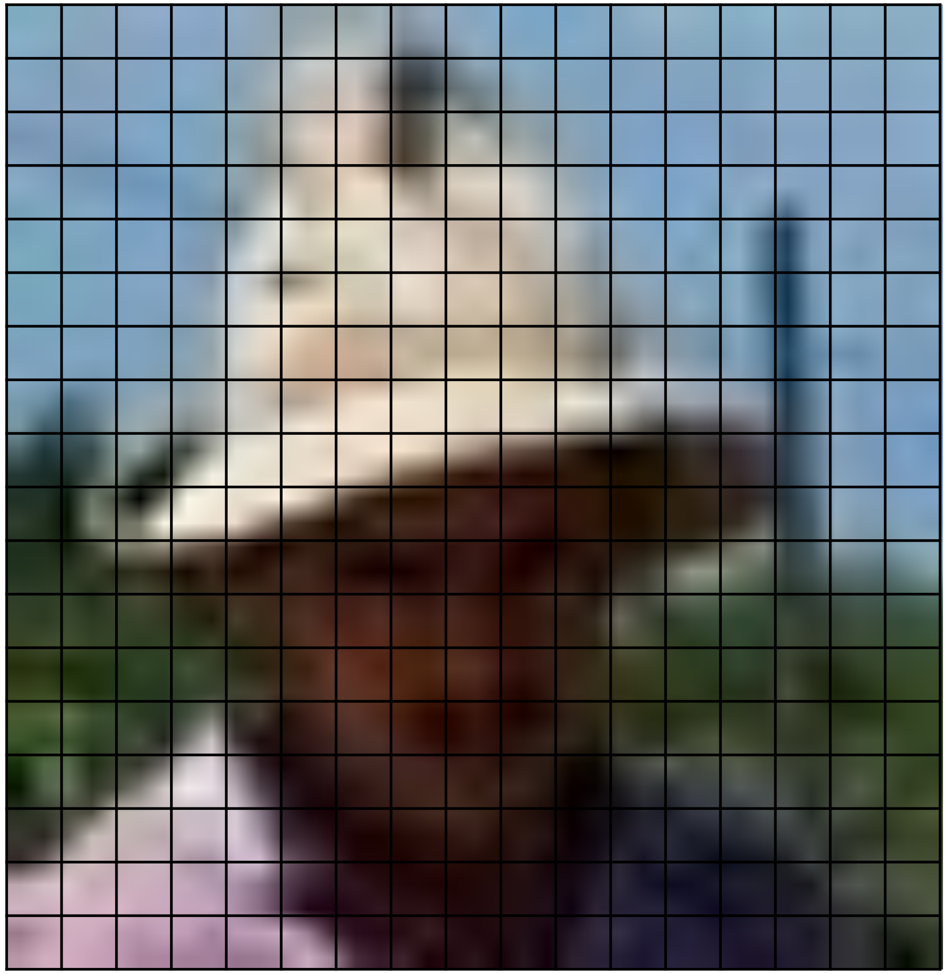
Lecture
Camera Calibration

Liangliang Nan

Today's Agenda

- Review: Camera models
- Camera calibration
- A1: Camera calibration

Images

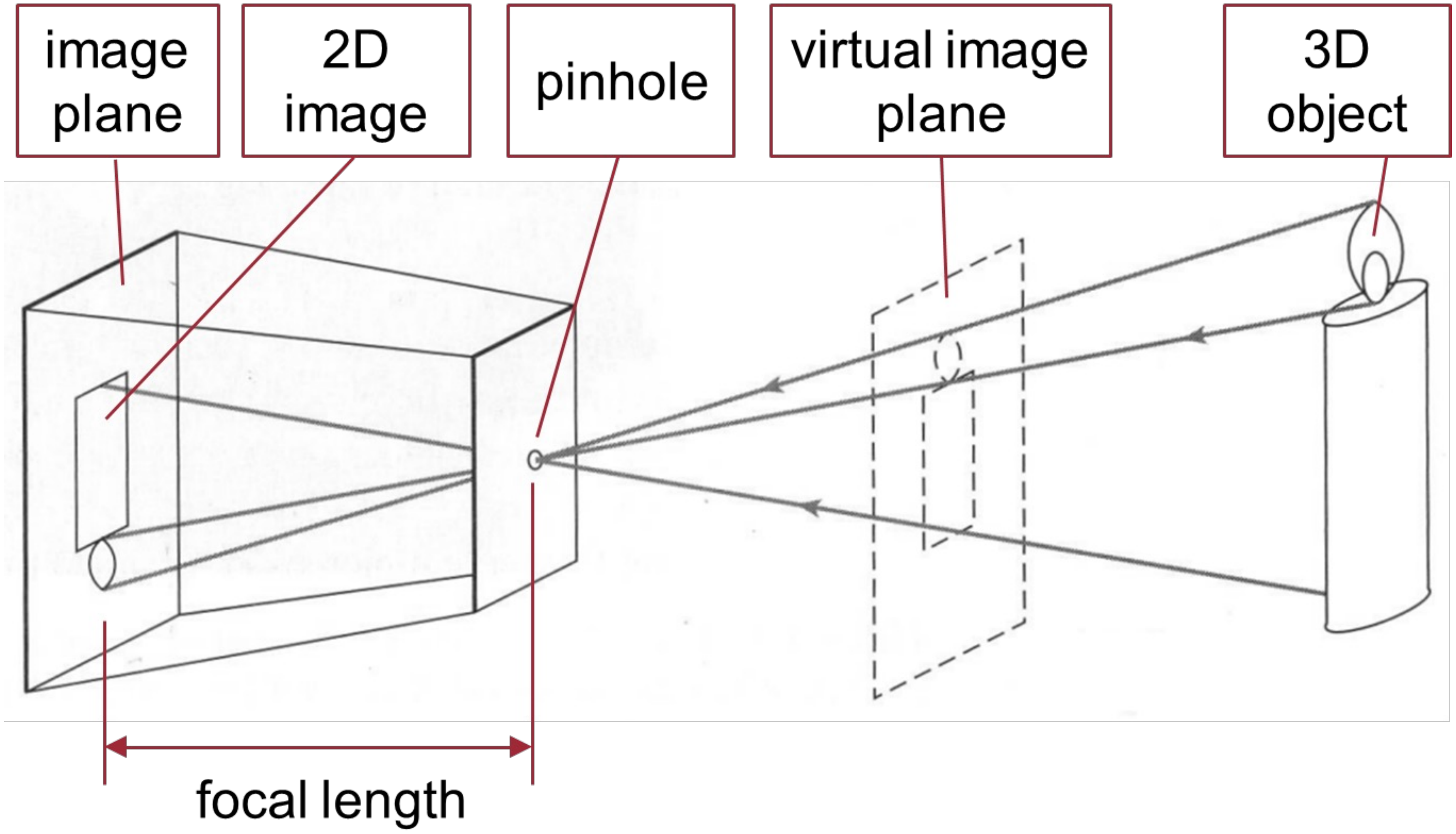


A color image: R, G, B channels

$$f(x,y) = \begin{bmatrix} r(x,y) \\ g(x,y) \\ b(x,y) \end{bmatrix}$$

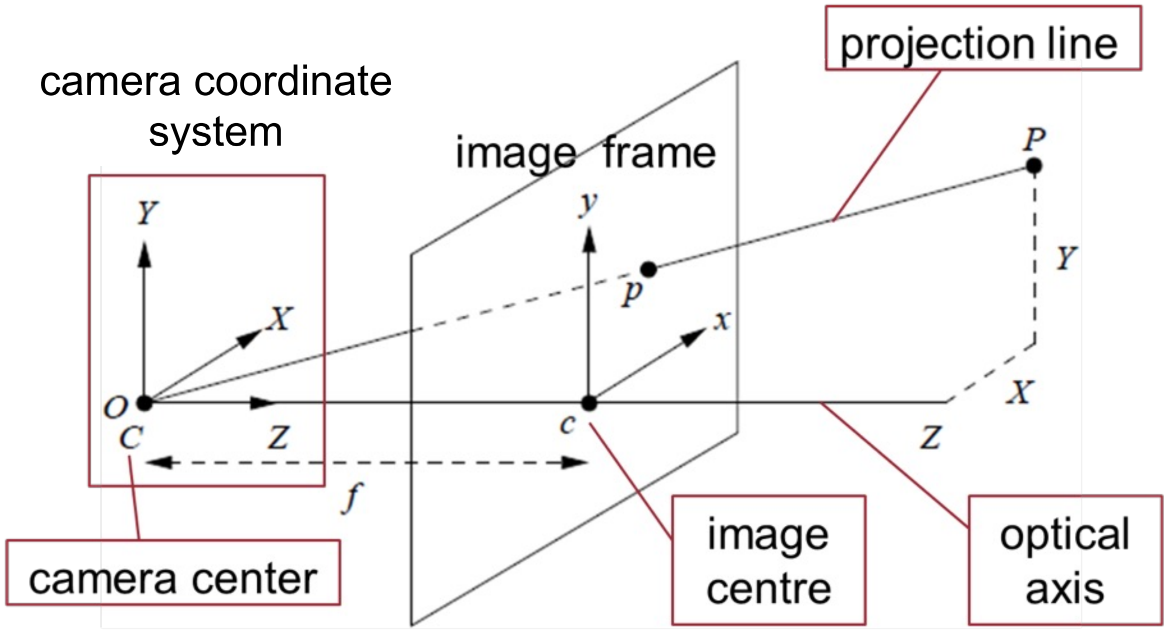
“vector-valued” function

Pinhole camera model



Pinhole camera model

- 3D point $\mathbf{P} = (X, Y, Z)^T$ projected to 2D image $\mathbf{p} = (x, y)^T$



$$x = f\frac{X}{Z}, \qquad y = f\frac{Y}{Z}$$

Perspective projection model

- Pinhole camera $x = f\frac{X}{Z}, \quad y = f\frac{Y}{Z}$
- Change of unit: physical measurements -> pixels
  - If $k = l$, camera sensor's pixels are exactly square

$$x = kf\frac{X}{Z}, \quad y = lf\frac{Y}{Z}$$

Denote $\alpha = kf, \beta = lf$

$$x = \alpha\frac{X}{Z}, \quad y = \beta\frac{Y}{Z}$$

- x, y: image coordinates (*pixels*)
- k, l: scale parameters (*pixels/mm*)
- f : focal length (*mm*)

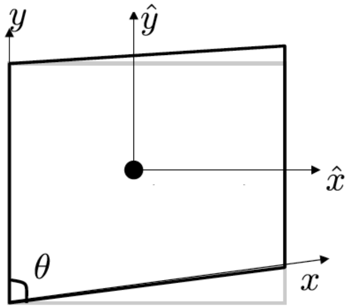
Perspective projection model

- Pinhole camera $x = f\frac{X}{Z}, \quad y = f\frac{Y}{Z}$
- Change of unit: physical measurements -> pixels $x = \alpha\frac{X}{Z}, \quad y = \beta\frac{Y}{Z}$
- Change of coordinate system
  - Image plane coordinates have origin at image center
  - Digital image coordinates have origin at top-left corner

$$x = \alpha\frac{X}{Z} + c_x, \quad y = \beta\frac{Y}{Z} + c_y$$

Perspective projection model

- Pinhole camera $x = f\frac{X}{Z}, \quad y = f\frac{Y}{Z}$
- Change of unit: physical measurements -> pixels $x = \alpha\frac{X}{Z}, \quad y = \beta\frac{Y}{Z}$
- Change of coordinate system $x = \alpha\frac{X}{Z} + c_x, \quad y = \beta\frac{Y}{Z} + c_y$
- Account for skewness
  - Image frame may not be exactly rectangular due to sensor manufacturing errors



$$x = \alpha\frac{X}{Z} - \alpha \cot\theta \frac{Y}{Z} + c_x, \quad y = \frac{\beta}{\sin\theta}\frac{Y}{Z} + c_y$$

θ: skew angle between x- and y-axis

Perspective projection model

$$x = \alpha \frac{X}{Z} - \alpha \cot\theta \frac{Y}{Z} + c_x, \qquad y = \frac{\beta}{\sin\theta} \frac{Y}{Z} + c_y$$

- Rewrite in matrix-vector product form

$$\mathbf{P} = [X, Y, Z]^{\mathrm{T}}, \quad \mathbf{p} = [x, y, 1]^{\mathrm{T}}$$

(homogeneous coordinates)

$$\mathbf{p} = K\mathbf{P}, \quad K = \begin{bmatrix} \alpha & -\alpha \cot\theta & c_x \\ 0 & \frac{\beta}{\sin\theta} & c_y \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

Intrinsic parameter matrix

Perspective projection model

- Camera motion
  - World frame may not align with the camera frame
  - Camera can move and rotate

$$\mathbf{P}^C = R_W^C \, \mathbf{P}^W + \mathbf{t}_W^C$$

1 3 2 4

Camera frame

World frame

1. Coordinates of 3D scene point in camera frame.
2. Coordinates of 3D scene point in world frame.
3. Rotation matrix of world frame in camera frame.
4. Position of world frame's origin in camera frame.

Perspective projection model

- The complete transformation

$$\mathbf{p} = M\mathbf{P}$$
$$= K \begin{bmatrix} R & \mathbf{t} \end{bmatrix} \mathbf{P}$$

Internal (intrinsic) parameters

External (extrinsic) parameters

- R: rotation matrix of the world coordinate system defined in the camera coordinate system
- **t**: the position of world coordinate system's origin in camera coordinate system
  (Note: **t** is often mistakenly interpreted as the position of the camera position in the world coordinate system)

Today’s Agenda

- Review: Camera models
- Camera calibration
- A1: Camera calibration

General Idea

- Why is camera calibration necessary?
  - Given 3D scene, knowing the precise 3D to 2D projection requires
    - Intrinsic and extrinsic parameters
  - Reconstructing 3D geometry from images also requires these parameters

$$\mathbf{p} = M\mathbf{P}$$
$$= K \begin{bmatrix} R & \mathbf{t} \end{bmatrix} \mathbf{P}$$

Internal (intrinsic) parameters: K

External (extrinsic) parameters: $\begin{bmatrix} R & \mathbf{t} \end{bmatrix}$

General Idea

- Why is camera calibration necessary?
- What information do we have?
  – Images only

General Idea

- Why is camera calibration necessary?
- What information do we have?
- Camera calibration
  - Recovering *K*
  - Recovering *R* and **t**
- How many parameters

$$\mathbf{p} = M\mathbf{P}$$
$$= K \begin{bmatrix} R & \mathbf{t} \end{bmatrix} \mathbf{P}$$

Internal (intrinsic) parameters

External (extrinsic) parameters

General Idea

- How many parameters to recover?
  - How many intrinsic parameters?

$$K = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

$$\begin{aligned} \mathbf{p} &= M\mathbf{P} \\ &= \boxed{K}\,[R \;\; \mathbf{t}]\,\mathbf{P} \end{aligned}$$

Internal (intrinsic) parameters

General Idea

- How many parameters to recover?
  - How many intrinsic parameters?
  - How many extrinsic parameters?

$$R = \begin{bmatrix} \mathbf{r}_1^T \\ \mathbf{r}_2^T \\ \mathbf{r}_3^T \end{bmatrix}, \quad \mathbf{t} = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}$$

$$\begin{aligned} \mathbf{p} &= M\mathbf{P} \\ &= K \begin{bmatrix} R & \mathbf{t} \end{bmatrix} \mathbf{P} \end{aligned}$$

External (extrinsic) parameters

General Idea

- How many parameters to recover: 11
  - 5 intrinsic parameters
    - 2 for focal lengths
    - 2 for offset (image center, or principal point)
    - 1 for skewness
  - 6 extrinsic parameters
    - 3 for rotation
    - 3 for translation

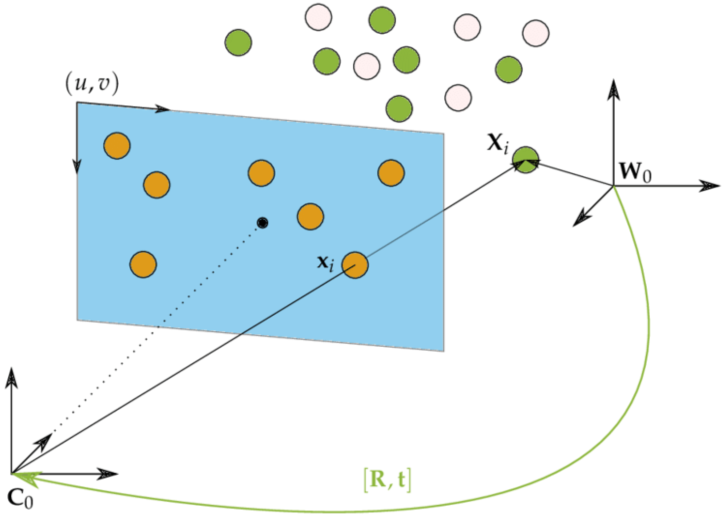
$$K = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}, \quad R = \begin{bmatrix} \mathbf{r}_1^T \\ \mathbf{r}_2^T \\ \mathbf{r}_3^T \end{bmatrix}, \quad \mathbf{t} = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}$$

General Idea

- What information to use?

  - Corresponding 3D-2D point pairs

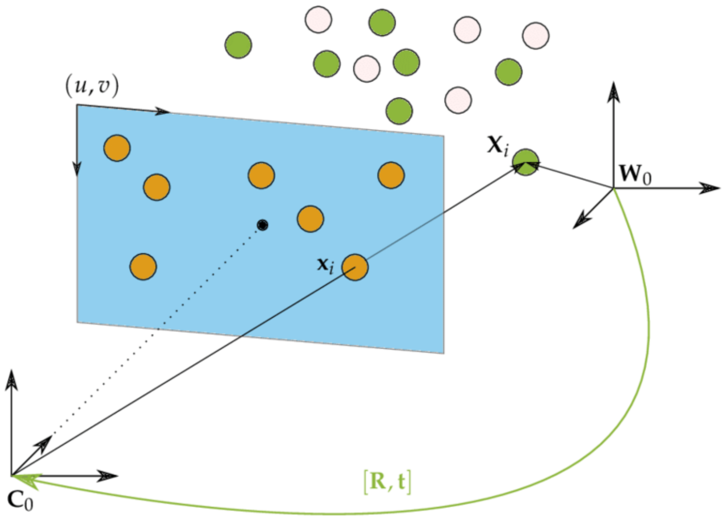
$$\mathbf{p} = M\mathbf{P}$$
$$= K\begin{bmatrix}R & \mathbf{t}\end{bmatrix}\mathbf{P}$$



General Idea

- What information to use?
  - Corresponding 3D-2D point pairs
    - How many pairs do we need?

$$\mathbf{p} = M\mathbf{P}$$
$$= K\begin{bmatrix} R & \mathbf{t} \end{bmatrix}\mathbf{P}$$



General Idea

- What information to use?
  - Corresponding 3D-2D point pairs
    - How many pairs do we need?
      - How much information does each pair of corresponding point provide?

$$\mathbf{p} = M\mathbf{P} \Rightarrow \mathbf{p}_i = \begin{bmatrix} u_i \\ v_i \end{bmatrix} = M\mathbf{P}_i = \begin{bmatrix} \frac{\mathbf{P}_i^T \mathbf{m}_1}{\mathbf{P}_i^T \mathbf{m}_3} \\ \\ \frac{\mathbf{P}_i^T \mathbf{m}_2}{\mathbf{P}_i^T \mathbf{m}_3} \end{bmatrix}$$

$\mathbf{m}_1$, $\mathbf{m}_2$, $\mathbf{m}_3$: the three rows of the projection matrix M

General Idea

- What information to use?
  - Corresponding 3D-2D point pairs
    - How many pairs do we need?
      - Each 3D-2D point pair -> 2 equations
      - 11 unknown -> 6 point correspondence
      - Use more to handle noisy data

$$\mathbf{p} = M\mathbf{P} \Rightarrow \mathbf{p}_i = \begin{bmatrix} u_i \\ v_i \end{bmatrix} = M\mathbf{P}_i = \begin{bmatrix} \frac{\mathbf{P}_i^T \mathbf{m}_1}{\mathbf{P}_i^T \mathbf{m}_3} \\ \\ \frac{\mathbf{P}_i^T \mathbf{m}_2}{\mathbf{P}_i^T \mathbf{m}_3} \end{bmatrix} \Rightarrow \begin{matrix} \mathbf{P}_i^T \mathbf{m}_1 - u_i(\mathbf{P}_i^T \mathbf{m}_3) = 0 \\ \mathbf{P}_i^T \mathbf{m}_2 - v_i(\mathbf{P}_i^T \mathbf{m}_3) = 0 \end{matrix}$$

$\mathbf{m}_1$, $\mathbf{m}_2$, $\mathbf{m}_3$: the three rows of the projection matrix M

General Idea

$$\mathbf{P}_i^T\mathbf{m}_1 - u_i(\mathbf{P}_i^T\mathbf{m}_3) = 0$$
$$\mathbf{P}_i^T\mathbf{m}_2 - v_i(\mathbf{P}_i^T\mathbf{m}_3) = 0$$

Constraints from one pair

$$\mathbf{P}_1^T\mathbf{m}_1 - u_1(\mathbf{P}_1^T\mathbf{m}_3) = 0$$
$$\mathbf{P}_1^T\mathbf{m}_2 - v_1(\mathbf{P}_1^T\mathbf{m}_3) = 0$$
$$\vdots$$
$$\mathbf{P}_n^T\mathbf{m}_1 - u_n(\mathbf{P}_n^T\mathbf{m}_3) = 0$$
$$\mathbf{P}_n^T\mathbf{m}_2 - v_n(\mathbf{P}_n^T\mathbf{m}_3) = 0$$

Equations from n pairs

$$\begin{bmatrix} \mathbf{P}_1^T & 0^T & -u_1\mathbf{P}_1^T \\ 0^T & \mathbf{P}_1^T & -v_1\mathbf{P}_1^T \\ & \vdots & \\ \mathbf{P}_n^T & 0^T & -u_n\mathbf{P}_n^T \\ 0^T & \mathbf{P}_n^T & -v_n\mathbf{P}_n^T \end{bmatrix} \begin{bmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \\ \mathbf{m}_3 \end{bmatrix} = P\mathbf{m} = 0$$

2n x 12

12 x 1

What is the dimension of the *P* matrix?
What is the dimension of **m**?

Details: the derivation of the linear system

- The equations $\mathbf{p} = M\mathbf{P}$ $\quad [X, Y, Z]^T \rightarrow [u, v]^T$

$$\begin{bmatrix} su \\ sv \\ s \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$$su = m_{11}X + m_{12}Y + m_{13}Z + m_{14}$$
$$sv = m_{21}X + m_{22}Y + m_{23}Z + m_{24}$$
$$s = m_{31}X + m_{32}Y + m_{33}Z + m_{34}$$

$$u = \frac{m_{11}X + m_{12}Y + m_{13}Z + m_{14}}{m_{31}X + m_{32}Y + m_{33}Z + m_{34}}$$
$$v = \frac{m_{21}X + m_{22}Y + m_{23}Z + m_{24}}{m_{31}X + m_{32}Y + m_{33}Z + m_{34}}$$

Details: the derivation of the linear system

- The equations

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

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

$$\Rightarrow \quad \begin{aligned} (m_{31}X + m_{32}Y + m_{33}Z + m_{34})\,u &= m_{11}X + m_{12}Y + m_{13}Z + m_{14} \\ (m_{31}X + m_{32}Y + m_{33}Z + m_{34})\,v &= m_{21}X + m_{22}Y + m_{23}Z + m_{24} \end{aligned}$$

$$\Rightarrow \quad \begin{aligned} m_{11}X + m_{12}Y + m_{13}Z + m_{14} - m_{31}uX - m_{32}uY - m_{33}uZ - m_{34}u &= 0 \\ m_{21}X + m_{22}Y + m_{23}Z + m_{24} - m_{31}vX - m_{32}vY - m_{33}vZ - m_{34}v &= 0 \end{aligned}$$

Details: the derivation of the linear system

- The equations

For every pair of 3D-2D corresponding points

$$m_{11}X + m_{12}Y + m_{13}Z + m_{14} - m_{31}uX - m_{32}uY - m_{33}uZ - m_{34}u = 0$$
$$m_{21}X + m_{22}Y + m_{23}Z + m_{24} - m_{31}vX - m_{32}vY - m_{33}vZ - m_{34}v = 0$$

Given n pairs of 3D-2D corresponding points

$$\begin{bmatrix} X_1 & Y_1 & Z_1 & 1 & 0 & 0 & 0 & 0 & -u_1X_1 & -u_1Y_1 & -u_1Z_1 & -u_1 \\ 0 & 0 & 0 & 0 & X_1 & Y_1 & Z_1 & 1 & -v_1X_1 & -v_1Y_1 & -v_1Z_1 & -v_1 \\ & & & & & & \vdots & & & & & \\ X_n & Y_n & Z_n & 1 & 0 & 0 & 0 & 0 & -u_nX_n & -u_nY_n & -u_nZ_n & -u_n \\ 0 & 0 & 0 & 0 & X_n & Y_n & Z_n & 1 & -v_nX_n & -v_nY_n & -v_nZ_n & -v_n \end{bmatrix} \begin{bmatrix} m_{11} \\ m_{12} \\ m_{13} \\ m_{14} \\ m_{21} \\ m_{22} \\ m_{23} \\ m_{24} \\ m_{31} \\ m_{32} \\ m_{33} \\ m_{34} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}$$

Details: the derivation of the linear system

- The equations

$$\begin{bmatrix} X_1 & Y_1 & Z_1 & 1 & 0 & 0 & 0 & 0 & -u_1X_1 & -u_1Y_1 & -u_1Z_1 & -u_1 \\ 0 & 0 & 0 & 0 & X_1 & Y_1 & Z_1 & 1 & -v_1X_1 & -v_1Y_1 & -v_1Z_1 & -v_1 \\ & & & & & & \vdots & & & & & \\ X_n & Y_n & Z_n & 1 & 0 & 0 & 0 & 0 & -u_nX_n & -u_nY_n & -u_nZ_n & -u_n \\ 0 & 0 & 0 & 0 & X_n & Y_n & Z_n & 1 & -v_nX_n & -v_nY_n & -v_nZ_n & -v_n \end{bmatrix} \begin{bmatrix} m_{11} \\ m_{12} \\ m_{13} \\ m_{14} \\ m_{21} \\ m_{22} \\ m_{23} \\ m_{24} \\ m_{31} \\ m_{32} \\ m_{33} \\ m_{34} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}$$

Simplified notation

$$\begin{bmatrix} \mathbf{P}_1^T & 0^T & -u_1\mathbf{P}_1^T \\ 0^T & \mathbf{P}_1^T & -v_1\mathbf{P}_1^T \\ & \vdots & \\ \mathbf{P}_n^T & 0^T & -u_n\mathbf{P}_n^T \\ 0^T & \mathbf{P}_n^T & -v_n\mathbf{P}_n^T \end{bmatrix} \begin{bmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \\ \mathbf{m}_3 \end{bmatrix} = P\mathbf{m} = 0$$

General Idea

- How to solve it?
  - It is a homogeneous linear system
  - It is overdetermined

$$\begin{bmatrix} \mathbf{P}_1^T & 0^T & -u_1\mathbf{P}_1^T \\ 0^T & \mathbf{P}_1^T & -v_1\mathbf{P}_1^T \\ & \vdots & \\ \mathbf{P}_n^T & 0^T & -u_n\mathbf{P}_n^T \\ 0^T & \mathbf{P}_n^T & -v_n\mathbf{P}_n^T \end{bmatrix} \begin{bmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \\ \mathbf{m}_3 \end{bmatrix} = P\mathbf{m} = 0$$

General Idea

- How to solve it?
  - $\mathbf{m} = 0$ is always a trivial solution
  - If $\mathbf{m} \neq 0$ is a solution, then any $k * \mathbf{m}$ is also a solution

$$\begin{bmatrix} \mathbf{P}_1^T & 0^T & -u_1\mathbf{P}_1^T \\ 0^T & \mathbf{P}_1^T & -v_1\mathbf{P}_1^T \\ & \vdots & \\ \mathbf{P}_n^T & 0^T & -u_n\mathbf{P}_n^T \\ 0^T & \mathbf{P}_n^T & -v_n\mathbf{P}_n^T \end{bmatrix} \begin{bmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \\ \mathbf{m}_3 \end{bmatrix} = P\mathbf{m} = 0$$

General Idea

- How to solve it?
  - $\mathbf{m} = 0$ is always a trivial solution
  - If $\mathbf{m} \neq 0$ is a solution, then any $k * \mathbf{m}$ is also a solution
  - Constrained optimization

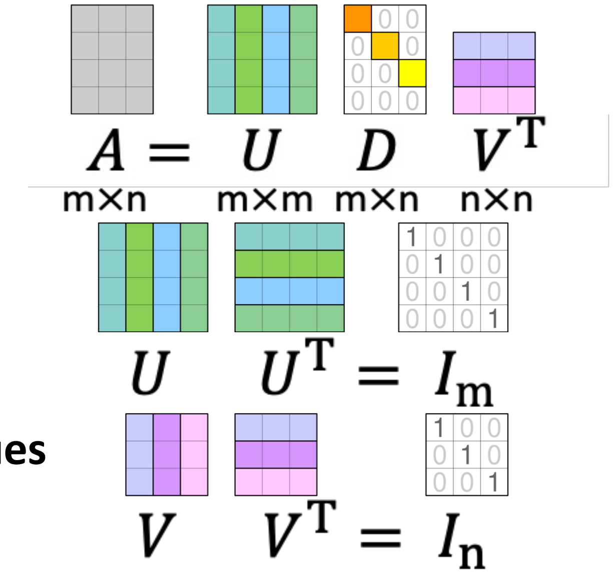
$$\begin{bmatrix} \mathbf{P}_1^T & 0^T & -u_1\mathbf{P}_1^T \\ 0^T & \mathbf{P}_1^T & -v_1\mathbf{P}_1^T \\ & \vdots & \\ \mathbf{P}_n^T & 0^T & -u_n\mathbf{P}_n^T \\ 0^T & \mathbf{P}_n^T & -v_n\mathbf{P}_n^T \end{bmatrix} \begin{bmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \\ \mathbf{m}_3 \end{bmatrix} = P\mathbf{m} = 0 \quad \Rightarrow \quad \begin{aligned} &\underset{\mathbf{m}}{\text{minimize}} && \|P\mathbf{m}\|^2 \\ &\text{subject to} && \|\mathbf{m}\|^2 = 1 \end{aligned}$$

SVD

- **S**ingular **V**alue **D**ecomposition
  - Generalization of the eigen-decomposition of a square matrix to any *m* by *n* matrix

$$A = UDV^{\mathrm{T}} \qquad D = \begin{bmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_N \end{bmatrix}$$

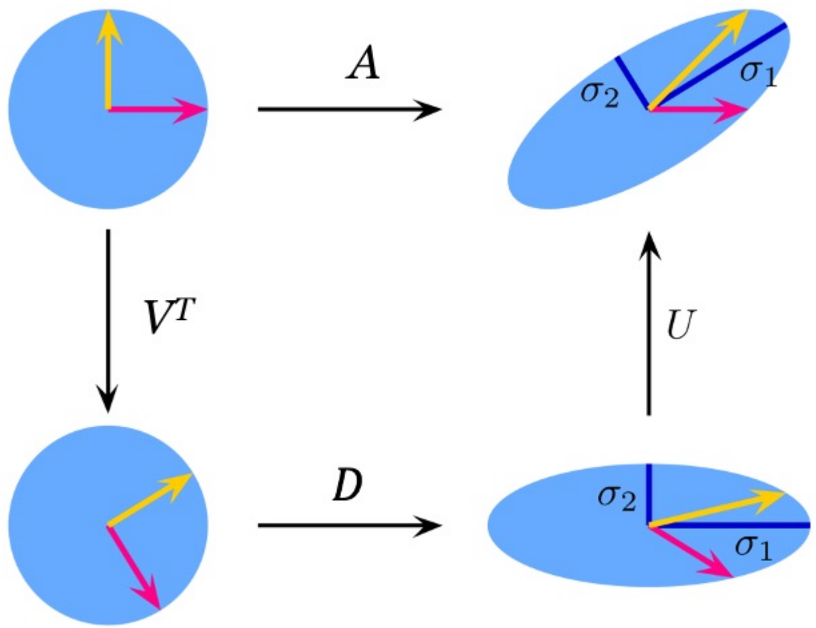
U: an m by m orthogonal matrix
D: an m by n diagonal matrix; entries on diagonal called **singular values**
V: an n by n orthogonal *matrix*



SVD

- Geometric meaning

$$A = UDV^{\mathrm{T}}$$



Example (square matrix)

$$\begin{bmatrix} 3 & -2 \\ 1 & 5 \end{bmatrix} = \begin{bmatrix} -.40 & .916 \\ .916 & .40 \end{bmatrix} \cdot \begin{bmatrix} 5.39 & 0 \\ 0 & 3.154 \end{bmatrix} \cdot \begin{bmatrix} -.05 & .999 \\ .999 & .05 \end{bmatrix}$$

A	U	D	V^{T}
Transformation	Rotation	Scaling	Rotation

Calibration: solve for projection matrix

$$Pm = 0 \Rightarrow \min_{\mathbf{m}} \|P\mathbf{m}\|^2 \quad \text{subject to} \quad \|\mathbf{m}\|^2 = 1$$

SVD of P

$$U_{2n\times 2n} D_{2n\times 12} V^T_{12\times 12}$$

Last column of V gives $\mathbf{m}$

(Why? See page 593 of Hartley & Zisserman. Multiple View Geometry in Computer Vision)

Least-squares solution of homogeneous equations

This problem is solvable as follows. Let $\mathtt{A} = \mathtt{UDV}^\mathsf{T}$. The problem then requires us to minimize $\|\mathtt{UDV}^\mathsf{T}\mathbf{x}\|$. However, $\|\mathtt{UDV}^\mathsf{T}\mathbf{x}\| = \|\mathtt{DV}^\mathsf{T}\mathbf{x}\|$ and $\|\mathbf{x}\| = \|\mathtt{V}^\mathsf{T}\mathbf{x}\|$. Thus, we need to minimize $\|\mathtt{DV}^\mathsf{T}\mathbf{x}\|$ subject to the condition $\|\mathtt{V}^\mathsf{T}\mathbf{x}\| = 1$. We write $\mathbf{y} = \mathtt{V}^\mathsf{T}\mathbf{x}$, and the problem is: minimize $\|\mathtt{D}\mathbf{y}\|$ subject to $\|\mathbf{y}\| = 1$. Now, $\mathtt{D}$ is a diagonal matrix with its diagonal entries in descending order. It follows that the solution to this problem is $\mathbf{y} = (0, 0, \ldots, 0, 1)^\mathsf{T}$ having one non-zero entry, 1 in the last position. Finally $\mathbf{x} = \mathtt{V}\mathbf{y}$ is simply the last column of $\mathtt{V}$. The method is summarized in algorithm A5.4.

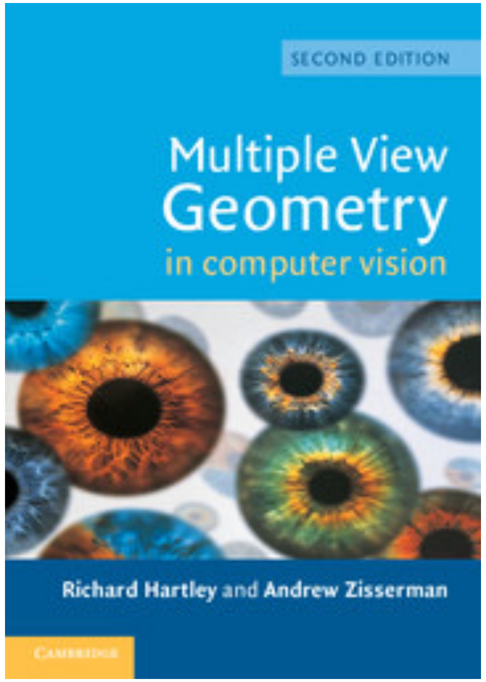
Objective

Given a matrix $\mathtt{A}$ with at least as many rows as columns, find $\mathbf{x}$ that minimizes $\|\mathtt{A}\mathbf{x}\|$ subject to $\|\mathbf{x}\| = 1$.

Solution

$\mathbf{x}$ is the last column of $\mathtt{V}$, where $\mathtt{A} = \mathtt{UDV}^\mathsf{T}$ is the SVD of $\mathtt{A}$.

Algorithm A5.4. *Least-squares solution of a homogeneous system of linear equations.*



Page 593 of Hartley & Zisserman. Multiple View Geometry in Computer Vision

Camera parameters from project matrix

$$M = K\begin{bmatrix} R & \mathbf{t} \end{bmatrix}$$

$$K = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \alpha & -\alpha\cot\theta & c_x \\ 0 & \frac{\beta}{\sin\theta} & c_y \\ 0 & 0 & 1 \end{bmatrix}, \quad R = \begin{bmatrix} \mathbf{r}_1^T \\ \mathbf{r}_2^T \\ \mathbf{r}_3^T \end{bmatrix}, \quad \mathbf{t} = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}$$

$$M = \begin{bmatrix} \alpha\mathbf{r}_1^T - \alpha\cot\theta\mathbf{r}_2^T + c_x\mathbf{r}_3^T & \alpha t_x - \alpha\cot\theta t_y + c_x t_z \\ \frac{\beta}{\sin\theta}\mathbf{r}_2^T + c_y\mathbf{r}_3^T & \frac{\beta}{\sin\theta}t_y + c_y t_z \\ \mathbf{r}_3^T & t_z \end{bmatrix}$$

SVD-solved projection matrix

SVD-solved projection matrix is known up to scale, i.e., $\rho\mathcal{M} = M$

The true values of project matrix

$$\mathcal{M} = \frac{1}{\rho}M = \frac{1}{\rho}\begin{bmatrix} \alpha\mathbf{r}_1^T - \alpha\cot\theta\mathbf{r}_2^T + c_x\mathbf{r}_3^T & \alpha t_x - \alpha\cot\theta t_y + c_x t_z \\ \frac{\beta}{\sin\theta}\mathbf{r}_2^T + c_y\mathbf{r}_3^T & \frac{\beta}{\sin\theta}t_y + c_y t_z \\ \mathbf{r}_3^T & t_z \end{bmatrix}$$

Camera parameters from project matrix

$$\mathcal{M} = \frac{1}{\rho}\begin{bmatrix} \alpha\mathbf{r}_1^T - \alpha\cot\theta\mathbf{r}_2^T + c_x\mathbf{r}_3^T & \alpha t_x - \alpha\cot\theta t_y + c_x t_z \\ \frac{\beta}{\sin\theta}\mathbf{r}_2^T + c_y\mathbf{r}_3^T & \frac{\beta}{\sin\theta}t_y + c_y t_z \\ \mathbf{r}_3^T & t_z \end{bmatrix}$$

$$\text{denote } \mathcal{M} = \begin{bmatrix} A & \mathbf{b} \end{bmatrix} = \begin{bmatrix} \mathbf{a}_1^T & b_1 \\ \mathbf{a}_2^T & b_2 \\ \mathbf{a}_3^T & b_3 \end{bmatrix}$$

$$\frac{1}{\rho}\begin{bmatrix} \alpha\mathbf{r}_1^T - \alpha\cot\theta\mathbf{r}_2^T + c_x\mathbf{r}_3^T & \alpha t_x - \alpha\cot\theta t_y + c_x t_z \\ \frac{\beta}{\sin\theta}\mathbf{r}_2^T + c_y\mathbf{r}_3^T & \frac{\beta}{\sin\theta}t_y + c_y t_z \\ \mathbf{r}_3^T & t_z \end{bmatrix} = \begin{bmatrix} \mathbf{a}_1^T & b_1 \\ \mathbf{a}_2^T & b_2 \\ \mathbf{a}_3^T & b_3 \end{bmatrix}$$

Solving for the intrinsic and extrinsic parameters

Camera parameters from project matrix

Intrinsic parameters:

$$\rho = \pm \frac{1}{\|\mathbf{a}_3\|}$$

$$c_x = \rho^2 (\mathbf{a}_1 \cdot \mathbf{a}_3)$$

$$c_y = \rho^2 (\mathbf{a}_2 \cdot \mathbf{a}_3)$$

$$\cos\theta = -\frac{(\mathbf{a}_1 \times \mathbf{a}_3) \cdot (\mathbf{a}_2 \times \mathbf{a}_3)}{\|\mathbf{a}_1 \times \mathbf{a}_3\| \cdot \|\mathbf{a}_2 \times \mathbf{a}_3\|}$$

$$\alpha = \rho^2 \|\mathbf{a}_1 \times \mathbf{a}_3\| \sin\theta$$

$$\beta = \rho^2 \|\mathbf{a}_2 \times \mathbf{a}_3\| \sin\theta$$

Extrinsic parameters:

$$\mathbf{r}_1 = \frac{\mathbf{a}_2 \times \mathbf{a}_3}{\|\mathbf{a}_2 \times \mathbf{a}_3\|}$$

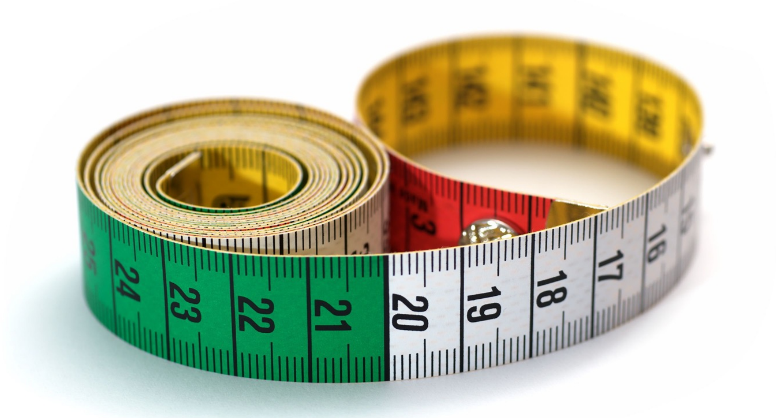
$$\mathbf{r}_3 = \rho \mathbf{a}_3$$

$$\mathbf{r}_2 = \mathbf{r}_3 \times \mathbf{r}_1$$

$$\mathbf{t} = \rho K^{-1} \mathbf{b}$$

Find 3D-2D corresponding points

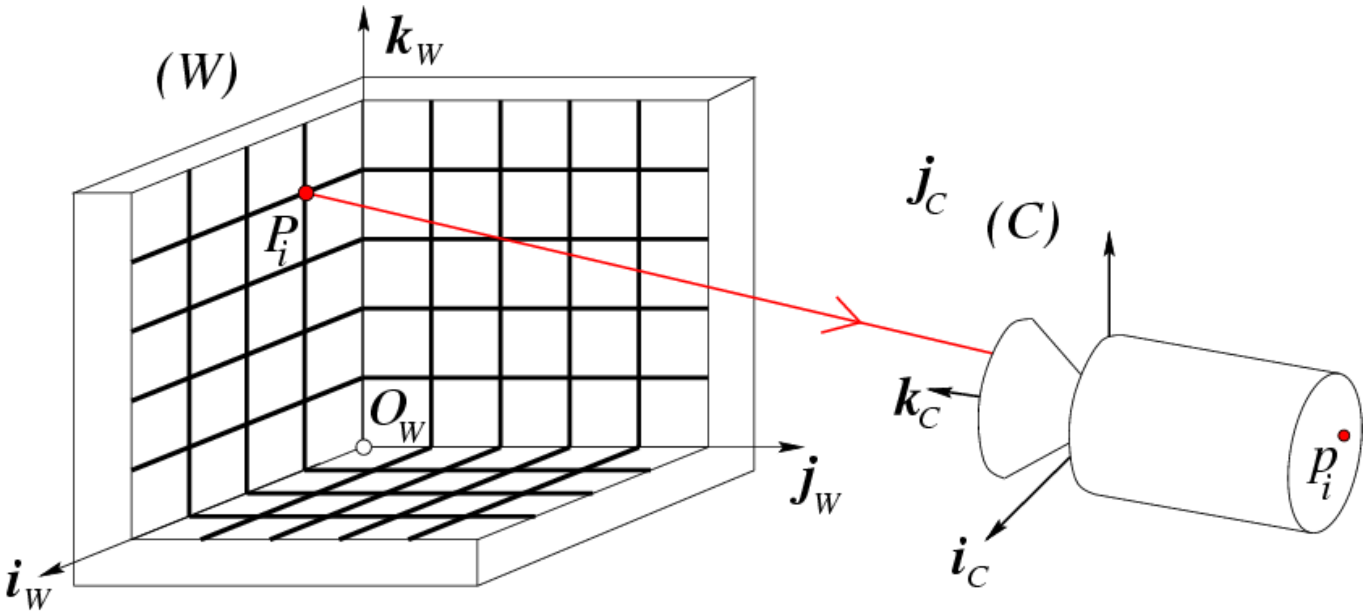
- At least 6 3D-2D point pairs
  - 3D points with known 3D coordinates
  - Corresponding image points with known 2D coordinates



tape measure

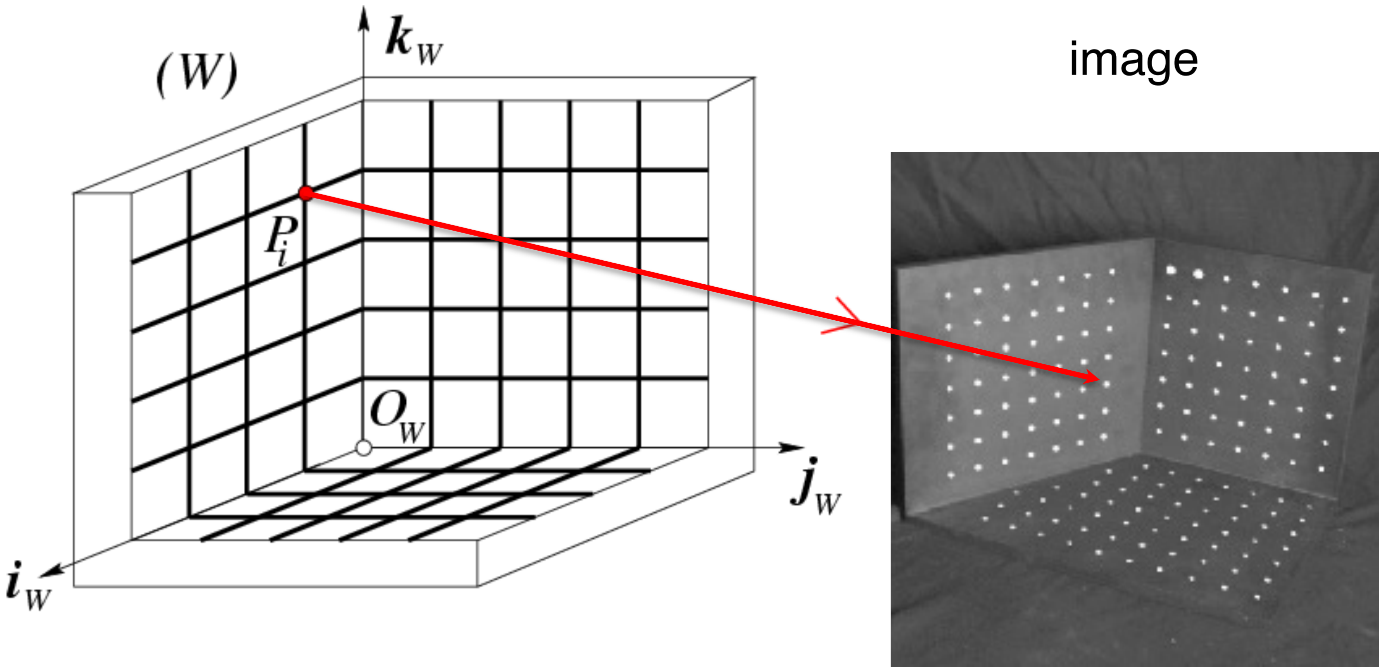
Find 3D-2D corresponding points

- Calibration rig - a special apparatus
  - $P_1, \ldots P_n$ with known positions in $[O_w, i_w, j_w, k_w]$



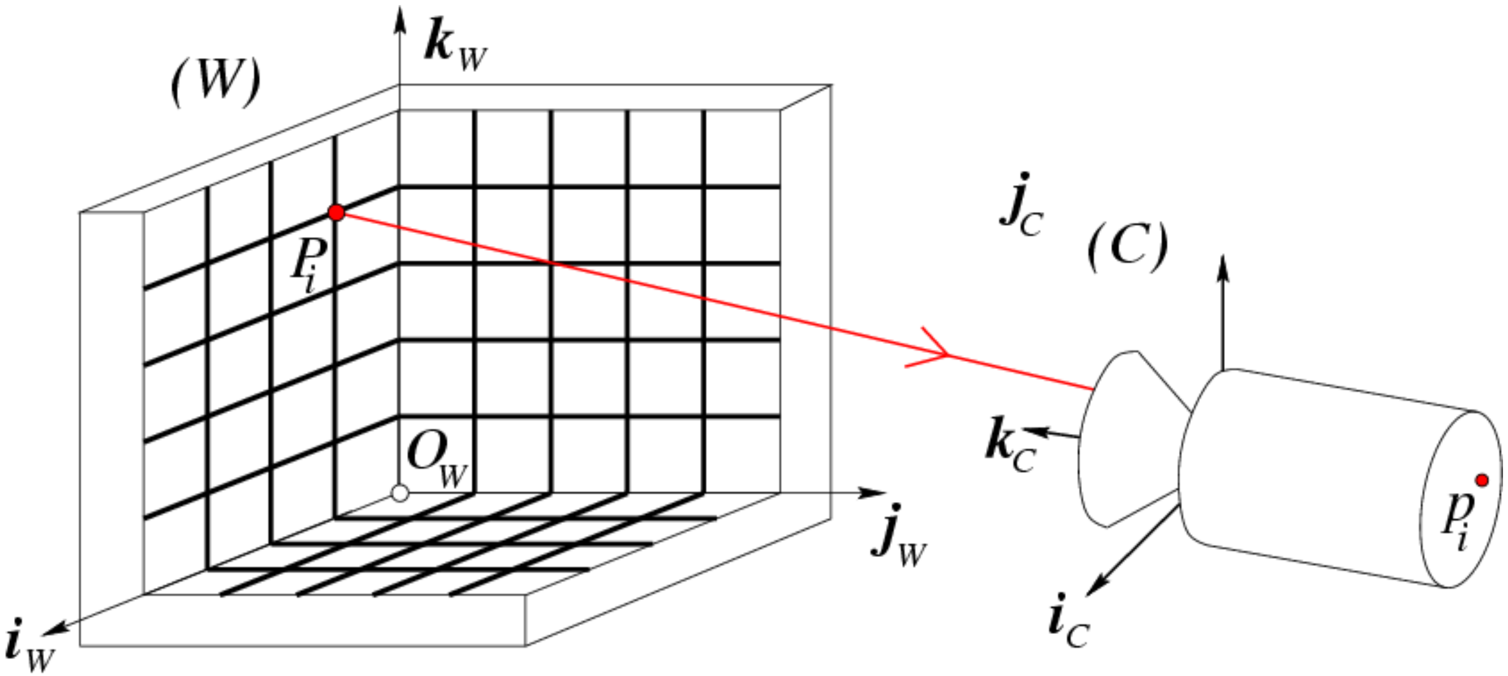
Find 3D-2D corresponding points

- Calibration rig - a special apparatus
  - $P_1, \ldots P_n$ with known positions in $[O_w, i_w, j_w, k_w]$
  - $p_1, \ldots p_n$ known positions in the image
  - At least 6 pairs
- Goal
  - Intrinsic parameters
  - Extrinsic parameters



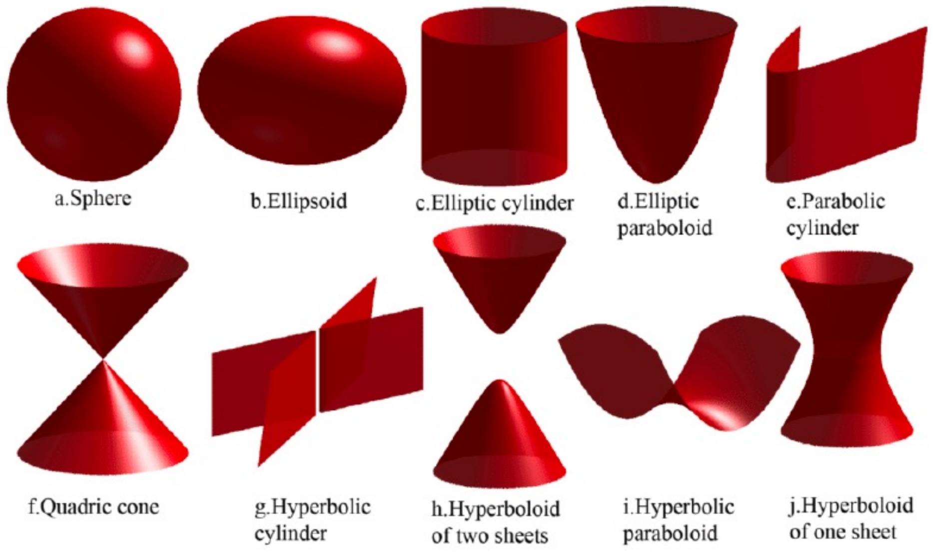
Calibration

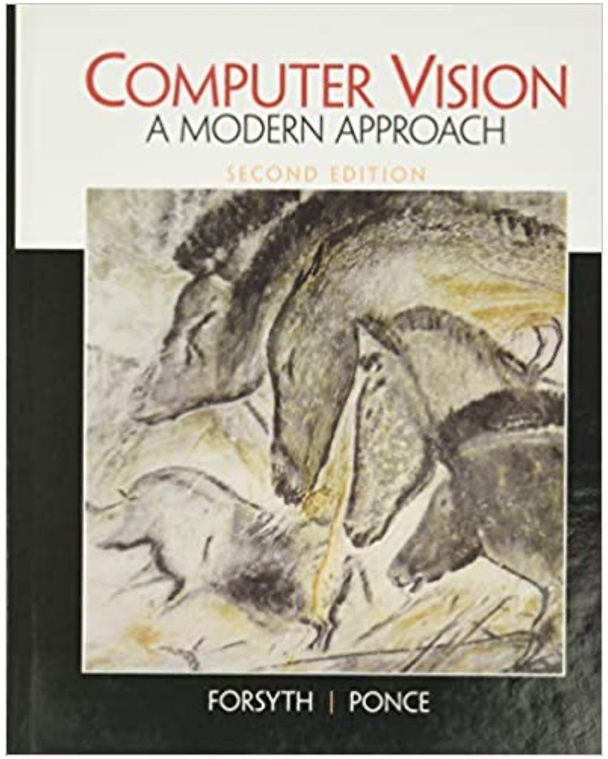
- Always solvable?



Calibration

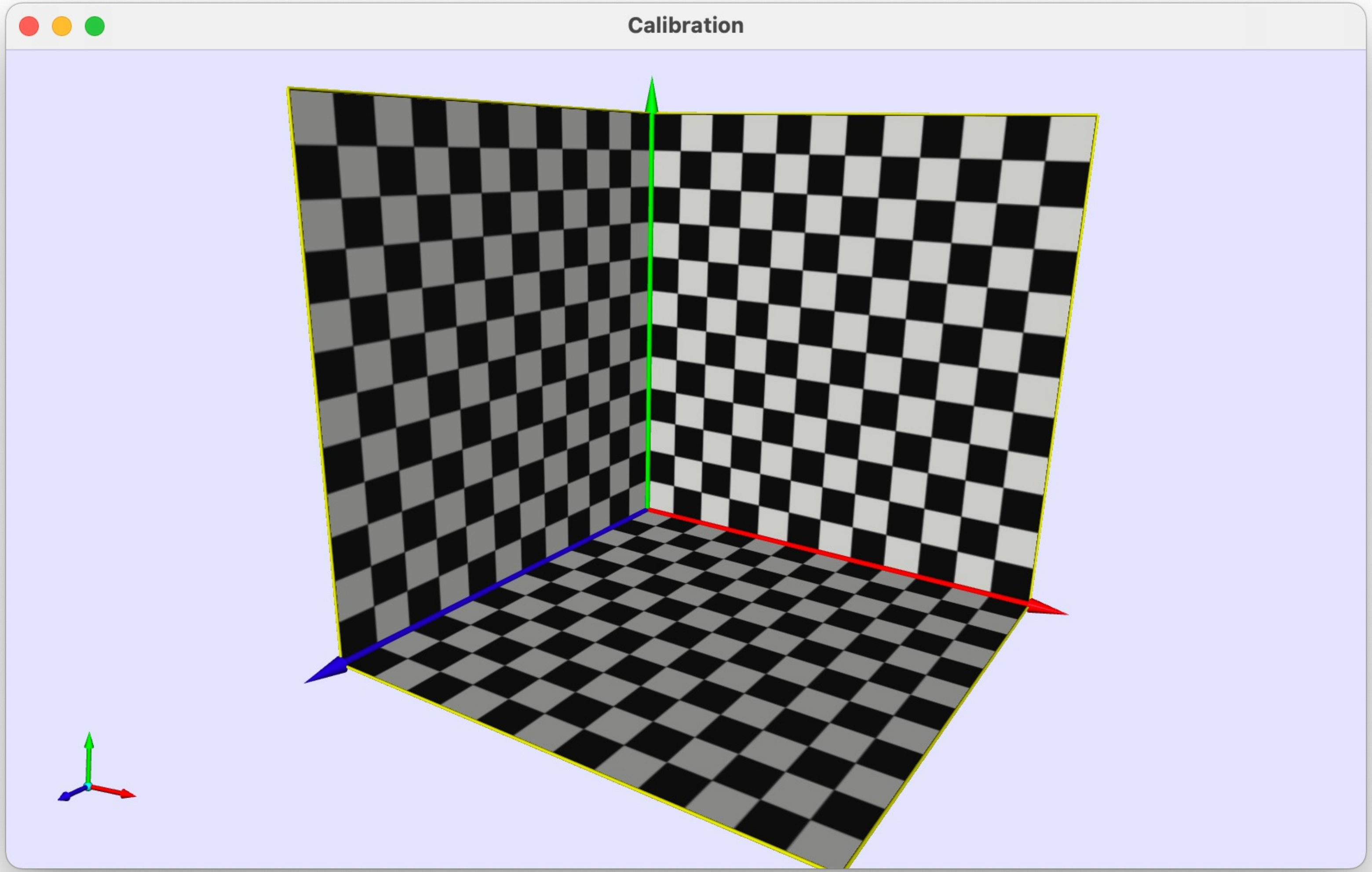
- Always solvable?
  - $\{P_i\}$ cannot lie on the same plane
  - $\{P_i\}$ cannot lie on the intersection curve of two quadric surfaces





See Section 1.3 of Forsyth & Ponce. Computer Vision: A Modern Approach

A1: Camera calibration



Next lecture

- Epipolar geometry