



特徴点マッチングの改良と実験の準備

M2 田川幸汰

1 前回までの結果と概要

以下に前回までの特徴点マッチングの結果（図1）と実行時間（表1）、自己位置推定結果（図2）を示す。

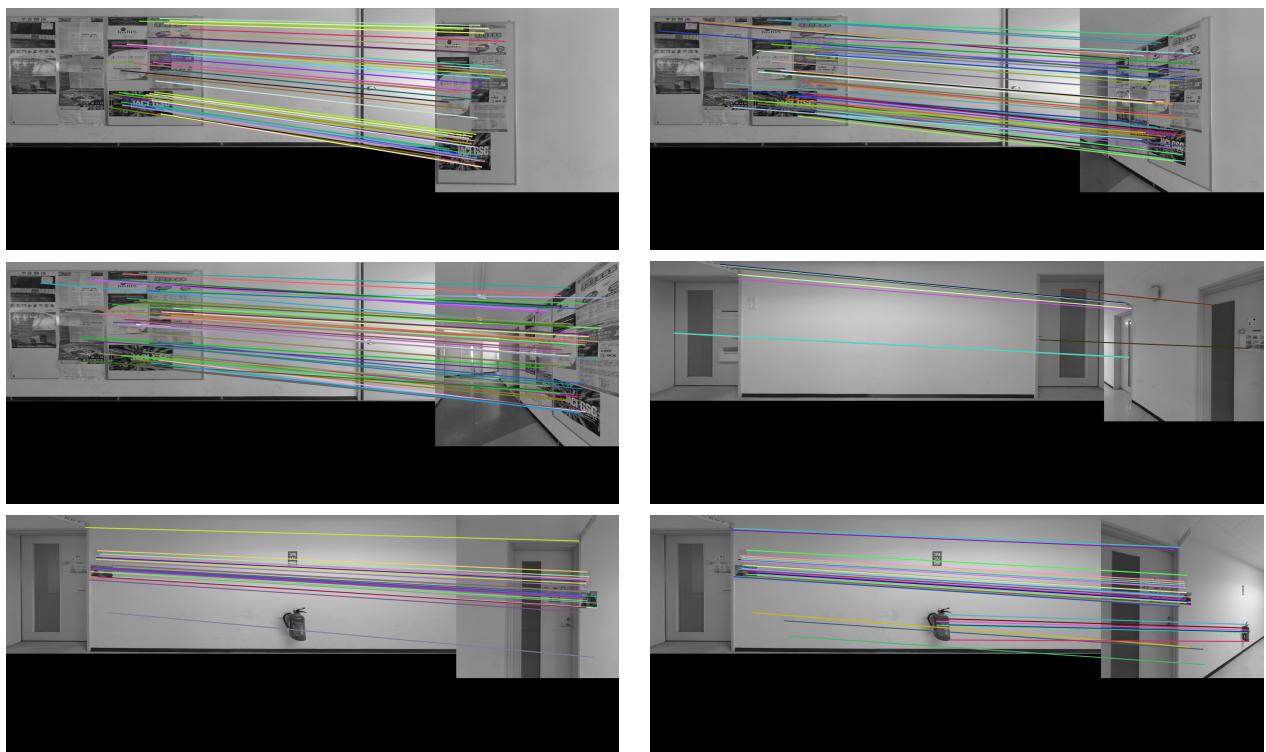


図1: SuperPoint, SuperGlueによる特徴点マッチング結果

表1: 実行時間

	特徴点の事前検出(秒)	特徴点検出(秒)	特徴点マッチング(秒)	自己位置推定(秒)
SuperPoint,SuperGlue	389.025	0.134	5.516	0.141



図 2: 自己位置推定結果

機械学習ベースの特徴点マッチング手法である SuperPoint [1] および SuperGlue [2] に対して、現在位置に基づく特徴点マッチングなどの修正を行うことで、マッチング精度および実行時間を改善した。さらに、自己位置推定の結果として一部のカメラ位置が壁の裏側に推定される問題が発生していたが、これについても幾何的な制約を導入することで、カメラ位置が正しく壁の表側に推定されるように改善した。

2 現在位置に基づく特徴点マッチング

表 1 より、処理全体における実行時間のボトルネックは特徴点マッチング処理であることが明らかとなった。従来は、入力された画像に対して毎回すべてのテクスチャ画像と特徴点マッチングを実施していて、処理時間が大幅に増加していた。この問題を解消するために、以下の 2 点の改善を行った。

1. 初回のみ全テクスチャとのマッチングを実施

最初の 1 枚の入力画像に対しては、従来通り全テクスチャ画像と特徴点マッチングを実施する。

2 回目以降は近傍テクスチャに限定

2 回目以降の入力画像については、1 つ前の画像から得られた自己位置推定結果（現在位置）を用いて、位置的に近いテクスチャ画像を 5 件選定し、それらとのみ特徴点マッチングを実施する。

また、この処理変更に伴い、各テクスチャ画像に対応する重心位置をあらかじめファイルに記述・保存する仕組みを追加した。

2.1 特徴点マッチングの修正

精度と実行時間の改良のための主な修正項目を以下に示す。



1. 特徴点検出およびマッチングのパラメータ調整

SuperPointにおける最大検出数(`max_keypoints`)を1500から500に削減し、処理時間を短縮した。加えて、SuperGlueにおけるSinkhorn反復回数(`sinkhorn_iterations`)を10から15に増やし、マッチング精度を向上させた。また、マッチングのしきい値(`match_threshold`)を0.4から0.3に変更することで、より信頼性の高いマッチのみを採用するようにした。

2. 距離スコアに基づくマッチ候補の絞り込み

SuperGlueによって得られたマッチのうち、スコアが高い上位100件のみを選択し、RANSACによる外れ値除去処理に入力するよう変更した。これにより、RANSAC処理における計算量を削減した。

3. 特徴点検出およびマッチングの際の解像度変更

入力画像およびデータベース画像の解像度を柔軟に変更できるようにした。SuperPointおよびSuperGlueは、低解像度でも十分に特徴点検出およびマッチングを行うことが可能である。

4. 不要処理の削減と前処理の導入

マッチング結果の画像描画処理は、最良マッチが更新された場合に限って実施するよう変更した。また、特徴点の正規化を事前にを行うことで、マッチングモデルへの入力サイズを削減した。

3 幾何的な制約による自己位置推定結果の修正

一部の入力画像において、カメラ位置が壁の裏側に推定される問題が発生していた。この原因を調査するため、12個の初期値から求めた自己位置推定結果のうち、目的関数のスコアが高い上位3件の解を分析した(図3)。その結果、上位2件はいずれもスコアが同一であり、カメラ位置は壁の裏側に投影され、さらにカメラ姿勢が天地反転していることが確認された。

```
===== self-localization for input\half\input_-90.png =====
input\half\input_-90.pngの特徴点検出時間: 0.066秒
input\half\input_-90.pngの特徴点マッチング時間: 3.981秒
最良マッチ数: 660
== ナンバーワンの推定結果 ==
誤差 J_best: 4.622894410195802e-06
R_best:
[[ -0.02094977 -0.9997724  0.00403082]
 [ 0.0016813   0.00399646  0.9999906 ]
 [-0.99977912  0.02095636  0.0015972 ]]
C_best: [-1.02943697  6.98679868  1.29191141]

== ナンバー2の推定結果 ==
誤差 J_2nd: 4.622894410195802e-06
R_2nd:
[[ -0.02094977  0.9997724 -0.00403082]
 [ 0.0016813   -0.00399646 -0.9999906 ]
 [-0.99977912 -0.02095636 -0.0015972 ]]
C_2nd: [1.02943697  6.98679868  1.29191141]
```

図3: カメラ位置

カメラ位置が壁の裏側に誤って推定される問題に対処するため、カメラ座標系のY軸が常に下向きであるという幾何学的制約を導入した。具体的には、カメラ姿勢行列Rの要素 $R[1, 2]$ が負($R[1, 2] < 0$)であることを条件とし、これはカメラのY軸が世界座標系のZ軸負方向と一致していることを意味する。この判定処理を加えることで、壁の裏側と判定された解を候補から除外することが可能となった。

3.1 特徴点検出およびマッチング結果

特徴点検出およびマッチングはSuperPoint、SuperGlueの公式実装([3])を参考にした。実行環境と、SuperPoint及びSuperGlueのパラメータを表2に示す。また、入力画像とデータベース画像の解像度は前回の0.5倍としている。



表2: 実行環境及びハイパーパラメータ

GPU	NVIDIA GeForce RTX 4060 Ti, NVIDIA driver version=560.94, CUDA version=12.6
SuperPoint	nms_radius=4, keypoint_threshold=0.005, max_keypoints=500
SuperGlue	weights='outdoor', sinkhorn_iterations=15, match_threshold=0.3

実行結果を図4に示す。前回までの結果（図1）と比較して、右中央の画像が正しく推定されていることが確認できる。これは、距離スコアによるマッチング候補の絞り込みにより、スコアの低いマッチを除外できたことが要因と考えられる。また、その他の画像についても解像度を変更してもなお精度の高いマッチングが実現されている。

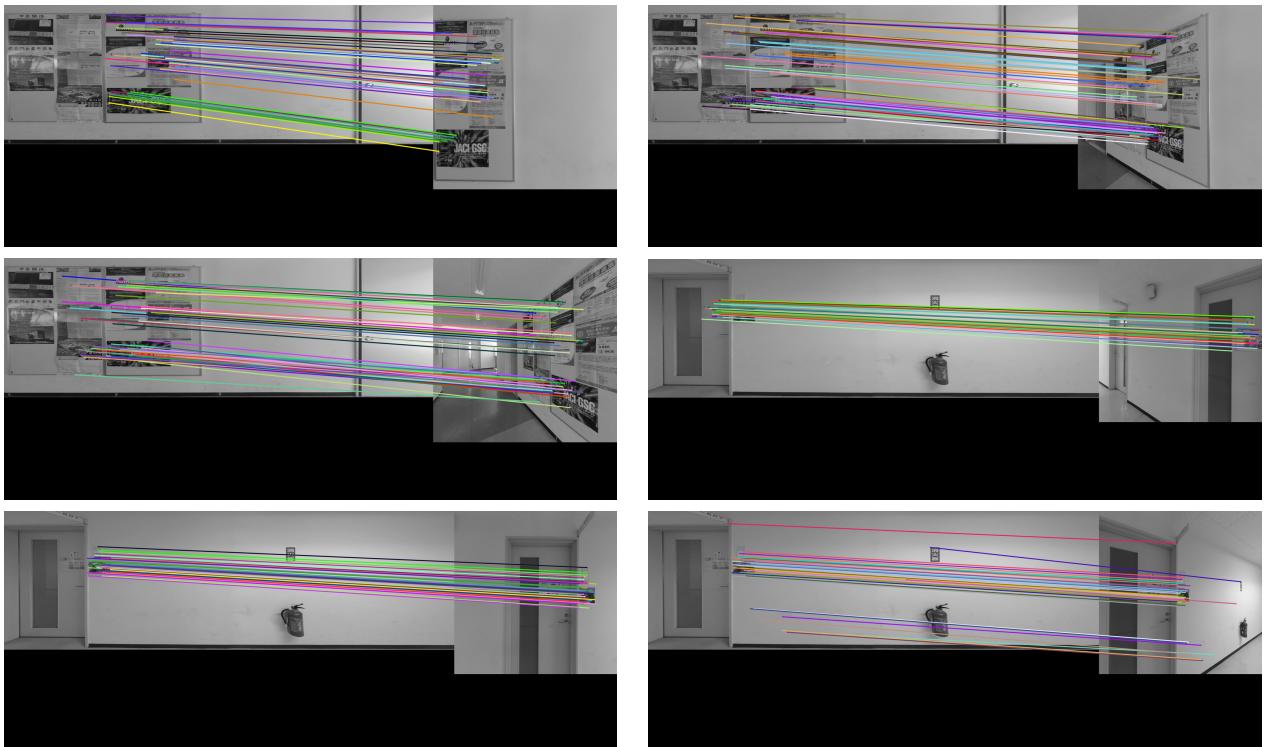


図4: SuperPoint, SuperGlueによるマッチング結果

また、特徴点マッチングの結果を用いた自己位置推定の結果を図5に示す。前回までの結果（図2）と比較して、壁の裏側の推定結果を除外し、正しい姿勢が描画されていることが確認できる。



図 5: 自己位置推定結果の改良版

実行時間を表3に示す。前回までの結果(表1)と比較して、特徴点の検出およびマッチングに要する時間を大幅に短縮できた。自己位置推定に関しては、現状では12個の初期値に対して逐次的に推定を行っているため、初期値の数を削減するか、あるいは初期値を用いない手法に置き換えることで、さらなる処理時間の短縮が期待される。

表3: 実行時間

SuperPoint, SuperGlue	特徴点の事前検出(秒)	特徴点検出(秒)	特徴点マッチング初回(秒)
	12.716	0.061	1.459
	特徴点マッチング2回目(秒)	自己位置推定(秒)	
	0.128	0.332	

4 今後の計画

今後の研究計画を以下に示す。

1. 8月5日(19時から20時): テクスチャの再撮影
2. 8月前半: テクスチャ補完した3次元モデルを生成、自己位置推定結果確認
3. 8月後半: 自己位置推定機能を応用した実用的なアプリケーションのデモ開発
 - 目的地までのルートを提示するリアルタイムナビゲーションシステムを構築(動画入力)
4. 9月以降: 線特徴の活用方法を検討



参考文献

- [1] D. DeTone, T. Malisiewicz, and A. Rabinovich, “Superpoint: Self-supervised interest point detection and description,” *arXiv preprint arXiv:1712.07629*, 2018.
- [2] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Geiger, “Superglue: Learning feature matching with graph neural networks,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4938–4947, 2020.
- [3] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, “Supergluepretrainednetwork: Superglue: Learning feature matching with graph neural networks.” <https://github.com/magicleap/SuperGluePretrainedNetwork>, 2020. Accessed: 2025-07-09.