No. 2

カメラ校正

Camera Calibration

担当教員：向川康博・田中賢一郎

Slide credits

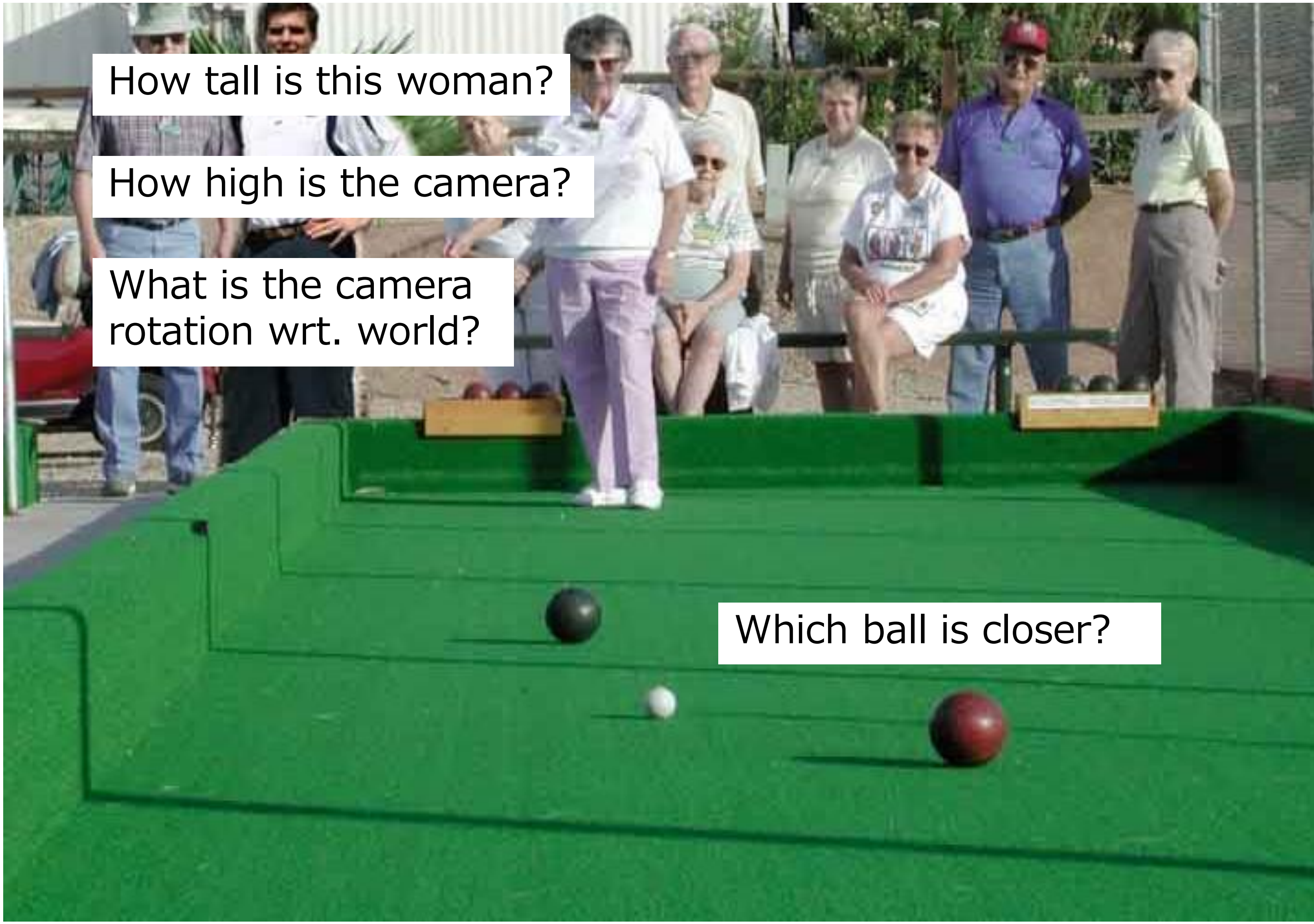
- Special Thanks: Some slides are adopted from other instructors’ slides.
  - Tomokazu Sato, former NAIST CV1 class
  - James Tompkin, Brown CSCI 1430 Fall 2017
  - Ioannis Gkioulekas, CMU 16-385 Spring 2018

  - We also thank many other instructors for sharing their slides.

Today’s mini-report

- There are 10 questions throughout the class.
  - Write your idea.
  - Correct answer is not important, but thinking your own idea is the most important.
  - The answer will be shown immediately.

Cameras and World Geometry

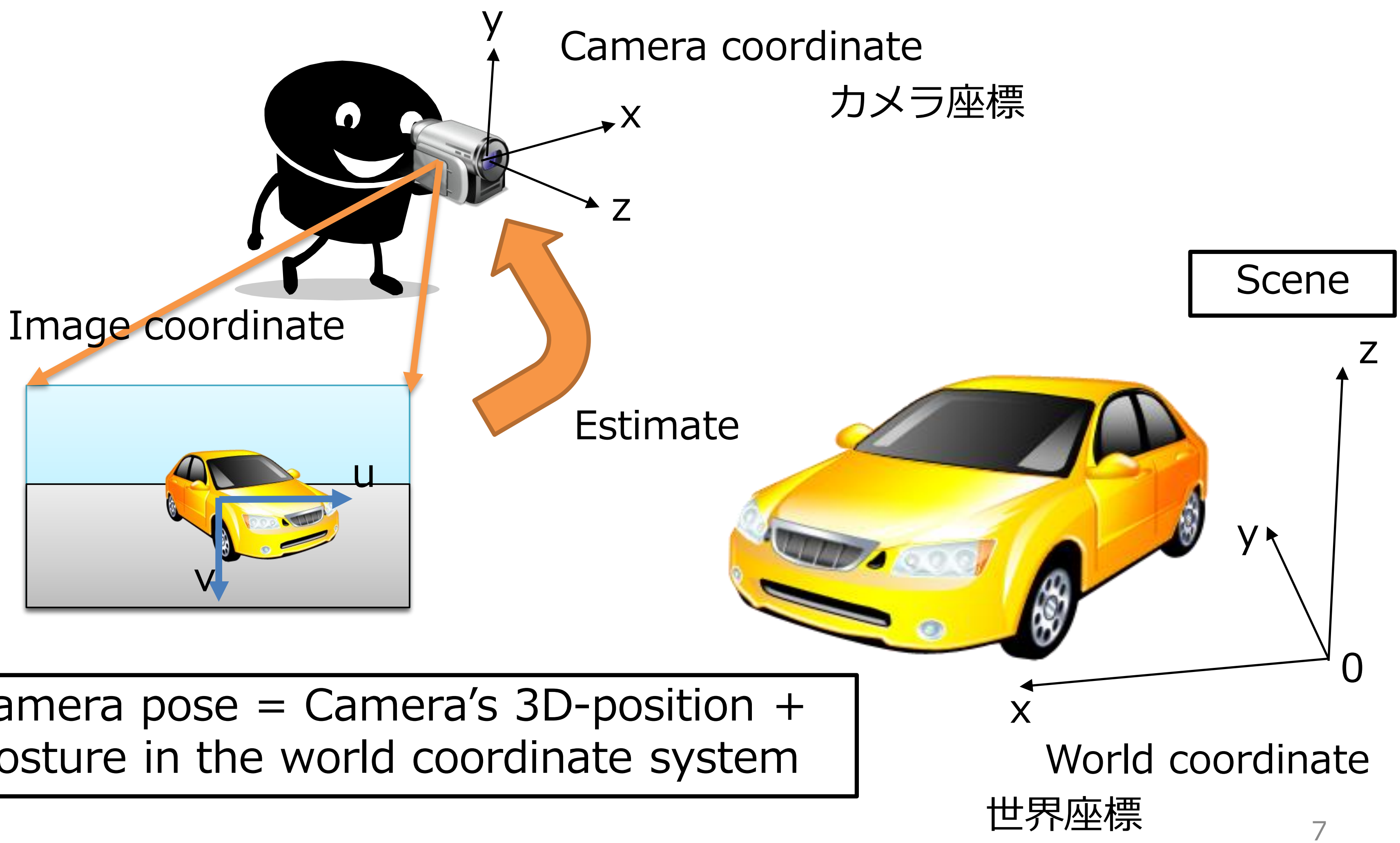


Today's topics

- What is camera pose?
- Coordinate system and transformation
- Projection from 3D to 2D
  - Intrinsic parameters
  - Extrinsic parameters
- Camera pose estimation
  - Linear algorithm
  - Iterative algorithm

What is camera pose?

What is 'camera pose' ?

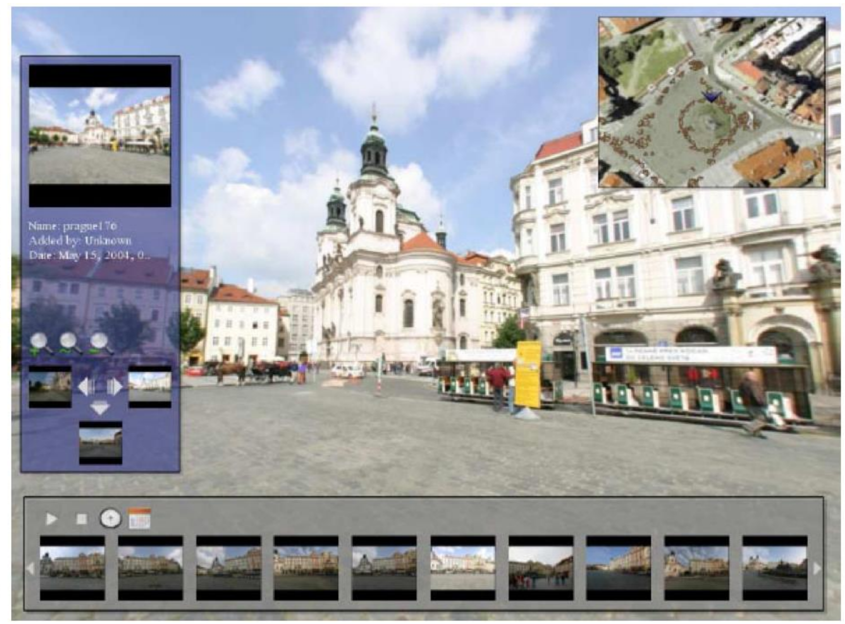


Camera pose = Camera's 3D-position + posture in the world coordinate system

Camera pose estimation

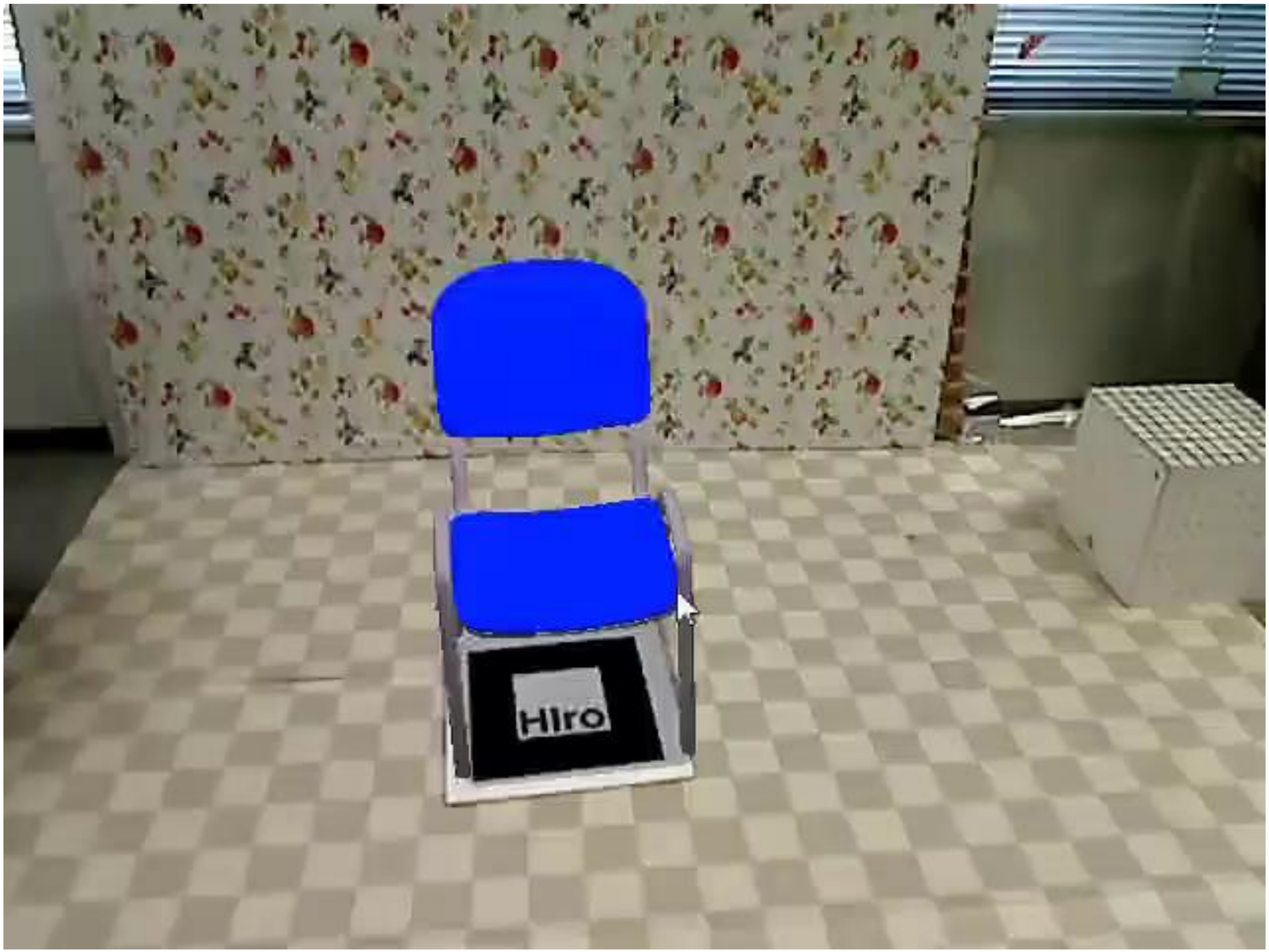
（カメラ位置姿勢推定）

- Given a single image, estimate the exact position of the photographer



Pose estimation

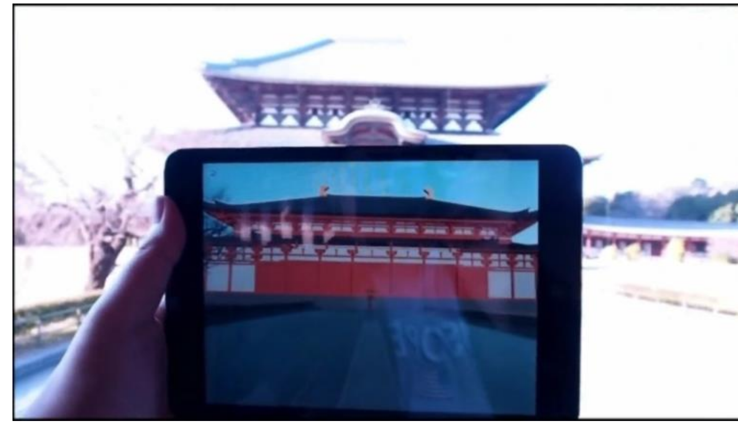
Applications of camera pose estimation
- Marker based Augmented Reality -



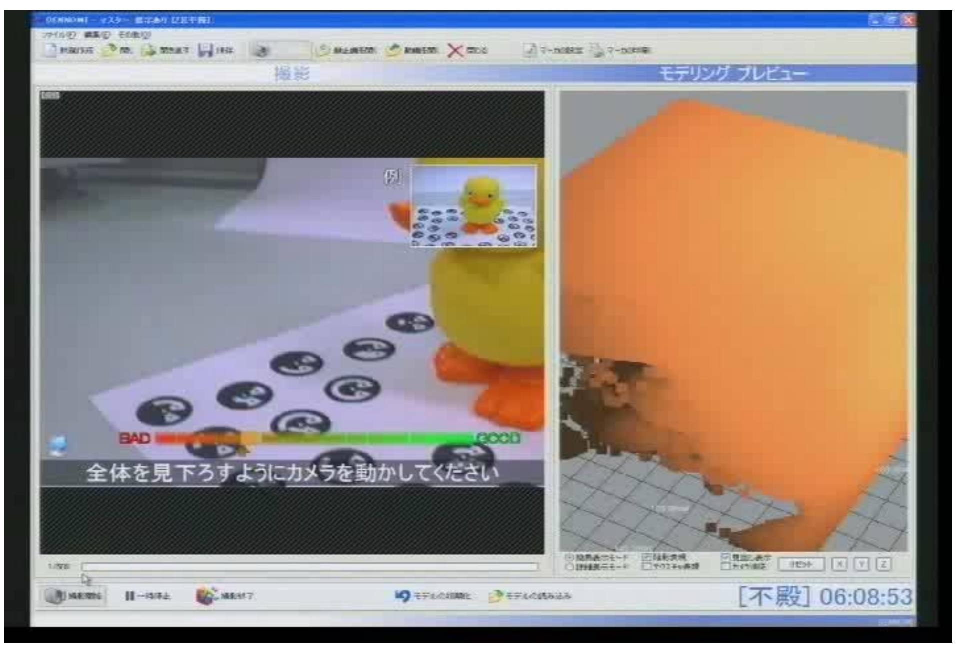
Applications of camera pose estimation

Image mosaic

Augmented reality

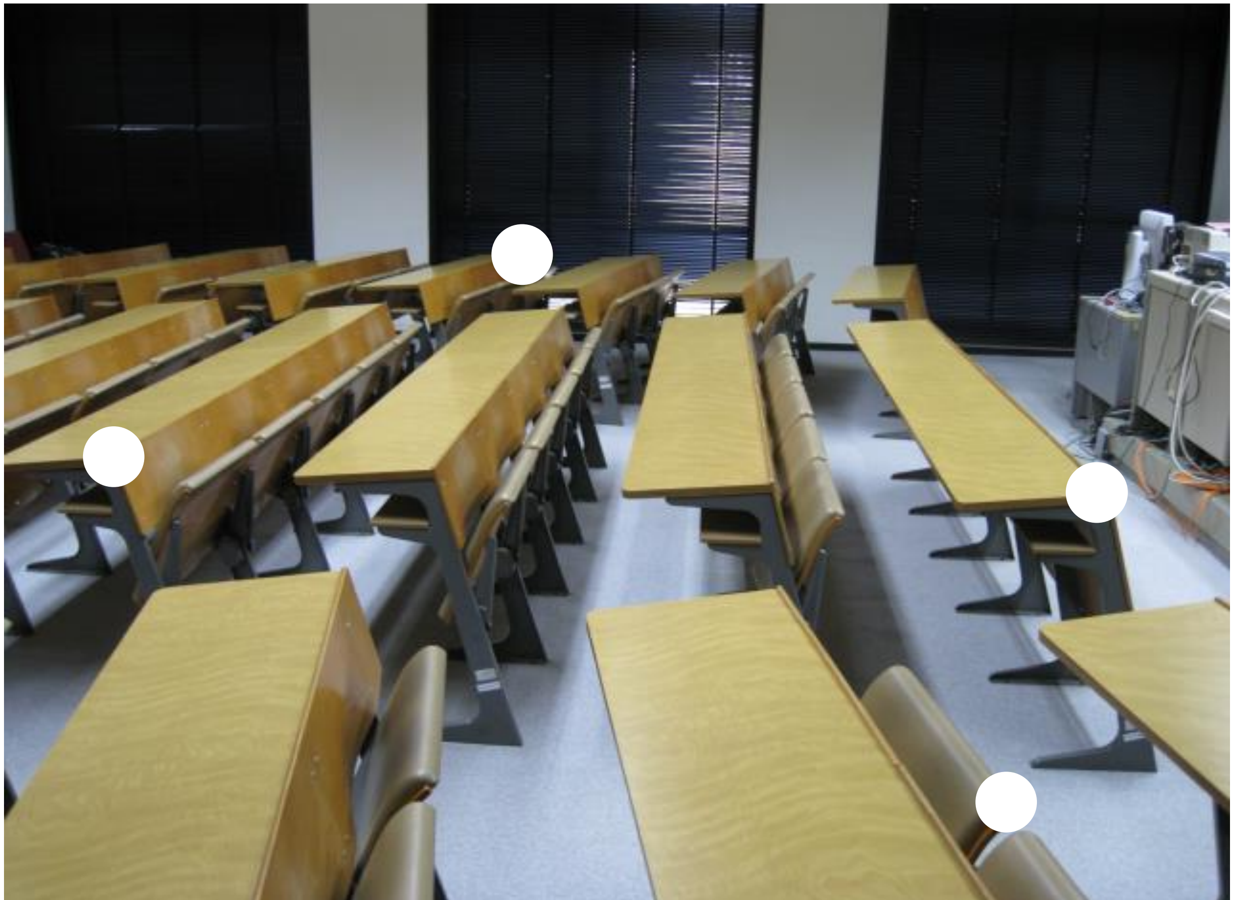


3D modeling



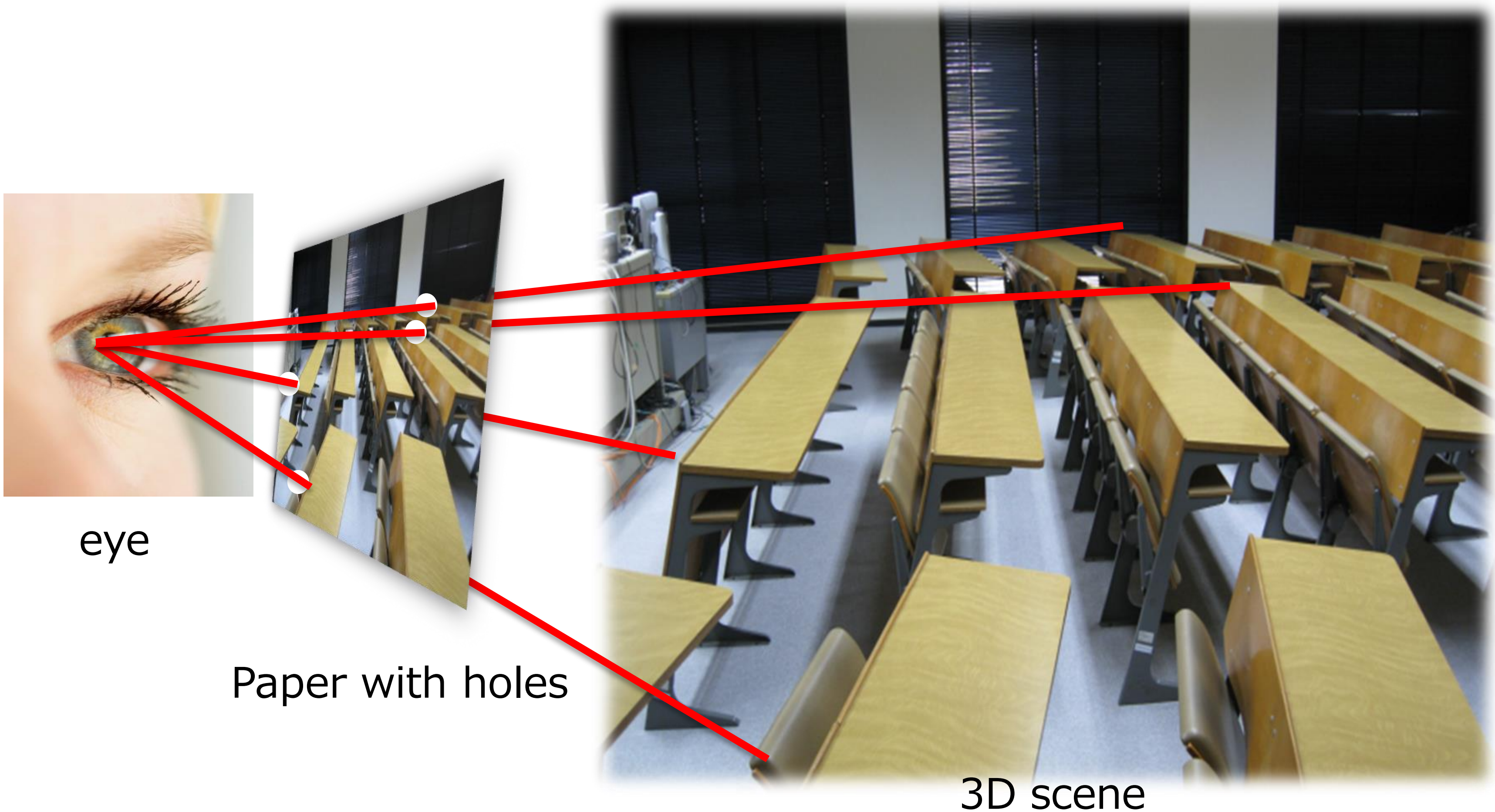
Free viewpoint image rendering

Small experiment:
Find the camera pose for taking this picture



Mini-report (1) Write your idea.
Hint: Why did I make some holes?

Idea for camera pose estimation



By aligning the positions of rays and holes,
you can estimate camera pose from an image.

Camera Pose Estimation

	Structure (scene geometry)	Motion (camera geometry)	Measurements
Pose Estimation	known	estimate	3D to 2D correspondences
Triangulation	estimate	known	2D to 2D correspondences
Reconstruction	estimate	estimate	2D to 2D correspondences

Today’s topics

- What is camera pose?
- **Coordinate system and transformation**
- Projection from 3D to 2D
  - Intrinsic parameters
  - Extrinsic parameters
- Camera pose estimation
  - Linear algorithm
  - Iterative algorithm

How can you express the position and posture?

Q. Where is he?
Which direction is he looking?

A. He is …

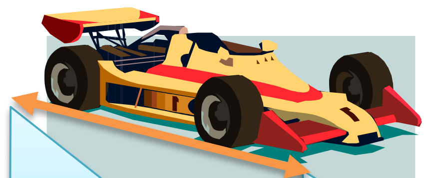
Position:
・He is in Aomori.
・He is at the place 600km north from Tokyo.
・He is at the position of latitude 135 40’xxx, longitude 34 39‘xxx.

Posture:
・He is looking east direction.
・He is looking the pacific.
・He is looking moon.

Common basis to specify the camera pose is necessary.

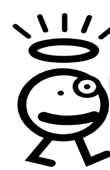
Is GPS coordinate system the best?

Latitude 135 40’xxx
Longitude 34 39‘xxx
Height 10^10km



Latitude 135 40’xxx
Longitude 34 39‘xxx

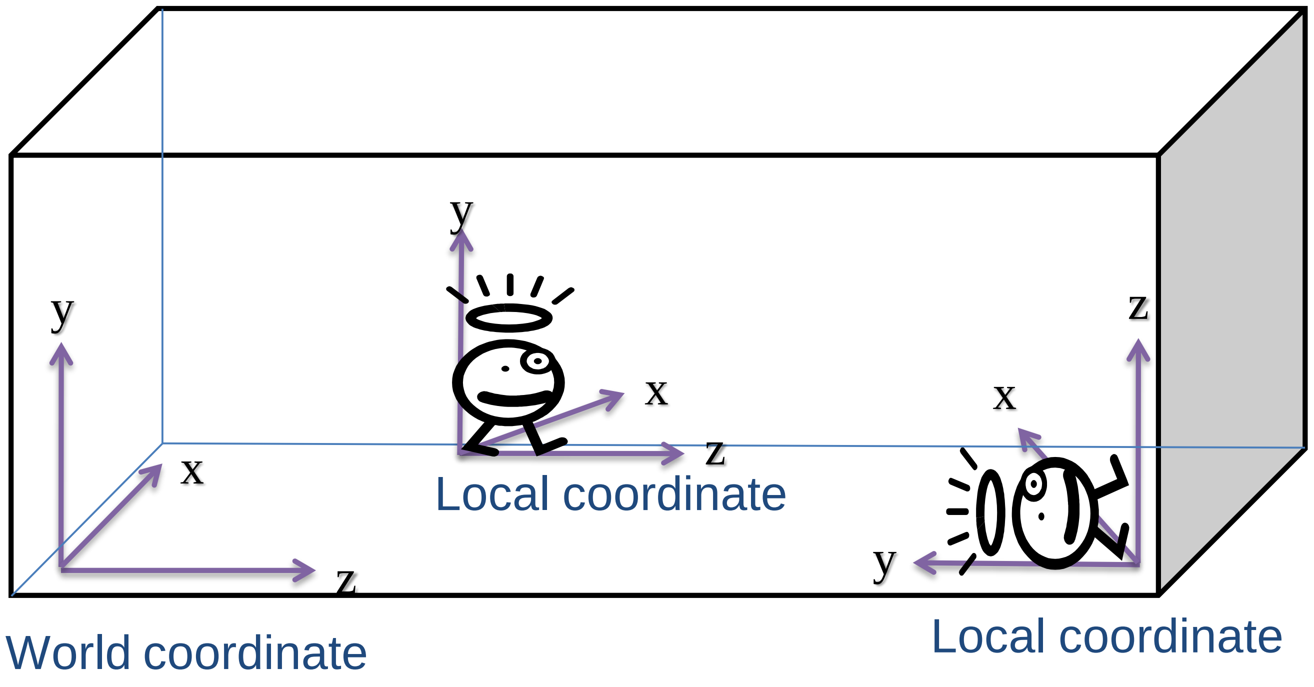
Latitude 135 40’xxx
Longitude 34 39‘xxx



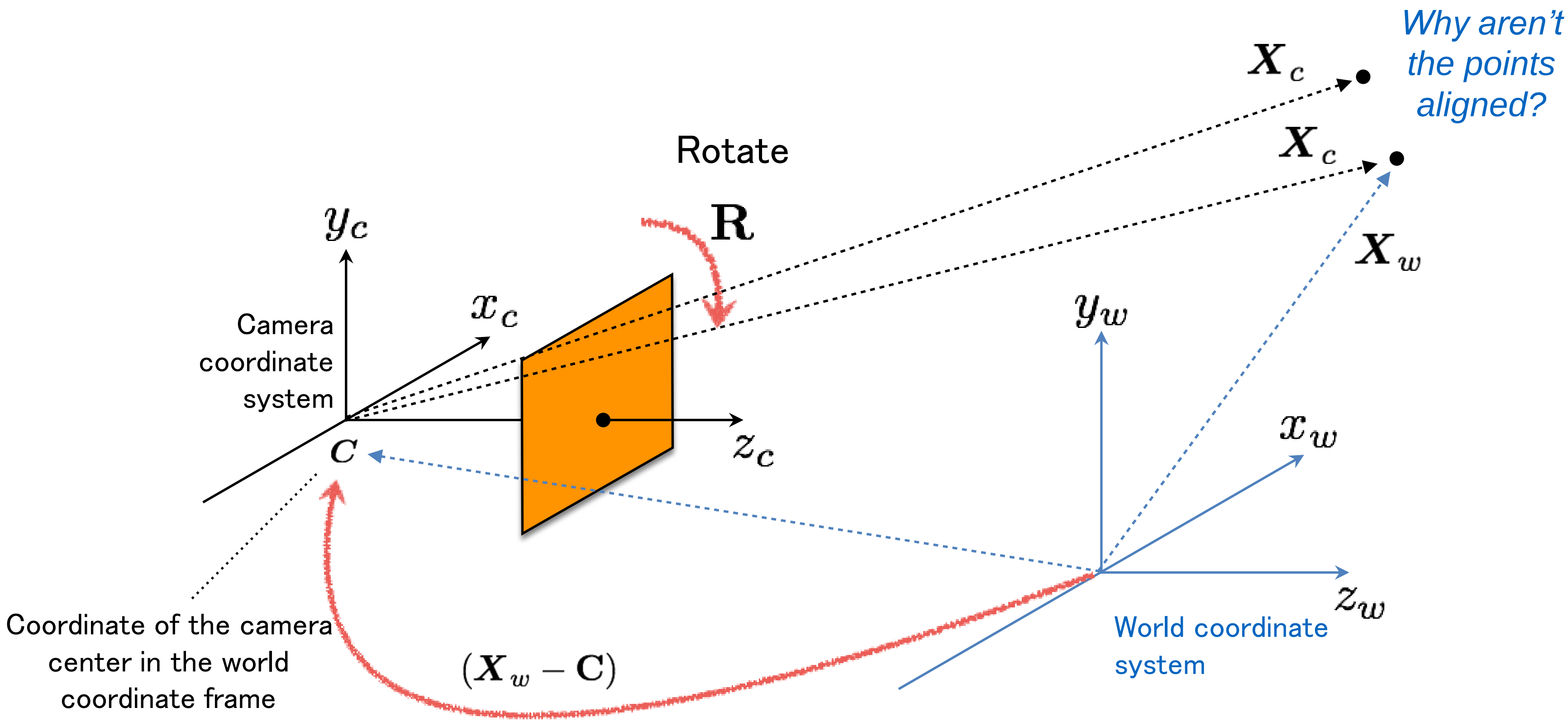
How long is this car?

Absolute and relative position / posture

Absolute position / posture: Position and posture that are defined in some fixed and common **world** coordinate system.
Relative position / posture: Position and posture that are defined in **local** coordinate system.



How to convert coordinate?



$$\boldsymbol{X}_c = \mathbf{R}\,(\boldsymbol{X}_w - \mathbf{C})$$

Rotate Translate

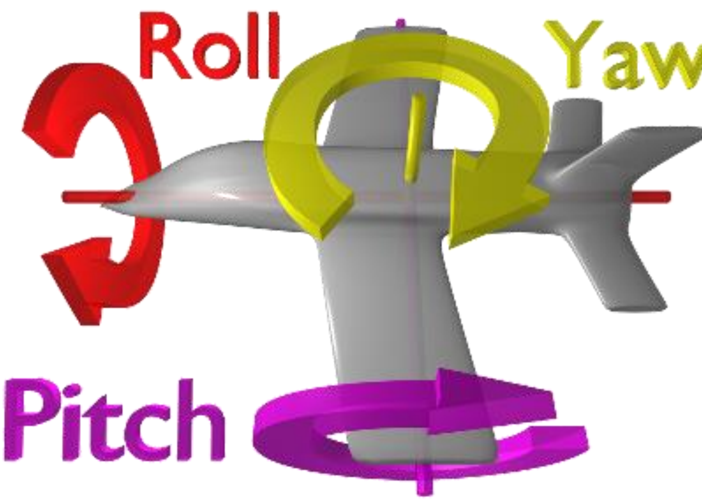
What are R and C?
3D rigid transformation

Rotation parameters

- Roll, Pitch, Yaw: Rotate along X, Y, Z.

$$\mathbf{R} = Z(\alpha)Y(\beta)X(\gamma)$$

$$= \begin{pmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\gamma & -\sin\gamma \\ 0 & \sin\gamma & \cos\gamma \end{pmatrix}$$



https://en.wikipedia.org/wiki/Aircraft_principal_axes

Homogeneous coordinate

(同次座標系・斉次座標系)

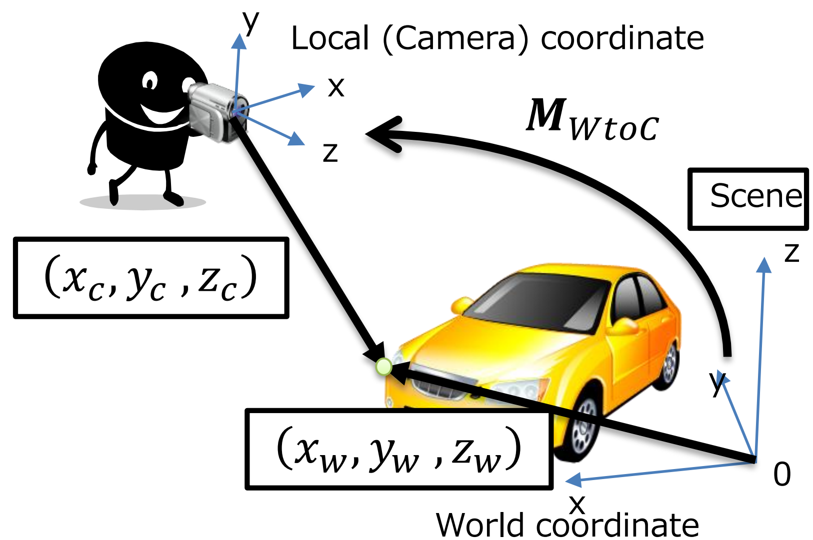
Heterogeneous（非同次）

$$\boldsymbol{X}_c = \mathrm{R}(\boldsymbol{X}_w - \mathbf{C})$$

$$\boldsymbol{X}' = \boldsymbol{R}_2(\boldsymbol{R}_1(\boldsymbol{X} - \boldsymbol{C}_1) - \boldsymbol{C}_2)$$

Homogeneous（同次）

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} \mathrm{R} & -\mathrm{RC} \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$



Transformation matrix $\boldsymbol{M}$ (Extrinsic parameter)

$$\boldsymbol{X}' = \boldsymbol{M}_2\boldsymbol{M}_1\boldsymbol{X}$$

many notations

$$\boldsymbol{M} = \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} \boldsymbol{R} & \boldsymbol{t} \\ \boldsymbol{0}^{\mathrm{T}} & 1 \end{pmatrix} \text{ or } [\boldsymbol{R}|t] \text{ (3x4)}$$

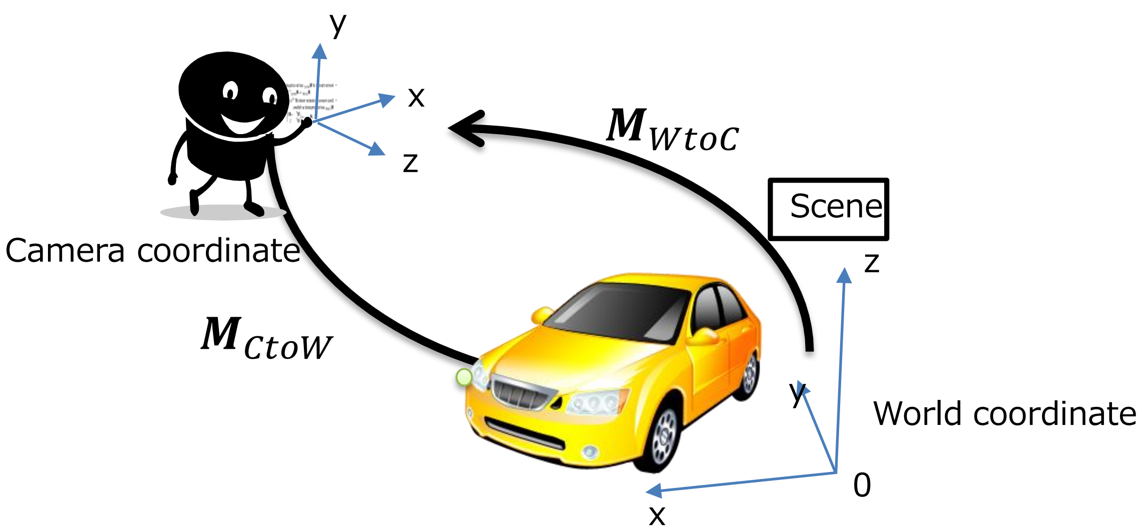
Important characteristics of rigid transform

- Inverse transform of $\boldsymbol{M}_{WtoC}$ can be computed by inverse matrix.

$$\boldsymbol{M}_{CtoW} = \boldsymbol{M}_{WtoC}^{-1}$$

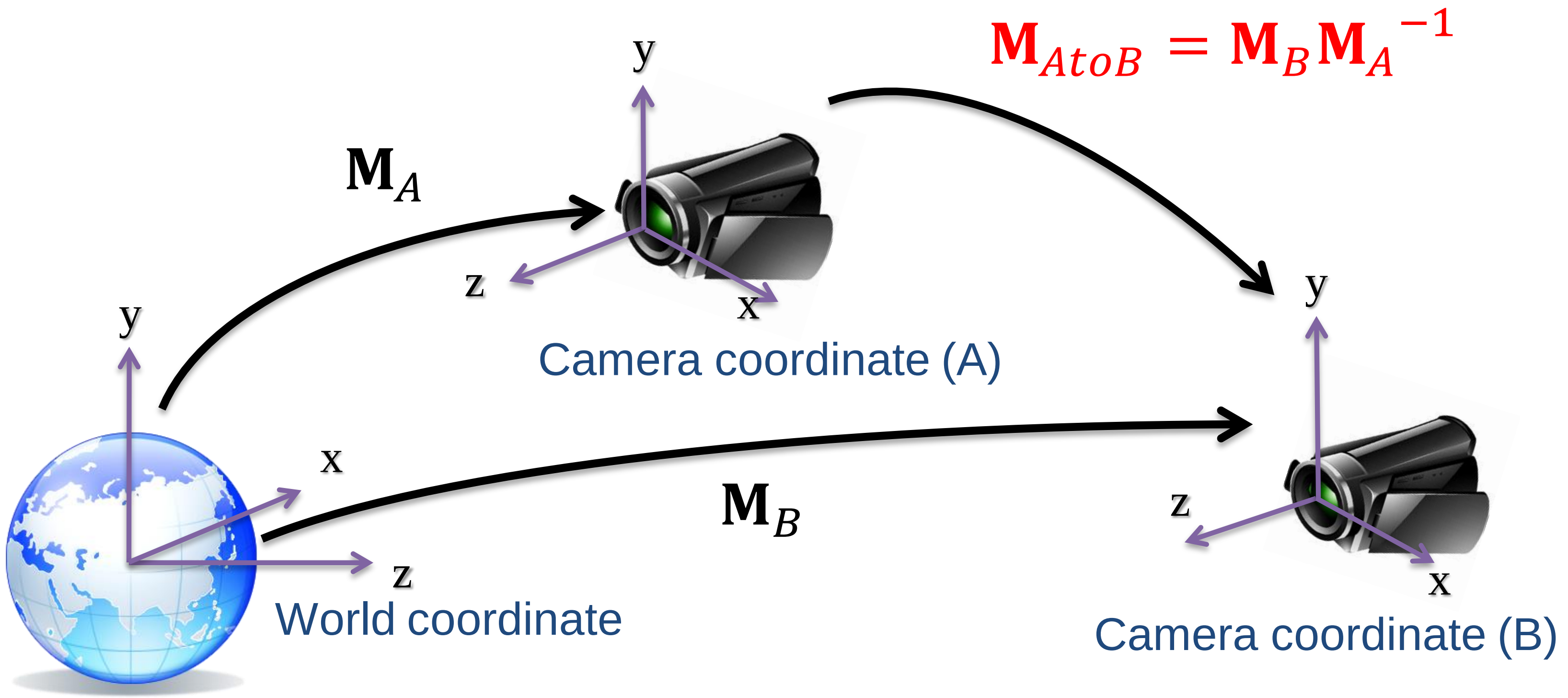
- Since inverse of rotation matrix $\boldsymbol{R}^{-1}$ is coincide with $\boldsymbol{R}^{\top}$, $\boldsymbol{M}_{CtoW}$ can be computed as follows.

$$\boldsymbol{M}_{CtoW} = \begin{pmatrix} \boldsymbol{R}^{\top} & -\boldsymbol{R}^{\top}\boldsymbol{t} \\ \boldsymbol{0}^{\top} & 1 \end{pmatrix} = \begin{pmatrix} \boldsymbol{R}^{\top} & \boldsymbol{c} \\ \boldsymbol{0}^{\top} & 1 \end{pmatrix}$$



Note: $\boldsymbol{t} = -\boldsymbol{R}\boldsymbol{c}$
and $\boldsymbol{R}^{\top} = \boldsymbol{R}^{-1}$

Relationship between coordinates



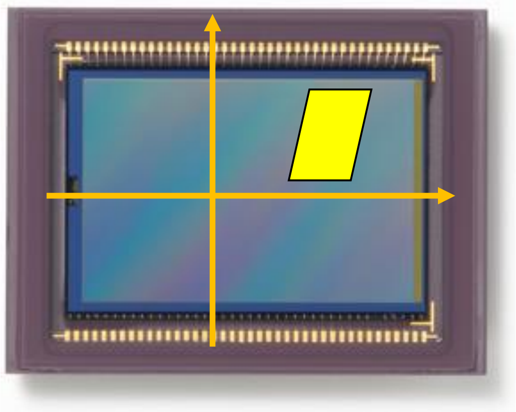
- Inverse transformation can be easily computed by inverse matrix.
- Transformation can be cascaded by simply multiplying matrixes.

Today's topics

- What is camera pose?
- Coordinate system and transformation
- **Projection from 3D to 2D**
  - **Intrinsic parameters**
  - **Extrinsic parameters**
- Camera pose estimation
  - Linear algorithm
  - Iterative algorithm

Intrinsic and Extrinsic parameters

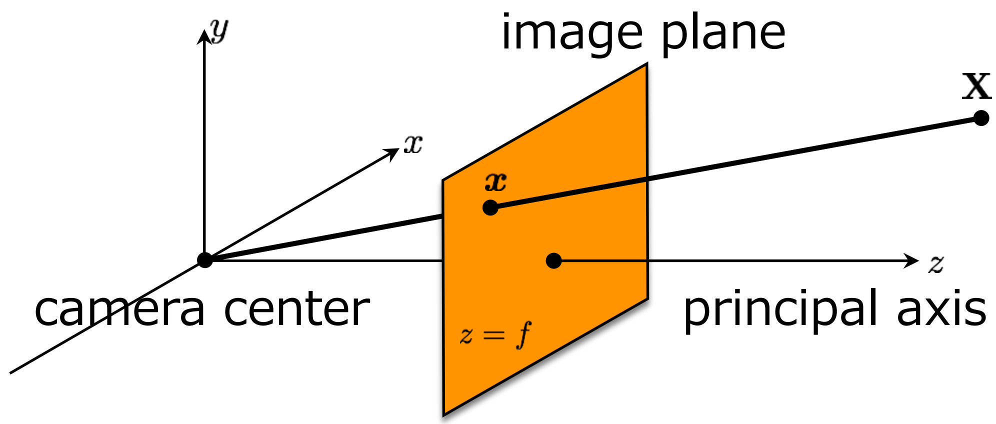
- Intrinsic parameters（内部パラメータ）
  - depend only on camera characteristics
  - 5 intrinsic parameters
    - focal length fx, fy in pixel unit
    - principal point cx, cy
    - skew s (usually s=0)



- Extrinsic parameters（外部パラメータ）
  - depend only on camera pose
  - rotation matrix $\boldsymbol{R}$, translation vector $\boldsymbol{t}$

Things to remember

The (rearranged) pinhole camera



3D to 2D mapping using projection matrix

$$\boldsymbol{x} = \boldsymbol{M}\boldsymbol{X}$$

$$\lambda \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} f_x & s & c_x & 0 \\ 0 & f_y & c_y & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} x_c \\ y_c \\ z_c \\ 1 \end{pmatrix}$$

Image coord. Projection matrix (Intrinsic parameter) Camera local coord.

Projection matrix decomposition

We can decompose the camera matrix like this:

$$\boldsymbol{M} = \begin{pmatrix} f_x & s & c_x & 0 \\ 0 & f_y & c_y & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} = \underbrace{\begin{pmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix}}_{\text{3x3 intrinsic}} \underbrace{\left(\begin{array}{ccc:c} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{array}\right)}_{\text{3x4 extrinsic}}$$

3x3 intrinsic 内部パラメータ

3x4 extrinsic 外部パラメータ

This means camera and world coordinates are the same.

What if world and camera coordinate systems are different?

$$\boldsymbol{M} = \begin{pmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{pmatrix} = \begin{pmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} [\boldsymbol{R} \mid \boldsymbol{t}]$$

3x3 intrinsic

3x4 extrinsic (camera pose)

Recap

- The camera matrix relates what two quantities?

$$\boldsymbol{x} = \boldsymbol{M}\boldsymbol{X}$$

3D points to 2D image points

- The camera matrix $\boldsymbol{M}$ can be decomposed into?

$$\boldsymbol{M} = \boldsymbol{K}[\boldsymbol{R}|\boldsymbol{t}]$$

intrinsic and extrinsic parameters

$$\boldsymbol{M} = \begin{pmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \left(\begin{array}{ccc:c} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{array}\right)$$

rotation translation

Today’s topics

- What is camera pose?
- Coordinate system and transformation
- Projection from 3D to 2D
  - Intrinsic parameters
  - Extrinsic parameters
- **Camera pose estimation**
  - **Linear algorithm**

Calibrating the Camera

Use an scene with **known** geometry

- Correspond image points to 3d points
- Get least squares solution (or non-linear solution)

Known 2d image coordinates

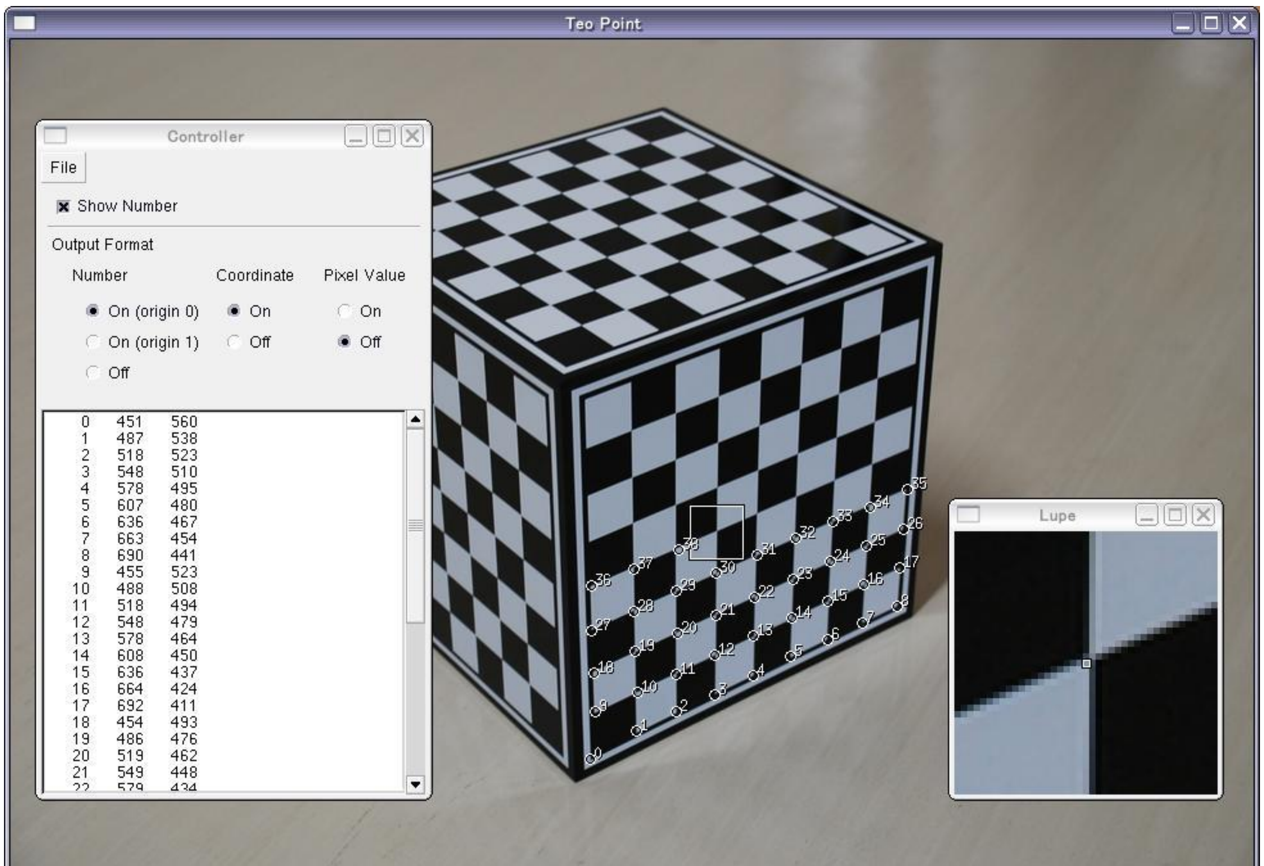
Known 3d world locations

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

Unknown Camera Parameters

Camera and image positions

- A calibration rig is observed by a camera and the image positions have been found manually or automatically.



Large scale calibration

- Calibration object
  - Marker for distant camera
  - Reflector for TotalStation

Algebraic manipulation

known 2D points

known 3D points

$$\begin{bmatrix} su \\ sv \\ s \end{bmatrix} = M \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} - & \boldsymbol{m}_1^\top & - \\ - & \boldsymbol{m}_2^\top & - \\ - & \boldsymbol{m}_3^\top & - \end{bmatrix} \begin{bmatrix} | \\ \boldsymbol{X} \\ | \end{bmatrix}$$

4 (width of the matrix) · 4 (height of $\boldsymbol{X}$)

estimate

- Expand

$$u = \frac{su}{s} = \frac{\boldsymbol{m}_1^\top \boldsymbol{X}}{\boldsymbol{m}_3^\top \boldsymbol{X}}$$

$$v = \frac{sv}{s} = \frac{\boldsymbol{m}_2^\top \boldsymbol{X}}{\boldsymbol{m}_3^\top \boldsymbol{X}}$$

- Manipulation to make it linear

$$\boldsymbol{m}_1^\top \boldsymbol{X} - \boldsymbol{m}_3^\top \boldsymbol{X} u = 0$$
$$\boldsymbol{m}_2^\top \boldsymbol{X} - \boldsymbol{m}_3^\top \boldsymbol{X} v = 0$$

Algebraic manipulation cont'd

$$\boldsymbol{m}_1^\top \boldsymbol{X} - \boldsymbol{m}_3^\top \boldsymbol{X} u = 0 \qquad \boldsymbol{m}_2^\top \boldsymbol{X} - \boldsymbol{m}_3^\top \boldsymbol{X} v = 0$$

- In matrix form

12

$$\begin{bmatrix} \boldsymbol{X}^\top & \boldsymbol{0} & -u\boldsymbol{X}^\top \\ \boldsymbol{0} & \boldsymbol{X}^\top & -v\boldsymbol{X}^\top \end{bmatrix} \begin{bmatrix} \boldsymbol{m}_1 \\ \boldsymbol{m}_2 \\ \boldsymbol{m}_3 \end{bmatrix} = 0$$

12

- For N points...

12

2N

$$\begin{bmatrix} \boldsymbol{X}_1^\top & \boldsymbol{0} & -u_1\boldsymbol{X}_1^\top \\ \boldsymbol{0} & \boldsymbol{X}_1^\top & -v_1\boldsymbol{X}_1^\top \\ \vdots & \vdots & \vdots \\ \boldsymbol{X}_N^\top & \boldsymbol{0} & -u_N\boldsymbol{X}_N^\top \\ \boldsymbol{0} & \boldsymbol{X}_N^\top & -v_N\boldsymbol{X}_N^\top \end{bmatrix} \begin{bmatrix} \boldsymbol{m}_1 \\ \boldsymbol{m}_2 \\ \boldsymbol{m}_3 \end{bmatrix} = 0$$

12

- Unknowns are 12, so more than 6 points are required to solve.

Known 2d image coordinates

Known 3d world locations

1st point

880	214
43	203
270	197
886	347
745	302
943	128
476	590
419	214
317	335
…	

(u_1, v_1)

312.747	309.140	30.086
305.796	311.649	30.356
307.694	312.358	30.418
310.149	307.186	29.298
311.937	310.105	29.216
311.202	307.572	30.682
307.106	306.876	28.660
309.317	312.490	30.230
307.435	310.151	29.318
…..		

(X_1, Y_1, Z_1)

Insert the 1st point values

$$\begin{bmatrix} 312.747 & 309.140 & 30.086 & 1 & 0 & 0 & 0 & 0 & -880\times312.747 & -880\times309.140 & -880\times30.086 & -880 \\ 0 & 0 & 0 & 0 & 312.747 & 309.140 & 30.086 & 1 & -214\times312.747 & -214\times309.140 & -214\times30.086 & -214 \\ & & & & & & \vdots & & & & & \\ X_n & Y_n & Z_n & 1 & 0 & 0 & 0 & 0 & -u_nX_n & -u_nY_n & -u_nZ_n & -u_n \\ 0 & 0 & 0 & 0 & X_n & Y_n & Z_n & 1 & -v_nX_n & -v_nY_n & -v_nZ_n & -v_n \end{bmatrix} \begin{bmatrix} m_{11} \\ m_{12} \\ m_{13} \\ m_{14} \\ m_{21} \\ m_{22} \\ m_{23} \\ m_{24} \\ m_{31} \\ m_{32} \\ m_{33} \\ m_{34} \end{bmatrix}$$

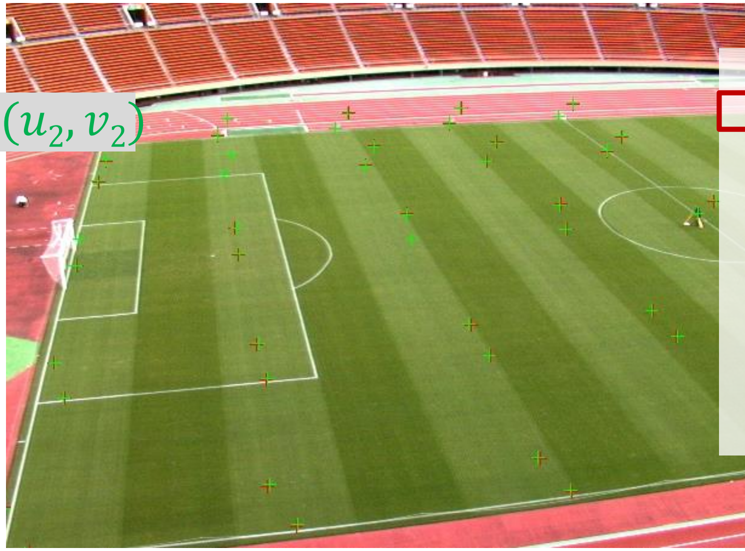
Known 2d image coordinates

Known 3d world locations

2nd point

880 214
43 203
270 197
886 347
745 302
943 128
476 590
419 214
317 335
…

(u_2, v_2)



(X_2, Y_2, Z_2)

312.747 309.140 30.086
305.796 311.649 30.356
307.694 312.358 30.418
310.149 307.186 29.298
311.937 310.105 29.216
311.202 307.572 30.682
307.106 306.876 28.660
309.317 312.490 30.230
307.435 310.151 29.318
…..

Insert the 2nd point values, and continue to the N-th point.

$$
\begin{bmatrix}
312.747 & 309.140 & 30.086 & 1 & 0 & 0 & 0 & 0 & -880\times 312.747 & -880\times 309.140 & -880\times 30.086 & -880 \\
0 & 0 & 0 & 0 & 312.747 & 309.140 & 30.086 & 1 & -214\times 312.747 & -214\times 309.140 & -214\times 30.086 & -214 \\
305.796 & 311.649 & 30.356 & 1 & 0 & 0 & 0 & 0 & -43\times 305.796 & -43\times 311.649 & -43\times 30.356 & -43 \\
0 & 0 & 0 & 0 & 305.796 & 311.649 & 30.356 & 1 & -203\times 305.796 & -203\times 311.649 & -43\times 30.356 & -203 \\
 & & & & & & \vdots & & & & & \\
X_n & Y_n & Z_n & 1_n & 0 & 0 & 0 & 0 & -u_n X_n & -u_n Y_n & -u_n Z_n & -u_n \\
0 & 0 & 0 & 0 & X_n & Y_n & Z_n & 1 & -v_n X_n & -v_n Y_n & -v_n Z_n & -v_n
\end{bmatrix}
\begin{bmatrix}
m_{11} \\ m_{12} \\ m_{13} \\ m_{14} \\ m_{21} \\ m_{22} \\ m_{23} \\ m_{24} \\ m_{31} \\ m_{32} \\ m_{33} \\ m_{34}
\end{bmatrix}
$$

Solve the equation by least squares

- For N points...

12 (columns) × 2N (rows)

$$\begin{bmatrix} \boldsymbol{X}_1^\top & \mathbf{0} & -u_1\boldsymbol{X}_1^\top \\ \mathbf{0} & \boldsymbol{X}_1^\top & -v_1\boldsymbol{X}_1^\top \\ \vdots & \vdots & \vdots \\ \boldsymbol{X}_N^\top & \mathbf{0} & -u_N\boldsymbol{X}_N^\top \\ \mathbf{0} & \boldsymbol{X}_N^\top & -v_N\boldsymbol{X}_N^\top \end{bmatrix} \begin{bmatrix} \boldsymbol{m}_1 \\ \boldsymbol{m}_2 \\ \boldsymbol{m}_3 \end{bmatrix} = 0$$

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

How to solve $\boldsymbol{A}\boldsymbol{x} = 0$?

Linear and Non-linear solutions

- Least squares problem

$$\boldsymbol{Ax} = 0 \text{ where } x \neq 0$$

A is a tall matrix. (縦長行列)

- Optimization form

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} \|\mathbf{A}\boldsymbol{x}\|^2 \text{ subject to } \|\boldsymbol{x}\|^2 = 1$$

- Closed-form solution
  - Singular value decomposition (SVD, 特異値分解)

    Solution **x** is the column of **V** corresponding to smallest singular value of $\mathbf{A} = \mathbf{U\Sigma V}^\top$

  - Eigen value decomposition (固有値分解)

    Equivalently, solution ***x*** is the Eigenvector corresponding to smallest Eigenvalue of $\mathbf{A}^\top\mathbf{A}$

Matrix decomposition

$$\boldsymbol{M} = \boldsymbol{K}[\boldsymbol{R}|\boldsymbol{t}]$$

- Can we factorize back to intrinsic and extrinsic parameters **K**, **R**, and **t**? → Yes!

$$\boldsymbol{M} = \boldsymbol{K}[\boldsymbol{R}|\boldsymbol{t}] = [\boldsymbol{KR}|-\boldsymbol{KRc}]$$

Note: $\boldsymbol{t} = -\boldsymbol{Rc}$, Refer Homogeneous coordinate slide

$$\begin{pmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{pmatrix}$$

$\boldsymbol{KR}$ (columns 1–3) $\qquad -\boldsymbol{KRc}$ (column 4)

$-\boldsymbol{KRc}$ → Multiply $-\boldsymbol{R}^{-1}\boldsymbol{K}^{-1}$ → $\boldsymbol{c}$ camera position

$\boldsymbol{KR}$ → QR decomposition → $\boldsymbol{K}, \boldsymbol{R}$

$$\begin{pmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \qquad \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix}$$

right upper triangle $\qquad$ orthogonal

Estimated parameters are too much?

Estimated parameters	5	12
Degrees of freedom	3	6

usually 0

$$\boldsymbol{M} = \begin{pmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{pmatrix}$$

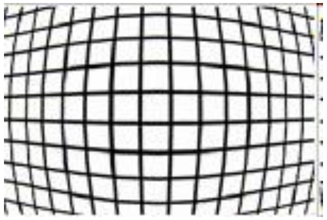
usually the same

Rotation DOF is 3!

- Rotation around x
- Rotation around y
- Rotation around z

Calibration with linear method

- Advantages
  - Easy to formulate and solve
  - Provides initialization for non-linear methods
- Disadvantages
  - Doesn’t directly give you camera parameters
  - Doesn’t model radial distortion
  - Can’t impose constraints, such as known focal length
- Non-linear methods are preferred
  - Define error as difference between projected points and measured points
  - Minimize error using Newton’s method or other non-linear optimization



radial distortion

Today’s topics revisit

- What is camera pose?
- Coordinate system and transformation
- Projection from 3D to 2D
  - Intrinsic parameters
  - Extrinsic parameters
- Camera pose estimation
  - Linear algorithm
  - Iterative algorithm

Today's mini-report 10

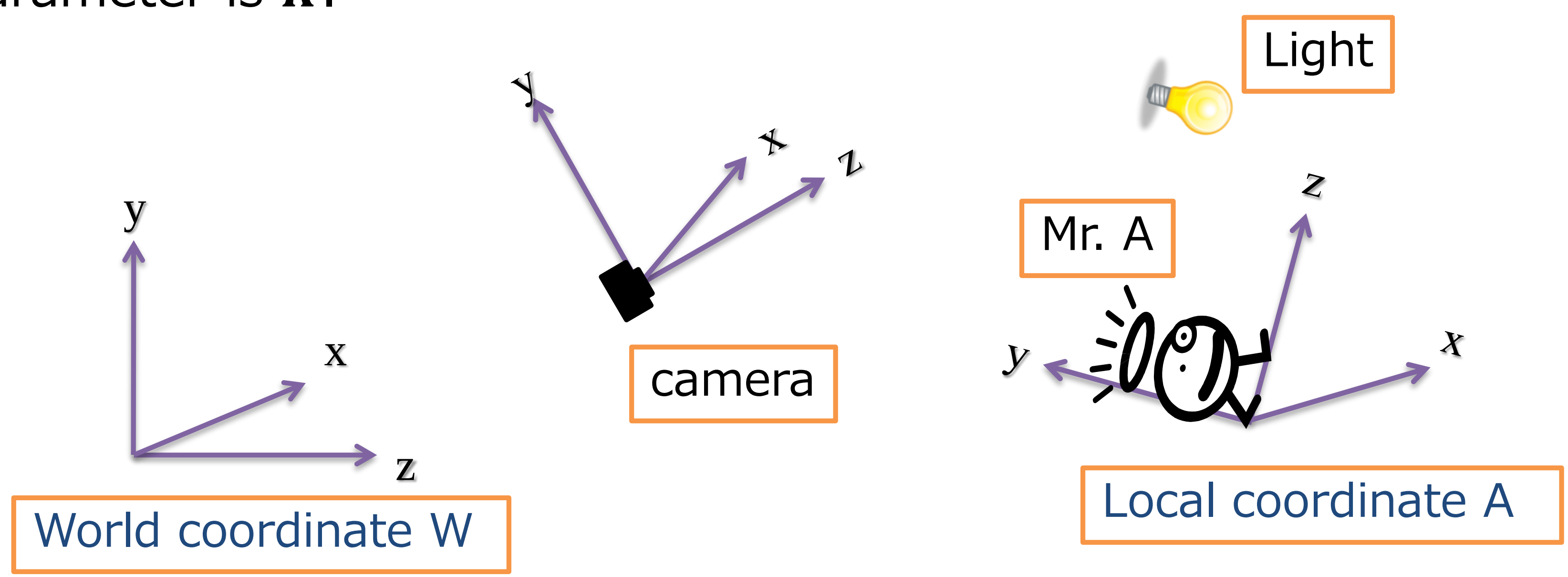
How to compute the position of light in the image coordinate of a camera?

Assumption:

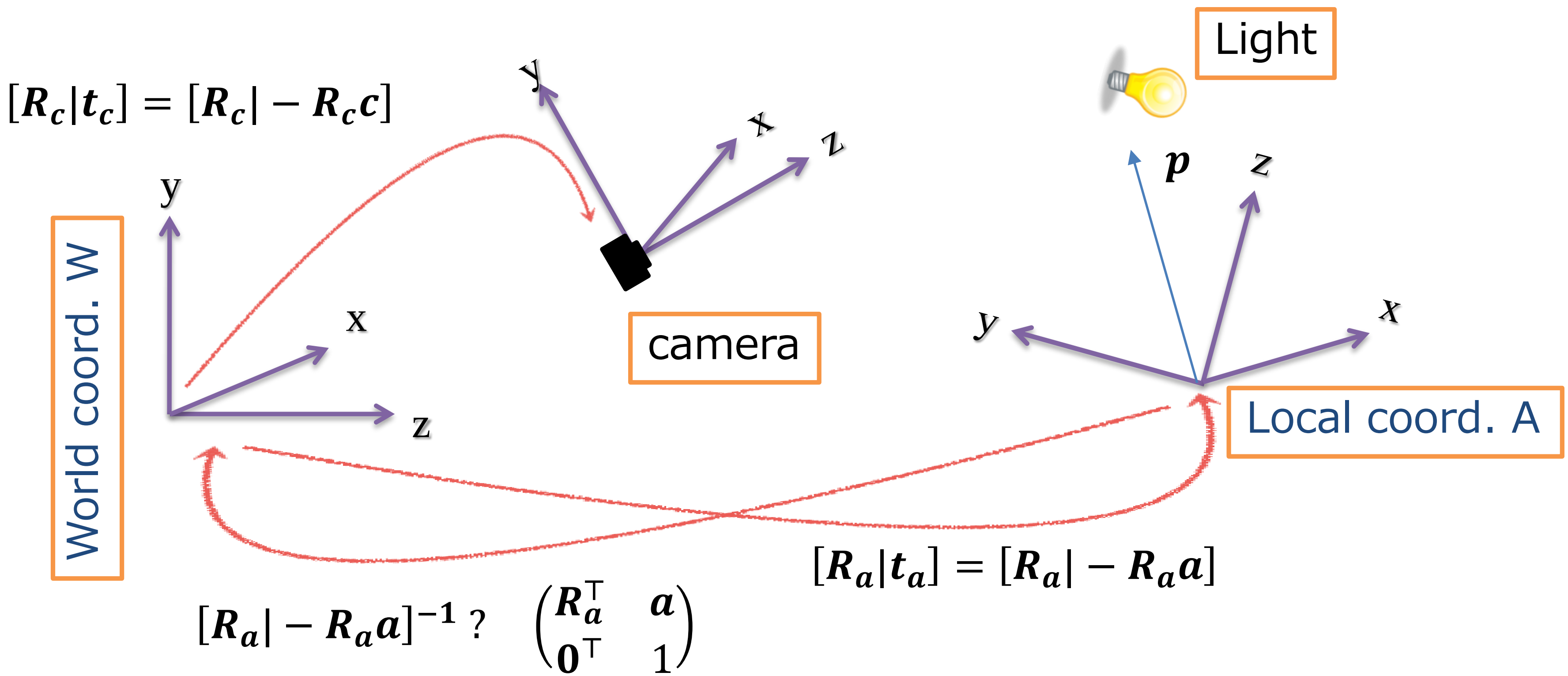
・ Light is placed at $\boldsymbol{p}$ in the coordinate system A.

・ Mr. A is staying at $\boldsymbol{a}$ in the world coordinate system, and is inclined from the world (rotation matrix $\boldsymbol{R_a}$).

・ A camera is put on $\boldsymbol{c}$, whose rotation matrix is $\boldsymbol{R_c}$, and intrinsic parameter is $\boldsymbol{K}$.



Today's mini-report 10



$$\lambda \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \boldsymbol{K} \quad [\boldsymbol{R}_c| - \boldsymbol{R}_c\boldsymbol{c}] \begin{pmatrix} \boldsymbol{R}_a^\top & \boldsymbol{a} \\ \boldsymbol{0}^\top & 1 \end{pmatrix} \boldsymbol{p}$$