

No. 2

カメラ校正

Camera Calibration

担当教員：向川康博・田中賢一郎

Slide credits

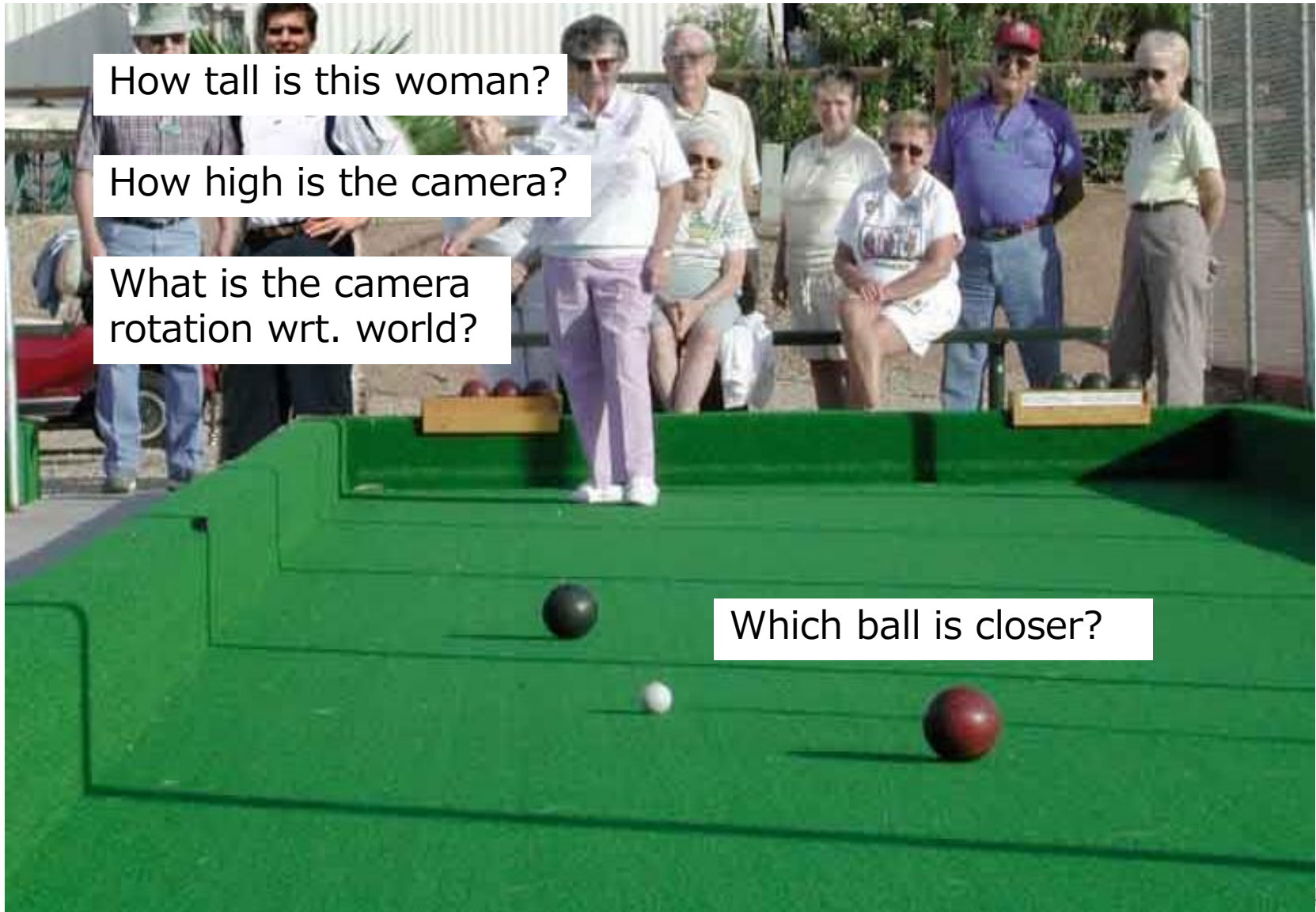
- Special Thanks: Some slides are adopted from other instructors' slides.
 - Tomokazu Sato, former NAIST CV1 class
 - James Tompkin, Brown CSCI 1430 Fall 2017
 - Ioannis Gkioulekas, CMU 16-385 Spring 2018

 - 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



How tall is this woman?

How high is the camera?

What is the camera rotation wrt. world?

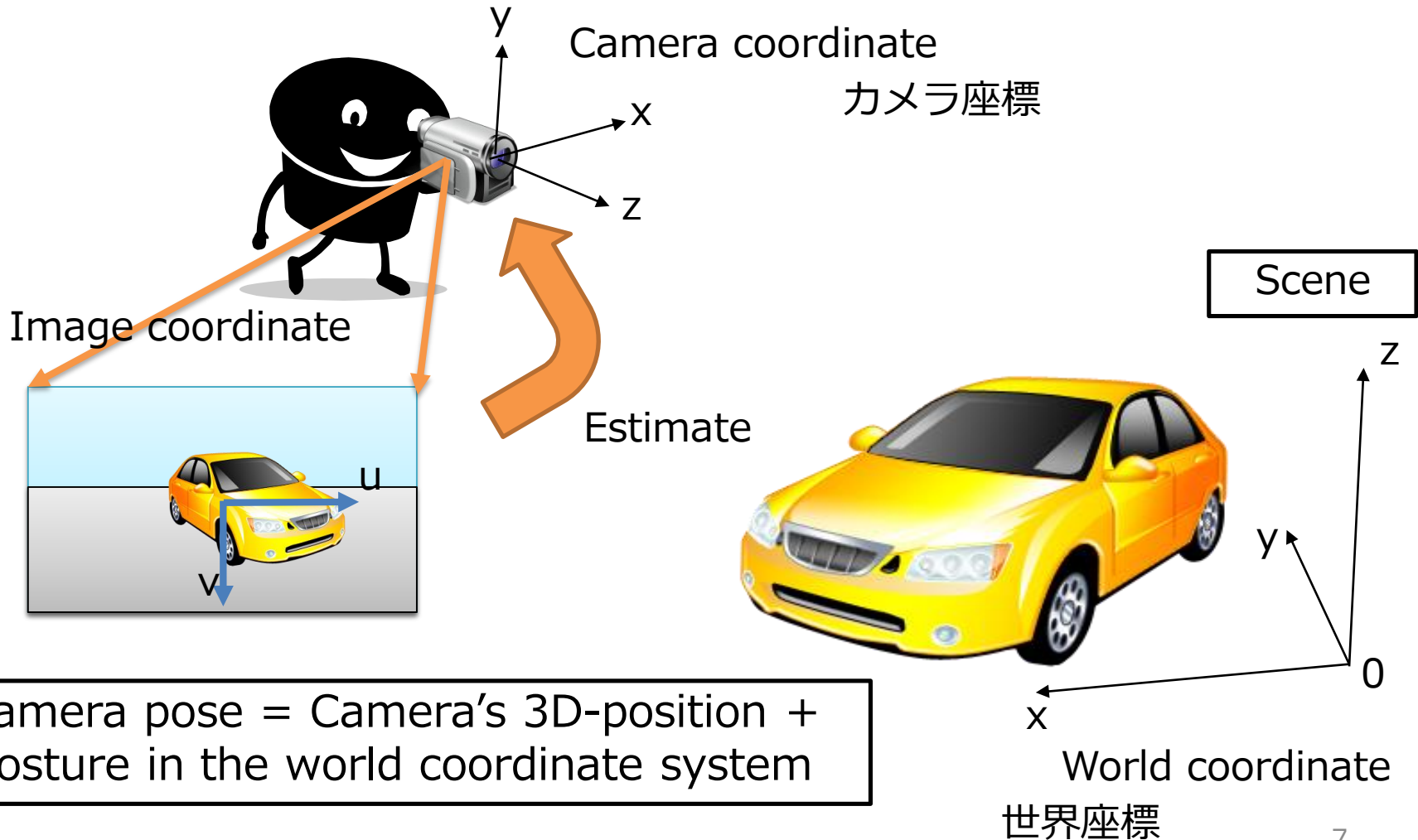
Which ball is closer?

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 estimation

(カメラ位置姿勢推定)

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

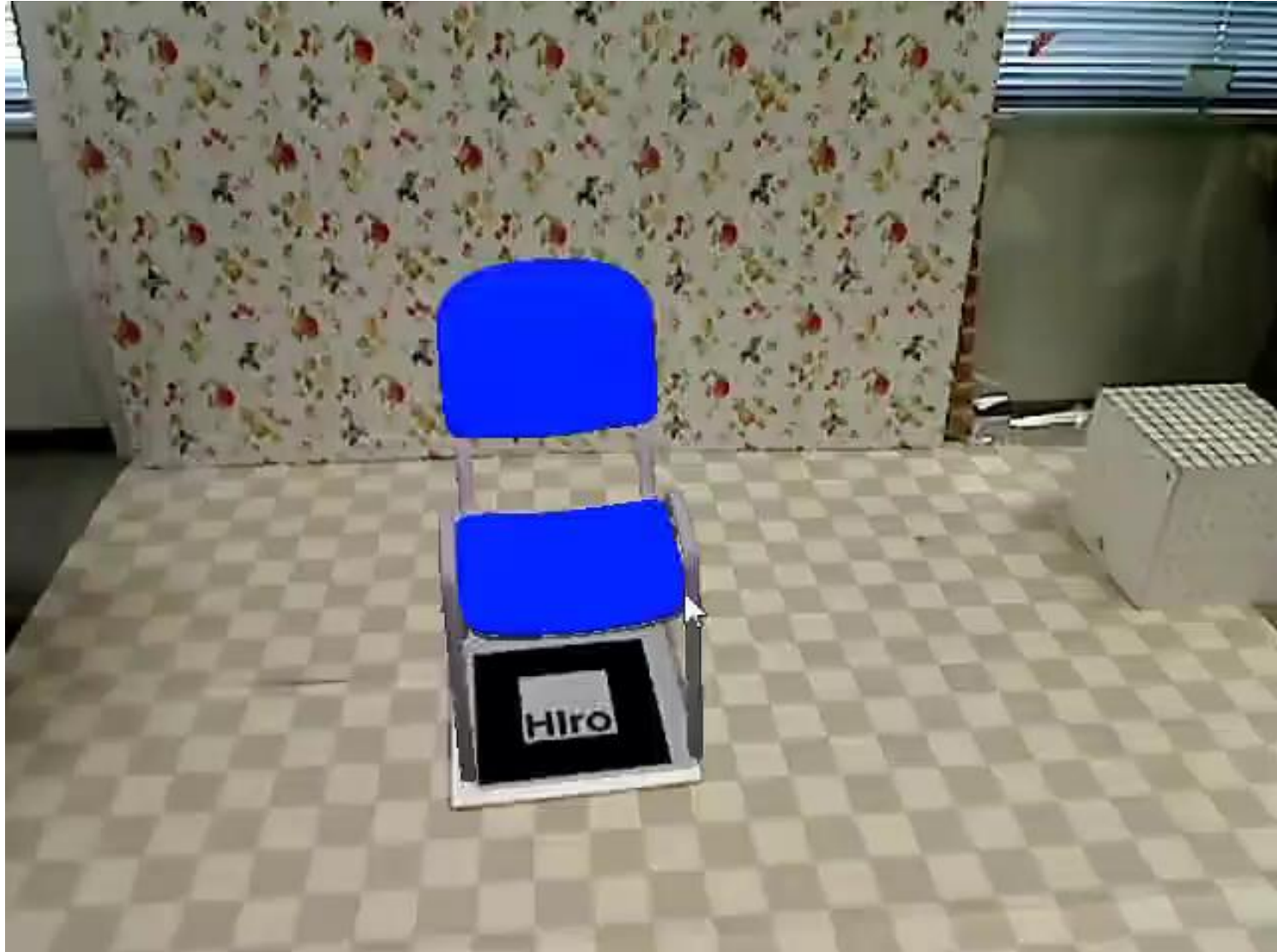


Pose estimation



Applications of camera pose estimation

- Marker based Augmented Reality -

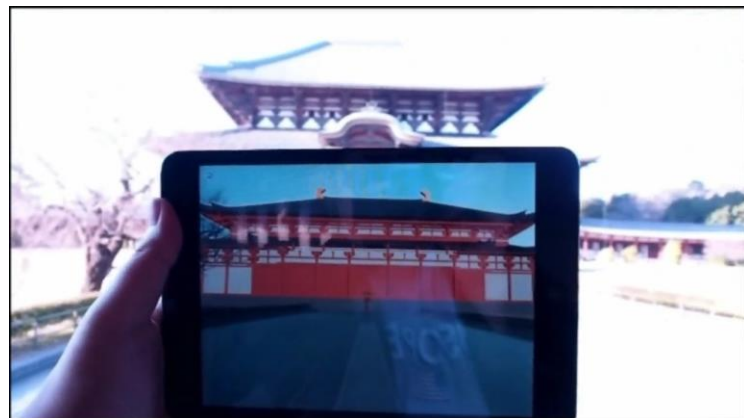


Applications of camera pose estimation

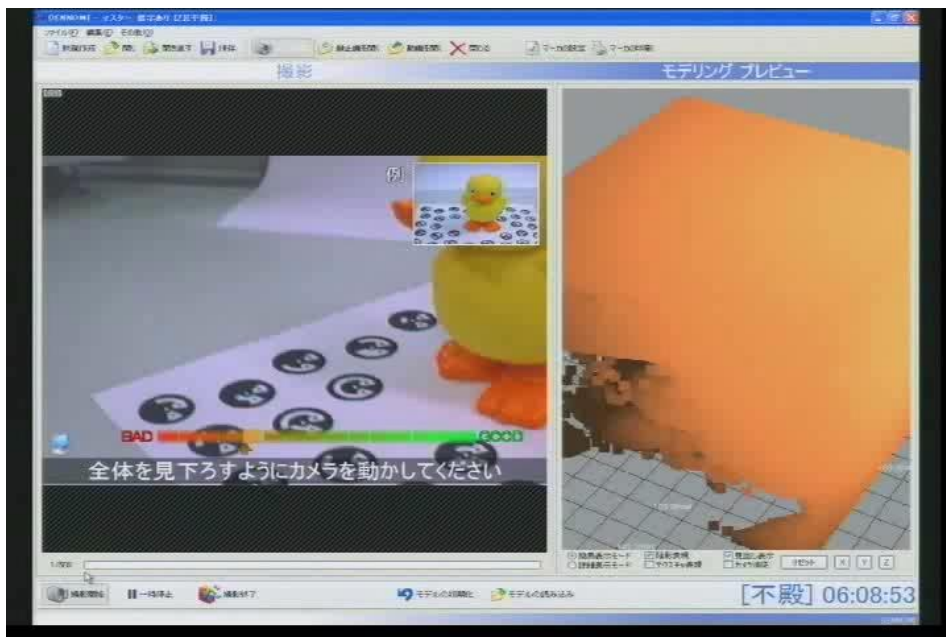
Image mosaic



Augmented reality



3D modeling



Free viewpoint image rendering

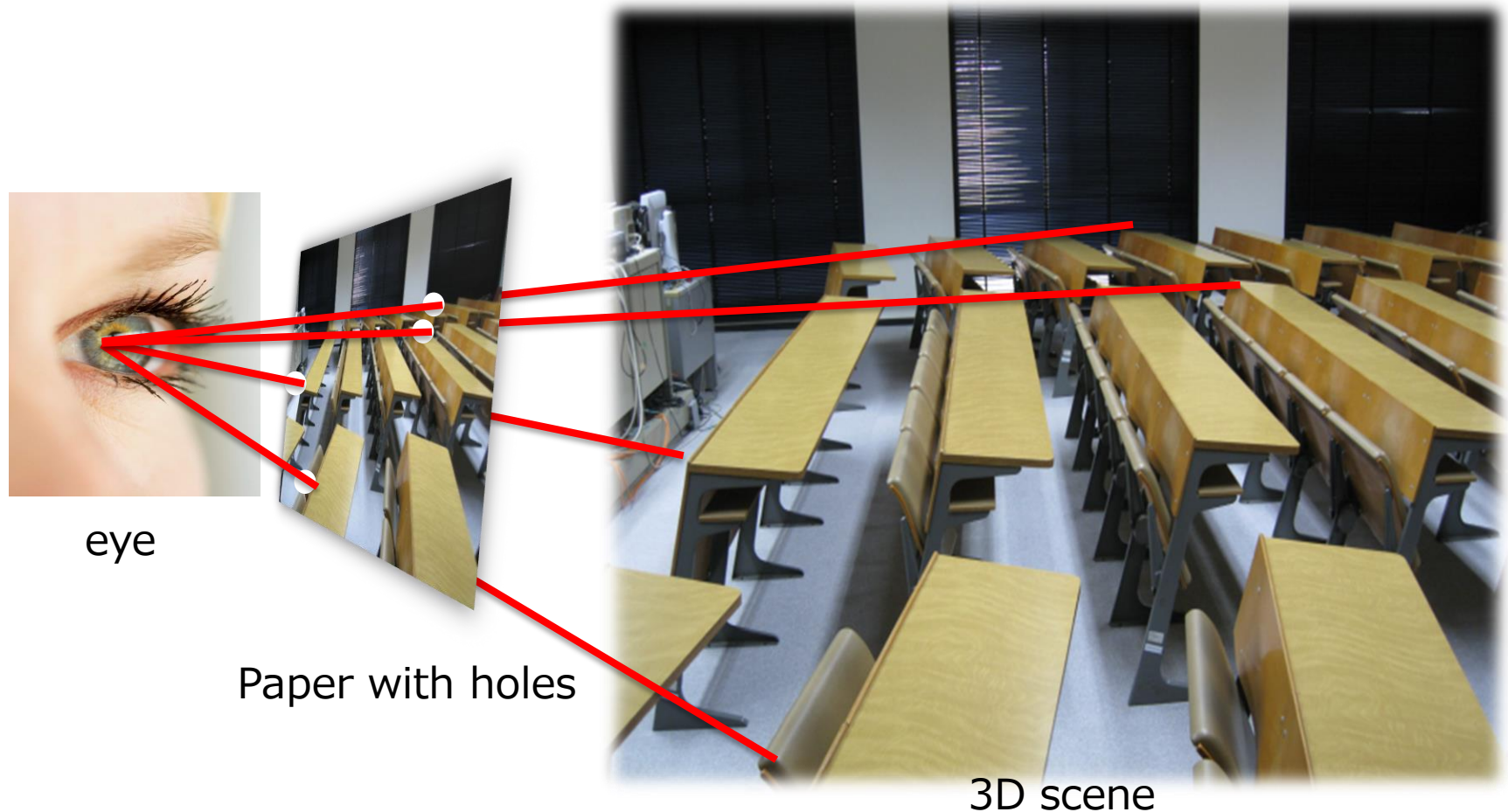


Small experiment: Find the camera pose for taking this picture



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

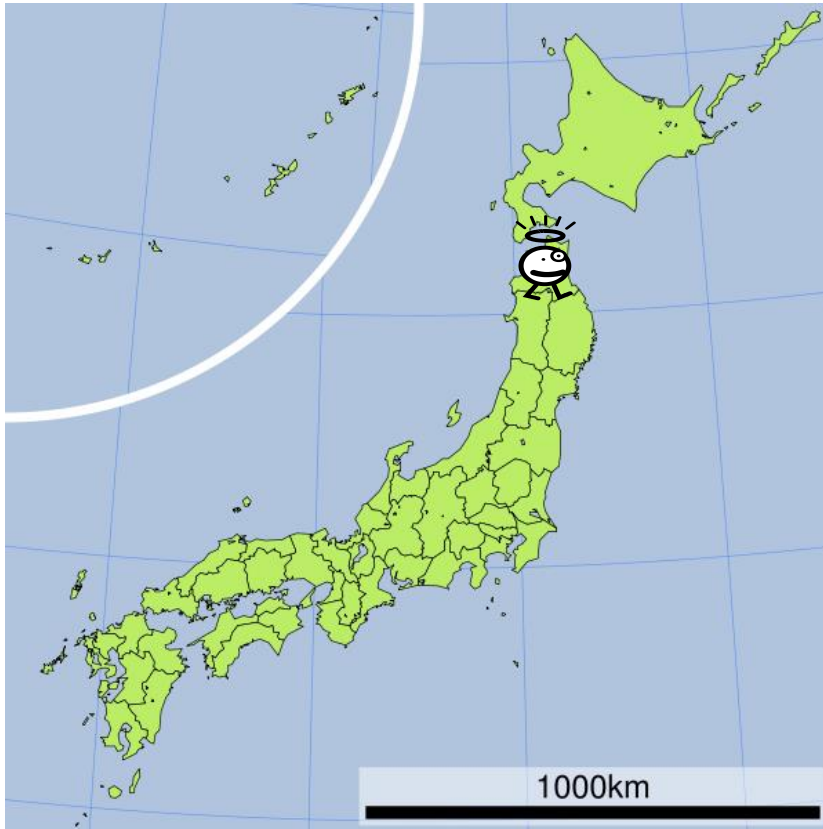
| | Structure (scene geometry) | Motion (camera geometry) | Measurements |
|------------------------|-------------------------------|-----------------------------|-----------------------------|
| Pose Estimation | known | estimate | 3D to 2D correspondences |
| Triangulation | estimate | known | 2D to 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 35° 40'xxx, longitude 139° 39'xxx.

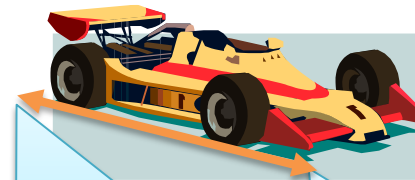
Posture:

- 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^{10} km



Latitude 135 40'xxx
Longitude 34 39'xxx

Latitude 135 40'xxx
Longitude 34 39'xxx

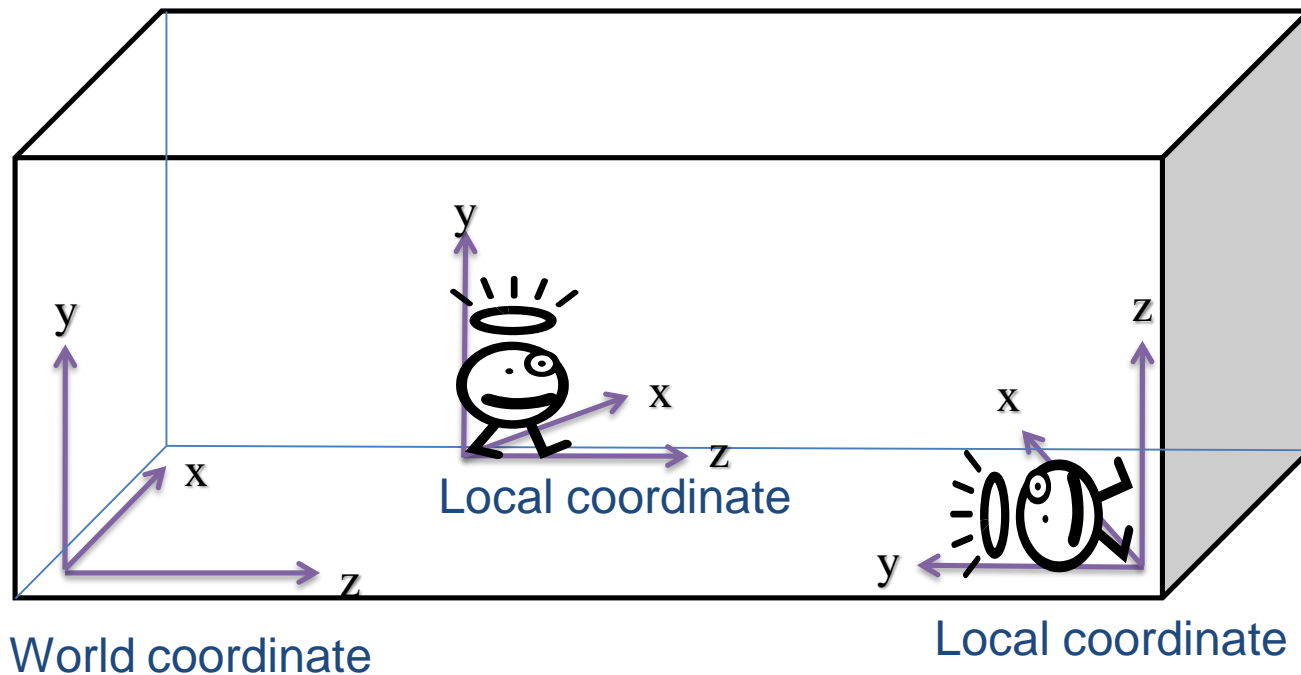


How long is this car?

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.

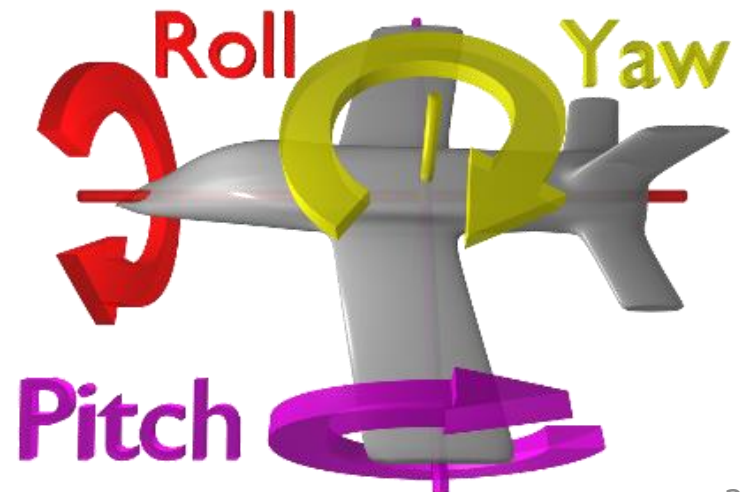


Rotation parameters

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

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

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



Homogeneous coordinate

(同次座標系 · 齊次座標系)

Heterogeneous (非同次)

$$X_c = R(X_w - C)$$

$$X' = R_2(R_1(X - C_1) - C_2)$$

Homogeneous (同次)

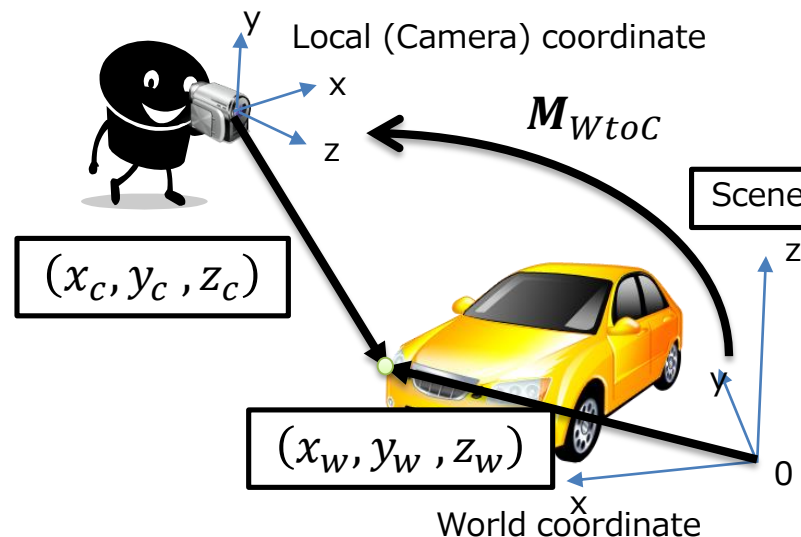
$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R} & -\mathbf{R}\mathbf{C} \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

Transformation matrix M (Extrinsic parameter)

$$X' = M_2 M_1 X$$

many notations

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



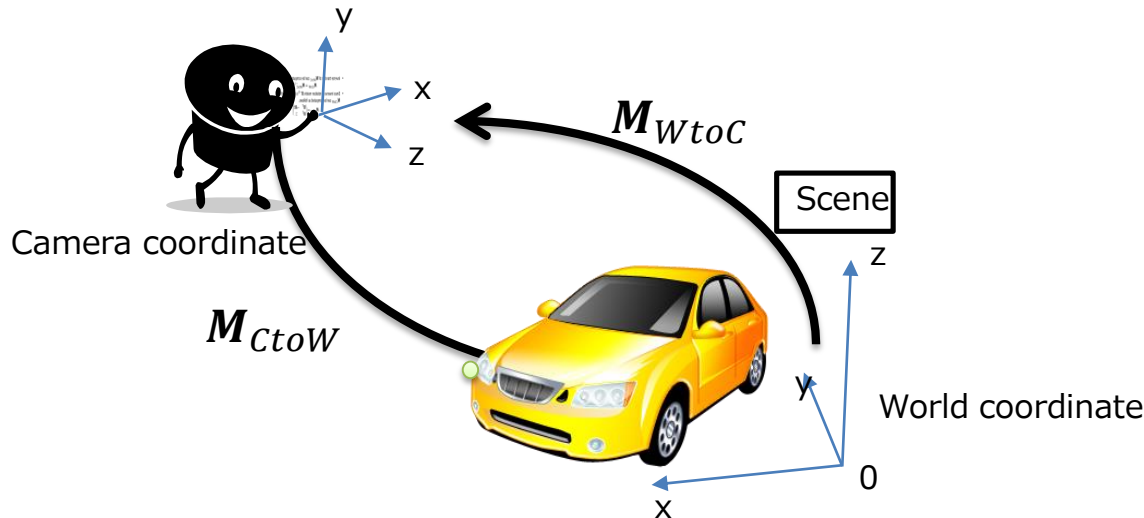
Important characteristics of rigid transform

- Inverse transform of M_{WtoC} can be computed by inverse matrix.

$$M_{CtoW} = M_{WtoC}^{-1}$$

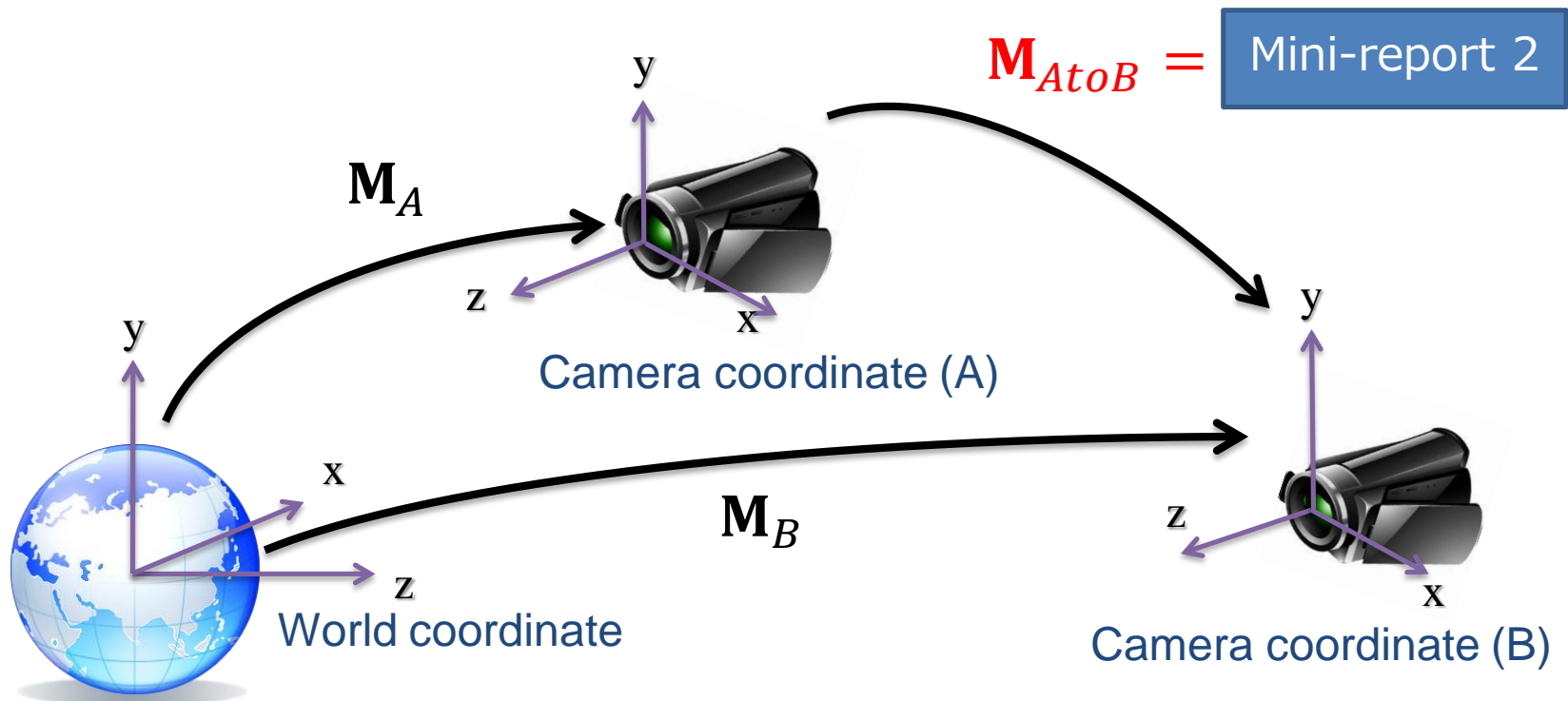
- Since inverse of rotation matrix R^{-1} is coincide with R^T , M_{CtoW} can be computed as follows.

$$M_{CtoW} = \begin{pmatrix} R^T & -R^T t \\ \mathbf{0}^T & 1 \end{pmatrix} = \begin{pmatrix} R^T & c \\ \mathbf{0}^T & 1 \end{pmatrix}$$



Note: $t = -Rc$
and $R^T = R^{-1}$

Relationship between coordinates



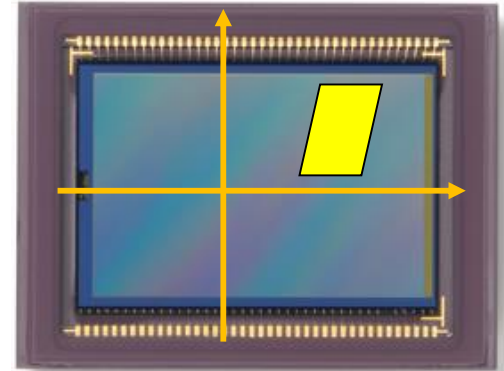
- Inverse transformation can be easily computed by inverse matrix.
- Transformation can be cascaded by simply multiplying matrixes.

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

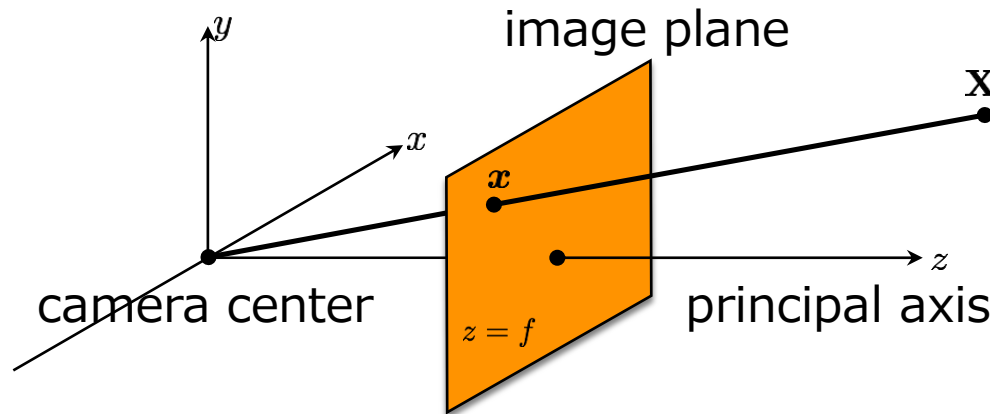
Intrinsic and Extrinsic parameters

- Intrinsic parameters (内部パラメータ)
 - depend only on camera characteristics
 - 5 intrinsic parameters
 - focal length f_x, f_y in pixel unit
 - principal point c_x, c_y
 - skew s (usually $s=0$)
- Extrinsic parameters (外部パラメータ)
 - depend only on camera pose
 - rotation matrix \mathbf{R} , translation vector \mathbf{t}



Things to remember

The (rearranged) pinhole camera



3D to 2D mapping using projection matrix

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

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

Image coord.

Projection matrix
(Intrinsic parameter)

Camera local coord.

Recap

- The camera matrix relates what two quantities?

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

Mini-report 3

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

$$\mathbf{M} =$$

Mini-report 4

Mini-report 5: What is \mathbf{K} , and what is $[\mathbf{R}|\mathbf{t}]$

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

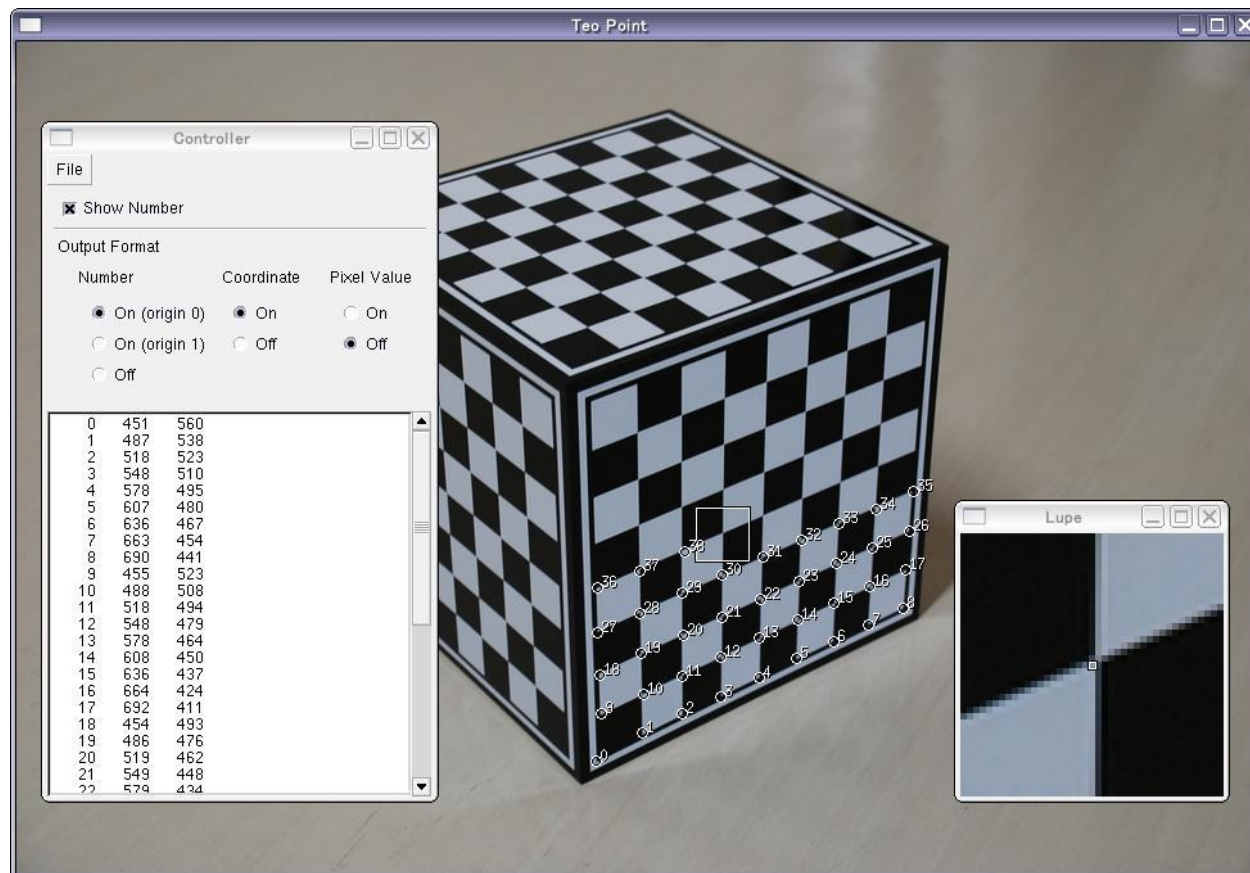
Known 3d world locations

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

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

$$\begin{bmatrix} s & u \\ s & v \\ s & \end{bmatrix} = M \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} - & \mathbf{m}_1^\top & - \\ - & \mathbf{m}_2^\top & - \\ - & \mathbf{m}_3^\top & - \end{bmatrix} \begin{bmatrix} | \\ \mathbf{X} \\ | \end{bmatrix}$$

known 3D points

estimate

- Mini-report 6: What is known and what is to estimate?

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

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

- Manipulation to make it linear

$$\mathbf{m}_1^\top \mathbf{X} - \mathbf{m}_3^\top \mathbf{X} u = 0$$

$$\mathbf{m}_2^\top \mathbf{X} - \mathbf{m}_3^\top \mathbf{X} v = 0$$

Algebraic manipulation cont'd

$$m_1^T X - m_3^T X u = 0$$

$$m_2^T X - m_3^T X v = 0$$

- In matrix form

$$\begin{matrix} & \xrightarrow{12} & \\ \begin{bmatrix} X^T & \mathbf{0} & -uX^T \\ \mathbf{0} & X^T & -vX^T \end{bmatrix} & \begin{bmatrix} m_1 \\ m_2 \\ m_3 \end{bmatrix} & \begin{matrix} \updownarrow 12 \\ = 0 \end{matrix} \end{matrix}$$

- For N points...

$$\begin{matrix} & \xrightarrow{12} & \\ \begin{matrix} \updownarrow 2N \\ \begin{bmatrix} X_1^T & \mathbf{0} & -u_1 X_1^T \\ \mathbf{0} & X_1^T & -v_1 X_1^T \\ \vdots & \vdots & \vdots \\ X_N^T & \mathbf{0} & -u_N X_N^T \\ \mathbf{0} & X_N^T & -v_N X_N^T \end{bmatrix} \end{matrix} & \begin{bmatrix} m_1 \\ m_2 \\ m_3 \end{bmatrix} & \begin{matrix} \updownarrow 12 \\ = 0 \end{matrix} \end{matrix}$$

- 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 |
| ... | |



| | | |
|---------|---------|--------|
| 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

| | | | | | | | | | | | |
|---------|---------|--------|---|---------|---------|--------|----------|-----------------------|-----------------------|----------------------|--------|
| 312.747 | 309.140 | 30.086 | 1 | 0 | 0 | 0 | 0 | -880×312.747 | -880×309.140 | -880×30.086 | -880 |
| 0 | 0 | 0 | 0 | 312.747 | 309.140 | 30.086 | 1 | -214×312.747 | -214×309.140 | -214×30.086 | -214 |
| | | | | | | | \vdots | | | | |
| X_n | Y_n | Z_n | 1 | 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 | $-v_n X_n$ | $-v_n Y_n$ | $-v_n Z_n$ | $-v_n$ |

- m_{11}
- m_{12}
- m_{13}
- m_{14}
- m_{21}
- m_{22}
- m_{23}
- m_{24}
- m_{31}
- m_{32}
- m_{33}
- m_{34}

Solve the equation by least squares

- For N points...
$$\begin{matrix} & \xrightarrow{12} \\ \downarrow 2N & \begin{bmatrix} X_1^T & 0 & -u_1 X_1^T \\ 0 & X_1^T & -v_1 X_1^T \\ \vdots & \vdots & \vdots \\ X_N^T & 0 & -u_N X_N^T \\ 0 & X_N^T & -v_N X_N^T \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ m_3 \end{bmatrix} = 0 \end{matrix}$$

Mini-report 7: what is the size of the matrix?

How to solve $Ax = 0$?

Linear and Non-linear solutions

- Least squares problem

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

A is a tall matrix. (縱長行列)

- Optimization form

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

- Closed-form solution

- Singular value decomposition (SVD, 特異值分解)

Solution \mathbf{x} is the column of \mathbf{V}
corresponding to smallest singular value of

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\top}$$

- Eigen value decomposition (固有値分解)

Equivalently, solution \mathbf{x} is the Eigenvector
corresponding to smallest Eigenvalue of

$$\mathbf{A}^{\top} \mathbf{A}$$

Matrix decomposition

$$M = K[R|t]$$

- 8: what is the next step?
9: what is M, K, R, and t?

 parameters **K**, **R**, and **t**? → Yes!

$$M = K[R|t] = [KR|-KRc]$$

Note: $t = -Rc$,
Refer Homogeneous coordinate slide

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



Multiply $-R^{-1}K^{-1}$

c
camera position

$$KR \quad -KRc$$



QR decomposition

$$K, R$$

$$\begin{pmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix}$$

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

right upper triangle

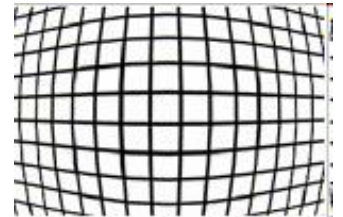
orthogonal

Estimated parameters are too much?

| | | |
|----------------------|--|--|
| Estimated parameters | 5 | 12 |
| | $\mathbf{M} = \begin{pmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix}$ <p>usually the same</p> | $\begin{pmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{pmatrix}$ <p>Rotation DOF is 3! - Rotation around x - Rotation around y - Rotation around z</p> |
| Degrees of freedom | 3 | 6 |

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 p in the coordinate system A.
- Mr. A is staying at a in the world coordinate system, and is inclined from the world (rotation matrix R_a).
- A camera is put on c , whose rotation matrix is R_c , and intrinsic parameter is K .

