CV_HW2 Structure from Motion 107081028 常安彥

3.1 Camera Pose from Essential Matrix

$Q = UWV^T$ or UW^TV^T

$R = \det(Q) \cdot Q$

$T = \pm u_3$

```
--------------------------------------------------------------------------------
Part A: Check your matrices against the example R,T
--------------------------------------------------------------------------------
Example RT:
 [[ 0.9736 -0.0988 -0.2056  0.9994]
 [ 0.1019  0.9948  0.0045 -0.0089]
 [ 0.2041 -0.0254  0.9786  0.0331]]

Estimated RT:
 [[[ 0.98305251 -0.11787055 -0.14040758  0.99941228]
  [-0.11925737 -0.99286228 -0.00147453 -0.00886961]
  [-0.13923158  0.01819418 -0.99009269  0.03311219]]

 [[ 0.98305251 -0.11787055 -0.14040758 -0.99941228]
  [-0.11925737 -0.99286228 -0.00147453  0.00886961]
  [-0.13923158  0.01819418 -0.99009269 -0.03311219]]

 [[ 0.97364135 -0.09878708 -0.20558119  0.99941228]
  [ 0.10189204  0.99478508  0.00454512 -0.00886961]
  [ 0.2040601  -0.02537241  0.97862951  0.03311219]]

 [[ 0.97364135 -0.09878708 -0.20558119 -0.99941228]
  [ 0.10189204  0.99478508  0.00454512  0.00886961]
  [ 0.2040601  -0.02537241  0.97862951 -0.03311219]]]
--------------------------------------------------------------------------------
```

3.2 Linear 3D Points Estimation

$$\begin{bmatrix} v_1M_{13} - M_{12} \\ M_{11} - u_1M_{13} \\ \dots \\ v_nM_{n3} - M_{n2} \\ M_{n1} - u_nM_{n3} \end{bmatrix}$$

SVD solution and normalization

```
--------------------------------------------------------------------------------
Part B: Check that the difference from expected point
is near zero
--------------------------------------------------------------------------------
Difference:  0.0029243053036643873
--------------------------------------------------------------------------------
```

(near 0)

3.3 Non-Linear 3D Points Estimation

$y = M_iP$

$p_i` = 1 / y_3 [y_1 \ y_2]$

$e_i = p_i` - p_i$

$$\frac{\partial e}{\partial Pi} = \frac{[m1\cdot(m3Pi) - m3\cdot(m1Pi)]}{(m3Pi)^2}$$

$$\frac{\partial e}{\partial Pi} = \frac{[w \cdot m3 - u \cdot m1]}{w^2}$$

```
--------------------------------------------------------------------------------
Part C: Check that the difference from expected error/Jacobian
is near zero
--------------------------------------------------------------------------------
Error Difference:  8.301300130674275e-07
Jacobian Difference:  1.817115702351657e-08
--------------------------------------------------------------------------------
```

(near 0)

$P = P - (J^T J)^{-1} J^T e$

```
--------------------------------------------------------------------------------
Part D: Check that the reprojection error from nonlinear method
is lower than linear method
--------------------------------------------------------------------------------
Linear method error: 98.7354235689419
Nonlinear method error: 95.59481784846034
--------------------------------------------------------------------------------
```

(Nonlinear lower than linear)

3.4 Decide the Correct RT

First, estimate initial R and T.

Next, for every pair of R and T, calculate the corresponding 3D point.

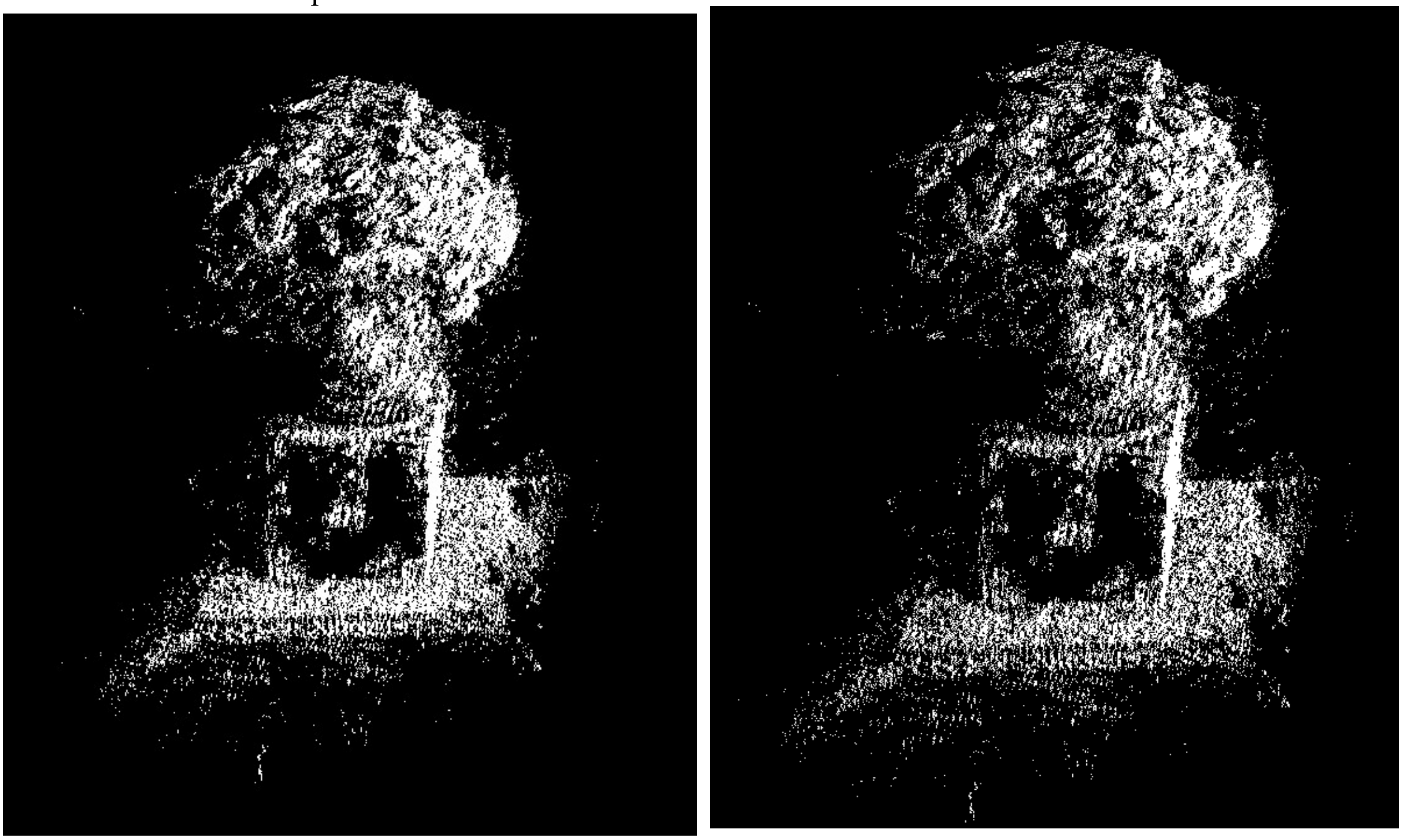
Then, transform the 3D point to another camera coordinate.

Finally, count the best pair with most positive z-coordinates.

```
--------------------------------------------------------------------------------
Part E: Check your matrix against the example R,T
--------------------------------------------------------------------------------
Example RT:
 [[ 0.9736 -0.0988 -0.2056  0.9994]
 [ 0.1019  0.9948  0.0045 -0.0089]
 [ 0.2041 -0.0254  0.9786  0.0331]]

Estimated RT:
 [[ 0.97364135 -0.09878708 -0.20558119  0.99941228]
 [ 0.10189204  0.99478508  0.00454512 -0.00886961]
 [ 0.2040601  -0.02537241  0.97862951  0.03311219]]
--------------------------------------------------------------------------------
```

3.4.2 SFM Pipeline



Left figure was done with non-linear estimation, and the right one was done with linear estimation.