

視覚と行動の end-to-end 学習により経路追従行動を オンラインで模倣する手法の提案 実環境における経路選択機能の検証と学習時間の短縮化の検討

A proposal for an online imitation method of path-tracking behavior by end-to-end
learning of vision and action

-Verification of route selection function in a real environment and study of reducing learning time-

学 藤原 証 (千葉工大) 学 春山 健太 (千葉工大) 馬場 琉生 (千葉工大)
正 上田 隆一 (千葉工大) 正 林原 靖男 (千葉工大)

Masaki FUJIWARA, Chiba Institute of Technology, s19c1101ga@s.chibakoudai.jp

Kenta HARUYAMA, Ryusei BABA,

Ryuichi UEDA and Yasuo HAYASHIBARA, Chiba Institute of Technology

We have proposed an online imitation method of path-following behavior by end-to-end learning of visual actions. However, the proposed method aims at orbiting a fixed path and cannot dynamically change the path in order to move the robot to the destination. Therefore, in this study, we add a route selection function so that the robot can move to any destination. We attempt to solve the problem of long learning time, which was identified through preliminary experiments in a real environment, by trying two approaches. In addition, we will verify the effectiveness of the route selection function through experiments in a real environment.

Key Words: Autonomous mobile robot, Navigation, End-to-end learning

1 緒言

我々は、入力データから出力を直接生成する end-to-end 学習により、経路追従行動をロボットの視覚に当たるカメラ画像に基づいてオンラインで模倣する手法を提案し、有効性を実験により検証してきた [1][2] (以下、「従来手法」と称する)。加えて、この岡田らの従来手法に経路を選択する機能を追加を提案し、シミュレータ上での実験により、有効性を検証してきた [6]。従来手法は、測域センサやオドメトリなどを入力とする地図を用いたルールベース制御器による走行を模倣学習し、似た行動を画像を用いて行う手法である。なお、地図を用いたルールベース制御器は、ROS Navigation_stack[8] へ目標位置 (waypoint) の指示を行う waypoint_nav[9] を組み合わせたものである。この手法は、多くの研究 [3][4][7] が人間のドライバーが操作するステアリング角度を用いて模倣学習を行っていたのに対して、オンラインで自動的にデータセットを収集して模倣学習を行う点で異なっているといえる。

この手法は、学習を行った一定の経路を走行できるが、目的地に対して経路を動的に選択して走行することはできない。それをするためには、ネットワークの入力に新たな要素の追加が必要となる。新たな要素として、カメラ画像とステアリング角度に条件を加えて学習を行う条件付き模倣学習の研究が現在までにいくつか行われている。Felipe ら [5] や Hawke ら [7] は、前方カメラ画像に加えて方向を指示するコマンドを入力とするネットワークを用いて模倣学習を行った。そのため、前報 [6] ではネットワークへ条件を追加し、分岐路において任意の経路を選択する機能を追加を提案した。また、実験により有効性を検証した。しかし、有効性の検証はシミュレータ上のみでしか確認しておらず、実環境においては未確認であった。

そこで本稿では、前報 [6] で追加した経路選択機能の有効性を実環境において検証する。さらに、実験で問題となった学習時間の長さについて、2つのアプローチにより学習時間を短縮することを目的とする。

条件付き模倣学習により、学習済みモデルによる走行時において経路を制御できるようになると、自律走行を行う手段が地図を用いたルールベース制御器と模倣学習など、複数持つことができる。たとえば、自律走行の手段を切り替えることでロボットの向上が期待できる。

2 従来手法

岡田らの従来手法に関して紹介する。図1に経路追従行動を視覚に基づいてオンラインで模倣するシステムを示す。手法は機械学習により、学習器の訓練を行う「学習フェーズ」と訓練した結果を検証する「テストフェーズ」に分かれる。

2.1 学習フェーズ

学習フェーズは、模倣学習によって学習器の訓練を行うフェーズである。測域センサとオドメトリを入力とする地図を用いたルールベース制御器で自律走行する。この経路追従行動を、カメラ画像を用いた end-to-end で模倣学習する。前報 [2] で述べたように、岡田らの手法ではデータセットの収集方法にいくつかの種類がある。本稿では、その中で最も経路追従の成功率が高い手法を用いてロボットに模倣学習をさせる。

2.2 テストフェーズ

テストフェーズは、訓練後の学習結果を評価するフェーズである。学習器にカメラ画像を入力し、出力されるヨー方向の角速度を用いて自律走行する。

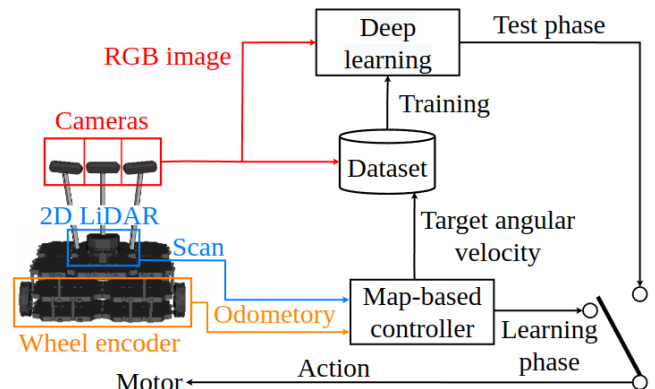


Fig.1 Imitation learning system of learning phase

3 経路選択機能の追加

経路選択機能の追加を目的として、データセットと学習器の入力へ、目標とする進行方向（以下、「目標方向」と称する）を追加する。なお、追加した要素以外は従来手法と同様である。図2に提案手法のシステムを示す。学習フェーズでは、地図を用いたルールベース制御器で自律走行しながら、角速度、カメラ画像に新たに目標方向をデータセットに加える。テストフェーズでは、学習器にカメラ画像に加えて目標方向を外部から入力する。本稿では議論しないが、最終的に目的地までカメラ画像のみで自律走行させることを目標としているため、目標方向を自動的に作成する仕組みが必要となる。これに関しては、カメラ画像からトポロジカルマップのどのノード上に到達したかを判定し、目標方向を自動的に切り替えるパッケージを作成している [10]。

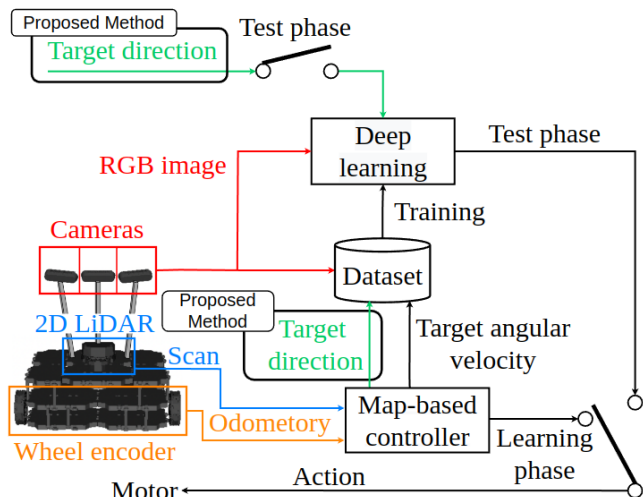


Fig.2 Imitation learning system of test phase

4 ネットワークとデータ形式

図3に経路選択機能を追加したネットワーク、表1にハイパーパラメータを示す。先行研究に倣って、前報で用いたネットワークの出力部へ2つの層を追加した。この層に目標方向として、表2に表されるデータを入力する。なお、前報 [6] では「Continue」、「Go straight」、「Turn left」、「Turn right」からなる4つのコマンドを用いていたが、「Continue」と「Go straight」のコマンドに対応する行動がほぼ同一であったため、本稿ではコマンドを1つ減らし、3コマンドとしている。

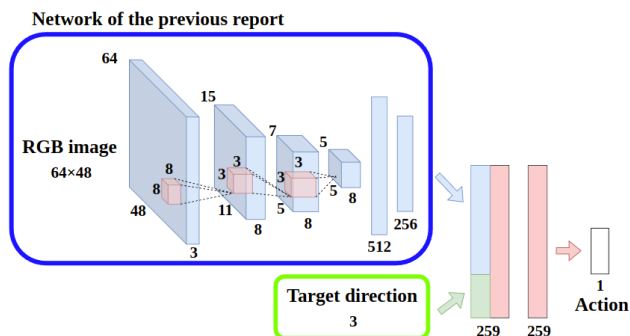


Fig.3 Structure of network

Table 2 Target direction and data for imitation learning

Target direction	Data
Go straight	[100, 0, 0]
Turn left	[0, 100, 0]
Turn right	[0, 0, 100]

Table 1 Parameters of network

Input data	Image (64 × 48 pixels, RGB channels), Target direction
Optimizer	Adam(alpha = 0.001, beta1 = 0.9, beta2 = 0.999, eps = 1e ⁻²)
Loss function	Softmax-cross-entropy
Output data	Angular velocity

5 提案手法の検証実験

前報 [6] で行った実験2を実環境に移して行う。

5.1 実験装置

ロボットは、図4に示すような3つのカメラを搭載したロボットを用いる。また、実験は千葉工業大学津田沼キャンパス2号館3階で行った。

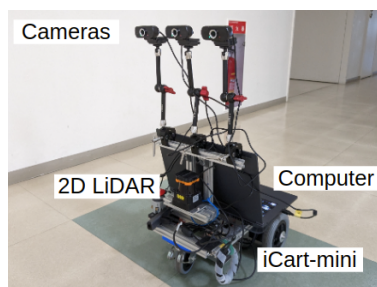


Fig.4 Experimental setup

5.2 実験方法

図5に示すようなA,Bの三叉路で全ての侵入方向と抜け出す経路を含めるように、前報 [6] の実験2と同様の経路を繰り返し走行させる。学習を60000step実行後、テストフェーズに移行する。テストフェーズでは、学習フェーズと同様の経路を、目標方向による制御によって正しく走行できるか評価する。しかし、実験を行ったのは1回のみである。実験が1回のみなのは、学習時間の長さが問題になったためである。具体的には、実験におよそ7時間を要し、バッテリーなどの関係で1日に1,2時間が限界であったため、1回の実験に1週間かけて行う必要があった。

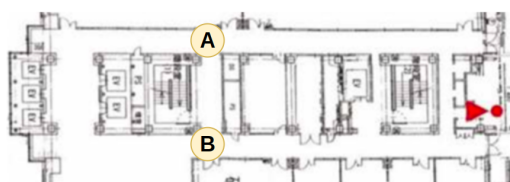


Fig.5 Experimental environment

5.3 実験結果

実験結果を図6に示す。この図は、それぞれの侵入方向に対して正しい経路を選択し、走行できた回数を表している。目標方向に従って9/12回 (75%), 正しい経路を選択する様子が見られた。

6 学習時間の短縮化の提案

6.1 問題

前報 [6] で行った実験により、シミュレータ上では提案手法が有効だと確認している。そのため、5章で述べたように、実環境における提案手法の有効性を検証することを試みた。しかし、学習時間の長さが問題となり、評価を複数回行うことが困難であった。そのため、学習時間の短縮化に関して2つのアプローチを提案し、学習時間の削減の効果を実験により確認する。

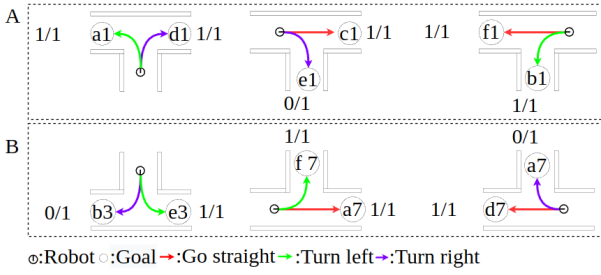


Fig.6 Number of times a correct path was selected

6.2 学習ステップ数の削減による影響

前報 [6] のシミュレータ上で行った実験 2 と同様の条件で、60000step から 10000step に学習ステップ数を変更し、実験を行った。実験結果を図 7 に示す。表 3 に侵入方向ごとの成功回数を合計した結果を示す。表に示すように、87/120 回 正しい経路を選択する様子が見られた。

60000step の実験と成功率を比較すると約 22%ほどの差がある。これまで述べた提案手法では、学習ステップ数を減らすことにより、成功率が大幅に低下することが判明した。そのため、後述する 2 つのアプローチを試みることで成功率の改善を図る。

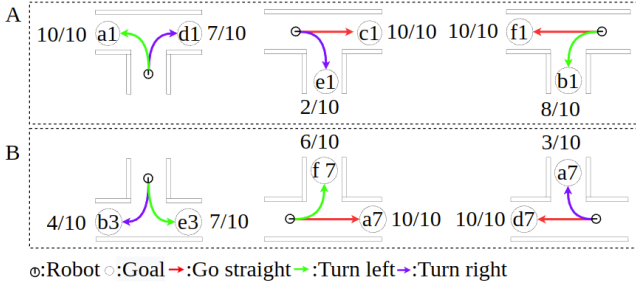


Fig.7 Number of times a correct path was selected

Table 3 Experimental results

Experiment	Step	Total results
Exp.2 from [6]	60000	113/120(94.2%)
	10000	87/120(72.5%)

6.3 アプローチ 1：データセットに加えるデータの割合の変更

6.2 章の実験における 10000step あたりのコマンドのデータ数を図 8 に示す。図 8(a) より、直進コマンドのデータ数が圧倒的に多く、不均衡データであることが分かる。不均衡データを調査した Haibo ら [12] によると、ほとんどの標準的な学習アルゴリズムがデータの不均衡により、パフォーマンスが大幅に低下すると述べている。そのため、オーバーサンプリングを行い、データの偏りを改善する。図 8(b) に示すように、左折と右折コマンドのデータを 7 倍に複製し、データセットへ加える。なお、データを何倍にするのが望ましいのか本稿では議論しない。実験により、データセットに加えるデータ数の割合の変更が有効か検証する。

6.3.1 実験装置

前報 [6] で行った実験 2 のシミュレータ環境とロボットで実験を行った。

6.3.2 実験方法

前報 [6] の実験 2 と同様の経路を繰り返し走行させる。学習を 10000step 実行後、テストフェーズに移行し、評価する。この一連の流れを 10 回行う。

6.3.3 実験結果

実験結果を図 9 に示す。表 4 に侵入方向ごとの成功回数を合計した結果を示す。表に示すように、99/120 回 正しい経路を選択する様子が見られた。6.2 章の学習ステップ数を減らした結果と比べると成功率は高いが、60000step の実験と成功率を比較すると約 12%ほど差があり、アプローチ 1 を試みた結果では成功率が高いとは言えないため、次にアプローチ 2 を試みる。

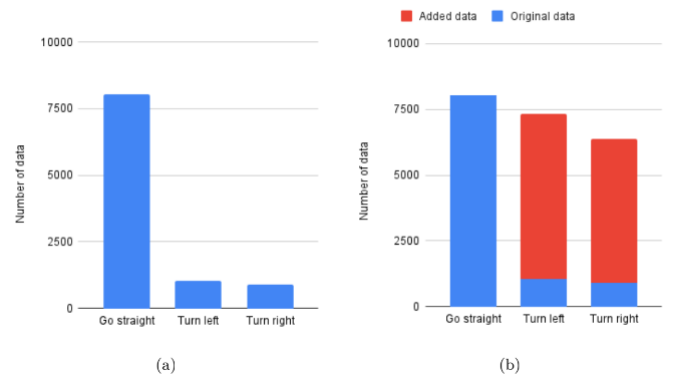


Fig.8 Number of data per command per 10000steps in conventional experiments

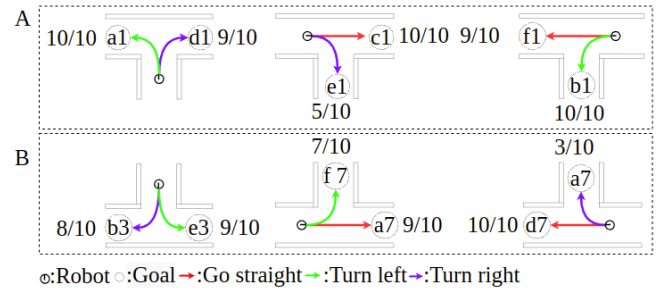


Fig.9 Number of times a correct path was selected

Table 4 Experimental results

Experiment	Step	Total results
Exp.2 from [6]	60000	113/120(94.2%)
	10000	87/120(72.5%)
Data 7 times	10000	99/120(82.5%)

6.4 アプローチ 2：学習フェーズにおける積極的な蛇行

清岡ら [11] により、目標経路上に加えて離れた状態を学習することが、テストフェーズでの走行において目標経路に復帰できるかに大きな影響を与えるため、重要だとされている。そのため、より積極的に蛇行を行い、目標経路から離れた状態を増加させることを検討する。

訓練中の学習器へ画像を入力し、図 10 のように出力される角速度を 1.5 倍にする。その結果、蛇行する頻度が高くなり、目標経路から離れた状態をより多く学習できる可能性がある。なお、得られた角速度を何倍にするのが望ましいのか本論文では議論しない。実験により、学習フェーズにおける積極的な蛇行が有効か検証する。

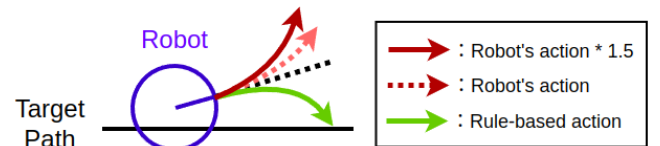


Fig.10 Robot behavior

6.4.1 実験装置

前報 [6] で行った実験 2 のシミュレータ環境とロボットで実験を行った。

6.4.2 実験方法

6.3 章と同様の条件で実験を行った。

6.4.3 実験結果

実験結果を図 11 に示す。表 5 に侵入方向ごとの成功回数を合計した結果を示す。表に示すように、109/120 回 正しい経路を選択する様子が見られた。

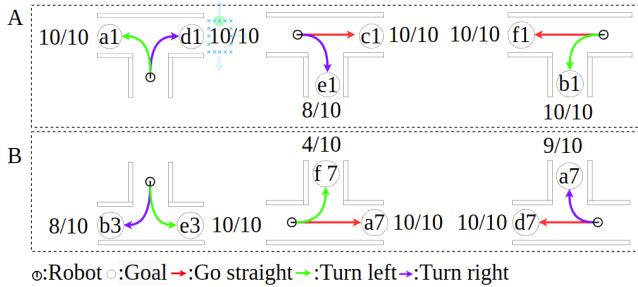


Fig.11 Number of times a correct path was selected

Table 5 Experimental results

Experiment	Step	Total results
Exp.2 from [6]	60000	113/120(94.2%)
	10000	87/120(72.5%)
Data 7 times	10000	99/120(82.5%)
Aggressive meandering	10000	109/120(90.8%)
	20000	114/120(95%)

6.3 章のデータの割合の変更した結果と比べると成功率が改善していることが分かる。以下に図 11 における成功率が低い場所と要因を示す。

- f7
前報 [6] で行った実験 2 と同様に、予め設定した経路の最後であるため、データセットに f7 のデータが加えられてから十分に学習がされていないおそれがある。
- a7, b3, e1
これらは、全て右折する場所である。図 8 より、他のコマンドのデータと比較して右折コマンドのデータが少ないことが分かる。このことから、右折コマンドのデータを十分に学習できていないおそれがある。

この 2 つの点から、学習ステップ数を増やすと成功率が改善する可能性がある。そのため、10000step から 20000step に学習ステップ数を増やして実験した。実験結果は、表 5 に示すように成功回数が 114/120 回 (95%) となり、成功率が 60000step の実験と同程度になった。このことから、学習時間を短縮するための 2 つのアプローチにより、学習のステップ数が約 67%削減できたといえる。

6.5 実環境の実験

5 章で行った実験に 2 つのアプローチを追加して実験を行った。実験装置は 5 章と同様である。

6.5.1 実験方法

シミュレータ上で、まずは学習を 10000step 行い、その学習器を実環境でファインチューニングする。学習フェーズでは、前報 [6] の実験 2 と同様の経路を繰り返し走行させる。学習を 10000step 実行後、テストフェーズに移行し、評価する。この一連の流れを 10 回行う。

6.6 実験結果

実験結果を図 12 に示す。目標方向に従って 78/120 回 (65%) 正しい経路を選択する様子が見られた。前章までに行ってきたシミュレータ上での実験では、成功率が 90%を超えていたが、実環境での成功率は 65%と低かった。なお、成功率が低い原因の解明には未だ至っていない。この問題の検討は、今後の課題となっている。

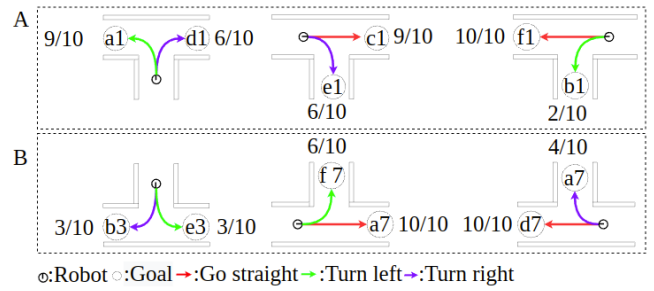


Fig.12 Number of times a correct path was selected

7 結言

本稿では、経路追従行動をカメラ画像を用いた end-to-end 学習で模倣する岡田らの従来手法をベースに、データセットと学習器の入力へ目標方向を加えることで、経路選択をする機能の追加を提案した。また、シミュレータ上での実験を実環境に移す際に、問題となった学習時間の長さについて、2 つのアプローチを試みることによって学習時間を約 67%削減した。加えて、実環境での実験を行い、有効性の検証を行った。実験結果より、学習器へ目標方向を与えることで、指定した経路へ走行する挙動が確認できた。

参考文献

- [1] 岡田真也, 清岡優祐, 上田隆一, 林原靖男: “視覚と行動の: end-to-end 学習により経路追従行動をオンラインで模倣する手法の提案”, 計測自動制御学会 SI 部門講演会 SICE-SI2020 予稿集, pp.1147–1152(2020)
- [2] 岡田真也, 清岡優祐, 春山健太, 上田隆一, 林原靖男: “視覚と行動の end-to-end 学習により経路追従行動をオンラインで模倣する手法の提案 -経路追従行動の修正のためにデータセットを動的に追加する手法の検討”, 計測自動制御学会 SI 部門講演会 SICE-SI2021 予稿集, pp.1066–1070(2021)
- [3] Bojarski, Mariusz, et al. “End to end learning for self-driving cars.” arXiv:1604.07316(2016)
- [4] Moridian, Barzin, Anurag Kamal, and Nina Mahmoudian. “Learning navigation tasks from demonstration for semi-autonomous remote operation of mobile robots.” 2018 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR). IEEE, (2018)
- [5] Felipe Codevilla et al. “end-to-end driving via conditional imitation learning.” arXiv: 1710.02410(2018)
- [6] 春山健太, 藤原柁, 清岡優祐, 岡田真也, 上田隆一, 林原靖男, “視覚と行動の end-to-end 学習により経路追従行動をオンラインで模倣する手法の提案 経路選択機能の追加”, 日本機械学会ロボティクス・メカトロニクス講演会 '22 予稿集,(2022).
- [7] Jeffrey hawke et al. “urban driving with conditional imitation learning”. arxiv: 1912.00177, 2019.
- [8] ros-planning, navigation レポジトリ.
<https://github.com/ros-planning/navigation>.
(最終閲覧日: 2 月 9 日)
- [9] waypoint_nav レポジトリ.
https://github.com/open-rdc/waypoint_nav.git.
(最終閲覧日: 2 月 9 日)
- [10] weighted_graph レポジトリ.
https://github.com/masakifujiwara1/weighted_graph.git
(最終閲覧日: 2 月 9 日)
- [11] 清岡優祐, 岡田真也, 岩井一輝, 上田隆一, 林原靖男, “視覚と行動の end-to-end 学習により経路追従行動をオンラインで模倣する手法の提案 データセットと生成された経路追従行動の解析”, 計測自動制御学会 SI 部門講演会 SICE-SI2021 予稿集, pp.1071–1075(2021)
- [12] He, Haibo, and Edwardo A. Garcia. “Learning from imbalanced data.” IEEE Transactions on knowledge and data engineering 21.9 (2009): 1263–1284.