



テクスチャ補完した3次元モデルの生成と自己位置推定

M2 田川幸汰

1 前回までの結果

図1に入力画像とデータベース画像とのマッチング結果を示す。左図は機械学習ベースの手法を用いた結果であり、特徴点検出にはSuperPoint [1]、マッチングにはSuperGlue [2]を適用した。一方、右図は従来手法を用いた結果であり、特徴点検出にはSIFT、マッチングにはFLANNを適用している。

今回の例では、どちらの手法を用いた場合でも正しくマッチングを行うことができた。ただし、一般的には機械学習を用いた手法の方が安定して特徴点マッチングを行うことができる。

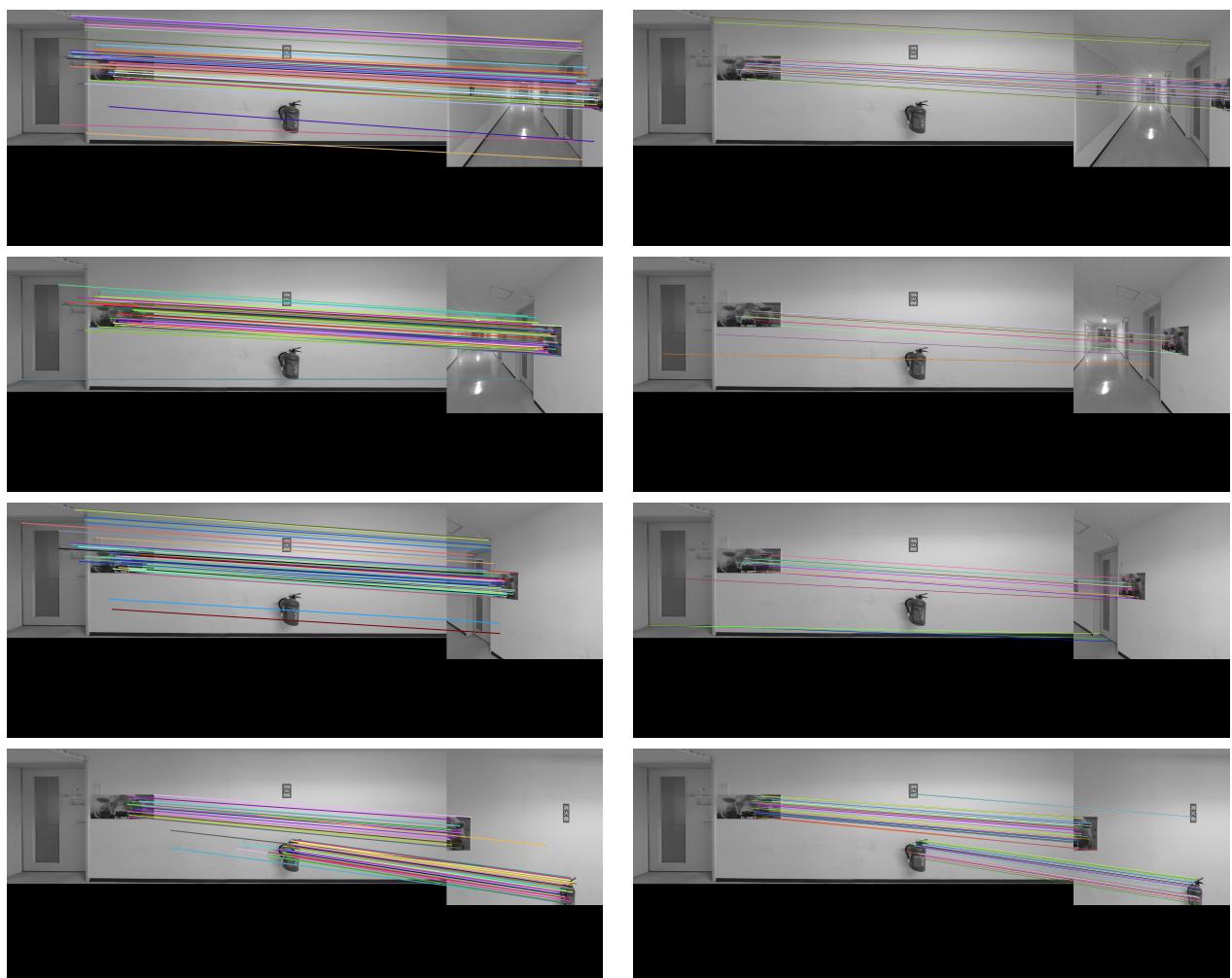


図1: SuperPoint, SuperGlueによるマッチング結果

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??に示す。左図が機械学習ベースの手法による結果、右図が従来手法による結果である。



図 2: 自己位置推定の結果

実行時間を表??に示す。特徴点マッチングは、初回はすべてのデータベース画像とマッチングをしているが、2回目以降は近傍のテクスチャとのみマッチングをしている。自己位置推定を行う際に12個の初期値に対してマッチングを行うため、少し時間がかかってしまっている。

表 1: 実行時間

	特徴点の事前検出(秒)	特徴点検出(秒)	特徴点マッチング初回(秒)
SuperPoint,SuperGlue	12.451	0.033	1.593
	特徴点マッチング2回目(秒)	自己位置推定(秒)	
	0.156	0.739	
SIFT	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	

2 概要

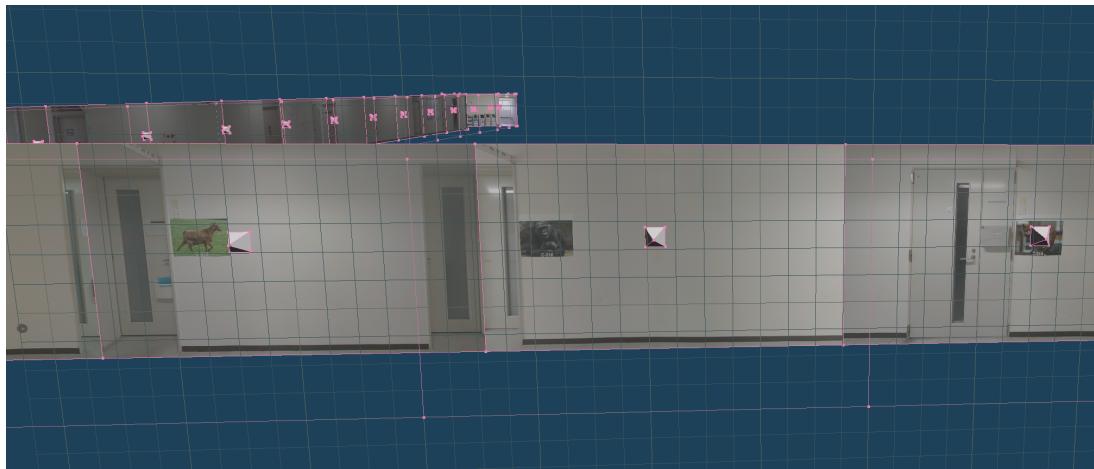
前回までの実験において課題となっていた「テクスチャの特徴不足によるマッチングおよび自己位置推定の困難さ」を改善するため、特徴の乏しい面にポスターを貼付し、より豊富なテクスチャ情報を有する3次元モデルを生成した。その3次元モデルを用いて特徴点マッチングおよび自己位置推定を実施した。さらに、今回は新たな試みとして、動画から切り出したフレームを入力画像として利用し、各フレームで得られた自己位置推定結果を連結することで、推定過程を動画として出力した。加えて、実行時間や精度の向上を目的として、特徴点マッチングや自己位置推定を含む一連の処理システムに対し、いくつかの改良を加えた点についても報告する。

3 テクスチャ補完

従来の三次元モデルでは、壁面や扉付近など模様が乏しい領域において特徴点が十分に検出されず、類似したパターンや平滑なテクスチャにより正確なマッチングが困難であった。この問題を改善するため、特徴点の少ない壁面にA2サ



イズのポスターを貼付し、人工的にテクスチャ情報を追加することで、より特徴量の多い三次元モデルを生成した。生成結果を図??に示す。



ポスターに使用した画像は Wikimedia Commons [3] より取得し、C棟5階の各部屋の扉の隣に貼付した。これにより、従来は特徴点が得られなかった領域でも安定したマッチングが可能となることを期待する。

4 システムの改良：動画入力への対応

従来は静止画を入力として自己位置推定を行っていたが、本研究では新たに動画入力に対応した。具体的には、動画から任意のフレームレートでフレームを切り出し、それぞれのフレームに対してデータベース画像との特徴点マッチングを行い、自己位置を推定する。さらに、各フレームで得られた自己位置推定結果を連結し、動画として出力する機能を追加した。なお、自己位置推定に失敗したフレームについては、処理を打ち切り、元の自己位置推定結果をそのまま保存する仕様とした。この改良は、将来的に道案内などリアルタイム処理を想定し、逐次的に自己位置推定結果を返す必要があることを考慮している。

加えて、動画入力に対応するにあたり、実行時間および精度を維持・向上させるために以下の修正を加えた。

1. 特徴点マッチング結果の評価

特徴点マッチ数が10未満の場合は、誤マッチの可能性が高いため処理を終了するよう変更した。

2. 近傍テクスチャとのマッチング

一定時間以上連続して自己位置推定に失敗した場合は現在位置をリセットし、再度すべてのデータベース画像とのマッチングを行うようにした。

3. 自己位置推定結果の評価

推定結果が前回の結果から大きく乖離している場合は誤マッチと判断し、処理を終了するよう変更した。

4. 自己位置推定手法の変更

初期位置を仮定しない大域最適化ベースのカメラ姿勢推定手法を導入し、実行時間の大幅な短縮を図った。

5 自己位置推定結果

入力動画は [video_30.mp4](#) に示す。動画から切り出すフレームレートは 2FPS である。

テクスチャ補完を行った三次元モデルの面と、動画から切り出した各フレームに対して特徴点マッチングを行った結果の一部を図 3 に示す。ここに示したフレームはいずれも正しく特徴点マッチングが行えたものである。

マッチングおよび自己位置推定に成功したフレーム数は、機械学習ベース手法が 44/84、従来手法 (SIFT+FLANN) が 9/84 であった。機械学習ベースの手法では、テクスチャ補完のために貼付したポスターが写り込んでいるフレームについて、ほぼ全てで正しくマッチングを行うことができた。一方、従来手法ではカメラ近傍に写っている場合のみマッ



チングが成立し、それ以外のフレームではマッチ数不足や誤った対応付けが生じる結果となった。さらに、一定時間以上連續して自己位置推定に失敗した場合に、すべてのデータベース画像との再マッチングを行う仕様としたことで、全体の処理時間に大きな差が生じる要因となった。

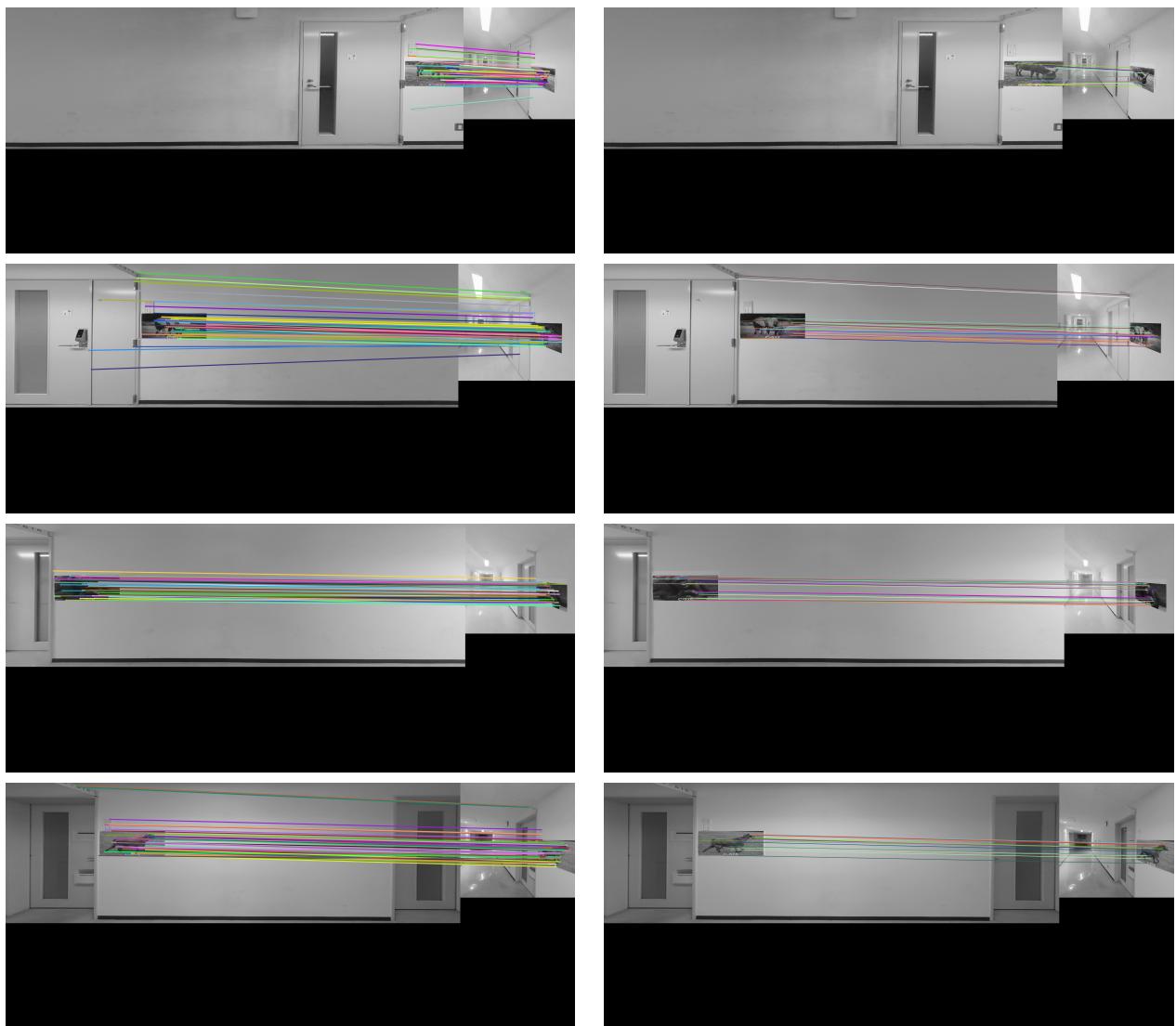


図3: 入力フレームに対する特徴点マッチング結果。左列は機械学習ベース手法、右列は従来手法

各フレームの自己位置推定結果を図4に示す。また、動画は [result.mp4](#)、[result_cuda.mp4](#) に示す。図より、機械学習ベース手法は従来手法に比べ、より多くのフレームで自己位置推定に成功していることが分かる。一方で、テクスチャ補完用のポスターを貼付していない領域では、いずれの手法においても自己位置推定が成立していないことが確認された。



図 4: 自己位置推定結果の比較。左は機械学習ベース手法、右は従来手法

実行時間を表 2 に示す。ただし、前回の報告と大きく変化のない部分については省略している。自己位置推定については、大域最適化ベースの手法を導入することで、大幅な処理時間の削減が可能となった。

一方で、全体の処理時間に大きな差が生じた要因として、一定時間以上連続して自己位置推定に失敗した場合に、すべてのデータベース画像との再マッチングを行う仕様を導入した点が挙げられる。この影響により、マッチングに失敗するフレームが多い従来手法では、処理時間が大きく増加する結果となった。

表 2: 実行時間

SuperPoint,SuperGlue	全体の実行時間 (秒)	自己位置推定 (秒)
	27.002	0.001
SIFT	全体の実行時間 (秒)	自己位置推定 (秒)
	253.323	0.001

6 考察

今回の実験結果から、今後アプリケーションとして実用的に構築するためには、以下のような改良が必要であると考えられる。

まず、撮影される画像中にテクスチャの乏しい面が極力含まれないようにする必要がある。そのために、より多くのポスターを貼付することで人工的にテクスチャを補完し、特徴点の検出・マッチングを安定化させることが有効である。ただし、利用環境が必ずしもテクスチャが豊富であるとは限らないため、線特徴を併用することや、一定時間自己位置推定ができなくても動作を継続できるような設計が求められる。

次に、自己位置推定の処理効率についてである。現時点では、リアルタイムで安定的に自己位置推定を行うためには、機械学習ベースの手法を GPU 上で実行することが望ましい。一方で、CPU 環境に限定される場合には、リアルタイム性を重視するのではなく、必要なタイミングでのみ自己位置推定を行う方式が現実的である。また、さらなる効率化のためには、マッチング処理の並列化が有効であると考えられる。

さらに、入力データの取得方法についても改善の余地がある。現在位置がリセットされてしまった場合には、ユーザに対してパノラマ撮影のように全周を撮影させ、そこから特徴の豊富な領域を切り出して利用することで、迅速に正しい位置を再推定できると考えられる。また、全方位カメラを利用できる環境であれば、全方位画像から多様なテクスチャを含む透視投影画像を取得し、マッチングに有効な領域を選択する手法が有望である。



7 今後の計画

今後の研究計画を以下に示す。

1. 9月：C棟全体でのテクスチャ補完、考察をもとにした改良
2. 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] Wikimedia Commons contributors, “Wikimedia Commons, the free media repository.” <https://commons.wikimedia.org/>. Accessed: 2025-08-27.