Assignment 2: Where is The Camera?

2022 Fall EECS205002 Linear Algebra

Due: 2022/11/16

Sometime when you look at an image, you may wonder from what angle the image was taken. Basically, the photo shooting is a process to project 3D objects onto 2D images. In this project, we will study the math of the projection, and from that we could know where the camera is.

First, the projection is a linear transformation, which we will talk about in Chapter 4. As we will learn, any linear transformation for a vector v can be represented as a matrix-vector multiplication, Av. However, if we use the ordinary coordinate system, such as using (x, y) for a 2D point and (x, y, z) for a 3D point, the linear transformation cannot represent the movement of object. For example, if we want to move a point (x_1, x_2) to (y_1, y_2), we need

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} t_1 \\ t_2 \end{bmatrix},$$

where $(t_1, t_2) = (y_1 - x_1, y_2 - x_2)$. To solve this problem, we can use the *homogeneous coordinate*, which adds 1 in the additional dimension. For example, (x, y) becomes $(x, y, 1)$ and (x, y, z) becomes $(x, y, z, 1)$. As the result, the translation (movement) can be represented as

$$\begin{bmatrix} y_1 \\ y_2 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_1 \\ 0 & 1 & t_2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ 1 \end{bmatrix}.$$

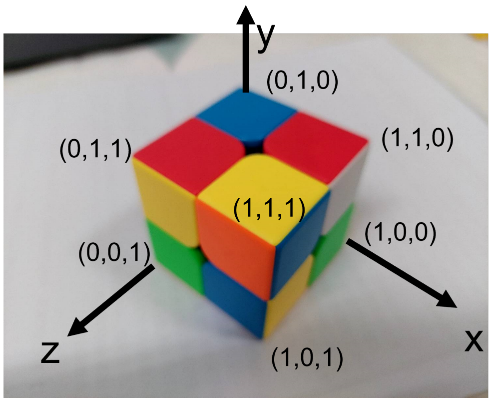
And this is a linear transformation.

To project a 3D point $(x, y, z, 1)$ to a 2D point $(x, y, 1)$, we need a 3×4 matrix, called a projection matrix. If we consider an ideal camera, whose projection matrix P can be expressed as

$$P = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & 1 \end{bmatrix},$$

which as 11 vairables.

In this project, we want to learn how to find the projection matrix from a single image. But first, we need to build the coordinate systems for 3D and for 2D. Figure 1 shows an example, which is a Rubik's Cube. We can decide the 3D



3D	2D
(0,0,1)	(46, 69)
(0,1,0)	(72, 16)
(0,1,1)	(38, 27)
(1,0,0)	(104, 65)
(1,0,1)	(79, 94)
(1,1,0)	(112, 33)
(1,1,1)	(83, 65)

coordinate by ourselves, so I just makes the most convenience one, as shown in the figure. Second, we need to decide the 2D coordinates for the corresponding points. The top-left corder of the image is $(0, 0)$ and the horizontal axis is for x and the vertical axis is for y. The table in Figure 1 shows the mapping between two different coordinates. I got those image coordinates using Microsoft Paint, and you can use any image viewers or editors that can show the position of mouse cursor to get those image coordinates.

Next, we need to use those mapped points to calculate the projection matrix P. Let $x = [x_1, x_2, x_3, 1]$ be a 3D point and $y = [y_1, y_2, 1]$ is the corresponding point in the 2D image. So we have $Px = \alpha y$,

$$Px = \begin{bmatrix} x_1p_{11} + x_2p_{12} + x_3p_{13} + p_{14} \\ x_1p_{21} + x_2p_{22} + x_3p_{23} + p_{24} \\ x_1p_{31} + x_2p_{32} + x_3p_{33} + 1 \end{bmatrix} = \alpha \begin{bmatrix} y_1 \\ y_2 \\ 1 \end{bmatrix}.$$

The last equation can help decide that $\alpha = x_1p_{31} + x_2p_{32} + x_3p_{33} + 1$. And therefore

$$x_1p_{11} + x_2p_{12} + x_3p_{13} + p_{14} = \alpha y_1 = (x_1p_{31} + x_2p_{32} + x_3p_{33} + 1)y_1,$$

$$x_1p_{21} + x_2p_{22} + x_3p_{23} + p_{24} = \alpha y_2 = (x_1p_{31} + x_2p_{32} + x_3p_{33} + 1)y_2.$$

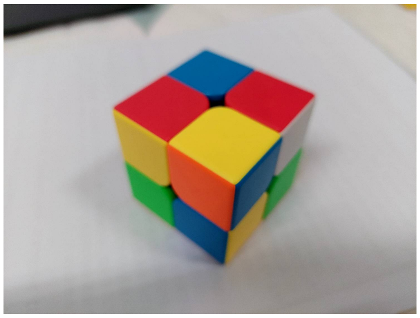
If we put the variables to the left-hand-side and the constants to the right-hand-side, we have

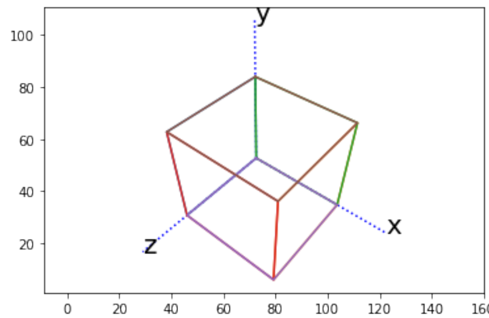
$$x_1p_{11} + x_2p_{12} + x_3p_{13} + p_{14} - y_1x_1p_{31} - y_1x_2p_{32} - y_1x_3p_{33} = y_1,$$

$$x_1p_{21} + x_2p_{22} + x_3p_{23} + p_{24} - y_2x_1p_{31} - y_2x_2p_{32} - y_2x_3p_{33} = y_2.$$

For each point, we can obtain two equations. And because we have 11 unknows, we need at least 11 equations. Fortunately, we have 7 points in this example, which means that there are 14 equations. So we can pick 11 equations and solve the equations to get the projection matrix P. The linear system $Ax = b$ is formulated as follows. The vector x contains 11 unknowns

$$x^T = [p_{11}, p_{12}, p_{13}, p_{14}, p_{21}, p_{22}, p_{23}, p_{24}, p_{31}, p_{32}, p_{33}].$$





Matrix A is a 11×11 matrix, which is a submatrix of a 14×11 matrix,

$$\begin{bmatrix}
x_1^{(1)} & x_2^{(1)} & x_3^{(1)} & 1 & 0 & 0 & 0 & 0 & -x_1^{(1)}y_1^{(1)} & -x_2^{(1)}y_1^{(1)} & -x_3^{(1)}y_1^{(1)} \\
0 & 0 & 0 & 0 & x_1^{(1)} & x_2^{(1)} & x_3^{(1)} & 1 & -x_1^{(1)}y_2^{(1)} & -x_2^{(1)}y_2^{(1)} & -x_3^{(1)}y_2^{(1)} \\
x_1^{(2)} & x_2^{(2)} & x_3^{(2)} & 1 & 0 & 0 & 0 & 0 & -x_1^{(2)}y_1^{(2)} & -x_2^{(2)}y_1^{(2)} & -x_3^{(2)}y_1^{(2)} \\
0 & 0 & 0 & 0 & x_1^{(2)} & x_2^{(2)} & x_3^{(2)} & 1 & -x_1^{(2)}y_2^{(2)} & -x_2^{(2)}y_2^{(2)} & -x_3^{(2)}y_2^{(2)} \\
x_1^{(3)} & x_2^{(3)} & x_3^{(3)} & 1 & 0 & 0 & 0 & 0 & -x_1^{(3)}y_1^{(3)} & -x_2^{(3)}y_1^{(3)} & -x_3^{(3)}y_1^{(3)} \\
0 & 0 & 0 & 0 & x_1^{(3)} & x_2^{(3)} & x_3^{(3)} & 1 & -x_1^{(3)}y_2^{(3)} & -x_2^{(3)}y_2^{(3)} & -x_3^{(3)}y_2^{(3)} \\
x_1^{(4)} & x_2^{(4)} & x_3^{(4)} & 1 & 0 & 0 & 0 & 0 & -x_1^{(4)}y_1^{(4)} & -x_2^{(4)}y_1^{(4)} & -x_3^{(4)}y_1^{(4)} \\
0 & 0 & 0 & 0 & x_1^{(4)} & x_2^{(4)} & x_3^{(4)} & 1 & -x_1^{(4)}y_2^{(4)} & -x_2^{(4)}y_2^{(4)} & -x_3^{(4)}y_2^{(4)} \\
x_1^{(5)} & x_2^{(5)} & x_3^{(5)} & 1 & 0 & 0 & 0 & 0 & -x_1^{(5)}y_1^{(5)} & -x_2^{(5)}y_1^{(5)} & -x_3^{(5)}y_1^{(5)} \\
0 & 0 & 0 & 0 & x_1^{(5)} & x_2^{(5)} & x_3^{(5)} & 1 & -x_1^{(5)}y_2^{(5)} & -x_2^{(5)}y_2^{(5)} & -x_3^{(5)}y_2^{(5)} \\
x_1^{(6)} & x_2^{(6)} & x_3^{(6)} & 1 & 0 & 0 & 0 & 0 & -x_1^{(6)}y_1^{(6)} & -x_2^{(6)}y_1^{(6)} & -x_3^{(6)}y_1^{(6)} \\
0 & 0 & 0 & 0 & x_1^{(6)} & x_2^{(6)} & x_3^{(6)} & 1 & -x_1^{(6)}y_2^{(6)} & -x_2^{(6)}y_2^{(6)} & -x_3^{(6)}y_2^{(6)} \\
x_1^{(7)} & x_2^{(7)} & x_3^{(7)} & 1 & 0 & 0 & 0 & 0 & -x_1^{(7)}y_1^{(7)} & -x_2^{(7)}y_1^{(7)} & -x_3^{(7)}y_1^{(7)} \\
0 & 0 & 0 & 0 & x_1^{(7)} & x_2^{(7)} & x_3^{(7)} & 1 & -x_1^{(7)}y_2^{(7)} & -x_2^{(7)}y_2^{(7)} & -x_3^{(7)}y_2^{(7)}
\end{bmatrix},$$

where the superscript (i) means the ith point. In the example program, I picked the first 11 rows. But in your assignment, you need to choose diiferent combinations. Last, the vector b is given as

$$b^T = [y_1^{(1)}, y_2^{(1)}, y_1^{(2)}, y_2^{(2)}, y_1^{(3)}, y_2^{(3)}, y_1^{(4)}, y_2^{(4)}, y_1^{(5)}, y_2^{(5)}, y_1^{(6)}, y_2^{(6)}, y_1^{(7)}, y_2^{(7)}].$$

To validate the correctness of the solved projection matrix, we create a 3D model for a cube, and apply the projection matrix to it. Figure 2 shows the result. It can be seen that the model matches the original image well.

1 Assignments

1. (20%) There are 14 equations, but we only have 11 unknows. Try at least 3 different selections of equations, and check which selection gives the best result.

2. (20%) When the number of equations is larger than the number of unknowns, we can solve the linear system using *linear least square* method, which we will talk about in Chapter 5. Check out the linear least square function provided in numpy, numpy.linalg.lstsq, and use it to solve the overdermined system. Compare the answer with the results in question 1. Do you think the answer is better than those in question 1 or not? Justify your thought.

3. (20%) Take a photo of a rectangular object (cubid), but not a cube like the example. Use the same algorithm to compute the projection matrix. Also, draw a 3D model that is similar to the object in your photo. Be careful about the order of input points, because the drawing function I provided makes some assumptions about the point order. Notice your 3D model should keep the same ratio as the real object to make the linear system correct.

4. (20%) Read the article on the website `https://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/EPSRC_SSAZ/epsrc_ssaz.html` for Single View Geometry, and learn the algorithm of how to find the camera position (from which angle) in the 3D coordinate system you create. Explain why and how the algorithm works.

5. (20%) In this assignment, we decide the coordinates of the corresponding points on the image by hand. Design an algorithm to automatically select the corresponding points, and give a simple example (using code) to show your algorithm works. You can assume the 3D model is given, and the desired points in 3D model are also known. Please give the citation of the algorithm if the algorithm is from some papers or books or websites. You will get 0 for the entire assignment if you fail to do so.

2 Submission

1. Write a report in PDF file that includes the answers of question (1), (2), (3), (4), and (5).

2. The code of (1), (2), (3), (5).

3. Zip them and submit the zip file to the eeclass.