

# CV HW2

111061553 謝霖泳

## Problems

### 1. Camera Pose from Essential Matrix

- estimate\_initial\_RT()

```
def estimate_initial_RT(E):
    Z = np.zeros((3, 3))
    W = np.zeros((3, 3))
    Z[0][1], Z[1][0] = 1, -1
    W[0][1], W[1][0], W[2][2] = -1, 1, 1
    U, sigma, VT = np.linalg.svd(E)
    M = np.matmul(np.matmul(U, Z), U.T)
    Q1 = np.matmul(np.matmul(U, W), VT)
    Q2 = np.matmul(np.matmul(U, W.T), VT)
    R1 = np.linalg.det(Q1) * Q1
    R2 = np.linalg.det(Q2) * Q2
    T1 = U[:, 2]
    T2 = -U[:, 2]
    R1T1 = np.concatenate([R1, np.expand_dims(T1, axis=1)], axis=1)
    R1T2 = np.concatenate([R1, np.expand_dims(T2, axis=1)], axis=1)
    R2T1 = np.concatenate([R2, np.expand_dims(T1, axis=1)], axis=1)
    R2T2 = np.concatenate([R2, np.expand_dims(T2, axis=1)], axis=1)
    return np.array([R1T1, R1T2, R2T1, R2T2])
```

將 $E$ 矩陣做SVD分解為 $U$ 、 $\Sigma$ 、 $V^T$ ，便可從 $U$ 得到兩種 $T$ ， $T_1$ 和 $T_2$ 。接著根據公式計算出相應的 $M$ 以及兩種 $Q$ ，即 $Q_1$ 和 $Q_2$ 。有了兩個 $Q$ ，再根據公式 $R = \det(Q) \cdot Q$ 得到 $R_1$ 和 $R_2$ ，最後將 $R_1T_1$ 、 $R_1T_2$ 、 $R_2T_1$ 、 $R_2T_2$ 包起來return回去即為所求。

## 2. Linear 3D Points Estimation

- `linear_estimate_3d_point()`

```
def linear_estimate_3d_point(image_points, camera_matrices):
    M = deepcopy(camera_matrices)
    n = M.shape[0]
    p = deepcopy(image_points)
    mat = np.zeros((n * 2, 4))
    for i in range(0, n):
        mat[i * 2] = p[i, 1] * M[i, 2] - M[i, 1]
        mat[i * 2 + 1] = M[i, 0] - p[i, 0] * M[i, 2]
    U, sigma, VT = np.linalg.svd(mat)
    P_temp = VT[-1]
    P_temp /= P_temp[-1]
    return P_temp[:3]
```

先利用 $M$ 和 $p$ 製造出下圖左邊的矩陣，名為 `mat`，接著將 `mat` 做SVD分解得到 $U$ 、 $\Sigma$ 、 $V^T$ ，將 $V^T$ 的最後一個row同除以最後一個row的最後一個數字之後，return前三個數字即為所求。

$$\begin{bmatrix} v_1 M_1^3 - M_1^2 \\ M_1^1 - u_1 M_1^3 \\ \vdots \\ v_n M_n^3 - M_n^2 \\ M_n^1 - u_n M_n^3 \end{bmatrix} \cdot P = 0.$$

## 3. Non-Linear 3D Points Estimation

- `reprojection_error()`
- `jacobian()`
- `nonlinear_estimate_3d_point()`

## 4. Decide the Correct RT

- `estimate_RT_from_E()`

# Result