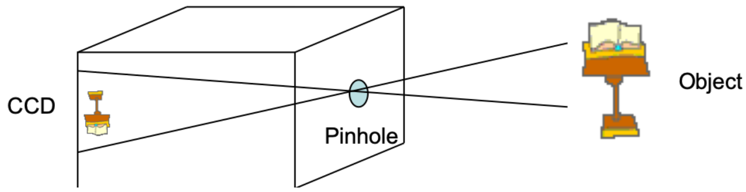
3-D Computer vision Mathematics

Richard Xu

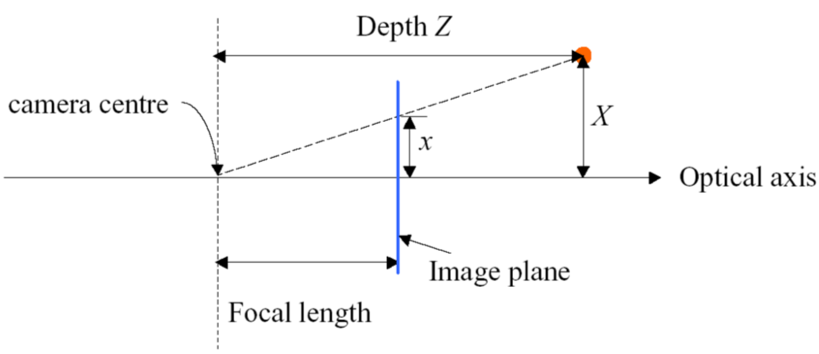
January 7, 2023

1 A Simple Pinhole Camera Model

For anyone with a digital camera, it's on every iPhone these days! CCD/CMOS is a type of imaging sensor:



It's rather strange to see things when the object and what appears on the CCD/CMOS sensor (we will refer to as image plane) are upside down, so we redraw it to look like:



Now it looks more geometrically pleasing.

1.1 homogeneous co-ordinate system

for all points (x, y, z) on the plane containing the origin and orthogonal to vector $[a \quad b \quad c]^\top$, they must satisfy the following:

$$ax + by + cz = 0 \tag{1}$$

Obviously, we can pick out some very specific z, for example $z = 1$:

$$[x \quad y \quad 1]^\top \tag{2}$$

now, if $[x \quad y \quad 1]^\top$ satisfies Eq.(1), then, if we times ϵ to each of the dimensions, i.e., $[\epsilon x \quad \epsilon y \quad \epsilon]^\top$ also satisfies:

$$a\epsilon x + b\epsilon y + c\epsilon = 0 \tag{3}$$

it makes sense, as one may not just "cut" the plane at $z = 1$, it is also possible to cut it at different z values.
A geometric example:

1. given a 2-d point $[1.6 \quad 1.2]$ at the plane $z = 1$, thus, $[1.6 \quad 1.2 \quad 1]$
2. it also represents the **same** object at position $[3.2 \quad 2.4]$ on plane $z = 2$, thus, $[3.2 \quad 2.4 \quad 2]$

therefore, under 2-d homogeneous co-ordinate system, $[1.6 \quad 1.2 \quad 1]$ and $[3.2 \quad 2.4 \quad 2]$ are in fact the same point!

1.1.1 practical importance

for practical operations, we need to keep all transformation in matrix multiplication, this is because:

$$\mathbf{A}_1\mathbf{A}_2 \dots \mathbf{A}_m\mathbf{X} = \big(\mathbf{A}_1\mathbf{A}_2 \dots \mathbf{A}_m\big)\mathbf{X} \tag{4}$$

however, if one is to look at translation operation:

$$\mathbf{X}' = \mathbf{X} + \mathbf{t} \tag{5}$$

there is no way we can write an translation operation in terms of a matrix multiplication. However, we can do so via homogeneous coordinate system, i.e.,

$$\begin{bmatrix} \mathbf{X}' \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix} \tag{6}$$

1.1.2 another interesting factor about homogeneous coordinate system

looking at figure (1), the **important** thing is that if every object is defined according to the **camera** coordinate system, where $[0 \quad 0 \quad 0]$ at **camera center**. Then, their positions on the CCD/image plane can be easily found by similar triangles:

$$\frac{X}{x} = \frac{Z}{f} \quad \implies \quad x = \frac{f}{Z}X \tag{7}$$

similarly,

$$\frac{Y}{y} = \frac{Z}{f} \quad \implies \quad y = \frac{f}{Z}Y \tag{8}$$

combine the two, one have:

$$\begin{aligned} \begin{bmatrix} x \\ y \\ w \end{bmatrix} &= \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \\ &= \begin{bmatrix} fX \\ fY \\ Z \end{bmatrix} = \begin{bmatrix} f\frac{X}{Z} \\ f\frac{Y}{Z} \\ 1 \end{bmatrix} \end{aligned} \tag{9}$$

The last equality is due to the fact that the coordinates are homogeneous. Then, we also need to add translation, since the image is from a different source than the CCD (usually in the center), so we have the following expression:

$$\begin{aligned} u &= m_u f \frac{X}{Z} + m_u t_u \\ v &= m_v f \frac{Y}{Z} + m_v t_v \end{aligned} \tag{10}$$

$$\begin{aligned} \begin{bmatrix} u \\ v \\ w \end{bmatrix} &= \begin{bmatrix} m_u f & 0 & m_u t_u \\ 0 & m_v f & m_v t_v \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \\ &= \begin{bmatrix} m_u f X + m_u t_u Z \\ m_v f Y + m_v t_v Z \\ Z \end{bmatrix} = \begin{bmatrix} m_u f \frac{X}{Z} + m_u t_u \\ m_v f \frac{Y}{Z} + m_v t_v \\ 1 \end{bmatrix} \end{aligned} \tag{11}$$

2 line and plane representation

2.1 2-d line

Back in high school, people preferred to use the following equation for straight line:

$$y = mx + b \tag{12}$$

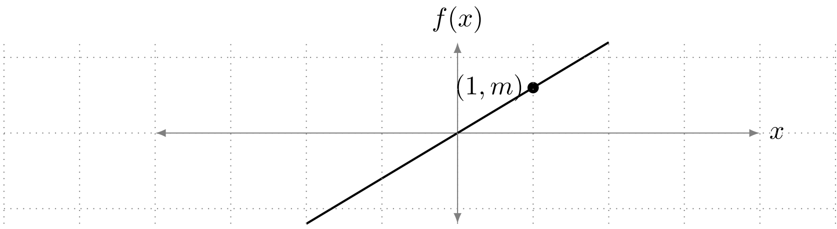


Figure 1: $y = mx$

However, this representation has some difficulty in for example vertical line, one would have to model it using:

$$y = \infty \times x \tag{13}$$

2.1.1 new representation: through origin

we can do something differently, imagine the following equation:

$$ax + by = 0 \tag{14}$$

or writing it out as a 2-d linear equation form:

$$\begin{bmatrix} a & b \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = 0 \tag{15}$$

Since $\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ is part of null-space of 1×2 matrix $[a \quad b]$ (null space always contains **zero** vector), then (x, y) must lie on a line passing origin, orthognal to 2-d normal vector $[a \quad b]^\top$. So, if you want to define a 2-d perpendicular line through the origin, then let:

$$[a \quad b] = d\,[1 \quad 0] \quad \forall d \in \mathbb{R} \tag{16}$$

2.1.2 new representation: not through origin

If one wants to have a generic line **not** passing through the origin, then:

$$ax + by + c = 0 \tag{17}$$

where c "shift" is parallel to the parallel of $[a \quad b]$, now we can see that the parameter becomes (a, b, c) and up to a certain scale, which means:

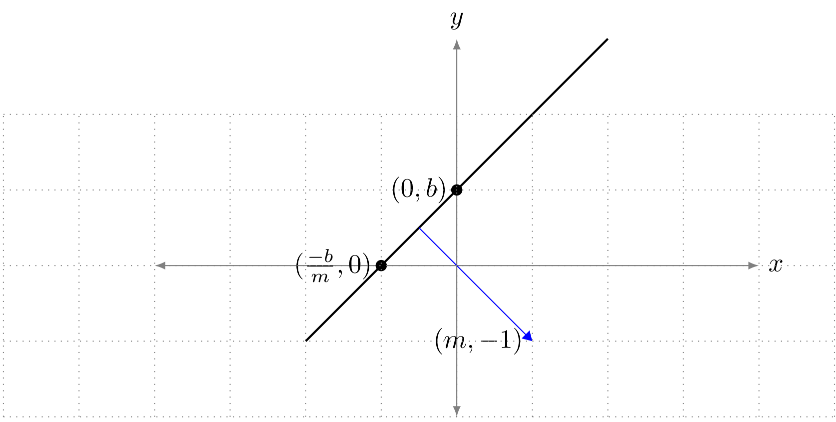
$$\begin{aligned} & ax + by + c = 0 \\ \implies & \frac{a}{d}x + \frac{b}{d}y + \frac{c}{d} = 0 \end{aligned} \tag{18}$$

2.1.3 Alternative way of looking at 2D line

One may argue we can still change the following into implicit representation:

$$\begin{aligned} & y = mx + b \\ \implies & [m \quad -1] \begin{bmatrix} x \\ y \end{bmatrix} + b = 0 \end{aligned} \tag{19}$$

Note that it means the normal vector $[m \quad -1]$ is decomposed by vectors $[m \quad 0]$ and $[0 \quad -1]$:



It can be seen that since b only plays the role of **shift** (because it is not directly multiplied by y), then m is changed by fixing b and if one. This means that a vertical line is achieved by letting $m \to \infty$, i.e. by "wiggling" $\frac{-b}{m}$ very close to the origin.

2.2 Equation of line and points: 3d

looking at $ax + by + c$ and extend it to 3d:

$$ax + by + cz = 0 \tag{20}$$

on a 3-d linear equation form:

$$\begin{bmatrix} a & b & c \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = 0 \tag{21}$$

Since $[0 \quad 0 \quad 0]^\top$ is an element of null-space of $[a \quad b \quad c]$, then (x, y, z) must lie on a **plane** passing origin, orthognal to 3D normal vector $[a \quad b \quad c]^\top$. If one wants to have a plane not passing through the origin, then:

$$ax + by + cz + d = 0 \tag{22}$$

where d "shifts" parallel planes orthogonal to $[a \quad b \quad c]$.

3 Camera Calibration

3.1 How object location relates to an image point?

Naturally:

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{P} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{23}$$

the inverse, i.e., determine the 3D ray from 2D image point $\mathbf{x}$ can be achieved by (this is to be discussed later):

$$\begin{aligned} \mathbf{X}_{\text{3D}}(\lambda) &= \mathbf{P}^{+}\mathbf{x} + \lambda \mathbf{C} \\ \text{where } \mathbf{P}\mathbf{P}^{+} &= \mathbf{I} \end{aligned} \tag{24}$$

Representing the entire projection matrix into a single 3×4 matrix $\mathbf{P}$ doesn't help. Because some parts of $\mathbf{P}$ are related to the camera itself, and other parts are related to the position of the camera.

3.2 what is camera calibration?

$$s\mathbf{x} = \quad \mathbf{K} \quad [\mathbf{R}|\mathbf{t}] \quad \mathbf{X} \tag{25}$$

$$\underbrace{s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}}_{\text{image}} = \underbrace{\begin{bmatrix} f_a & \gamma & u_0 \\ 0 & f_b & v_0 \\ 0 & 0 & 1 \end{bmatrix}}_{\text{intrinsic}} \underbrace{\begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix}}_{[\mathbf{R}|\mathbf{t}]} \underbrace{\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}}_{\text{object}} \tag{26}$$

3.3 Camera calibration details

$$s\,\mathbf{x} = \mathbf{K}\quad[\mathbf{R}\,|\mathbf{t}]\quad\mathbf{X}$$

$$\Longrightarrow s\begin{bmatrix}u\\v\\1\end{bmatrix} = \begin{bmatrix}f_a & \gamma & u_0\\0 & f_b & v_0\\0 & 0 & 1\end{bmatrix}\begin{bmatrix}r_{11} & r_{12} & r_{13} & t_x\\r_{21} & r_{22} & r_{23} & t_y\\r_{31} & r_{32} & r_{33} & t_z\end{bmatrix}\begin{bmatrix}x\\y\\z\\1\end{bmatrix} \tag{27}$$

1. Intrinsic parameter

$$\mathbf{K} = \begin{bmatrix}f_a & \gamma & u_0\\0 & f_b & v_0\\0 & 0 & 1\end{bmatrix} \tag{28}$$

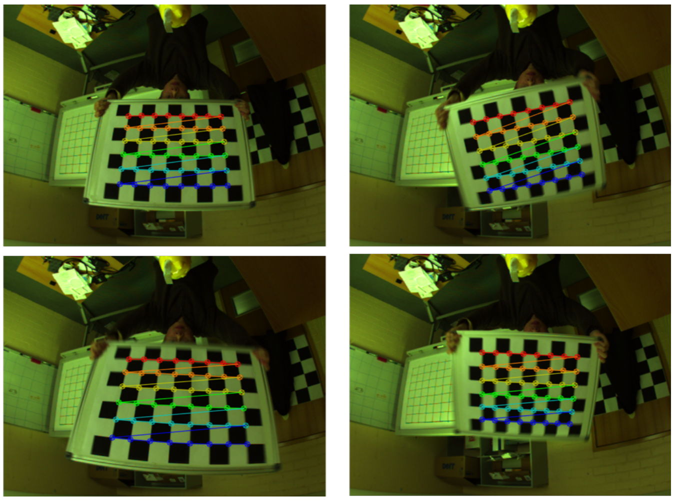
   compare this with Eq.(11) , we now have an extra skew coefficient γ to account for mechanism inaccuracies.

2. Extrinsic parameter

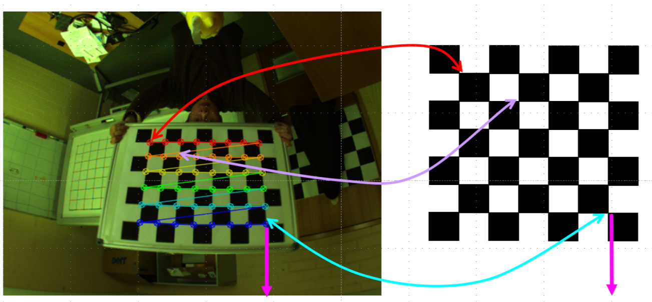
$$[\mathbf{R}\,|\mathbf{t}] = \begin{bmatrix}r_{11} & r_{12} & r_{13} & t_x\\r_{21} & r_{22} & r_{23} & t_y\\r_{31} & r_{32} & r_{33} & t_z\end{bmatrix} \tag{29}$$

4 Intrinsic Camera calibration

This rest is largely paraphrasing from [1]



4.1 "Data" collection: use Homography H as data



$$\mathbf{x} \qquad\qquad\qquad \mathbf{M} \tag{30}$$

Imagine we know that the 4^{th} and 5^{th} grid corner (i.e., $\mathbf{M} = [4 \quad 5]$ has image co-ordinates of $\mathbf{x} = [34.12 \quad 65.21]$, then, there exist a homography $\mathbf{H}$ such that:

$$\begin{gathered}\mathbf{x} = \mathbf{H}\mathbf{M} \\ \begin{bmatrix}34.12\\65.21\\1\end{bmatrix} = \mathbf{H}\begin{bmatrix}4\\5\\1\end{bmatrix} \qquad \text{as an example}\end{gathered} \tag{31}$$

Homography $\mathbf{H}$ acts like our "data", because it can be computed **beforehand** without needing any camera geometry geometry information. As long as you have 4 pairs of matching points, you can calculate a particular $\mathbf{H}$ up to a scale constant. So that's why I wave the checkerboard in mulitple poses, just to collect different $\mathbf{H}$.

Let us define $\mathbf{M}$ to be $\mathbf{X}$ without z^{th} component

$$\begin{gathered}\mathbf{x} = \mathbf{H}\mathbf{M} \\ \underbrace{\begin{bmatrix}u\\v\\1\end{bmatrix}}_{\mathbf{x}} = \underbrace{\mathbf{H}\begin{bmatrix}x\\y\\1\end{bmatrix}}_{\hat{\mathbf{x}}}\end{gathered} \tag{32}$$

Get 4 pair of points and we are done, yeah? Where is the catch? Image points have noises!

$$\sum_i \left[\left(\mathbf{x}_i - \hat{\mathbf{x}}_i\right)^\top \mathbf{\Lambda}^{-1}\left(\mathbf{x}_i - \hat{\mathbf{x}}_i\right)\right] \tag{33}$$

For simplicity, can just assume: $\mathbf{\Lambda} = \sigma^2\mathbf{I}$, i.e., noises are the same for each corner:

$$\min_{\mathbf{H}} \sum_i \|\mathbf{x}_i - \hat{\mathbf{x}}_i\| \tag{34}$$

4.2 alternative computation of H:

During the calculation of Homography, the following steps can be used (or you can use other methods if you prefer). However, I would like you to use more than 4 matching grid corners to reflect what computer vision people usually do.

Looking at each of the matching corner $(x, y, 1)$ and image corner $(u, v, 1)$, for example:

$$\begin{gathered}\begin{bmatrix}u\\v\\1\end{bmatrix} = \mathbf{H}\begin{bmatrix}x\\y\\1\end{bmatrix} \\ \begin{bmatrix}34.12\\65.21\\1\end{bmatrix} = \mathbf{H}\begin{bmatrix}4\\5\\1\end{bmatrix} \qquad \text{as an example}\end{gathered} \tag{35}$$

then we have:

$$
\begin{aligned}
\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} &= \begin{bmatrix} h_{1,1} & h_{1,2} & h_{1,3} \\ h_{2,1} & h_{2,2} & h_{2,3} \\ h_{3,1} & h_{3,2} & h_{3,3} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \\
\implies u &= \frac{h_{1,1}x + h_{1,2}y + h_{1,3}}{h_{3,1}x + h_{3,2}y + h_{3,3}} \\
v &= \frac{h_{2,1}x + h_{2,2}y + h_{2,3}}{h_{3,1}x + h_{3,2}y + h_{3,3}}
\end{aligned} \tag{36}
$$

4.2.1 Direct minimization method

you may just try to solve it as a minimization problem:

$$
\underset{h_{1,1},\ldots,h_{3,3}}{\arg\min} \sum_{i=1}^{N} \left(u_i - \frac{h_{1,1}x_i + h_{1,2}y_i + h_{1,3}}{h_{3,1}x_i + h_{3,2}y_i + h_{3,3}} \right)^2 + \left(v_i - \frac{h_{2,1}x_i + h_{2,2}y_i + h_{2,3}}{h_{3,1}x_i + h_{3,2}y_i + h_{3,3}} \right)^2 \tag{37}
$$

where $N > 4$

4.2.2 $\mathbf{Ax} = \mathbf{0}$ method

the last two equations from Eq.(36), it gives you the following:

$$
\begin{aligned}
u\big(h_{3,1}x + h_{3,2}y + h_{3,3}\big) &= h_{1,1}x + h_{1,2}y + h_{1,3} \\
v\big(h_{3,1}x + h_{3,2}y + h_{3,3}\big) &= h_{2,1}x + h_{2,2}y + h_{2,3}
\end{aligned} \tag{38}
$$

Rearrange them to give you two equations in a linear system:

$$
\begin{aligned}
-h_{1,1}x - h_{1,2}y - h_{1,3} + uxh_{3,1} + uyh_{3,2} + uh_{3,3} &= 0 \\
-h_{2,1}x - h_{2,2}y - h_{2,3} + vxh_{3,1}x + vyh_{3,2} + vh_{3,3} &= 0 \\
\implies \begin{bmatrix} -x & -y & -1 & 0 & 0 & 0 & ux & uy & u \\ 0 & 0 & 0 & -x & -y & -1 & vx & vy & v \end{bmatrix} \begin{bmatrix} h_{1,1} \\ h_{1,2} \\ h_{1,3} \\ h_{2,1} \\ h_{2,2} \\ h_{2,3} \\ h_{3,1} \\ h_{3,2} \\ h_{3,3} \end{bmatrix} &= \mathbf{0}
\end{aligned} \tag{39}
$$

If this is the first pair of matching corners, then let's add some indices to it such that $x \to x_i$ and $y \to y_i$. Suppose we have $N > 4$ matching angles, therefore, we have our linear system:

$$
\underbrace{\begin{bmatrix}
-x_1 & -y_1 & -1 & 0 & 0 & 0 & u_1x_1 & u_1y_1 & u_1 \\
0 & 0 & 0 & -x_1 & -y_1 & -1 & v_1x_1 & v_1y_1 & v_1 \\
-x_2 & -y_2 & -1 & 0 & 0 & 0 & u_2x_2 & u_2y_2 & u_2 \\
0 & 0 & 0 & -x_2 & -y_2 & -1 & v_2x_2 & v_2y_2 & v_2 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
-x_N & -y_N & -1 & 0 & 0 & 0 & u_Nx_N & u_Ny_N & u_N \\
0 & 0 & 0 & -x_N & -y_N & -1 & v_Nx_N & v_Ny_N & v_N
\end{bmatrix}}_{A_{(N\times 2)\times 9}}
\underbrace{\begin{bmatrix} h_{1,1} \\ h_{1,2} \\ h_{1,3} \\ h_{2,1} \\ h_{2,2} \\ h_{2,3} \\ h_{3,1} \\ h_{3,2} \\ h_{3,3} \end{bmatrix}}_{\mathbf{h}_{9\times 1}} = \mathbf{0} \tag{40}
$$

Therefore, we have a system:

$$\mathbf{A}\mathbf{h} = \mathbf{0} \tag{41}$$

you can then compute SVD for $\mathbf{A}$ and let it be:

$$\begin{aligned}\mathbf{A} &= \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top \\ &= \sum_{i=1}^{9} \sigma_i \mathbf{u}_i (\mathbf{v}_i)^\top\end{aligned} \tag{42}$$

Now if you sort the $\{\sigma_i\}$ in descending order, i.e., $\sigma_1 \geq \sigma_2 \geq \dots \sigma_9$, then you just pick $\hat{\mathbf{h}} = \mathbf{v}_9$ which correspond to σ_9 (smallest) as the best approximation to solution of $\mathbf{A}\hat{\mathbf{h}} \approx \mathbf{0}$,

Why? think about SVD again: if $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$, then:

$$\begin{aligned}\mathbf{A} &= \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top \\ &= \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{-1} \\ \implies \mathbf{A}\mathbf{V} &= \mathbf{U}\mathbf{\Sigma} \\ \implies \mathbf{A}\begin{bmatrix}\mathbf{v}_1 & \dots & \mathbf{v}_9\end{bmatrix} &= \begin{bmatrix}\sigma_1\mathbf{u}_1 & \dots & \sigma_9\mathbf{u}_9\end{bmatrix} \qquad \sigma_1 \geq \sigma_2 \geq \dots \sigma_9\end{aligned} \tag{43}$$

Since all $\|\mathbf{u}_i\|_2 = 1 \quad \forall i$, then:

$$\|\mathbf{A}\mathbf{v}_9\|_2 = \sigma_9 \tag{44}$$

4.2.3 $\mathbf{A}\mathbf{x} = \mathbf{b}$ method

looking at Eq.(36), we have:

$$\begin{aligned}\begin{bmatrix}u\\v\\1\end{bmatrix} &= \begin{bmatrix}h_{1,1} & h_{1,2} & h_{1,3}\\ h_{2,1} & h_{2,2} & h_{2,3}\\ h_{3,1} & h_{3,2} & h_{3,3}\end{bmatrix}\begin{bmatrix}x\\y\\1\end{bmatrix} \\ \implies u &= \frac{h_{1,1}x + h_{1,2}y + h_{1,3}}{h_{3,1}x + h_{3,2}y + h_{3,3}} \\ v &= \frac{h_{2,1}x + h_{2,2}y + h_{2,3}}{h_{3,1}x + h_{3,2}y + h_{3,3}}\end{aligned} \tag{45}$$

and by letting $h_{3,3} = 1$:

$$\begin{aligned}&\begin{cases} u &= \frac{h_{1,1}x+h_{1,2}y+h_{1,3}}{h_{3,1}x+h_{3,2}y+h_{3,3}} \\ v &= \frac{h_{2,1}x+h_{2,2}y+h_{2,3}}{h_{3,1}x+h_{3,2}y+h_{3,3}}\end{cases} \\ \implies &\begin{cases} u &= \frac{h_{1,1}x+h_{1,2}y+h_{1,3}}{h_{3,1}x+h_{3,2}y+1} \\ v &= \frac{h_{2,1}x+h_{2,2}y+h_{2,3}}{h_{3,1}x+h_{3,2}y+1}\end{cases}\end{aligned} \tag{46}$$

this means that:

$$\begin{aligned}&\begin{cases} uh_{3,1}x + uh_{3,2}y + u &= h_{1,1}x + h_{1,2}y + h_{1,3} \\ vh_{3,1}x + vh_{3,2}y + v &= h_{2,1}x + h_{2,2}y + h_{2,3}\end{cases} \\ \implies &\begin{cases} h_{1,1}x + h_{1,2}y + h_{1,3} - uh_{3,1}x - uh_{3,2}y &= u \\ h_{2,1}x + h_{2,2}y + h_{2,3} - vh_{3,1}x - vh_{3,2}y &= v\end{cases}\end{aligned} \tag{47}$$

therefore, we end up with a linear system of $\mathbf{Ax} = \mathbf{b}$:

$$\underbrace{\begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -u_1 - x_1 & -u_1 y_1 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -v_1 x_1 & -v_1 y_1 \\ x_2 & y_2 & 1 & 0 & 0 & 0 & -u_2 x_2 & -u_2 y_2 \\ 0 & 0 & 0 & x_2 & y_2 & 1 & -v_2 x_2 & -v_2 y_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_N & y_N & 1 & 0 & 0 & 0 & -u_N x_N & -u_N y_N \\ 0 & 0 & 0 & x_N & y_N & 1 & -v_N x_N & -v_N y_N \end{bmatrix}}_{A_{(N\times 2)\times 8}} \underbrace{\begin{bmatrix} h_{1,1} \\ h_{1,2} \\ h_{1,3} \\ h_{2,1} \\ h_{2,2} \\ h_{2,3} \\ h_{3,1} \\ h_{3,2} \end{bmatrix}}_{\mathbf{h}_{8\times 1}} = \underbrace{\begin{bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ \vdots \\ u_N \\ v_N \end{bmatrix}}_{\mathbf{b}_{(N\times 2)\times 1}} \tag{48}$$

4.3 back to the projection matrix

now let's examine the projection matrix again!

$$\begin{aligned} s\,\mathbf{x} &= \mathbf{K}[\mathbf{R} \quad \mathbf{t}]\mathbf{X} \\ s\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} &= \begin{bmatrix} f_a & \gamma & u_0 \\ 0 & f_b & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \end{aligned} \tag{49}$$

let's assume the board is a planar surface, and $z = 0$:

$$\begin{aligned} s\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} &= \mathbf{K} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix} \begin{bmatrix} x \\ y \\ 0 \\ 1 \end{bmatrix} \\ &= \mathbf{K}\left[\mathbf{r}_1 \quad \mathbf{r}_2 \quad \mathbf{t}\right] \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \\ &= \mathbf{K}\left[\mathbf{r}_1 \quad \mathbf{r}_2 \quad \mathbf{t}\right]\mathbf{M} \end{aligned} \tag{50}$$

obviously, we need to re-arrange to **cancel** auxiliary variable $\mathbf{r}$ and $\mathbf{t}$

4.4 Combine the two case together

substitute $\mathbf{x} = \mathbf{HM}$

$$\begin{aligned} s\,\mathbf{x} &= \mathbf{K}\left[\mathbf{r}_1 \quad \mathbf{r}_2 \quad \mathbf{t}\right]\mathbf{M} \\ &= \mathbf{HM} \\ \implies \mathbf{H} &= \lambda \mathbf{K}\left[\mathbf{r}_1 \quad \mathbf{r}_2 \quad \mathbf{t}\right] \qquad \lambda = \frac{1}{s} \end{aligned} \tag{51}$$

kept on going:

$$\begin{aligned} \mathbf{H} = [\mathbf{h}_1 \quad \mathbf{h}_2 \quad \mathbf{h}_3] &= \lambda \mathbf{K}\left[\mathbf{r}_1 \quad \mathbf{r}_2 \quad \mathbf{t}\right] \\ \implies [\mathbf{h}_1 \quad \mathbf{h}_2] &= \lambda \mathbf{K}\left[\mathbf{r}_1 \quad \mathbf{r}_2\right] \qquad \text{we do not need } \mathbf{h}_3 \text{ and } \mathbf{t} \\ \implies \mathbf{h}_1^\top \mathbf{K}^{-\top}\mathbf{K}^{-1}\mathbf{h}_2 &= 0 \quad \text{case 1} \\ \text{also} \implies \mathbf{h}_1^\top \mathbf{K}^{-\top}\mathbf{K}^{-1}\mathbf{h}_1 &= \mathbf{h}_2^\top \mathbf{K}^{-\top}\mathbf{K}^{-1}\mathbf{h}_2 \quad \text{case 2} \end{aligned} \tag{52}$$

so $\mathbf{r}$ and $\mathbf{t}$ are completely disappeared

4.5 prove $\mathrm{h}_1^\top \mathrm{K}^{-\top}\mathrm{K}^{-1}\mathrm{h}_2 = 0$ case 1

$$
\begin{aligned}
\mathbf{H} &= \mathbf{K}\left[\mathbf{r}_1 \quad \mathbf{r}_2 \quad \mathbf{t}\right] \\
\implies \left[\mathbf{h}_1 \quad \mathbf{h}_2 \quad \mathbf{h}_3\right] &= \mathbf{K}\left[\mathbf{r}_1 \quad \mathbf{r}_2 \quad \mathbf{t}\right] \\
\mathbf{h}_1 &= \mathbf{K}\mathbf{r}_1 \implies \mathbf{r}_1 = \mathbf{K}^{-1}\mathbf{h}_1 \\
\mathbf{h}_2 &= \mathbf{K}\mathbf{r}_2 \implies \mathbf{r}_2 = \mathbf{K}^{-1}\mathbf{h}_2 \\
\mathbf{r}_1^\top \mathbf{r}_2 &= \left(\mathbf{K}^{-1}\mathbf{h}_1\right)^\top \mathbf{K}^{-1}\mathbf{h}_2 \\
&= \mathbf{h}_1^\top \mathbf{K}^{-\top}\mathbf{K}^{-1}\mathbf{h}_2 = 0
\end{aligned}
\tag{53}
$$

because rotation matrix $\mathbf{R}$ is orthogonal: $\mathbf{r}_i^\top \mathbf{r}_j = 0 \forall i \neq j$ λ won't matter:

$$
\begin{aligned}
\mathbf{h}_1 = \lambda \mathbf{K}\mathbf{r}_1 &\implies \mathbf{r}_1 = \frac{1}{\lambda}\mathbf{K}^{-1}\mathbf{h}_1 \\
\mathbf{h}_2 = \lambda \mathbf{K}\mathbf{r}_2 &\implies \mathbf{r}_2 = \frac{1}{\lambda}\mathbf{K}^{-1}\mathbf{h}_2 \\
&\implies \frac{1}{\lambda^2}\mathbf{h}_1^\top \mathbf{K}^{-\top}\mathbf{K}^{-1}\mathbf{h}_2 = 0
\end{aligned}
\tag{54}
$$

4.6 prove $\mathrm{h}_1^\top \mathrm{K}^{-\top}\mathrm{K}^{-1}\mathrm{h}_1 = \mathrm{h}_2^\top \mathrm{K}^{-\top}\mathrm{K}^{-1}\mathrm{h}_2$ case 2

$$
\begin{aligned}
\mathbf{r}_1^\top \mathbf{r}_1 &= \left(\mathbf{K}^{-1}\mathbf{h}_1\right)^\top \mathbf{K}^{-1}\mathbf{h}_1 \\
&= \mathbf{h}_1^\top \mathbf{K}^{-\top}\mathbf{K}^{-1}\mathbf{h}_1 = 1
\end{aligned}
\tag{55}
$$

similarly,

$$
\begin{aligned}
\mathbf{r}_2^\top \mathbf{r}_2 &= \left(\mathbf{K}^{-1}\mathbf{h}_2\right)^\top \mathbf{K}^{-1}\mathbf{h}_2 \\
&= \mathbf{h}_2^\top \mathbf{K}^{-\top}\mathbf{K}^{-1}\mathbf{h}_1 = 1
\end{aligned}
\tag{56}
$$

together:

$$
\implies \mathbf{h}_1^\top \mathbf{K}^{-\top}\mathbf{K}^{-1}\mathbf{h}_1 = \mathbf{h}_2^\top \mathbf{K}^{-\top}\mathbf{K}^{-1}\mathbf{h}_2 \tag{57}
$$

again, because rotation matrix $\mathbf{R}$ is orthogonal

4.7 now you have a linear system

a linear system:

$$
\begin{aligned}
&\mathbf{h}_1^\top \mathbf{K}^{-\top}\mathbf{K}^{-1}\mathbf{h}_2 = 0 \\
&\mathbf{h}_1^\top \mathbf{K}^{-\top}\mathbf{K}^{-1}\mathbf{h}_1 - \mathbf{h}_2^\top \mathbf{K}^{-\top}\mathbf{K}^{-1}\mathbf{h}_2 = 0 \\
\implies &\mathbf{h}_1^\top \mathbf{B}\mathbf{h}_2 = 0 \\
&\mathbf{h}_1^\top \mathbf{B}\mathbf{h}_1 - \mathbf{h}_2^\top \mathbf{B}\mathbf{h}_2 = 0 \qquad \text{let: } \mathbf{B} = \mathbf{K}^{-\top}\mathbf{K}^{-1}
\end{aligned}
\tag{58}
$$

knowing $\mathbf{K} = \begin{bmatrix} f_a & \gamma & u_0 \\ 0 & f_b & v_0 \\ 0 & 0 & 1 \end{bmatrix}$
you can perform python code to get expression of $\mathbf{B} = \mathbf{K}^{-\top}\mathbf{K}^{-1}$

4.8 Solve for B

notice $\mathbf{B}$ is symmetrical matrix, so there are only 6 degree-of-freedom

$$\begin{bmatrix} B_{1,1} & B_{12} & B_{13} \\ B_{12} & B_{22} & B_{23} \\ B_{13} & B_{23} & B_{33} \end{bmatrix} \tag{59}$$

so we let $\mathbf{B} = \begin{bmatrix} B_{11}, B_{12}, B_{22}, B_{13}, B_{23}, B_{33} \end{bmatrix}^\top$

$$\begin{aligned} \mathbf{h}_1^\top \mathbf{B} \mathbf{h}_2 &= 0 \\ \mathbf{h}_1^\top \mathbf{B} \mathbf{h}_1 - \mathbf{h}_2^\top \mathbf{B} \mathbf{h}_2 &= 0 \qquad \text{can be written as:} \end{aligned} \tag{60}$$

$$\begin{bmatrix} h_{11}h_{21} & h_{11}h_{22} + h_{12}h_{21} & h_{12}h_{22} & h_{11}h_{23} + h_{13}h_{21} & h_{13}h_{22} + h_{12}h_{23} & h_{13}h_{23} \\ h_{11}h_{11} - h_{21}h_{21} & 2h_{11}h_{12} - 2h_{21}h_{22} & h_{12}h_{12} - h_{22}h_{22} & 2h_{11}h_{13} - 2h_{21}h_{23} & 2h_{12}h_{13} - 2h_{22}h_{23} & h_{13}h_{13} - h_{23}h_{23} \end{bmatrix}$$
$$\times \begin{bmatrix} B_{11} \\ B_{12} \\ B_{21} \\ B_{22} \\ B_{13} \\ B_{23} \\ B_{33} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{61}$$

then you can solve for $\mathbf{K}$ from $\mathbf{B}$

5 calibrate extrinsic parameters

Extrinsic parameter transforms object defined in terms of world coordinate, to a new co-ordinate system where the origin is at the camera center. It is actually much easier to do it only requires a single image/pose:

(62)

but it has many object and image pairs.

$$
\begin{aligned}
s\begin{bmatrix} u' \\ v' \\ w' \end{bmatrix} &= \underbrace{\begin{bmatrix} p_{1,1} & p_{1,2} & p_{1,3} & p_{1,4} \\ p_{2,1} & p_{2,2} & p_{2,3} & p_{2,4} \\ p_{3,1} & p_{3,2} & p_{3,3} & p_{3,4} \end{bmatrix}}_{\mathbf{P}} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \\
&= \begin{bmatrix} - & \mathbf{p}_1 & - \\ - & \mathbf{p}_2 & - \\ - & \mathbf{p}_3 & - \end{bmatrix} \begin{bmatrix} | \\ \mathbf{X} \\ | \end{bmatrix} \\
&= \begin{bmatrix} \mathbf{p}_1^\top \mathbf{X} \\ \mathbf{p}_2^\top \mathbf{X} \\ \mathbf{p}_3^\top \mathbf{X} \end{bmatrix}
\end{aligned} \tag{63}
$$

convert $\begin{bmatrix} u' \\ v' \\ w' \end{bmatrix} \rightarrow \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$ in image point

$$
\begin{aligned}
&\Longrightarrow u = \frac{\mathbf{p}_1^\top \mathbf{X}}{\mathbf{p}_3^\top \mathbf{X}} \qquad v = \frac{\mathbf{p}_2^\top \mathbf{X}}{\mathbf{p}_3^\top \mathbf{X}} \\
&\Longrightarrow \mathbf{p}_1^\top \mathbf{X} - \mathbf{p}_3^\top \mathbf{X} u = 0 \qquad \mathbf{p}_2^\top \mathbf{X} - \mathbf{p}_3^\top \mathbf{X} v = 0
\end{aligned} \tag{64}
$$

5.1 another system of linear equation

by looking at just a single point:

$$
\mathbf{p}_1^\top \mathbf{X} - \mathbf{p}_3^\top \mathbf{X} u = 0 \qquad \mathbf{p}_2^\top \mathbf{X} - \mathbf{p}_3^\top \mathbf{X} v = 0 \Longrightarrow \begin{bmatrix} \mathbf{X}^\top & \mathbf{0} & -u\mathbf{X}^\top \\ \mathbf{0} & \mathbf{X}^\top & -v\mathbf{X}^\top \end{bmatrix} \begin{bmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \\ \mathbf{p}_3 \end{bmatrix} = \mathbf{0} \tag{65}
$$

N points:

$$
\underbrace{\begin{bmatrix} \mathbf{X}_1^\top & \mathbf{0} & -u\mathbf{X}_1^\top \\ \mathbf{0} & \mathbf{X}_1^\top & -v\mathbf{X}_1^\top \\ \vdots & \vdots & \vdots \\ \mathbf{X}_N^\top & \mathbf{0} & -u\mathbf{X}_N^\top \\ \mathbf{0} & \mathbf{X}_N^\top & -v\mathbf{X}_N^\top \end{bmatrix}}_{\mathbf{A}} \begin{bmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \\ \mathbf{p}_3 \end{bmatrix} = \begin{bmatrix} \mathbf{X}_1^\top & \mathbf{0} & -u\mathbf{X}_1^\top \\ \mathbf{0} & \mathbf{X}_1^\top & -v\mathbf{X}_1^\top \\ \vdots & \vdots & \vdots \\ \mathbf{X}_N^\top & \mathbf{0} & -u\mathbf{X}_N^\top \\ \mathbf{0} & \mathbf{X}_N^\top & -v\mathbf{X}_N^\top \end{bmatrix} \begin{bmatrix} p_{1,1} \\ p_{1,2} \\ p_{1,3} \\ p_{1,4} \\ p_{2,1} \\ p_{2,2} \\ p_{2,3} \\ p_{2,4} \\ p_{3,1} \\ p_{3,2} \\ p_{3,3} \\ p_{3,4} \end{bmatrix} = \mathbf{0} \tag{66}
$$

5.2 Solving for $\hat{\mathbf{p}}$

if $\mathbf{P}$ were the original projection matrix, then we let $\mathbf{p} = \text{vect}(\mathbf{P})$:
if we were to solve:

$$
\hat{\mathbf{p}} = \arg\min_{\mathbf{p}} \|\mathbf{A}\mathbf{p}\|^2 \tag{67}
$$

the most obvious solution is $\mathbf{p} = \mathbf{0}$!, adding the constraint, then the objective function becomes:

$$
\hat{\mathbf{p}} = \arg\min_{\mathbf{p}} \|\mathbf{A}\mathbf{p}\|^2 \qquad \text{s.t. } \|\mathbf{p}\|^2 = 1 \tag{68}
$$

5.2.1 equivalence of Frobenius norm constraint

although the objective function in Eq.(68) is to put the vector L2 norm in the vectorized $\mathbf{p} = \text{vec}(\mathbf{P})$, when converting it back to the matrix $\mathbf{P}$ (I use Same notation here), using the Frobenious norm is equivalent to constraint:

imagine let $\|\mathbf{P}\|_F = s$, i.e., Frobenius norm $= s$

$$
\begin{aligned}
\begin{bmatrix} x \\ y \\ z \end{bmatrix} &= \underbrace{\begin{bmatrix} p_{1,1} & p_{1,2} & p_{1,3} & p_{1,4} \\ p_{2,1} & p_{2,2} & p_{2,3} & p_{2,4} \\ p_{3,1} & p_{3,2} & p_{3,3} & p_{3,4} \end{bmatrix}}_{\mathbf{P}} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \\
\Longrightarrow s \begin{bmatrix} x \\ y \\ z \end{bmatrix} &= \begin{bmatrix} s & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & s \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} s & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & s \end{bmatrix} \begin{bmatrix} p_{1,1} & p_{1,2} & p_{1,3} & p_{1,4} \\ p_{2,1} & p_{2,2} & p_{2,3} & p_{2,4} \\ p_{3,1} & p_{3,2} & p_{3,3} & p_{3,4} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \\
&= \begin{bmatrix} sp_{1,1} & sp_{1,2} & sp_{1,3} & sp_{1,4} \\ sp_{2,1} & sp_{2,2} & sp_{2,3} & sp_{2,4} \\ sp_{3,1} & sp_{3,2} & sp_{3,3} & sp_{3,4} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}
\end{aligned} \tag{69}
$$

and obviously,

$$
\left\| \begin{bmatrix} sp_{1,1} & sp_{1,2} & sp_{1,3} & sp_{1,4} \\ sp_{2,1} & sp_{2,2} & sp_{2,3} & sp_{2,4} \\ sp_{3,1} & sp_{3,2} & sp_{3,3} & sp_{3,4} \end{bmatrix} \right\|_F = s \left\| \begin{bmatrix} p_{1,1} & p_{1,2} & p_{1,3} & p_{1,4} \\ p_{2,1} & p_{2,2} & p_{2,3} & p_{2,4} \\ p_{3,1} & p_{3,2} & p_{3,3} & p_{3,4} \end{bmatrix} \right\|_F \tag{70}
$$

scale the matrix $\mathbf{P}$ by s won't change image points

5.2.2 Rayleigh quotient's view

$$
\begin{aligned}
\hat{\mathbf{p}} &= \arg\min_{\mathbf{p}} \|\mathbf{A}\mathbf{p}\|^2 \qquad \text{s.t. } \|\mathbf{p}\|^2 = 1 \\
\Longrightarrow \mathbf{p}^* &= \arg\min_{\mathbf{p}} \left\| \mathbf{A} \frac{\mathbf{p}}{\|\mathbf{p}\|} \right\|^2 \quad \text{same as finding unconstrained } \mathbf{p} \\
&= \arg\min_{\mathbf{p}} \left(\frac{\mathbf{p}^\top \mathbf{A}^\top \mathbf{A}\mathbf{p}}{\mathbf{p}^\top \mathbf{p}} \right)
\end{aligned} \tag{71}
$$

generically, the above is a form of Rayleigh quotient:

$$
R(\mathbf{M}, \mathbf{x}) := \frac{\mathbf{x}^\top \mathbf{M}\mathbf{x}}{\mathbf{x}^\top \mathbf{x}} \qquad \text{where } \mathbf{M} \text{ is S.P.D} \tag{72}
$$

Rayleigh quotient reaches its min value:

$$
R(\mathbf{M}, \mathbf{x}_{\min}) = \lambda_{\min}(\mathbf{M}) \tag{73}
$$

smallest eigenvalue of $\mathbf{M}$, when $\mathbf{x} = \mathbf{v}_{\min}$ the corresponding eigenvector.
and

$$
R(\mathbf{M}, \mathbf{x}_{\max}) = \lambda_{\max}(\mathbf{M}) \tag{74}
$$

where have you seen this before?

5.3 Decompose further: $\mathrm{P} \rightarrow (\mathrm{R}, \mathrm{t})$

$$
\mathbf{P} = \begin{array}{c} \begin{bmatrix} p_{1,1} & p_{1,2} & p_{1,3} & | & p_{1,4} \\ p_{2,1} & p_{2,2} & p_{2,3} & | & p_{2,4} \\ p_{3,1} & p_{3,2} & p_{3,3} & | & p_{3,4} \end{bmatrix} \\ \mathbf{KR} \qquad\qquad -\mathbf{KR}\mathbf{c} \end{array} = \mathbf{K}[\mathbf{R}\ |\mathbf{t}] = \mathbf{K}[\mathbf{R}\ |\underbrace{-\mathbf{Rc}}_{\mathbf{t}}] \tag{75}
$$

where $\mathbf{c}$ is the camera center, in case you may wonder why $\mathbf{t} = -\mathbf{Rc}$:

leave out $\mathbf{K}$ for now: if we were to just transform $\begin{bmatrix}\mathbf{X} \\ 1\end{bmatrix}$ by just the **extrinsic/pose matrix** $[\mathbf{R} \quad \mathbf{t}]$:

$$
\begin{aligned}
[\mathbf{R} \quad \mathbf{t}] \begin{bmatrix}\mathbf{X} \\ 1\end{bmatrix} &= \mathbf{RX} + \mathbf{t} \\
&= [\mathbf{R} \quad -\mathbf{Rc}] \begin{bmatrix}\mathbf{X} \\ 1\end{bmatrix} \qquad \text{if we were to substitute } \mathbf{t} = -\mathbf{Rc} \\
&= \mathbf{RX} - \mathbf{Rc} \\
&= \mathbf{R}(\mathbf{X} - \mathbf{c})
\end{aligned} \tag{76}
$$

then, if we let:

$$
\mathbf{X} = \mathbf{c} \quad \Longrightarrow \quad [\mathbf{R} \quad \mathbf{t}] \begin{bmatrix}\mathbf{c} \\ 1\end{bmatrix} = \mathbf{0} \tag{77}
$$

yes, the transformation of $\mathbf{c}$ is correct!

then if we to transform a generic point $\mathbf{X}$ (defined in some "world coordinate") to the "camera coordinate" (with camera center $= \mathbf{c}$ defined by world coordinate), we need:

1. subtract $\mathbf{X}$ by $\mathbf{c}$
2. perform rotation $\mathbf{R}$

alternative is to perform rotation $\mathbf{R}$ first, and then translate by $-\mathbf{Rc}$. Both are the same!

5.4 finding t and R

knowing $\mathbf{K}$, then we can easily find $\mathbf{t}$ and $\mathbf{R}$. This is why when we calibrate extrinsic parameter, we need to specify intrinsic parameter first. OpenCV commands for finding extrinsic parameter:

References

[1] Zhengyou Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000.