

線分特徴とエッジのマッチングによる全天球カメラの位置推定

○陽 東旭 †, 樋口 寛 †, Sarthak PATHAK ‡, Alessandro MORO †, 山下 淳 †, 浅間 一 †

○ Dongxu YANG † Hiroshi HIGUCHI † Sarthak PATHAK ‡ Alessandro MORO †
Atsushi YAMASHITA † and Hajime ASAMA †

† : 東京大学, yangdongxu@robot.t.u-tokyo.ac.jp

‡ : 中央大学

<要約>環境の3Dラインマップと全天球カメラで取得した画像の3D-2Dマッチングを利用し、環境中から大域的に位置推定する手法を提案する。画像の線分検出の際の誤差を避けるため、線分検出をせずに、検出されたエッジピクセルを直接利用し、環境の3Dラインマップとマッチングを行う。提案手法はオクルージョンに対してロバスト性が高いことを実験により確認できた。

<キーワード> 位置推定, 全天球カメラ, 線分特徴, オクルージョン

1 序論

カメラの位置姿勢推定はさまざまな分野において非常に重要な課題である。例えば近年、インフラや工場の点検など、さまざまな目的でより一般的にロボットが使用され始めている。点検場所に着くためには、ロボットの位置姿勢推定が必要である。GNSSを使用した位置推定方法は屋外で効率的に機能するが[1]、屋内環境では信号が届かないという問題点がある。そこで、カメラを使用した屋内でのロボットの位置姿勢推定に関する研究が多くある。

画像から位置姿勢を推定する方法では、カメラの移動追跡を行う手法と1枚の画像の位置姿勢を環境中から大域に推定する手法がある。カメラの移動追跡を行う手法では、Visual Odometry や Simultaneous Localization and Mapping (SLAM) [2] [3] [4] [5] などによる研究があるが、カメラの移動量にともなって累積誤差が発生する問題やカメラの初期値が推定できない問題がある。それに対して、環境中から大域的に位置推定する手法は累積誤差が発生しないことや、初期位置が必要ではない利点があり、さらに、累積誤差の校正や初期位置を特定する目的にも使える。そこで本研究は、1枚の画像の位置姿勢を環境中から大域に推定する手法に着目する。

1枚の画像の位置姿勢を環境中から大域的に推定す

る手法では、機械学習を利用する研究 [6] [7] が多数あるが、画像のデータセットを作るために大量の画像を事前に用意する必要がある。環境のラインマップと画像から検出した線分特徴とマッチングする位置姿勢推定をする手法 [8] もあるが、線分特徴がオクルージョンの影響で検出が失敗すると推定の精度が低下する問題点がある。そこで、本研究では、線分検出際の誤差を避けるため、画像の線分検出せずに、検出されたエッジピクセルを直接利用し、環境のラインマップとマッチングして推定を行い、オクルージョンにロバストな手法を提案する。

大域的アプローチの中でも、姿勢推定については、消失点を利用する方法など、高速かつ正確に推定できる手法があるが、位置推定については未だに難しい。従って、本研究は位置推定に着目する。

また、カメラの視野が広いほど、一度に取得できる情報が多いため、本研究では360degの視野がある全天球カメラを使用する。

2 提案手法

2.1 問題設定

本研究では、1枚の画像の位置姿勢を環境中から大域に推定する手法を提案する。大域的アプローチでは、リアルタイムでカメラの移動追跡を行わないため、アプリケーションとしてはオフラインで行う作業を想定

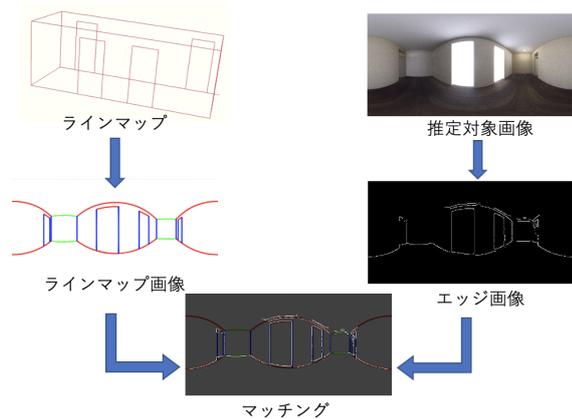


図 1 アプローチの全体像

する。また、事前準備をより効果的にするため、環境のモデルを参照情報として利用する。屋内環境では、建設用の CAD モデルなどの 3 次元モデルを入手可能な場合や、建設用の CAD モデルがない場合にもレーザーレンジファインダ等で取得可能な場合を想定する。従って、環境の 3 次元モデルを参照情報として利用する。

2.2 アプローチ

3 次元モデルと全天球カメラ画像の比較を行う 3D-2D マッチング手法を構築する。本研究では、屋内環境を対象とするため、壁と床の交線など、線分が多数存在することから、線分特徴を用いる。3 次元モデルから線分だけを抽出したラインマップを事前に準備する。

アプローチの全体像を図 1 に示す。ラインマップから、任意のある視点のラインマップ画像を生成する。多数の位置でのラインマップ画像を生成し、その中から推定対象画像からエッジ抽出をして作成したエッジ画像と最もマッチングするラインマップ画像を探す。2 つの画像の線分特徴がマッチングしたとき、同一視点から撮られた可能性が最も高いことを利用して位置推定を行う。

2.3 線分特徴とエッジのマッチング

2.3.1 全天球カメラ画像

まず、本研究で使用する全天球カメラの投影モデルについて説明する。全天球カメラ投影モデルを図 2 に示す。全天球カメラの投影面は、光学中心を中心とする単位球面であり、単位ピクセル $\hat{\mathbf{x}} = [x, y, z]^T$ で構成されている。全天球カメラ画像は、空間内の 3 次元

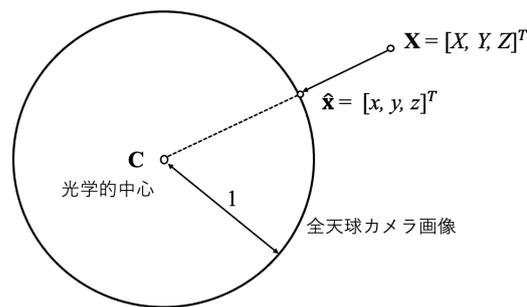


図 2 全天球カメラの投影モデル

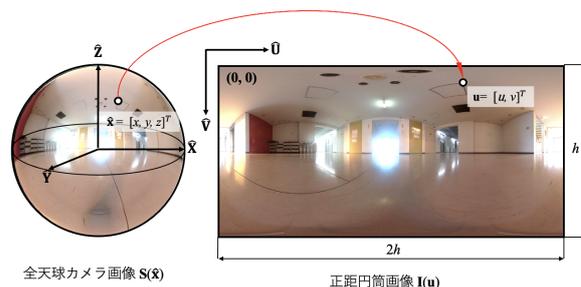


図 3 全天球カメラ画像と正距円筒画像

点 $\mathbf{X} = [X, Y, Z]^T$ の情報を球面上に投影した画像である [9]。全天球カメラ画像は単位球面上にあるため、スケールは全て同じになり、マッチングの際のスケールによる問題を考慮する必要がない利点がある。

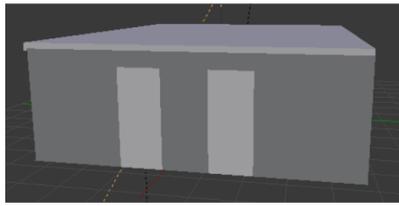
本研究では、マッチングは全天球カメラ画像の球面上に行う。そのため、正距円筒画像を全天球カメラ画像に変換する必要がある。全天球カメラ画像とそれに対応する正距円筒画像を図 3 に示す。全天球カメラ画像の各ピクセルと正距円筒画像の対応したピクセルの変換を以下の式に示す。

$$x = \sin \frac{\pi u}{h} \sin \frac{\pi v}{h}, \quad (1)$$

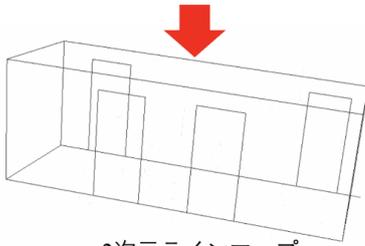
$$y = \cos \frac{\pi u}{h} \sin \frac{\pi v}{h}, \quad (2)$$

$$z = \cos \frac{\pi v}{h}, \quad (3)$$

ただし、 $\mathbf{u} = [u, v]^T$ は正距円筒画像の座標である。 h は正距円筒画像の高さである。

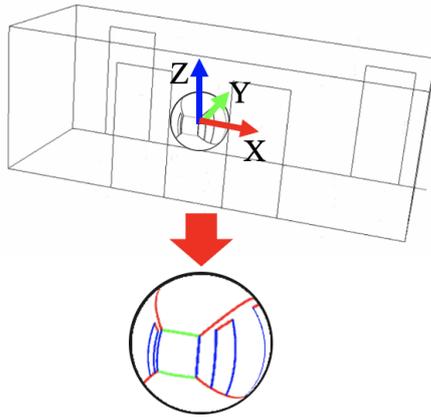


3次元モデル



3次元ラインマップ

図 4 3次元ラインマップ



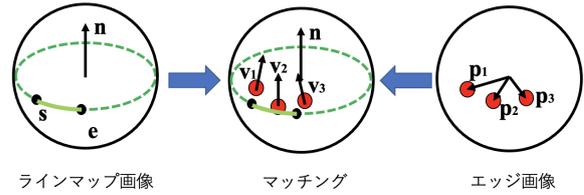
ラインマップ画像
(全天球カメラ画像)

図 5 ラインマップ画像の作成

2.3.2 ラインマップ画像の作成

環境の3次元モデルと3次元ラインマップの例を図4に示す。3次元ラインマップは事前に準備するため、3次元モデルから3次元ラインマップの変換は手動で行う。

3次元ラインマップからラインマップ画像を作成するプロセスを図5に示す。全天球カメラをある視点に置き、3次元ラインマップにある線分を全天球カメラに投影することで、その視点のラインマップ画像を作成することができる。図5のラインマップ画像では、 x , y , z の3方向をそれぞれ赤、緑、青で表示している。



ラインマップ画像

マッチング

エッジ画像

図 6 線分特徴とエッジのマッチング

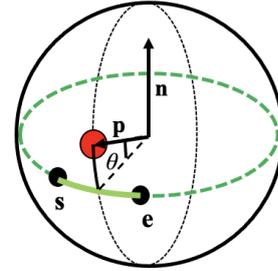


図 7 線分特徴とエッジピクセルの距離

2.3.3 エッジ画像の作成

本研究では、推定対象画像からCanny法[10]によるエッジ検出でエッジ画像を作成する。

2.3.4 線分特徴とエッジのマッチングの評価関数

ラインマップ画像の線分特徴とエッジ画像のエッジピクセルのマッチングを図6に示す。マッチングは全天球カメラ画像の球面上に行う。線分特徴は始点、終点、そして通る大円の法線ベクトル \mathbf{n} で表す。 \mathbf{p} はエッジピクセルのベクトルを表し、 \mathbf{v} はエッジピクセルの球面勾配ベクトルを表す。

マッチングを行うためには、まず線分特徴とエッジピクセルの対応付けが必要である。ここでは、エッジピクセルを一番近い線分特徴に対応付ける。全天球カメラ画像上の距離は、端点のベクトルのなす角度で表すことができる。線分特徴とエッジピクセルの距離は、エッジピクセルのベクトル \mathbf{p} と線分特徴が通る線のなす角度 θ で表す(図7)。対応付ける際は、エッジピクセルとラインマップ中の全線分について距離を計算し、距離が一番小さい線分に対応付ける。ただし、一番小さい距離でも、ある閾値 $\hat{\theta}$ より大きい場合、対応する線分特徴がないとし、そのエッジピクセルを外れ値だと判定する。

マッチングしたとき、エッジピクセルの球面勾配ベ

クトル \mathbf{v} は、線分特徴と垂直になる。線分特徴の法線ベクトル \mathbf{n} も線分特徴と垂直であることから、エッジピクセルの球面勾配ベクトル \mathbf{v} は法線ベクトル \mathbf{n} と同じ方向になる。従って、以下のように評価関数 F を定義する。

$$F_i = \begin{cases} \|\mathbf{v}_i \times \mathbf{n}_i\| & (|\theta_i| < \hat{\theta}) \\ \pi & (\text{otherwise}) \end{cases}, \quad (4)$$

$$F = \sum_{i=1}^N F_i, \quad (5)$$

ただし、 N はエッジピクセルの数であり、 \mathbf{v}_i は i 番目のエッジピクセルの球面勾配ベクトルであり、 \mathbf{n}_i は i 番目のエッジピクセルと対応する線分特徴の法線ベクトルである。 θ_i は i 番目のエッジピクセルと対応する線分特徴のなす角度であり、 $\hat{\theta}$ は外れ値を判定する角度の閾値である。

エッジピクセルが外れ値であると判断された場合、球面上の距離を表す角度の最大値、つまり直線を球面上に投影した際の角度 π に設定する。

2.4 最適化手法を使った位置推定

本研究では、式 (4)、(5) を評価関数とし、パーティクルフィルタ [11] とレーベンバーグ・マーカート法 [12] の組み合わせを使って推定を行っている。

パーティクルフィルタは大域的推定に使えるが、パーティクルの生成にはランダム性があり、真値の近くにパーティクルが生成されなかった場合の推定誤差が大きくなる可能性がある。

レーベンバーグ・マーカート法は計算速度が早く、性能の良い手法であるが、ローカルミニマムに落ちると推定が失敗する欠点がある。そのため、真値と離れた位置を初期位置とした推定はローカルミニマムに落ちる可能性が大きく、大域的推定には向いていない。それに対して、真値と近い位置を初期位置とした場合、ローカルミニマムに落ちることなく正しく推定することができる可能性が大きい。

以上から、パーティクルフィルタとレーベンバーグ・マーカート法の組み合わせを提案する。パーティクルフィルターを用いてグローバルに推定し、推定結果を初期値としてレーベンバーグ・マーカート法で推定することで、精度向上が期待できる。



図 8 推定対象画像

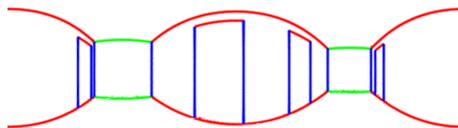


図 9 ラインマップ画像

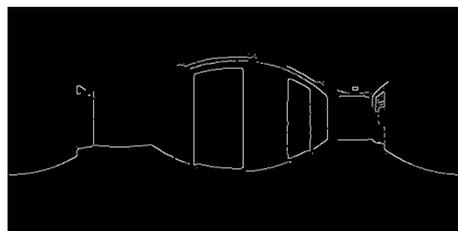


図 10 エッジ画像

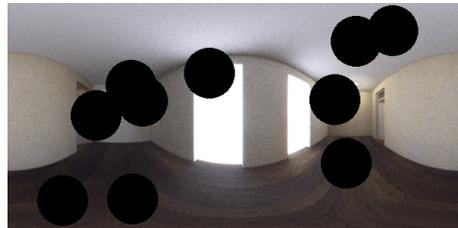


図 11 オクルージョンを追加した画像 (10 個の円)

3 実験と結果

3.1 実験条件

本研究では、シミュレーション実験を行った。実験環境は $7.9\text{m} \times 2.8\text{m} \times 2.9\text{m}$ の部屋である。10 箇所推定対象画像生成した。パーティクルフィルタはランダム性があるため、各画像について 5 回実験した。推定対象画像、ラインマップ画像、Canny 法で検出したエッジ画像の例を図 8, 9, 10 に示す。ただし、図 9 のラインマップ画像では、 x , y , z の 3 方向をそれぞれ赤、緑、青で表示している。画像の解像度を 600×300 に設定し。パーティクルの数を 2000 に設定した。エッジピクセルと対応する線分特徴のなす角度の閾値 $\hat{\theta}$ は 3 deg に設定した。推定誤差が 0.5 m 以上の場合は推

表 1 平均誤差と成功率

最適化手法	推定誤差 [m]	成功率 [%]
パーティクルフィルタ	0.199	96
組み合わせ	0.192	96

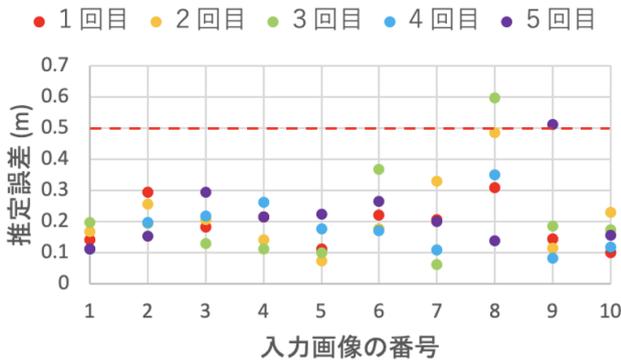


図 12 推定誤差

表 2 オクルージョンの影響検証実験の平均誤差と成功率

円の数	遮蔽面積 [%]	推定誤差 [m]	成功率 [%]
2	4	0.216	98
4	8	0.226	92
6	12	0.217	96
8	16	0.207	98
10	20	0.202	94

定失敗とする。

また、オクルージョンによる影響を調べるために、推定対象画像に障害物として画像全体の面積の2%の円をランダムな位置に追加した。円の数に2個、4個、6個、8個、10個のそれぞれについて実験し、10箇所生成した推定対象画像画像で、各5回実験した。このとき、最大で画像の20%が遮蔽される。円を追加した推定対象画像の例を図11に示す。最適化手法はパーティクルフィルタとレーベンバーグ・マーカート法の組み合わせを用いた。

3.2 実験結果

推定誤差を図12に示す。横軸は推定対象画像の番号を表し、縦軸は推定誤差を表す。赤い点線は0.5 mで、誤差がそれ以上の場合には推定失敗であることを示す。

表1にパーティクルフィルタのみを用いる場合と、パーティクルフィルタとレーベンバーグ・マーカート法の組み合わせの場合のそれぞれ10箇所×5回の50回結果の平均誤差と成功率を示す。ただし、平均誤差は成功したときの誤差のみを含めた平均である。

また、提案手法の平均計算時間は52.8 sであった。

オクルージョンによる影響の検証実験の平均誤差と成功率を表2に示す。

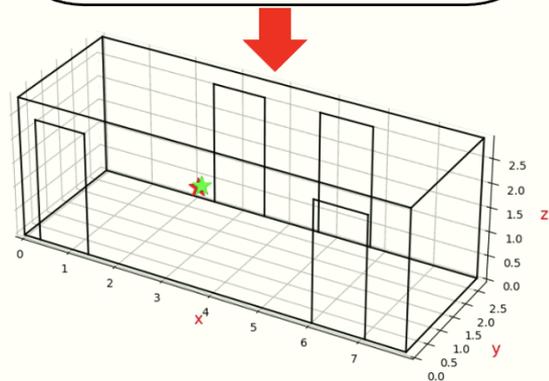
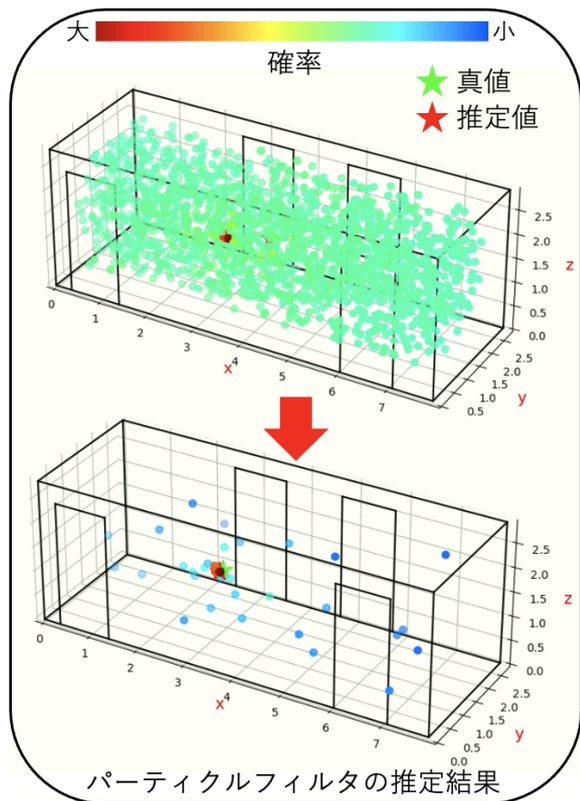
推定の様子を図13に示す。ただし、緑の星のマークは真値を表し、赤い星のマークは推定値を表している。パーティクルの色は、そのパーティクルが正しい確率を相対的に表している。黒い線分は、ラインマップにある線分である。

3.3 考察

表1から、推定誤差はパーティクルフィルタとレーベンバーグ・マーカート法の組み合わせにより、パーティクルフィルタのみの場合より精度が上がったことが明らかになった。平均誤差が0.2 m以下であった。

また、提案手法の推定成功率は高く、96%であった。最適化手法のパーティクルフィルタのみの場合と比べ、パーティクルフィルタとレーベンバーグ・マーカート法の組み合わせを用いた場合の成功率が上がらなかった理由としては、パーティクルフィルタで失敗した場合、レーベンバーグ・マーカート法の初期値は真値より遠く、ローカルミニマムに落ちやすいため、正しく推定できないことが考えられる。

推定誤差(図12)は、10箇所×5回の50回の結果の推定誤差と分散である。誤差が一番大きかった8番の推定対象画像のマッチング結果の失敗例と成功例を図14に示す。ここでは、推定された位置でのラインマップ画像を緑、外れ値ではないエッジピクセルを黄色、外れ値であるエッジピクセルを赤で示す。失敗例では、正しくない位置にもかかわらず、線分特徴とエッジピクセルが偶然一致し、評価関数値が小さかったことからその位置に推定された。成功例では、ラインマップには存在しないにもかかわらず、推定対象画像から検出されたエッジピクセルは、外れ値として検出できていることがわかる。また、エッジ画像(図10)から、線分があるのにエッジとして検出できなかった部分があることがわかる。障害物によるオクルージョンが多い環境で、1本の線分が遮られ、線分検出をすると2本

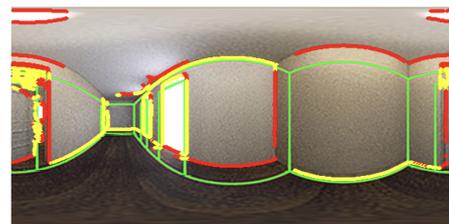


レーベンバーク・マーカート法の推定結果

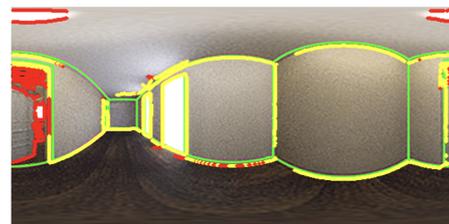
図 13 推定の様子

として検出される場合や、ラインマップにないエッジが検出された場合などにも提案手法は高い成功率で推定できる。

オクルージョンの影響を検証する実験のマッチング例を図 15 に示す。推定された位置でのラインマップ画像を緑、外れ値ではないエッジピクセルを黄色、外れ値であるエッジピクセルを赤で示す。オクルージョンの円の輪郭はエッジとして検出され、遮蔽された線分が多かったにもかかわらず、適切にマッチングがなされ、ラインマップにないエッジピクセルは外れ値であると判定できていることが確認できる。推定結果 (表 2)



推定失敗例



推定成功例

図 14 マッチング失敗例と成功例

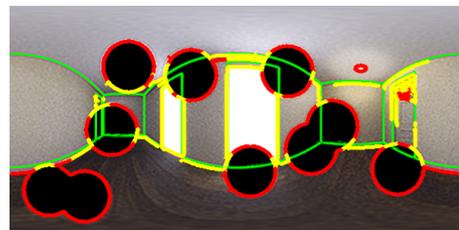


図 15 オクルージョン検証実験のマッチング例 (10 個の円)

では、オクルージョンがある場合、オクルージョンがない場合より推定誤差が少し大きいものの、平均誤差は 0.25m 以下である。オクルージョンが増えることによって推定誤差は大きくなるにつれ、推定成功率が低下し、オクルージョンが 20% の場合、成功率が 60% まで下がる。それに比べ提案手法の成功率は 90% 以上であり、オクルージョンによって推定成功率が低下しないことが示された。この実験により、提案手法のオクルージョンに対するロバスト性が高いことが確認できた。

4 結論

本研究では、環境の 3D ラインマップと全天球カメラで取得した画像の 3D-2D マッチングを利用し、オクルージョンにロバストな環境中から大域的に位置推定する手法を提案した。シミュレーション実験によって有効性を確かめた。また、オクルージョンによる影響の検証実験も行い、その結果、オクルージョンがある

場合でも、平均誤差が 0.25 m 以下であり、成功率が 90% 以上であった。オクルージョンが多い環境でもロバストに推定することが確認できた。

参考文献

- [1] S. H. Kim, C. W. Roh, S. C. Kang and M. Y. Park: “Outdoor Navigation of a Mobile Robot Using Differential GPS and Curb Detection,” Proceedings of the 2007 IEEE International Conference on Robotics and Automation (ICRA2007), pp. 3414-3419 (2007)
- [2] A. Rituerto, L. Puig and J. J. Guerrero: “Visual SLAM with an Omnidirectional Camera,” Proceedings of the 20th Conference on Pattern Recognition (ICPR2010), pp.141-148 (2010)
- [3] D. Caruso, J. Engel and D. Cremers: “Large-Scale Direct SLAM for Omnidirectional Cameras,” Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp.141-148 (2015)
- [4] J. Li, X. Wang, and S. Li: “Spherical-Model-Based SLAM on Full-View Images for Indoor Environments,” Applied Sciences, vol. 8, no. 11 pp. 2268 (2018)
- [5] 陽 東旭, 樋口 寛, S. Pathak, A. Moro, 山下 淳, 浅間 一: “色情報と 3 次元距離情報に基づく全天球カメラの位置姿勢推定,” 精密工学会誌, Vol. 86, No. 12, pp. 1014-1019 (2020)
- [6] J. Xia and J. Gong: “Precise Indoor Localization with 3D Facility Scan Data,” Computer-Aided Civil and Infrastructure Engineering, pp. 1-17 (2021)
- [7] P. E. Sarlin, A. Unagar, M. Larsson, H. Germain, C. Toft, V. Larsson, M. Pollefeys, V. Lepetit, L. Hammarstrand, F. Kahl and T. Sattler: “Back to the Feature: Learning Robust Camera Localization from Pixels to Pose,” Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR2021), pp. 3247-3257 (2021)
- [8] T. Goto, S. Pathak, Y. Ji, H. Fujii, A. Yamashita and H. Asama: “Line-Based Global Localization of a Spherical Camera in Manhattan Worlds,” Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA2018), pp. 2296-2303 (2018)
- [9] J. Courbon, Y. Mezouar, L. Eckt and P. Martinet: “A Generic Fisheye Camera Model for Robotic Applications,” Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS2007), pp.1683-1688 (2007)
- [10] J. Canny: “A computational approach to edge detection,” Proceedings of IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 8, pp. 679-714 (1986)
- [11] M. S. Arulampalam, S. Maskel, N. Gordon, and T. Clapp: “A Tutorial on Particle Filters for Online Nonlinear/non-Gaussian Bayesian Tracking”, IEEE Transactions on Signal Processing, Vol. 50, No. 2, pp. 174-188 (2002)
- [12] J. J. More: “The Levenberg-Marquardt Algorithm: Implementation and Theory,” Numerical Analysis, Springer-Verlag, Vol. 630, pp 105-116 (1977)