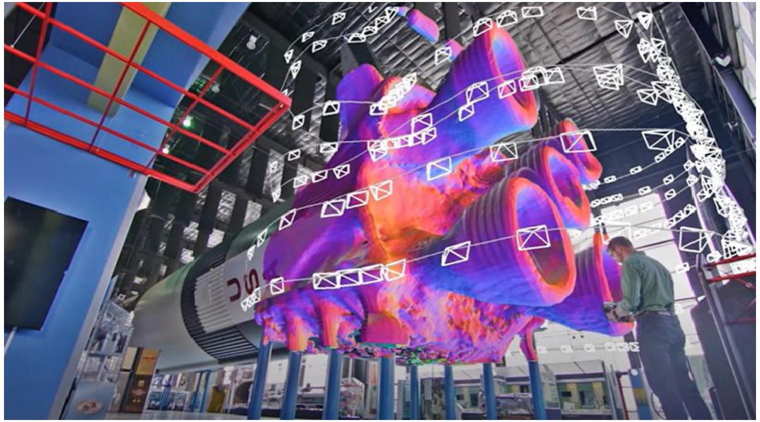
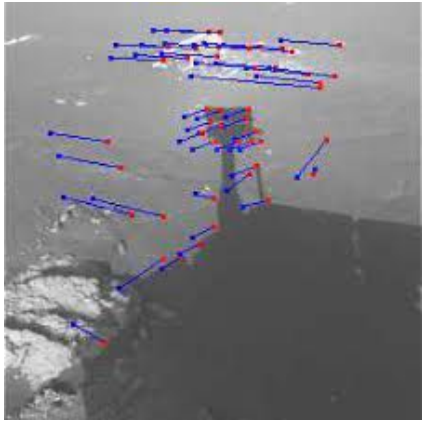
Multi-View Geometry - 1
Camera Modeling and Calibration

RRC Summer School 2024

Rohit Jayanti

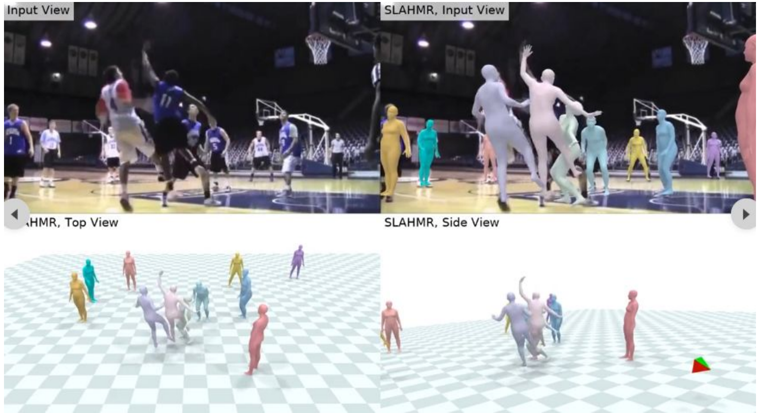


[Skydio's 3D Scan Autonomous Drone Inspection]



[Visual Odometry on Mars!]

[Oreo AR Box Demo]

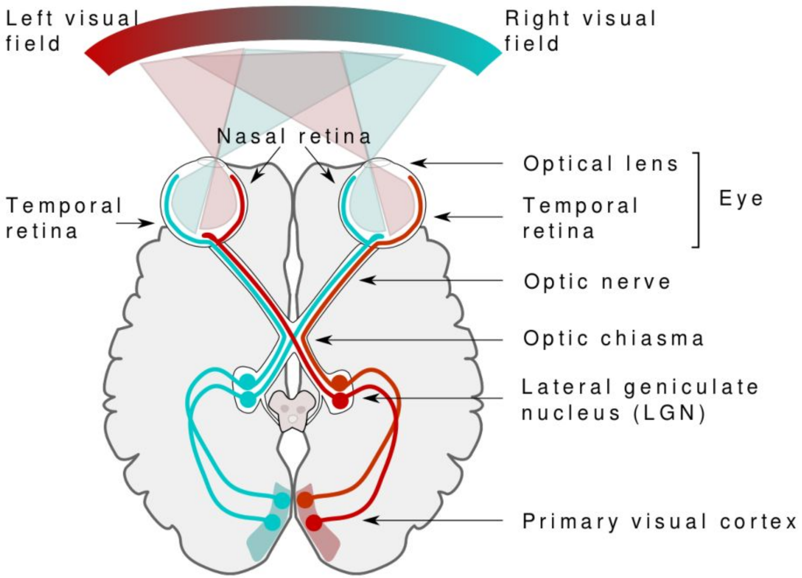


[Decoupled Human and Camera Pose Estimation]

[Language Embedded Radiance Fields]

Why Vision ?

- Most powerful sense - half of primate cerebral cortex is devoted to visual processing
- Enormous data-rate of ~3GB/s
- Extract meaning, both geometric, semantic, from light (nothing travels faster!)
- J. Malik's 3R's - Reorganization (segmentation), Recognition, and Reconstruction

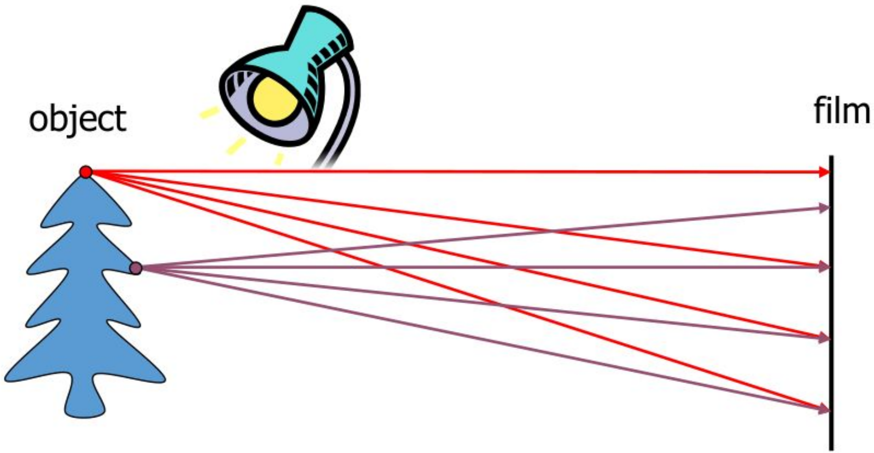


Lecture Objectives

- Camera Modeling
  - Geometric Image Formation and Pinhole Camera
  - Transformations and Projective Geometry
  - General Camera Equation
- Camera Calibration
  - Direct Linear Transform
  - Anatomy of the Projection Matrix (M)

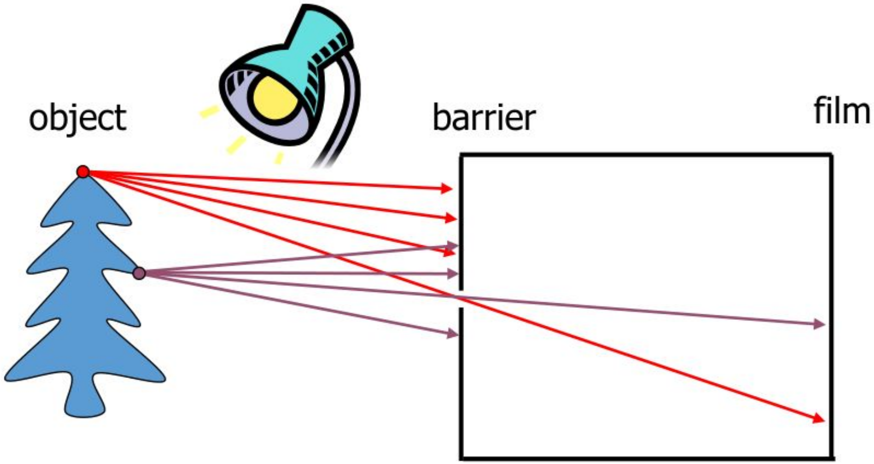
Camera Modeling

Geometric Image Formation



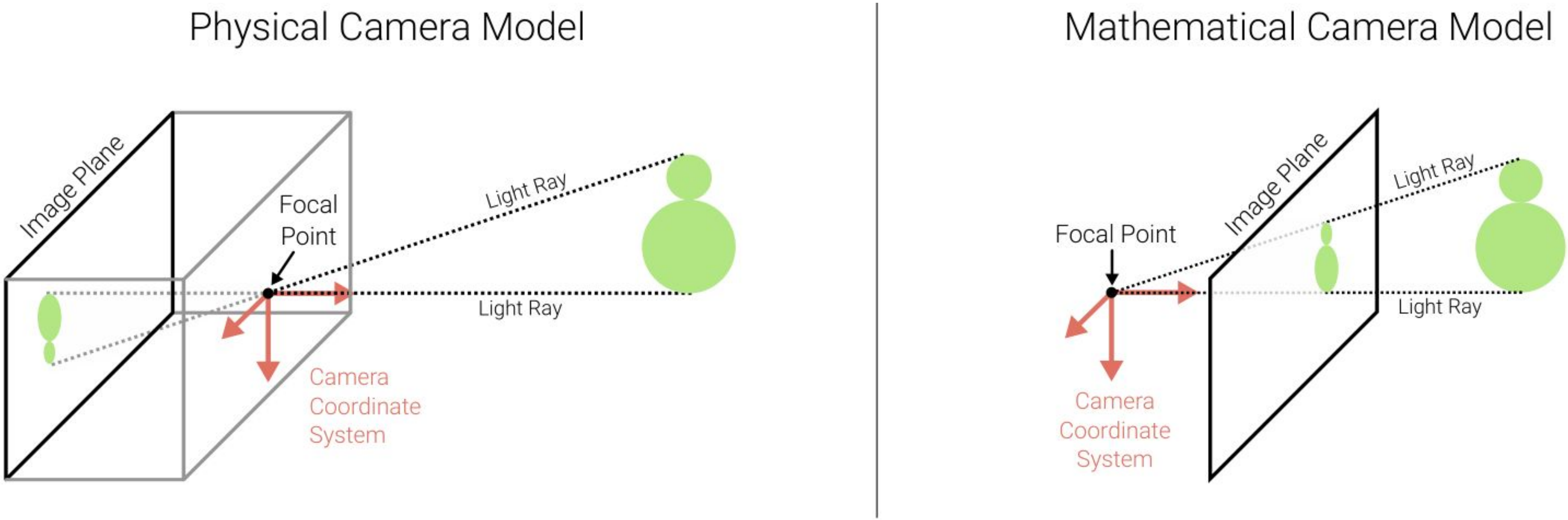
Would this work ?
What would the resulting image look like ?

Geometric Image Formation



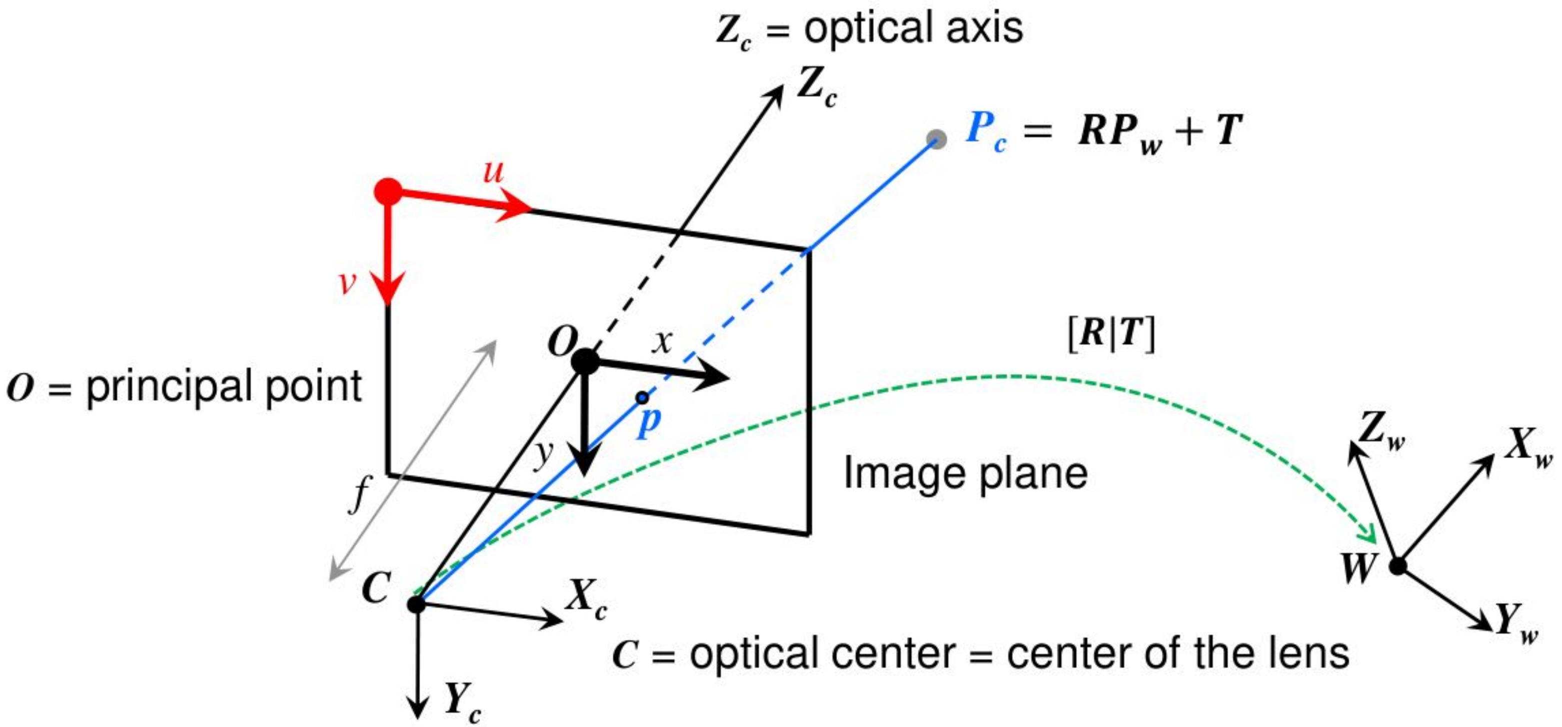
Placing a barrier helps - opening called aperture
What happens if it’s too small ?

Pinhole Camera Model

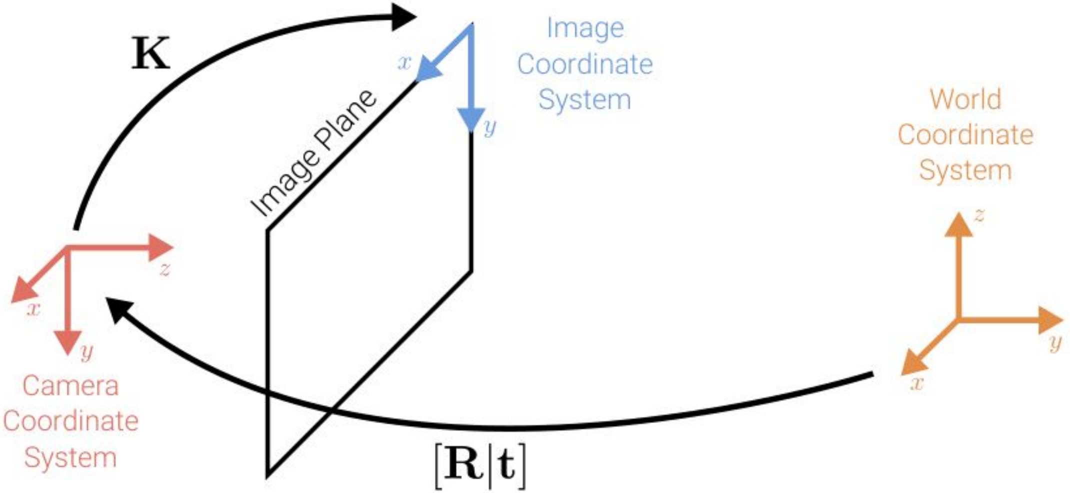


Both models are equivalent, with appropriate change of image coordinates

Pinhole Camera Model



Perspective Projection - Chaining Transformations



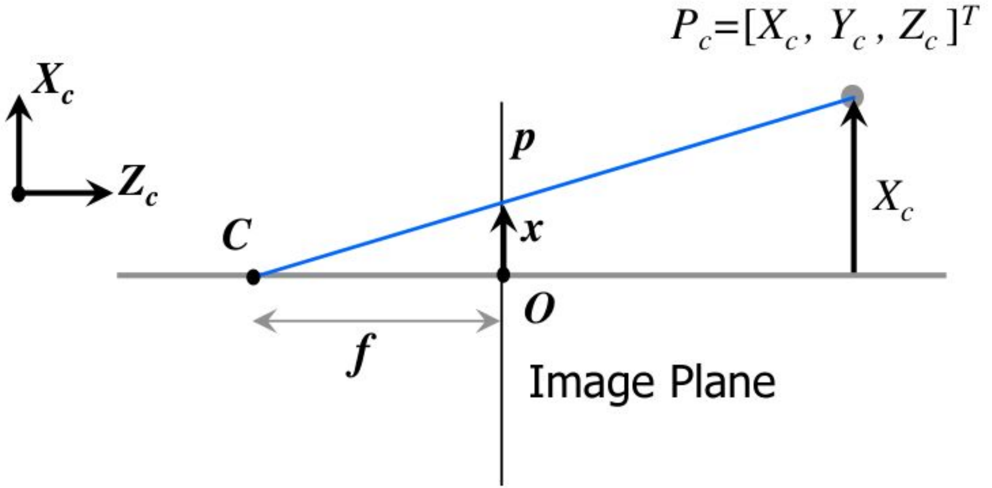
$$\tilde{\mathbf{x}}_s = \begin{bmatrix} \mathbf{K} & \mathbf{0} \end{bmatrix} \bar{\mathbf{x}}_c = \underbrace{\boxed{\begin{bmatrix} \mathbf{K} & \mathbf{0} \end{bmatrix}}}_{\text{Intrinsics}} \underbrace{\boxed{\begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix}}}_{\text{Extrinsics}} \bar{\mathbf{x}}_w = \mathbf{K} \begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix} \bar{\mathbf{x}}_w = \mathbf{P}\, \bar{\mathbf{x}}_w$$

Perspective Projection 1/4 - Camera Frame to Image Plane

$P_c = [X_c, Y_c, Z_c]^T$ projects to $p = (x, y)$ onto the image plane

$$\frac{x}{f} = \frac{X_c}{Z_c} \Rightarrow x = \frac{fX_c}{Z_c}$$

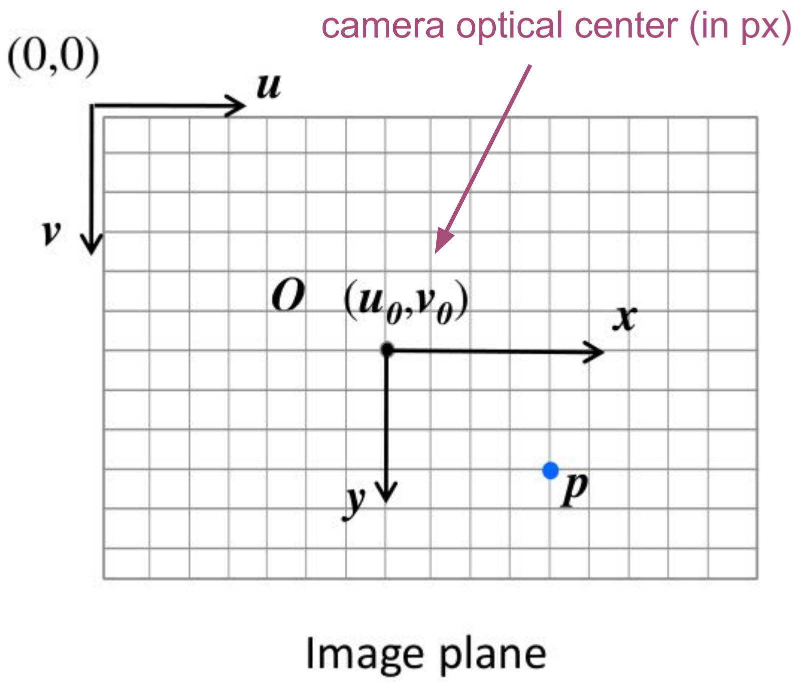
$$\frac{y}{f} = \frac{Y_c}{Z_c} \Rightarrow y = \frac{fY_c}{Z_c}$$



Perspective Projection 2/4 - Image Plane to Pixel Coords.

$$u = u_0 + k_u x \rightarrow u = u_0 + \frac{k_u f X_C}{Z_C}$$

$$v = v_0 + k_v y \rightarrow v = v_0 + \frac{k_v f Y_C}{Z_C}$$



Perspective Projection 3/4 - Mapping from 3D to 2D

- Homogeneous Coordinates to the rescue!

$$p = \begin{pmatrix} u \\ v \end{pmatrix} \Rightarrow \tilde{p} = \begin{bmatrix} \tilde{u} \\ \tilde{v} \\ \tilde{w} \end{bmatrix} = \lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix}}_{\text{K / Calibration matrix / Intrinsic Parameter Matrix}} \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix}$$

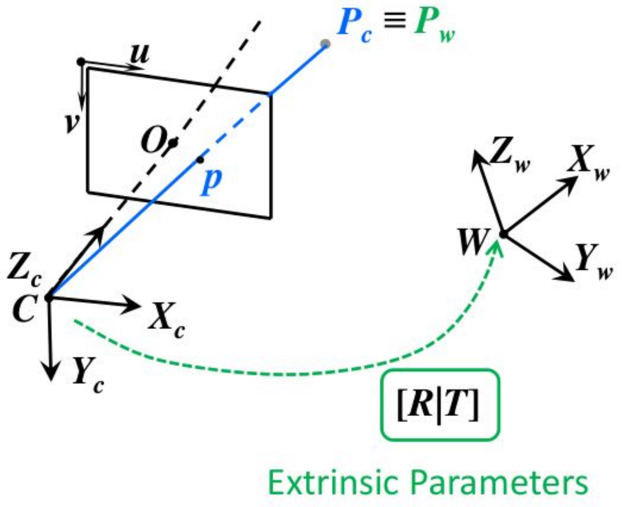
Note: In the past, it was common to assume a skew factor ($K_{12} \neq 0$) to account for sensor manufacturing errors. However, nowadays with updated build quality, it is safe to assume ($K_{12} = s = 0$)

Perspective Projection 4/4 - From World to Camera Frame

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = \left[\begin{array}{ccc|c} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{array}\right] \cdot \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} = \left[\begin{array}{c|c} R & T \end{array}\right] \cdot \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

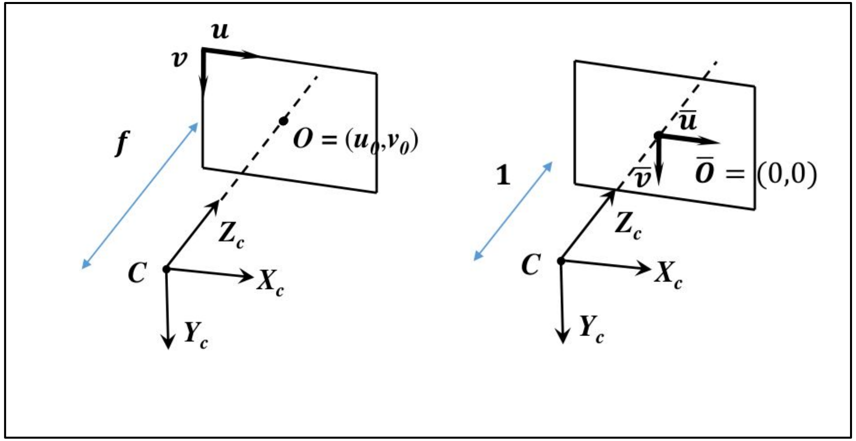
Substituting this into our Perspective Projection Equation

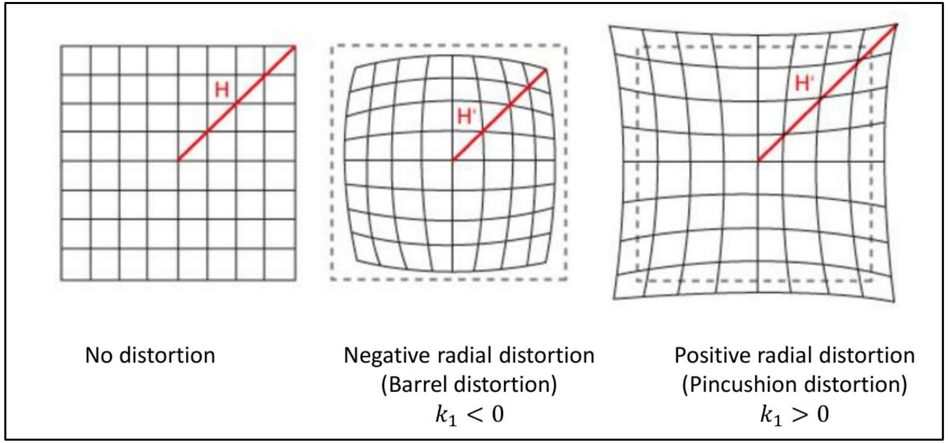
$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} \qquad \lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \underbrace{\boxed{K\left[R \mid T\right]}}_{\text{Projection Matrix (M)}} \cdot \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$



Geometric Image Formation - Self Study

- [Normalized Image Coordinates]
- [Lens Distortion Models]
  - [Simple Quadratic Model of Radial Distortion]





Camera Calibration

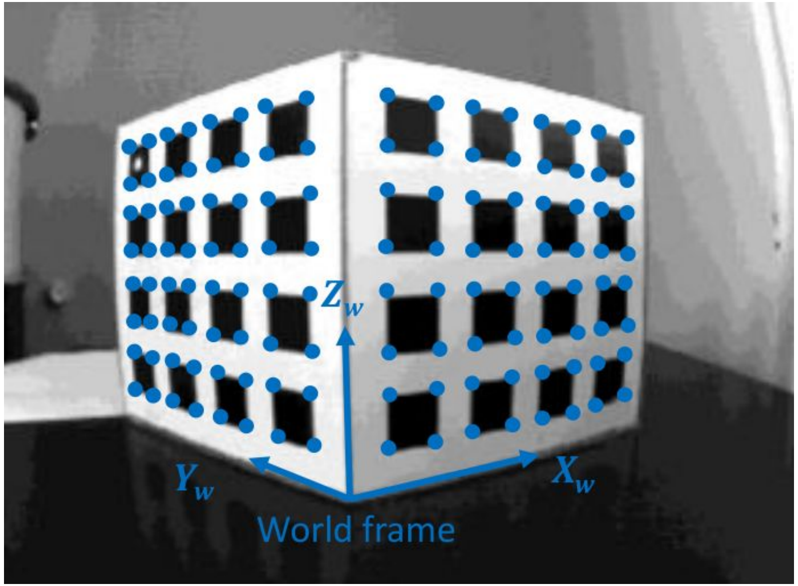
Camera Calibration

- What are we determining ? Intrinsic parameters (*K* plus lens distortion) and extrinsic parameters (*R, T*) - for now we'll neglect lens distortion
- *K, R, T* can be determined by applying the perspective projection equation to known 3D-2D point correspondences
- Mainstream Methods:
  - Tsai's Method: Uses 3D objects
  - Zhang's Method: Uses Planar Grids

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K\left[R \mid T\right] \cdot \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

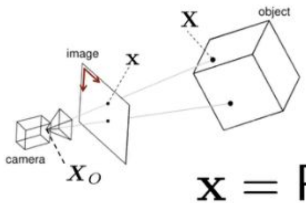
Tsai's Method: Calibration from 3D Objects

- Method proposed in 1987 by Tsai and consists of measuring the 3D position of $\boldsymbol{n \geq 6}$ control points on a 3D calibration target and the 2D coordinates of their projection in the image.



Applying the Direct Linear Transform (DLT) algorithm

DLT simply refers to rewriting the projection equation as a **homogeneous linear equation** and solve it using standard techniques from linear algebra



$$\mathbf{x} = \mathsf{P}\mathbf{X}$$

$$\mathbf{x} = \mathsf{K}R[I_3| - \boldsymbol{X}_O]\mathbf{X}$$

observed image point

c, s, m, x_H, y_H

3 rotations

3 translations

control point

Applying the Direct Linear Transform (DLT) algorithm

- Each 3D point gives two observation equations, one for each image coordinate

$$x = \frac{p_{11}X + p_{12}Y + p_{13}Z + p_{14}}{p_{31}X + p_{32}Y + p_{33}Z + p_{34}}$$

$$y = \frac{p_{21}X + p_{22}Y + p_{23}Z + p_{24}}{p_{31}X + p_{32}Y + p_{33}Z + p_{34}}$$

$$\mathbf{x}_i = \underset{3\times 4}{\mathsf{P}}\, \mathbf{X}_i = \begin{bmatrix} \boxed{p_{11} \quad p_{12} \quad p_{13} \quad p_{14}} \\ \boxed{p_{21} \quad p_{22} \quad p_{23} \quad p_{24}} \\ \boxed{p_{31} \quad p_{32} \quad p_{33} \quad p_{34}} \end{bmatrix} \mathbf{X}_i$$

So what we can rewrite the equation as

$$\begin{bmatrix} u_i \\ v_i \\ w_i \end{bmatrix} = \mathbf{x}_i = \mathsf{P}\mathbf{X}_i = \begin{bmatrix} \boxed{\mathbf{A}^\top} \\ \boxed{\mathbf{B}^\top} \\ \boxed{\mathbf{C}^\top} \end{bmatrix} \mathbf{X}_i = \begin{bmatrix} \mathbf{A}^\top \mathbf{X}_i \\ \mathbf{B}^\top \mathbf{X}_i \\ \mathbf{C}^\top \mathbf{X}_i \end{bmatrix}$$

Applying the Direct Linear Transform (DLT) algorithm

$$x_i = \frac{\mathbf{A}^\top \mathbf{X}_i}{\mathbf{C}^\top \mathbf{X}_i} \quad \Rightarrow \quad x_i\, \mathbf{C}^\top \mathbf{X}_i - \mathbf{A}^\top \mathbf{X}_i = 0$$

$$y_i = \frac{\mathbf{B}^\top \mathbf{X}_i}{\mathbf{C}^\top \mathbf{X}_i} \quad \Rightarrow \quad y_i\, \mathbf{C}^\top \mathbf{X}_i - \mathbf{B}^\top \mathbf{X}_i = 0$$

Leads to a system of equations, which is <u>linear in parameters A, B, and C</u>

$$\begin{aligned} -\mathbf{X}_i^\top \mathbf{A} \quad &+ x_i\, \mathbf{X}_i^\top \mathbf{C} = 0 \\ -\mathbf{X}_i^\top \mathbf{B} \quad &+ y_i\, \mathbf{X}_i^\top \mathbf{C} = 0 \end{aligned}$$

Applying the Direct Linear Transform (DLT) algorithm

Rewriting

$$
\begin{aligned}
-\mathbf{X}_i^\top \mathbf{A} \quad &+ x_i\, \mathbf{X}_i^\top \mathbf{C} = 0 \\
-\mathbf{X}_i^\top \mathbf{B} \quad &+ y_i\, \mathbf{X}_i^\top \mathbf{C} = 0
\end{aligned}
$$

as

$$
\begin{aligned}
\boldsymbol{a}_{x_i}^\top \boldsymbol{p} &= 0 \\
\boldsymbol{a}_{y_i}^\top \boldsymbol{p} &= 0
\end{aligned}
$$

where

$$
\boxed{
\begin{aligned}
\boldsymbol{p} \quad &= \quad (p_k) = \mathrm{vec}(\mathsf{P}^\top) \\
\boldsymbol{a}_{x_i}^\top \quad &= \quad (-\mathbf{X}_i^\top, \mathbf{0}^\top, x_i \mathbf{X}_i^\top) \\
&= \quad (-X_i, -Y_i, -Z_i, -1, 0, 0, 0, 0, x_i X_i, x_i Y_i, x_i Z_i, x_i) \\
\boldsymbol{a}_{y_i}^\top \quad &= \quad (\mathbf{0}^\top, -\mathbf{X}_i^\top, y_i \mathbf{X}_i^\top) \\
&= \quad (0, 0, 0, 0, -X_i, -Y_i, -Z_i, -1, y_i X_i, y_i Y_i, y_i Z_i, y_i)
\end{aligned}
}
$$

Applying the Direct Linear Transform (DLT) algorithm

For each point, we have

$$\boldsymbol{a}_{x_i}^\mathsf{T} \boldsymbol{p} = 0$$
$$\boldsymbol{a}_{y_i}^\mathsf{T} \boldsymbol{p} = 0$$

stacking everything together

$$\begin{bmatrix} \boldsymbol{a}_{x_1}^\mathsf{T} \\ \boldsymbol{a}_{y_1}^\mathsf{T} \\ \cdots \\ \boldsymbol{a}_{x_i}^\mathsf{T} \\ \boldsymbol{a}_{y_i}^\mathsf{T} \\ \cdots \\ \boldsymbol{a}_{x_I}^\mathsf{T} \\ \boldsymbol{a}_{y_I}^\mathsf{T} \end{bmatrix} \boldsymbol{p} = \underset{2I \times 12}{\mathsf{M}} \; \underset{12 \times 1}{\boldsymbol{p}} \stackrel{!}{=} 0$$

Applying the Direct Linear Transform (DLT) algorithm

- How do we solve this **Ax=0 ?** (Hint: We're finding the null-space of A)

Applying the Direct Linear Transform (DLT) algorithm

- How do we solve this **Ax=0 ?** (Hint: We're finding the null-space of A)
- We can apply Singular Value Decomposition (SVD) on M
- p is the singular vector belonging to singular value of 0
- Which corresponds to ?

$$\underset{2I\times 12}{\mathsf{M}} = \underset{2I\times 12}{U} \; \underset{12\times 12}{\mathsf{S}} \; \underset{12\times 12}{V^{\mathsf{T}}} = \sum_{i=1}^{12} s_i \boldsymbol{u}_i \boldsymbol{v}_i^{\mathsf{T}}$$

Applying the Direct Linear Transform (DLT) algorithm

- How do we solve this **Ax=0 ?** (Hint: We're finding the null-space of A)
- We can apply Singular Value Decomposition (SVD) on M
- p is the singular vector belonging to singular value of 0
- Which corresponds to ?
- **Pick the last row of V^T**

$$\underset{2I\times 12}{\mathsf{M}} = \underset{2I\times 12}{U} \; \underset{12\times 12}{\mathsf{S}} \; \underset{12\times 12}{V^{\mathsf{T}}} = \sum_{i=1}^{12} s_i \boldsymbol{u}_i \boldsymbol{v}_i^{\mathsf{T}}$$

Applying the Direct Linear Transform (DLT) algorithm

- Reshaping 12x1 to 3x4

$$\boldsymbol{p} = \begin{bmatrix} p_{11} \\ \vdots \\ p_{34} \end{bmatrix} \Rightarrow \mathsf{P} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix}$$

- Now, we need to arrive at K, X_0, and R

Decomposition of P

- Structure of P_{3x4}

$$P = \left[KR \mid -KRX_0\right] = \left[H \mid h\right]$$

- Projection Center

$$X_0 = -H^{-1}h$$

- Some things we know about H = KR
  - K is an upper triangular matrix
  - R is a rotation matrix
- Is there a matrix decomposition into a rotation matrix and a triangular one?

Decomposition of P

QR Decomposition of H^{-1} yields *R & K*

$$H^{-1} = (K\,R)^{-1} = R^{-1}\,K^{-1} = \underset{Q}{\underset{\uparrow}{R^{T}}}\;\underset{R}{\underset{\uparrow}{K^{-1}}}$$

- H = KR is homogenous
- Therefore recovered K matrix is in homogenous form too
- Normalize as $K{:}K/K_{33}$

Anatomy of the Projection Matrix

- Any 3x4 P with a nonsingular left submatrix represents a valid camera! It can be decomposed as shown in the previous slides as
  - A non-singular upper diagonal matrix K
  - An orthonormal matrix R and a vector C with their usual meanings!
- The 4-vector **C** with **PC=0** is the camera center
- Can you find the world origin just by looking at the projection matrix ?

Anatomy of the Projection Matrix

- Any 3x4 P with a nonsingular left submatrix represents a valid camera! It can be decomposed as shown in the previous slides as
  - A non-singular upper diagonal matrix K
  - An orthonormal matrix R and a vector C with their usual meanings!
- The 4-vector **C** with **PC=0** is the camera center
- Can you find the world origin just by looking at the projection matrix ?
  - [0, 0, 0] in homogeneous coordinates is the world origin
  - Thus $\mathbf{p}_4$ (last column of **P**) is the image of the world origin
  - $\mathbf{p}_4 = \mathbf{P}[0\ 0\ 0\ 1]^T$

More on Calibration in upcoming sessions...

- Zhang’s Method - Complete Intrinsics and Extrinsics using Planar Grids
- Perspective N-Point (PnP) - Only Extrinsics using 2D-3D correspondences

Resources

- Cyrill Stachniss' Mobile Sensing and Robotics 2 (2021) [Youtube]
  - Lectures 23-27: Camera Parameters and DLT
- Shree K Nayar's First Principles of Computer Vision [Youtube]
  - Image Formation Sub-Playlist [Youtube]
  - Camera Calibration Series [Youtube]
  - Fun fact: The first student who'd go on to do their PhD from IIIT-H went under Prof. Shree K Nayar
- Andreas Geiger's Computer Vision @ Tübingen [Youtube][Course Page]
  - Lecture 02: Image Formation [Slides]

Acknowledgments:
Slides and Images adapted from Andreas Geiger, Wikiwand, Davide Scaramuzza, RRC SS-23