

# 適応的閾値を用いた相互最近傍マッチングによる 全天球画像の2視点間位置姿勢推定精度向上

○安藤 大生 (東京大学), 小松 廉 (東京大学), Junwoon Lee (東京大学),  
篠崎 充 (株式会社クボタ), 北島 利浩 (株式会社クボタ),  
浅間 一 (東京大学), 安琪 (東京大学), 山下 淳 (東京大学)

## Improving Two-View Pose Estimation for Spherical Images Using Adaptive Threshold Mutual Nearest Neighbor Matching

○Taisei ANDO (The University of Tokyo), Ren KOMATSU (The University of Tokyo),  
Junwoon LEE (The University of Tokyo), Mitsuru SHINOZAKI (Kubota Corporation),  
Toshihiro KITAJIMA (Kubota Corporation), Hajime ASAMA (The University of Tokyo),  
Qi AN (The University of Tokyo), and Atsushi YAMASHITA (The University of Tokyo)

Abstract: Selection of effective feature extraction techniques is essential for Visual SLAM using spherical images. We propose a method to estimate pose by utilizing mutual nearest neighbor matching by calculating the similarity among all feature points and adaptively filtering them. By using the proposed method on the feature points detected by SuperPoint, we achieved an improvement in the accuracy of pose estimation.

### 1. 序論

近年, 日本の農業人口は減少傾向にあり<sup>[1]</sup>, 農作物の需要に対して供給のみ減少しているため農作業の更なる効率化が求められている. その実現のために科学的アプローチとして Information and Communication Technology (ICT) の導入が進められており, 中でも自動運転技術は農業において大きな生産性向上をもたらすものとして期待されている. ただし現状の農業自動化技術では作業員がロボットを圃場にて近傍監視したうえで動作させる必要があり, 遠隔からの監視に切り替えて無人化を進めるにあたっては圃場間の自律移動が必要である.

圃場間の無人移動のための要素技術として, Global Navigation Satellite System (GNSS) や Light Detection and Ranging (LiDAR) を用いた Simultaneous Localization and Mapping (SLAM)<sup>[2]</sup> が存在する. GNSS は広大な領域での高精度な位置情報を提供することができ, LiDAR SLAM は高精度な 3D 環境の地図生成が可能になっている. しかし, GNSS には山中や森林などで信号が物理的に遮断されることでその利用が困難になる問題が存在する. また, LiDAR SLAM は圃場のように人工物の少ない環境では, 地形の特徴が似ている部分が複数存在するためにロボットが正確な自己位置情報を失うことがある.

これらの問題を解決するため, カメラを用いた Visual SLAM を用いることができる. Visual SLAM はカメラ映像を基に自己位置姿勢と地図情報を推定する技術である. 視覚情報は物体の色や形状などの豊富な情報を含むため, 特定の状況や環境下で GNSS や LiDAR SLAM が効果的でない場合でも Visual SLAM は有用な

情報を提供することができる. Visual SLAM は GNSS や LiDAR SLAM の補完として, またそれらの技術が適用できない場面での代替手法として考えることができる.

Visual SLAM においては通常のカメラを使用するのが一般的だが, 全天球カメラの導入により SLAM の可能性が大きく広がりつつある. 全天球カメラは 360 度の視野を持っており, 1 回の撮影で周囲の全環境を捉えることができるため一般的なカメラに対して環境把握における優位性を持つ. 特に物体が移動する環境下や広範囲の地図生成においてその効果を発揮する<sup>[3]</sup>.

その一方で, 全天球画像には特有の歪みを持つという弱点が存在する. この歪みにより一般的な画像, つまり透視投影形式で用いられる画像処理を適用すると精度が劣るという問題がある. 実際, 全天球画像に対応した既存の SLAM (具体的には OpenVSLAM<sup>[4]</sup> を指す) を農場の環境で適用した結果, 複数の映像で位置推定の失敗が確認された. 実行例を図 1 に示す. ただし特徴点の色に関して, 青, 緑および赤はそれぞれ新しく検出された特徴点, トラッキング中の特徴点およびトラッキングに失敗した特徴点を意味する.

これは農場において近傍の地面や植物から特徴点が十分に取れていないことが原因と考えられる. OpenVSLAM で用いられているアルゴリズムは Oriented FAST and Rotated BRIEF (ORB)<sup>[5]</sup> であり, 全天球画像特有の歪みを考慮した特徴量抽出が行われていない. また ORB のような従来の手法は多くの環境での性能が確認されている<sup>[6]</sup> が, 特定の状況や条件下での性能が限定的である<sup>[7]</sup> ことも知られている. この問題を克服するために学習ベースの特徴点抽出手法を導入するこ



Fig.1 Detection of feature points when self-location estimation by OpenVSLAM fails on a farm.

とで、地面や植物等からも特徴点を検出できる可能性がある。

よって、全天球画像に対する特徴量抽出アルゴリズムに関して更なる検討が必要であると考えられる。したがって、本論文では同じ環境について異なる視点から撮影した画像ペアに対して相対的な位置姿勢推定を扱うことで Visual SLAM に対する効果的な特徴量抽出手法および特徴点マッチングについての議論を行う。ただし SLAM はリアルタイムで実行する必要があるため、位置姿勢推定における精度とともに計算速度に関しても評価を行う。

以上より、本論文では 2 視点間での位置姿勢推定に関して高い精度と速度を有する手法の開発を目的とする。具体的には、SuperPoint<sup>[8]</sup> という学習ベースの特徴点検出アルゴリズムを使用し、それに適した特徴点のマッチング手法を提案する。

## 2. 関連研究

以下では特徴点検出・特徴量抽出・特徴点マッチング、全天球画像の歪みを軽減するための接平面投影処理および全天球画像における位置姿勢推定についての先行研究を記載する。

### 2.1 特徴点検出・特徴量抽出・特徴点マッチング

特徴点検出・特徴量抽出は Visual SLAM において中心的な役割を果たしており、ORB<sup>[5]</sup>、Spherical ORB (SPHORB)<sup>[9]</sup>、Scale-Invariant Feature Transform (SIFT)<sup>[10]</sup>、SuperPoint<sup>[8]</sup> および Accurate and Lightweight Keypoint Detection and Descriptor Extraction (ALIKE)<sup>[11]</sup> など複数のアルゴリズムが存在する。

また、特徴点マッチングとは異なる画像内の特徴点間の類似度を計算して対応関係を確立する技術である。

### 2.2 全天球画像の平面投影

本研究においては、全天球画像を用いて姿勢推定を行う。全天球画像は環境の全方向から特徴点を検出することができるため検出を高精度で行うことができる<sup>[3]</sup>が、一方で画像の歪みが著しく大きいという難点が存在する。

歪みを軽減する代表的な手法としてキューブマップと接平面投影<sup>[12]</sup>が存在する。キューブマップは全天球画像を立方体の 6 つの面に投影する手法であり、接平面投影とは全天球画像を複数の接平面に投影する手法である。接平面投影は投影する接平面の数を制御できるためキューブマップよりも歪み軽減に効果的であり、投影する面の数が増加するに連れて各投影画像

での歪みは小さくなる。ただし計算コストは増大していくので、歪みと計算はトレードオフの関係になっている。

### 2.3 全天球画像における位置姿勢推定

全天球画像で用いることのできるアルゴリズムに関して、キーポイントマッチングに基づく 2 視点間の位置姿勢推定での比較分析を行った研究<sup>[13]</sup>が存在する。全天球画像に対して接平面投影<sup>[12]</sup>を行う場合と行わない場合の 2 種類の画像ペアを用意し、それぞれに対して ORB、SPHORB、SIFT および SuperPoint を用いて特徴点と特徴量を抽出する。そして画像ペアについて回転行列・並進ベクトルを算出して角度誤差を算出し、複数の閾値に基づいて評価している。最終的には位置姿勢推定において SPHORB と接平面投影による処理を加えた SIFT が最も精度の高いアルゴリズムと結論付けられている。しかし、この研究には特徴点の個数に関する制約下での考察が行われておらず、リアルタイム性を必要とする場合の計算時間の議論が不十分である。

また任意に移動する全天球カメラの動きを推定する自己教師あり学習ネットワークを提案した研究<sup>[14]</sup>が存在する。この研究ではラベルなしのデータを使用して学習を行い、カメラの起動と動きを推定することに成功している。ただしこの研究では実行速度についての言及がない。

以上より、全天球画像における位置姿勢推定に関して特徴点の個数を制限したうえで位置姿勢推定精度とともに計算速度を評価する研究が必要である。

## 3. 提案手法

先行研究<sup>[13]</sup>では SuperPoint を用いた場合の精度が ORB・SPHORB・SIFT のものよりも劣る結果となっていた。しかし SuperPoint の提案論文<sup>[8]</sup>や CVPR2020 でのマッチングコンペティション<sup>[15]</sup>ではその優れた性能が示されていることを踏まえると、先行研究<sup>[13]</sup>では SuperPoint の性能を十分に発揮できていない可能性がある。そしてこれを改善することで、SPHORB を使う場合や接平面に投影した画像に対して SIFT を使う場合よりも高い精度で位置姿勢推定を行えるのではないかと考える。

### 3.1 k 近傍マッチングと Lowe's ratio test

位置姿勢推定の先行研究<sup>[13]</sup>では k 近傍マッチングを行い、Lowe's ratio test<sup>[16]</sup>を用いて誤ったマッチングを除去してアルゴリズムを比較していた。

k 近傍マッチングの詳細を式 (1) および (2) に示す。画像 A と画像 B のマッチングを行うとすると、k 近傍マッチングではまず画像 A の特徴点に対して、画像 B の中で特徴量記述子の距離が最も近い特徴点  $b_1$  と次に近い特徴点  $b_2$  を探す。ただし距離とは、ORB の場合にはハミング距離、それ以外の場合にはユークリッド距離を意味し、distance 関数にて 2 点間の距離を計算している。また、 $F_B$  は画像 B の特徴点の集合を意味する。

$$b_1 = \operatorname{argmin}_{b' \in F_B} \operatorname{distance}(a, b'), \quad (1)$$

$$b_2 = \operatorname{argmin}_{b' \in F_B \setminus b_1} \operatorname{distance}(a, b'). \quad (2)$$

また Lowe's ratio test の詳細を式 (3) に示す。Lowe's ratio test では距離の比を計算し、ratio が閾値 (一般的

には 0.75 や 0.8 が用いられる<sup>[16]</sup> よりも小さい場合にマッチングを受け入れる。

$$\text{ratio} = \frac{\text{distance}(a, b_1)}{\text{distance}(a, b_2)}. \quad (3)$$

しかしこの手法では、シーンにおいて類似した特徴点が存在する場合に正しいマッチングを Lowe's ratio test により過度に除去してしまう可能性がある。

そこで本研究では、特徴点のマッチング手法を新たに提案する。具体的には、SuperPoint で得られた特徴点を相互最近傍探索を用いてマッチングした。相互最近傍マッチングの閾値は標準偏差を基に適応的に決定することで位置姿勢推定の精度向上を目指す。

### 3.2 適応的に閾値を決定した相互最近傍マッチング

相互最近傍マッチングは双方向の手法であり、双方の特徴点が互いに最も近い場合にのみマッチングを認める。以下、提案する相互最近傍マッチングのアルゴリズムを記載する。画像 A と画像 B のマッチングを行うとすると、まず式 (4) の通り画像 A の特徴点  $a$  について画像 B の中で最も近い特徴点  $b$  を探し、次に式 (5) の通り画像 B の特徴点  $b$  について、画像 A の中で最も近い特徴点  $a'$  を探す。もし  $a$  と  $a'$  が一致した場合、特徴点  $a$  と特徴点  $b$  は最近傍となり、一致したと判断する。ただし一致したと判断する閾値は各特徴量の類似度の最小値と標準偏差を利用して設定する。

$$b = \underset{b' \in \mathbf{F}_B}{\text{argmin}} \text{distance}(a, b'), \quad (4)$$

$$a' = \underset{a'' \in \mathbf{F}_A}{\text{argmin}} \text{distance}(b, a''). \quad (5)$$

具体的な処理を示す。画像 A および画像 B の特徴量記述子を並べた行列  $\mathbf{D}_A \in \mathbb{R}^{P_A \times N}$  および  $\mathbf{D}_B \in \mathbb{R}^{P_B \times N}$  の転置について行列積を取り、全特徴点間の類似度を含んだ行列  $\mathbf{S}$  を算出する。ただし  $P_A$  および  $P_B$  はそれぞれ画像 A および B の特徴点の個数であり、 $N$  は特徴量記述子の長さである。 $\mathbf{S}$  の各要素について式 (6) を用いて閾値  $t$  を算出し、それを用いて式 (7) でフィルタリングを行う。ただし  $\min(\mathbf{S})$ ,  $\text{std}(\mathbf{S})$  はそれぞれ行列  $\mathbf{S}$  の要素の最小値および標準偏差を意味する。この処理により、データセットの性質を踏まえつつ誤ったマッチングを抑制することができる。

$$t = \min(\mathbf{S}) + 4\text{std}(\mathbf{S}), \quad (6)$$

$$s_{ij} = \begin{cases} 0 & \text{if } s_{ij} < t \\ s_{ij} & \text{otherwise} \end{cases}. \quad (7)$$

そして  $\mathbf{S}$  の各行および各列について最大値となるインデックスを取得し、双方向で最大となった場合にマッチングを認める。この双方向性により確実なマッチング結果を得ることができる。特に類似した特徴点が多数存在するシーンに関して、SuperPoint のマッチング手法として相互最近傍探索を用いることで、より高い精度をもたらせる可能性がある。



Fig.2 Example of indoor scene used in experiment.



Fig.3 Example of outdoor scene used in experiment.

## 4. 実験

画像について相対的な位置姿勢推定を行って回転行列と並進ベクトルの誤差を算出し、誤差が設定した閾値内に収まっているかどうかで判定して精度評価を行う。またこの閾値を変動させ、閾値ごとに特徴点検出アルゴリズムの精度を比較する。位置姿勢推定を用いる場合には誤差の許容範囲があり、閾値を設定してその中で性能を評価することが重要と考えたためこの評価方法を採用した。

ただし最終的に SLAM で処理を行うにあたってリアルタイム性を必要とするため、マッチングの対象とする特徴点は検出時の応答値が上位の 500 個に限定し、また投影する接平面の数は 20 とした。

以下に詳細な設定および実験結果を記載する。

### 4.1 実験環境

#### 4.1.1 使用するデータセット

データセットとして、先行研究<sup>[13]</sup>で使用されている仮想環境の画像を用いる。これは Blender や UnrealCV などのオープンソースの 3D モデリングソフトウェアを使用して 9 つの 3D モデルから 100 枚の画像ペアが生成されており、画像ペアを屋内と屋外のシーンで分けて評価を行う。また真値は画像生成の際に予め求められている。画像の解像度は 1024x512 である。

実験で使用した屋内シーンおよび屋外シーンの例をそれぞれ図 2, 3 に示す。

#### 4.1.2 位置姿勢の誤差算出

本研究においては抽出した特徴について 5 点法<sup>[17]</sup>を用いて相対的な位置姿勢を回転行列・並進ベクトルとして算出しており、回転行列はクォータニオンに変換して平均化、並進ベクトルは単位ベクトルに変換して平均化して誤差を計算した。これに関して、クォータニオン  $\mathbf{q}_1, \mathbf{q}_2 \in R^4$  についてそれらのなす角度、 $\theta_{\text{quat}}$

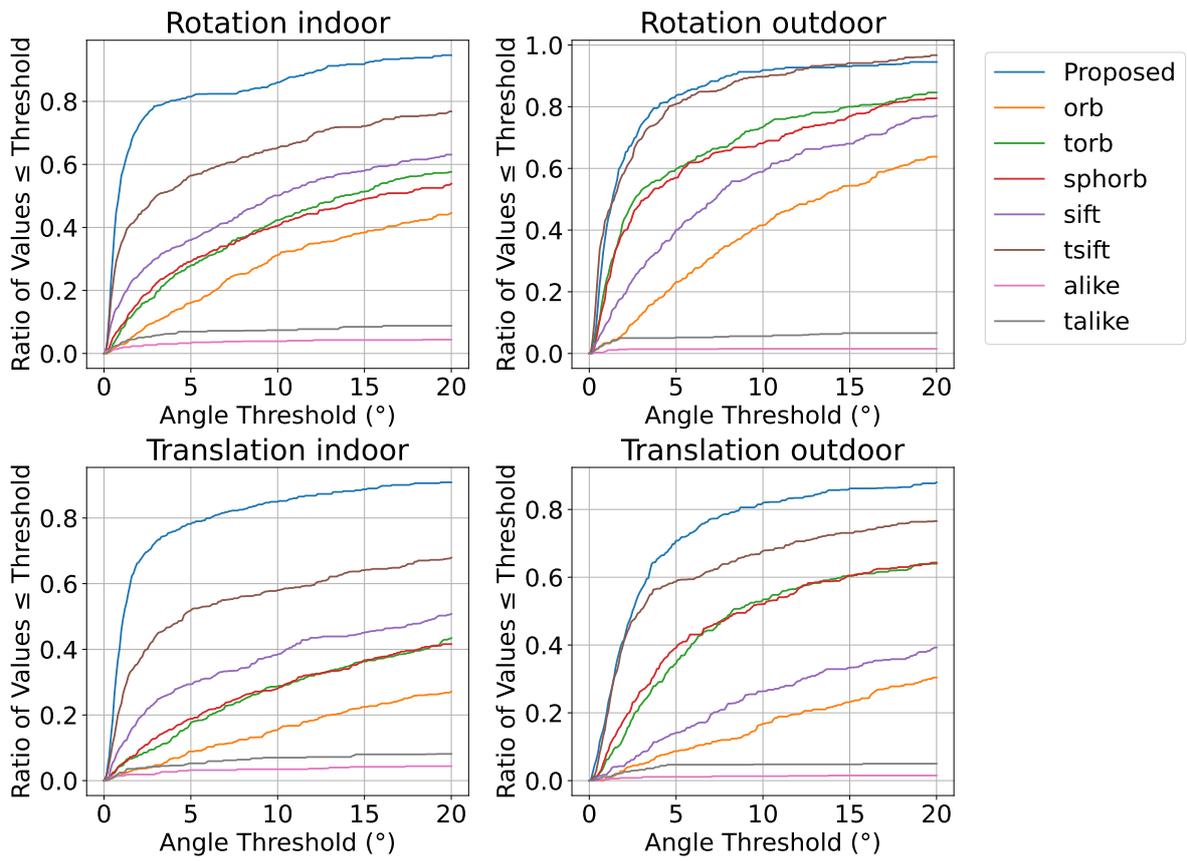


Fig.4 Error by threshold for pose estimation.

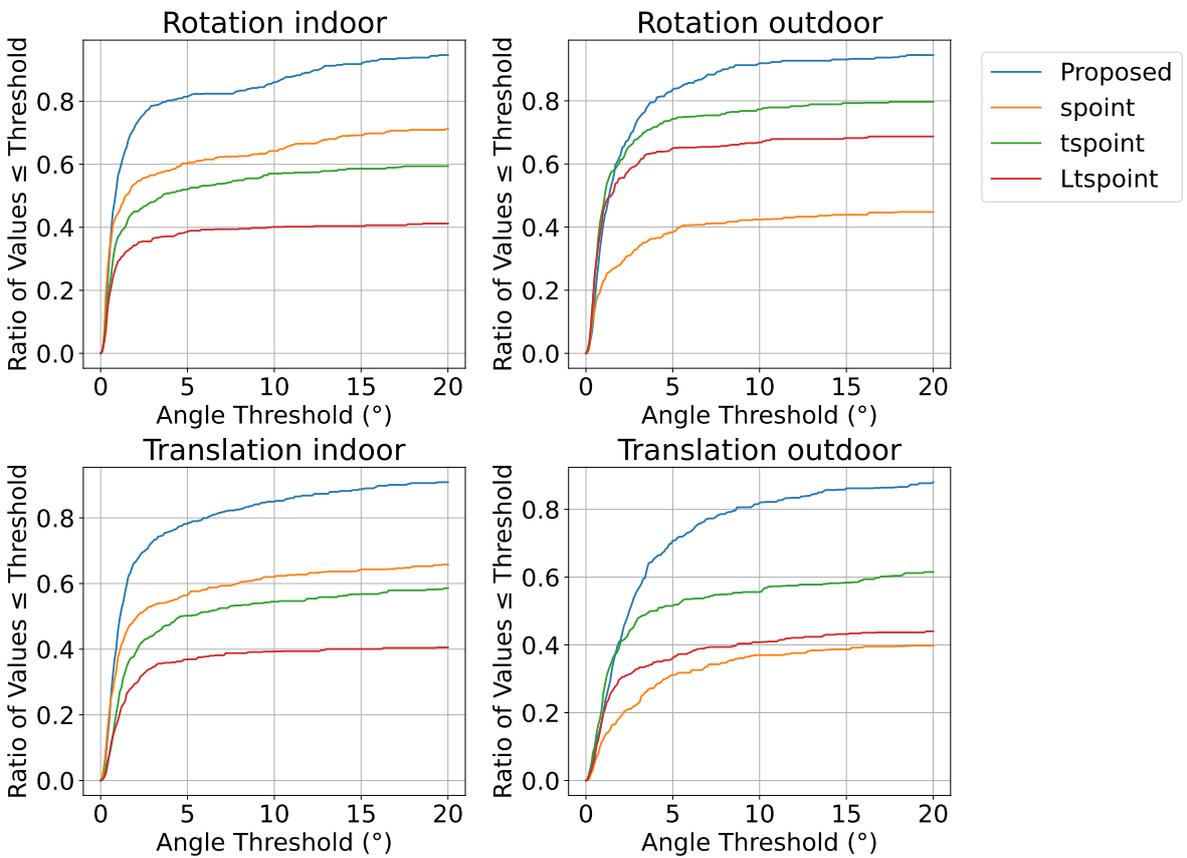


Fig.5 Ablation study on SuperPoint feature point matching.



Fig.6 Feature point matching on farm using tangent plane projection, SuperPoint, and the proposed method.

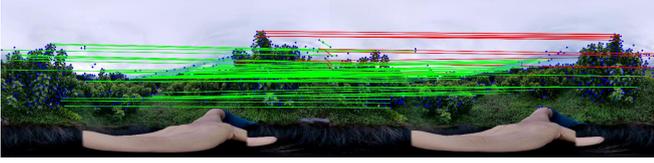


Fig.8 Feature point matching on farm using tangent plane projection, SuperPoint, k-nearest neighbor matching, and Lowe's ratio test.

は式 (8) を用いて定義される。また 3 次元のベクトル  $\mathbf{t}_1, \mathbf{t}_2 \in R^3$  のなす角度,  $\theta_{\text{trans}}$  は式 (9) を用いて定義される。

$$\theta_{\text{quat}} = 2 \arccos(\mathbf{q}_1 \cdot \mathbf{q}_2), \quad (8)$$

$$\theta_{\text{trans}} = \arccos\left(\frac{\mathbf{t}_1 \cdot \mathbf{t}_2}{\|\mathbf{t}_1\| \|\mathbf{t}_2\|}\right). \quad (9)$$

#### 4.1.3 比較した特徴量

今回比較に使用したアルゴリズムは, ORB, SPHORB, SIFT, SuperPoint および ALIKE である。さらに SPHORB 以外は 2.2 項で述べた接平面投影の手順を踏まえて球面画像に適応させたものも比較に加える。これを以下それぞれ TORB, TSIFT, TSuperPoint および TALIKE と呼ぶ。

#### 4.1.4 実行環境

我々の実験について, OS は Ubuntu 20.04.6 LTS, プロセッサは 12th Gen Intel (R) Core (TM) i9-12900, メモリは 64GB DDR4, GPU は NVIDIA RTX A4500 を搭載したデバイスを使用する。また本研究で使用したプログラムは Python 3.7 によって実装されている。

## 4.2 結果

以下, 図表内では簡単のために ORB, SPHORB, SIFT, SuperPoint および ALIKE をそれぞれ orb, sphorb, sift, spoint および alike と表記し, 接平面投影を用いた場合にはそれぞれ torb, tsift, tspoint および talike と表記する。また提案手法は Proposed, 提案手法と Lowe's ratio test の併用は Ltspoint と表記する。

### 4.2.1 位置姿勢推定の閾値別精度

本研究における位置姿勢推定の結果を図 4 に示す。図 4 では閾値となる角度の誤差を横軸の値とし, その閾値以下に誤差が収まるデータの割合を縦軸の値として示している。ここで"Rotation"と"Translation"はそれぞれ姿勢および位置を表し, "indoor"と"outdoor"はそれぞれ屋内シーンおよび屋外シーンを指す。屋内シーンでの姿勢推定結果を例にとると, 各手法における姿勢推定の誤差が 5 度以内に収まる割合は"Rotation indoor"のグラフで x 軸が 5 度の部分を参照することで確認で



Fig.7 Feature point matching on farm using SuperPoint, k-nearest neighbor matching, and Lowe's ratio test.



Fig.9 Feature point matching on farm using tangent plane projection, SuperPoint, mutual nearest neighbor matching, and Lowe's ratio test.

きる。この例では, 提案手法は約 0.8, TSIFT は約 0.55 の割合で誤差が 5 度以内に収まることがわかる。

図 4 より位置・姿勢, また屋内シーン・屋外シーンのどの組み合わせでも提案手法を用いた場合の精度が最良であり, 次に TSIFT を用いた場合の精度が高いことがわかる。また, どの結果も閾値の変化に対してアルゴリズムの精度の順序は大きく変動しない。

### 4.2.2 SuperPoint のマッチング手法の寄与分析

次に SuperPoint に関して, マッチング手法を比較した結果を図 5 に示す。図 5 を見ると, 提案手法は他のマッチング手法を用いた場合と比べて精度が高くなっていることがわかる。

現実の農場で取得したデータから SuperPoint を用いて特徴点マッチングを行った結果を図 6, 7, 8 および 9 に示す。ただし図 6, 7, 8 および 9 はそれぞれ図 5 の Proposed, spoint, tspoint および Ltspoint に対応しており, 緑色および赤色の線はそれぞれ Random Sample Consensus (RANSAC) により判定された正しいマッチングおよび誤ったマッチングを意味する。また, 画像下部は手や頭部が写っているのでマスク処理を行った。

提案手法を用いることで正しいマッチングの数が大きく増加していることがわかる。

### 4.2.3 計算時間

各特徴量において, 画像ペアに対し特徴点検出・特徴量抽出を行い, 位置姿勢推定を行う際の計算時間を表 1 に示す。表 1 を見ると接平面投影を用いた SuperPoint での特徴点検出時間は ORB の約 2 倍, また TSIFT の約 2/3 倍である。またマッチング時間に関して, 提案手法は k 近傍マッチングを用いた場合よりも小さい。

## 5. 結論

本研究では SuperPoint を活用した位置姿勢推定のマッチング手法を新たに提案した。この手法は, 適応的な閾値を使用してフィルタリングを行った上で相互最近傍マッチングを適用するものであり, データセットの特性を考慮することで堅牢なマッチングを実現した。提案手法により, 特徴点の数が制限された状況での複数のシーンでの位置姿勢推定において従来手法に比べて高い精度を達成することができた。

一方で, 本研究の限界として実際の農場のデータを

Table 1 Computation time per single pair when number of feature points is limited to 500.

	Methods										
	Proposed	orb	torb	sphorb	sift	tsift	alike	talike	spoint	tspoint	Ltspoint
Feature point detection (s)	0.49	0.24	0.24	0.18	0.25	0.75	0.20	0.37	0.24	0.49	0.49
Matching (s)	0.006	0.011	0.014	0.001	0.011	0.011	0.010	0.011	0.012	0.012	0.013

使用した評価を行っていない点が挙げられる。この課題を克服するため、今後は実際の農場データを使用した精度と速度の評価を進める予定である。

## 参考文献

- [1] 農林水産省: (1) 基幹的農業従事者. アクセス日: 2023-10-09. (2023). URL: [https://www.maff.go.jp/j/wpaper/w\\_maff/r3/r3\\_h/trend/part1/chap1/c1\\_1\\_01.html](https://www.maff.go.jp/j/wpaper/w_maff/r3/r3_h/trend/part1/chap1/c1_1_01.html).
- [2] H. Durrant-Whyte and T. Bailey: “Simultaneous Localization and Mapping: Part I”. *IEEE Robotics & Automation Magazine* 13.2, pp. 99–110, (2006).
- [3] Alejandro Rituerto, Luis Puig, and J.J. Guerrero: “Visual SLAM with an Omnidirectional Camera”. *Proceedings of the 20th International Conference on Pattern Recognition*, pp. 348–351, (2010).
- [4] Shinya Sumikura, Mikiya Shibuya, and Ken Sakurada: “OpenVSLAM: A Versatile Visual SLAM Framework”. *Proceedings of the 27th ACM International Conference on Multimedia*, pp. 2292–2295, (2019).
- [5] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski: “ORB: An Efficient Alternative to SIFT or SURF”. *Proceedings of the 2011 International Conference on Computer Vision*, pp. 2564–2571, (2011).
- [6] Carlos Campos, Richard Elvira, Juan J. Gómez Rodríguez, José M. M. Montiel, and Juan D. Tardós: “ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial, and Multimap SLAM”. *IEEE Transactions on Robotics* 37.6, pp. 1874–1890, (2021).
- [7] Yong Dai and Jiaxin Wu: “An Improved ORB Feature Extraction Algorithm Based on Enhanced Image and Truncated Adaptive Threshold”. *IEEE Access* 11, pp. 32073–32081, (2023).
- [8] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich: *SuperPoint: Self-Supervised Interest Point Detection and Description*. (2018). arXiv: 1712.07629.
- [9] Qiang Zhao, Wei Feng, Liang Wan, and Jiawan Zhang: “SPHORB: A Fast and Robust Binary Feature on the Sphere”. *International Journal of Computer Vision* 113.2, pp. 1573–1405, (2015).
- [10] David G Lowe: “Distinctive Image Features from Scale-Invariant Keypoints”. *International Journal of Computer Vision* 60.2, pp. 1573–1405, (2004).
- [11] Xiaoming Zhao, Xingming Wu, Jinyu Miao, Weihai Chen, Peter C. Y. Chen, and Zhengguo Li: “ALIKE: Accurate and Lightweight Keypoint Detection and Descriptor Extraction”. *IEEE Transactions on Multimedia* 25, pp. 3101–3112, (2022).
- [12] Marc Eder, Mykhailo Shvets, John Lim, and Jan-Michael Frahm: “Tangent Images for Mitigating Spherical Distortion”. *Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12423–12431, (2020).
- [13] Jeffri Murrugarra-Llerena, Thiago L. T. Da Silveira, and Claudio R. Jung: “Pose Estimation for Two-View Panoramas based on Keypoint Matching: a Comparative Study and Critical Analysis”. *Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 5198–5207, (2022).
- [14] Dabae Kim, Sarthak Pathak, Alessandro Moro, Atsushi Yamashita, and Hajime Asama: “SelfSphNet: Motion Estimation of a Spherical Camera via Self-Supervised Learning”. *IEEE Access* 8, pp. 41847–41859, (2020).
- [15] *IMW 2020: Leaderboard*. Accessed: 2023/10/13. (2020). URL: <https://www.cs.ubc.ca/research/image-matching-challenge/2020/leaderboard/>.
- [16] Jiri Matas, Ondrej Chum, Martin Urban, and Tomas Pajdla: “Robust Wide-baseline Stereo from Maximally Stable Extremal Regions”. *Image and Vision Computing* 22.10, pp. 761–767, (2004).
- [17] D. Nister: “An Efficient Solution to the Five-Point Relative Pose Problem”. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26.6, pp. 756–770, (2004).