

全方位動画像を用いた自己位置推定高速化のための 特徴空間の構築

Construction of feature space for speeding up self-localization using time series omnidirectional images

佐々木 勇人† 濱上 知樹†
Hayato SASAKI Tomoki HAMAGAMI

1 はじめに

自律移動ロボットだけではなくロボットが利用する空間側を智能化することで、より高度なサービスを提供しようとする研究がなされてきた。そのような空間を智能化空間という [1]。特に、空間智能化の手法としては、RFID タグのような物理的なタグを実空間中に配置する手法がある。これは情報を実空間中に埋め込み、ロボット等の空間利用者がその情報にアクセスすることで、ロボットの行動をより高度にするという取り組みであった。しかし RFID タグのような物理的なタグでは、設置によるコストや設置できる場所が限られるという課題がある。その改善策として、動画像により智能化を行う実空間の情報を収集し、そこから得られる時系列画像とタグを紐付ける手法が研究された [2]。ただしこの手法ではタグにアクセスするために、画像からの現在位置の同定が必要となる。

時系列画像を用いた自己位置推定では観測画像とデータベース上の画像との比較を行うが、データベースの規模が大きくなると比較に要する計算コストも高くなる。データベースの規模は智能化を行う環境の規模に応じて大きくなるため、計算コストを抑えることは重要な課題である。さらに今日では、様々な場所に設置された監視カメラ、ドライブレコーダ、ライフログカメラ等の普及により、様々な時系列画像を大量に収集できる環境が整ってきた。このことから、大量の時系列画像から効率的に観測画像とのマッチングをとる手法が求められる。

本稿ではカメラの向きによらずに情報を取得できる全方位カメラをセンサとして用い、これにより自己位置推定を行うことでタグの読み取りおよび空間の智能化を行う。そしてオフィスの一室といった小規模な環境ではなく、それよりも規模の大きな建物全体もしくは複数の建物というような環境を智能化することを想定して、自己位置推定によるタグ読み取りの高速化手法を提案する。

2 時系列情報を含んだ局所画像特徴量による自己位置推定

画像同士のマッチング手法は、領域ベースな手法と特徴点ベースな手法とに分類される。テンプレートマッチングのような領域ベースな比較手法で位置推定を行う場合、移動物体や照明変化といった環境変化の影響を大きく受けてしまう。対して特徴点ベースな比較手法の場合、領域ベースな手法と比べて局所的な画像領域の特徴を用いて比較を行う。

局所画像特徴量として有名なものには SIFT や SURF があるが、これらは画像中の勾配情報を元に特徴点の抽出を行う。そのため、テンプレートマッチングと比較すると影響は小さくなるものの、環境変動による影響を完全には排除できない。

そこで環境変動に対してロバストな位置推定を行うために、時系列情報を含んだ局所画像特徴量が研究されてきた。本稿では位置推定 (位置識別) を連続する時系列画像群からなる

シーンの識別に帰着している。PIRF(Position-Invariant Robust Features)[3] は環境変動の影響が生じている特徴点を取り除いて抽出される特徴量であり、シーン識別において有効な特徴量である。PIRF-nav[4] や PIRF-nav2[5] では、この PIRF を用いて自己位置推定を行う。

環境変動に対してロバストであるという性質の他にも、PIRF はカメラの移動に対してセンシティブな特徴点を排除できるという利点も存在する。本研究では、画像を取得するカメラとして全方位カメラを使用している。また、智能化を行う環境としては屋内に重点を置いている。全方位カメラで屋内を撮影する際、カメラに対して離れた位置にある物体 (カメラに対して前方の物体) の見え方は変化が小さいが、近くにある物体 (主にカメラに対して側方の物体) の見え方はカメラの移動に対して大きく変化する。このカメラ移動に対してあまりに見え方の変化が大きい物体から得られた局所画像特徴量を用いて画像のマッチングを行おうとすると、あまりにピンポイントな位置で撮られた画像にのみマッチするようになってしまう。PIRF は見え方の変化が大きい特徴点は排除するため、全方位カメラを用いた位置の推定に適した特徴量だといえる。

3 PIRF-nav2

本手法は PIRF-nav2 を応用して実空間をモデル化する。PIRF-nav2 では、移動時に取得される時系列画像群から PIRF を抽出し、観測画像とデータベース内の画像群とのマッチングを行う。

PIRF-nav2 において実空間はトポロジカルなマップでモデリングされる。本稿では実空間を位置ノードという単位に分割し、各位置ノードでの観測情報から得られる特徴ベクトルを用いて自己位置推定を行う。また、トポロジカルなマップに加えて、PIRF 辞書と位置ノード特徴ベクトル辞書を構築していくことで実空間をモデリングしている。このことを踏まえて PIRF-nav2 により実空間をモデル化するためには、観測画像ごとに図 1 で示される処理を行う。

3.1 PIRF

PIRF は連続する時系列画像から SIFT もしくは SURF を抽出することで、環境変動に対するロバスト性を獲得している。これは、時系列画像群を用いて連続的にマッチングを取ることによって、移動物体から抽出される特徴点群を排除しているためである。この PIRF は次のようにして抽出される。

1. 時刻 t で撮影された画像 I_t と時刻 $t-1$ で撮影された画像 I_{t-1} それぞれから SURF 集合 S_t 、 S_{t-1} を抽出し、 $d \in S_t$ とマッチングが取れる特徴点を S_{t-1} から探索する

SURF のマッチングにおいて、 I_t から抽出されたある SURF 特徴点 d に対して、 I_{t-1} から抽出される特徴点のうちユークリッド距離が最も小さい特徴点を d_1 、2 番目に距離が小さい特徴点を d_2 とした時、ratio test という条件式 (1) を満たす d_1 を対応する特徴点として決定している。この ratio test は、マッチする特徴点の候補が複数存在する場合にはマッチングを棄却するという考え方である。こ

† 横浜国立大学大学院工学府

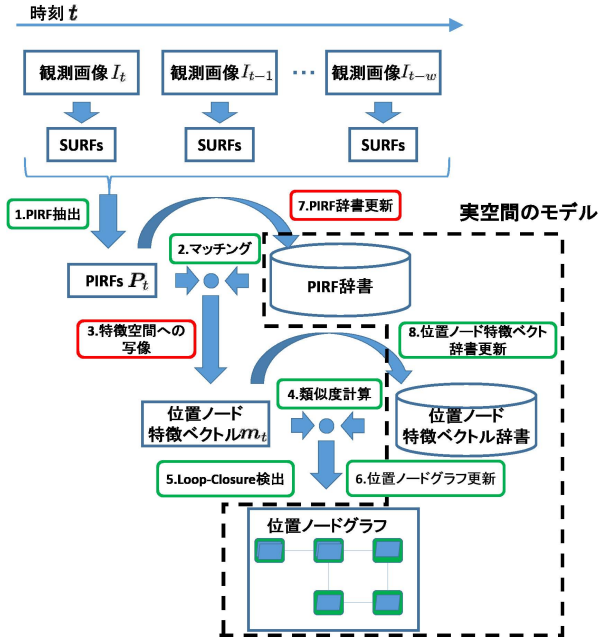


図1: 実空間のモデリングの流れ

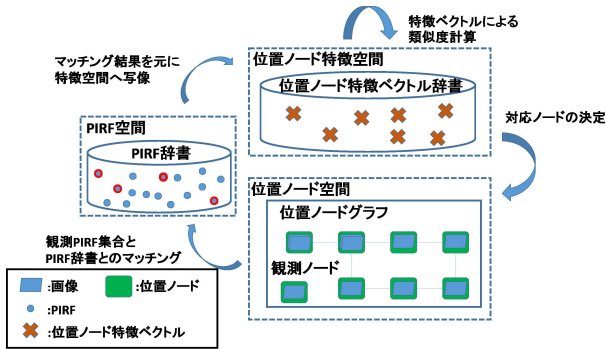


図2: モデル内の論理関係

の方法によりマッチングの再現率は低下してしまうものの、適合率を高めることができる。

$$\frac{\|d - d_1\|_2}{\|d - d_2\|_2} < D_{surf} \quad (1)$$

ただし、 D_{surf} は閾値定数。

2. 窓サイズ w_{size} の範囲内で連続的にマッチングをとり、その範囲内の全てでマッチングがとれた特徴点を PIRF 特徴点として抽出する

まず、項目1でマッチングのとれた特徴点 $d_1 \in S_{t-1}$ に対して、再び項目1と同様のマッチングを行う。すなわち、 S_{t-1} と S_{t-2} とでマッチングを行い、これを S_{t-2} と $S_{t-3}, \dots, S_{t-w_{size}-1}$ と $S_{t-w_{size}}$ と繰り返していく。

PIRF-nav2 において、時刻 t に位置ノード L_t で観測された全方位画像 I_t から PIRF 集合 P_t を抽出する際には、窓サイズ w_{size} を動的に調節することで I_t から抽出される PIRF の個数を制御する。具体的には抽出される PIRF の個数 $|P_t|$ を2つの定数 n_{pmin} , n_{pmax} を用いて

$$n_{pmin} \leq |P_t| \leq n_{pmax} \quad (2)$$

と指定し、 $|P_t| < n_{pmin}$ の場合には w_{size} をデクリメントし、 $|P_t| > n_{pmax}$ の場合には w_{size} をインクリメントすることで $|P_t|$ を制御する。この操作を、 $w_{size} = w_{sizeDefault}$ から初めて、式(2)を満たすまで繰り返す。

3.2 位置ノード特徴ベクトル

位置ノード特徴ベクトル m_t は、位置ノード L_t で観測された PIRF 集合 P_t を時刻 $t - \varepsilon$ での PIRF 辞書 $Q_{t-\varepsilon}$ の要素を用いて表現したバイナリベクトルである。ここで PIRF 辞書 Q_t は、時刻 t までに観測された PIRF が登録された集合であり、 ε は時刻 t の直近の観測を除外するためのオフセット値を表す。

$$m_t = (b_1^t, b_2^t, \dots, b_{|Q_{t-\varepsilon}|}^t) \quad (3)$$

ただし、 $b_i^t \in \{0, 1\}$, ($i = 1, 2, \dots, |Q_{t-\varepsilon}|$) であり、

$$b_i^t = \begin{cases} 1 & (q_i \in Q_{t-\varepsilon} \text{ と マッチング が 取 れ る } p \in P_t \text{ が 存 在 す る}) \\ 0 & (\text{otherwise}) \end{cases} \quad (4)$$

3.3 位置ノード特徴ベクトルの類似度計算

現在の観測が過去に経験済みの観測であるか判定することを Loop-Closure 検出とよび、 L_t と $L_j \in \mathcal{L}_{t-1}$, ($j = 1, \dots, |\mathcal{L}_{t-1}|$) に対応する位置ノード特徴ベクトルを用いて Loop-Closure 検出を行う。なお、 \mathcal{L}_t は時刻 t での位置ノード集合を表す。

L_t に対応する特徴ベクトルを m_t , L_j に対応する特徴ベクトルを $m_j \in M_{t-1}(\mathcal{L}_{t-1})$ の各要素に対応する位置ノード特徴ベクトル集合) として、2つの特徴ベクトル間の類似度 z_j^{norm} が閾値 τ 以上であった場合に Loop-Closure であると判定する。

$$s_j = \frac{\|m_t\|_1}{n_{pmax}} \times \frac{\|m_t \wedge m_j\|_1}{n_{pmax}} \quad (5)$$

$$z_j = s_j \times p_T(0, \sigma, 0) + \sum_{i \in I_j} s_i \times p_i(0, \sigma, 1) \quad (6)$$

$$p_T(\mu, \sigma, x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x - \mu)^2}{2\sigma^2}\right) \quad (7)$$

$$z_j^{norm} = \frac{z_j + \sum_{i \in I_j} z_i}{1 + |I_j|} \quad (8)$$

ここで s_j はバイナリベクトルの値を用いて計算される類似度で、 P_t 中に過去の観測が多く含まれ、また m_t と m_j との間で共通する PIRF が多く含まれるほど類似度は大きくなる。そして z_j は s_j を位置ノードの隣接情報を元に補正を行った値である。ただし、式(6)中の I_j は L_j に隣接する位置ノードのインデックス集合を表す。また、 σ の値は PIRF-nav2 の利用時に設定する定数である。最終的に得られる類似度 z_j^{norm} は、 z_j を正規化した値となっている。

Loop-Closure が検出された場合には L_t に対応する位置ノード $L_c \in \mathcal{L}_{t-1}$ を次のように決定する。

$$c = \arg \max_{j=1, \dots, |M_{t-1}|} (z_j^{norm}) \quad (9)$$

3.4 PIRF 辞書の更新

時刻 t で得られた PIRF 集合 P_t のうち、PIRF 辞書 Q_{t-1} に未登録な PIRF を辞書に追加する。すなわち Q_{t-1} に存在しない、 P_t で新たに観測された PIRF 集合を P_{new} としたとき、次のように辞書を更新する。

$$Q_t \leftarrow Q_{t-1} \cup P_{new} \quad (10)$$

ただし、 $Q_0 = \emptyset$ である。

4 既存手法の改善

既存システムでは、位置ノード特徴ベクトルをバイナリベクトルで表現することで特徴ベクトル同士の類似度計算を高速化している。

しかし位置ノード特徴ベクトルを計算する際に、 $|P_i| \times |Q_{t-\varepsilon}|$ 回の距離計算が必要となる。この計算が自己位置推定において時間的なボトルネックとなってしまっていた。さらに既存システムでの PIRF 辞書の更新方法では、辞書中に類似した PIRF が複数登録されてしまう可能性があった。これは P_{new} の中に類似している PIRF が複数存在する可能性を考慮していないためである。

そこでこれらの課題を解決するために本研究では PIRF 辞書の更新方法の改善と PIRF 辞書のインデックス化を提案している。

なお、本研究では図 1 より実空間のモデル構築が完了した後で、ロボットが知能化空間を利用することを考えている。したがって、PIRF 辞書の構築やインデックス化にかかる時間は考慮しない。

4.1 PIRF 辞書の更新方法の改善

既存手法では 3.4 のように、PIRF 辞書の更新の際に観測 PIRF 集合 P_t 中で PIRF 辞書 Q_{t-1} に未登録な PIRF 集合 P_{new} を全て登録していた。しかし P_{new} 全てを Q_{t-1} に対して追加すると、 P_{new} の中に類似した PIRF 特徴点が複数存在した場合に、辞書に類似した PIRF 特徴点が複数登録されてしまう。このことは PIRF 辞書の要素数を無駄に増やしてしまうだけでなく、位置ノード特徴ベクトルを計算する際に、外乱により同一の特徴点が区別される危険性をもつ。

また、PIRF 同士のマッチングを行う際には式 (1) に示すような ratio test を用いており、これによりマッチングの適合率を上げている。しかしながら、ratio test を用いると次のような場合でマッチング結果が異なる。

1. $p \in P_{new}$ とマッチする PIRF が Q_{t-1} 中に存在するかどうか
2. $p \in P_{new}$ とマッチする PIRF が $P_{new} \setminus p$ 中に存在するかどうか判定した後、さらに 1 を判定する
3. $p \in P_{new}$ とマッチする PIRF が $Q_{t-1} \cup (P_{new} \setminus p)$ 中に存在するかどうか

ratio test は $p \in P_{new}$ に最も近い点と 2 番目に近い点との比によりマッチング判定を行うということから、更新後の辞書 Q_t 内に類似した特徴点が存在しないためには、3 の手法のように Q_{t-1} と $P_{new} \setminus p$ の両方を同時に考慮する必要がある。ただし、実際には辞書 Q_{t-1} に P_{new} を全て登録するわけではないので、 $Q_{t-1} \cup (P_{new} \setminus p)$ ではなく、次に示すような PIRF 集合 $Q_{t-1}^{(i)}$ との比較を行う。

- $P_{new}^{(i)} \in P_{new}^{sorted}$ とマッチングが取れる特徴点が $Q_{t-1}^{(i-1)}$ 中に存在しない場合

$$Q_{t-1}^{(i)} \leftarrow Q_{t-1}^{(i-1)} \cup (P_{new}^{(i)} \in P_{new}^{sorted}), (i = 1, \dots, |P_{new}^{sorted}|) \quad (11)$$

- $P_{new}^{(i)} \in P_{new}^{sorted}$ とマッチングが取れる特徴点が $Q_{t-1}^{(i-1)}$ 中に存在する場合

$$Q_{t-1}^{(i)} \leftarrow Q_{t-1}^{(i-1)} (i = 1, \dots, |P_{new}^{sorted}|) \quad (12)$$

ただし 2 点 p, q 間の距離を $\text{dist}(p, q)$ として $p_k \in P_{new}$ と PIRF 辞書 Q_{t-1} 中の最近傍点との距離 d_k を式 (13) のように定義し、この d_k の降順で P_{new} 中の要素を並べ替えたものを P_{new}^{sorted} と定義する。

$$d_k = \min_{q_j \in Q_{t-1}} \text{dist}(p_k, q_j) \quad (13)$$

$P_{new}^{(i)}$ は P_{new}^{sorted} の i 番目の要素である。

さらに式 (11), (12) の境界条件は

$$Q_{t-1}^{(0)} = Q_{t-1} \quad (14)$$

$$Q_t = Q_{t-1}^{(|P_{new}^{sorted}|)} \quad (15)$$

となっている。

なお、この手法では P_{new}^{sorted} について辞書への登録を打ち切る処理を明示的に定めてはいないが、 i が大きくなるにしたがって (1) 式が成り立つ可能性が高くなるため、結果として登録が棄却されるようになる。

このように特徴点の登録判定を d_i が大きな順から一個ずつ行うことによって、既存手法での辞書更新方法で得られる辞書の分布をある程度保ちつつ辞書の要素数を削減することが可能となる。

4.2 PIRF 辞書のインデックス化

位置ノード特徴ベクトルを計算するためには観測 PIRF 集合 P_t と PIRF 辞書 Q_{t-1} とのマッチングが必要となる。しかし、PIRF 辞書の要素数が大きくなった場合に、全探索でマッチングを行うのは非効率である。

したがって本研究では PIRF 辞書に対してインデックス化を行い、全探索 (BF search (Brute Force Search)) ではなく近似近傍探索 (ANN search (Approximate Nearest Neighbor Search)) を行うことで位置ノード特徴ベクトルの計算時間を短縮化する。

ただし本研究では、PIRF 辞書の更新が全て更新し終わったあとで辞書のインデックス化を行う。これは、実空間中を移動するロボット (もしくは人) は知能化空間のユーザであり、空間の知能化はユーザが利用する前に構築済みであることを想定しているためである。

なお、近似近傍探索の手法として Marius らの研究 [6][7] で提案されている手法を用いる。Marius らの研究では、近似近傍探索を行うためのライブラリ FLANN (Fast Library for Approximate Nearest Neighbor) を提供しており、SIFT や SURF のような高次元なデータに対する探索において階層的 k-means tree や Randomized kd-tree による探索木が有効であることが示されている。また、探索木の探索は priority search という手法で行われ、この手法では高次元なデータに対してモクエリの周辺の空間を効率的に探索することができる。

4.2.1 PIRF 辞書の探索木の構築

PIRF 辞書の探索木として、階層的 k-means tree もしくは Randomized kd-tree を使い、PIRF 辞書中のデータ分布に応じて適当なものを選択する。

階層的 k-means tree は訓練データセットに対して階層的に k-means 法を適用していくことで構築され、パラメータとしては木の分岐数と木の深さが存在する。

一方 Randomized kd-tree では、 k 次元のユークリッド空間を再帰的に分割する kd-tree を複数構築する。kd-tree はデータセットの分散が大きくなる次元を選択し、その次元でのデータセットの中央値で空間を分割していたが、Randomized kd-tree では、分散が大きくなる次元の上位 D 個の中からランダムで選択することで空間を分割していく。Randomized kd-tree はこのように多様な kd-tree を複数構築する方法で、パラメータとしては木の構築数が存在する。

FLANN ではこの 2 つのアルゴリズムの内、どちらがデータセットのインデックス化に適しているかということも探索木のパラメータとして考え、最適な探索木の構築をパラメータ最適化問題に帰着している。

本システムでは、この FLANN を用いて自動的に決定されるパラメータの探索木により PIRF 辞書のインデックス化を行っている。[6] では、128 次元のベクトルである SIFT について、近似近傍探索時に適合率を高くするようにした場合、階層的 k-means tree のほうが探索コストが小さいことが示されて

いる。実際に FLANN によるパラメータ探索の結果、実験で使用したデータセットに対しては階層的 k-means が最適なアルゴリズムとして判定された。

4.2.2 未探索ノードの優先度

構築した木の探索を行う際には priority search という手法が用いられる。階層的 k-means tree および Randomized kd-tree とともに探索時に priority queue を一つ保持し、priority queue に保存された未探索ノードの優先度にしたがって探索を行う。なお、Randomized kd-tree の場合には全ての探索木の間で共通の priority queue を用いている。

priority queue は、クエリが属する葉ノードに到達するまでの間に優先度が計算され初期化される。葉ノードに到達した後は優先度の高いノードから順に探索を行い、未探索ノードの展開を行うたびに priority queue の値が更新される。本研究で用いた階層的 k-means tree では priority queue に保存する優先度として、クエリと分岐ノードのクラスタ中心点との距離を用い、その距離が小さいほど優先度を高く設定する。

4.2.3 近似近傍探索

このように priority queue を用いることでクエリの周辺の空間を効率的に探索する。さらに探索葉ノード数 L で探索を打ち切ることにより、近似的な探索にはなるものの全探索と比べて計算コストを大幅に抑えることができる。なお、探索葉ノード数 L は探索木の構築時に設定する precision に応じて決定される。また、近似近傍探索での計算コストはデータ数 n の log オーダーであり、探索葉ノード数 L に比例する。

5 実験および考察

実環境で撮影された全方位動画画像を用いて、実空間のモデリングと自己位置推定の実験を行った。これらの実験により提案手法の有効性を確認する。

5.1 実空間のモデリング

5.1.1 データセット

実空間のモデリングに用いた全方位動画画像は次のものを利用した。

- データセット枚数: 9040 枚
- 撮影デバイス: RICOH THETA m15
- 全方位画像の投影方法: 正距円筒図法
- 解像度: 480x240[pixel]
- 撮影間隔: 5 枚/s

この時系列画像は、全方位カメラを人が持ち歩きながら連続的に撮影したものである。また、このデータセットは屋内もしくは屋外に限定せずに、両方の環境で撮影されている。

5.1.2 実験パラメータ

実空間をモデリングする際の各種パラメータは表 1 のように設定した。

表 1: モデリングに用いたパラメータ

D_{surf} (ratio test における閾値)	0.7
$w_{sizeDefault}$ (PIRF 抽出時のデフォルトの窓幅)	3
n_{min} (PIRF 抽出数の下限)	40
n_{max} (PIRF 抽出数の上限)	200
ε (直近の観測を除外するためのオフセット値)	30
σ (正規分布関数におけるシグマ)	2.0
τ (Loop-Closure 検出の際の閾値)	0.015

5.1.3 PIRF 辞書の更新方法の比較

PIRF 辞書の更新方法に関する既存手法と提案手法とで、PIRF 辞書の要素数と Loop-Closure 検出数にどのような違いが現れるかを実験で確認した。なお、使用したデータセットに



(a)



(b)

図 3: 全方位画像の例

はラベル付が行われていないため、Loop-Closure の判定が正しく行われたかどうかは人手で判断している。このとき、 L_c から半径 1m 以内の位置に L_l が存在していると判断した場合に Loop-Closure が正しく行われているとした。

表 2: 実空間モデリング結果

	辞書更新の改善なし	辞書更新の改善あり
辞書の要素数	483,727	320,364
Loop-Closure 検出数	2,524	3,202
Loop-Closure 検出の正解数	2,523	3,196
precision	0.9996	0.9981

結果を見ると提案手法では既存手法と比べて辞書の要素数が減少しているのにも関わらず、Loop-Closure 検出数は増加している。このことから提案手法では自己位置推定に不必要な PIRF を登録しないような辞書の更新が行われていることが分かる。

5.2 近似近傍探索による位置ノード特徴ベクトルの生成

実空間のモデリングを行った後で PIRF 辞書のインデックス化を行い、PIRF 辞書を全探索した場合と近似近傍探索した場合とで、計算される位置ノード特徴ベクトルの値にどのような影響が出るのかを確認する。

ここでは全探索 (BF search) した場合の位置ノード特徴ベクトルを m_{bf} 、近似近傍探索 (ANN search) を行った場合の位置ノード特徴ベクトルを m_{ann} と定義し、この 2 種類のベクトルについて比較を行う。

また、2 つのベクトルを比較する指標として、この実験では Jaccard 係数, Recall, Precision を用いる。

$$Jaccard = \frac{\|m_{ann} \wedge m_{bf}\|_1}{\|m_{ann} \vee m_{bf}\|_1} \quad (16)$$

$$Precision = \frac{\|m_{ann} \wedge m_{bf}\|_1}{\|m_{ann}\|_1} \quad (17)$$

$$Recall = \frac{\|m_{ann} \wedge m_{bf}\|_1}{\|m_{bf}\|_1} \quad (18)$$

これら3つの指標を計算する際に m_{bf} と m_{ann} の次元が一致していない場合には、次元が小さい方のベクトルの末尾に0を追加して次元を揃えている。

Jaccard 係数は一般的に2つのバイナリベクトルの類似度を求める際に利用される。

また、Precision は近似近傍探索による探索結果のうち、最近傍点に適合する結果がどの程度の割合なのかを表す。

さらに、Recall は近似近傍探索による探索結果のうち、どの程度の最近傍点が再現されているかを表す。

以上の3つの使用を用いて2種類のベクトル m_{ann} , m_{bf} を比較するため、実空間のモデリングに用いたデータセットをテスト用のデータセットとして用いて実験を行った。Jaccard 係数などの値は、全データセット9040枚で得られた値を平均したものをを用いている。

表3: 近似近傍探索時の位置ノード特徴ベクトルの比較

	辞書更新の改善なし	辞書更新の改善あり
Jaccard	0.816	0.739
Precision	0.847	0.802
Recall	0.958	0.903

結果は表3に示す。結果をみると、既存手法と提案手法ともにバイナリベクトルの類似度を表す Jaccard が0.7~0.8程度となっており、全探索と近似近傍探索とで同一の特徴ベクトルが得られていないことが確認できる。

続いて Precision の値をみると既存手法では0.847、提案手法では0.802というように、近似近傍探索の結果の中に約15~20%の非最近傍点が含まれていることが分かる。

また、Recall の値は既存手法と提案手法ともに9割以上の値となった。これは近似近傍探索の結果のなかに、全探索での結果の9割以上が含まれていることを意味している。

以上、Jaccard, Precision, Recall の結果から、近似近傍探索では位置ノード特徴ベクトルの値が変わってしまうものの、全探索での結果がおおむね近似近傍探索の結果に含まれていることが分かる。特徴ベクトル同士の類似度の大小は(5)式中の $\|m_i \wedge m_j\|_1$ に依存する (j は辞書中の PIRF のインデックス)。Recall の値が大きいということは、近似近傍探索と全探索とで $\|m_i \wedge m_j\|_1$ の値 (特に異なる j 間での値の大小関係) が変わりにくいことを意味する。

5.3 特徴ベクトルの計算方法による比較

近似近傍探索 (ANN search) と全探索 (BF search) とで、位置ノード特徴ベクトルの計算時間や自己位置推定精度にどのような違いが生じるかを確認した。

5.3.1 特徴ベクトルの計算時間

位置ノード特徴ベクトルの計算時間は、実空間モデリングの際のデータセット数を変化させ、それぞれの辞書サイズの下で比較を行っている。この結果は図4のようになった。

全探索 (BF search) では位置ノード特徴ベクトルの計算時間が辞書の要素数に対して線形である一方、近似近傍探索 (ANN search) では辞書の要素数に対して計算時間に大きな変化は見られなかった。図5は図4を拡大したものである。priority search による計算コストが探索葉ノード数 L に比例しており、 L の値がインデックス化の度に異なっているため、正確な log のグラフにはなっていないが、上に凸なグラフになっていることが確認できる。したがって、全探索では実時間での自己位置

推定が困難な規模の環境でも、近似近傍探索では実時間での自己位置推定が可能であることがわかった。

5.3.2 自己位置推定の正解率

自己位置推定に関しては、推定した位置ノードと正解位置ノードとのノード間距離が一定の範囲内 (gap tolerance) にあるかどうかで判定しており、正解率は (正解回数)/(試行回数) とした。なお、この実験では、近似近傍探索と全探索との位置推定の差を調べるのが目的であるため、テストデータセットには実空間のモデリングに用いたデータセットと同様のものを用いた。以上のような設定のもと、結果は図6および図7のようになった。

図6および図7の結果から、全探索と近似近傍探索とでは位置推定の精度にほとんど差が生じないことが確認できる。また、図6および図7の比較を行うと、提案手法の辞書更新では gap tolerance が0の場合に、正解率が既存手法と比べて10%ほど低くなった。しかし、gap tolerance の値を1,2と大きくしていくと差が生じなくなることから、提案手法では隣接する位置ノード同士の類似度が既存手法と比べて高くなっていると考えられる。

図4および、図6,7の結果から、近似近傍探索により計算コストを大幅に抑えることができ、さらに全探索と位置推定の精度がほぼ変わらないことが確認できた。

5.4 未知データに対する自己位置推定

実空間のモデリングに用いたデータとは異なるデータセットをテストデータとして自己位置推定を行った。その結果は表4のとおりである。なお、自己位置推定が正しく行われているかどうかの判定は、Loop-Closure 検出のときと同様に人手で判断している。また、自己位置推定の正解率は (正解回数)/(テストデータセットの枚数) である。

表4: 未知データに対する自己位置推定の正解率

	全探索	近似近傍探索
辞書更新の改善なし	4416/4856 = 0.9094	4509/4856 = 0.9248
辞書更新の改善あり	4547/4856 = 0.9364	4597/4856 = 0.9467

この結果から、近似近傍探索や辞書の要素数を削減したことによる自己位置推定精度の低下はほとんど生じないことが確認できた。なお、全探索に比べて近似近傍探索での正解率が上昇しているのは、ratio test における (本来の) 2番目に近い特徴点が探索されず、3番目 (もしくはそれ以降) に近い特徴点が2番目に近い特徴点として判定されているためであると考えら

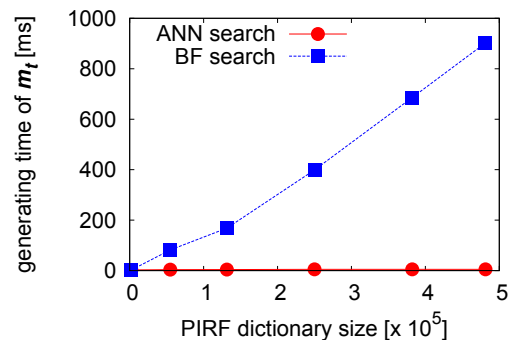


図4: PIRF 辞書の要素数と位置ノード特徴ベクトルの計算時間との関係

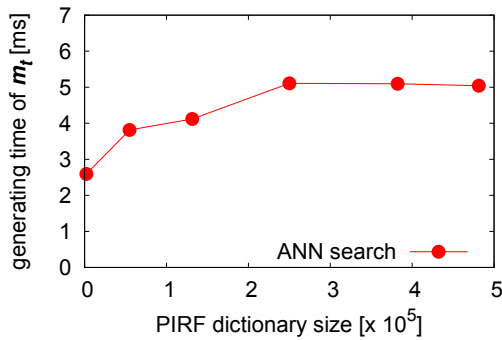


図5: PIRF辞書の要素数と位置ノード特徴ベクトルの計算時間との関係 (拡大)

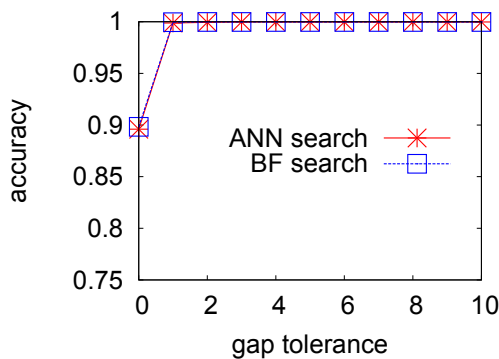


図6: 辞書更新を改善しない場合の位置正解率

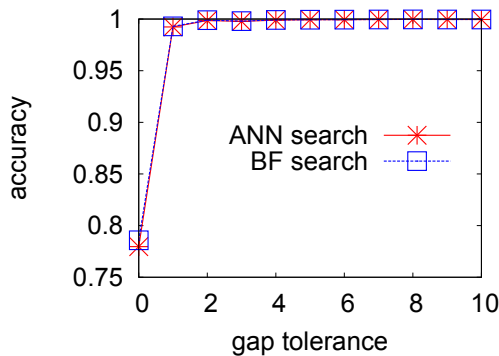


図7: 辞書更新を改善した場合の位置正解率

れる。もともと ratio test は画像内の局所画像特徴点のマッチング適合率を高めるために再現率を犠牲にした手法である。ratio test でマッチングが棄却された特徴点であっても実際にはマッチしている可能性があり、本実験で正解率が上昇したのは近似近傍探索によって特徴点同士のマッチングにおける再現率が上がったためだと考えられる。

6 おわりに

本稿では、PIRF を用いて自己位置推定を行う既存手法の改良方法を提案した。

PIRF は画像から抽出される SURF 集合を時系列情報によりフィルタリングし、外乱を取り除く手法である。PIRF 辞書の

更新方法の改善は、シーン識別に不必要な項目を PIRF 辞書の登録の際に除外することを狙っており、PIRF 辞書の要素数を減らすことで、計算機リソースの有効活用を目的としている。

一方 PIRF 辞書のインデックス化は、PIRF 空間を効率的に探索することで、位置ノードの特徴空間への写像時間短縮を目的とした手法である。

PIRF 辞書の更新は位置ノードの特徴空間を構築することと同義であり、シーン識別を行うために必要十分な特徴項目が選択されることが理想的である。そこで本研究では PIRF 辞書の更新時に一つずつ登録判定を行い、辞書中に類似した PIRF が複数登録されることを防いでいる。しかし、この手法の結果構築された位置ノードの特徴空間がシーン識別に必要な特徴項目により構成されているかの保証は行えていない。この点は今後考慮すべき課題である。

本研究では、PIRF 辞書の更新方法を変更することによって、隣接する位置ノード間での類似度が高くなることが確認された。すなわち提案手法では既存手法に比べて実空間のモデルの抽象度が高くなっており、このことは未知データに対する汎化性能の向上につながる。加えて PIRF 辞書の要素数が抑えられることは位置ノード特徴空間への写像時間を抑えること以外にも、PIRF 辞書が専有するメモリを節約する効果も期待される。

PIRF 辞書のインデックス化については、全探索により位置ノード特徴ベクトルを計算する場合と比べて近似近傍探索では計算時間が短縮された一方で、シーン識別の精度には大きな差が生じないことが確認された。むしろ、ratio test によるマッチングの条件が近似近傍探索により緩められることによりシーン識別の精度に向上が見られた。すなわち、提案手法では従来手法と比較して自己位置推定の精度を維持しながら、大規模環境においても実時間での自己位置推定が行えると期待される。

参考文献

- [1] 橋本秀紀, 新妻実保子, 佐々木毅: 空間知能化 インテリジェント・スペース, 日本ロボット学会誌, Vol.23, No.6, pp.674-677, 2005
- [2] 杉山貴彦, 濱上知樹: 動画像を用いた自律的な知能化空間の構築, 計測自動制御学会, システム・情報部門学術講演会 (SSI2008) 講演論文集, pp.503-506, 2008
- [3] Aram Kawewong, Sirinart Tangruamsub and Osamu Hasegawa: Position-invariant Robust Features for Long-term Recognition of Dynamic Outdoor Scenes, IEICE Transactions on Information and Systems, Vol.E93-D, No.9, pp.2587-2601, 2010
- [4] A.Kawewong, N.Tongprasit, S.Tangruamsub, O.Hasegawa,: Online and Incremental Appearance-based SLAM in Highly Dynamic Environments, International Journal of Robotics Research (IJRR), 2010
- [5] Aram Kawewong, Tongprasit Nopparit and Osamu Hasegawa: PIRF-Nav 2.0: Fast and Online Incremental Appearance-Based Loop-Closure Detection in an Indoor Environment, Elsevier Robotics and Autonomous Systems, 2011
- [6] Marius Muja, David G. Lowe: Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration, in International Conference on Computer Vision Theory and Applications (VISAPP'09), 2009
- [7] Marius Muja, David G. Lowe: Scalable Nearest Neighbor Algorithms for High Dimensional Data, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.36, pp.2227-2240, 2014