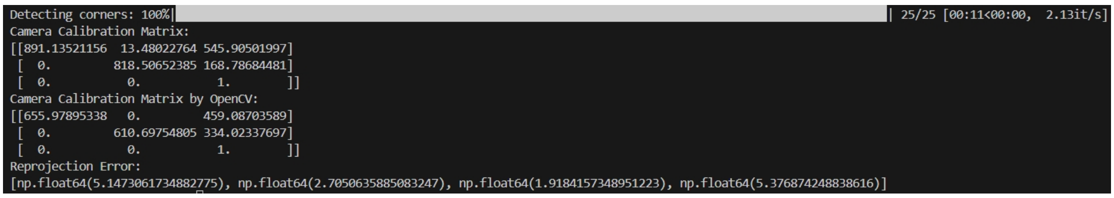
Computer Vision Homework 2 Report

涂宇清 522030910152

1. Programming Assignment

1. In `init_world_points()`, we use `cv2.findChessboardCorners()` to find the corners of the chessboard. And we use `cv2.cornerSubPix()` to refine the corner positions. We store the 3D world points and 2D image points in `world_points` and `image_points` respectively.
2. As we have learnt in class, we can get H of each picture by eigenvalue decomposition. And we can also get B from H by eigenvalue decomposition. Then, using Cholesky factorization, we can get K, which is the intrinsic matrix.
3. Then, we can use P, the camera matrix, to get reprojection error of each image.



```
Detecting corners: 100%|████████████████████| 25/25 [00:11<00:00,  2.13it/s]
Camera Calibration Matrix:
[[891.13521156  13.48022764 545.90501997]
 [  0.         818.50652385 168.78684481]
 [  0.           0.           1.        ]]
Camera Calibration Matrix by OpenCV:
[[655.97895338   0.         459.08703589]
 [  0.         610.69754805 334.02337697]
 [  0.           0.           1.        ]]
Reprojection Error:
[np.float64(5.1473061734882775), np.float64(2.7050635885083247), np.float64(1.9184157348951223), np.float64(5.376874248838616)]
```

图 **1..1 Results of camera calibration.**

The results are shown in Figure 1..1.

2. Written Assignment

2.1 Problem 1

a.

$$
\begin{aligned}
E(A,T) &= \sum_{i=1}^{N} ||Y_i - AX_i - T||_2^2 \\
&= \sum_{i=1}^{N} (Y_i - AX_i - T)^T (Y_i - AX_i - T)
\end{aligned}
$$

To minimize $E(A,T)$, we need to take the derivative of $E(A,T)$ with respect to A and T and set them to zero.

$$
\begin{aligned}
\frac{\partial E(A,T)}{\partial A} &= \sum_{i=1}^{N} 2(AX_i + T - Y_i)X_i^T = O \\
\frac{\partial E(A,T)}{\partial T} &= \sum_{i=1}^{N} 2(AX_i + T - Y_i) = O
\end{aligned}
$$

It is given by the second equation that $T = \frac{1}{N}\sum_{i=1}^{N}(Y_i - AX_i) = \hat{Y} - A\hat{X}$ Then we can substitute T with $\hat{Y} - A\hat{X}$ in the first equation and get

$$
\begin{aligned}
\sum_{i=1}^{N}(AX_i + \hat{Y} - A\hat{X} - Y_i)X_i^T = O \\
A\sum_{i=1}^{N}(X_i - \hat{X})X_i^T = \sum_{i=1}^{N}(Y_i - \hat{Y})X_i^T
\end{aligned}
$$

And we know that

$$
\begin{aligned}
\sum_{i=1}^{N}(X_i - \hat{X})(X_i - \hat{X})^T &= \sum_{i=1}^{N}(X_i - \hat{X})X_i^T - \sum_{i=1}^{N}(X_i - \hat{X})\hat{X}^T \\
&= \sum_{i=1}^{N}(X_i - \hat{X})X_i^T - \left(\sum_{i=1}^{N} X_i\hat{X}^T - N\hat{X}\hat{X}^T\right) \\
&= \sum_{i=1}^{N}(X_i - \hat{X})X_i^T
\end{aligned}
$$

and

$$
\sum_{i=1}^{N}(Y_i - \hat{Y})(X_i - \hat{X})^T = \sum_{i=1}^{N}(Y_i - \hat{Y})X_i^T
$$

So we can get

$$
\begin{aligned}
A\sum_{i=1}^{N}(X_i-\hat{X})X_i^T &= \sum_{i=1}^{N}(Y_i-\hat{Y})X_i^T \\
A\sum_{i=1}^{N}(X_i-\hat{X})(X_i-\hat{X})^T &= \sum_{i=1}^{N}(Y_i-\hat{Y})(X_i-\hat{X})^T \\
A(XX^T) &= YX^T \\
A &= (YX^T)(XX^T)^{-1}
\end{aligned}
$$

To sum up, min$(E(A,T))$ is given by $T^* = \hat{Y} - A^*\hat{X}$, $A^* = (YX^T)(XX^T)^{-1}$.

b. For every correspondence (x_i, y_i), we have

$$
\begin{aligned}
y_i &= Ax_i + T \\
\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} &= \begin{bmatrix} A_{11} & A_{12} & X_{13} \\ A_{21} & A_{22} & X_{23} \\ A_{31} & A_{32} & X_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} T_1 \\ T_2 \\ T_3 \end{bmatrix}
\end{aligned}
$$

three equations totally. And A and T have 12 unknowns. So we need at least 4 correspondences to estimate the transformation.