ECE 5554: CV: HW-4

1.a) Essential Matrix

$$E = [t]_x R$$

t_x : Skew Symmetric matrix
R : Rotation Matrix, which is I - Identity matrix in our case

t translation is $(f, 0, f)^T$ as camera translates by f in x' and z' directions not in y'

R = I is the ~~the~~ identity matrix

$$[t]_x = \begin{bmatrix} 0 & -f & 0 \\ f & 0 & -f \\ 0 & f & 0 \end{bmatrix}$$

$$E = [t_x]$$

$$E = \begin{bmatrix} 0 & -f & 0 \\ f & 0 & -f \\ 0 & f & 0 \end{bmatrix}$$

f) Finding Epipole

Epipole in 2^{nd} image is projection of the first camera center into the second image. It lies at point e, where $Ee=0$

$(F,0,F)^T \rightarrow$ The camera translated by

epipole (e) in the second image

$e = (F, 0, F)^T$

Normalizing to image coordinates divide by last coordinate

$$e_{image} = \left(\frac{F}{F}, 0, \frac{F}{F}\right)^T = (1,0,1)^T$$

$$e_{image} = (1,0) \text{ in normalized coordinates}$$

2.a) $E = [t]_x R$

$$t = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

$$R = \begin{bmatrix} \cos 30^\circ & -\sin 30^\circ & 0 \\ \sin 30^\circ & \cos 30^\circ & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\boxed{t}_x = \begin{bmatrix} 0 & -3 & 2 \\ 3 & 0 & -1 \\ -2 & 1 & 0 \end{bmatrix}$$

b) $E = [t]_x R$

$$= \begin{bmatrix} -1.5 & -2.598 & 2 \\ 2.598 & -1.5 & -1 \\ -1.232 & 1.866 & 0 \end{bmatrix}$$

b) Rank of E:

Using (SVD) Singular Value Decomposition

$\Sigma = [3.7417, 3.7417, 0]$

Rank of E is 2 as there are two significant non-zero values.

```
[1] import numpy as np

    # Define the translation vector t
    t = np.array([1, 2, 3])

    # Define the rotation matrix R
    theta = np.radians(30)  # Convert angle to radians
    R = np.array([
        [np.cos(theta), -np.sin(theta), 0],
        [np.sin(theta), np.cos(theta), 0],
        [0, 0, 1]
    ])

    # Compute the skew-symmetric matrix [t]_x
    t_cross = np.array([
        [0, -t[2], t[1]],
        [t[2], 0, -t[0]],
        [-t[1], t[0], 0]
    ])

    # Compute the essential matrix E = [t]_x * R
    E = np.dot(t_cross, R)

    # Perform Singular Value Decomposition (SVD) to compute the rank of E
    U, S, Vt = np.linalg.svd(E)
    rank_E = np.sum(S > 1e-10)  # Count non-zero singular values

    E, S, rank_E
```

```
(array([[-1.5       , -2.59807621,  2.        ],
        [ 2.59807621, -1.5       , -1.        ],
        [-1.23205081,  1.8660254 ,  0.        ]]),
 array([3.74165739e+00, 3.74165739e+00, 1.98024198e-17]),
 2)
```