ECE 484
Lecture 6
3D Vision

Sayan Mitra

“The eye is an extension of the brain.”

Cameras now span 12+ orders of magnitude in scale and 6+ orders in time:

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

Problems

- Reconstructing the 3D structure of the scene from images
- Reconstructing 6D pose of camera from images
- Camera calibration

Input: image with points in pixels

Output:

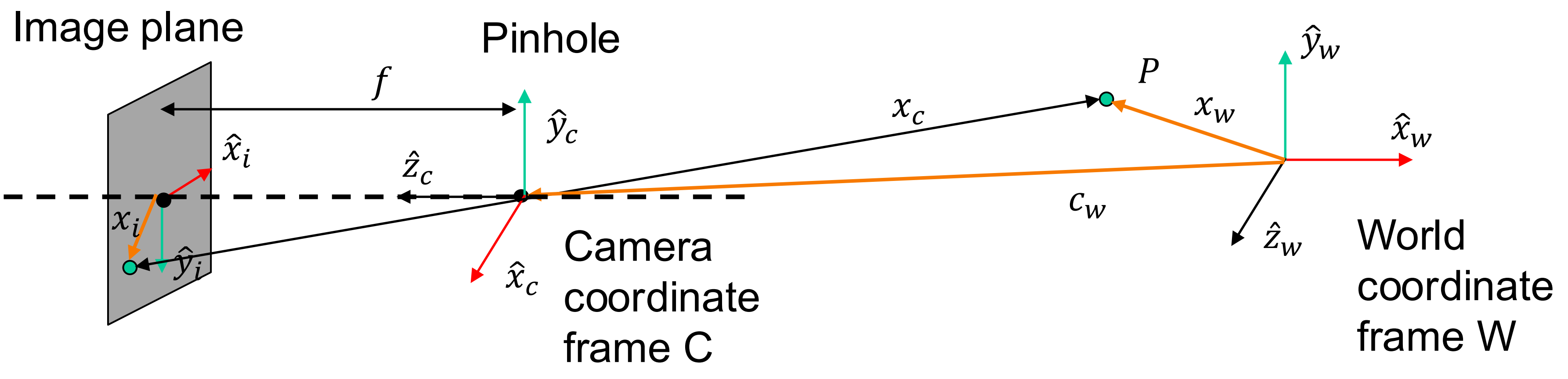
- Position of objects in millimeters in world frame
- Camera pose
- Camera intrinsic parameter

Outline

Forward camera model

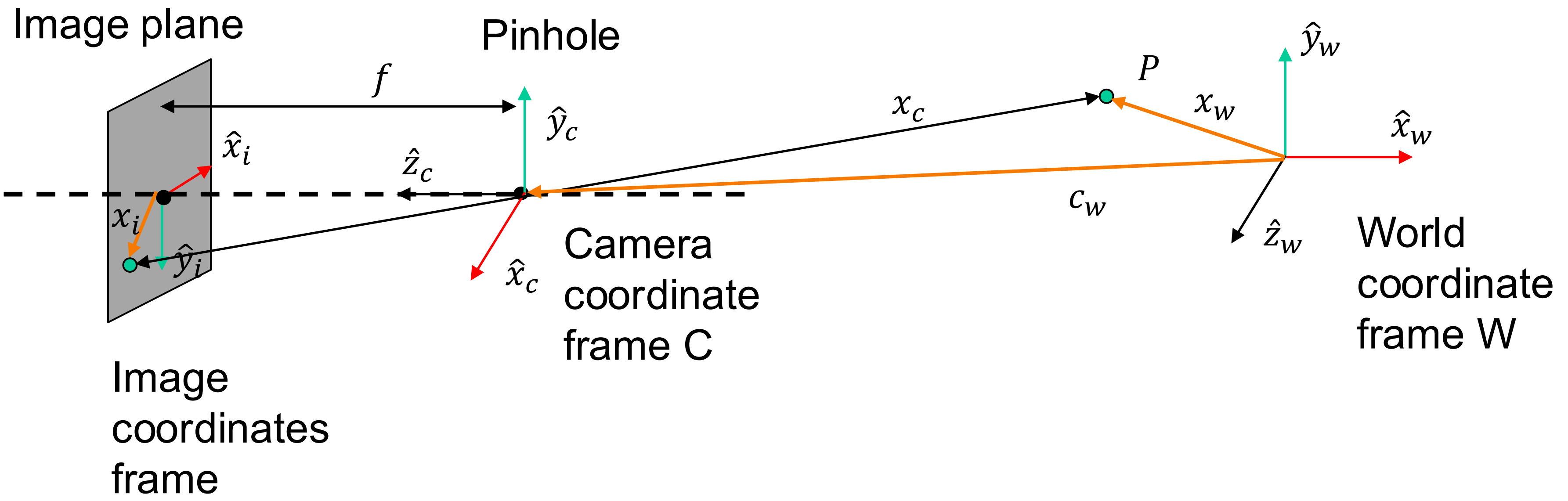
Camera calibration

Forward Imaging Model: 3D to 2D



Slide ideas from First Principles of Computer Vision by Shree Nayar

Forward Imaging Model: 3D to 2D



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

3D-2D

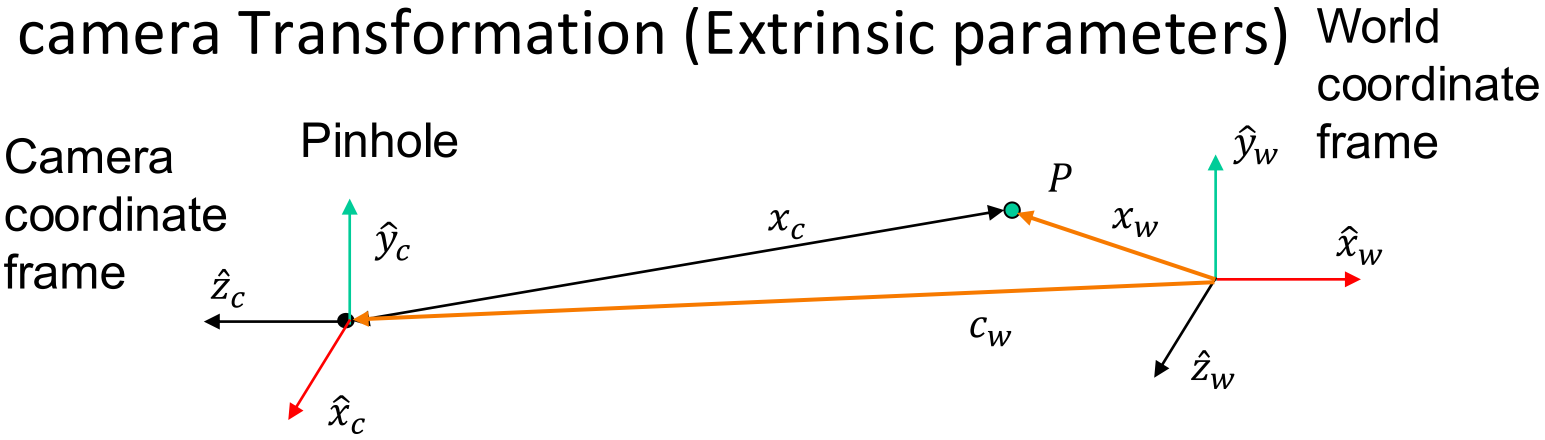
$$\boldsymbol{x}_c = \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix}$$

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

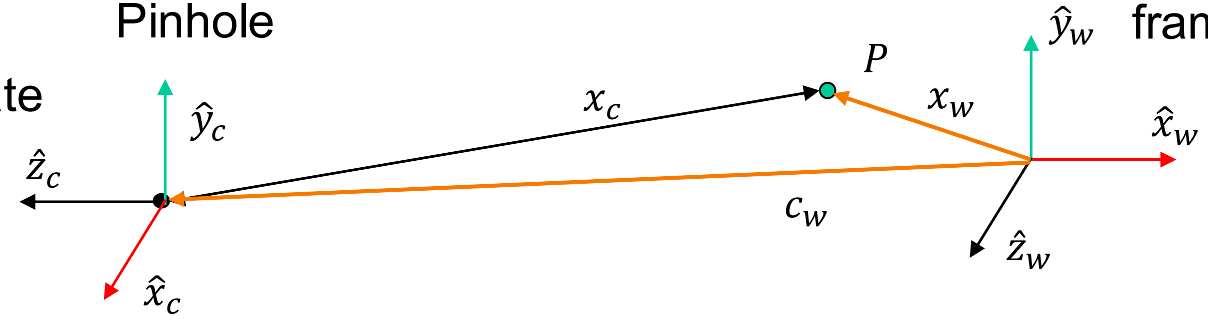
World to camera Transformation (Extrinsic parameters)



Suppose we give you the position c_w of the camera's optical center and the its rotation R in the world coordinate frame (W) and the coordinates of P in the world coordinates P^w then what is the coordinate of P in the camera coordinates?

Slide ideas from First Principles of Computer Vision by Shree Nayar

World to camera Transformation (Extrinsic parameters)

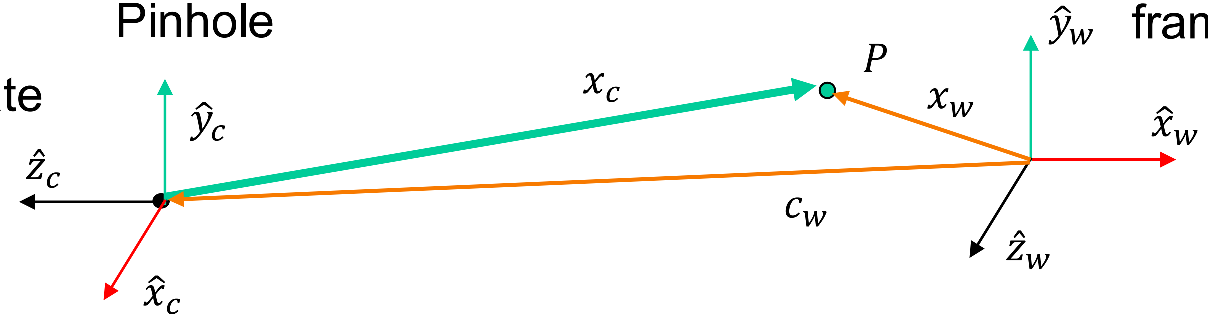


Position c_w and the orientation R of the camera rotation in the world coordinate frame (W) are the camera's **Extrinsic Parameters**

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

This is a **rotation matrix**, i.e., the row vectors or the column vectors are orthonormal
$R^{-1} = R^T$ i.e., $R^T R = R R^T = I$

World to camera Transformation



Position c_w and rotation R of the camera in the world coordinate frame (W)

These are camera's **Extrinsic Parameters**

Recall $p^W = t_{WC} + R_{WC}\, p^C \quad p^C = R_{CW}(p^W - t_{WC})$

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

define $\boldsymbol{t} = -Rc_w$

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

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: $x_c = Rx_w + t$

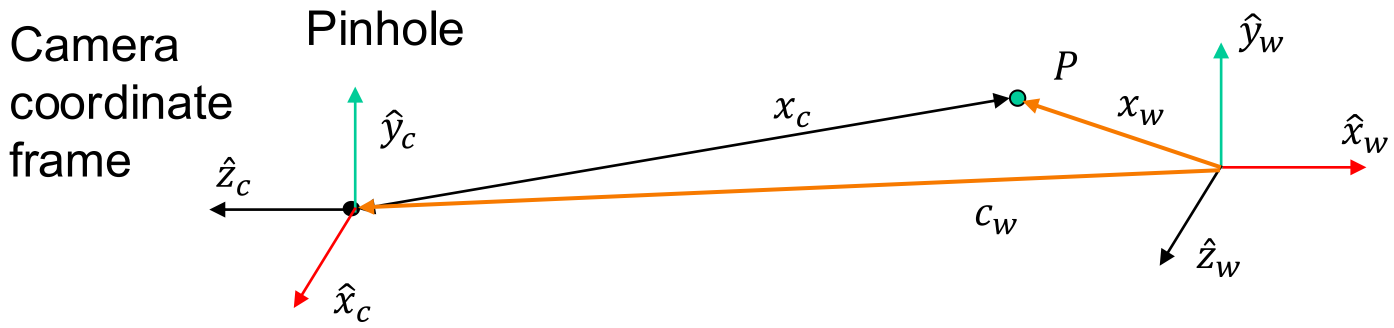
Can we represent it as $x_c = Mx_w$? No

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 $\tilde{x}_c = M\tilde{x}_w$

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

Summary: Extrinsics define transformation from W to C frame



Given camera's extrinsic parameters (R, c_w), the coordinates of P in camera coordinates

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

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

Slide ideas from First Principles of Computer Vision by Shree Nayar

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: $x_c = Rx_w + t$

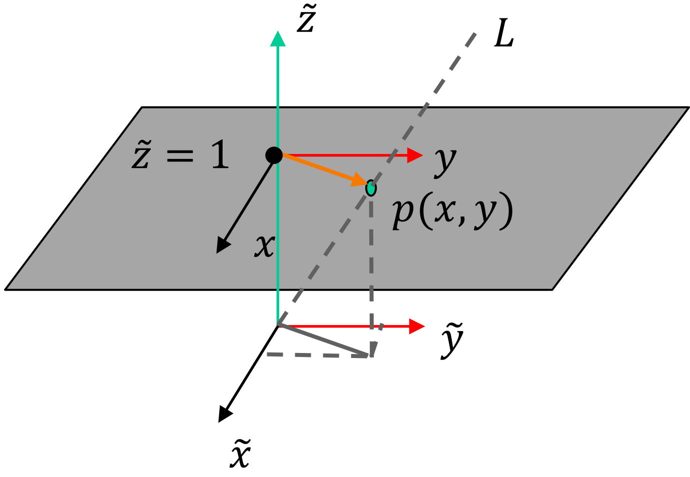
How to represent this as $\tilde{x}_c = M\tilde{x}_w$

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:

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

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

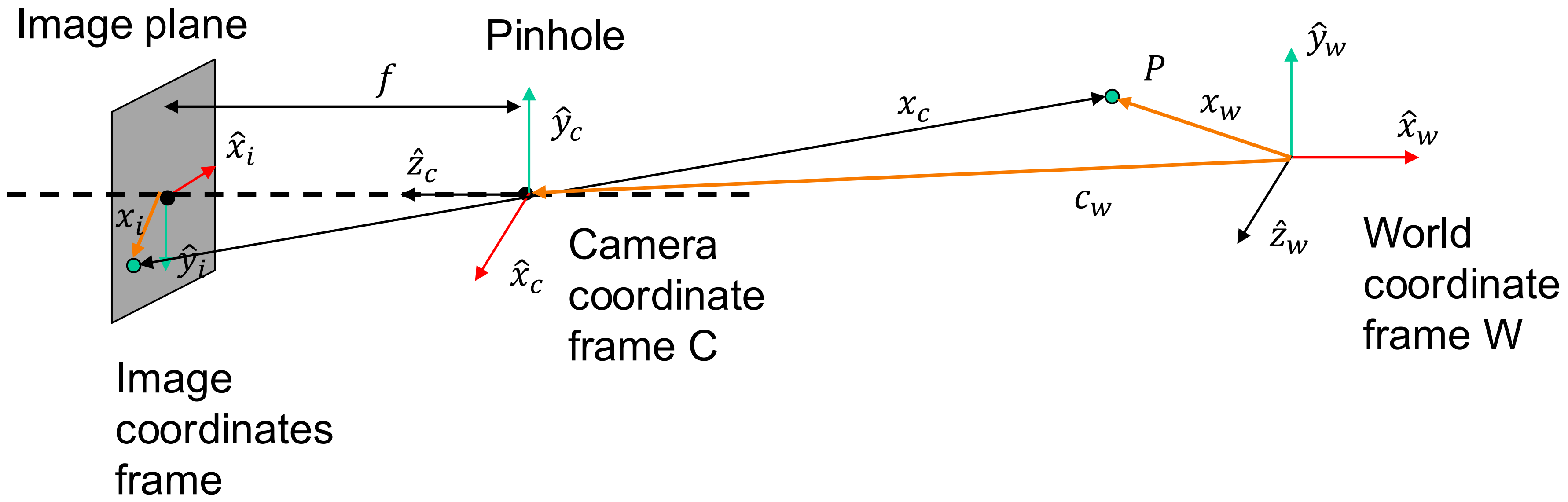


Geometric interpretation: all points on the line L (except origin) represent homogeneous coordinate $p(x, y)$

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

Slide ideas from First Principles of Computer Vision by Shree Nayar

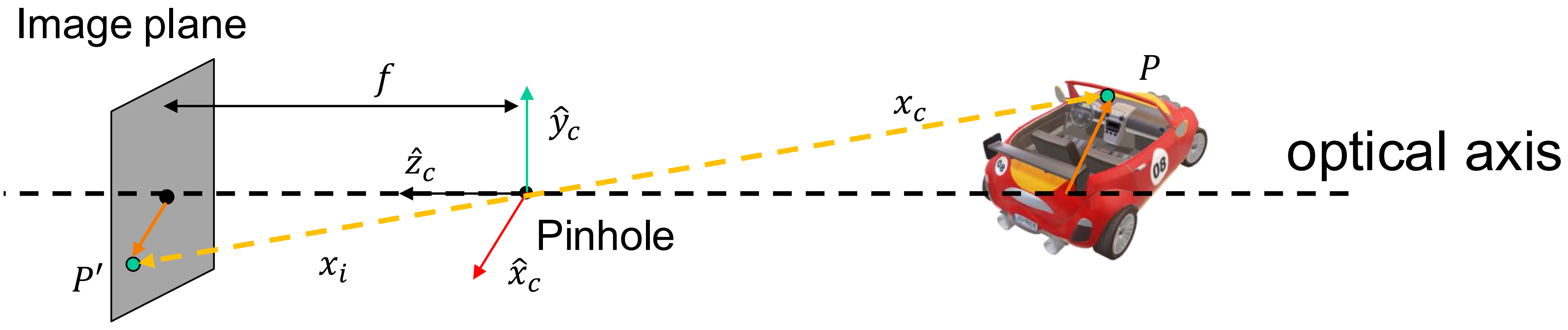
Forward Imaging Model: 3D to 2D



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

Slide ideas from First Principles of Computer Vision by Shree Nayar

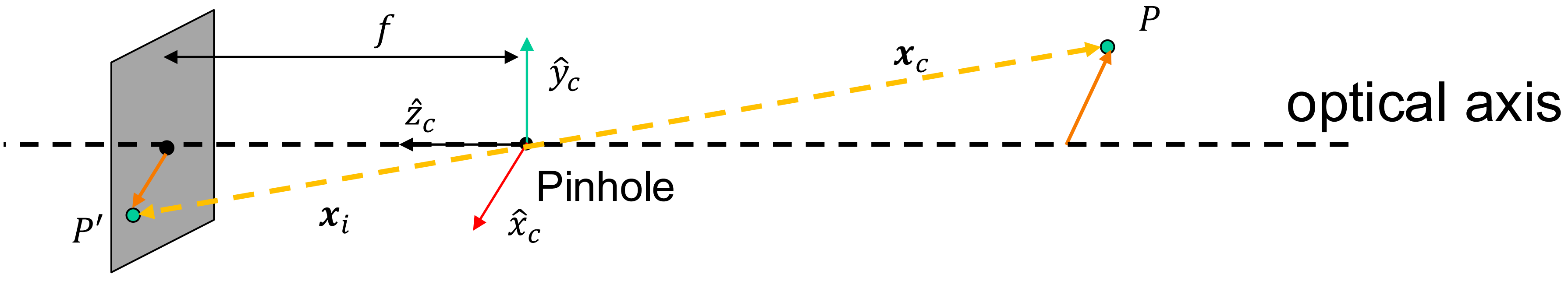
Perspective imaging with pinhole



f: Effective focal length

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

Perspective imaging with pinhole



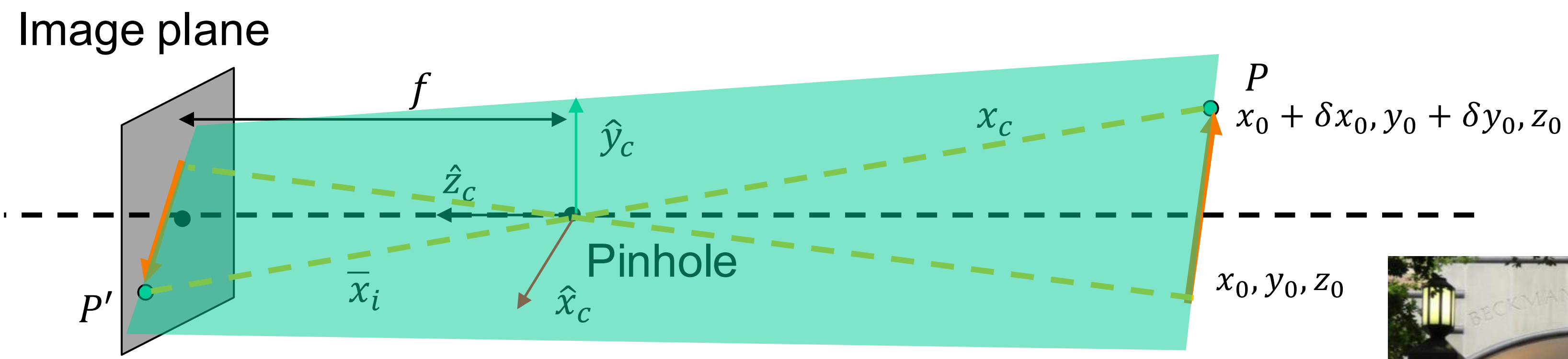
f: Effective focal length

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

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

Not a linear transformation because division by z_c

Perspective projection of a line and magnification



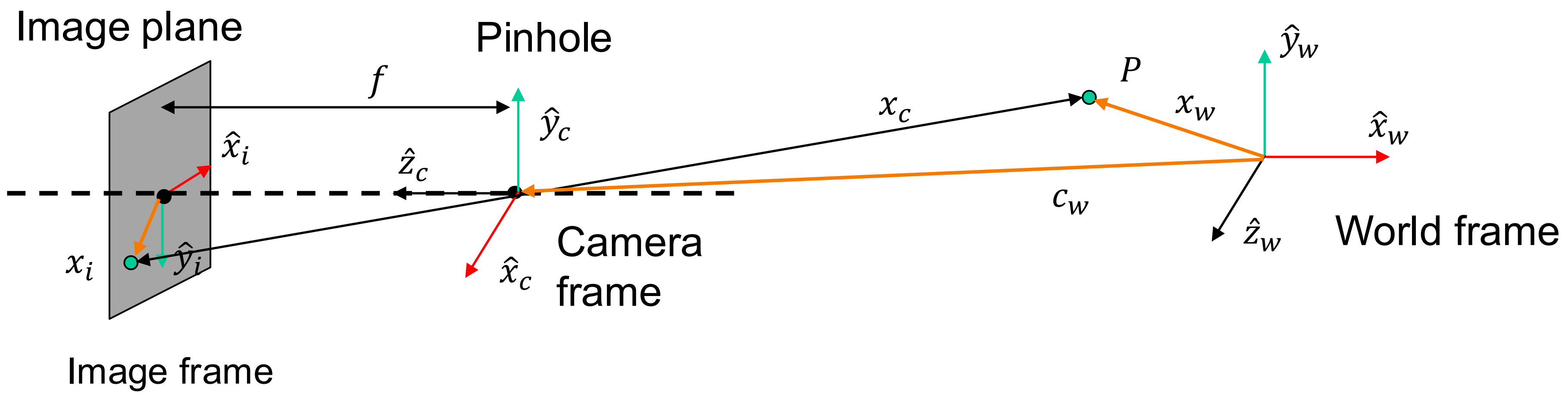
A line in 3D gets mapped to a line in the image plane

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

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

Slide ideas from First Principles of Computer Vision by Shree Nayar

Camera coordinates to image plane coordinates



Perspective projection

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

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

Slide ideas from First Principles of Computer Vision by Shree Nayar

Image plane to image sensor mapping

Image plane

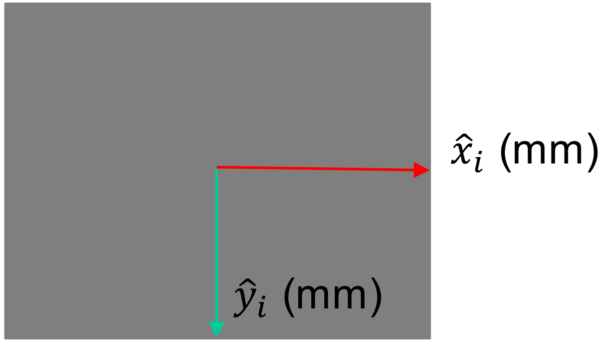
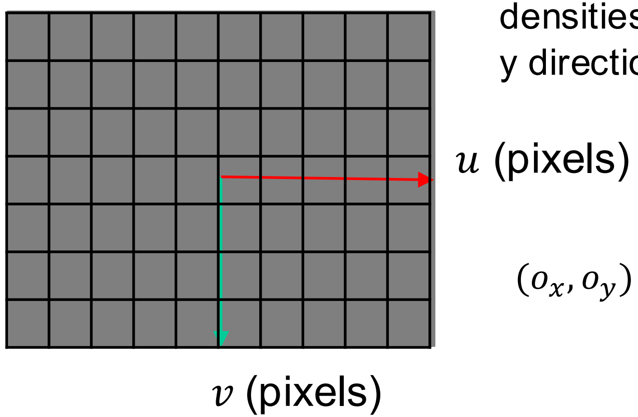


Image sensor



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) Principle point

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

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

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

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

Intrinsic parameters: f_x, f_y, o_x, o_y

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 $v = f_y \frac{y_c}{z_c} + o_y$

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

Adding $z_c = 0. x_c + 0. y_c + 1. z_c$

Stacking them

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

Homogeneous representation of (u, v) as a 3D point $\tilde{u} = (\tilde{u}, \tilde{v}, \tilde{w})$
$(uz_c, vz_c, z_c) \equiv (u, v, 1)$

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

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

$$\tilde{u} = M_{int}\ \tilde{x}_w = [K|0]\tilde{x}_w$$

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

$$\tilde{u} = M_{int} M_{ext}\ \tilde{x}_w = P\ \tilde{x}_w = K[R|t]\tilde{x}_w$$

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

P: **Projection matrix**

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

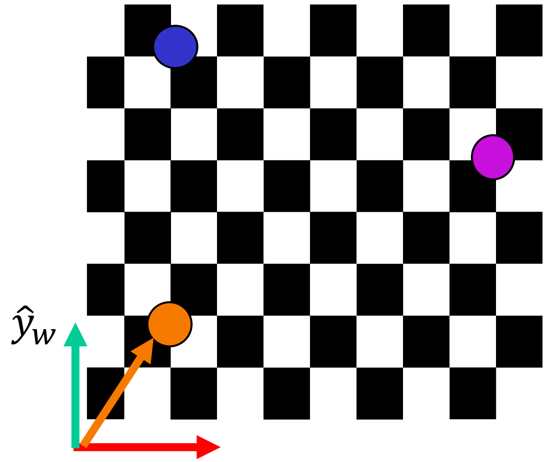
Outline

Forward camera model

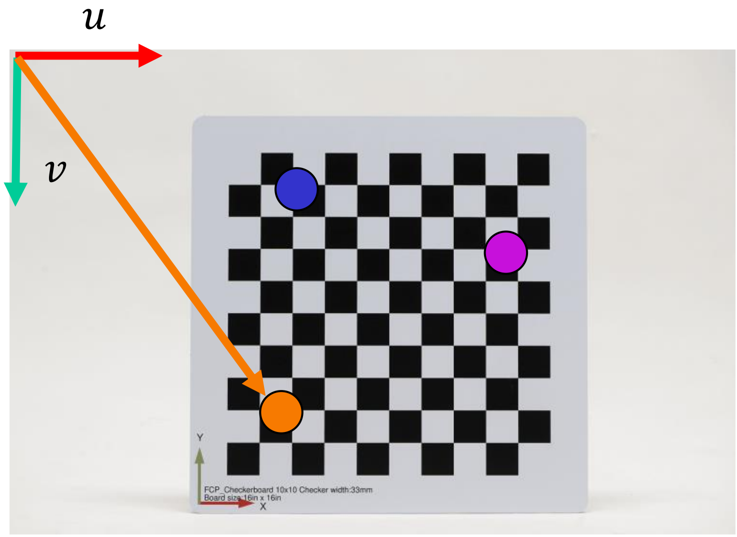
Calibration

Camera Calibration Procedure

Step 1. Capture image of object with known geometry



known geometry object



captured image

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

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

Camera Calibration

Step 2. For each point i with $(x_w^{(i)}, y_w^{(i)}, z_w^{(i)})$ 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 3. 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 and

rearranging $p = [p_{11}\ p_{12}\ \dots p_{34}]^T$ as a vector $\mathbb{R}^{12}$ we get $A\boldsymbol{p} = 0$ where a single row of $A_{u,i} = \left[x_w^{(i)}\ y_w^{(i)}\ z_w^{(i)}\ 1\ 0\ 0\ 0\ -u_i x_w^{(i)}\ -u_i y_w^{(i)}\ -u_i z_w^{(i)}\ \ -u_i\right]$

$A_{v,i} = [0\ \ 0\ \ 0\ \ x_w^{(i)}\ y_w^{(i)}\ z_w^{(i)}\ 1 - v_i x_w^{(i)}\ - v_i y_w^{(i)}\ - v_i z_w^{(i)}\ \ - v_i]$

Step 4. Solve for $\boldsymbol{p}$

$$A\boldsymbol{p} = 0$$

Find $\boldsymbol{p}$ such that $A\boldsymbol{p} = 0$

Find a vector in the null space of A

With noisy measurements the null space will collapse to $\boldsymbol{p} = \boldsymbol{0}$

Also, remember we can only find $\boldsymbol{p}$ up to scale

- We can set one of the elements to be 1 arbitrarily OR
- Add $|p| = 1$ as a constraint

Projection matrix scale

Since projection matrix works on homogeneous coordinates

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

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 $||\boldsymbol{p}||^2 = 1$

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

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

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

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

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

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

Eigenvector corresponding to $\lambda_{\boldsymbol{min}}$ makes $|\lambda_{\boldsymbol{min}}\boldsymbol{p}|$ closest to 0

From projection matrix to Mint Mext

1. Reshape $P \in \mathbb{R}^{3\times4}$ $P = [M \mid \boldsymbol{p_4}]$ with $M \in \mathbb{R}^{3\times3}$
2. Factor M into intrinsic K and rotation R using RQ factorization:
   $M = K\ R$ where K is upper triangular R is orthonormal
3. Solve for translation
   From $P = K\,[R \mid t] = [K\,R \mid K\,t]$ gives:
   $$\boldsymbol{p_4} = Kt \Rightarrow t = K^{-1}\boldsymbol{p_4}$$

From data we have derived the intrinsic $M_{int} = [K|0]$ and the extrinsic $M_{ext} = [R|t]$ matrices and we have a fully calibrated camera

Summary: Monocular vision

- From pixel and 3D coordinate data we can solve the eigenvalue problem to get the camera's projection matrix
- From projection matrix and RQ decomposition we obtain the cameras intrinsic and extrinsic matrices
- The intrinsic and extrinsic matrices define the camera's forward model: transformation from world $W \rightarrow C \rightarrow i$ (pixel coordinates)
- The inverse transform from pixels to world will be useful for perception: pose estimation, visual odometry, structure from motion

Homographies

A homography $H \in \mathbb{R}^{3\times3}$ maps points between two 2D planes in homogeneous coordinates:

$x' \equiv H\,x$ with $x = [u, v, 1]^T$ and $x' = [u', v', 1]^T$

How to use it (dehomogenize)

$x' \equiv H\,x = [\mathrm{a}, \mathrm{b}, \mathrm{c}]^T$ then $u' = \frac{a}{c}$ and $v' = b/c$

Examples: Any two views of the same planar surface (e.g., road, wall)

Image transform for BEV, panorama stitching, etc.

H has 8 DOF (scale is arbitrary), i.e., Need at least 4 point correspondences (each gives 2 equations) to solve for H, then refine by minimizing reprojection error.

Special case: ground plane Z=0 in world frame simplifies to $H = K\,[r_1\,r_2\,t]$

where $r_1\,r_2$ are the first two rows of the rotation matrix