1. Given the 2D-to-3D corresponding points, calculate the **projection matrix**.
2. Based on the projection matrix, calculate the **calibration** matrix, **rotation** matrix and **translation** matrix.
3. Use projection matrix to calculate the projected 2D points from 3D points. Calculate the **average projection error**.

Note:

You need to handin the python code and the report

Your report should include:

1) Method description

2) Experimental results

3) Discussion of results

4) Problems or difficulties you have encountered

Upload your assignment to E-Course

File Format

Zip all your files into a SINGLE file

Name your file by

StudentID_hw1_version

ex： 602410143_hw1_v1

資工四 孫渝鈞 409510049

(1)Method description

計算 projection matrix: 先列出矩陣 A (其中 X, Y, Z 為 3D points x, y 為 2D points)，並計算 A^T * A，最後再找 A^T *A 矩陣的最小 eigenvalue 的對應的 eigenvactor 就是 projection matrix.

$$\begin{bmatrix} X_1 & Y_1 & Z_1 & 1 & 0 & 0 & 0 & 0 & -x_1X_1 & -x_1Y_1 & -x_1Z_1 & -x_1 \\ 0 & 0 & 0 & 0 & X_1 & Y_1 & Z_1 & 1 & -y_1X_1 & -y_1Y_1 & -y_1Z_1 & -y_1 \\ : & & & & & & : & & & & & : \\ : & & & & & & : & & & & & : \\ X_n & Y_n & Z_n & 1 & 0 & 0 & 0 & 0 & -x_nX_n & -x_nY_n & -x_nZ_n & -x_n \\ 0 & 0 & 0 & 0 & X_n & Y_n & Z_n & 1 & -y_nX_n & -y_nY_n & -y_nZ_n & -y_n \end{bmatrix}$$

A

計算 calibration matrix, rotation matrix and translation matrix: 利用 QR 分解方式，先求出 Q, R 矩陣，其中 M 矩陣為 projection matrix 的左上角 3*3 的部分。接著再利用 Q, R 矩陣去計算 K, R, T 矩陣(藍色公式)

Apply QR decomposition to M^{-1}

$M^{-1}=QR$, $M=(QR)^{-1}=R^{-1}Q^{-1}$, $K=R^{-1}$ and $R=Q^{-1}$

Then, matrix T is easy.

$P_4 = KT$

Since P_4 and K is known, $T = K^{-1}P_4$.

計算 average projection error: 利用以下公式將第一題算出來的 projection matrix 把 3D points 轉成 2D points，接著再與原本 ground truth 的 2D points 去計算 average project error

$$x_i = \frac{P_{11}X_i + P_{12}Y_i + P_{13}Z_i + P_{14}}{P_{31}X_i + P_{32}Y_i + P_{33}Z_i + P_{34}}$$

$$y_i = \frac{P_{21}X_i + P_{22}Y_i + P_{23}Z_i + P_{24}}{P_{31}X_i + P_{32}Y_i + P_{33}Z_i + P_{34}}$$

$$\text{project error} = \frac{1}{N}\sum_{i=1}^{N}\left\|x_{gt} - PX_i\right\|_2$$

(2) Experimental results

```
(base) sunyujun@sunyujundeMacBook-Pro 電腦視覺 % python hw1.py
Projection Matrix (P):
[[ 6.26520493e-01  1.36940564e-02 -3.74796008e-01 -6.04059962e-02]
 [-2.27506930e-04 -6.24059076e-01 -2.71498499e-01 -1.22952221e-03]
 [-4.86110469e-08  3.09567150e-05 -9.39823684e-04  1.50042782e-05]]
Calibration Matrix (K):
[[-0.62653939 -0.00156044 -0.37501129]
 [ 0.          0.63265885 -0.2508067 ]
 [-0.         -0.         -0.00094033]]
Rotation Matrix (R):
[[-9.99999941e-01  3.37224951e-04  6.28313818e-05]
 [-3.39110632e-04 -9.99457900e-01 -3.29209802e-02]
 [ 5.16955450e-05 -3.29209996e-02  9.99457956e-01]]
Translation Matrix (T):
[ 0.10598329 -0.00826904 -0.01595634]
Average Projection Error: 0.42708937691426857
```

(3) Discussion of results

首先 Projection matrix，我依照找 A^T * A 的最小 eigenvalue 對應的 eigenvactor 的方式算出來後，可以得到一個 3*4 的矩陣。並且我還有額外實驗用另外一種 SVD 求 projection matrix 的方式來雙重確認答案，並且發現兩種方式算出來的 projection matrix 都一樣。

```
Projection Matrix (P)/ eigenvalue method:
[[ 6.26520493e-01  1.36940564e-02 -3.74796008e-01 -6.04059962e-02]
 [-2.27506930e-04 -6.24059076e-01 -2.71498499e-01 -1.22952221e-03]
 [-4.86110469e-08  3.09567150e-05 -9.39823684e-04  1.50042782e-05]]
Projection Matrix (P)/ SVD method:
[[ 6.26520493e-01  1.36940565e-02 -3.74796008e-01 -6.04059959e-02]
 [-2.27506919e-04 -6.24059076e-01 -2.71498499e-01 -1.22952192e-03]
 [-4.86110233e-08  3.09567152e-05 -9.39823684e-04  1.50042789e-05]]
```

接下來就是 K, R, T 矩陣，為了驗證答案是否正確，我利用 P = K[R|T]的公式來驗證 K, R, T 算出來 projection matrix 是否與我第一題算出來的 projection matrix 相等，結果實驗結果是一樣的，所以判斷我求 K, R, T 的方式應該沒問題。

並且可以發現 K 矩陣是 upper triangular 沒錯，R 矩陣是 orthogonal。

```
驗證答案  P = M[R|T]
[[ 6.26520493e-01  1.36940564e-02 -3.74796008e-01 -6.04059962e-02]
 [-2.27506930e-04 -6.24059076e-01 -2.71498499e-01 -1.22952221e-03]
 [-4.86110469e-08  3.09567150e-05 -9.39823684e-04  1.50042782e-05]]
```

最後是 average project error，我求得的 error 是 0.427 左右，會有 error 而不是百分百正確的原因在於一開始求的 projection matrix 本來就沒有最佳解，我們是利用找最小 eigenvalue 對應的 eigenvactor 來求得 projection matrix 的近似最佳解，所以在 3D points 轉成 2D points 時是會存在誤差的。

(4) Problems or difficulties you have encountered

在這次作業中我所遇到的困難在於第二題求K, R, T矩陣的部分，因為一開始不太了解QR分解的方式，還很納悶三個未知數的矩陣到底要怎麼求出來，也不知道M^-1的q, r矩陣怎麼求，最後查資料發現了一個python函數np.linalg.qr可以直接求出q, r矩陣，之後就再直接把求得的q, r矩陣帶入公式求出K, R, T即可。

```python
# 使用QR分解分解M^-1矩陣
q, r = np.linalg.qr(np.linalg.inv(P[:, :3]))
```