**Report: Matching using Sinkhorn & 3D stereo constraints**

**August 11th, 2022**

1. **Previous works**

In my two latest reports, I experimented with a frame-to-frame point-2D-matching method based on descriptors. The matching score is the L2 Norm distance between binary descriptors and the Hungarian algorithm for maximum likelihood.

I developed an automatic point matching evaluation tool for the TartanAir dataset; then, I compared my result to the SuperGlue (2020) method (point matching state-of-the-art) in three different scenarios. SuperGlue gave better results than my current method, but it does not work well in the Office scenario of the TartanAir dataset, which has many line segments and repetitive patterns.

*(I also introduced my line-segment coding method with the accompanying matching algorithm. However, I want to focus on improving my points matching algorithm first because I did not find any good baseline for the lines-matching task.)*

In this report, I introduce two improvement methods:

* First: using Sinkhorn algorithms to improve matching precision.
* Second: using estimated pose (by MSCKF) and 3D stereo improves matching recall.

1. **Current works**
2. **Matching using Sinkhorn algorithms evaluation**

In the Office sequence in the TartanAir dataset, similar patterns will give similar descriptors so that L2 Norm distance values between them are too small to distinguish.



Figure 1: Example for the source image and target image in Office sequence

Applying the Hungarian method to the Cross matrix will give low precision because it is no longer a suitable cost matrix in this situation. This problem can be visualized in the Cross L2-Norm matrix and 02 Self L2-Norm matrices (for the source and target images).

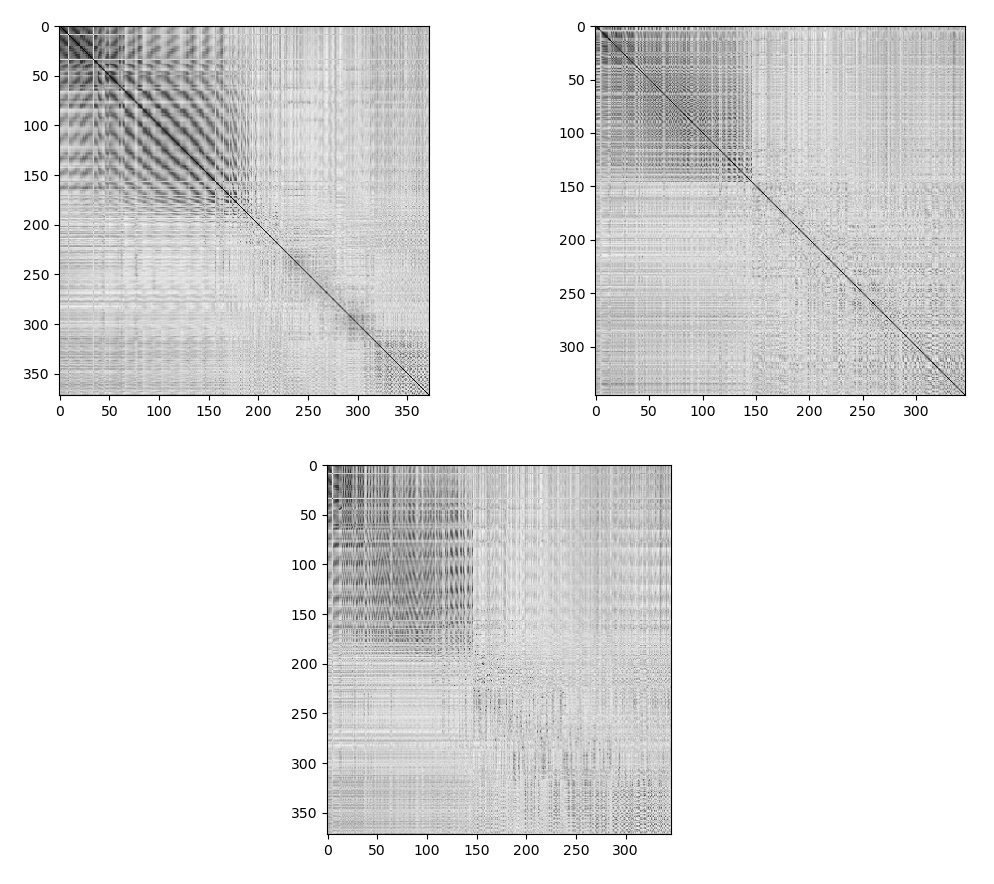


Figure 2: Two Self L2-Norm matrices of keypoints in source/target image (top) and the Cross L2-Norm matrix between 2 keypoints sets.

I use the Sinkhorn-Knopp solver to generate a regularized cost matrix for Hungarian algorithms. The histograms (for Sinkhorn-Knopp) are extracted from Self L2-norm matrices by the following equation:

His = mean (L2) – 1 (1)

The input cost matrix for Sinkhorn is still Cross L2, the regulation term is 0.1, and the number of iterative is 20. A new optimal matrix (Gs) is used to formulate a new cost matrix (C):

C = 1 – Gs (2)

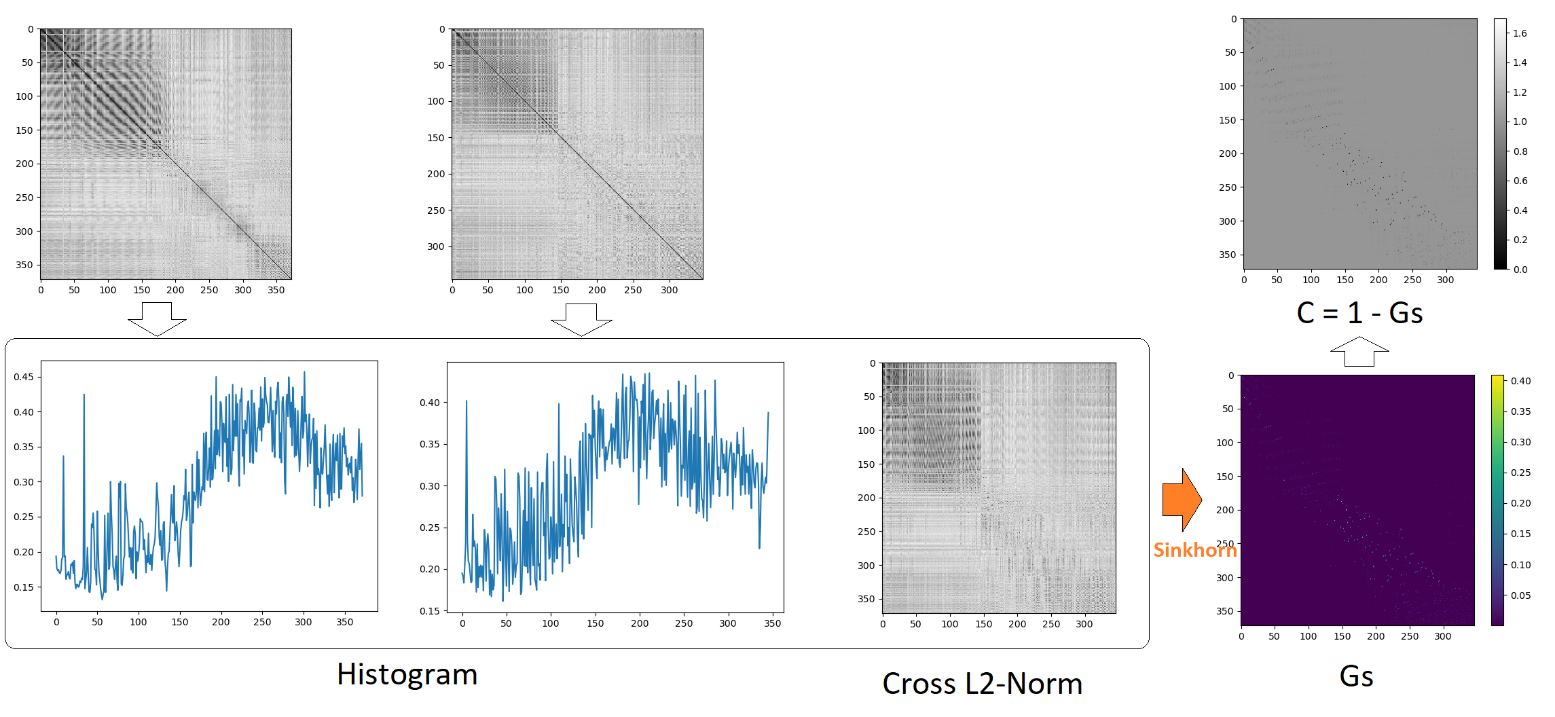


Figure 3: Using the Sinkhorn algorithm to generate the new cost matrix C

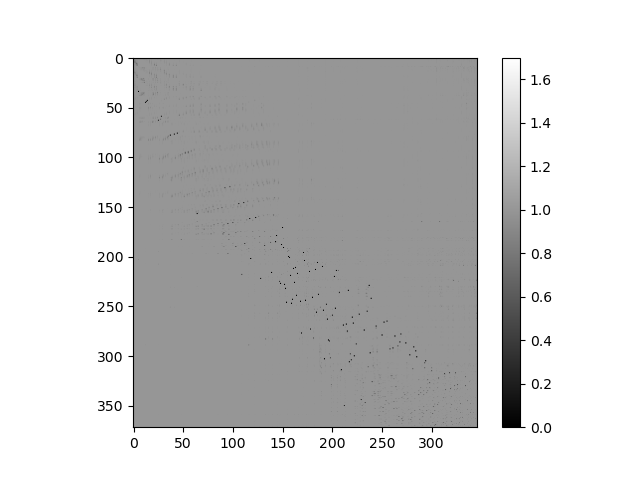


Figure 4: Final cost matrix for Hungarian algorithm

The Hungarian method optimizes the cost matrix (C) and gives the final matching result.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Distance | Precision | Recall | F1 |
| Hungarian | *05-step:* | 0.728445648 | 0.769618991 | 0.74846651 |
| *10-step:* | 0.546011455 | 0.608357436 | 0.57550083 |
| *20-step:* | 0.376334234 | 0.42755992 | 0.40031498 |
| Sinkhorn | *05-step:* | **0.961448381** | 0.347707498 | 0.51071506 |
| *10-step:* | **0.939177775** | 0.25065063 | 0.39569656 |
| *20-step:* | **0.898606474** | 0.155797659 | 0.26555432 |
| Superglue (S.O.T.A) | *05-step:* | **0.937973349** | **0.79267253** | **0.8592234** |
| *10-step:* | **0.770154347** | **0.68707055** | **0.7262439** |
| *20-step:* | **0.581928661** | **0.55220678** | **0.5666783** |

Figure 5: Results in Office scenario.

My Sinkhorn-based method has improved matching precision in the Office scenario; however, the recall and the F1 score are much lower than other methods. The iterative regulation process in the Sinkhorn algorithm has blurred the responses of repetitive keypoints in the final cost matrix.

1. **3D Stereo constraint**

The distance between keypoints can be represented as constraints in the cost matrix C. The condition for generating this constraint is the stereo camera and estimated poses given by MSCKF. In the previous meeting, I mentioned the simulation for MSCKF (by OpenVINS) in the TartanAir dataset. At the same time, depth from the stereo is provided by the block-matching method.

By using the estimated transform (given by MSCKF), we projected source keypoints to the target image to have expected 2D points. Geometry distance (g) between source keypoint and target keypoint can express by bellowing equation:

(3)

Where is the 2D location of source keypoints in the target image view, **p1** is the 2D location of target keypoint, and **d** is distance tolerance (I chose **d=64** in the Office scenario). The geometry distance matrix G formulated new cost matrices for Hungarian matching:

CHungarian-3D = Cross\_L2 + G (4a)

CSinkhorn-3D = (1 – Gs) + G (4b)

So, we have:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Distance | Precision | Recall | F1 |
| Hungarian | *05-step:* | 0.7284456 | 0.769619 | 0.748467 |
| *10-step:* | 0.5460115 | 0.6083574 | 0.575501 |
| *20-step:* | 0.3763342 | 0.4275599 | 0.400315 |
| Hungarian-3D | *05-step:* | 0.8385701 | **0.823137** | **0.83078** |
| *10-step:* | 0.7012934 | **0.714638** | 0.7079 |
| *20-step:* | 0.5302445 | **0.541864** | 0.53599 |
| Sinkhorn | *05-step:* | **0.961448** | 0.3477075 | 0.510715 |
| *10-step:* | **0.939178** | 0.2506506 | 0.395697 |
| *20-step:* | **0.898606** | 0.1557977 | 0.265554 |
| Sinkhorn-3D | *05-step:* | 0.8862197 | 0.7597507 | 0.81242 |
| *10-step:* | **0.817389** | **0.655074** | **0.71614** |
| *20-step:* | **0.706716** | 0.4813608 | **0.55247** |
| Superglue | *05-step:* | **0.937973** | **0.792673** | **0.85922** |
| *10-step:* | **0.770154** | **0.687071** | **0.72624** |
| *20-step:* | 0.581929 | **0.552207** | **0.56668** |

Figure 6: Comparison methods in Office scenario: best = highlighted green, second best = highlighted black

The geometry constraint significantly improves matching recall, especially in short-distance (05-step ~ 10-step). In large distance (20-step) Hungarian-3D & Sinkhorn-3D is comparable (F1 scores 53.5% and 55.2% respectively). SuperGlue is still the best F1 score (56.7%) but has lower precision than Sinkhorn 3D.

There are differences between indoor (office) and outdoor (neighborhood) data. It’s sensible because stereo depth with a 25cm baseline is insufficient.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Distance | Precision | Recall | F1 |
| Hungarian | *05-step:* | 0.8939709 | 0.6659299 | **0.76328** |
| *10-step:* | 0.8217074 | 0.5245622 | 0.640342 |
| *20-step:* | 0.6831667 | 0.3337097 | 0.448392 |
| Hungarian-3D | *05-step:* | 0.839973 | **0.718186** | **0.77217** |
| *10-step:* | 0.768358 | **0.6218** | **0.68153** |
| *20-step:* | 0.653362 | **0.469105** | **0.53427** |
| Sinkhorn | *05-step:* | **0.95645** | 0.3979353 | 0.562034 |
| *10-step:* | **0.949718** | 0.2451893 | 0.389755 |
| *20-step:* | **0.918732** | 0.1333736 | 0.232932 |
| Sinkhorn-3D | *05-step:* | 0.8789597 | **0.69355** | **0.7713** |
| *10-step:* | 0.8358405 | **0.594191** | **0.68507** |
| *20-step:* | **0.767121** | 0.4417392 | **0.5422** |
| Superglue | *05-step:* | **0.934941** | **0.640739** | **0.76037** |
| *10-step:* | **0.881966** | 0.5837558 | **0.70252** |
| *20-step:* | **0.780125** | **0.51625** | **0.62133** |

Figure 7: Comparation methods in Neighborhood scenario: best = highlighted green, second best = highlighted black, worse (w.r.t: poor depth) = highlighted red

1. **Future works**

The prior is to develop line matching algorithms:

* Sophisticated line encoding method (based on Sinkhorn & 3D stereo)
* Line 2D matching (Local)
* Line 2D to 3D matching (Global)
* Find a baseline method