I. Multiple View Geometry hw2

student: 임규범, student No.: 2018-22219

A. Problem 1

$\mathbf{x} = \mathbf{PX}$ 일 때 $\mathbf{P}^{\dagger} = \mathbf{P}^T(\mathbf{PP}^T)^{-1}$ 라고 하면 $\mathbf{PP}^{\dagger} = \mathbf{I}$ 가 되므로 $\mathbf{P}^{\dagger}$ 는 $\mathbf{P}$ 의 pseudo inverse가 된다. 따라서 $\mathbf{X} = \mathbf{P}^{\dagger}\mathbf{x}$ 가 된다.

카메라의 중심점을 $\mathbf{C}$ 라고 하면 $\underline{\mathbf{X}(\lambda) = \mathbf{P}^{\dagger}\mathbf{x} + \lambda\mathbf{C}}$ 와 같이 나타낼 수 있다. ($\because \mathbf{x} = \mathbf{PP}^{\dagger}\mathbf{x} + \lambda\mathbf{PC} = \mathbf{x}$, where $\mathbf{PC} = 0$)

B. Problem 2

(a) $H(\mathbf{PX}) = \mathbf{P}'\mathbf{X}, \forall \mathbf{X} \in \{Z = 0\}$

$\mathbf{x} = \mathbf{PX},\ \ \mathbf{x}' = \mathbf{P}'\mathbf{X}$ 라고 하면 $H(\mathbf{x}) = \mathbf{x}'$ 가 된다. 3차원 공간 상 점의 Z 값이 0이므로 $\mathbf{r}_3$ 를 제거하여 간결하게 표현할 수 있다. Projection Matrix를 자세히 풀어쓰면 다음과 같다.

$$\begin{aligned}
&\mathbf{x} = \mathbf{PX} = K\begin{bmatrix}\mathbf{r}_1 & \mathbf{r}_2 & \mathbf{t}\end{bmatrix}\mathbf{X} \\
&\mathbf{x}' = \mathbf{P}'\mathbf{X} = K\begin{bmatrix}\mathbf{r}_1' & \mathbf{r}_2' & \mathbf{t}\end{bmatrix}\mathbf{X} \\
&\mathbf{P}, \mathbf{P}' \in \mathbb{R}^{3\times 3}
\end{aligned}$$

Projection Matrix가 정방행렬이므로 determinant가 0보다 크다는 조건 아래 역행렬을 통해 두 점 사이의 관계를 나타낼 수 있다.

$$\begin{aligned}
&\mathbf{X} = \mathbf{P}^{-1}\mathbf{x} \\
&\mathbf{x}' = \mathbf{P}'\mathbf{X} = \mathbf{P}'\mathbf{P}^{-1}\mathbf{x} \\
&\underline{\therefore \mathbf{x}' = K\mathbf{P}'\mathbf{X} = K\begin{bmatrix}\mathbf{r}_1' & \mathbf{r}_2' & \mathbf{t}\end{bmatrix}\left(K\begin{bmatrix}\mathbf{r}_1 & \mathbf{r}_2 & \mathbf{t}\end{bmatrix}\right)^{-1}\mathbf{x}} \\
&\text{where} \quad \det\left(K\begin{bmatrix}\mathbf{r}_1 & \mathbf{r}_2 & \mathbf{t}\end{bmatrix}\right) > 0
\end{aligned}$$

(b) $\mathbf{X} \in \{Z \neq 0\}$ 인 경우 $H(\mathbf{PX})$ 는 $\underline{\mathbf{x}' = \mathbf{P}'\mathbf{P}^{\dagger}\mathbf{x} = \mathbf{P}'\mathbf{P}^T(\mathbf{PP}^T)^{-1}\mathbf{x}}$ 를 통해 구할 수 있다.

C. Problem 3

6.5.2 (i) $\mathbf{I}_o$ 는 $\mathbf{x} = \mathbf{PX} = K\mathbf{R}\left[\mathbf{I}| - \tilde{\mathbf{C}}\right]\mathbf{X}$ 로 나타낼 수 있다. 그리고 $\mathbf{I}' = H(\mathbf{x}) = H\mathbf{PX} = HK\mathbf{R}\left[\mathbf{I}| - \tilde{\mathbf{C}}\right]$ 로 표현할 수 있고 이 때 $H \in \mathbb{R}^{3\times 3}$ 이다.

Homography는 $\tilde{\mathbf{C}}$ 값을 변경시키지 않으므로 카메라 중심점에 위치한 초상화의 눈은 관찰자가 방 안을 돌아다니는 동안 항상 계속 자신을 쳐다보는 것처럼 보인다.

6.5.2 (ii) $\mathbf{L}^*$ 는 $\mathbb{P}^3$ 공간 상의 두 평면 P,Q의 교차선을 구함으로써 계산할 수 있다.

$$\mathbf{L}^* = PQ^T - QP^T$$

또한 다음과 같이 직선 $\mathbf{L}$ 로부터 바로 구할 수도 있다.

$$l_{12} : l_{13} : l_{14} : l_{23} : l_{42} : l_{34} = l^*_{34} : l^*_{42} : l^*_{23} : l^*_{14} : l^*_{13} : l^*_{12}$$

3차원 공산 상의 점 $\mathbf{X}$ 와 직선 $\mathbf{L}$ 을 지나는 평면은 다음과 같이 구할 수 있다.

$$\pi = \mathbf{L}^*\mathbf{X}$$

만약 $\mathbf{X}$ 가 $\mathbf{L}$ 위에 있으면 $\mathbf{L}^*\mathbf{X} = 0$ 이 된다. 이 성질을 활용하면 (ii) 문제의 식을 증명할 수 있다.

$$\begin{aligned}
\mathbf{L}^* &= \mathbf{P}^T[\mathbf{x}]_\times\mathbf{P} \\
\mathbf{L}^*\mathbf{X}(\lambda) &= \mathbf{P}^T[\mathbf{x}]_\times\mathbf{PX}(\lambda) \\
&= \mathbf{P}^T[\mathbf{x}]_\times\mathbf{P}(\mathbf{P}^{\dagger}\mathbf{x} + \lambda\mathbf{C}) \\
&= \mathbf{P}^T[\mathbf{x}]_\times\mathbf{x} + \mathbf{P}^T[\mathbf{x}]_\times\mathbf{PC} \\
&= 0
\end{aligned}$$

D. Problem 4

7.5.2 (i) $\mathbf{x} = \mathbf{P}\mathbf{X} = [\mathbf{p}_1\mathbf{p}_2\mathbf{p}_3\mathbf{p}_4]\mathbf{X}$ 를 자세하게 나타내면 다음과 같다.

$$s\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{p}_1 & \mathbf{p}_2 & \mathbf{p}_3 & \mathbf{p}_4 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

이를 전개하면 다음과 같이 s, u, v 를 계산할 수 있다.

$$\begin{aligned} s &= X_i p_{31} + Y_i p_{32} + Z_i p_{33} + 1 \\ u_i &= (X_i p_{11} + Y_i p_{12} + Z_i p_{13} + p_{14})/s \\ v_i &= (X_i p_{21} + Y_i p_{22} + Z_i p_{23} + p_{24})/s \end{aligned}$$

이를 풀어서 전개하면 다음과 같다.

$$\begin{aligned} X_i p_{11} + Y_i p_{12} + Z_i p_{13} + p_{14} - (u_i X_i p_{31} + u_i Y_i p_{32} + u_i Z_i p_{33}) = u_i \\ X_i p_{21} + Y_i p_{22} + Z_i p_{23} + p_{24} - (v_i X_i p_{31} + v_i Y_i p_{32} + v_i Z_i p_{33}) = v_i \end{aligned}$$

다음으로 위 식을 $\mathbf{AP} = 0$ 꼴로 해당 방정식을 정리하면 다음과 같다.

$$\begin{bmatrix} X_i & Y_i & Z_i & 1 & \mathbf{0}_4 & -u_i X_i & -u_i Y_i & -u_i Z_i & -u_i \\ \mathbf{0}_4 & X_i & Y_i & Z_i & 1 & -v_i X_i & -v_i Y_i & -v_i Z_i & -v_i \end{bmatrix}_{2N\times 12} \begin{bmatrix} p_{11} \\ \vdots \\ p_{33} \\ p_{34} \end{bmatrix}_{12\times 1} = 0$$

위 식은 $\mathbf{AP} = 0$ 꼴이므로 null(A)를 찾아야 한다. 만약 rank(A)=12 인 경우 null(A)=0 이다. 즉, 해가 없다. 만약 rank(A)=11 인 경우 singular value 중 하나가 0이며 이와 상응하는 eigenvector가 $\mathbf{AP}$ 의 해가 된다. 만약 rank(A)<11 인 경우 해가 무수히 많이 존재하며 degenerate case 데이터를 제거한 후 다시 캘리브레이션을 해야 한다.

$\mathbf{P}$ matrix를 분해해보면 다음과 같다.

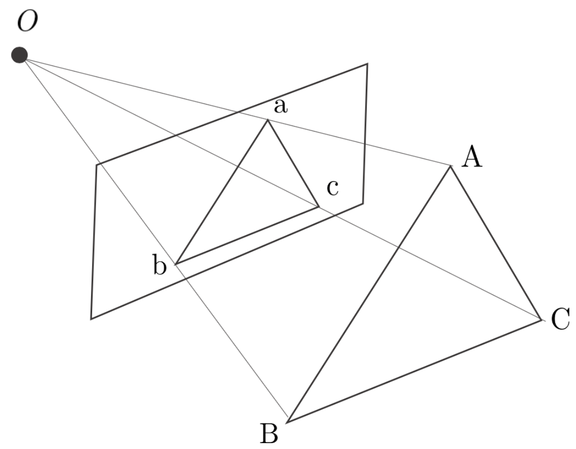
$$\begin{aligned} \mathbf{P} &= \begin{bmatrix} \alpha & 0 & u_0 & 0 \\ 0 & \beta & v_0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{r}_1^T & t_x \\ \mathbf{r}_2^T & t_y \\ \mathbf{r}_3^T & t_z \\ \mathbf{0}^T & 1 \end{bmatrix} \\ &= \begin{bmatrix} \alpha\mathbf{r}_1^T + u_0\mathbf{r}_3^T & \alpha t_x + u_0 t_z \\ \beta\mathbf{r}_2^T + v_0\mathbf{r}_3^T & \beta t_y + v_0 t_z \\ \mathbf{r}_3^T & t_z \end{bmatrix} \\ &= [\mathbf{M}| - \mathbf{MC}] = [\mathbf{M}|\mathbf{c}] \\ &= K[\mathbf{R}| - \mathbf{RC}] = K[\mathbf{R}|\mathbf{t}] \end{aligned}$$

따라서 $\mathbf{M} = K\mathbf{R}$ 이고 $\mathbf{c} = K\mathbf{t}$ 이다. 다음으로 $\mathbf{B} = \mathbf{M}\mathbf{M}^T$ 행렬을 전개해보면

$$\begin{aligned} \mathbf{B} = \mathbf{M}\mathbf{M}^T &= K\mathbf{R}\mathbf{R}^T K^T \\ &= KK^T \\ &= \begin{bmatrix} \alpha^2 + u_0^2 & u_0 v_0 & u_0 \\ u_0 v_0 & \beta^2 + v_0^2 & v_0 \\ u_0 & v_0 & 1 \end{bmatrix} \end{aligned}$$

따라서 Intrinsic 파라미터들은 다음과 같이 구할 수 있다. $u_0 = B_{13}, v_0 = B_{23}$ 이고 $\alpha = \sqrt{B_{11} - u_0^2}, \beta = \sqrt{B_{22} - v_0^2}$ 이 계산된다. 이 때, $\alpha, \beta > 0$ 이므로 음수는 고려하지 않는다.

Extrinsic 파라미터의 경우 $\mathbf{R} = K^{-1}\mathbf{M}$ 그리고 $\mathbf{t} = K^{-1}\mathbf{c}$ 를 통해 계산할 수 있다. 이 때 $\mathbf{M}$ 은 QR Decomposition을 통해 계산할 수 있는데 QR Decomposition은 유일해가 계산되지 않고 일반적으로 4개의 해가 계산된다. 이 때 Rotation Matrix 의 제약조건(SO(3) property)을 고려하여 유일해를 결정해야 한다.



(ii) P3P 알고리즘을 사용하면 세 쌍의 상응점들을 사용하여 카메라의 Projection Matrix를 계산할 수 있다. 이는 위 그림과 같이 삼각형의 일치 성질을 활용한다.

$$\Delta Oab \equiv \Delta OAB$$
$$\Delta Obc \equiv \Delta OBC$$
$$\Delta Oac \equiv \Delta OAC$$

ΔOab 와 ΔOAB 의 관계를 보면 다음과 같이 제2코사인법칙이 성립한다.

$$OA^2 + OB^2 - 2OA \cdot OB \cdot \cos(\alpha) = AB^2$$

이를 다른 삼각형에도 확장해보면 다음과 같다.

$$OA^2 + OB^2 - 2OA \cdot OB \cdot \cos(\alpha) = AB^2$$
$$OB^2 + OC^2 - 2OB \cdot OC \cdot \cos(\beta) = BC^2$$
$$OA^2 + OC^2 - 2OA \cdot OC \cdot \cos(\gamma) = AC^2$$

위 세 식을 OC^2 으로 나누고 $x = OA/OC, y = OB/OC$ 로 치환하면 다음과 같다.

$$x^2 + y^2 - 2xy\cos(\alpha) = AB^2/OC^2$$
$$y^2 + 1 - 2y\cos(\beta) = BC^2/OC^2$$
$$x^2 + 1 - 2x\cos(\gamma) = AC^2/OC^2$$

다시 $v = AB^2/OC^2, uv = BC^2/OC^2, wv = AC^2/OC^2$ 로 치환하고 정리하면

$$x^2 + y^2 - 2xy\cos(\alpha) - v = 0$$
$$y^2 + 1 - 2y\cos(\beta) - uv = 0$$
$$x^2 + 1 - 2x\cos(\gamma) - wv = 0$$

이 때, $v = x^2 + y^2 - 2xy\cos(\alpha)$ 이므로 이를 위 두,세번 째 공식에 대입하면

$$y^2 + 1 - 2y\cos(\beta) - u(x^2 + y^2 - 2xy\cos(\alpha)) = 0$$
$$x^2 + 1 - 2x\cos(\gamma) - w(x^2 + y^2 - 2xy\cos(\alpha)) = 0$$

이 되고 결국 다음과 같이 정리할 수 있다.

$$(1-u)y^2 - ux^2 - 2\cos(\beta)y + 2uxy\cos(\alpha) + 1 = 0$$
$$(1-w)y^2 - wy^2 - 2\cos(\gamma)y + 2wxy\cos(\alpha) + 1 = 0$$

위 식에서 이미 알고 있는 값들은 $\cos(\cdot), u = BC^2/AB^2, w = CA^2/AB^2$ 이 되어서 결국 x,y에 대한 이차방정식의 형태가 된다. 이 식을 풀면 일반적으로 8개의 해가 나오지만 해가 길이의 의미를 지니므로 양수들만 유효하다. 따라서 최종적으로 4개의 양의 해가 계산된다.

(iii) (a) $\tilde{\mathbf{C}}$ is known.
$[\mathbf{x}_i]_\times \mathbf{P}\mathbf{X_i} = 0$ 식에서 이미 알고 있는 값들을 제거하여 컴팩트하게 치환하면 다음과 같다.

$$\begin{aligned} 0 &= [\mathbf{x}_i]_\times \mathbf{P}\mathbf{X}_i \\ &= [\mathbf{x}_i]_\times K[\mathbf{I}|0]\begin{bmatrix}\mathbf{R} & 0 \\ 0 & 1\end{bmatrix}\begin{bmatrix}\mathbf{I} & -\tilde{\mathbf{C}} \\ 0 & 1\end{bmatrix}\mathbf{X}_i \\ &= [\mathbf{x}_i]_\times K[\mathbf{R}|0]\mathbf{X}_i' \qquad \text{where} \quad \mathbf{X}_i' = \begin{bmatrix}\mathbf{I} & -\tilde{\mathbf{C}} \\ 0 & 1\end{bmatrix}\mathbf{X}_i \\ &= \begin{bmatrix} 0 & -w_i\tilde{\mathbf{X}}_i'^T & y_i\tilde{\mathbf{X}}_i'^T \\ w_i\tilde{\mathbf{X}}_i'^T & 0 & -x_i\tilde{\mathbf{X}}_i'^T \\ -y_i\tilde{\mathbf{X}}_i'^T & x_i\tilde{\mathbf{X}}_i'^T & 0 \end{bmatrix}\begin{bmatrix}\mathbf{p}_1^T \\ \mathbf{p}_2^T \\ \mathbf{p}_3^T\end{bmatrix} \end{aligned}$$

이 때 $\mathbf{p}_i \in \mathbb{R}^3$ 이며 $\mathbf{P}$ 의 i번째 행을 의미한다. 위 식은 9 자유도를 가지므로 4.5개의 점을 통해 Projection Matrix를 계산할 수 있다.

(b) $\mathbf{v} = \det(\mathbf{M})\mathbf{m}_3$ is known.

$$\begin{aligned} 0 &= [\mathbf{x}_i]_\times \mathbf{P}\mathbf{X}_i \\ &= \begin{bmatrix} 0 & -w_i\mathbf{X}_i^T & y_i\mathbf{X}_i^T \\ w_i\mathbf{X}_i^T & 0 & -x_i\mathbf{X}_i^T \\ -y_i\mathbf{X}_i^T & x_i\mathbf{X}_i^T & 0 \end{bmatrix}\begin{bmatrix}\mathbf{p}_1^T \\ \mathbf{p}_2^T \\ \mathbf{p}_3^T\end{bmatrix} \qquad \text{where} \quad \mathbf{p}_i \in \mathbb{R}^4 \quad \text{and} \quad \mathbf{p}_3 = \begin{bmatrix}\mathbf{m}_3 & p_{34}\end{bmatrix} \\ &= \begin{bmatrix} 0 & -w_i\mathbf{X}_i^T & y_i\mathbf{X}_i^T \\ w_i\mathbf{X}_i^T & 0 & -x_i\mathbf{X}_i^T \\ -y_i\mathbf{X}_i^T & x_i\mathbf{X}_i^T & 0 \end{bmatrix}\begin{bmatrix}\mathbf{p}_1^T \\ \mathbf{p}_2^T \\ \mathbf{m}_3 \\ p_{34}\end{bmatrix} \\ &= \begin{bmatrix} 0 & -w_i\mathbf{X}_i^T & y_i \\ w_i\mathbf{X}_i^T & 0 & -x_i \\ -y_i\mathbf{X}_i^T & x_i\mathbf{X}_i^T & 0 \end{bmatrix}\begin{bmatrix}\mathbf{p}_1^T \\ \mathbf{p}_2^T \\ p_{34}\end{bmatrix} + \begin{bmatrix} y_i\tilde{\mathbf{X}}_i^T\mathbf{m}_3 \\ -x_i\tilde{\mathbf{X}}_i^T\mathbf{m}_3 \\ 0 \end{bmatrix} \end{aligned}$$

위 식은 9 자유도를 가지므로 4.5개의 대응쌍을 사용하여 Projection Matrix를 계산할 수 있다.

(c) $\tilde{\mathbf{C}}, \mathbf{v} = \det(\mathbf{M})\mathbf{m}_3$ are known.

$$\begin{aligned} 0 &= [\mathbf{x}_i]_\times \mathbf{P}\mathbf{X}_i \\ &= [\mathbf{x}_i]_\times K[\mathbf{R}|0]\mathbf{X}_i' \qquad \text{where} \quad \mathbf{X}_i' = \begin{bmatrix}\mathbf{I} & -\tilde{\mathbf{C}} \\ 0 & 1\end{bmatrix}\mathbf{X}_i \\ &= \begin{bmatrix} 0 & -w_i\tilde{\mathbf{X}}_i'^T & y_i\tilde{\mathbf{X}}_i'^T \\ w_i\tilde{\mathbf{X}}_i'^T & 0 & -x_i\tilde{\mathbf{X}}_i'^T \\ -y_i\tilde{\mathbf{X}}_i'^T & x_i\tilde{\mathbf{X}}_i'^T & 0 \end{bmatrix}\begin{bmatrix}\mathbf{p}_1^T \\ \mathbf{p}_2^T \\ \mathbf{p}_3^T\end{bmatrix} \qquad \text{where} \quad \mathbf{p}_i \in \mathbb{R}^3 \quad \text{and} \quad \mathbf{p}_3 = \mathbf{m}_3 \\ &= \begin{bmatrix} 0 & -w_i\tilde{\mathbf{X}}_i'^T \\ w_i\tilde{\mathbf{X}}_i'^T & 0 \\ -y_i\tilde{\mathbf{X}}_i'^T & x_i\tilde{\mathbf{X}}_i'^T \end{bmatrix}\begin{bmatrix}\mathbf{p}_1^T \\ \mathbf{p}_2^T\end{bmatrix} + \begin{bmatrix} y_i\tilde{\mathbf{X}}_i^T\mathbf{m}_3 \\ -x_i\tilde{\mathbf{X}}_i^T\mathbf{m}_3 \\ 0 \end{bmatrix} \end{aligned}$$

위 식은 6 자유도를 가지므로 3개의 대응쌍을 사용하여 Projection Matrix를 계산할 수 있다.

(d) $\tilde{\mathbf{C}}, \mathbf{R}$ are known.

$$
\begin{aligned}
0 &= [\mathbf{x}_i]_\times \mathbf{P}\mathbf{X}_i \\
&= [\mathbf{x}_i]_\times K\mathbf{X}_i'' \qquad \text{where} \quad \mathbf{X}_i'' = \begin{bmatrix} \mathbf{R} & -\mathbf{R}\tilde{\mathbf{C}} \\ 0 & 1 \end{bmatrix} \mathbf{X}_i \\
&= [\mathbf{x}_i]_\times \begin{bmatrix} f_x & s & x_0 \\ 0 & f_y & y_0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{X}_i'' \\
&= [\mathbf{x}_i]_\times \begin{bmatrix} \mathbf{k}_1 \\ \mathbf{k}_2 \\ \mathbf{k}_3 \end{bmatrix} \mathbf{X}_i'' \\
&= \begin{bmatrix} 0 & -w_i\mathbf{X}_i''^T & y_i\mathbf{X}_i''^T \\ w_i\mathbf{X}_i''^T & 0 & -x_i\mathbf{X}_i''^T \\ -y_i\mathbf{X}_i''^T & x_i\mathbf{X}_i''^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{k}_1^T \\ \mathbf{k}_2^T \\ \mathbf{k}_3^T \end{bmatrix} \qquad \text{where} \quad \mathbf{k}_i \in \mathbb{R}^3 \\
&= \begin{bmatrix} 0 & -w_i\mathbf{X}_i''^T & y_i\mathbf{X}_i''^T \\ w_i\mathbf{X}_i''^T & 0 & -x_i\mathbf{X}_i''^T \\ -y_i\mathbf{X}_i''^T & x_i\mathbf{X}_i''^T & 0 \end{bmatrix} \begin{bmatrix} k_{11} \\ k_{12} \\ k_{13} \\ 0 \\ k_{22} \\ k_{23} \\ 0 \\ 0 \\ 1 \end{bmatrix}
\end{aligned}
$$

위 식은 5 자유도를 가지므로 2.5개의 대응쌍을 사용하여 Projection Matrix를 계산할 수 있다.

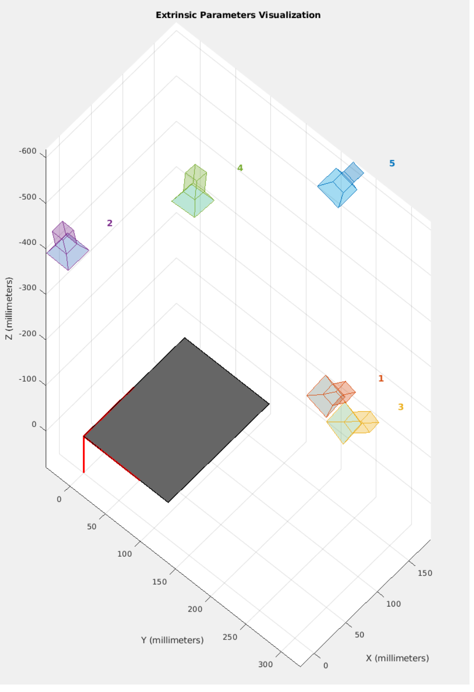
(e) $\tilde{\mathbf{C}}, \mathbf{v} = \det(\mathbf{M})\mathbf{m}_3, \mathbf{R}$ are known.

위 경우는 (d)보다 더 많은 변수를 알고 있는 경우이므로 5 이하의 자유도를 가진다. 따라서 2.5개 이하의 대응쌍을 사용하여 Projection Matrix를 계산할 수 있다.

E. *Problem 5*

실험에 사용한 사진을 40% 크기로 나타낸 그림은 아래와 같다 스마트폰은 갤럭시 노트10+를 사용하였고 체커보드는 7x9 크기의 각 셀의 간격은 20[mm]인 보드를 사용하였다.

cameraCalibrator를 통해 계산한 Extrinsic을 시각화하면 다음과 같다.



zhang_method.m 파일은 다음과 같다.

```
for i=1:5
images{i} = imread(sprintf('%d.jpg',i));
end

corners=zeros(63,2,5);

for i=1:5
[corners(:,:,i), ~] = detectCheckerboardPoints(images{i});
end

X = zeros(63,4);
a = 20; % [mm]

for j=1:9
for i=1:7
X(7*(j-1)+i,:) = [(j-1)*a (i-1)*a 0 1];
end
end

H = zeros(3,3,5);

for i=1:5
H(:,:,i) = homography2d(X, corners(:,:,i));
end

k = sym('k', [1 6]);
KtinvKinv = [k(1) k(2) k(3); k(2) k(4) k(5); k(3) k(5) k(6)];
eqs = zeros(10,6);

for i=1:5
eqs(2*i-1,:) = equationsToMatrix(transpose(H(:,1,i))*KtinvKinv*H(:,2,i)==0, k);
eqs(2*i,:) = equationsToMatrix(transpose(H(:,1,i))*KtinvKinv*H(:,1,i) - transpose(H(:,2,i))*KtinvKinv*H(:,2,i)==0,
↪ k);
end

[U,D,V] = svd(eqs);
KtinvKinv_ = subs(KtinvKinv, k, transpose(V(:,6)));
KtinvKinv_ = double(KtinvKinv_);

% cholesky decompose
pseudoKinv = chol(KtinvKinv_);
K = inv(pseudoKinv);
K = K/K(3,3);
Kinv = inv(K);

R = zeros(3,3,5);
t = zeros(3,5);

for i=1:5
r1 = Kinv * H(:,1,i);
r2 = Kinv * H(:,2,i);
r3 = cross(r1,r2);
R_ = [r1 r2 r3];
[u,d,v] = svd(R_);
R(:,:,i) = u*v'; % rotation

lambda = 1 / norm(r1); % scaling
t(:,i) = lambda * Kinv * H(:,3,i); % translation
end
```

homography2d.m 파일은 다음과 같다.

```
function H = homography2d(X1, x)

H = fitgeotrans(X1(:,1:2), x, 'projective');
H = (H.T)';
H = H / H(3,3);

end
```

이를 통해 측정한 결과는 다음과 같다. (zhang)은 직접 구현한 코드를 의미하며 (matlab)은 cameraCalibrator를 통해 측정한 결과를 의미한다.

K (zhang)			K (matlab)		
3354.284	12.401	1923.950	3331.893	0	1939.064
0	3355.181	989.724	0	3329.977	1000.860
0	0	1	0	0	1

R1 (zhang)			R2 (matlab)		
0.998	-0.060	0.008	0.998	-0.055	0.001
0.045	0.827	0.559	0.045	0.827	0.560
-0.040	-0.557	0.828	-0.032	-0.559	0.828

R2 (zhang)			R2 (matlab)		
0.995	0.035	-0.082	0.995	0.036	-0.089
-0.040	0.997	-0.059	-0.043	0.996	-0.068
0.080	0.063	0.994	0.086	0.072	0.993

R3 (zhang)			R3 (matlab)		
0.992	-0.0906	0.080	0.994	-0.081	0.067
0.010	0.726	0.687	0.0137	0.732	0.681
-0.120	-0.681	0.721	-0.104	-0.676	0.729

R4 (zhang)			R4 (matlab)		
0.986	-0.0153	0.161	0.985	-0.012	0.171
0.023	0.998	-0.047	0.022	0.998	-0.058
-0.160	0.051	0.985	-0.170	0.061	0.983

R5 (zhang)			R5 (matlab)		
0.985	-0.058	0.160	0.986	-0.051	0.155
-0.009	0.920	0.390	-0.005	0.937	0.347
-0.170	-0.386	0.906	-0.163	-0.343	0.924

t (zhang)			t (matlab)		
-69.152	-58.150	427.349	-71.230	-59.702	427.161
-27.320	-8.228	422.081	-29.158	-9.540	422.498
-47.328	-36.324	424.989	-49.457	-37.856	423.639
-92.258	-39.531	382.977	-93.934	-40.686	382.775
-64.523	5.164	662.446	-67.532	2.961	658.088