

ARマーカーを追従する自律移動台車の開発

佐野 峻輔

Development of an AR Marker-Based Autonomous Following Cart

Shunsuke SANO

抄 録

近年、労働力不足や省力化ニーズの高まりを背景に、人の移動に追従して物品を搬送する移動ロボットの需要が高まっている。特に物流・製造・サービス分野では、簡便に導入可能な追従機構の開発が求められている。本研究では、ARマーカーとUSBカメラを用いた追従機構を構築し、自律移動台車に搭載してその基本性能を評価した。追従対象に貼付したマーカーの相対位置を取得し、PID制御によって自律移動台車を対象に追従させる実験を行った結果、安定したマーカー追従動作、および認識・制御系の有効性が確認された。

キーワード：自律移動台車、ARマーカー、移動ロボット

1. はじめに

近年、少子高齢化の進行により、多くの分野で深刻な労働力不足が問題となっている。それに伴い、限られた人員で効率的に作業を行うための省力化や自動化のニーズが急速に高まっている。こうした背景のもと、物流、製造、サービス業などを中心に、現場作業を支援する移動ロボットの導入が広がっている。

なかでも、作業者の移動に追従して物品を搬送する「追従型移動ロボット」は、作業者にとって特別な操作や事前の操作習得が必要なく、普段どおりの作業者の動線に合わせて移動する点が大きな特長である。日常業務の流れを妨げることなく現場へ導入することが可能であり、既存の作業環境に適応しやすい種類のロボットである。

この移動ロボットの走行を制御するためには、追従対象との相対位置を正確に把握する必要がある。屋内において追従対象と移動ロボットの相対位置を計測する手段として、Wi-Fi、BLEビーコン、IMES (Indoor MEssaging System)、モーションキャプチャ、UWBタグなど、さまざまな計測技術が提案されている。これらの方式は作業環境に送信機やセンサーを設置する必要があり、初期導入や維持管理にかかるコストが大きいという課題がある。

一方で、作業環境に特別な機器を設置する必要がない相対位置計測技術として、LiDAR¹⁾ やステレオカメラ²⁾、画像認識ベース³⁾ の手法などがある。これらは拡張性に優れるが、

画像などからの情報量が多いため、一般的に高性能な演算装置を必要とする場合が多く、リアルタイム処理が困難となる場合がある。特に、機械学習にもとづく認識手法を用いる場合には、事前に学習用の大量のデータセットを準備する必要があり、その構築・調整にも多大な労力とコストを要する。

しかし、簡易なマーカーを単眼カメラで撮像し認識することはそれほど計算量を必要としないため、これらを用いて追従対象の位置を取得し、その情報にもとづいて自律的に対象を追従する移動ロボットを構築できれば、専用センサーや高価な通信機器を用いずに安価に実用的な追従動作が実現可能となる。

そこで本研究では簡易なマーカーとして、導入コストが低く利用実績が豊富なARマーカーに着目した。ARマーカーは一般的なプリンターで印刷でき、単眼カメラで撮像、認識することも容易である。この方式は低コストで導入できるため、小規模な現場での運用に適した、簡便かつ汎用性の高い追従方式として期待できる。

本報告では、ARマーカーと単眼カメラによる追従システムの開発に関して、その実装と評価結果について述べる。

2. システム構成

2.1 概要

本研究で構築した自律移動台車は、USBカメラ、制御用PC、移動台車で構成される(図1)。移動台車前方に取り付

けたカメラで追従対象物のマーカーを撮影し、制御用PCはその画像からマーカーの位置を算出する。制御用PCがマーカーの位置情報をもとに移動台車の移動方向と速度を計算し移動台車に送信し、移動台車が制御指令にもとづいてモーターを制御することで追従対象の動きに自動で追従する。

移動台車には市販の二輪駆動ロボットを用い、自動追従プログラムを実行する制御用PCとして汎用ノートPCを使用した。さらに、異なる計算機を搭載する移動台車でも同様に動作可能とするために、本研究ではコンテナ型仮想化技術であるDocker上にシステムを構築した。

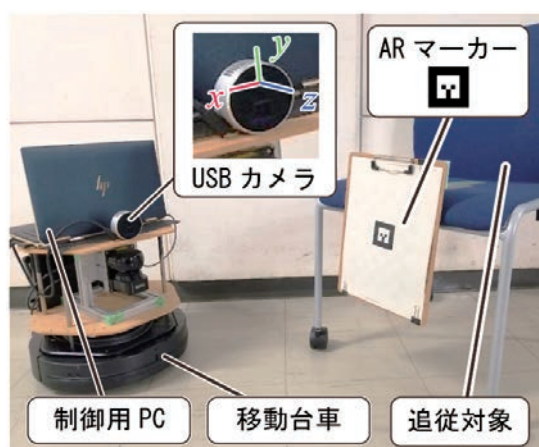


図1 ロボットハードウェア構成と座標

2.2 ハードウェア構成

本システムでは、小型の移動台車として市販の二輪駆動ロボット（Kobuki・Yujin Robot社製）を用いた。このロボットは左右の車輪を独立に駆動する方式を採用しており、比較的単純な構成ながら安定した走行を実現している。サイズはおおよそ幅35cm、奥行33cm、高さ15cm、重さ約5.6kgであり、内蔵バッテリーで動作する。

台車前方にはUSBカメラ（RealSense L515・Intel Corp. 製）を固定し、画角内のマーカーを検出する。解像度はフルHD相当で、水平画角約70度と広めの画角をもつため通常の移動速度であれば追従対象を十分に画角内に収められる。

移動台車の制御およびマーカーの認識処理は、ノートPC（Intel社製Core i7-1065G7搭載機）で実行した。台車およびカメラとの接続はすべてUSBとした。

2.3 ソフトウェアプラットフォーム

自動追従プログラムを実行するソフトウェア環境を、他のロボットやPCでも容易に再利用できるよう、再現性と可搬性に優れる手法で構築した。具体的には、プログラムはロボット開発の共通基盤として広く使われている「ROS(Robot Operating System)」に準拠して作成し、その実行環境をコンテナ型仮想化技術のDocker上に構築した。

ROSとは、センサー情報の取得、モーター制御、位置推定、

画像処理といったロボットシステムを構成する各種プログラム（ノード）を部品化し、それらをメッセージ通信によって連携させるためのフレームワークである。オープンソースとして提供されており、世界中の研究者や技術者によって開発が進められている。多数のハードウェアや外部ライブラリとの互換性があるため、実験用プラットフォームとしても汎用性が高い。ただし、ROSは多くの依存ライブラリや設定ファイルを必要とするため開発環境の構築が煩雑であり、バージョン管理や機材変更のたびに再設定が必要になるという課題がある。

そこでDockerと呼ばれる、コンテナ型仮想化技術を活用し、ROS環境をあらかじめ構築した仮想的な作業空間としてパッケージ化した。このコンテナを用いることで、ハードウェア構成が異なる別のPCでも、同一の動作環境を短時間で再現できる。これにより、実際の現場環境への導入が円滑になることが期待される。

本研究においては、LinuxベースのOS（Ubuntu 18.04）上でDockerコンテナを起動し、カメラ画像の取得、ARマーカーの検出、ロボットへの制御指令の生成と送信などを、コンテナ内部で自動的に実行する構成とした。これにより、コンテナの起動と数行のコマンド入力だけで、複雑な環境構築を行うことなくシステムを実行可能とした。

3. 認識および制御の設計

3.1 ARマーカーライブラリ

ARマーカーライブラリにはALVAR、AprilTag⁴⁾ や ChromaTag⁵⁾、LentiMark⁶⁾ など複数の方式があるが、本研究ではROS環境において安定して動作するラッパーライブラリ（ar_track_alvar）が利用可能であり、導入・運用が容易な、ALVARを採用した。

3.2 ARマーカーの認識処理

今回使用した52mm四方のマーカーを図1に示す。これを今回簡易的な追従対象としたキャスター付きの事務用椅子に貼り付け、以下の方法により認識処理を行う。

マーカーの認識処理に使用したARマーカーライブラリ（ALVAR）は、映像入力に加えてカメラの内部パラメーター（レンズの歪みや焦点距離など）をもとに、マーカーのIDと三次元空間における位置（x, y, z）および姿勢（3軸まわりの回転角度）を出力する。

出力されるマーカーの位置情報は、図1のようにカメラ中心を原点とし、前方を+Z軸、右方向を+X軸、上方向を+Y軸とする左手系の座標系で出力される。なお、位置・姿勢推定の精度を高めるためには、カメラ画像を補正するためのキャリブレーション情報を事前に取得する必要がある。具体的には、所定の寸法で作成した市松模様のプレートを様々な

角度・距離から撮影することで、ライブラリがレンズの歪みや焦点距離などのキャリブレーション情報を自動で算出する。

3.3 追従制御アルゴリズム

ARマーカの検出により取得した追従対象の位置情報は、自律移動台車の進行方向と速度の制御に利用した。処理の流れを図2に示す。

カメラから見た対象の位置を「左右方向のずれ(x軸方向)」と「前後方向の距離(z軸方向)」の2つの成分に分けて扱い、それぞれにもとづいてモーターの制御信号を生成した。左右方向のずれに応じて台車を旋回させ、前後方向の距離に応じて速度を調整することで対象に追従する動作を実現した。目標とする追従距離は50cmとした。

この制御には、目標値と現在値の誤差に応じて出力を調整するPID制御を用いた。高い汎用性があり実装が容易であるため広く普及している制御手法であり、今回の用途に適している。PID制御は、現の誤差だけでなく、過去の誤差蓄積や変化の速さをも考慮することで、過度な変動を抑えながら現在値を目標値に収束するよう調整する制御手法である。本自律移動台車では、滑らかな応答が得られるゲイン（調整係数）を実験により求めて設定した。

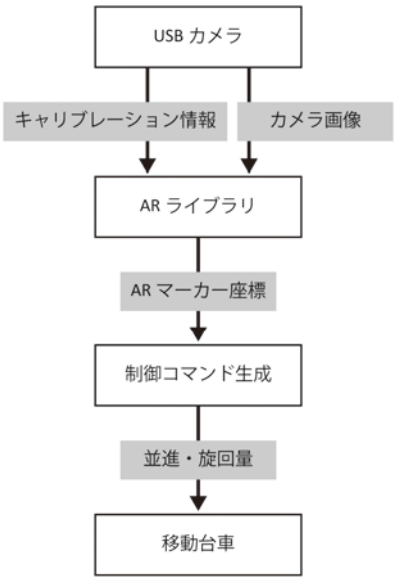


図2 移動台車の制御における処理の流れ

4. 性能評価および追従実験

4.1 実験の概要

はじめに、計測精度に関する基礎的な実験として、USBカメラをロボットに搭載せず、単体で静的な位置推定の精度を評価した。次に、USBカメラを自律移動台車に搭載し、追従対象に貼付したARマーカに対する追従動作確認実験に

より、本手法の有効性を確認した。

4.2 カメラ単体による距離計測精度の評価

4.2.1 実験方法

図3のとおりUSBカメラとARマーカをそれぞれ三脚に固定し、互いに向かい合う姿勢で距離精度を測定した。実距離と出力値との平均誤差を算出するため、カメラとマーカの距離をメジャーで測定し、1.0-2.5mの範囲において距離(z軸方向)の出力を取得した。

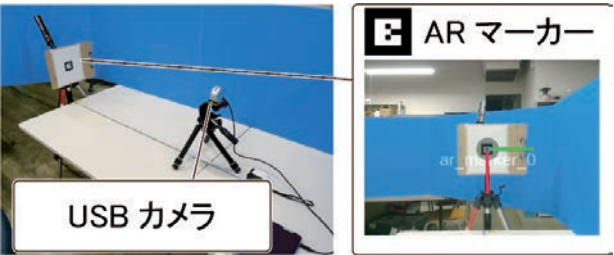


図3 ARマーカとカメラ

4.2.2 結果

計測結果を表1に示す。マーカの位置の計測値はメジャーで計測した真値とおおむね良好に一致し、誤差は全体として3%程度に収まった。

表1 ARマーカの距離測定精度

| 真値 [m] | 計測値 [m] | 誤差 [%] |
|--------|---------|--------|
| 1.00 | 1.018 | 1.80 |
| 2.00 | 2.063 | 3.15 |
| 2.50 | 2.494 | 0.24 |

4.3 ロボット搭載時における追従動作の確認

4.3.1 実験方法

追従動作を確認するため、前節で示した構成に従ってUSBカメラを移動台車の前面に固定し、マーカ検出・追従制御を実行可能な状態とした。効率的にシステム開発を進めるため、前後左右の移動、および旋回が可能なキャスター付きの事務用椅子にマーカを貼付した。

追従対象の最大の移動速度は台車の移動速度以内とした。台車の最大直進速度は0.7m/s、旋回角速度は180deg/sである。

4.3.2 結果

実験の様子を図4に示す。自律移動台車はARマーカが貼付された追従対象の移動に応じて、前進・旋回・停止といった動作を行って滑らかに追従する動作を示した。特に、追従対象が画角から外れる方向に移動した際にもロボットの向きが適切に変化し、ARマーカがカメラ画角に保たれる

状態が維持されることを確認した。



図4 ARマーカーに向かって追従する自律移動台車

5. 考察

本研究では、ARマーカーを用いた追従対象の位置計測システムと、それにもとづく自律移動台車の追従制御システムを開発した。実験により、簡易なハードウェア構成で一定の認識精度と追従性能が得られることを確認した。特に、ARマーカーを用いた追従制御の手軽さは、現場への簡易導入を目的とした自律移動台車の構築において有用である。

一方で、本システムにはいくつかの制約がある。特に、自律移動台車の走行経路が追従対象と完全には一致しないため、狭路など特定の状況では台車が通過できない場合がある。この点を改善するには、走行環境の地図にもとづく高度な処理が必要となる。

また、マーカー検出性能にも制約がある。現行のアルゴリズムでは、追従対象がその場で旋回するとマーカーを見失う場合がある。本研究では、マーカーを見失った場合には移動を停止し、再検出後に自動で移動を再開する設計とした。マーカー検出が途絶しにくい頑強なシステム構成の実現は今後の課題である。

6. まとめ

本研究では、USBカメラとARマーカーを用いた簡易な追従型自律移動台車の構築を試み、誤差約3%の測定精度と、基本的な追従動作の実現を確認した。汎用的なソフトウェアフレームワークを採用したことで、市販されている別のROS対応移動台車であっても容易に移植できるシステム構成とした。これにより、移動台車に追従機能を低コストで導入する際の有力な選択肢となることを示した。

一方で、移動台車の移動性能やマーカーの認識性能の制約が課題としてあげられた。実運用においては、導入する場面を精査のうえ、より高い自由度を有する台車を採用することや複数のマーカーを多方向に向けて配置するなどの対策が必要である。

参考文献

- 1) W. Chung, H. Kim, et al.: *IEEE Transactions on Industrial Electronics*, vol. 59, no. 8, pp. 3156–3166, (2012)
- 2) Zhichao Chen and S. T. Birchfield,: 2007 *IEEE/RSJ International Conference on Intelligent Robots and Systems*, San Diego, CA, USA, 2007, pp. 815–820 (2007)
- 3) Redhwan Algabri et al.: *Sensors (Basel, Switzerland)* (2020)
- 4) Olson, Edwin: *2011 IEEE international conference on robotics and automation*. IEEE, (2011)
- 5) DeGol, Joseph, Timothy Bretl, and Derek Hoiem: *Proceedings of the IEEE international conference on computer vision*. 2017.
- 6) Hideyuki Tanaka: 2020 *IEEE International Conference on Robotics and Automation*. IEEE, (2020)