# Assignment for Lecture 10

LARGE-SCALE MATRIX COMPUTATIONS AND 3D RECONSTRUCTION

Lecture Date: 5/6/2026

"C" denotes for "computational" problems, language suggestion: Python/Julia

please include codes and results with analyses for computational problems

***please write in pdf format and submit to*** `bjcai@fudan.edu.cn` ***before the lecture of 5/13/2026***

1. Write down the rotation matrix in 4D.

2. Given a few noisy points, i.e., $(x^{(i)}, y^{(i)})$, how can one fit out an ellipse? Figure out the connection between this problem and the calculation of the fundamental matrix.

3. [C] Implement a programming simulation for the following multi-view geometry setup. There are in total $m = 10^4$ 3D points uniformly randomly distributed with depth between 10 cm and 500 cm. The maximum noise level of each 3D point is characterized by the parameter $n_{\max}$ in the unit of pixel, and the final result is obtained by averaging over 10 independent runs. The parameters are

$$\mathbf{K} = \begin{pmatrix} 750\,\text{pixel} & 0 & 250\,\text{pixel} \\ 0 & 500\,\text{pixel} & 250\,\text{pixel} \\ 0 & 0 & 1 \end{pmatrix}, \quad \mathbf{R}_j = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_j & -\sin\theta_j \\ 0 & \sin\theta_j & \cos\theta_j \end{pmatrix}, \quad \mathbf{t}_j = \begin{pmatrix} x_j \\ 0 \\ 0 \end{pmatrix}, \tag{1-1}$$

together with $\mathbf{R}_1 = \vec{1}, \mathbf{t} = \vec{0}$, and $\theta_2 = \pi/6, x_2 = 10\,\text{cm}, \theta_3 = \pi/6, x_3 = -8\,\text{cm}, \theta_4 = -\pi/6, x_4 = 15\,\text{cm}$, and similarly $\theta_5 = -\pi/6, x_5 = -12\,\text{cm}, \theta_6 = \pi/6, x_6 = -18\,\text{cm}$. Write a program to: (1) generate the 3D points according to the above specification; (2) project them onto each camera using the given $\mathbf{K}$, $\mathbf{R}_j$, and $\mathbf{t}_j$; (3) add random image noise to each projected point with maximum magnitude controlled by $n_{\max}$ (in pixel units); (4) repeat the experiment for different values of $n_{\max}$; (5) for each $n_{\max}$, perform 10 independent runs and compute the averaged error (e.g., reprojection error or deviation from ground truth); (6) visualize the final result by plotting the averaged error as a function of $n_{\max}$ and briefly analyze the trend.