**Report: Matching evaluations – Dataset & VIO integration**

**July 28th, 2022**

1. **Previous works**

I proposed a semantic SLAM method using “point & lines” data association to construct a factor graph for the structure-from-motion (SFM) stage. This transition stage will provide a 3D structure before representing a semantic 3D map at the higher level. My expected high-level semantic graph will encode an individual object as a subgraph, where nodes are interest points, and edges are line segments.

Under development modules have depicted in bellowing diagram:

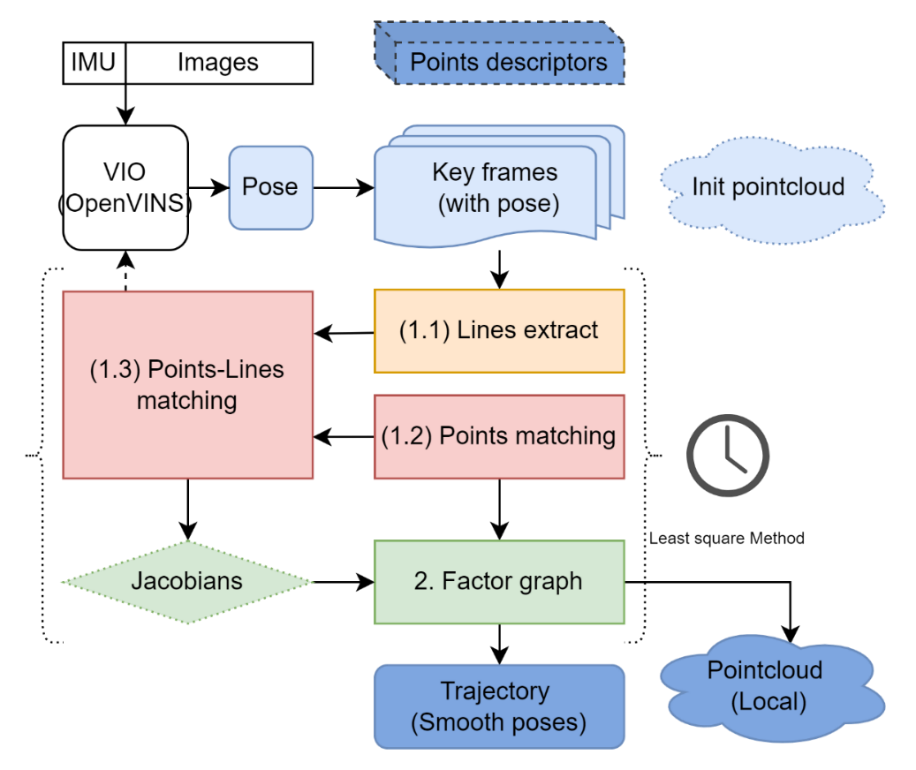


Figure 1: Proposed SLAM framework.

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

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

The main focus is data association in keyframes with approximate poses given by GPS signals or the drone’s Visual Inertial Odometry (VIO). I chose the pose from VIO measurement because this approach is the mainstream of visual SLAM research, and practically, it’s also suitable for drone inspection applications.

In my last report, I experimented with a frame-to-frame point-2D-matching method based on descriptors. The matching score is the L2 Norm distance between binary descriptor and Hungarian algorithm for maximum likelihood. I also introduced my line-segment coding method with the accompanying matching algorithm. The processes were just performed in some sample images, so I need an evaluation tool.

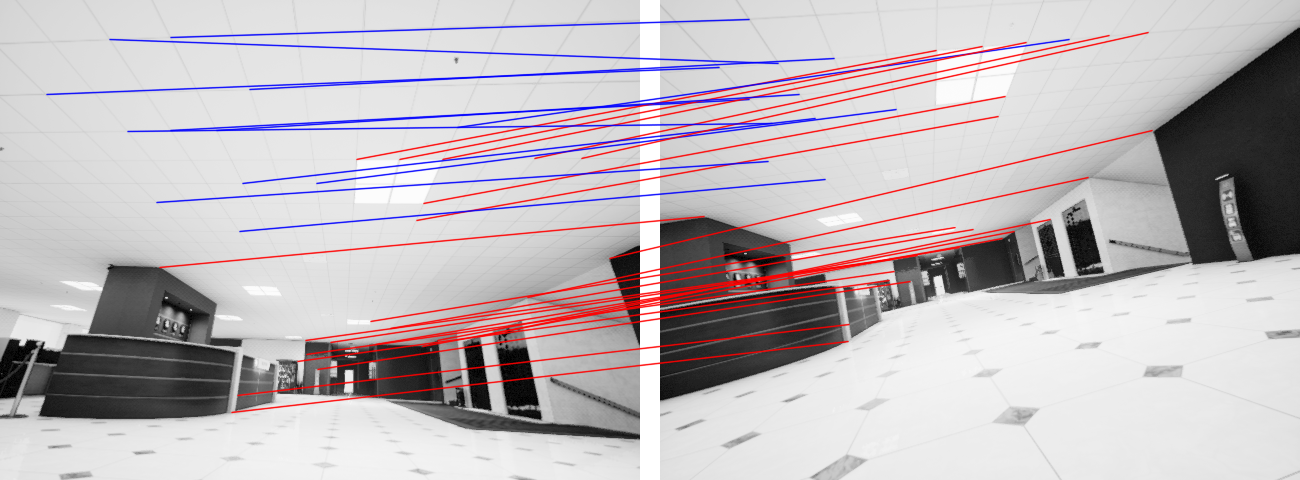


Figure 2: Matching result by using Hungarian algorithm in Cross-L2-norm matrix: red lines = accurate matching, blue lines = inaccurate matching.

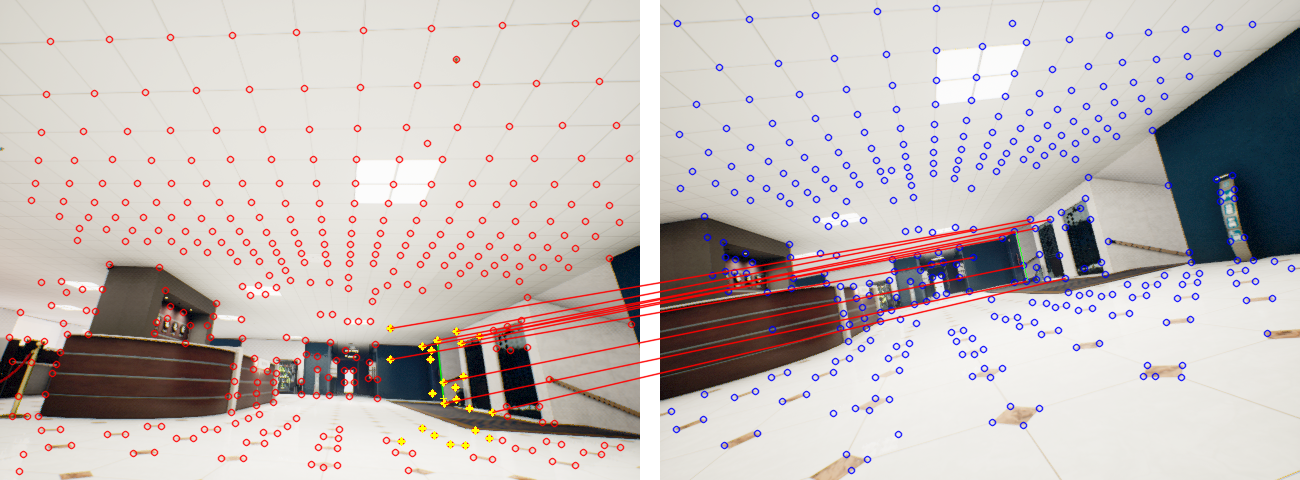


Figure 3: Line matching result: Green line segment = detected line, Yellow cross = neighbors, RED/ BLUE lines = right/wrong matched points.

I use the TartanAir dataset in this report to benchmark my semantic SLAM system because its ground truth has depth images. First, I try my automatic evaluation tool for matching; second, I generate extra IMU information to perform validation of the dataset with the Visual-inertial-odometry stage (MSCKF).

1. **Current works**
2. **Matching evaluation**

I’m using precision & recall metrics for evaluating point matching algorithms.

* Precision = TP / (TP+FP)
* Recall = TP / (TP+FN)

Precision is matching accuracy, while recall expresses the throughput of the algorithm. The recall metric is crucial because it provides various viewpoints for the landmarks, enhancing 3D estimation results.

To determine the 2D location of source points in the target frame (matching ground truth), I projected 3D source points generation to the target camera plane. Mathematically, the true positive matching would have the exact 2D location and depth distance. Quantization errors may influence this method; however, this problem has little statistical impact.

My evaluation performs under three levels based on the order distance of the source image and target image: 05-step, 10-step, and 20-step. I also compare my method to the current state of the art (SuperGlue).

Figure 4: Matching precision overtime in Office\_004 sequence

I also evaluate matching methods in different scenarios:

* Office: indoor – have repetitive landmarks
* Seaside town: outdoor – simple trajectory
* Neighborhood: outdoor – complex trajectory

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | **Hungarian** | | **SuperGlue** | |
|  |  | *Precision* | *Recall* | *Precision* | *Recall* |
| **Office  004** | *05-step:* | 73% | 77% | 94% | 79% |
| *10-step:* | 55% | 61% | 77% | 69% |
| *20-step:* | 38% | 43% | 58% | 55% |
| **Seaside town 000** | *05-step:* | 95% | 68% | 96% | 66% |
| *10-step:* | 94% | 53% | 94% | 61% |
| *20-step:* | 88% | 33% | 91% | 56% |
| **Neighborhood 000** | *05-step:* | 89% | 67% | 93% | 64% |
| *10-step:* | 82% | 52% | 88% | 58% |
| *20-step:* | 68% | 33% | 78% | 52% |

Figure 5: Evaluation in 03 sequences.

As shown in figure 5, the simple Hungarian match is still acceptable in Seaside Town and Neighborhood sequences. SuperGlue is state-of-the-art but has struggled with the Office scenario where 3D geometry constrain is more helpful than using descriptors match.

SuperGlue method is for 2D-2D matching, but SLAM also has the 3D-2D match and 3D-3D match. Its approach (Sinkhorn algorithms & Graph Neural Network) is notable but directly adopting SuperGlue is not sensible in the SLAM context.

1. **Convert TartanAir to VIO dataset**

Unlike the EuroC dataset, TartanAir does not provide IMU for VIO algorithms like MSCKF in my proposed system, so it needs extra IMU simulation to adapt to the current SLAM approach. First, I add time stamps to the camera pose ground-truth; the frame rate is the same as EUROC (20fps). Second, I use OpenVINS to interpolate the b-spline of the trajectory, then calculate velocity and acceleration.

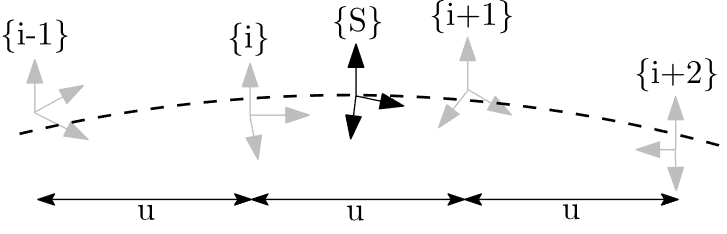


Figure 6: B-Spline interpolation: S = control point / camera pose, i = time\_stamp of IMU, u = time interval of IMU

Now, I have made TartanAir sequences to be valid with MSCKF in OpenVINS. The predicted pose of VIO is possible to improve the matching results.

1. **Future works**

The prior is to improve the matching with extra information:

* Improve 2D-2D match by using the Sinkhorn algorithm
* Perform SFM from the init pose given by MSCKF
* 3D-2D match with depth from stereo pair & previous motion