

# 卒業論文

## 視覚と行動の end-to-end 学習により経路追従行動を オフラインで模倣する手法の提案

(学習用データを走行中に自動収集するシステムの構築と検証)

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

(Development and validation of an automated training data collection system during driving)

2025年12月29日 提出

指導教員 林原 靖男 教授

千葉工業大学 先進工学部 未来ロボティクス学科

22C1048 小山雄矢



# 概要

## 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

(学習用データを走行中に自動収集するシステムの構築と検証)

本論文では、高橋らによって提案された、事前に収集した画像と行動を用いて経路追従を学習するオフライン模倣学習手法に着目し、実環境への適用に向けたデータ収集システムの構築および有効性の検証を行う。

高橋らの手法は、オフライン学習による経路追従の可能性を示したもの、学習用データの収集はロボットを手動で配置し直す作業に依存しており、走行中に自動でデータを蓄積するシステムは構築されていなかった。そのため、多数の姿勢・位置条件を網羅するには多大な時間と労力を要し、実環境への適用における大きな障壁となっていた。そこで本研究では、ロボットの走行中に複数カメラ画像と行動データを同期して自動収集可能なシステムを新たに構築した。これにより、従来手法で課題であったロボットの置き直し作業を排除し、効率的なデータ収集を可能とした。また、実機実装におけるコストと設置負荷を考慮し、従来研究の9台の物理カメラ構成を見直し、3台のカメラ画像から射影変換を用いて9台相当の視野情報を再構成する手法を導入した。

シミュレータを用いた実験の結果、本システムを用いることで、わずか1周の走行データから既存手法と同等の経路追従性能を実現し、学習にかかる作業時間を大幅に短縮できることを確認した。さらに、実環境における走行実験を通じて、提案システムの有効性を検証するとともに、一部区間において経路追従が困難となる事例を確認した。これにより、本手法の実環境適用に向けた課題が明らかとなった。

キーワード: end-to-end 学習、ナビゲーション、オフライン

# abstract

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

(Development and validation of an automated training data collection system during driving)

This paper focuses on the offline imitation learning method for path-tracking proposed by Takahashi et al. and proposes an automated data collection system designed for real-world applications. Although the previous method demonstrated the potential of offline learning, it relied heavily on manual robot repositioning to gather training data. This dependency made it labor-intensive to cover a comprehensive range of poses and positions, posing a significant barrier to practical implementation.

To overcome this, we developed a new system capable of automatically synchronizing and collecting multi-camera images and action data while the robot is in motion. To reduce the hardware cost and installation burden for real-world deployment, we revised the original nine-camera configuration. Instead, we introduced a method that uses projective transformation to reconstruct the equivalent field-of-view of nine cameras from only three physical cameras.

Simulation results confirmed that the proposed system achieves path-tracking performance comparable to existing methods using only a single lap of driving data, significantly reducing the effort required for training. Furthermore, real-world experiments validated the system's effectiveness and clarified remaining challenges for its practical application.

keywords: End-to-End Learning, Navigation, Offline

# 目次

<b>第 1 章</b>	<b>序論</b>	1
1.1	背景 . . . . .	1
1.2	関連研究 . . . . .	2
1.3	目的 . . . . .	5
1.4	論文構成 . . . . .	5
<b>第 2 章</b>	<b>要素技術</b>	6
2.1	地図を用いたルールベース制御器によるナビゲーション . . . . .	6
2.2	ディープラーニング . . . . .	7
2.3	end-to-end 学習 . . . . .	8
2.4	データセット . . . . .	9
2.5	オフライン学習 . . . . .	9
2.6	ミニバッチ学習 . . . . .	9
<b>第 3 章</b>	<b>オンライン手法</b>	10
3.1	オンライン手法の概要 . . . . .	10
3.2	ネットワークの構造 . . . . .	12
<b>第 4 章</b>	<b>オフライン手法</b>	13
4.1	オフライン学習 . . . . .	13
4.2	自動データ収集システム . . . . .	13
4.3	視覚情報の構成と学習設定 . . . . .	14

<b>第 5 章 実験</b>	16
5.1 実験 1 . . . . .	16
5.1.1 実験目的 . . . . .	16
5.1.2 実験装置 . . . . .	16
5.1.3 実験方法 . . . . .	19
5.1.4 実験結果 . . . . .	19
5.2 実験 2 . . . . .	21
5.2.1 実験目的 . . . . .	21
5.2.2 実験方法 . . . . .	21
5.2.3 実験結果 . . . . .	23
5.2.4 検証 . . . . .	23
5.3 実験 3 . . . . .	25
5.3.1 実験目的 . . . . .	26
5.3.2 実験方法 . . . . .	26
5.3.3 実験結果 . . . . .	27
<b>第 6 章 結言</b>	29
<b>参考文献</b>	30
<b>付録</b>	32
<b>謝辞</b>	33

# 図目次

1.1	Path-tracking by end-to-end imitation learning . . . . .	2
1.2	Training the neural network ( source: [3] ) . . . . .	2
1.3	Framework overview ( source: [4] ) . . . . .	3
1.4	Robot navigation in a pedestrian environment ( source: [4] ) . . . . .	4
2.1	Map based navigation using navigation package . . . . .	6
2.2	Structure of deep learning network . . . . .	7
2.3	Structure of general processing . . . . .	8
2.4	Structure of end-to-end learning . . . . .	8
2.5	MNIST dataset from [6] . . . . .	9
3.1	System configuration during network training ( source: [7] ) . . . . .	11
3.2	System configuration after network training ( source: [7] ) . . . . .	11
3.3	Imitatation of angular velocity of navigation using end-to-end learning . .	12
3.4	Structure of network . . . . .	12
4.1	Method of collecting data around the target route . . . . .	15
5.1	Experimental environment in simulator . . . . .	17
5.2	Course to collect data . . . . .	18
5.3	Turtlebot3 waffle with 3 cameras . . . . .	18
5.4	Failure point of the experiment . . . . .	20
5.5	Loss value in the experiment1 . . . . .	20

5.6	Method of collecting data around the target route . . . . .	23
5.7	Loss value in the experiment2 . . . . .	24
5.8	Verification method in previous research ( source: [11] ) . . . . .	24
5.9	Verification of the output of the learning machine . . . . .	25
5.10	Example of the verification results . . . . .	25
5.11	Projection transformation for reducing camera count . . . . .	26
5.12	Projective transformation example . . . . .	27
5.13	Loss value in the experiment2 . . . . .	28

# 表目次

5.1	Number of successes in the experiment . . . . .	19
5.2	目標経路復帰のための各カメラ視点に対する角速度補正量 $v_{\omega\_offset}$ [rad/s]	22
5.3	Number of successes in the experiment . . . . .	24
5.4	Number of successes in the experiment . . . . .	28

# 第1章

## 序論

### 1.1 背景

近年, 配膳ロボットや警備案内ロボットなどの需要が高まり, 自律的に移動できることが求められている. これらのロボットは LiDAR, ホイールオドメトリ, IMU などの様々なセンサから得られるデータに基づいて作成された占有格子地図を用いてナビゲーションを行う.

本研究グループでは, end-to-end 学習により, 視覚に基づく経路追従行動をオンラインで模倣する手法を提案し, その有効性を実験により検証してきた. 岡田らが提案した手法では Fig. 1.1 に示すような LiDAR やホイールオドメトリなどを入力として生成したナビゲーションの行動を, カメラ画像を入力とする行動に模倣する [1] [2]. これにより, 地図を用いたナビゲーションと視覚に基づく経路追従行動の 2 つのナビゲーション手段が得られるため, 状況に応じて高い信頼性が見込まれる方を選択することで, 経路追従を継続できる可能性が高まる.

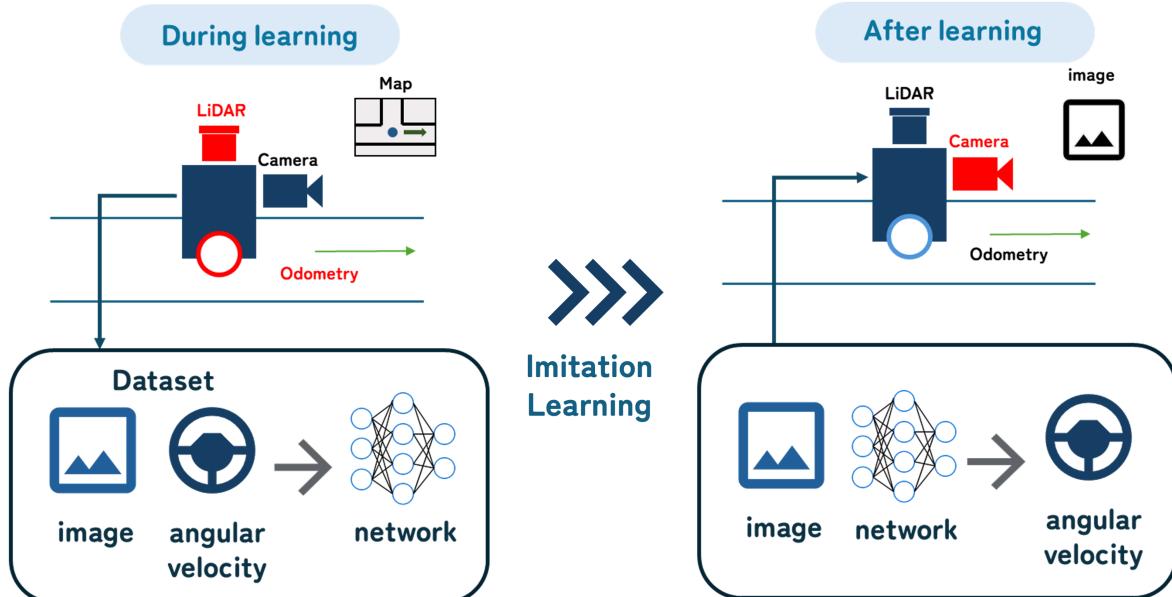


Fig. 1.1: Path-tracking by end-to-end imitation learning

## 1.2 関連研究

視覚を入力として, end-to-end 学習により経路追従を模倣する手法をいくつか紹介する. 例えば, Bojarski らは人が操作したステアリングの角度を end-to-end 学習することで経路追従する手法を提案した. Fig. 1.2 に示すように学習器に, 3 つのカメラ (左・中央・右) の画像と対応するステアリングの角度を入力する. 学習後は出力されたステアリングの角度のみで経路追従することが確認された [3].

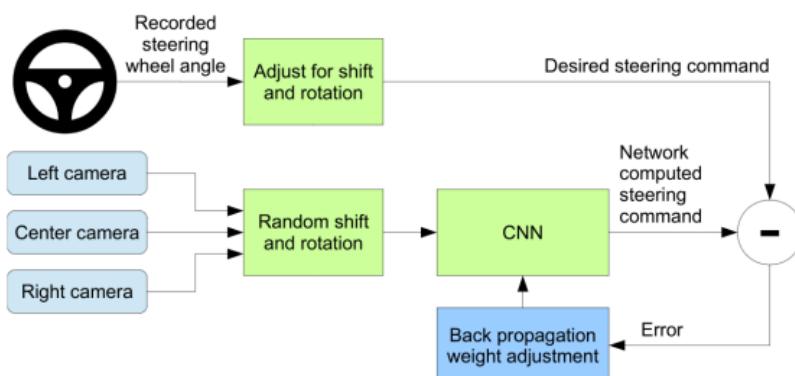


Fig. 1.2: Training the neural network ( source: [3] )

また, Jing Bi らは, 歩行者の多い環境における経路追従行動の模倣に取り組んだ. 具体的には, Fig. 1.3 に示すようなフレームワークを用いて, Fig. 1.4 のような歩行者のいる状況に合わせて, 人が操作しながらデータ収集を行う. ステージ 1 では「どの状況にいるのか」を判断する. ステージ 2 では分類結果に基づいて「具体的にどう動くか」を決定する. 経路から外れたり, 人と衝突したりなどのエラーに遭遇しそうな場合には人が介入を行う. そして, 介入の前後を含めてポリシーを更新する. これにより, 歩行者がいる場合でも経路追従できることが確認された [4].

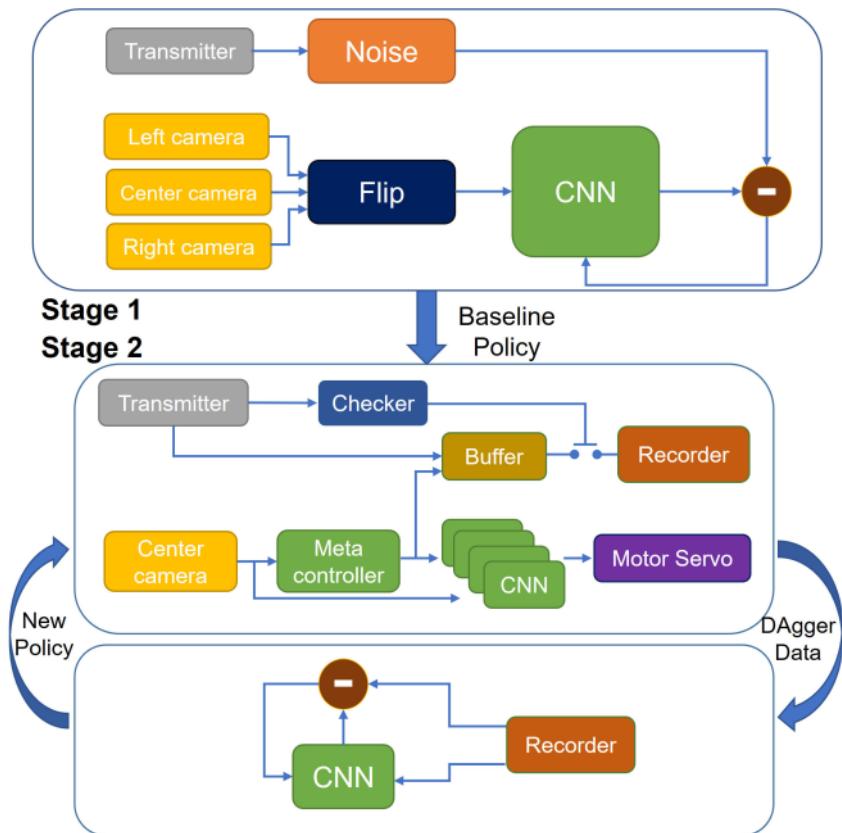


Fig. 1.3: Framework overview ( source: [4] )



Fig. 1.4: Robot navigation in a pedestrian environment (source: [4])

これらの手法が人の手動操作を模倣しているのに対して、岡田らの手法は自動運転を模倣する点が異なる。しかし、岡田らの手法では、データ収集及び学習を行うために、ロボットを経路に沿って走行させ続けることが必要である。そのため、経路追従の成功率を上げるためにロボットを長時間走行させ続ける必要があり、それが問題となっていた。

### 1.3 目的

本論文では、岡田らの手法において問題となっていた学習に要する時間を短縮することを目的とし、高橋らによって提案されたオフラインによる経路追従行動の模倣学習手法を基に、ロボット走行中に複数カメラ画像と行動データを同時に収集可能なシステムを構築する。そして、構築したシステムを用いることで、学習時間の短縮と経路追従性能の両立が可能であるかを検証する。

オフライン模倣学習は他の研究においても行われているが、本研究で提案する手法は、データセットを自動的に収集できる点に特徴がある。さらに、実ロボットへの適用を念頭に置き、経路周辺の視覚情報が経路追従行動に与える影響について、シミュレータを用いた実験により明らかにすることも目的とする。

### 1.4 論文構成

本論文の構成は以下に述べる通りである。第1章では、研究を行う背景や目的を述べた。第2章では、研究に関連する要素技術、第3章では、従来手法について説明する。第4章では、提案手法について説明し、第5章では、実験について説明する。そして、第6章では、本研究の結論を述べる。

## 第2章

# 要素技術

### 2.1 地図を用いたルールベース制御器によるナビゲーション

教師信号としている地図を用いたルールベース制御器によるナビゲーションについて説明する。このナビゲーションには、ROS のパッケージである navigation[5] を使用している。移動ロボットは、Fig. 2.1 のように LiDAR のスキャンデータやオドメトリを入力として自己位置推定と経路計画を行い、これらに基づいて自律走行をする。また、自己位置推定には、amcl(Adaptive Monte Carlo Localization)、経路計画とモータ指令には move\_base[5] を使用している。



Fig. 2.1: Map based navigation using navigation package

## 2.2 ディープラーニング

ディープラーニングとは、人間の神経細胞を模したネットワーク構造のことである。主に、入力層と出力層、その間に中間層（隠れ層）という構成である。中間層を多層化することで複雑な入力情報を処理し、パターンを認識することや、ルールを読み解くことができる。近年では、画像や物体認識、自然言語処理などで活用されている。Fig. 2.2 に構造の一例を示す。

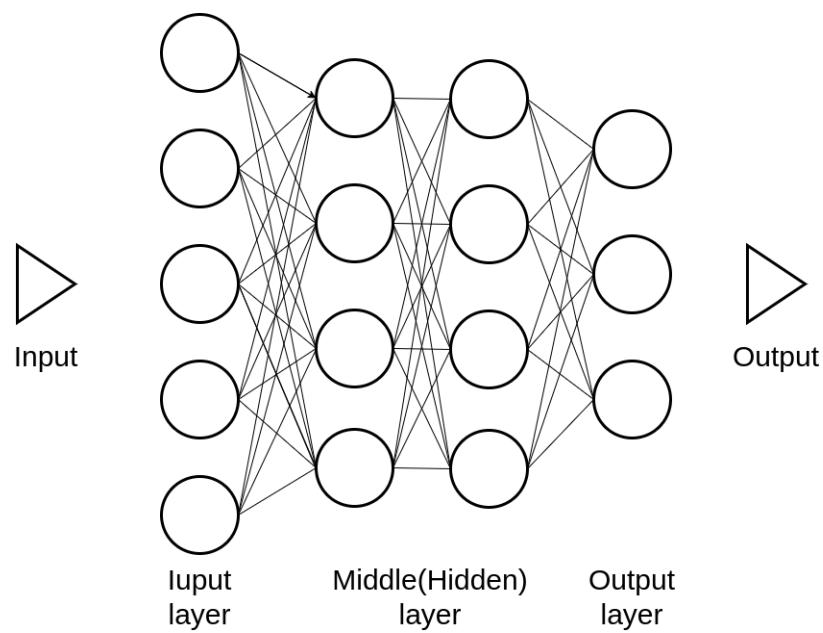


Fig. 2.2: Structure of deep learning network

### 2.3 end-to-end 学習

end-to-end 学習とは、入力から出力までの流れを一括に学習することができる手法である。例として、画像中からの文字認識を行う処理を挙げる。一般的な処理では、Fig. 2.3 のように画像から文字検出を行い、その後に文字分割、最終的に文字認識をする。しかし、end-to-end 学習では、Fig. 2.4 に示すような入力から出力までの流れを一括して学習することができる。

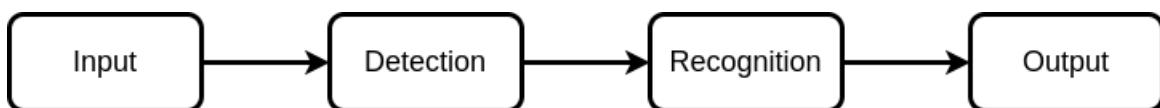


Fig. 2.3: Structure of general processing

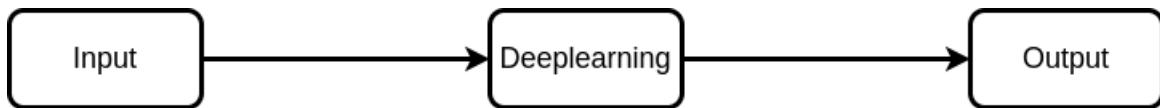


Fig. 2.4: Structure of end-to-end learning

## 2.4 データセット

データセットとは、学習に使用する学習（訓練）データの集合のことである。例として、Fig. 2.5 に示すような 0 から 9 の手書きで書かれた数字の画像セットである MNIST が挙げられる。機械学習や画像認識において多く利用されており、訓練画像 6000 枚とテスト画像 1000 枚で構成されている。



Fig. 2.5: MNIST dataset from [6]

## 2.5 オフライン学習

オフライン学習とは、あらかじめ用意したデータセットを使用して学習を行うことである。これに対して、先行研究用で用いたオンライン学習とは、タスクを行いながらデータ収集をし、そのデータを使用して学習することを指す。

## 2.6 ミニバッチ学習

ミニバッチ学習とは、訓練データをいくつかのグループ（バッチ）に分けて順番に学習を行うことである。特徴としては、勾配更新の頻度が高く、計算量が少なくて済むといったことが挙げられる。これらの特徴を踏まえて、ミニバッチ学習を使用している。

## 第3章

# オンライン手法

### 3.1 オンライン手法の概要

従来から岡田らにより提案してきた手法 [1] (以後, オンライン手法と呼ぶ) に関して述べる。オンライン手法では、地図を用いたルールベース制御器の出力を模倣して、経路追従行動を獲得する。Fig. 3.1 にシステム概要を示す。学習時は、LiDAR とオドメトリを入力として、ROS の navigation [5] により目標角速度を求めて、ロボットを経路追従させる。同時に  $64 \times 48$  にリサイズした 3 つのカメラ画像 (RGB 画像) を入力、目標角速度を出力とするデータを、データセットに加える。そのデータをランダムにピックアップしてリアルタイムに学習する。左右のカメラ画像に対する目標角速度には、それぞれ経路に戻るためのオフセットを加える。これらは経路から外れた場合に、経路に戻る行動が選択されるように加えている。

一定ステップ数の学習を行った後、学習したモデルで経路追従できるか確認する。Fig. 3.2 にシステム概要を示す。中央のカメラ画像を深層学習器に入力して、出力された目標角速度を用いてロボットを制御する。なお、目標の並進速度は  $0.2[m/s]$  で一定とする。

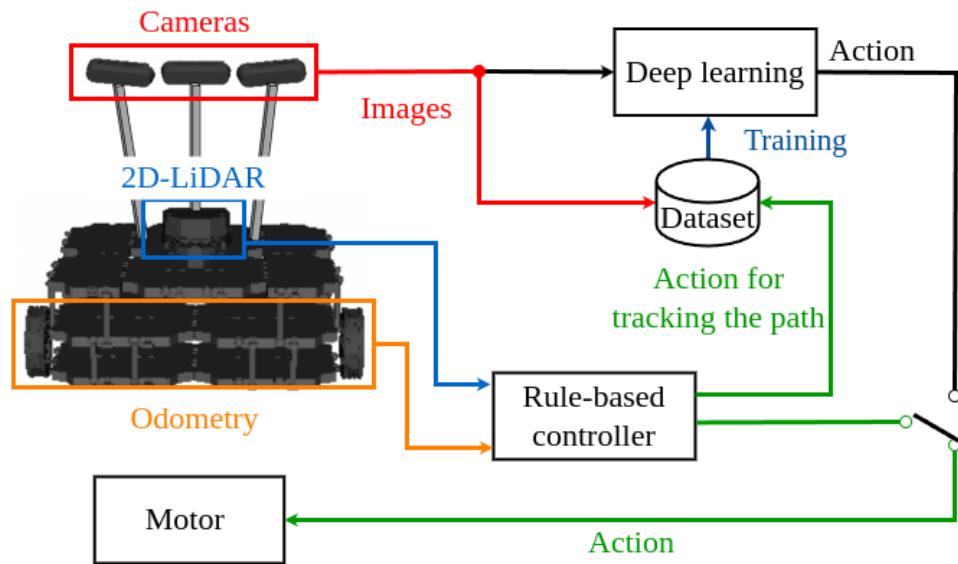


Fig. 3.1: System configuration during network training ( source: [7] )

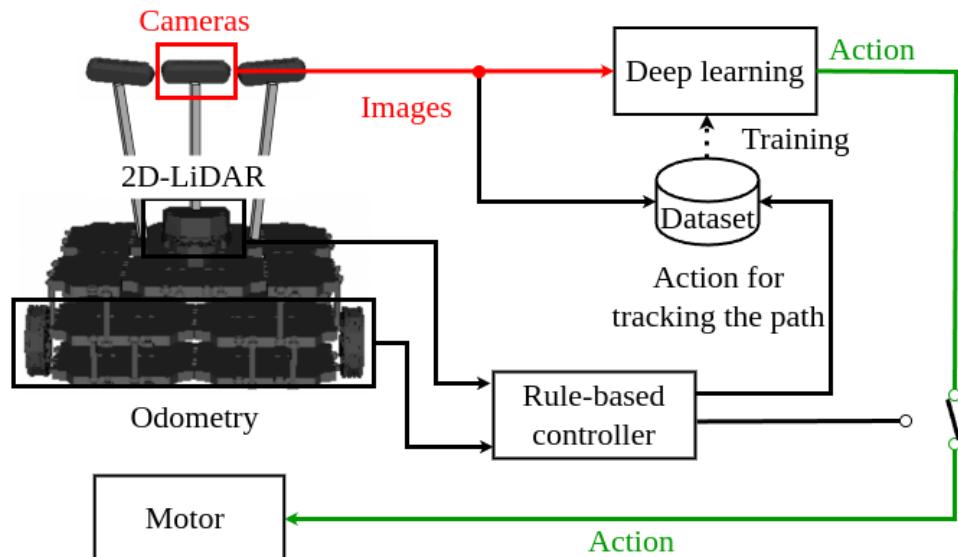


Fig. 3.2: System configuration after network training ( source: [7] )

### 3.2 ネットワークの構造

オンライン手法で用いた学習器のネットワーク構造について説明する。オンライン手法では、Fig. 3.3 に示すように、画像を入力、角速度を出力として end-to-end 学習する。そこで、使用するネットワークを Fig. 3.4 に示す。ネットワーク構造は、入力層 1、畳み込み層 3、全結合層 2、出力層 1 の計 7 層から構成されている。また、ネットワークは畳み込みニューラルネットワーク (CNN) を基にしている。

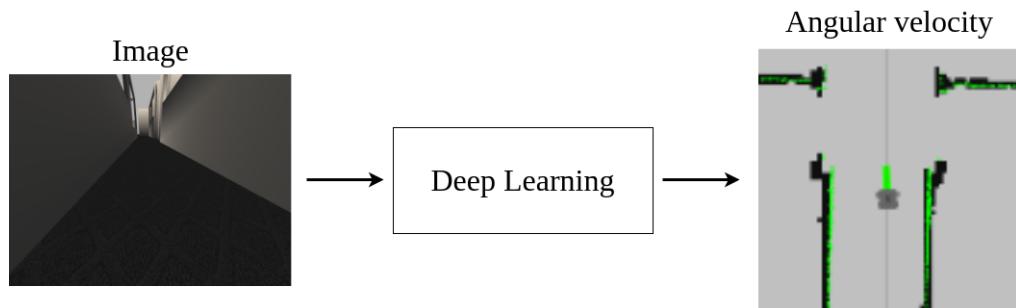


Fig. 3.3: Imitation of angular velocity of navigation using end-to-end learning

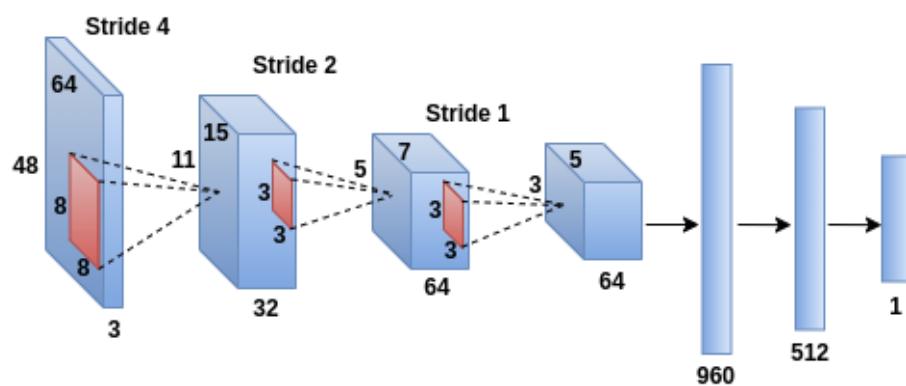


Fig. 3.4: Structure of network

## 第4章

# オフライン手法

### 4.1 オフライン学習

本章では、本研究の基盤となるオフライン模倣学習の概念と、自動でデータセットを収集するシステムについて述べる。オフライン模倣学習とは、熟練者や既存の制御則による行動データを事前に蓄積し、そのデータセットのみを用いて方策を学習する手法である。オンライン学習とは異なり、学習時間短縮の面で利点がある。しかし、高橋らによる先行研究において示された通り、経路追従の堅牢性を高めるには、目標経路から逸脱した位置、姿勢の誤差を含む多様なデータが必要となる。従来の手法では、これらの多様な状態を作り出すために、ロボットを一つずつ手動で配置し直してデータを取得していた。一方で、これらのデータをロボットの走行中に自動的に収集し、そのまま学習に用いるシステムの構築および評価については、十分な検証が行われていない。そこで本研究では、ロボットが経路追従走行を行いながら、多様な視覚情報と行動データを同一タイミングで自動収集するシステムを構築し、データ収集から学習までの工程を効率化するシステムの有効性を検証する。

### 4.2 自動データ収集システム

本研究で構築したシステムは、先行研究 [1] 同様にロボットの走行中に複数台のカメラ画像と制御入力を同期して記録するものである。データ収集には、先行研究に基づくシミュレーション環境およびルールベース制御器を用いる。

Fig. 4.1 に示すようにロボットを目標経路に沿って走行させ、走行中に取得した複数台の力

メラ画像と、ルールベース制御器によるナビゲーション出力である角速度をペアとして保存する。データ収集は、ロボットが 0.2 m 走行するごとにを行い、取得した画像は  $64 \times 48$  画素にリサイズした。制御入力：並進速度を  $0.2 \text{ m/s}$  の一定値とし、ルールベース制御器から出力される角速度のみを学習対象とする。

本研究では、先行研究における実験を再現可能とするため、ロボット走行中に複数台のカメラ画像と行動データを同一タイミングで取得・保存可能なデータ収集システムを構築した。構築したシステムでは、前方に向けて配置した 3 台もしくは 9 台の RGB カメラ画像と、それに応するロボットの角速度指令を同期して取得し、学習用データセットとして自動的に保存することができる。

### 4.3 視覚情報の構成と学習設定

本手法の特徴は、複数台のカメラから得られる広範な視覚情報を学習入力として扱う点にある。実環境への適用を想定し、カメラの台数や配置が学習性能に与える影響を検証する。さらに、システム構成を簡略化するため、少数の物理カメラ画像から射影変換を用いて視野情報を拡張する処理を導入する。これにより、ハードウェアの制約と学習性能の両立を図る。収集されたデータセットを用いたオフライン学習は、バッチサイズ 16 のミニバッチ学習により行う。エポック数は実験条件に応じて最適化し、少量のデータセットでも過学習を抑え、安定した経路追従モデルが獲得できるよう設定した。

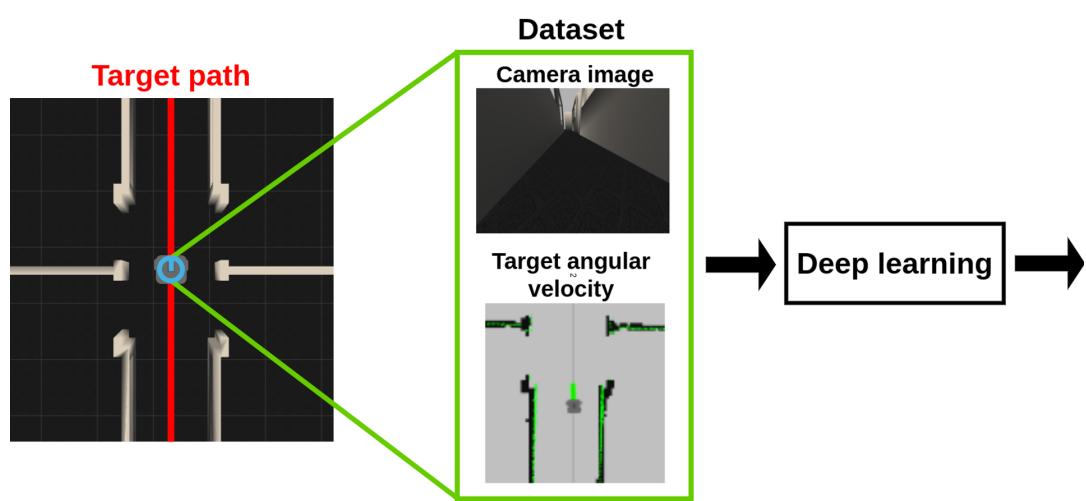


Fig. 4.1: Method of collecting data around the target route

## 第5章

# 実験

### 5.1 実験 1

#### 5.1.1 実験目的

シミュレータ上において、岡田らおよび高橋らの研究の一条件として採用されていた、前方に向けた3台のカメラのみを入力とした経路追従実験を行う。本実験は、先行研究の結果を再現可能であるかを確認することを目的とした追試であり、高橋らの研究ではコースアウトが生じることがあるとされている条件であるが、構築したシステムの挙動を確認するために実施する。

#### 5.1.2 実験装置

本実験では、先行研究における実験を再現可能とするため、ロボット走行中に複数台のカメラ画像と行動データを同一タイミングで取得・保存可能なデータ収集システムを構築した。構築したシステムでは、前方に向けて配置した3台のRGBカメラ画像と、それに対応するロボットの速度指令（並進速度および角速度）を同期して取得し、学習用データセットとして自動保存することができる。

実験環境には、Fig. 5.1 に示すロボットシミュレータ Gazebo[8] 上に構築された Willow Garage 環境 [9] を使用した。本環境内に設けられたコース (Fig. 5.2) を1周走行することで、走行中の画像および行動データを収集した。

ロボットモデルには、Fig. 5.3 に示すように前方に3台のカメラを搭載した TurtleBot3[10]

を使用した。

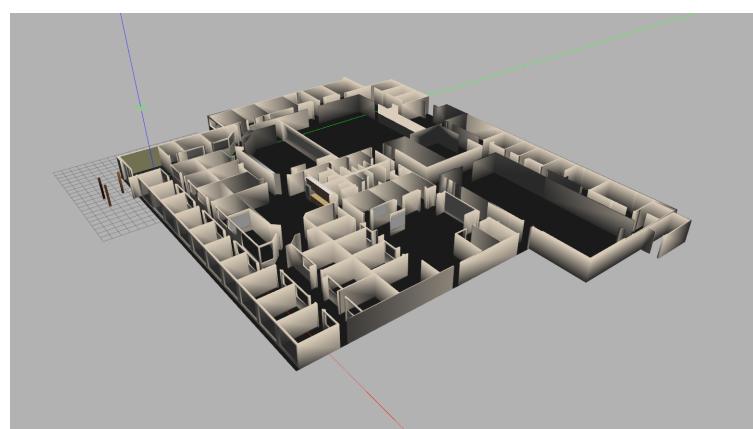


Fig. 5.1: Experimental environment in simulator

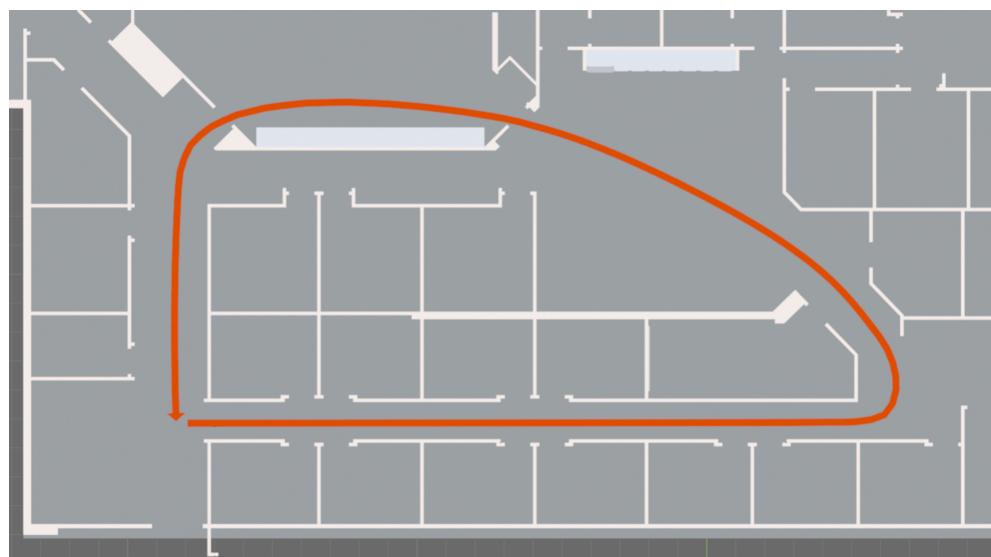


Fig. 5.2: Course to collect data



Fig. 5.3: Turtlebot3 waffle with 3 cameras

### 5.1.3 実験方法

#### 1. データ収集フェーズ

Fig. 5.2 に示す目標経路に沿ってロボットを走行させ、走行中に取得した前方 3 台のカメラ画像とナビゲーション出力である角速度を学習用データとして収集した。カメラの配置間隔および向きは、高橋らの先行研究における議論に基づいて設定した。[7]

#### 2. 訓練フェーズ

データ収集フェーズで取得した 1935 個のデータを用いて、バッチサイズ 16、エポック数 100 のミニバッチ学習によりモデルを学習した。

#### 3. テストフェーズ

Fig. 5.2 に示すコースにおいて、学習済みモデルを用いてロボットを走行させた。ロボットの並進速度は 0.2 m/s とし、コースを 1 周完走できた場合を成功、壁への衝突、またはコースから 10 m 以上逸脱した場合を失敗と定義した。

### 5.1.4 実験結果

実験方法に記載する、2, 3 の手順を 100 回繰り返し、100 のデータを収集する。実験結果を表 5.1 に示す。また、失敗箇所を Fig. 5.4 に示す。図より、×印の地点でロボットが曲がり切れずコースアウトしたことが確認された。

訓練時の損失関数の推移を Fig. 5.5 に示す。学習は収束しているものの、カーブで曲がり切れなかった要因として、目標経路周辺のデータが不足していた可能性がある。

Table 5.1: Number of successes in the experiment

Experiments	Number of successes
Exp.1(epoch100)	42/100

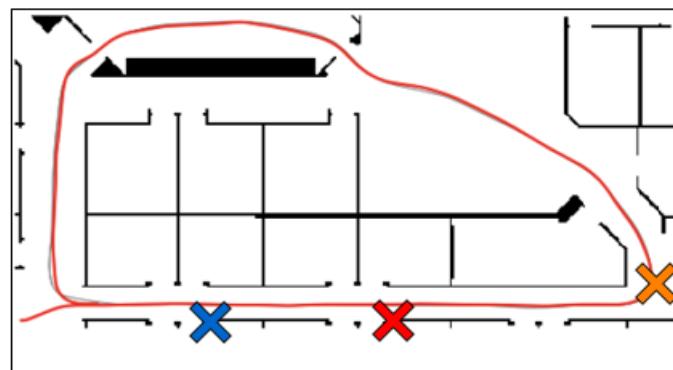


Fig. 5.4: Failure point of the experiment

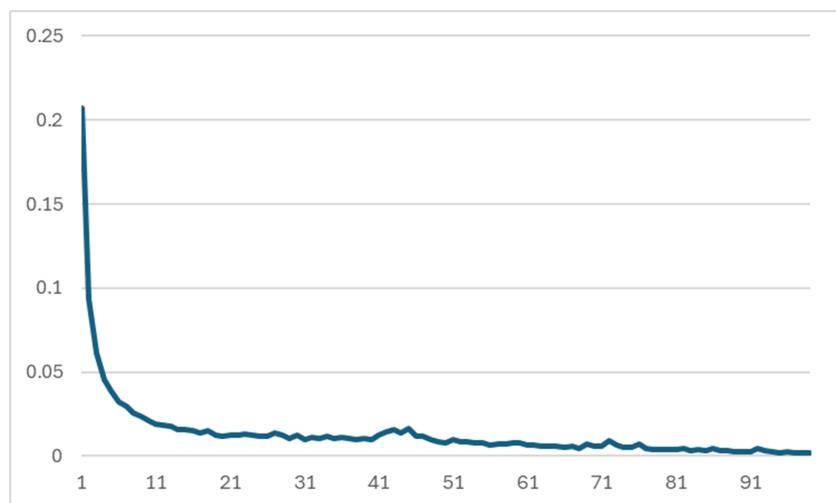


Fig. 5.5: Loss value in the experiment1

## 5.2 実験 2

実験装置・テストフェーズは実験 1 と同様である。

### 5.2.1 実験目的

実験 1 では、構築した走行中データ収集システムを用いて取得したデータにより、経路追従モデルの学習および走行実験を行った。その結果、学習は収束するものの、カーブ区間において失敗が確認された。

高橋らの先行研究では、目標経路に対して平行方向に  $\pm 0.20\text{ m}$  ずらし、さらに  $\pm 5^\circ$  傾けたデータで学習したモデルが最も経路追従できたと述べられている。本研究でも同様に、平行方向に  $\pm 0.20\text{ m}$  離した位置および  $\pm 5^\circ$  の角度から撮影したデータを走行中データ収集システムを用いて追加収集する。これにより、走行中に自動収集された拡張データを用いた場合においても、先行研究と同様の性能向上が得られるかを検証し、本研究で構築したシステムの有効性を評価することを目的とする。

### 5.2.2 実験方法

#### 1. データ収集フェーズ

Fig. 5.6 にデータ収集方法を示す。また、ロボット走行中に学習データを自動蓄積するためのシステム構成について述べる。本システムは、ロボットが経路追従走行を行う過程で、複数視点の画像データとそれに対応する制御入力を同期して自動保存する機能を備えている。具体的な視界構成として、Fig. 5.6 に示す通り、赤線で示す目標経路を基準として左右に平行な  $\pm 0.20\text{ m}$  のオフセットを設けたレーンを定義し、各レーン上に前方へ向けた仮想カメラを配置する。さらに、各カメラのヨー方向に  $\pm 5^\circ$  の回転を加えることで、合計 9 通りの視点（3 レーン  $\times$  3 角度）からの画像を同時に取得可能なシステムを構築した。これにより、ロボットを物理的に置き直すことなく、走行中の 1 ステップごとに、経路逸脱状態からの復帰に必要な視覚情報を網羅的に収集することが可能となった。

Table 5.2: 目標経路復帰のための各カメラ視点に対する角速度補正量  $v_{\omega\_offset}$  [rad/s]

レーン	中央カメラ	左カメラ	右カメラ
lane1 ( 中央 )	0.0226	-0.0487	0.0862
lane2 ( 左 )	-0.2897	-0.2524	-0.3023
lane3 ( 右 )	0.2895	0.2965	0.2770

## 2. オフセット決定フェーズ

走行中にデータを自動収集する際、取得できる実測値はその瞬間のロボットの視点に対応する制御入力のみである。しかし、先行研究において最も経路追従できたモデルは経路から逸脱した状態から復帰するためのデータを学習させたモデルである。本来、これらの経路から逸脱したデータはロボットを手動でずらして配置し直すことで取得していたが、本研究では走行中の自動収集を可能にするため、角速度オフセットを用いた疑似的な教師データを生成する手法を導入する。

本実験では、走行中に取得した中央カメラ画像と角速度のペアに対し、左右に配置された他視点カメラの画像を組み合わせる。この際、各視点画像に対応する角速度は、基準とする経路で得られた実測の角速度に以下の手順で算出したオフセット値を加算することで再現する。信頼性の高いオフセット値を算出するため、まず先行研究の手法に倣い、ロボットを目標経路に対して平行方向に  $\pm 0.20\text{ m}$  ずらし、さらに  $\pm 5^\circ$  傾けた姿勢に手動で配置し、1周分の角速度データを収集する。収集した1周分の角速度データを平均化し、各姿勢における定数的な復帰に必要な操作量をオフセットとして決定する。

そして学習時には、走行中に自動収集された中央の角速度  $v_{\omega\_center}$  に対し、表 5.2 に示すオフセット  $v_{\omega\_offset}$  を加算し、各カメラ視点に対応する角速度  $v_{\omega\_pseudo} = v_{\omega\_center} + v_{\omega\_offset}$  を生成する。なお、本研究では角速度の正方向を左回転と定義する。

## 3. 訓練フェーズ

収集した 5805 個のデータを用い、バッチサイズ 16、エポック数 200 のミニバッチ学習で訓練を行った。

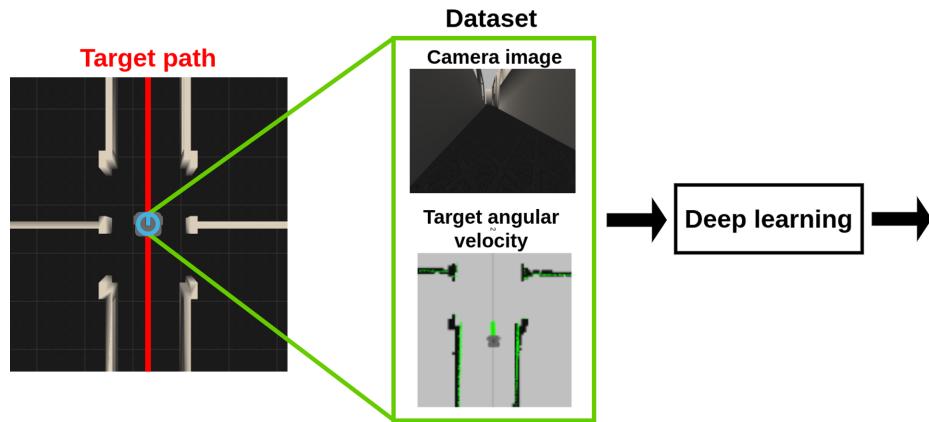


Fig. 5.6: Method of collecting data around the target route

### 5.2.3 実験結果

実験結果を表 5.3 に、損失関数の推移を Fig. 5.7 に示す。経路周辺のデータを入力に含めることでデータ数を増やし、ミニバッチ学習を適用することで安定した経路追従が可能となった。

### 5.2.4 検証

実験 2 では 150/150 の試行すべてが成功し、安定したモデルを生成できた。また、実験中にロボットを意図的に経路から逸脱させたところ、いずれの試行でも経路へ復帰する挙動が確認された。

しかし、この復帰挙動がどの地点においても一様に発生するかは不明である。そこで、先行研究（清岡ら）の方法 Fig. 5.8 を参考に、一定間隔で配置した地点からのモデル出力を記録した。結果を Fig. 5.9 に、拡大図を Fig. 5.10 に示す。

いずれの地点においても経路から外れた際に元の経路へ戻ろうとする挙動が確認でき、構築したシステムの有効性が検証された。

Table 5.3: Number of successes in the experiment

Experiments	Number of successes
Exp.2(epoch200)	150/150

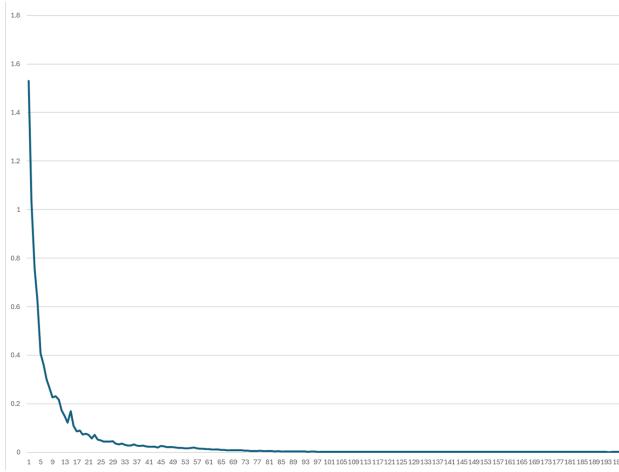


Fig. 5.7: Loss value in the experiment2

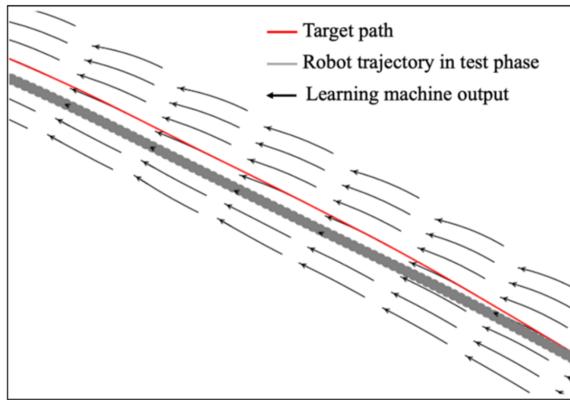


Fig. 5.8: Verification method in previous research ( source: [11] )

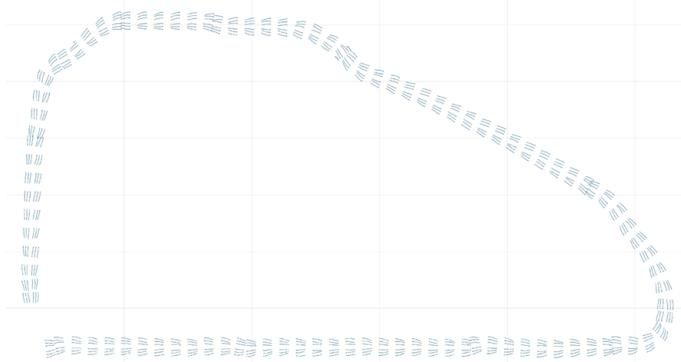


Fig. 5.9: Verification of the output of the learning machine

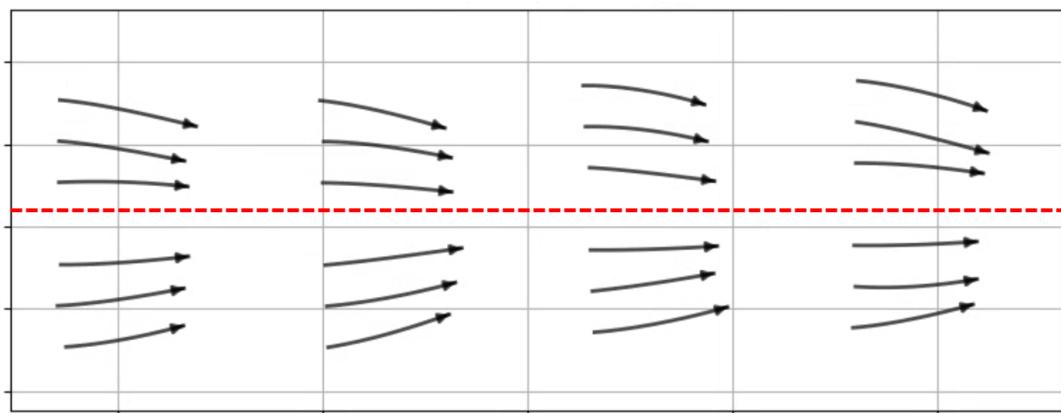


Fig. 5.10: Example of the verification results

### 5.3 実験 3

実験 2 では、平行方向  $\pm 0.20\text{ m}$  と角度  $\pm 5^\circ$  の視点から取得した画像（計 9 台のカメラ相当）を使用したが、実環境ではそのような多数のカメラを搭載することは現実的ではない。このような制約の下でも視点拡張の効果を得るための手法が求められる。本実験では、その一手法として射影変換を用いた視点拡張に着目する。例を Fig. 5.11 に示す。実験装置・データ収集フェーズ・テストフェーズは実験 2 と同様である。

### 5.3.1 実験目的

本実験では、実環境を意識して前方3台のカメラのみを用いて走行中にデータを収集し、射影変換によって視点を拡張するシステムを構築する。さらに、構築したシステムを用いることで、カメラ台数を削減した条件においても十分な経路追従性能が得られるかを検証することを目的とする。

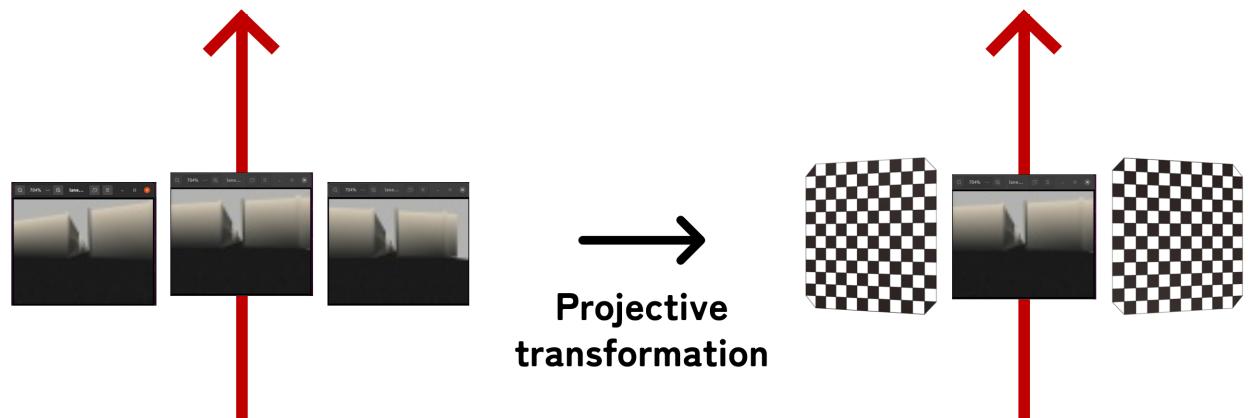


Fig. 5.11: Projection transformation for reducing camera count

### 5.3.2 実験方法

#### 1. データ収集フェーズ

Fig. 5.6 にデータ収集方法を示す。本システムでは、各走行レーン(中央, 左右 $\pm 0.20\text{m}$ )に配置した前方カメラの画像に対し、以下訓練フェーズに示す射影変換を適用することで、ヨー方向に $\pm 5^\circ$ 回転させた視界画像をソフトウェア上で再現する。そのため、本実験野データ収集フェーズでは赤線で示す目標経路を基準として左右に平行な $\pm 0.20\text{m}$ のレーン上に配置した仮想カメラのみを使用する。これにより、ハードウェア構成を簡略化しつつ、経路追従の学習に不可欠な経路から逸脱した姿勢を網羅的に収集することが可能となった。

#### 2. 訓練フェーズ

収集した 1935 個のデータを用い、バッチサイズ 16、エポック数 200 で学習を行った。  
射影変換に使用する焦点距離  $f$  は以下で求める。変換後の画像の例を Fig. 5.12 に示す

$$f_x = \frac{W_{\text{px}}}{2 \tan\left(\frac{\text{FOV}_h}{2}\right)}$$

今回は水平視野角  $\text{FOV}_h = 2.09 \text{ rad}$ 、画像サイズ  $64 \times 48$  を用いた。

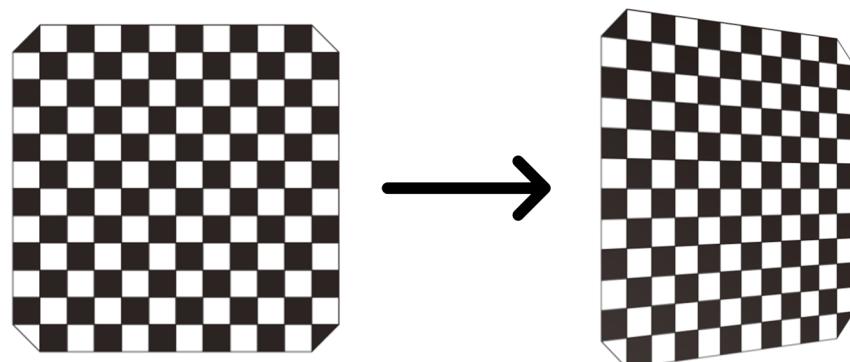


Fig. 5.12: Projective transformation example

### 5.3.3 実験結果

実験結果を表 5.4 に示す。分母の 150 は実験回数を表し、分子は成功回数を示している。その結果、目標経路上および  $\pm 0.2 \text{ m}$  の位置で取得したカメラ画像に対して、 $\pm 5^\circ$  の向きに射影変換を行った場合、100% (150 回中 150 回) の成功率が得られた。また、オンライン手法では学習に約 40 分を要するのに対し、オフライン手法では約 4 分で学習が完了したことから、学習に要する時間を約 1/10 に短縮できることを確認した。ただし、オフライン手法における学習時間 4 分には、実験方法で述べたデータ収集に要する時間は含まれていない。ここで、Fig. 5.13 に、この実験条件で学習したときの loss グラフを示す。図より、100 epoch 程度で loss がほぼ収束しており、その後は緩やかに下降している様子が確認できる。

Table 5.4: Number of successes in the experiment

Experiments	Number of successes
Exp.3(4000step)	150/150

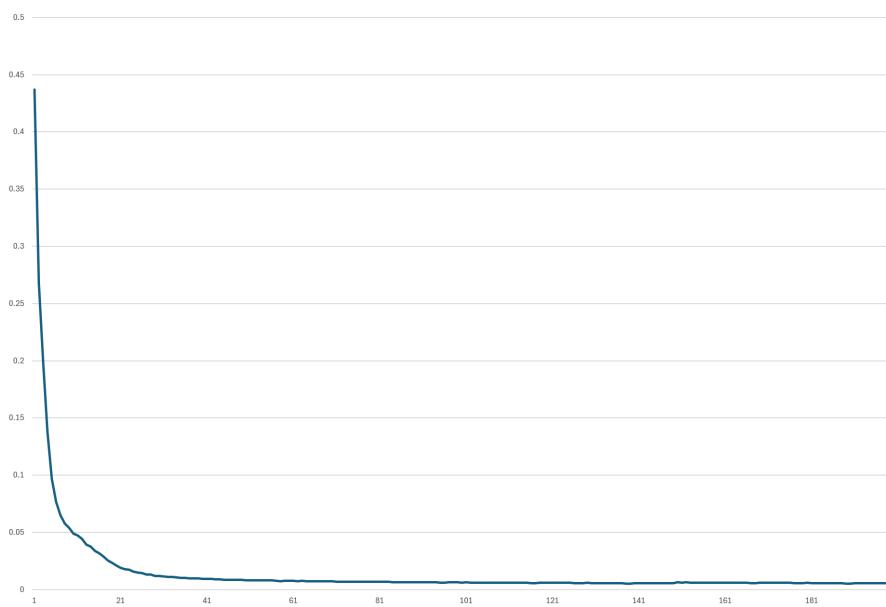


Fig. 5.13: Loss value in the experiment2

## 第6章

### 結言

本論文では、岡田らの手法において課題となっていた学習に要する時間の短縮を目的として、高橋らにより提案されたオフラインによる経路追従行動の模倣学習手法を基に、ロボット走行中に複数カメラ画像と行動データを同時に自動収集可能なシステムを構築した。本研究を通じて得られた知見を以下にまとめる。

第一に、ロボット走行中に複数台のカメラ画像と制御入力を同期して収集するシステムを構築することで、従来手法で不可欠であったロボットの手動配置作業を排除した。これにより、わずか1周の走行データから経路追従に必要なデータセットを網羅的に生成することが可能となり、学習準備に要する時間を大幅に短縮できることを示した。

第二に、実用性を考慮した視覚情報の構成について検証した。9台の物理カメラを用いた構成に加え、3台の物理カメラ画像に射影変換を適用して視野情報を拡張する手法を導入した。シミュレータを用いた実験の結果、射影変換を用いた構成においても、従来の多台数カメラ構成と同等の経路追従性能を維持できることを確認した。これにより、ハードウェア構成の簡略化と効率的なデータ収集の両立が可能となった。

以上の結果から、本研究で構築したシステムは、オフライン模倣学習におけるデータ収集の自動化を可能とし、学習効率と経路追従性能の両立に有効であることが示唆された。また、シミュレーション環境を用いた評価を通して、経路周辺の視覚情報が経路追従行動に与える影響を明らかにし、実ロボットへの適用に向けた有用な知見を得た。

# 参考文献

- [1] 真也岡田, 優祐清岡, 隆一上田, 靖男林原. 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, 2020. 計測自動制御学会 SI 部門講演会 (SICE-SI2020) 予稿集, pp.1147–1152.
- [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] Mariusz Bojarski et al. ”end to end learning for self-driving cars.”. arXiv: 1604.07316(2016).
- [4] Jing Bi, Tianyou Xiao, Qiuyue Sun, and Chenliang Xu. Navigation by imitation in a pedestrian-rich environment, 2018. arXiv:1811.00560.
- [5] ros-planning, navigation.  
<https://github.com/ros-planning/navigation>.  
最終閲覧日 2025 年 12 月 29 日.
- [6] The mnist database of handwritten digits.  
<http://yann.lecun.com/exdb/mnist/>.  
最終閲覧日 2025 年 12 月 29 日.
- [7] 高橋祐樹, 白須和暉, 藤原恆, 上田隆一, 林原靖男. ”視覚と行動の end-to-end 学習による経路追従行動の模倣-データセットを収集してオフラインで訓練する手法の検討- ”. 日本機械学会ロボティクス・メカトロニクス講演会 2023 講演論文集, 2023. 2P1-G07.
- [8] gazebo.  
<http://gazebosim.org/>.

最終閲覧日 2025 年 12 月 29 日

- [9] Koenig, Nathan, and Andrew Howard. "design and use paradigms for gazebo, an open-source multi-robot simulator.". 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566). Vol. 3. IEEE, pp.2149-2154(2004).

最終閲覧日 2025 年 12 月 29 日.

- [10] Turtlebot3 robotis emanual.robotis.

<https://emanual.robotis.com/docs/>.

最終閲覧日 2025 年 12 月 29 日.

- [11] 清岡優祐, 岡田眞也, 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.1072-1075(2021).

# 付録

## 謝辞

本研究を進めるにあたり、2年に渡り、熱心にご指導を頂いた林原靖男教授に深く感謝いたします。また、日頃から研究へのアドバイス、指導、サポートしてくださった高橋祐樹様、今井悠月様、並びにロボット設計制御研究室の皆様には、心から深く感謝を申し上げます。

そして何より、生活面で私を支えてくださった両親に感謝します。