計測自動制御学会東北支部第343回研究集会(2023.7.7)
資料番号343-4

動的物体の影響を削減したRGB-D SLAMの開発

Development of RGB-D SLAM with Reduced Effect of Moving Objects

○菊地京陽*，釜谷博行*，原元司**，工藤憲昌*
Kyo Kikuchi*, Hiroyuki Kamaya*, Motoshi Hara**, Norimasa Kudoh*
*八戸工業高等専門学校，**松江工業高等専門学校
*National Institute of Technology, Hachinohe College,
**National Institute of Technology, Matsue College

キーワード: SLAM(Simultaneous Localization and Mapping)，動的物体(Moving Objects)
セマンティックセグメンテーション(Semantic Segmentation)，特徴点(Feature Point)
連絡先: 〒039-1192 青森県八戸市田面木字上野平16-1 八戸工業高等専門学校　産業システム工学専攻
Tel.: 0178-27-7283，E-mail: kamaya-e@hachinohe-ct.ac.jp

1. はじめに

近年、画像認識技術や自動運転技術の開発が盛んになってきており、掃除ロボットやAGVなどの移動ロボットが身近なところから産業まで普及している。このような移動ロボットには、SLAM（Simultaneous Localization and Mapping）という未知の環境で地図構築・自己位置推定をする機能が備わっており、自動化に欠かせない技術として幅広く研究されている。SLAM は静的な環境で実行することを前提として設計しているものが多く、人が動いているような動的環境では、正常に動作できないという問題がある。一例として、デスクの周りを人が歩いているような環境で、実際にSLAMを行った場合、Fig.1の赤丸で示したように人が地図に登録されてしまう。結果として、地図生成に失敗してしまう。

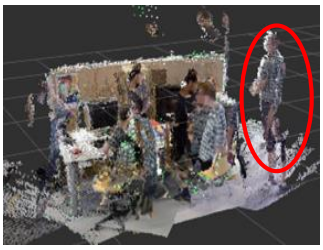

Fig.1 人が動いている環境での地図生成

そこで本研究では、人のような動的物体の影響を削減するために、RGB画像と深度情報を入力としたRGB-D SLAMに、セマンティックセグメンテーション[1]を適用して、動的物体の領域を検出、除外する手法を提案する。セマンティックセグメンテーションは画像内の物体をピクセル単位で分類する方法である。RGB-D SLAMにはリアルタイムで動作するRTAB-Map(Real-Time Appearance Based Mapping)[2]のSLAM手法を用いる。

本稿では、セマンティックセグメンテーションの予備実験と、人が動いている環境にお

ける、提案手法による自己位置推定の評価実験について述べる。

2. SLAM

開発には、RGB 画像と深度情報を利用した RGB-D SLAM を使用する。RGB-D SLAM は、カメラで取得した画像から、周囲の３次元環境を計算する既存の Visual SLAM に、深度情報を追加して精度を向上させたものである。本研究では、RGB-D SLAM の中でも、画像内の特徴点を利用した手法に着目する。このような特徴点を扱う、特徴ベースの SLAM は、画像内の一部のデータしか扱わないため情報量は減るが処理時間を短縮できるという利点がある。本稿では、リアルタイムで動作可能な RTAB-Map を用いる。

3. システム概要

Fig.2 に処理の流れを示す。基本的な処理は、RTAB-Map の SLAM 手法に準ずる。右側に示す色付けしたブロックが、今回提案する動的物体検出プロセスである。入力画像から動的物体を検出し、それらの領域を除いて、特徴点の抽出を行うことで動的物体の影響を削減する。

動的物体の検出のため、入力画像に対して、セマンティックセグメンテーションを適用する。セマンティックセグメンテーションは、画像内のピクセルごとに物体のクラスに分類することで、物体の種類と位置、領域を検出することができる画像認識技術の１つである。

動的物体検出プロセス以降は、RGB-D SLAM の基本的なフレームワークと同じで、RGB-D 画像からの特徴点抽出、特徴点マッチング、地図生成・自己位置推定で構成される。

本研究で提案する手法は、RGB-D 画像の入力から特徴点抽出プロセスまでに接続されるため、他の特徴ベースの SLAM システムにも適用できる。

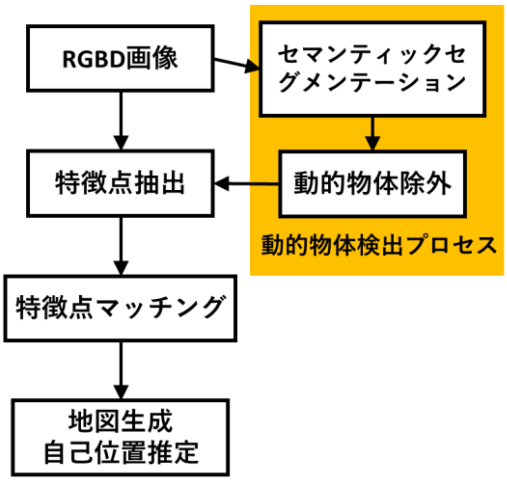

Fig.2 処理の流れ

4. セマンティックセグメンテーションの予備実験

4.1 概要

セマンティックセグメンテーションには様々な手法がある。リアルタイムで実行する SLAM では、処理時間が重要な要素になるので、セマンティックセグメンテーションのいくつかの手法について処理時間や精度を比較した。

なお、学習モデルの作成には TensorFlow 2.12.0、学習用のデータセットには oxford_iiiit_pet[3]を用いた。このデータセットには 37 品種の猫と犬の画像が合計で 7,349 枚含まれる。

4.2 処理時間・精度の比較

有名なセマンティックセグメンテーション手法として知られる、U-Net[4]、SegNet[5]、DeepLabV3[6]の３つのモデルについて処理時間や精度を比較した。入力画像のサイズは 128*128 ピクセルである。Fig.3 にセマンティックセグメンテーションで動物の領域を検出した結果を示す。縦の列は左から順に入

力画像、正解画像、出力画像である。横の行は上から(a)U-Net、(b)SegNet、(c)DeepLab-V3 である。どのモデルにおいても動物の領域を検出できていることがわかる。

Table 1 にエポック数が 50 のときの損失、精度、処理時間を示す。SegNet は処理時間が約 100ms と他のモデルより時間がかかるので適さない。処理速度が最も速いのは U-Net で、約 10ms であるが、損失と精度は DeepLabV3 が最も優れている。DeepLabV3 の処理時間は約 15ms であり、U-Net との差は約 5ms であった。

以上の結果より、本研究では損失が低く、精度が高い DeepLabV3 モデルを使用したセマンティックセグメンテーションを実装する。

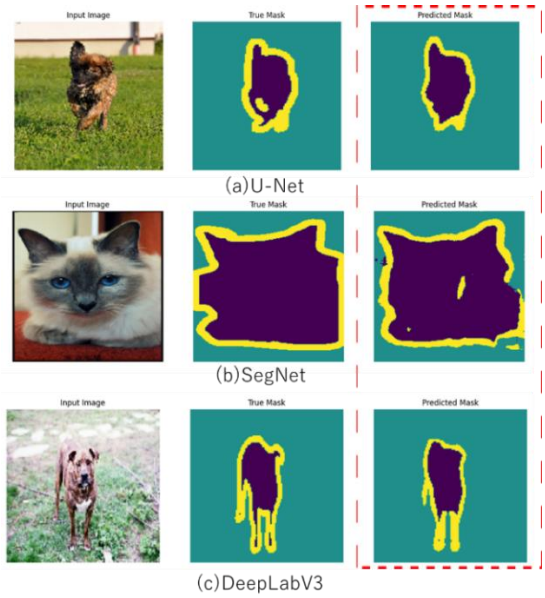

Fig.3 セマンティックセグメンテーション

Table 1 モデルの評価

	学習時		推論時
	損失	精度	処理時間[ms]
U-Net	0.189	0.923	約10
SegNet	0.0758	0.968	約100
DeepLabV3	0.0676	0.971	約15

5. セマンティックセグメンテーションの適用実験

5.1 実験内容

人が含まれる環境において、人を検出、除外するために、画像にセマンティックセグメンテーションを適用した。本研究では、ミュンヘン工科大学が提供している TUM データセット[7]を用いて、実験を行う。

5.2 使用するデータセット

TUM データセットには様々な環境でのセンサ測定値が含まれ、今回は RGB-D カメラで測定されたデータセットを使用する。使用するデータは以下の 2 つのデータセットである。2 つのデータセットの違いは、RGB-D カメラの移動の有無である。

(1)カメラは人の手によって固定され、歩行者が 2 人存在する動的環境。

(2)カメラが xyz の 3 次元上で移動し、歩行者が 2 人存在する動的環境。

5.3 実行環境

・プロセッサー: Intel®CoreTM i7-8700 CPU @3.20GHz×12

・RAM: 16GB

・グラフィック: NVIDIA GeForce GTX 1080 メモリ 16GB

・OS: Ubuntu 20.04.5 LTS 64bit

5.4 動的領域(人)の除外

セマンティックセグメンテーションを 2 つのデータセットに適用して、人間の領域を検出したマスク画像を生成した。

Fig.4 に人間の領域を検出したマスク画像を示す。左が入力画像、右が出力画像である。画像内の人間の領域を検出した画像が出力されている。

Fig.5 に Fig.4 のマスク画像を適用して、特徴点を抽出した結果を示す。画像内のカラフルな点が特徴点を可視化したものである。左がマスクを適用しないもの、右がマスクを適用したものである。マスクを適用したもの

は、人の服や顔から特徴点を抽出していないことがわかる。

Fig.4 人間の領域検出

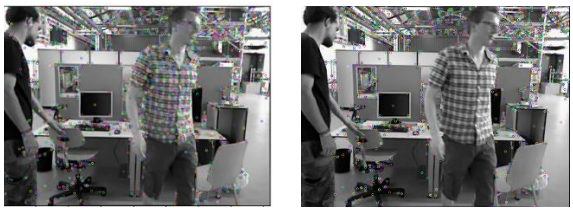

Fig.5 特徴点の比較

6. 自己位置推定

6.1 実験内容

カメラを固定した場合と動かした場合について、セマンティックセグメンテーションを適用したものを含めた 4 つのデータで SLAM を行い、自己位置推定の精度の比較を行う。自己位置推定の評価には、SLAM の評価ツールである evo[8]を使用した。また、実行環境は 5.3 項で示したものと同様である。

6.2 カメラ固定

カメラを固定した場合のデータセットについて実験結果を比較した。Table 2 にカメラの位置の真値と推定値の絶対位置誤差(Absolute Pose Error ; APE)の最大値と平均値、二乗平均平方根誤差(Root Mean Squared Error ; RMSE)を示す。RMSE の計算式を式(1)に示す。

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2} \qquad (1)$$

ここで、y_i が実際のカメラ座標であり、$\hat{y}_i$が SLAM を用いて推定した座標である。そして N 個のデータに対して平均化される。

Table 2 の結果から RMSE 値で比較すると、改善前は 0.36[m]であったが、改善後は 0.07[m]となり、改善後は 0.29[m]小さくなった。このことから、自己位置推定の精度が向上していることがわかった。

次に、Fig.6、Fig.7 に改善前、改善後それぞれの SLAM を開始してからの時間 t と APE の関係を示す。Fig.6 の改善前は時間が経つにつれて APE が増加しているが、Fig.7 の改善後は APE が増加せず、非常に小さい値を保っていることがわかる。

Table 2　自己位置推定の誤差(カメラ固定)

	Maximum error[m]	Mean[m]	RMSE[m]
改善前	0.55	0.24	0.36
改善後	0.09	0.07	0.07

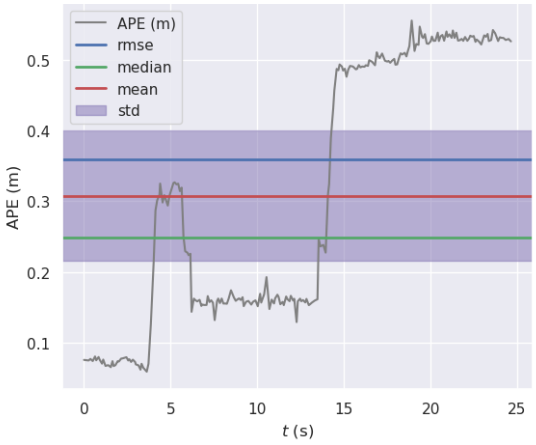

Fig.6 改善前の APE の変化（カメラ固定）

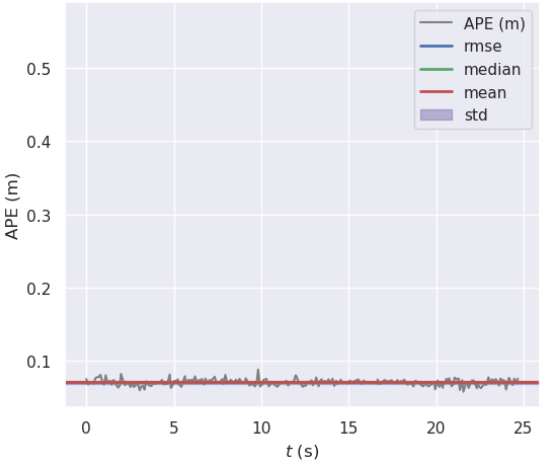

Fig.7 改善後の APE の変化（カメラ固定）

6.3 カメラ移動

カメラ固定の場合と同様に、改善前と改善後について実験を行なった。Table 3 にカメラが移動した軌跡の真値と推定値の APE の最大値と平均値、RMSE を示す。RMSE 値で比較すると、改善前は 0.14[m]であったが、改善後は 0.09[m]となり、改善後は 0.05[m]小さくなった。このことから、自己位置推定の精度が向上していることがわかった。

次に、Fig.8、Fig.9 に改善前、改善後それぞれの SLAM を開始してからの時間 t と APE の関係を示す。Fig.8 の改善前では 17 秒付近で APE が急激に増加しているが、Fig.9 の改善後ではそのような増加はみられないことがわかる。

Table 3　自己位置推定の誤差(カメラ移動)

	Maximum error[m]	Mean[m]	RMSE[m]
改善前	0.20	0.13	0.14
改善後	0.15	0.08	0.09

Fig.8 改善前の APE の変化（カメラ移動）

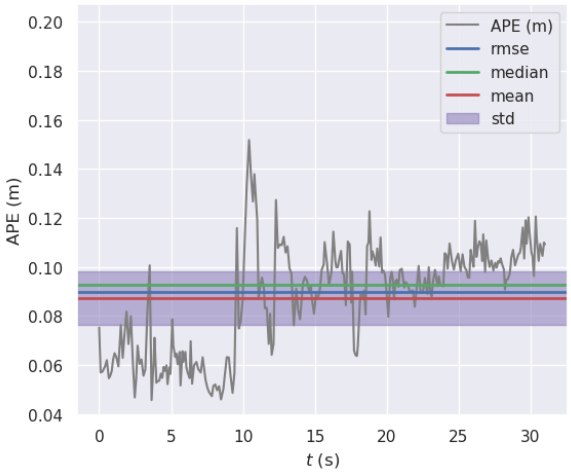

Fig.9 改善後の APE の変化（カメラ移動）

7. おわりに

本研究では、セマンティックセグメンテーションを用いた動的物体の影響を削減する SLAM を提案した。セマンティックセグメンテーションの予備実験において、3 つの手法について評価を行い、DeepLabV3 の性能が高いことを確認した。つぎに、人が動いている環境において、提案手法を自己位置推定に適用した結果、推定精度の向上を確認できた。

今後は、地図生成についても、どの程度精度向上が見込めるかなどについて詳しく検討していきたい。

8. 謝辞

実験に協力頂いたフランス Université du Littoral Côte d'Opale の Sailly Kevin 氏に感謝する。

参考文献

[1] Anil Chandra, Naidu Matcha, A 2021 guide to Semantic Segmentation, https://nanonets.com/blog/semantic-image-segmentation-2020/, 2021 (2023 年 7 月 3 日 閲覧)

[2] M. Labbé and F. Michaud, RTAB-Map as an Open-Source Lidar and Visual SLAM Library for Large-Scale and Long-Term

Online Operation, in Journal of Field Robotics, vol. 36, no. 2, pp. 416–446, 2019.

[3] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman and C. V. Jawahar, The Oxford-IIIT Pet Dataset, https://www.robots.ox.ac.uk/~vgg/data/pets/ (2023 年 6 月 27 日 閲覧)

[4] Olaf Ronneberger, Philipp Fischer, and Thomas Brox, U-Net: Convolutional Networks for Biomedical Image Segmentation, arXiv:1505.04597, 2015.

[5] Badrinarayanan V, Kendall A, Cipolla R. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 12, pp. 2481-2495, 2017.

[6] Liang-Chieh Chen, George Papandreou, Florian Schroff, Hartwig Adam, Rethinking Atrous Convolution for Semantic Image Segmentation, arXiv:1706.05587, 2017

[7] TUM dataset, https://vision.in.tum.de/data/datasets (2023 年 6 月 27 日 閲覧)

[8] Grupp, Michael, evo: Python package for the evaluation of odometry and SLAM, https://github.com/MichaelGrupp/evo, 2017. (2023 年 6 月 27 日 閲覧)