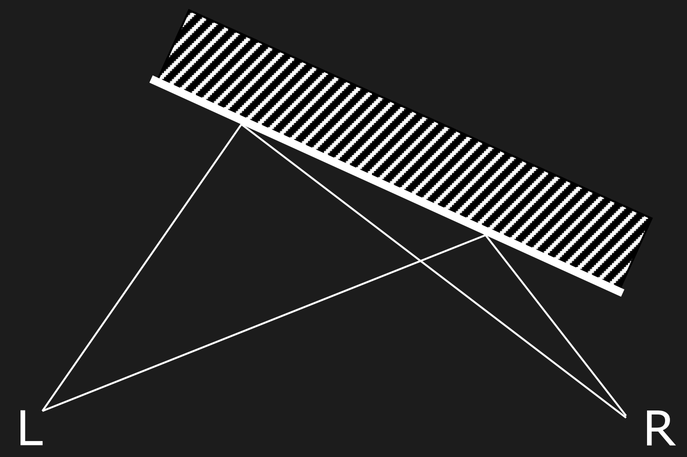
Find pixel $(k, l) \in L$ with Minimum Normalized Cross-Correlation:

$$NCC(k,l) = \frac{\sum_{(i,j)\in T} |E_l(i,j) - E_r(i+k, j+l)|^2}{\sqrt{\sum_{(i,j)\in T} |E_l(i,j)|^2 \sum_{(i,j)\in T} |E_r(i+k, j+l)|^2}}$$

(See Image Processing Lecture 1)

Issues with Window Based Methods

Sensitive to foreshortening effects

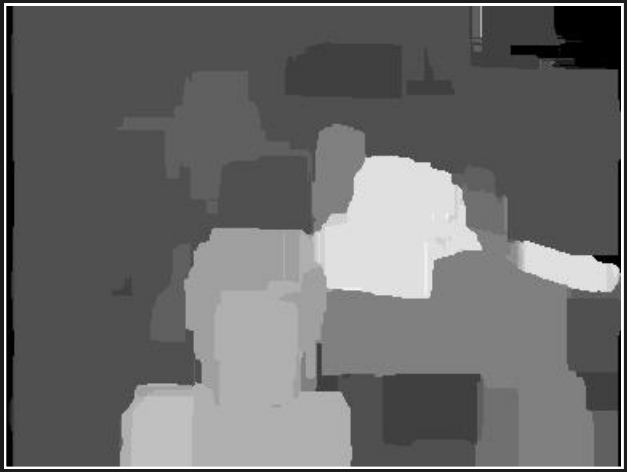


An inclined area will have different amounts of foreshortening in the two images. This affects correlation. (Try Warping)

Issues with Window Based Methods

How large should the window be?

Window size = 5 pixels



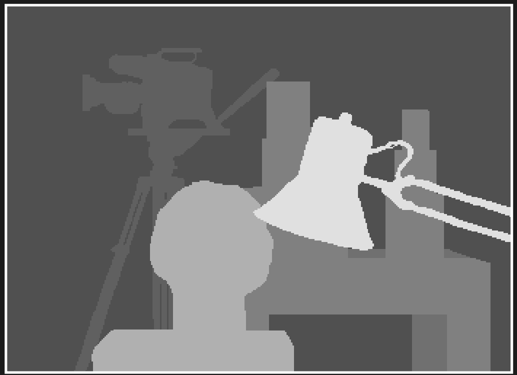
Window size = 30 pixels

Adaptive Window Method Solution: For each point, match using windows of multiple sizes and use the disparity that is a result of the best similarity measure.

Stereo Matching Results

Left Image

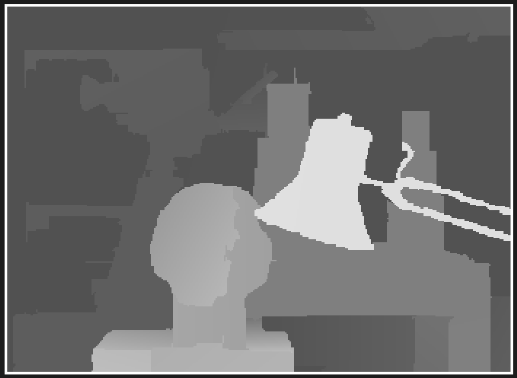
Right Image



Ground Truth

SSD (Window size=21)

SSD – Adaptive Window



State of the Art

Appendix A: Least Squares Solution for P

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

We know that:

$$\|A\mathbf{p}\|^2 = (A\mathbf{p})^T(A\mathbf{p}) = \mathbf{p}^T A^T A\mathbf{p} \qquad \text{and} \qquad \|\mathbf{p}\|^2 = \mathbf{p}^T\mathbf{p} = 1$$

Create a Loss function $L(\mathbf{p})$ and find $\mathbf{p}$ that minimizes it.

$$\min_{\mathbf{p}}\{L(\mathbf{p}) = \mathbf{p}^T A^T A\mathbf{p} + \lambda(\mathbf{p}^T\mathbf{p} - 1)\}$$

Taking derivatives w.r.t $\mathbf{p}$ and λ: $\boxed{A^T A\mathbf{p} + \lambda\mathbf{p} = 0}$ Eigenvalue Problem

Clearly, eigenvector $\mathbf{p}$ with smallest eigenvalue λ of matrix $A^T A$ minimizes the loss function $L(\mathbf{p})$.

Appendix B: RQ Matrix Factorization

Definition: Any Invertible Matrix A can be uniquely decomposed as a product of Upper Right Triangle matrix R and Orthonormal matrix Q.

$$\boxed{A = RQ}$$

Note that according to our camera model (projection matrix) notation:

$$\underbrace{\begin{bmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \\ p_{31} & p_{32} & p_{33} \end{bmatrix}}_{P_{3\times 3}} = \underbrace{\begin{bmatrix} f_x & s & o_x \\ 0 & f_y & o_y \\ 0 & 0 & 1 \end{bmatrix}}_{K} \underbrace{\begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}}_{R}$$

where K is the upper right triangle matrix and R is the orthonormal matrix.

Appendix B: RQ Matrix Factorization

Step 1:

Let $A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \\ p_{31} & p_{32} & p_{33} \end{bmatrix}$

Step 2:

Let $R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_x & s_x \\ 0 & -s_x & c_x \end{bmatrix}$ where $c_x = \frac{a_{33}}{\sqrt{{a_{32}}^2 + {a_{33}}^2}}$ $s_x = \frac{a_{32}}{\sqrt{{a_{32}}^2 + {a_{33}}^2}}$

Set $A = AR_x$ (This makes $a_{32} = 0$)

Appendix B: RQ Matrix Factorization

Step 3:

Let $R_y = \begin{bmatrix} c_y & 0 & -s_y \\ 0 & 1 & 0 \\ s_y & 0 & c_y \end{bmatrix}$ where $c_y = \frac{a_{33}}{\sqrt{a_{31}{}^2 + a_{33}{}^2}}$ $\quad s_y = \frac{-a_{31}}{\sqrt{a_{31}{}^2 + a_{33}{}^2}}$

Set $A = AR_y$ (This makes $a_{31} = 0$)

Step 4:

Let $R_z = \begin{bmatrix} c_z & s_z & 0 \\ -s_z & c_z & 0 \\ 0 & 0 & 1 \end{bmatrix}$ where $c_z = \frac{a_{22}}{\sqrt{a_{21}{}^2 + a_{22}{}^2}}$ $\quad s_z = \frac{a_{21}}{\sqrt{a_{21}{}^2 + a_{22}{}^2}}$

$\boxed{\text{Set } K = AR_z}$ (This makes $a_{21} = 0$)

Step 5: $\boxed{R = R_z^T R_y^T R_x^T}$

References: Textbooks

Robot Vision (Chapter 13)
Horn, B. K. P., MIT Press

Computer Vision: A Modern Approach (Chapter 10)
Forsyth, D and Ponce, J., Prentice Hall

Multiple View Geometry (Chapters 8-10)
Hartley, R. and Zisserman, A., Cambridge University Press

Computer Vision: Algorithms and Applications (Chapter 7)
Szeliski, R., Springer

An introduction to 3D Computer Vision (Chapter 3)
Cyganek, B., Siebert, J. P., Wiley Pub

Image Credits

I.1 http://www.cs.rochester.edu/~sanders/cali_box/box_offlights.jpg

I.2 http://www.orderlymayhem.com/images/terrain_scene2.jpg

I.3 http://vision.middlebury.edu/stereo/data/scenes2001/data/imagehtml/tsukuba.html