**Report: Point-lines matching by using Cross L2 Norm & Hungarian algorithm**

**June 30th, 2022**

1. **Previous works**

I proposed a SLAM method by leveraging Multi-Stage Constraint Kalman Filter (MSCKF) with a data association based on “SuperPoint – Line.” Under development modules have depicted in bellowing diagram:

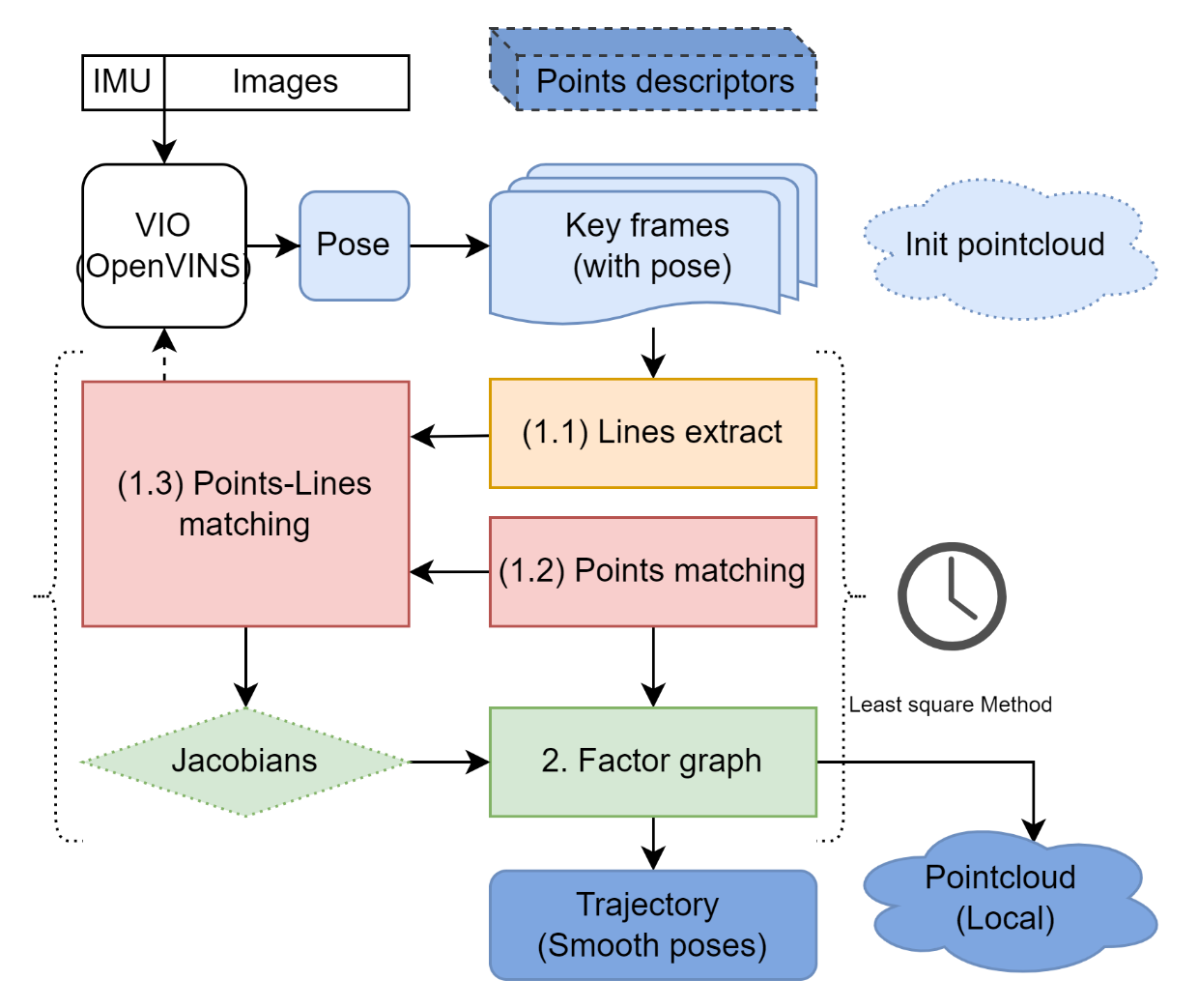


Figure 1: Proposed SLAM framework.

To give contributions, I have to deal with the following questions:

1. How to generate a *point-line graph* from image efficiency.
2. How to *track & match* (*based on* *point-line feature)* efficiency.

The main focus is data association in keyframes given by MSCKF in my previous paper. Precisely, matching methods for points and line features between two key images (source and target images). The significant transformation between the source and target images requires more sophisticated matching algorithms.

In my last report, I experimented with a point-matching method based on point descriptors. I extracted two set 2D keypoints with descriptors in the images. I call them source keypoints & target keypoints, respectively.

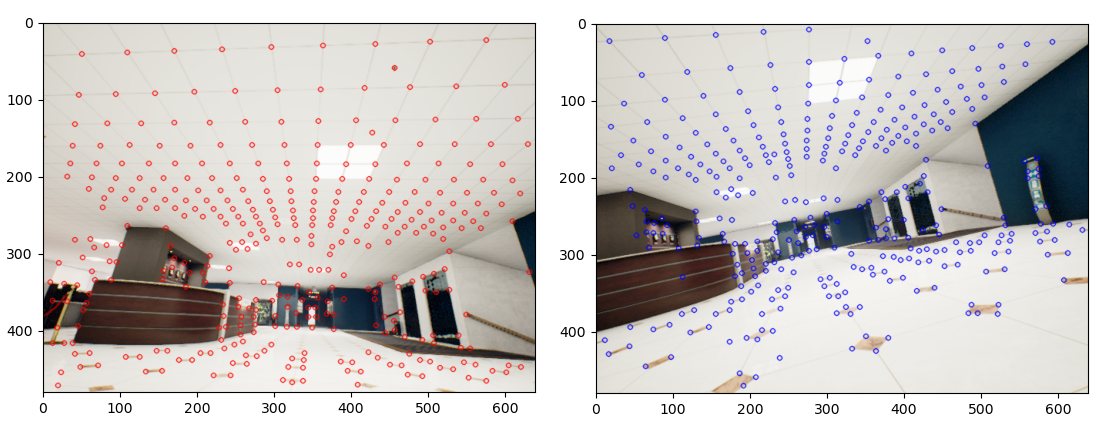


Figure 2: Keypoints in the source image (red) & target image (blue)

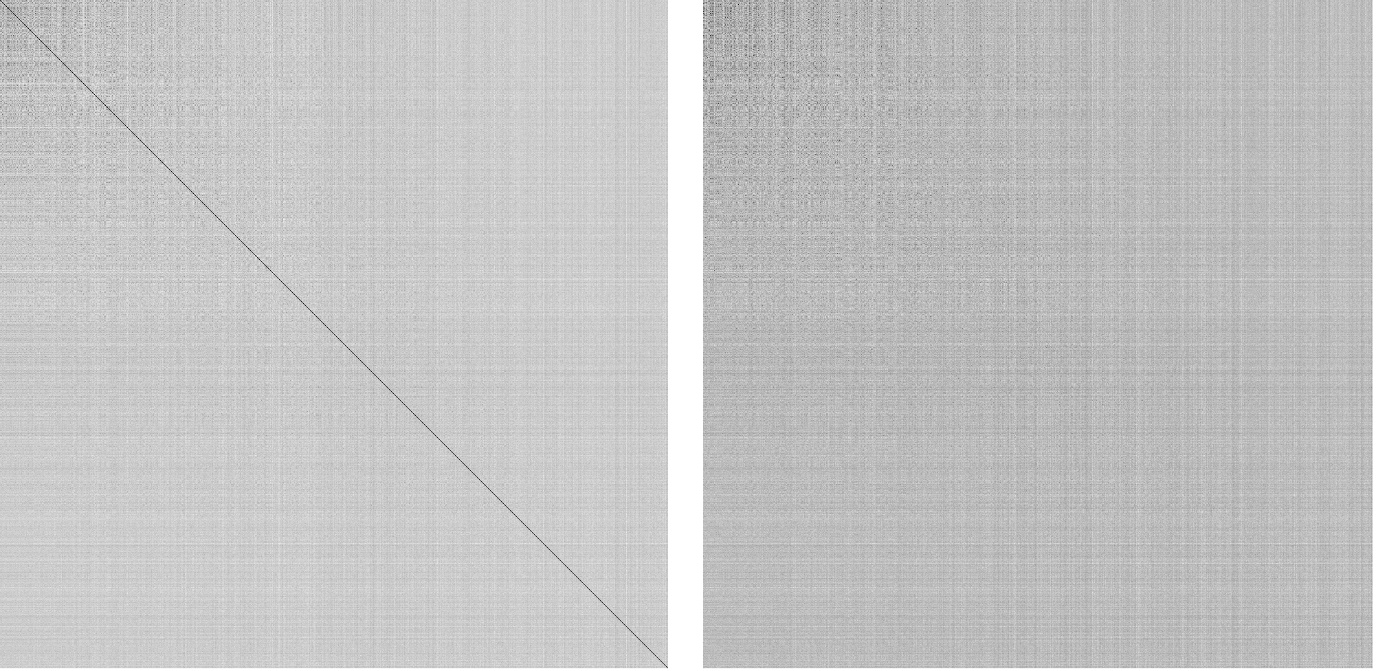


Figure 3: L2 distance matrices: Self-L2-norm matrix (left) for the source image, Cross-L2-norm matrix (Right) for two keypoint sets in the source & target image. (black is zero)

To analyze the similarity between keypoints, I calculated the L2 distance between its descriptors and represented them in two matrices, that called Self-L2-norm and Cross-L2-norm matrix. We can obtain a matching pair by selecting the minimum value in a row or using a threshold to select candidate matching pairs. My experiments showed that both methods are inefficient in a scenario where they have repetitive texture.

In this report, I continue my point-matching method by using the Hungarian algorithm. After that, I introduce my line-segment coding method with an accompanying matching algorithm.

1. **Current works**
2. **Points matching with Hungarian algorithm**

Hungarian method is an optimization algorithm that can solve the assignment problems (assign a feature point in the source image to a feature point in the target image). I used the Cross-L2-norm matrix as the input for Hungarian algorithm.

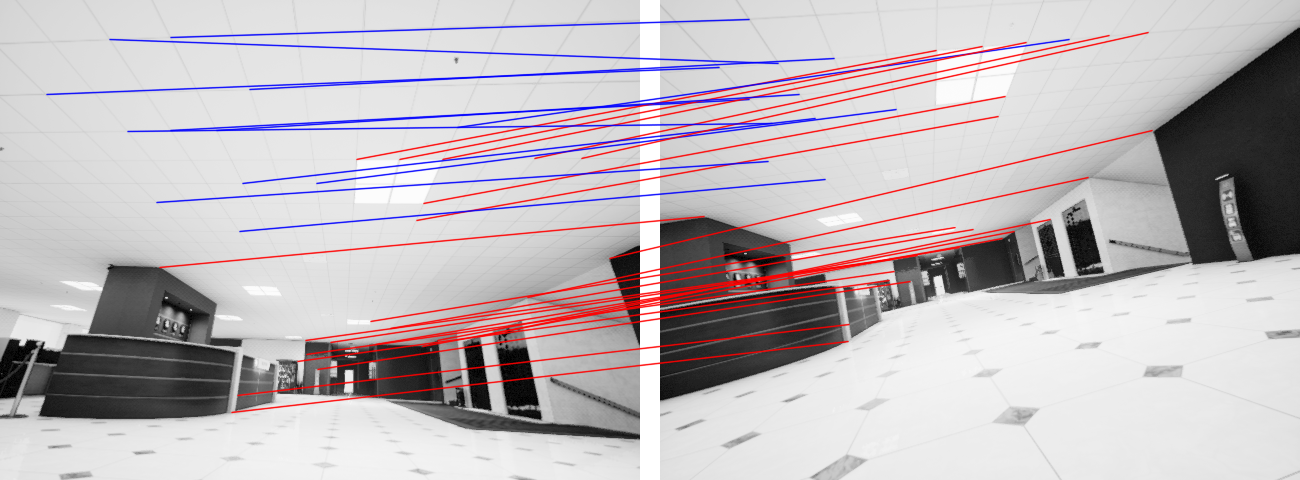


Figure 4: Matching result by using Hungarian algorithm in Cross-L2-norm matrix: red lines = accurate matching, blue lines = inaccurate matching.

The advantage of this method is not to depend on any parameters. The result is acceptable in SLAM, however, similar descriptors still corrupt the matching result, as figure 4.

In my opinion, I have to use other information (such as the Self-L2-norm matrix or predicted pose) to segment the Cross-L2-norm into smaller matrices before optimization.

1. **Line segment encoding & matching**

The most crucial part of my proposal is the line matching method. At first, I formulate the detected line by the endpoints & its neighbor features.

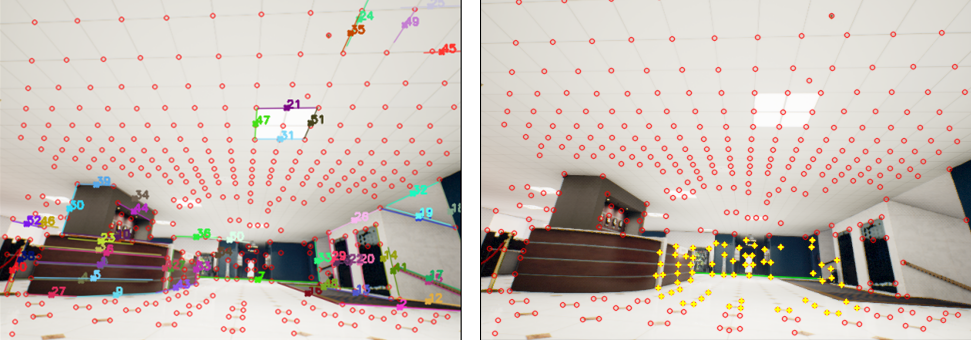


Figure 5: (a) point lines detection result. (b) line encoding method in the source image : green = line, yellow cross = endpoints + neighbors

Line features are becoming a sub-set of the source feature points so we can formulate a new smaller Cross-L2-norm. This matrix column number is unchanged because we do not drop any target feature point.

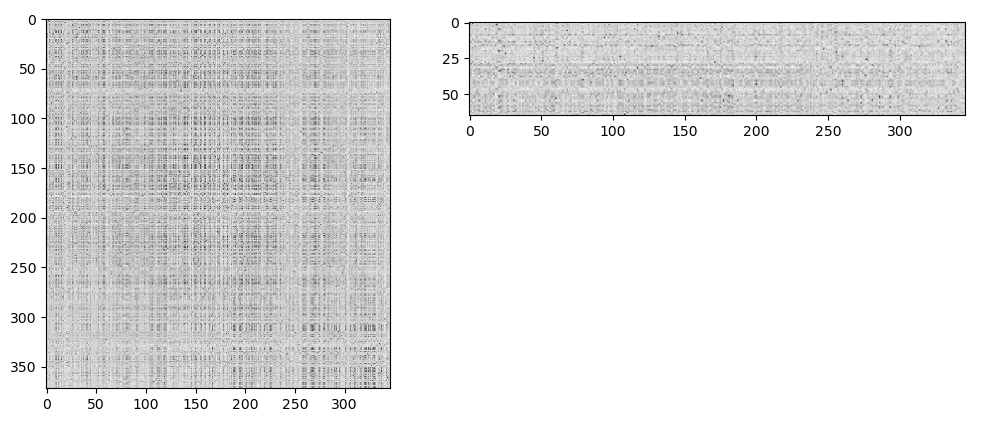


Figure 6: Full Cross-L2-norm & Sub Cross-L2-norm matrice

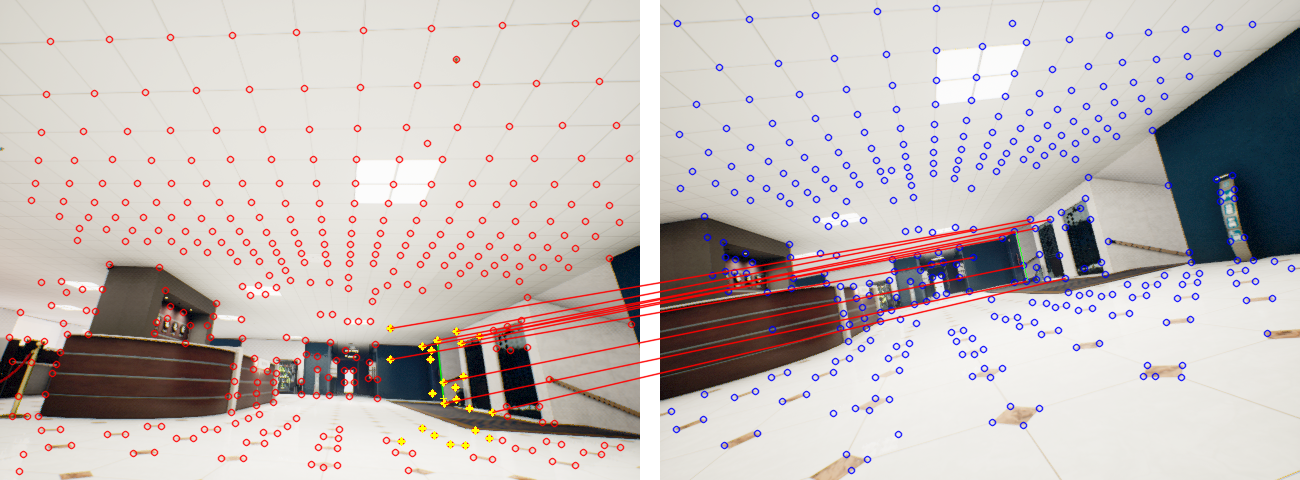
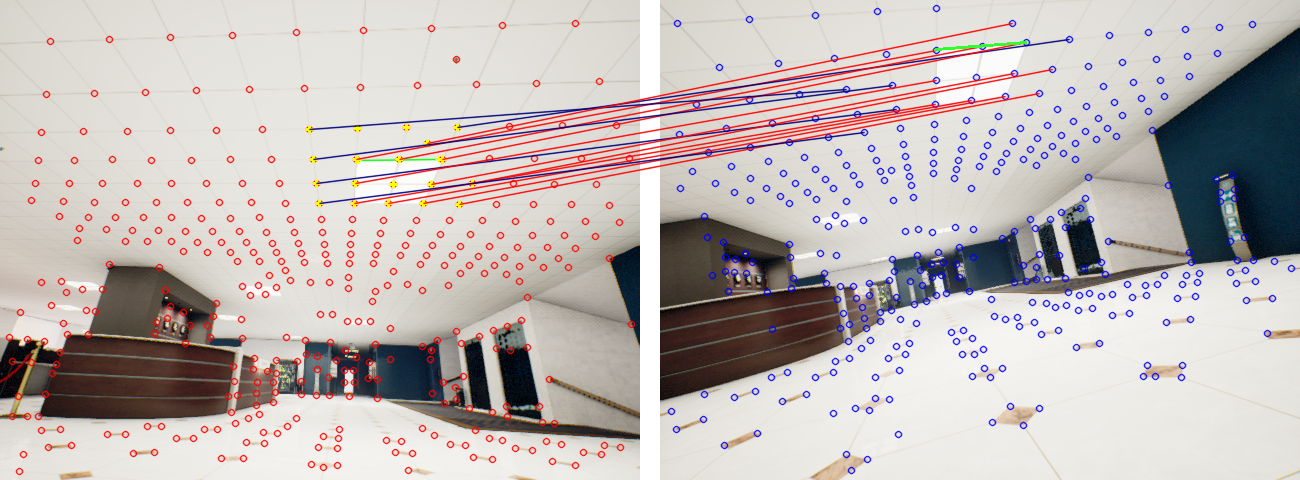
Again, I use the Hungarian method to estimate matching from coded line to target image. 

Figure 7: Single line matching result: Green line segment = detected line, Yellow cross = neighbors, RED/ BLUE lines = right/wrong matched points.

This method has some improvements to the previous way (part 2a). The illustration is shown in fig 8 and fig 9.

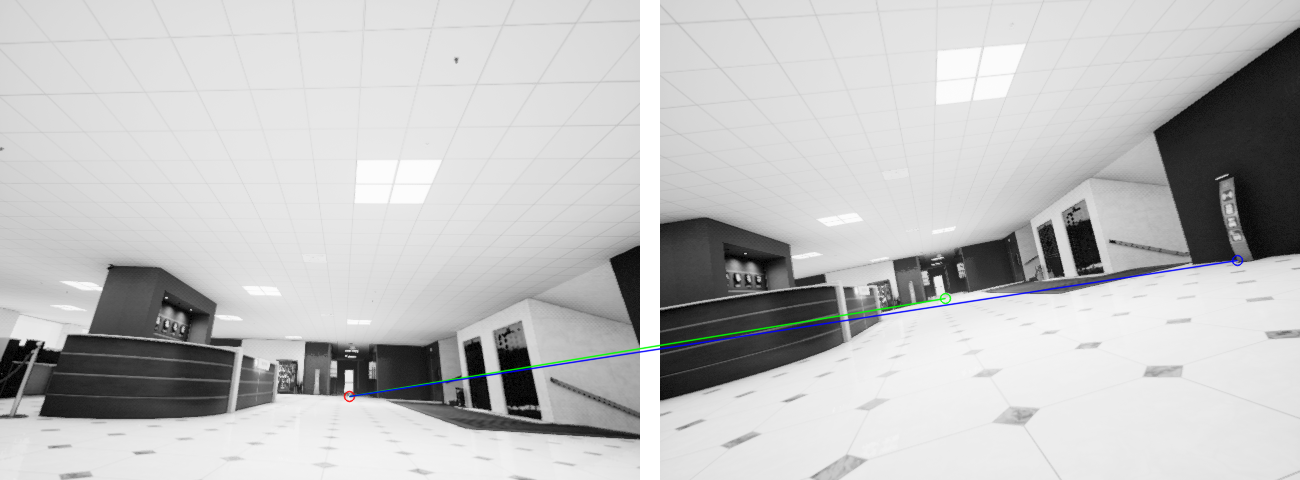


Figure 8: Single match error by the method in 2a (green line = ground-truth)

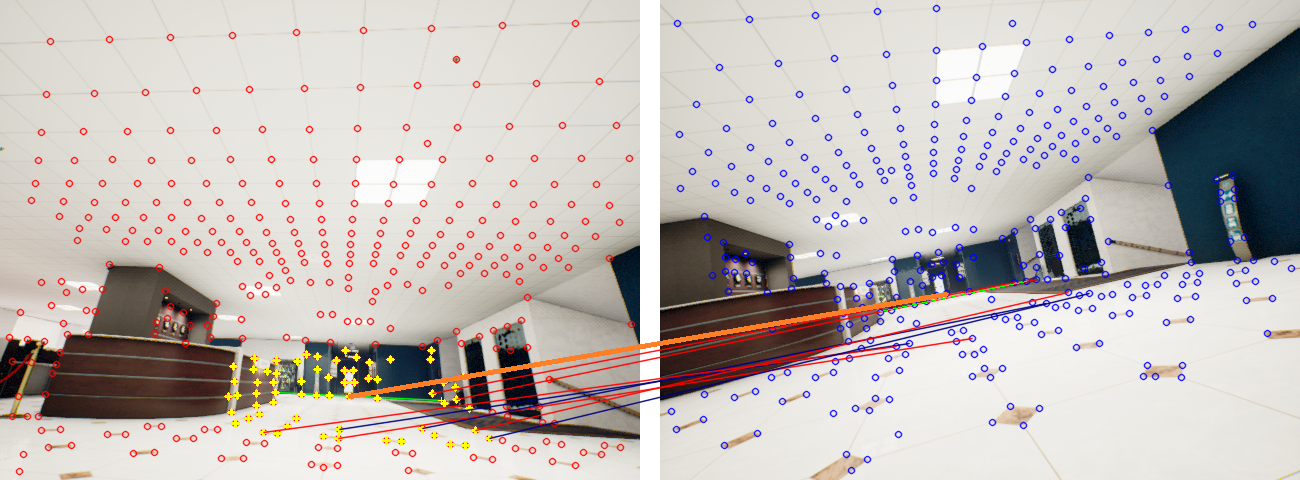


Figure 9: Improvement by the line-matching method in 2b. (Orange bold line is equal to the green line in figure 8)

1. **Future works**

Using only the Cross\_L2\_norm matrix is the weak point of both matching methods in part 2. Further improvement is using the Self\_L2\_norm matrix to improve matching. There are two metrics to consider in the line matching method: matching accuracies & success rate.

There are two other works I have to consider:

* Second, extend the point-lines association method in 3D for creating a 3D reconstruction framework.
* Third, adopting a learning-based method point-matching method is also to develop a line matching method (that also minimizes Cross\_L2\_norm too)