

連続画像比較による事前マップ屋内位置推定の精度向上

高橋 和太郎¹ 高橋 淳二² 兼子 朋也¹ 戸辺 義人³

概要：本研究では、事前マップを用いた屋内位置推定インフラ (Universal Map) における探索範囲限定をすることによる精度向上を目指す。Universal Map はサーバクライアント方式を取っており、サーバでは屋内環境を3次元ワイヤフレームで構築し、事前マップを様々な視点で投影することで線分画像を生成する。生成した線分画像をデータベースとし、クライアントから送られてきたセンサ画像を線分処理した画像と線分比較をすることで位置推定を行っていた。今回、連続的に取得した画像データを特徴点比較を行い、クライアントの移動量を推定することによってデータベース探索範囲を限定する。50枚のデータセットを用いた実験において従来手法では最大誤差が76.11mであったのに対し、提案手法では最大誤差が1.42mとなり精度向上したことを確認した。

Accuracy Improvement of Indoor Localization Using Pre-Map by Continuous Image Comparison

WATARO TAKAHASHI¹ JUNJI TAKAHASHI² TOMOYA KANEKO¹ YOSHITO TOBE³

1. はじめに

近年、スマートフォンを始めとしたモバイル型コンピュータが多く開発され広く普及してきている。それと同時にこれらの端末を用いた自己位置推定を利用したアプリケーションやサービスが多く提供されている。また、ロボット技術の発達に伴い、人間と居住空間を共にしたり、工場内で働いたりする自律移動ロボットの導入が期待されている。こういったロボットを安全に運用していくためにはロボット自身が自分の位置を知ることが必要不可欠である。そのため、高精度な屋内での位置推定技術が求められている。

屋外位置推定技術として多く利用されているものはGPS (Global Positioning System) である。GPSは人工衛星の電波を利用しており、電波の届かない屋内では正確な位置推定を行うことができない。

屋内位置推定ではWi-Fi FingerprintやBLE (Bluetooth Low Energy) などの受信信号強度 (RSSI: Received Signal Strength Indicator) やセンサフュージョンを用いた手法がある [1][2]。しかしこれらの手法はインフラを用いている

のでインフラの設置やメンテナンスにコストがかかってしまい、精度とコストのトレードオフが発生してしまう。また自律移動ロボットを安全に運用する上での必要とする精度の位置推定ができていない。

もう1つの屋内位置推定の手法としてSLAM (Simultaneous Localization and Mapping) がある [3]。SLAMは環境マップと自己位置推定を同時に行う技術であり、高精度な自己位置推定を行うことができる。しかし、環境マップに誤差が生じると自己位置推定にも誤差が生まれ、誤差の蓄積が生じてしまうことや、高価なセンサを用いていることから非常にコストがかかるという問題がある。

これらの問題を解決するために我々はすでに Universal Map (UMap) という事前マップと画像を用いた低演算低コストの位置推定インフラを開発した [4]。しかし UMap の問題として類似度の高い内部構造環境では全く違う位置推定がなされてしまうことがある。そこで本論文では、連続画像比較によって探索範囲を限定することで精度向上を目指す。またそのために時間的整合性の取れた屋内位置推定手法を提案する。本論文では、システムの設計、実装、評価結果について述べる。

¹ 青山学院大学理工学研究科専攻知能情報コース

² 鹿児島大学学術研究院理工学域工学系機械工学専攻

³ 青山学院大学理工学部情報テクノロジー学科

2. 関連研究

位置推定において推定結果が異なるという問題はあり、それを解決する研究は行われてきた。

和田らはレーザレンジスキャナと Wi-Fi Fingerprint を用いた位置推定手法を提案している [5]。誤差の多かった従来の Wi-Fi Fingerprint 法にレーザレンジスキャナ (LRS) の高い測距性能を利用することで誤差の小さい位置推定を実現している。2 種類の位置推定技術を組み合わせることにより従来の Wi-Fi Fingerprint の位置推定手法よりも精度向上を実現した。

また Grisetti らはロボットのオドメトリなどのセンサ情報を用いてロボットの存在位置を確率的に絞り込み位置推定をする Graph-Based SLAM という手法を提案している [6]。Graph-Based SLAM とは環境地図を作成する前段階として、ロボットの移動と観測の関係性をグラフとして表し、それを用いて地図作成とロボットの位置推定を行う手法である。グラフのノードとしてよく用いられているのは時刻 t の姿勢 x_t と次の時刻 $t+1$ の姿勢 x_{t+1} 、またはそれぞれの時刻で観測されたランドマークの位置である。グラフを用いることでロボットの存在確率を求めることができ、従来手法で求めた地図や位置推定をオフラインで修正を行う。

本研究では Graph-Based SLAM の理論を参考にし、時間的整合性の取れた位置推定を可能にする。

3. Universal Map (UMap)

UMap とは線分をランドマークとした事前マップを用いた位置推定インフラである。UMap はサーバクライアント方式を取っており、1 枚のカメラ画像から位置推定を行う。

事前にサーバ内で位置推定環境を 3 次元ワイヤフレームで構築した事前マップ (図 1) を様々な視点で投影した 2 次元画像 (図 2) を生成する。この 2 次元画像群をデータベースとする。この際、2 次元画像それぞれに事前マップにおける生成された地点の 3 次元座標を紐付けている。

クライアントから送信されたセンサ画像を LSD (Line Segment Detector) [7] を用いて線分処理を行い、それを入力画像とする。入力画像とデータベース画像の論理積を求め、最もマッチングしたデータベース画像を生成した際に紐付けた地点の 3 次元座標を推定位置とし、クライアントに送信することで位置推定をしている。マッチングには UMap における自己評価関数 (式 1) を用いている。 $A()$ は非ゼロピクセル数、 DB_i は i 番目のデータベース画像、 SEN は入力線分画像、 LC は DB_i と SEN の論理積、 M_i は i 番目のデータベース画像のマッチング率をそれぞれ表している。 M_i の値が最も大きいものを最もマッチングしたデータベース画像としている。

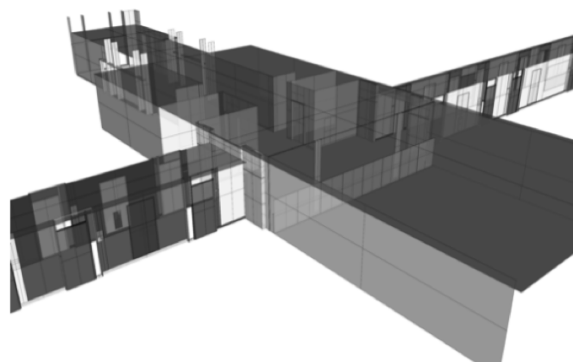


図 1 青山学院大学相模原キャンパス O 棟 5 階廊下の事前マップ

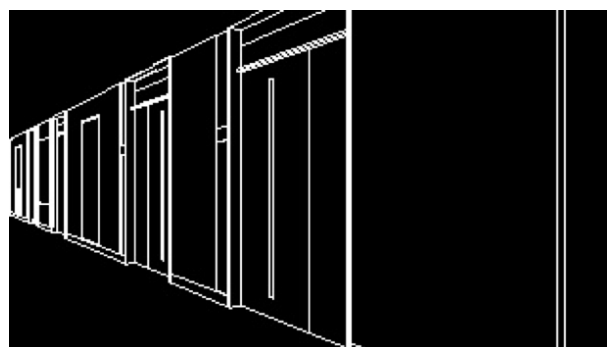


図 2 事前マップから生成された任意視点画像

$$M_i = \frac{A(LC)}{A(DB_i)} + A(SEN) - A(LC). \quad (\text{式 1})$$

図 3 は UMap においてマッチングを行った際の結果である。緑線がデータベース画像、青線がセンサ画像に線分処理を行った入力画像、白線が共通する線分である。

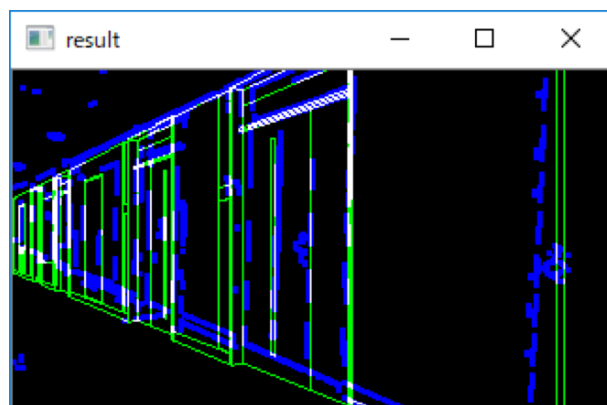


図 3 UMap によるマッチング

UMap ではサーバで事前マップを所持し、位置推定のための演算もサーバで行われるためクライアントにおける演算、電源資源といったものを節約することができる。

しかし、現状の UMap では建物の壁、床、天井のみの事前マップを用いており、他の後から追加されたポスターや家具といった情報がない。また、画像 1 枚でのみ位置推定を行っているため、内部構造が似ている環境では全く別の位置が推定されてしまうことがある。

4. Consistency Adjustment (CA)

この問題を解決するために、連続的に取得されたセンサ画像の時間的整合性を考慮する Consistency Adjustment (CA) システムを提案する。図 4 は CA の処理のフローチャートである。緑色が UMap の処理であり、青色が今回 UMap に追加する CA の処理を表している。

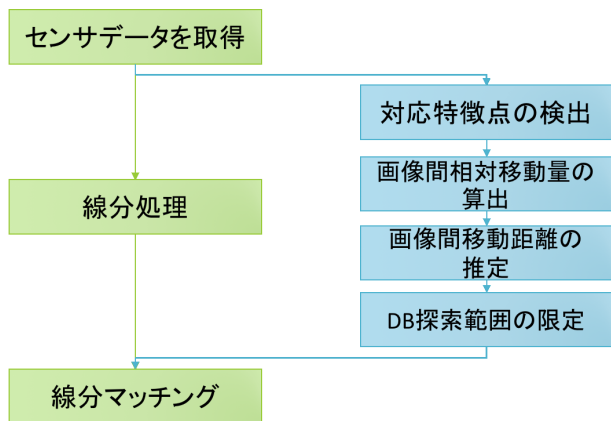


図 4 処理の流れ

特徴点検出には、Loew の SIFT (Scale Invariant Feature Transform)[8] や Herbert Bay らの SURF (Speed-Up Robust Features)[9], Alcantarilla らの KAZE[10] があるがその中でも、非線形、非等方的なスケール空間を用い、かつロバスト性の向上と高速化がなされた AKAZE[11] を本研究では用いる。AKAZE を用いて検出した特徴点を画像間移動距離を求めるためのランドマークとする。図 5 は実際にセンサ画像に対し、AKAZE を用いて特徴点検出を行ったものである。色がついている点が検出された特徴点である。

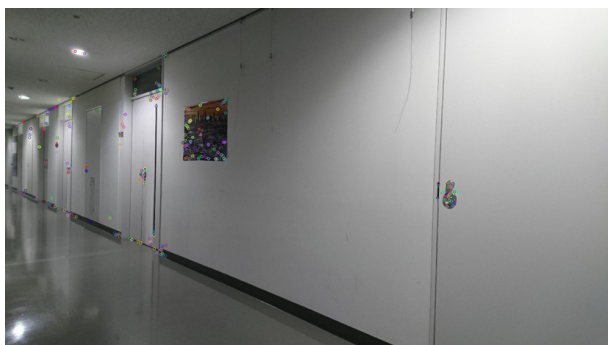


図 5 AKAZE を用いた特徴点検出

特徴点検出を行った連続する 2 枚の画像に対し、総当たりマッチングを用いて対応特徴点を検出する。図 6 は実際に連続するセンサ画像に対応特徴点検出を行ったものである。線で結ばれているのが対応した特徴点である。しかし中には誤対応しているものがある。これらを取り除くために多数決を行い選定する。それぞれの対応点の線がどのよ

うな引かれ方をしているのを見、最終的に多く同じ引かれ方をしている線の対応点のみを残すという手法である。図 7 は実際に図 6 に対し、多数決を行ったものである。

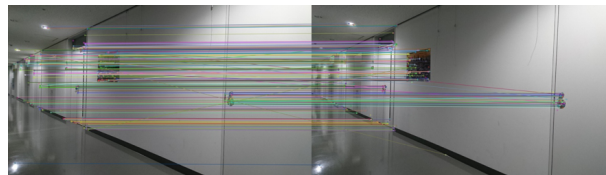


図 6 総当たりマッチングを用いた対応特徴点の検出

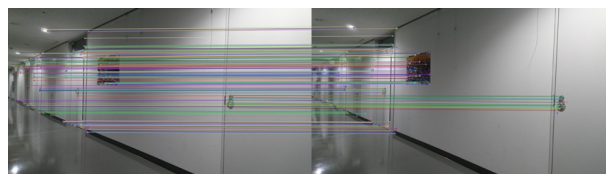


図 7 図 6 に対して多数決によって選定した対応特徴点

この対応特徴点を用いて、図 8 のように画像間相対移動量を算出していく。それぞれの対応特徴点の重心 (図 8 赤点) と画像の中心からのピクセル数を求める。それらの差を画像間相対移動量とする。図 8 の場合、画像間相対移動量は $n_N - n_{N-1}$ である。

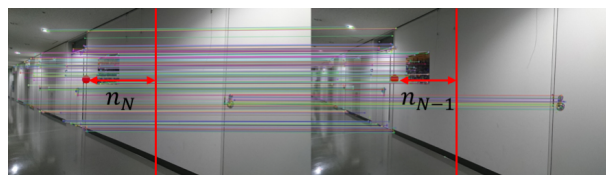


図 8 画像間相対移動量の算出

求めた画像間相対移動量と今回の研究ではセンサ画像を取得した際に簡易なレーザ距離計を用いて壁からの距離を用いて画像間移動距離を推定していく。推定には図 9 の模式図を用いた。

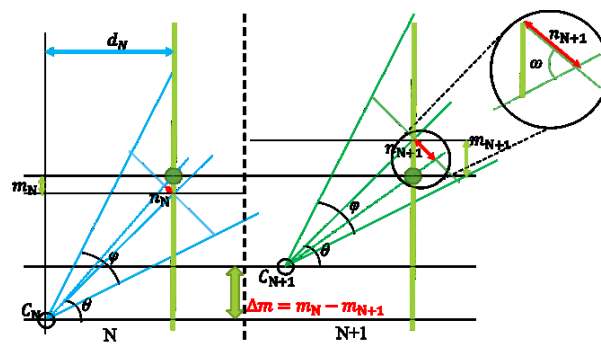


図 9 模式図

太線が壁を表しており、壁にある点が特徴点重心である。カメラの角度 θ 、画角 ϕ 、壁までの距離 d を用いることで相対移動量から移動距離 Δm を推定できる。 m_N, m_{N+1}

を求める式を式 2, 3 に示す。また式中の $width$ は画像の横のピクセルサイズである。

$$\omega = 90 - \arctan\left(\frac{|n_N| * \tan(\phi)}{width}\right). \text{ (式 2)}$$

$$m_N = d * \tan(\theta + 90 - \omega) - d * \tan(\theta) \text{ (式 3)}$$

CA によって推定した移動距離を UMap で位置推定する際に、存在範囲として限定することによって時間的整合性を考慮し、位置推定精度向上を目指す。また CA の推定誤差を考慮するために、存在範囲として用いる距離は移動距離に誤差の平均値と標準偏差を足し合わせたものを用いる。

5. 評価実験

評価実験は青山学院大学相模原キャンパス O 棟 5 階にて実験を行った。図 10 は O 棟 5 階の見取り図である。

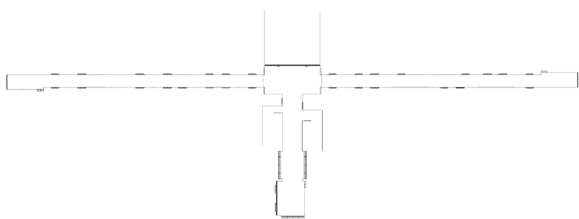


図 10 青山学院大学相模原キャンパス O 棟 5 階の見取り図

今回実験に用いるデータベースとして O 棟 5 階の事前マップから $0.0 < x < 90.0 \text{ m}$, $-1.5 < y < 4 \text{ m}$, $z = 1.14 \text{ m}$ の範囲で、 x 軸方向は 0.10 m 刻み、 y 軸方向は 0.20 m 、 z 軸方向は固定とし 2 次元画像を生成した。図 11 はデータベースの生成範囲である。

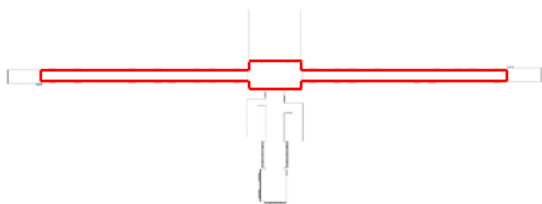


図 11 データベース生成範囲

また各視点座標において、注視点の水平方向を $45 < \theta < 315 \text{ deg}$ の範囲で 90 deg ずつ回転させた画像を生成し、全 36032 枚のデータベースを用いた。

今回の実験ではクライアントカメラ端末として、Xperia X Performance を用いた。また壁からの距離を測定するレーザ距離計には KINGTOP レーザ距離計 SW-E100 を用いた。データをとる際にはそれぞれの機器を図 12, 図 13 のように台車に固定し、撮影間隔は $0.1 \text{ m} \sim 2.0 \text{ m}$ の間でランダムに撮影し、全 50 枚の真値付きデータセットを作成した。また本実験では大量の画像を扱うため、比較的高画



図 12 実験装置



図 13 実験装置取り付け方法

質であり、ファイルサイズが小さい JPG を画像のフォーマットとして用いる。図 14 は真値データを取得した範囲である。

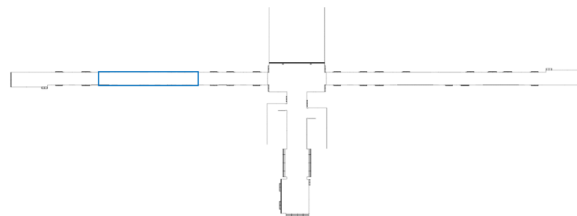


図 14 真値データ取得範囲

実験方法としては、作成したデータベース画像と真値データセットを用いて従来手法 (UMap), 提案手法 (UMap+CA) でそれぞれ位置推定を行う。実験の手順としては、UMap で位置推定を行う。その際に UMap に搭載されている自己評価関数を用いてマッチング結果が一番高いものを選択する。その後、マッチング結果が一番高かったものを基準とし、UMap+CA を用いてそれぞれのセンサ画像の画像間移動距離を推定し、探索範囲を決め、位置推定を行う。

図 15 はそれぞれの実験結果と真値データを見取り図にプロットしたものである。緑点が UMap, 赤点が UMap+CA, 青点が真値である。

またそれぞれの手法の推定誤差の棒グラフと累積割合グ

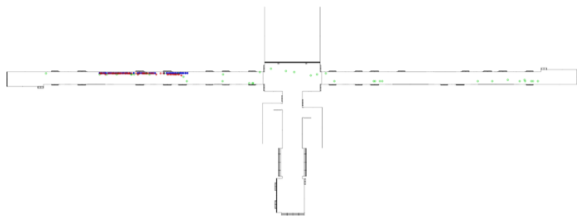


図 15 実験結果

ラフを図 16, 図 17 に示す。

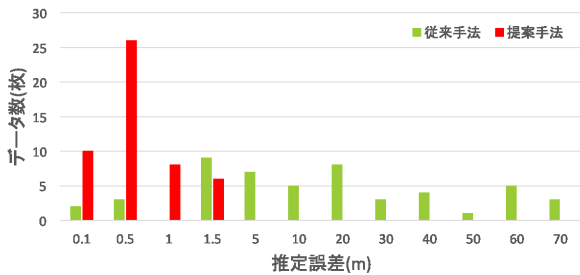


図 16 誤差の比較 (棒グラフ)

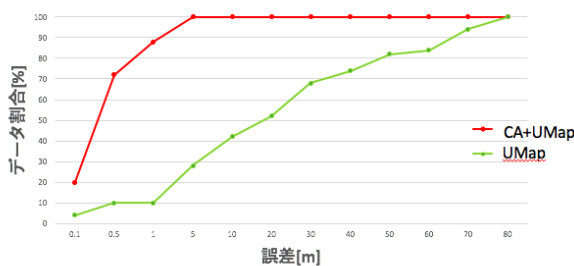


図 17 誤差の累積割合グラフ

UMap での位置推定誤差は最大で 76.11 m という結果になった。これは実験環境の内部構造が左右対称となっており、同じようなデータベースが別の箇所でも生成されてしまい、そちらにヒットしてしまったためだと考えられる。また図 15 を見ると反対側以外にも多く位置推定されていることがわかる。これは UMap の持っている事前マップが建物の線分のみで後から追加された家具やポスターといった情報を持っていない。それを元にデータベースを作成したため、マッチングする際にデータベースにはない線分情報によって誤って推定したものだと考えられる。UMap+CA では位置推定誤差は最大で 1.42 m という結果となった。これにより今回提案した手法は従来の手法よりも精度が向上したといえる。

また図 17 のグラフをみると UMap では 0.5 m 以内誤差推定が 10% しかないが、UMap+CA では 70% を超えていることがわかる。これからも UMap に CA を用いることで精度向上したと言える。

6. まとめ

本論文では、連続するセンサ画像の移動距離を特徴点と撮影位置の壁までの距離を用いて推定し、探索範囲を限定し、時間的整合性の取れた位置推定ができるアルゴリズム Consistency Adjustment を提案した。評価実験ではこのアルゴリズムを Universal Map に搭載し、従来の Universal Map と比較したところ、最大 76.11 m あった誤差を 1.42 m にまで小さくすることができた。また 0.5 m 以内誤差推定も 10% から 70% にまで向上させることができた。

今後の課題としては、今回最大で 1.42 m という誤差という結果になったが、これは自律移動ロボットを運用していくには決して小さくない誤差である。その誤差の原因としては、限定した探索範囲から真値が漏れてしまったものがあるためである。これにより誤差の蓄積が起きてしまった。これを解決する 1 つの方法としては、今回 50 枚を一度に推定したが、より少ない枚数で行うことで解決できると考えられる。また今回用いた手法はオフラインで処理を行うものであり、リアルタイムに位置推定を行うことができない。そこでリアルタイムでも行えるオンラインでの処理を実現していくことで実際にロボットに搭載した際により高精度な自己位置推定をすることができると考えられる。

参考文献

- [1] 吉澤菜津子, 遠藤貴裕, 永見健一: “屋内位置情報における推定技術の開発と新しいサービスの展開について”, Intec Tech J, 13 号, pp. 44-51, 2013.
- [2] 渡辺雄太, 松本倫子, 吉田紀彦: “無線モバイル端末の Wi-Fi Direct による電波強度を用いた位置推定”, 情報処理学会, 2013.
- [3] 原祥亮: “ベイズ理論に基づく移動ロボットの自己位置推定と地図生成に関する研究”, つくばリポジトリ, 12102 甲第 7298 号, 2015.
- [4] 徳永悠希, 高橋淳二, 兼子朋也, 花見唯, 戸辺義人: “線分マッチングに基づく位置推定アルゴリズム GPU 並列演算による高速化”, 研究報告モバイルコンピューティングとハイパーページシステム (MBL), vol. 21 no. 2017-MBL-82, pp.1-8, 2017.
- [5] 和田悠佑, 山口 弘純, 東野輝夫: “レーザレンジスキャナと Wi-Fi Fingerprint を併用した歩行者の位置推定手法の提案”, モバイルコンピューティングとユビキタス通信研究, 2013.
- [6] G. Grisetti, R. Kummerle, C. Stachniss and W. Burgard: “A Tutorial on Graph-Based SLAM”, Vol. 2, No. 4, pp. 31-43, 2010.
- [7] Von Gioi, Rafael Grompone, et al. “LSD: a line segment detector,” Image Processing On Line 2 35-55, 2012.
- [8] D.G. Lowe. : “Object recognition from local scale-invariant features” In Proceedings of the Seventh IEEE International Conference on Computer Vision, Vol. 2, pp. 1150 - 1157, 1999.
- [9] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool: “SURF: Speeded Up Robust Features,” computer vision-ECCV Lecture Notes in Computer Science, 2006.
- [10] Pablo F. Alcantarilla, Adrien Bartoli, and Andrew J. Davison.: “KAZE features” European Conference on

Computer Vision (ECCV). Springer Berlin Heidelberg, 2012.

- [11] P. F. Alcantarilla, J. Nuevo and A. Bartoli: “ Fast Explicit Diffusion for Accelerated Features in Nonlinear Scale Spaces, ” British Machine Vision Conf. (BMVC). Bristol, UK, 2013.