**Report: Lines representation by anchor points**

**September 8th, 2022**

1. **Previous works**

In the last meetings, I surveyed my matching method runtime in CPU/GPU and started line-matching tasks. My point-matching algorithms can work at 20-25 FPS, and it is possible to improve VIO’s state estimation.

However, point-matching is not a significant contribution to my thesis, while semantics with line features are more attractive in recent research and guarantee journal publications. In this report, I continue my work with line-matching algorithms in a keyframe:

* First: introduce my lines encoding method in a single image that formulate a line segment as a graph of inbound points & anchor points.
* Second: anchor-points matching when VIO estimation is unknown.

1. **Current works**
2. **Line encoding methods**

There are two problems related to line features:

* It’s hard to have an accurate line equation using a few views, so line equation determination should be carried out in the bundle adjustment (BA) stages. That includes local BA or global BA – depending on application or trajectory.

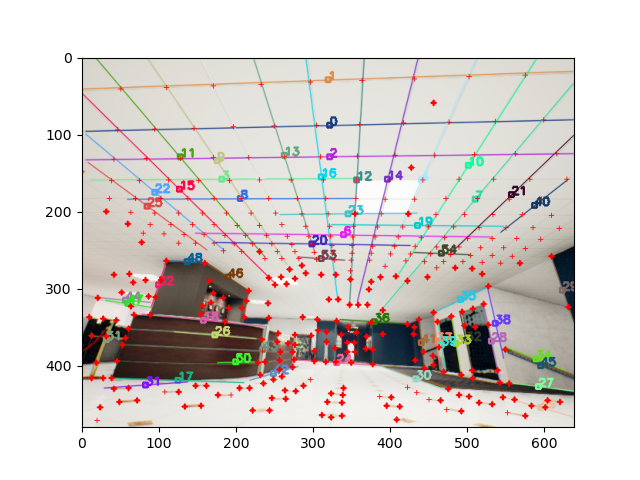


Figure 1: Point-Lines detected in a keyframe. Thin red crosses are repetitive keypoints.

* A line is an artificial object so it will have many repetitive patterns. It will make more effort to find endpoints and merge lines segment to polylines.

Because of these difficulties, I encode a line segment as a graph of inbound points and anchor points, and edges are distances between them.

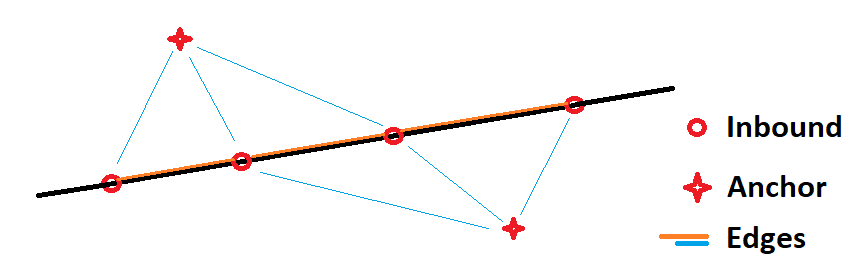


Figure 2: Visualize Line-encoding with Inbound points and Anchor points

In particular:

* Inbound points are keypoints that satisfy the 2D line equation.
* Anchor points are keypoints that do not satisfy the 2D line equation and are easily distinguished in the image. (I use self L2 Norm matrix to identify this property)
* Edges are the distance between 3D points acquired from stereo depth.

Generating lines equation in 3D needs many inbound points under various viewpoints. Incorrect inbound point matchings could be acceptable because RANSAC can reject the outlier candidates.

The line-segments merging method has two stages:

* Matching anchor points based on descriptors & Sinkhorn algorithms (matching precision by this method has the highest accuracy)
* Matching inbound points by minimizing anchor-inbound edge differences.

This encoding method makes line matching problems into two more minor issues: anchor points matching and 3D distance measurement. The point-matching was discussed in previous meetings, while 3D distance can be acquired in stereo depth or local BA.

1. **Anchor-points matching**

When VIO estimation is corrupted, my Local BA needs a robust method to determine anchor points and good matching.

This situation is like trying the Sinkhorn-3D method without prior poses by VIO. my Sinkhorn-Anchor-3D matching method has three steps to solve this problem:

* Find anchor points in the source and target image by the Sinkhorn-Knopp algorithm. If the number of anchor points is too small, the algorithm is stopped.
* Estimate 3D transform between source & target (stereo) image by Levenberg-Marquardt optimization and RANSAC
* Generate cost matrix to find matching points like Sinkhorn-3D.

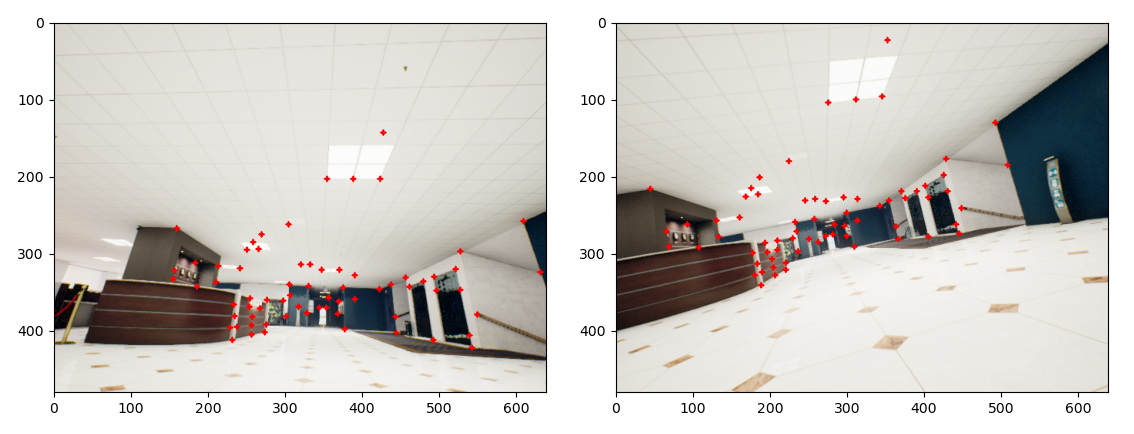


Figure 3: Detected anchor points in source and target image in the first step

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Distance | Precision | Recall | F1 |
| Sinkhorn | *05-step:* | **96.14%** | 34.77% | 51.07% |
| *10-step:* | **93.92%** | 25.07% | 39.57% |
| *20-step:* | **89.86%** | 15.58% | 26.56% |
| Sinkhorn-3D (VIO) | *05-step:* | 88.62% | 75.98% | 81.24% |
| *10-step:* | **81.74%** | **65.51%** | **71.61%** |
| *20-step:* | **70.67%** | 48.14% | **55.25%** |
| Superglue | *05-step:* | **93.80%** | **79.27%** | **85.92%** |
| *10-step:* | **77.02%** | **68.71%** | **72.62%** |
| *20-step:* | 58.19% | **55.22%** | **56.67%** |
| Sinkhorn-Anchor-3D | *05-step:* | 87.32% | 76.92% | 81.14% |
| *10-step:* | 76.75% | 66.69% | 70.64% |
| *20-step:* | 62.43% | 49.67% | 53.73% |

Figure 4: Comparing to previous methods

Following the above table, Sinkhorn-Anchor-3D has the same F1 score as Sinkhorn-3D (using VIO pose) in the short distance, but it’s getting worse in long distances. The matching precision of anchor points only is the same as the Sinkhorn method. This method helps the SLAM system become more independent of VIO and the foundation of fail-safe algorithms in SLAM.

1. **Future works**

Because of the current line encoding method related to anchor points & edge distances, line matching & merging become two problems: anchor point matching & accuracy of local BA.

So that there are two works I need to focus on:

1. Generate lines’ ground truth in the office dataset
2. Line matching & merging based on the proposed encoding method
3. Lines reconstruction and comparing them to the generated ground truth