Vision-based Navigation Exercise 3

Kerem Yildirir

May 2021

1 Part 1

Consider a 3D point X and it's correspondent 2D points x_1 and x_2 in two different images and for simplicity, pinhole camera model. When we unproject the point $x1$, we get a normalized vector $\overrightarrow{x_1}$. Due to scale ambiguity, we say that this vector needs to be scaled by factor λ_1 to be equal to original 3D point X.

$$\lambda_1 \overrightarrow{x}_1 = \mathbf{X}$$

We assume that camera rotated by $\mathbf{R}$ and translated by $\mathbf{t}$ then observed x_2. Doing the same thing as above to x_2:

$$\lambda_2 \overrightarrow{\mathbf{x}}_2 = R\mathbf{X} + \mathbf{t}$$

Inserting X from above equation,

$$\lambda_2 \overrightarrow{\mathbf{x}}_2 = R\lambda_1 \overrightarrow{x_1} + \mathbf{t}$$

Cross product with t from left leads to:

$$\lambda_2 t \times \overrightarrow{x_2} = t \times R\lambda_1 \overrightarrow{x_1}$$

Since $t \times \overrightarrow{x_2}$ results in a vector perpendicular to $\overrightarrow{x_2}$, applying dot product from left makes left hand side 0.

$$x_2^T t \times \overrightarrow{x_2} = 0 = \overrightarrow{x_2}^T t \times R\lambda_1 \overrightarrow{x_1}$$

We can omit λ_1 and represent $t\times$ with the skew-symmetric matrix $\hat{T}$built from t:

$$0 = \overrightarrow{x_2}^T \hat{T} R \overrightarrow{x_1}$$

This equation is the epipolar constraint, and $E = \hat{T}R$ is the essential matrix.

2 Part 2

When we do brute force matching over all frames, we consider all pairwise combinations minus stereo pairs. For N = 2n images, this leads to:

$$\frac{2n * (2n - 1)}{2} - n$$

where n is the number of frames For N = 2000 images we need to evaluate 1998000 pairs. When BoW is used, this number is limited by number of returned query images retrieved from the database. For Q returned images, we receive at most Q candidates for each of the N images, which leads to Q*N=2*Q*n pairs. However, we do not consider same frame pairs as matches, so this reduces to number of pairs to 2*n*Q-n. We can also observe that we won't always get Q matches, due to the cases when some words are not in the database, which reduces the number of pairs even further. Overall, it can be seen that when BoW is used number of pairs to match grows linearly while when we do brute force matching, it grows quadratically.