VISNAV Exercise02

Yiğit Aras Tunalı

April 2021

1 Part 2

We have x_L and x_R, as well as the R the rotation matrix and T the translation vector between the camera reference systems. The location of the x_R in left camera's reference system can be written as $Rx_R + T$. We also know that $Rx_R + T$ and T lie in the epipolar plane, thus:

$$T \times (Rx_R + T) = T \times (Rx_R)$$

We can also use the fact that x_L lies in the epipolar plane normal to $T \times (Rx_R)$ and write:

$$x_L^T \cdot [T \times (Rx_R)] = 0$$

By writing T as skew-symmetric matrix to make cross-product a dot product we get:

$$x_L^T [T]_x R x_R$$

Where $E = [T]_x R$ is the Essential Matrix.

2 Part 4

Difference between match_bow and match_all is that match_all does a brute-force among all pairs as candidates while match_bow uses a bag of words to only match have high correspondences (in our case the score function is L1). The parameter num_bow_candidates is used to filter the amount of candidates you want to receive from the query of the DB with the smallest L1 distances.

For 2×82 images the match_all returns 13284 candidates, which is $2 \times 81 \times 82$, i.e. $2 \times (n-1) \times (n)$. So for 2×1000 images we would have $2 \times 99 \times 100$ candidates. For the match_bow we get 3649 candidates which is 41×89. I wasn't able to find any parameters that related to this number so I am not sure if there is a mathematical relation for how many candidates we get for match_bow with the number of images. It is mostly controlled with the score value we get for bow vectors and the number_of_candidates variable.