University at Buffalo
Department of Computer Science and and Engineering
CSE 473/573 - Computer Vision and Image Processing

Fall 2021

Project #1

Due Date: 10/1/21, 11:59PM

1 Rotation Matrix (5 points)

Figure 1 illustrates the transformation from coordinate xyz to XYZ: 1)rotate around z axis with $\alpha = 45°$ to get $x'y'z'$ axis; 2) rotate around x' axis with $\beta = 30°$ to get $x''y''z''$; 3) rotate around z'' axis with $\gamma = 60°$ to get XYZ.

- Design a program to get the rotation matrix from xyz to XYZ.
- Design a program to get the rotation matrix from XYZ to xyz.

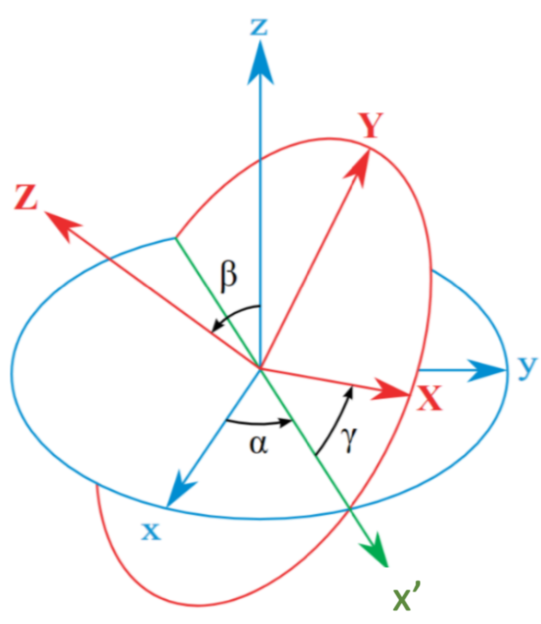

Figure 1: Illustration of Eular angles.

2 Camera Calibration (5 points)

Preliminary.1.

The projection from world coordinate to image plane can be indicated by intrinsic parameters (Camera) and extrinsic parameters(World). From world coordinate to camera coordinate, the extrinsic parameters can be used as

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} + \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix}$$

$$\begin{aligned} M = M_{in} \cdot M_{ex} &= \begin{bmatrix} f_x & 0 & o_x \\ 0 & f_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} r_{11} & r_{12} & r_{13} & T_x \\ r_{21} & r_{22} & r_{23} & T_y \\ r_{31} & r_{32} & r_{33} & T_z \end{bmatrix} \\ \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix} &= \begin{bmatrix} f_x r_{11} + o_x r_{31} & f_x r_{12} + o_x r_{32} & f_x r_{13} + o_x r_{33} & f_x T_x + o_x T_z \\ f_y r_{21} + o_y r_{31} & f_y r_{22} + o_y r_{32} & f_y r_{23} + o_y r_{33} & f_y T_y + o_y T_z \\ r_{31} & r_{32} & r_{33} & T_z \end{bmatrix}. \end{aligned} \tag{1}$$

Let's define $\mathbf{m_1} = (m_{11}, m_{12}, m_{13})^T, \mathbf{m_2} = (m_{21}, m_{22}, m_{23})^T, \mathbf{m_3} = (m_{31}, m_{32}, m_{33})^T, \mathbf{m_4} = (m_{14}, m_{24}, m_{34})^T$. Also define $\mathbf{r}_1 = (r_{11}, r_{12}, r_{13})^T, \mathbf{r}_2 = (r_{21}, r_{22}, r_{23})^T, \mathbf{r}_3 = (r_{31}, r_{32}, r_{33})^T$. Observe that $(\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3)$ is the rotation matrix, then

$$(\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3) \begin{pmatrix} \mathbf{r}_1^T \\ \mathbf{r}_2^T \\ \mathbf{r}_3^T \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Then we have $\mathbf{r}_i^T \mathbf{r}_i = 1, \quad \mathbf{r}_i^T \mathbf{r}_j = 0 \ (i \neq j)$.

From M we have

$$\begin{aligned} \mathbf{m}_1^T \mathbf{m}_3 &= r_{31}(f_x r_{11} + o_x r_{31}) + r_{32}(f_x r_{12} + o_x r_{32}) + r_{33}(f_x r_{13} + o_x r_{33}) \\ &= f_x(r_{11} r_{31} + r_{12} r_{32} + r_{13} r_{33}) + o_x(r_{31}^2 + r_{32}^2 + r_{33}^2) \\ &= f_x(\mathbf{r}_1^T \mathbf{r}_3) + o_x(\mathbf{r}_3^T \mathbf{r}_3) \\ &= o_x \end{aligned} \tag{2}$$

Similarly, Next, from M we have

$$\begin{aligned} \mathbf{m}_1^T \mathbf{m}_1 &= (f_x r_{11} + o_x r_{31})^2 + (f_x r_{12} + o_x r_{32})^2 + (f_x r_{13} + o_x r_{33})^2 \\ &= f_x^2 \cdot \mathbf{r_1}^T \mathbf{r_1} + 2 f_x o_x \cdot \mathbf{r_1}^T \mathbf{r_3} + o_x^2 \cdot \mathbf{r_3}^T \mathbf{r_3} = f_x^2 + o_x^2 \end{aligned} \tag{3}$$

So $f_x = \sqrt{\mathbf{m}_1^T \mathbf{m}_1 - o_x^2}$. Similarly we have $o_y = \mathbf{m}_2^T \mathbf{m}_3$, $f_y = \sqrt{\mathbf{m}_2^T \mathbf{m}_2 - o_y^2}$. Overall, we come to the conclusion as follows

$$o_x = \mathbf{m}_1^T \mathbf{m}_3 \quad o_y = \mathbf{m}_2^T \mathbf{m}_3 \tag{4}$$

$$f_x = \sqrt{\mathbf{m}_1^T \mathbf{m}_1 - o_x^2} \quad f_y = \sqrt{\mathbf{m}_2^T \mathbf{m}_2 - o_y^2} \tag{5}$$

Preliminary.2.
Let $X_w Y_w Z_w$ be the world coordinate and xy be the image coordinate, we have the transformation matrix $M \in \mathbb{R}^{3\times 4}$:

$$s \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix} \cdot \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \tag{6}$$

$$\begin{aligned} sx &= m_{11} X_w + m_{12} Y_w + m_{13} Z_w + m_{14}, \\ sy &= m_{21} X_w + m_{22} Y_w + m_{23} Z_w + m_{24}, \\ s &= m_{31} X_w + m_{32} Y_w + m_{33} Z_w + m_{34}. \end{aligned} \tag{7}$$

We can solve m_{ij} with the equation below:

$$\begin{bmatrix} X_w^1 & Y_w^1 & Z_w^1 & 1 & 0 & 0 & 0 & 0 & -x^1X_w^1 & -x^1Y_w^1 & -x^1Z_w^1 & -x^1 \\ 0 & 0 & 0 & 0 & X_w^1 & Y_w^1 & Z_w^1 & 1 & -y^1X_w^1 & -y^1Y_w^1 & -y^1Z_w^1 & -y^1 \\ & & & & & & . & & & & & \\ & & & & & & . & & & & & \\ & & & & & & . & & & & & \\ & & & & & & . & & & & & \\ & & & & & & . & & & & & \\ & & & & & & . & & & & & \\ X_w^n & Y_w^n & Z_w^n & 1 & 0 & 0 & 0 & 0 & -x^nX_w^n & -x^nY_w^n & -x^nZ_w^n & -x^n \\ 0 & 0 & 0 & 0 & X_w^n & Y_w^n & Z_w^n & 1 & -y^nX_w^n & -y^nY_w^n & -y^nZ_w^n & -y^n \end{bmatrix} \cdot \begin{bmatrix} m_{11} \\ m_{12} \\ m_{13} \\ m_{14} \\ m_{21} \\ m_{22} \\ m_{23} \\ m_{24} \\ m_{31} \\ m_{32} \\ m_{33} \\ m_{34} \end{bmatrix} = \mathbf{0}, \tag{8}$$

where the first matrix is with size $2n \times 12$ (n is the number of available points).

Preliminary.3.

Solve the homogeneous linear equation $\mathbf{Ax} = 0$, where $\mathbf{x}$ is the vector of N unknowns, and $\mathbf{A}$ is the matrix of $M \times N$ coefficients. A quick observation is that there are infinite solutions for $\mathbf{Ax} = 0$, since we can randomly scale x with a scalar λ such that $\mathbf{A}(\lambda\mathbf{x}) = 0$. Therefore, we assume $\|\mathbf{x}\| = 1$. Solving the equation can be converted to

$$\min \|\mathbf{Ax}\| \tag{9}$$

The minimization problem can be solved with Singular Value Decomposition (SVD). Assume that $\mathbf{A}$ can be decomposed to $\mathbf{U\Sigma V}^T$, we have

$$\min \|\mathbf{Ax}\| = \|\mathbf{U\Sigma V}^T\mathbf{x}\| = \|\mathbf{\Sigma V}^T\mathbf{x}\|. \tag{10}$$

Note that $\|\mathbf{V}^T\mathbf{x}\| = \|\mathbf{x}\| = 1$, then let $\mathbf{y} = \mathbf{V}^T\mathbf{x}$, so we have

$$\begin{aligned} \min \|\mathbf{Ax}\| &= \|\mathbf{\Sigma y}\| \\ &= \left\| \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_n \\ 0 & \cdots & 0 \\ \vdots & \vdots & \vdots \\ 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} \right\|, \end{aligned} \tag{11}$$

where $\sigma_1 \geq \cdots \geq \sigma_n \geq 0$. Recall that $\|\mathbf{y}\| = 1$, we can set

$$\begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 1 \end{bmatrix}. \tag{12}$$

So $\mathbf{x}$ should be the last row of $\mathbf{V}^T$.

Question

Figure 2 shows an image of the checkerboard, where XYZ is the world coordinate and xy is marked as the image coordinate. The edge length of each grid on the checkerboard is $10mm$ in reality. Suppose one pixel of the image is equivalent to $1mm$. You can calculate the projection matrix from world coordinate to image coordinate based on the 32 marked points on the checkerboard. From the projection matrix you can get the intrinsic matrix which is indicated as $\begin{bmatrix} f_x & 0 & o_x \\ 0 & f_y & o_y \\ 0 & 0 & 1 \end{bmatrix}$ (f_x and f_y are not necessarily be the same).

- Design a program to obtain the intrinsic parameters f_x, f_y, o_x, o_y.
- If the original point of world coordinate changed, would the intrinsic parameters be the same?

In this task you are only allowed to use the library and library function already imported in the script "task2.py".

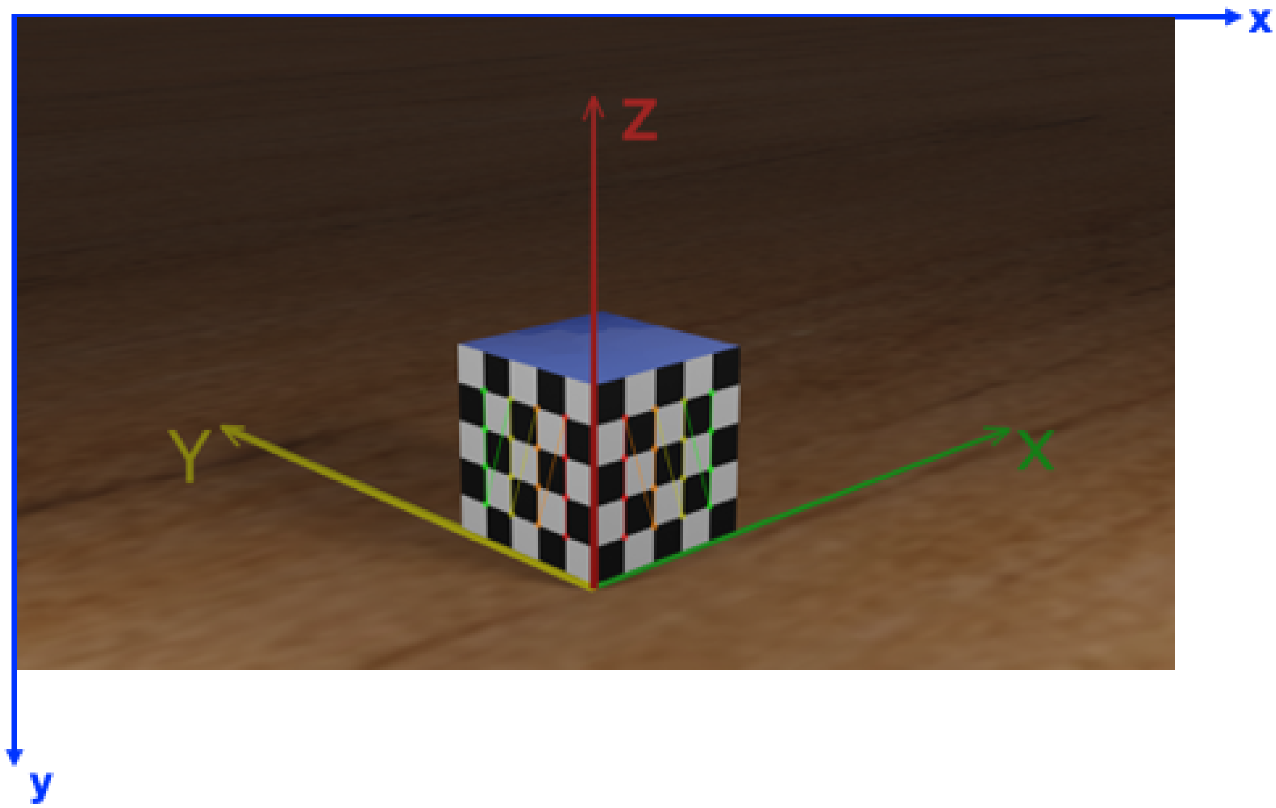

Figure 2: Image of the checkerboard

Instructions:

- Compress the two python files, i.e., "task1.py", "task2.py", the given image ("checkerboard.png") into a zip file, name it as "UBID.zip" (replace "UBID" with your eight-digit UBID, e.g., 51399256) and upload it to UBLearns before the due. The zip file you upload should not contain files other than the three aforementioned files.
- Anyone whose code is evaluated as plagiarism, your grade will be 0 for this project.
- For all students whose code raise "RuntimeError", your grade will be 0 for this task.
- Strictly follow the format in the scripts, i.e., "task1.py", "task2.py". **Do Not** modify the code provided to you.
- **Do Not** import any library or APIs besides what has been listed. For task2, you are **ONLY** allowed to use the library and library function already imported in the script.
- Late submissions within one day is allowed and will result in a 50% penalty. One day is defined as 24 hours after the day/time the assignment is due (excluding weekends or school holidays). After that, submissions will not be accepted.