

修士論文

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とシナリオに基づく目的地へのナビゲーション-

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

-Adding route selection function and scenario-based
navigation to the destination-

2024 年 3 月 提出

指導教員 林原 靖男 教授

千葉工業大学 先進工学研究科 未来ロボティクス専攻

22S1031 春山健太

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

-経路選択機能の追加とシナリオに基づく目的地へのナビゲーション-

自律移動ロボットの多くは、占有格子地図などのメトリックマップに基づくナビゲーションを用いている。本研究室ではこれまで、このメトリックマップに基づくナビゲーションの行動を視覚を入力として模倣学習することで、視覚に基づくナビゲーションを獲得する手法を提案してきた。また、実験により視覚を入力として、ロボットが一定の経路を周回できることを確認している。この手法では、経路を追従する行動の獲得を目的としており、走行する経路は一定に制限されていた。任意の目的地に向かって移動するためには、環境中の分岐路の検出、分岐路での適切な進行方向の提示、動的に経路を選択して移動する機能が必要であると考えられる。

本論文では、これらの機能を岡田らの従来手法に追加することで、走行する経路を一定の経路から、任意の目的地に向けた経路への拡張を試みた。はじめに動的に経路を選択して移動する機能を追加した。検証のため、シミュレータを用いた実験を行い、視覚に基づくナビゲーションにおいて、同一の分岐路で、動的に経路を選択して移動する様子を確認した。次に、視覚から分岐路を検出し、目的地に向けた進行方向を提示する機能を追加することで、視覚に基づいて目的地まで自律移動するシステムを構築した。さらに、実ロボットを用いた実験から、構築したシステムにより視覚に基づいて経路を追従して目的地へ到達可能であることを確認した。

キーワード: end-to-end 学習 自律移動ロボット ナビゲーション

abstract

title

Many autonomous mobile robots use navigation based on metric maps, such as occupancy grid maps. In our laboratory, we have been proposing a method to acquire vision-based navigation by imitating actions of navigation based on these metric maps using vision as input. Furthermore, experiments have confirmed that robots can repeatedly navigate a set course using vision as input. This method aims to acquire actions for following a route, and the routes for navigation were limited to a fixed set. To move towards an arbitrary destination, it is considered necessary to have functions for detecting branch points in the environment, presenting the appropriate direction at these branch points, and dynamically selecting and moving along a route. In this paper, we attempt to expand the route for navigation from a fixed course to a route towards an arbitrary destination by adding these functions to the conventional methods of Okada et al. Firstly, we added the function to dynamically select and move along a route. For verification, experiments were conducted using a simulator, and it was confirmed that the system could dynamically select and navigate different routes at the same branch point in vision-based navigation. Next, by adding the function to detect branch points from vision and present the direction towards the destination, we constructed a system capable of autonomous movement to the destination based on vision. Furthermore, experiments using real robots confirmed that the constructed system could follow a route based on vision and reach the destination.

keywords: end-to-end learning, autonomous mobile robot, navigation

目次

第 1 章	序論	1
1.1	背景	1
1.2	関連研究	4
1.3	目的	7
1.4	論文構成	7
第 2 章	要素技術	8
2.1	メトリックマップに基づくナビゲーション	8
第 3 章	先行研究	10
3.1	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	10
3.2	トポロジカルマップとシナリオ	13
3.2.1	トポロジカルマップ	13
3.2.2	シナリオ	14
第 4 章	動的に経路を選択して移動する機能の追加	15
4.1	概要	15
4.2	経路選択機能を追加したシステム	15
4.3	シミュレータを用いた経路選択機能を確認する実験	18
第 5 章	視覚に基づいて目的地まで自律移動するシステムの構築	23
5.1	構築したシステムの概要	23
5.2	経路追従モジュール	25

5.3	通路分類モジュール	28
5.4	シナリオモジュール	31
5.5	実ロボットを用いた実験	33
5.5.1	実験装置	33
5.5.2	実験方法	34
5.5.3	実験結果	39
第6章 おわりに		41
6.1	結論	41
参考文献		42
謝辞		45

図目次

1.1	Imitation method of path-tracking behavior	2
1.2	Path selection and following behavior based on camera images and target direction by imitation learning.	3
1.3	Path selection and following behavior based on camera images and target direction by imitation learning.	3
1.4	End-to-end driving via conditional imitation learning (Quoted from[8])	4
1.5	Overview of Seiya and others proposed method (Quoted from[18]) . .	5
1.6	Embodied instruction following with LM-Nav (Quoted from[19]) . . .	5
1.7	Miyamoto and others used topological map (Quoted from[15])	6
1.8	Observation of robot behavior using semantic segmentation (Quoted from[15])	6
2.1	Metric map based navigation (Quoted from[26])	9
3.1	Structure of the Okada and others proposed system (Quoted from[22])	11
3.2	Rule-based actions apart from the robot's actions Quoted from[22] . . .	12
3.3	Structure of the network used in the method of Okada and others. (Quoted from [22])	12
3.4	Topological map format proposed by Shimada and others (Quoted from[26])	13
3.5	Example scenarios (Quoted from [25])	14
4.1	System of the imitation learning with added function of selecting path .	16

4.2	Path selection and following behavior based on camera images and target direction by imitation learning (Quoted from [25])	16
4.3	Structure of network with added function of selecting path	17
4.4	TurtleBot3 Waffle Pi	18
4.5	Experimental environment of selecting path (Quoted from [24])	19
4.6	Selecting a path at the T-junction (Quoted from [24])	19
4.7	Route for experiment of selecting path (Quoted from [24])	20
4.8	Number of times a correct path was selected (Quoted from [24]) . . .	21
4.9	The robot moving while selecting a path (Quoted from [24])	22
5.1	Overview of constructed system (Quoted from [25])	24
5.2	Number of data in the target direction per 10000 steps in the previous experiment Quote from [27]	25
5.3	Aggressive meandering (Quote from [27])	26
5.4	Structure of the network of route selection module	27
5.5	Overview of the corridor classification module (Quoted from [25]) . . .	28
5.6	Types of corridor features (Quoted from [25])	28
5.7	Network of corridor classification module	29
5.8	Classification labels registered in the metric map Quoted from [25] . . .	30
5.9	Example of topological map and created scenario (Quoted from [25]) . .	32
5.10	Experimental setup (Quoted from [25])	33
5.11	Experimental environment (Quoted from [25])	34
5.12	Route used for learning (Quoted from [25])	35
5.13	Scenarios used in the experiment (Quoted from [25])	36
5.14	Example of narrow passage in experimental environment	37
5.15	An example of the robot applied the constructed system (Quoted from [25])	40

表目次

4.1	Target direction and data for imitation learning	17
4.2	Location where the robot went off course	21
5.1	Target direction and data for path-following module	27
5.2	Number of classes in the dataset	38
5.3	The weights assigned to each class in the experiment	39
5.4	Evaluation results of the corridor classification module on test data.	39

第1章

序論

1.1 背景

一般的に，移動ロボットを目的地まで誘導する制御はナビゲーションと呼ばれ，工場内の巡回や，警備，配送業務などを行う自律移動ロボットに活用されている。これらの多くのロボットは，ナビゲーションに占有格子地図などの事前に作成したメトリックマップを用いている。例として，つくば市内の公道で行われる技術チャレンジであるつくばチャレンジにおいて，2019年度の原らの技術調査 [9] では，参加した57チームの中で，50のチームがメトリックマップに基づくナビゲーションを用いている。また，多くのチームがメトリックマップに基づくナビゲーションに，LiDAR や IMU などのセンサ，ホイールエンコーダから算出したオドメトリを利用している。

これらの LiDAR やオドメトリ，メトリックマップに基づくナビゲーションの他に，カメラ画像と深層学習に基づくナビゲーションが研究 [12][17] されている。その中で，本研究室の岡田ら [22][21] は Fig. 1.1 に示すようにメトリックマップに基づくナビゲーションによる行動を，視覚を入力とした行動に模倣することで，視覚に基づくナビゲーションを獲得する手法を提案した（以後，岡田らの従来手法と呼ぶ）。また，提案した手法の有効性を，シミュレータと実ロボットを用いて検証する実験を行い，視覚に基づいて一定の経路を追従できることを確認している。この手法の利点として，一般的な模倣学習が人の挙動を模倣するのに対して，本手法では，メトリックマップに基づくナビゲーションの行動を模倣するため，データセットを収集する手間を省くことが可能という特長がある。

岡田らの従来手法では、経路を追従する行動の獲得を目的としていた。そのため、走行する経路は一定という制限があり、任意の目的地に向けて移動することはできない。Fig. 1.2 に示す目的地に向けて移動するためには、環境中の分岐路を検出し、分岐路で適切な進行方向の提示、動的に経路を選択して移動する機能が必要と考えられる。

そこで、本論文では、岡田らの従来手法に対し、Fig. 1.3 のような分岐路を視覚に基づいて検出し、目的地に向けた分岐路での進行方向の提示、動的に経路を選択して移動する機能を追加する。これにより、走行する経路を一定から、任意の目的地に向けた経路へ拡張することで、視覚に基づいて、経路を追従して目的地へ到達できる可能性がある。

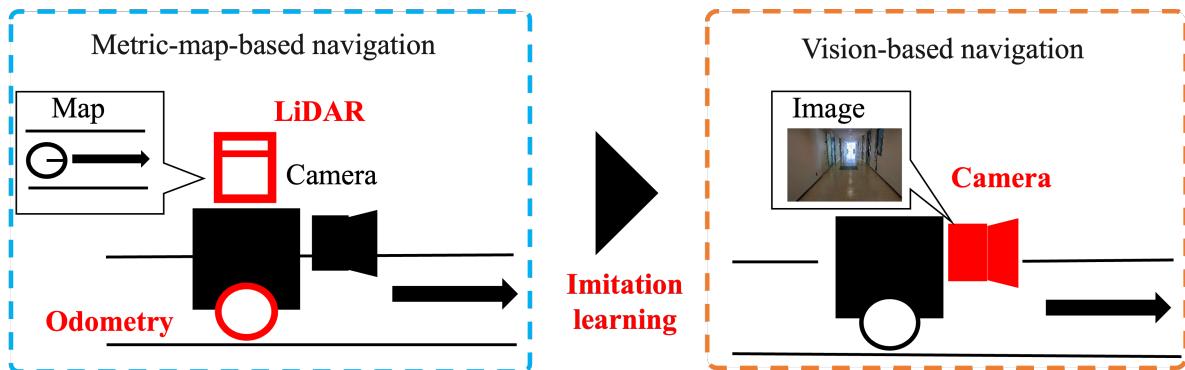


Fig. 1.1: Imitation method of path-tracking behavior

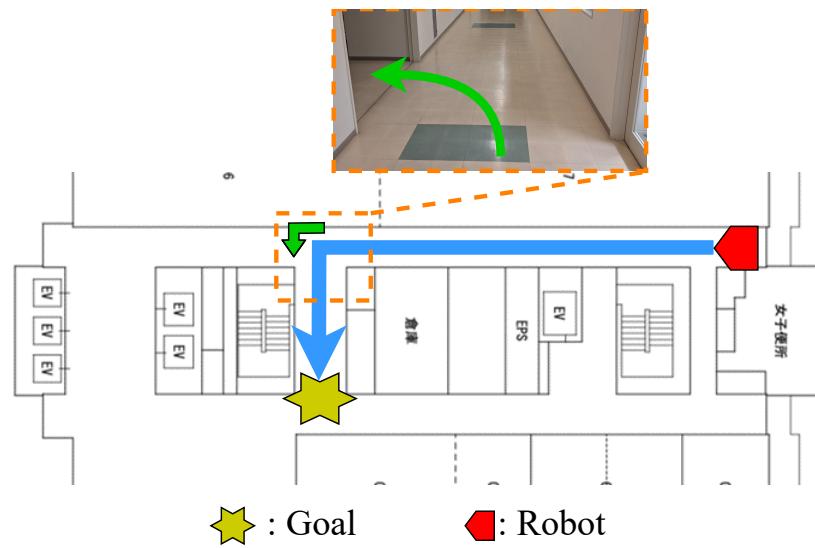


Fig. 1.2: Path selection and following behavior based on camera images and target direction by imitation learning.



Fig. 1.3: Path selection and following behavior based on camera images and target direction by imitation learning.

1.2 関連研究

分岐路で動的に経路を選択して移動する機能を有する視覚に基づくナビゲーションを、模倣学習により獲得する手法はいくつか提案されている。Felipe ら [8] は視覚を入力とする模倣学習を、左折や右折などの行動ごとにモデルを分けて行った。このモデルを目標とする経路方向のベクトルによって切り替えることで、Fig. 1.4 に示す経路選択が可能な視覚に基づく自律移動を行なっている。



Fig. 1.4: End-to-end driving via conditional imitation learning (Quoted from[8])

Seiya ら [18] は Fig. 1.5 に示すようにカメラ画像と目的地への方向を示すベクトルを入力、ステアリングの角度を出力として模倣学習した。これにより屋内外の分岐路で特定の経路が選択できることを確認している。

これらの研究により、経路の情報を含めて模倣学習することで、経路選択が可能な自律移動を獲得できることが確認されている。第 4 章ではこれらの研究を参考に、岡田らの従来手法に対し、経路を選択する機能の追加を試みる。

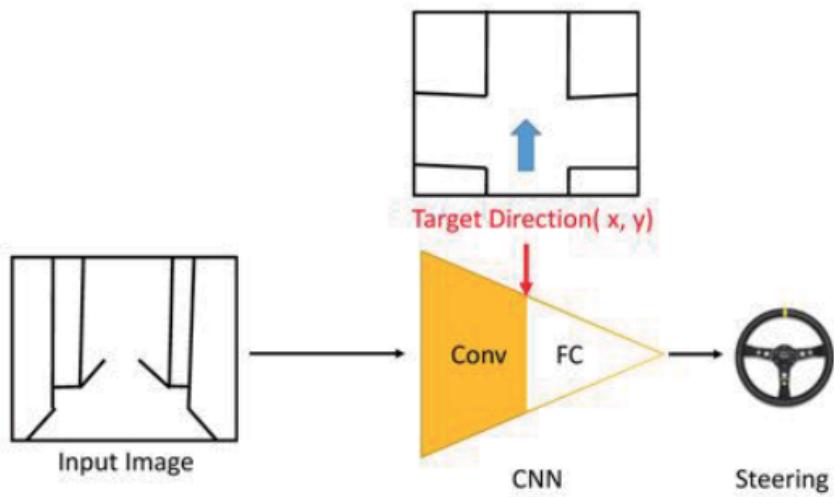


Fig. 1.5: Overview of Seiya and others proposed method (Quoted from[18])

次に任意の目的地に向けて移動が可能な視覚に基づくナビゲーションに関する研究について述べる。Dhruv ら [19] は、大規模な事前学習モデルを用いて、自然言語指示と画像を組み合わせたナビゲーションを行う手法を提案している。この研究では、Fig. 1.6 に示すように、自然言語で指示されたランドマークを視覚によって検出し、これらのランドマークを経由して目的地へのナビゲーションをしている。実験では、停車中の車、道路標識、曲がり角といった道路の特徴をランドマークとして利用している。



Fig. 1.6: Embodied instruction following with LM-Nav (Quoted from[19])

Miyamoto ら [15] は環境中のランドマークを含む Fig. 1.7 に示すトポロジカルマップと, Fig. 1.8 に示すセマンティックセグメンテーションを用いた, 視覚に基づくナビゲーション手法を提案している. 実験では, 木などの植物や, 建物, 交差点などのランドマークを視覚によって検出し, これらをトポロジカルマップ上での経路の決定に利用している.



Fig. 1.7: Miyamoto and others used topological map (Quoted from[15])



Fig. 1.8: Observation of robot behavior using semantic segmentation (Quoted from[15])

これらの研究では, 視覚に基づくナビゲーションにおいて, 目的地への自律移動に標識や交差点といったランドマークを用いている. 本論文では, 分岐路で動的な経路を選択した移動を行うことから, 「分岐路」などの通路の特徴をランドマークとし, これを視覚に基づいて検出する.

1.3 目的

本論文では、岡田らの従来手法に対し、視覚から通路の特徴を検出し、目的地に向けた進行方向を提示、動的に経路を選択して移動する機能を追加する。これにより、視覚に基づいて任意の目的地まで自律移動するシステムを構築する。また、構築したシステムにより目的地までカメラ入力のみで自律移動できるかを、実ロボットを用いた実験により検証することを目的とする。

1.4 論文構成

1章では本論文における、背景や関連研究、目的を述べた。2章では本論文に関連する要素技術を述べる。4章では、経路選択機能の追加とその機能をシミュレータを用いて確認する実験について述べる。5章では、構築するシステムとその有効性を実ロボットを用いて検証する実験について述べる。6章では、本論文の結論と展望を述べる。

第2章

要素技術

本章では、本研究に関連する要素技術を述べる

2.1 メトリックマップに基づくナビゲーション

メトリックマップに基づくナビゲーションについて述べる。このナビゲーションは、LiDAR やオドメトリなどのセンサとメトリックマップを用いて、自己位置推定、経路計画を行い、目的地まで自律移動する。メトリックマップに基づくナビゲーションには ROS Navigation stack[5] がよく用いられている。ROS Navigation stack は主に下記の 3 つの機能を提供する。

- パーティクルフィルタによって自己位置推定を行うモンテカルロ自己位置推定 (MCL)
- 障害物認識などの局所的または地図全体の大域的なコスト計算と、その結果に基づく経路計画
- 経路に追従する並進速度や角速度などの速度指令

Fig. 2.1 に ROS Navigation stack によるメトリックマップに基づくナビゲーションの様子を示す。本論文では、図中の赤の経路を追従することを経路追従と呼ぶ。岡田らの従来手法、本論文で述べる視覚に基づくナビゲーションでは、この ROS Navigation stack による経路追従行動を模倣する。

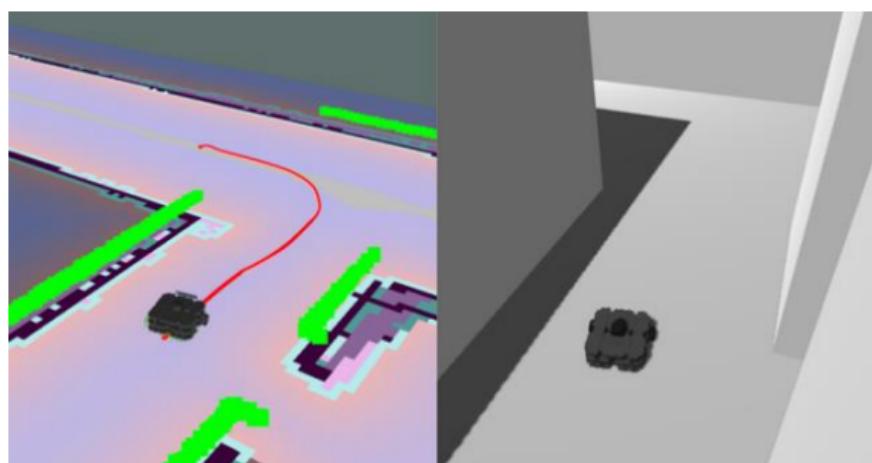


Fig. 2.1: Metric map based navigation (Quoted from[26])

第3章

先行研究

本論文の議論のベースとなる、岡田らの従来手法と島田らが提案したトポロジカルマップの形式、単語の組み合わせで経路を表現したシナリオについて述べる

3.1 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

岡田らの従来手法について述べる。岡田らの従来手法では、メトリックマップに基づくナビゲーションの経路追従行動を end-to-end 学習により、視覚を入力とした行動へ模倣する。これにより、視覚に基づくナビゲーションを獲得できる。手法に基づいて構築されたシステムを Fig. 3.1 に示す。

まずははじめに、学習器の訓練を行う。学習器の訓練時、LiDAR やオドメトリを入力とするメトリックマップに基づくルールベース制御器を用いて、設定した経路を追従する。その際、ロボットに取り付けた、カメラから得た RGB 画像とルールベース制御器が出力するヨー方向の角速度をペアとして 0.2 秒の周期でデータセットへ加える。次にデータセットから、バッチサイズを 8 として教師データを抽出し、0.2 秒の周期で end-to-end 学習する。このデータセットへのデータの追加から学習までの 1 連の流れを 1 ステップとしている。カメラ画像の収集中には、3 つのカメラを用いる。左右のカメラ画像に対するヨー方向の角速度は、経路に戻るようなオフセット ($\pm 0.2\text{rad/s}$) を加える。

学習器の訓練後は、学習器へ中央のカメラから得た RGB 画像を入力し、出力されるヨー方

向の角速度を用いて経路を追従する。この時、並進速度は 0.2m/s の一定の値を用いる。

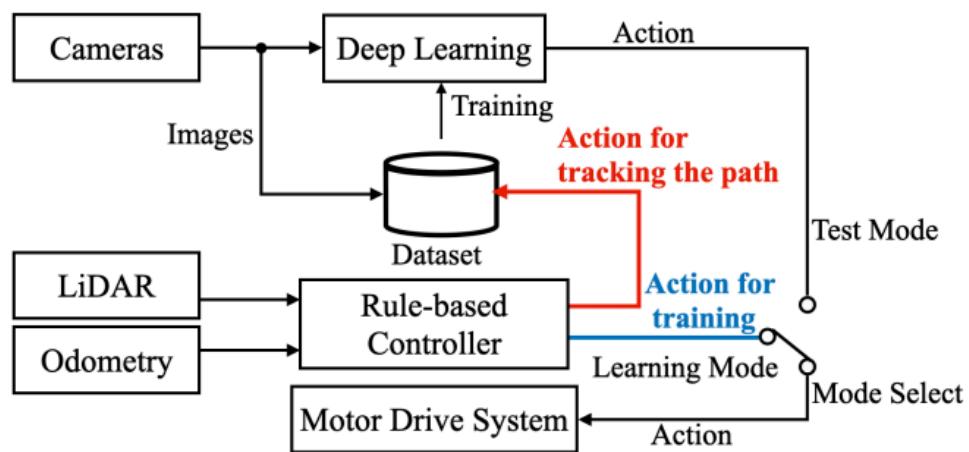


Fig. 3.1: Structure of the Okada and others proposed system (Quoted from[22])

この岡田らの従来手法は、ルールベース制御器を教師データの収集に用いることで、データセットの収集に人間の操作が不要となること、Fig. 3.2 に示すように学習のデータセットに加える行動と学習時にロボットを制御する行動を別々に扱うことで、常に経路に戻る行動のみをデータセットに追加できるといった特長がある。

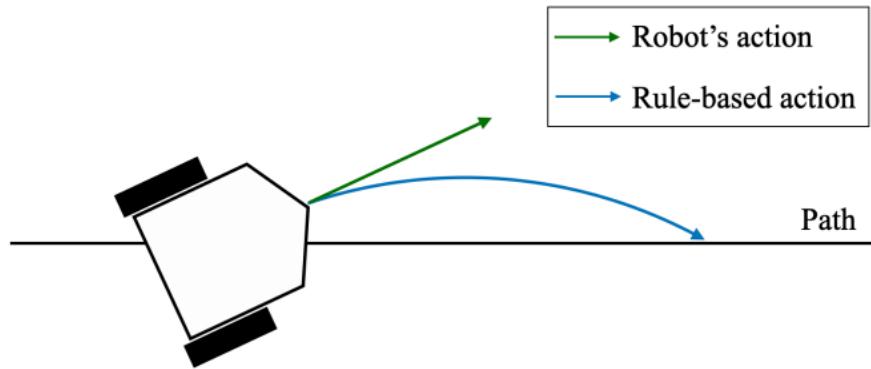


Fig. 3.2: Rule-based actions apart from the robot's actions Quoted from[22]

Fig. 3.3 に岡田らの従来手法における学習器のネットワーク構造を示す。ネットワークは、入力層 1, 畳み込み層 3, 全結合層 2, 出力層 1 の全 7 層で構成される。活性化関数として ReLU[16], 最適化アルゴリズムに Adam[13], 損失関数としては Softmax-cross-entropy を用いる。学習器は入力を 64×48 の RGB 画像データ、出力をロボットのヨー方向の角速度として end-to-end 学習する。

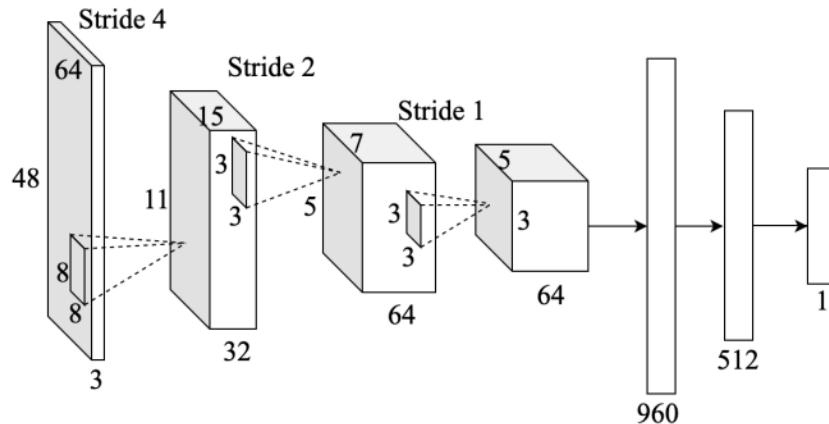


Fig. 3.3: Structure of the network used in the method of Okada and others. (Quoted from [22])

3.2 トポロジカルマップとシナリオ

ここでは、分岐路で適切な進行方向を提示する機能のベースとなる、島田ら [26] が提案したトポロジカルマップの形式とシナリオについて述べる。

3.2.1 トポロジカルマップ

トポロジカルマップは、環境中のランドマークなどの特徴的な箇所（ノード）とその繋がり（エッジ）によって環境を表現したマップである。島田らは Fig. 3.4 に示すようなトポロジカルマップの形式 [26] を提案している。島田らのトポロジカルマップでは、ノードは通路の特徴的な箇所に配置され、エッジはそれらのノードを接続するように配置される。ノードには ID、通路の特徴 (Type), エッジの ID と相対角度 (Edge) のデータ、エッジには ID のデータのみが含まれている。このトポロジカルマップの形式は、人の道案内に関するアンケートを実施し、その結果に基づいて決定された。アンケートでは、人は道案内において「通路の特徴」や「向いている方向」を重視していることが明らかになっている。

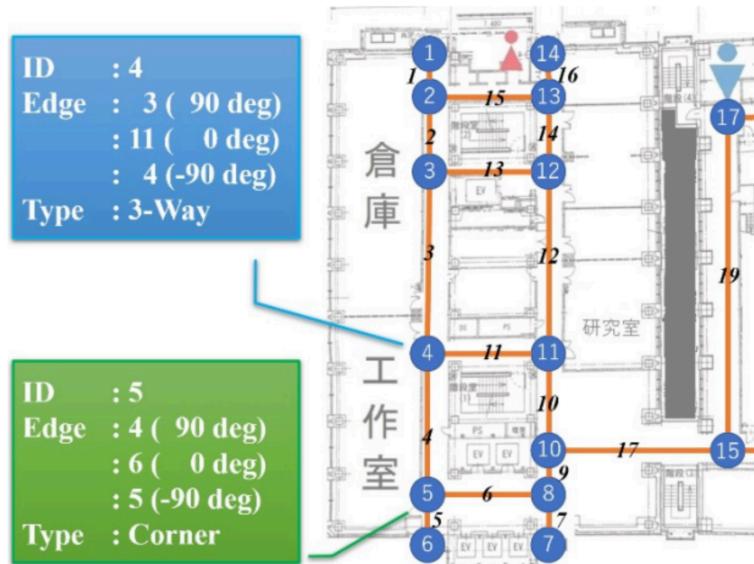


Fig. 3.4: Topological map format proposed by Shimada and others (Quoted from[26])

3.2.2 シナリオ

シナリオはトポジカルマップ上での、目的地までの経路を単語の組み合わせで表現したものである。具体的には、「次の角」や「突き当たりまで」のような「条件」と「直進」、「右折」のような「行動」の組み合わせにより作成される。このシナリオの形式は、3.2節のトポジカルマップと同様に人の道案内に関するアンケートを実施し、その結果に基づいて決定された。アンケートでは、人は「条件」と「行動」を組み合わせて道案内をしていることが明らかになっている。例として、Fig. 3.5 に示す経路をシナリオで表現すると、「3番目の三叉路まで直進。停止。」となる。

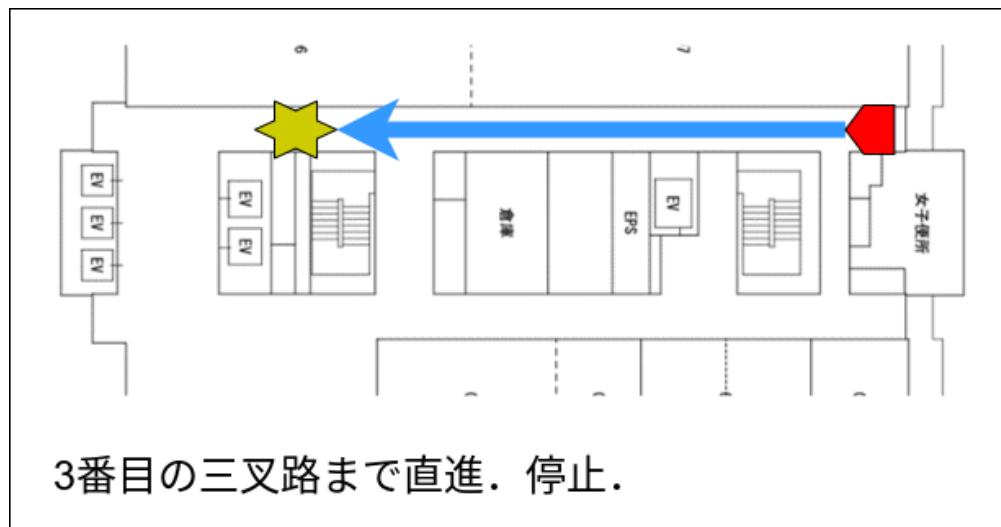


Fig. 3.5: Example scenarios (Quoted from [25])

第4章

動的に経路を選択して移動する機能 の追加

4.1 概要

本章では、岡田らの従来手法に対して、目標とする進行方向のデータを加えることで、動的に経路を選択して移動する機能の追加を試みる。これにより、視覚に基づくナビゲーションにおいて、分岐路で「直進」「左折」などの任意の経路への移動が可能となることを目指す。以後、この動的に経路を選択して移動する機能を「経路選択機能」と呼ぶ。

4.2 経路選択機能を追加したシステム

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のデータ（以後、目標方向と呼ぶ）を新たに追加した。Fig. 4.1 に経路選択機能を追加したシステムを示す。なお、上記の追加した要素を除き、他の部分は岡田らの従来手法と同様である。学習時は、カメラ画像とメトリックマップに基づくルールベースの制御器から出力される目標方向を入力としてデータセットへ加える。岡田らの従来手法では、データセットの収集法に関して、いくつかバリエーションがある。本論文では、その中で、最も経路追従の成功率が高い、学習器の出力とルールベース制御器の出力を比較し、差の絶対値が閾値 (0.05rad/s) を超えた場合のみ訓練データへ加える方法 [21] を用いる。学習後は、Fig. 4.2 に示すように学習器へ外部から目標方向を入力する。ロボットは視覚に基づくナビゲーションにおいて、入力された

目標方向に従った経路を選択して移動する。

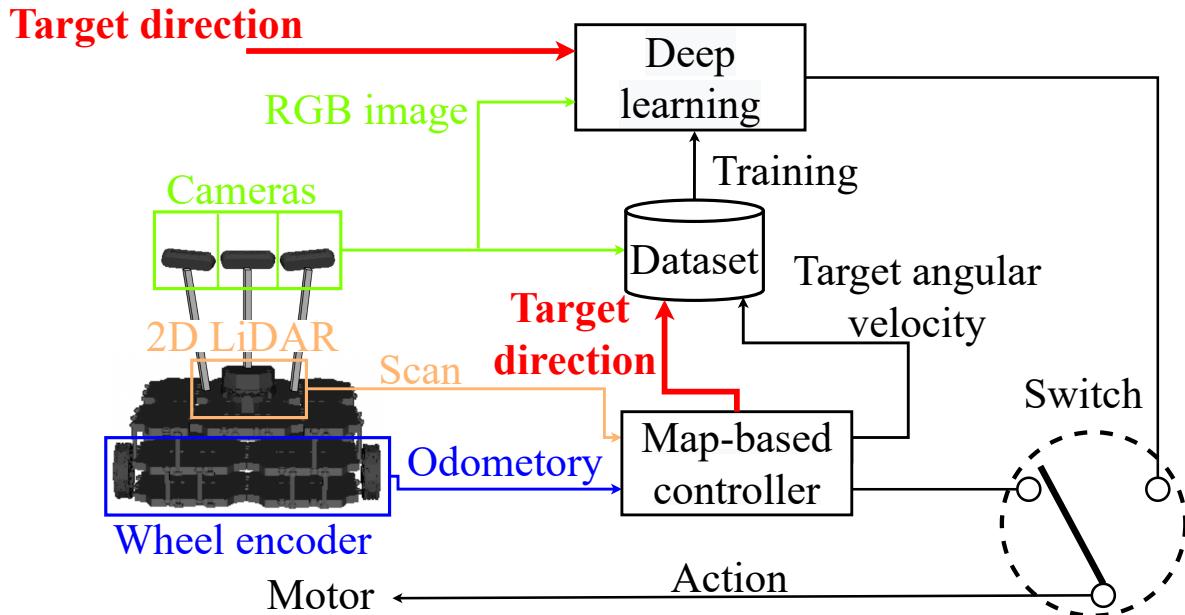


Fig. 4.1: System of the imitation learning with added function of selecting path

Target direction given externally

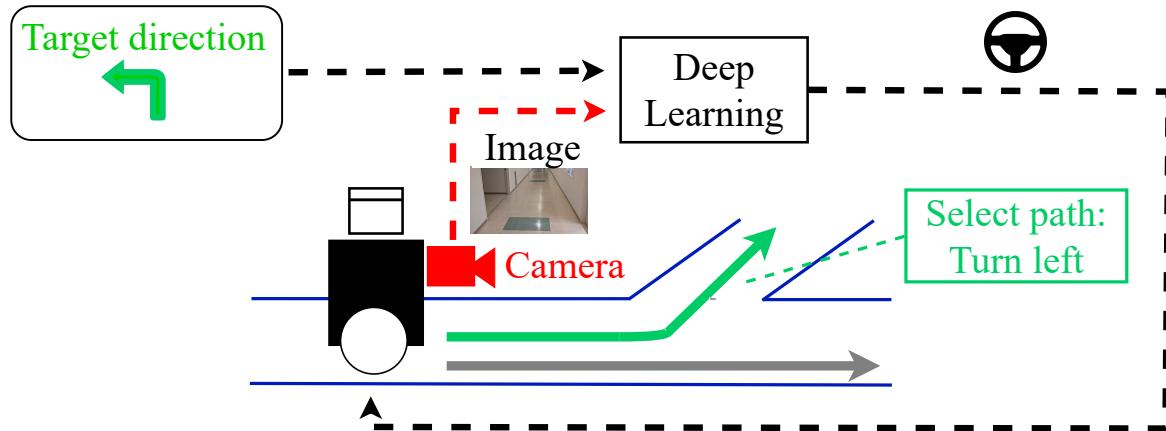


Fig. 4.2: Path selection and following behavior based on camera images and target direction by imitation learning (Quoted from [25])

ネットワークの構造と目標方向のデータ形式

経路選択機能を追加した学習器のネットワークの構造を Fig. 4.3 に示す。岡田らの従来手法で用いたネットワークの出力部に、目標方向の入力部を追加した。この層に Table 4.1 に示すワンホットベクトルのデータを入力する。ネットワークは、画像を処理する CNN アーキテクチャ、CNN の出力と目標方向を入力とする全結合層で構成されている。損失関数や活性化関数などのパラメータは岡田らの従来手法と同様である。

学習器は RGB 画像と、目標方向を入力、ヨー方向の角速度を出力として end-to-end で学習する。つまり、視覚を入力とした行動を、目標方向によって条件付けながら学習する。

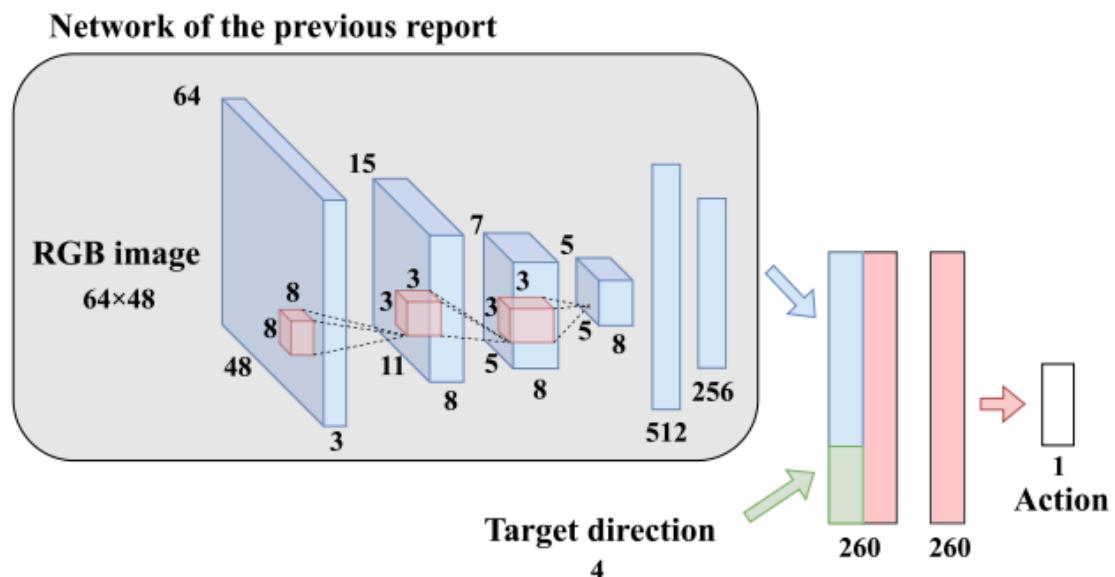


Fig. 4.3: Structure of network with added function of selecting path

Table 4.1: Target direction and data for imitation learning

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

4.3 シミュレータを用いた経路選択機能を確認する実験

経路選択機能を追加したシステムで、指定した進行方向にロボットが移動できることをシミュレータを用いた実験により確認する。

実験装置

実験はシミュレータ上で行い、シミュレータには Gazebo[1] を使用し、ロボットには Fig. 4.4 に示す TurtleBot3 Waffle Pi[4] へ 3 つのカメラを追加したモデルを用いる。実験環境には、Fig. 4.5 に示す千葉工業大学津田沼キャンパス 2 号館 3 階を模したモデルを用いた。環境中の A, B の地点においては、Fig. 4.6 に示すように侵入する方向が 3 つ、脱出する経路はそれぞれ 2 つあるため、バリエーションは 6 つある。つまり、目標方向に従って適切に経路を選択して移動することが求められる場所となる

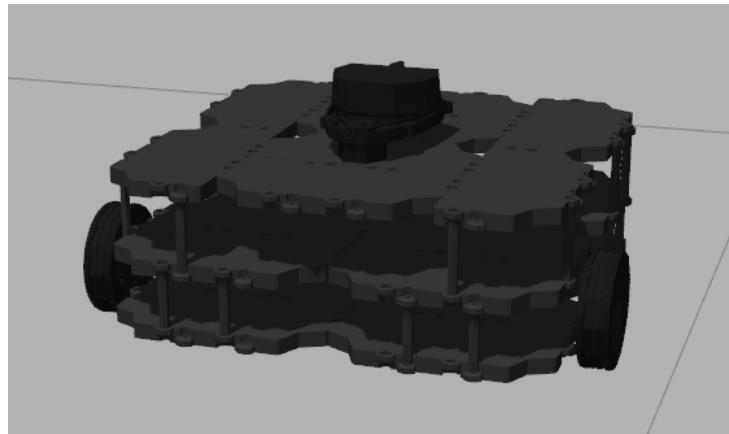


Fig. 4.4: TurtleBot3 Waffle Pi

実験方法

実験は、Fig. 4.7 に示した経路を a から f の順番で繰り返し走行しながら、模倣学習を行う。目標方向のデータはメトリックマップに基づくルールベース制御器から出力され、データセットに加えられる。学習は 60000 ステップ実行し、その後視覚に基づくナビゲーションへ移行する。視覚に基づくナビゲーションにおいても a から f までの経路をロボットに走行させる。ロボットが壁に衝突した場合は、ロボットを経路の中央まで戻して実験を継続する。この実験を 10 回実施する。

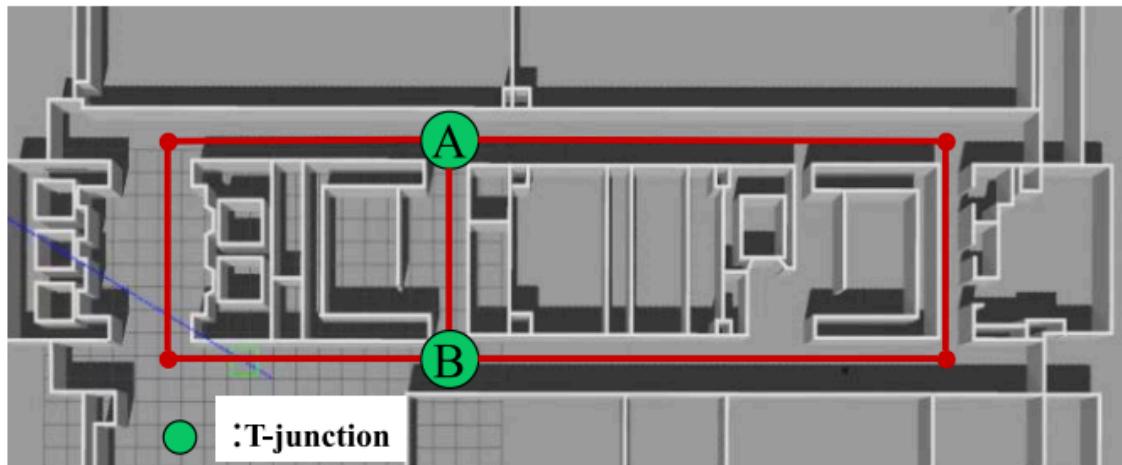
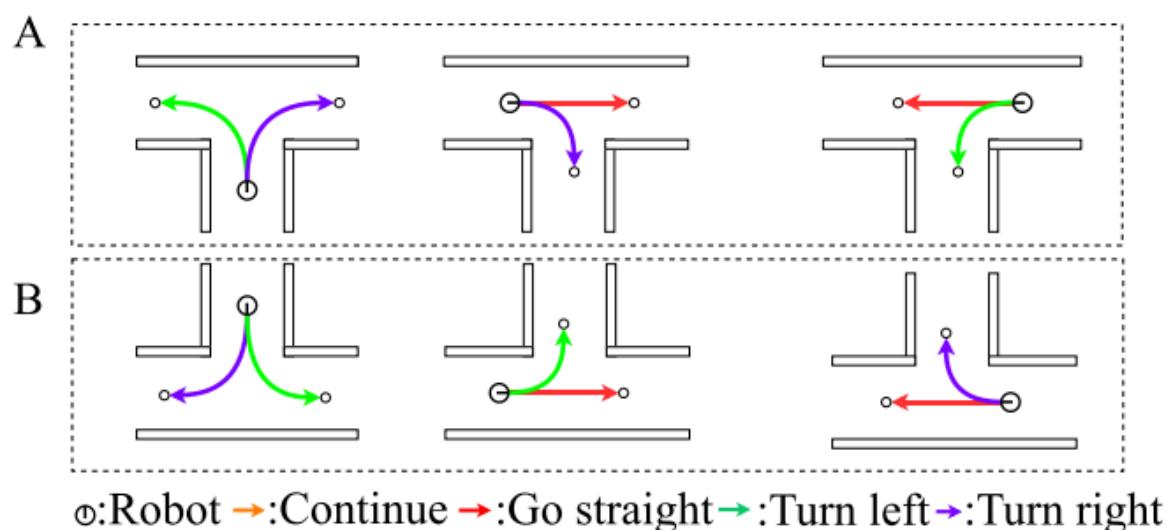
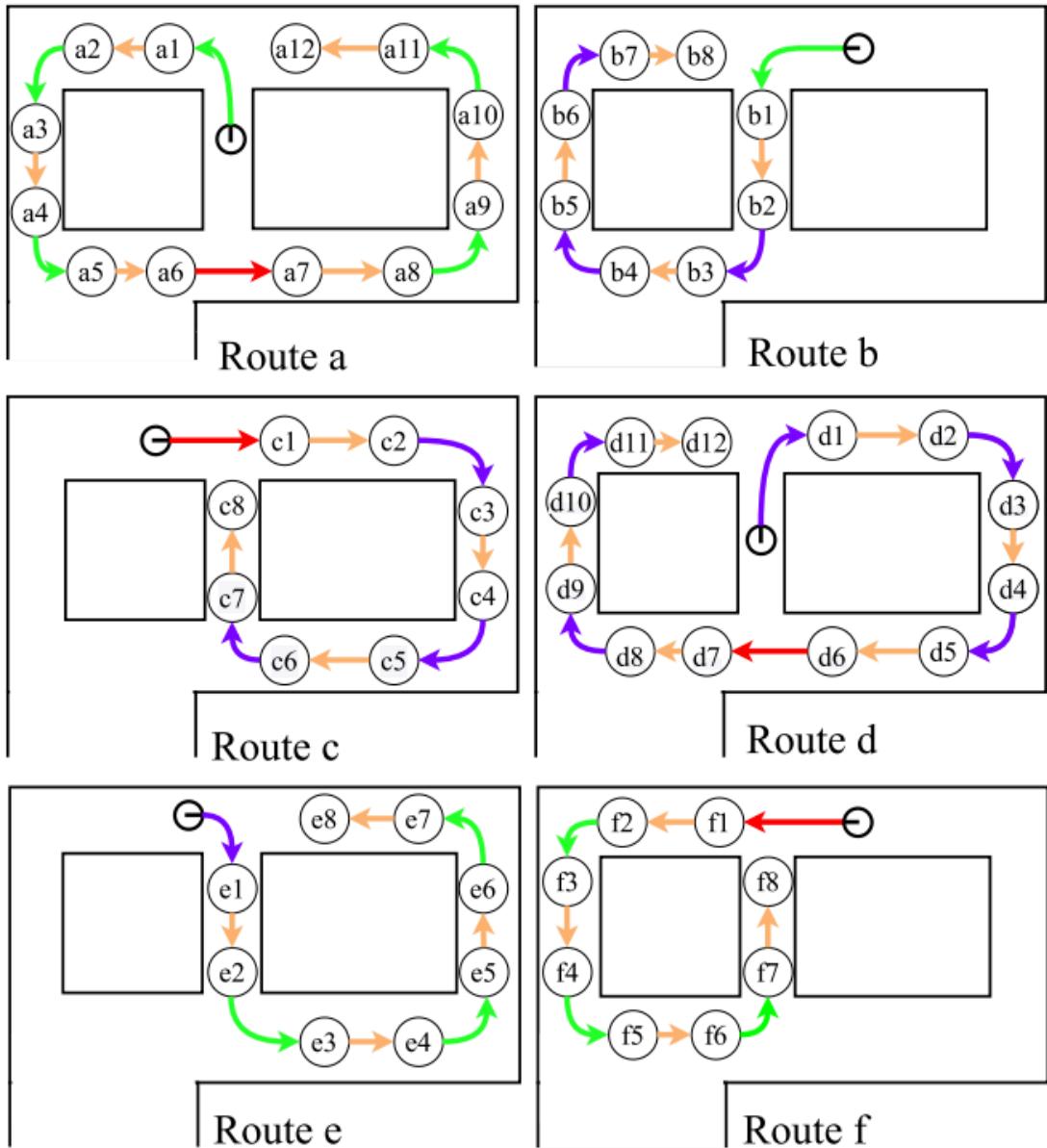


Fig. 4.5: Experimental environment of selecting path (Quoted from [24])



○:Robot →:Continue →:Go straight→:Turn left→:Turn right

Fig. 4.6: Selecting a path at the T-junction (Quoted from [24])



⦿:Robot →:Continue →:Go straight
○:Goal →:Turn left →:Turn right

Fig. 4.7: Route for experiment of selecting path (Quoted from [24])

実験結果

実験結果を Fig. 4.8 に示す。図は A, B それぞれの分岐路で、ロボットが正しく経路を選択した回数を示している。図に示すように、目標方向に従い 113/120 回適切に経路を選択する様子が確認できた。ただし、Table 4.2 に示した箇所では、壁に衝突するなどして自律移動が継続できない様子も見られた。

Fig. 4.9 に A 地点で目標方向のコマンドに従い、ロボットが経路を選択して走行する様子を示す。このように、同一の分岐路であっても目標方向の入力に従い、ロボットが適切に経路を選択して走行する様子が見られた。この結果から、岡田らの従来手法に対し、任意の目的地に向けた移動に必要な、動的に経路を選択した移動する機能を追加できたと考えられる、

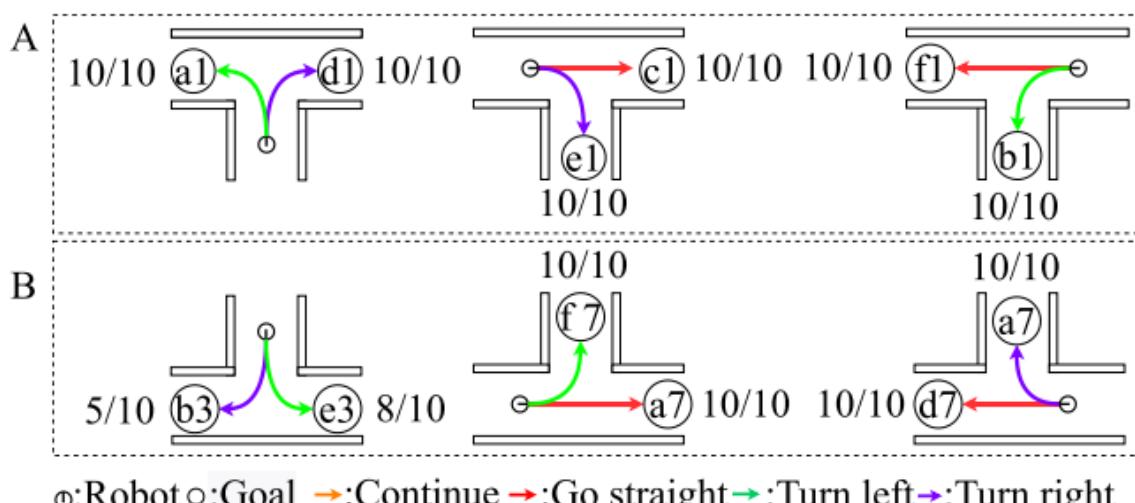


Fig. 4.8: Number of times a correct path was selected (Quoted from [24])

Table 4.2: Location where the robot went off course

Location	Number of Failed
a5	2/10
b3	5/10
d11	1/10
e1	2/10
f5	2/10

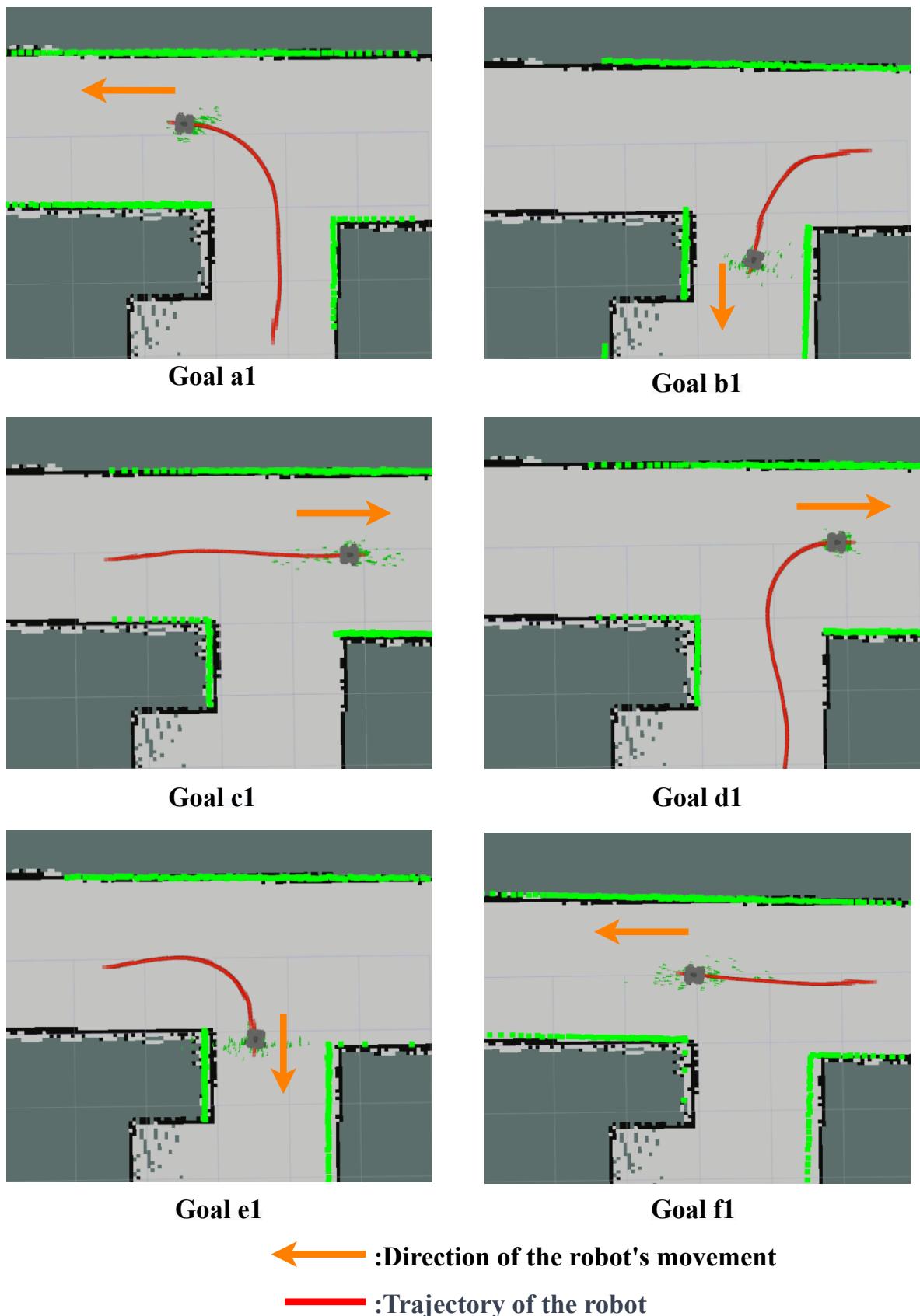


Fig. 4.9: The robot moving while selecting a path (Quoted from [24])

第5章

視覚に基づいて目的地まで自律移動するシステムの構築

第4章では岡田らの従来手法に対し、経路選択機能を追加した。これにより視覚に基づくナビゲーションにおいて、分岐路で目標方向の入力に従った経路を選択して移動できることを確認した。本章では、視覚に基づいて通路の特徴を検出し、目標方向を提示する機能を追加する。目標方向の提示には島田ら [26] が提案したトポロジカルマップとシナリオを用いる。これにより、カメラ画像とトポロジカルマップから作成されるシナリオに基づいて、目的地まで自律移動するシステムを構築する。

5.1 構築したシステムの概要

構築したシステムでは、人間が作成した「次の角まで直進。左折。」などのシナリオから指示された道順に従い、カメラ画像に基づいて目的地まで自律移動する。Fig. 5.1 に構築したシステムの概要と自律移動の流れを示す。構築したシステムは

- 1) シナリオを分解し、「条件」と「行動」を抽出するモジュール（以後、シナリオモジュールと呼ぶ）
- 2) カメラ画像と目標方向に基づいて、経路を追従するモジュール（以後、経路追従モジュールと呼ぶ）
- 3) カメラ画像を基に通路の特徴を分類するモジュール（以後、通路分類モジュールと呼ぶ）

3つのモジュールで構成されている。ロボットは a から d の一連の流れにより指示された道順に従って目的地まで自律移動する。

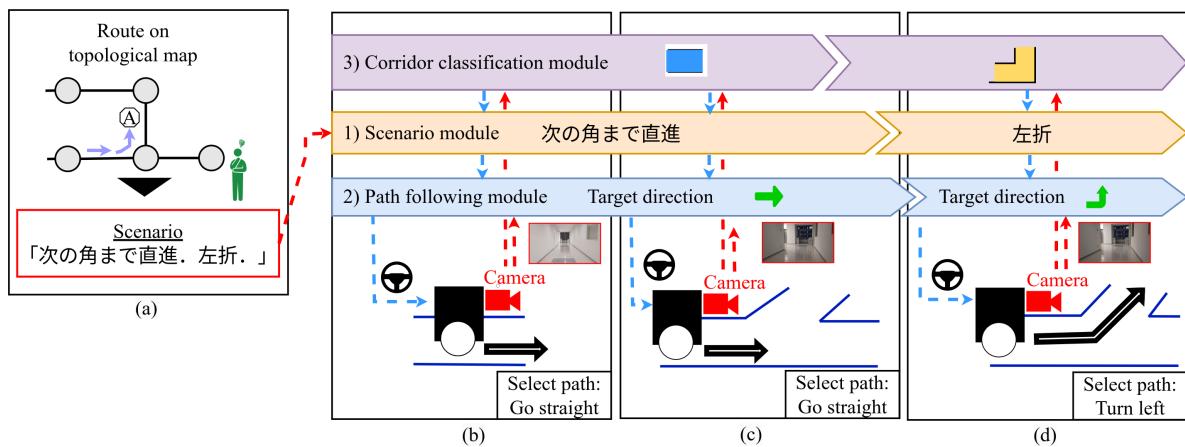


Fig. 5.1: Overview of constructed system (Quoted from [25])

- (a) トポロジカルマップ上の目的地に応じて、人間が「条件」と「行動」で構成されるシナリオを作成する。例えば、図のトポロジカルマップ上で A を目的地とするシナリオは「次の角まで直進. 左折.」となる。
- (b) 作成したシナリオをシナリオモジュールへ入力する。シナリオモジュールは入力されたシナリオを分解し、「条件」と「行動」を抽出する。1つ目の条件と行動のセットは「次の角まで」と「直進」となる。この「直進」を目標方向として経路追従モジュールへ与える。経路追従モジュールは、カメラ画像と与えられた目標方向に基づいて、経路に沿って直進する。
- (c) ロボットが角に近づくと、通路分類モジュールがカメラ画像に基づいて通路を「角」と分類して、それをシナリオモジュールに与える。シナリオモジュールはそれを基に「次の角まで」という条件を満たしたかを判定する。この場合は条件を満たしているため、2つ目の行動である「左折」へ遷移する。
- (d) 「左折」に基づいて、経路追従モジュールは経路に沿って角を左折する。

5.2 経路追従モジュール

経路追従モジュールについて述べる。このモジュールは4章で述べたシステムに倣って構築した、カメラ画像に基づいて経路を追従するモジュールであり、分岐路では入力された目標方向によって経路を選択して走行する。なお、4章で述べたシステムに対し、目標方向とネットワークのパラメータ変更、データセットの収集方法の追加をしている。

追加した学習時のデータセットの収集方法は、藤原ら [27] が学習量の削減を目的に行った以下の2つの手法である。

- データセットに加えるデータの不均衡を改善する手法
- 学習時に積極的に蛇行する手法

データセットに加えるデータの不均衡を改善する手法について述べる。Fig. 5.2 に、第4章の実験における10000ステップあたりの目標方向を青で示す。直進のデータが他に比べ圧倒的に多く、不均衡あることが分かる。Haibo らの調査 [10] では、ほとんどの標準的な学習アルゴリズムは、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されている。そこで、データ前処理手法であるオーバーサンプリングを用いて、データの偏りを改善する。具体的には、図中赤で示すように左折と右折を7倍に複製する。

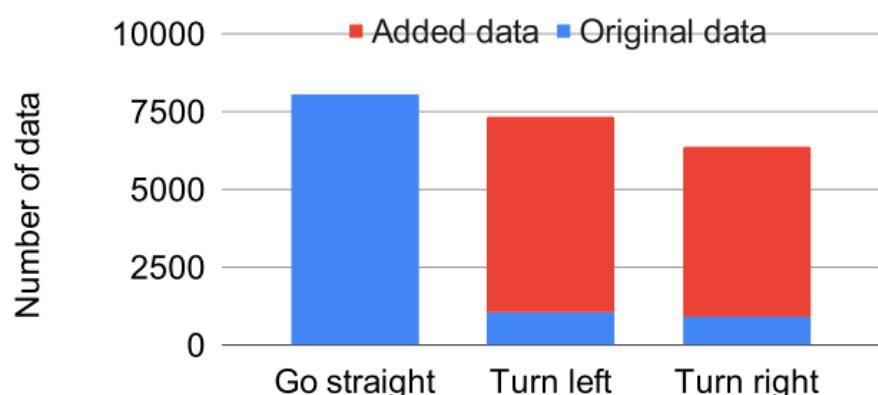


Fig. 5.2: Number of data in the target direction per 10000 steps in the previous experiment Quote from [27]

次に、学習時に積極的に蛇行する手法について述べる。学習時に経路から離れた状態が増加すると、訓練済みの学習器での走行時に経路から外れることが減少し、より少ない学習時間で成功率が高いモデルができる可能性がある。そのため、学習時に積極的に蛇行する手法では Fig. 5.3 に示すように、学習時のロボットの制御に用いるヨー方向の角速度を 1.5 倍にする。これにより、大きく蛇行して、目標経路から離れた状態を多く収集する。

藤原らは実験により、これらの 2 つのデータセットの収集手法によって、経路追従の成功率を下げる無く、必要な学習量を 60000 ステップから 20000 ステップまで削減できることを確認している。

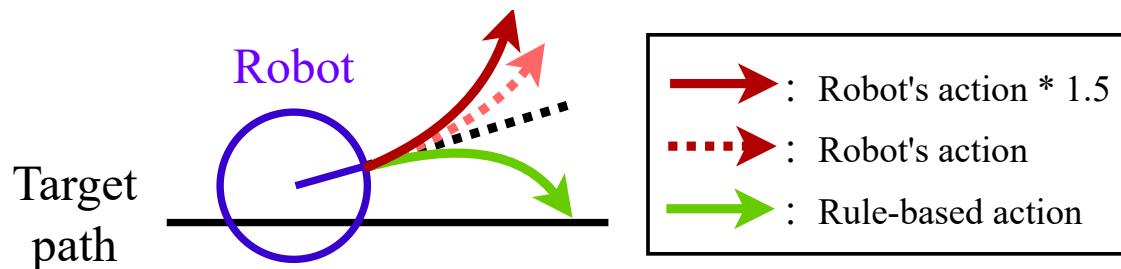


Fig. 5.3: Aggressive meandering (Quote from [27])

Table 5.1 に変更した目標方向とそのデータ形式を示す。第4章における目標方向の中で、Continue と Go straight は同じ方向を指しているため、本章ではこれらを1つとして扱う。これに伴い、ワンショットベクトルの次元数を4から3へ変更している。さらに、目的地に到着後、ロボットを停止させることを目的として、Stop(停止)の目標方向を追加した。停止の目標方向が入力された際には、ロボットの並進速度は0.0m/sとなり、ロボットはその場で停止する。

目標方向の次元数に伴い、変更を加えたネットワークの構造を Fig. 5.4 に示す。具体的には、CNN の出力と目標方向を入力する全結合層の入力サイズを 260 から 259 へ調整している。

Table 5.1: Target direction and data for path-following module

Target direction	Data
Go straight	[1, 0, 0]
Turn left	[0, 1, 0]
Turn right	[0, 0, 1]
Stop	[0, 0, 0]

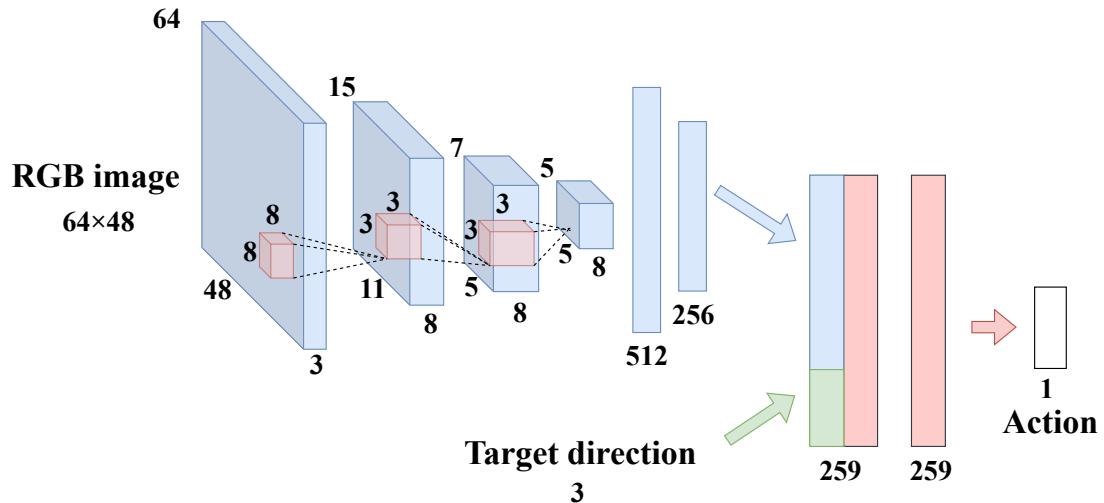


Fig. 5.4: Structure of the network of route selection module

5.3 通路分類モジュール

通路分類モジュールについて述べる。このモジュールは、シナリオの「条件」が満たされたかの判定に必要な通路の特徴を、カメラ画像を入力として分類する。通路分類モジュールの概要を Fig. 5.5 に示す。モジュールはフレーム数 16、画像サイズ 64x48 の連続した画像データを入力とし、通路の特徴を分類した結果を出力する。通路の特徴の分類は、島田らの手法に倣い、Fig. 5.6 に示す 8 つとしている。使用するカメラは、経路追従モジュールと同様に、データセットの収集時は 3 つ、学習後は 1 つである。

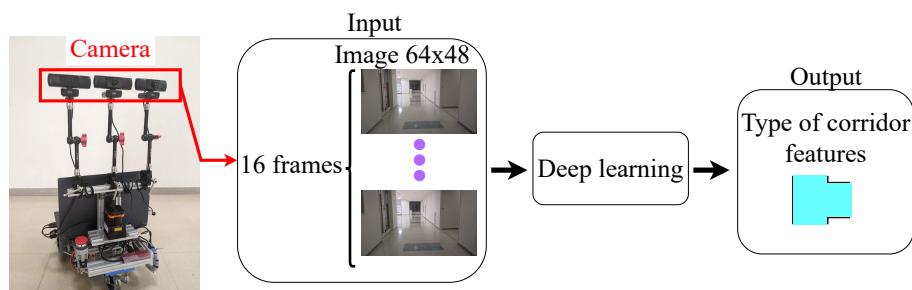


Fig. 5.5: Overview of the corridor classification module (Quoted from [25])

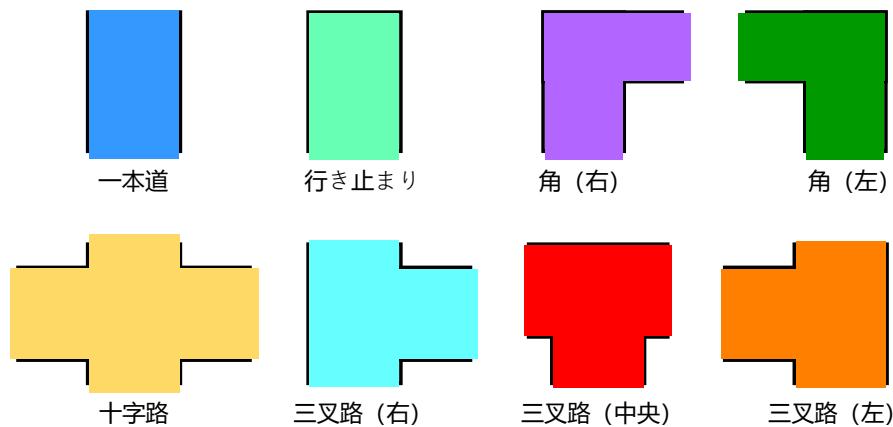


Fig. 5.6: Types of corridor features (Quoted from [25])

ネットワークの構成を Fig. 5.8 に示す。このネットワークアーキテクチャは Dhaivat ら [7] が提案する CNN と LSTM を組み合わせた IntersectNet に倣って構築した。なお CNN アーキテクチャは実時間性の観点から AlexNet[14] から MobileNetV3-Large[11] へ変更している。

ネットワークはフレーム数 16、画像サイズ 64×48 の連続した RGB 画像データを入力とする。画像データは各フレームごとに CNN で処理され、この特徴ベクトルは LSTM へ入力される。各 LSTM の出力は分類層（全結合層）へ渡される。最後に、全ての分類層の出力を融合層へ渡し、融合層は入力の平均を取ることで、最終的な分類結果を出力する。活性化関数として ReLU、最適化アルゴリズムには Adam、損失関数として CrossEntropyLoss を使用する。

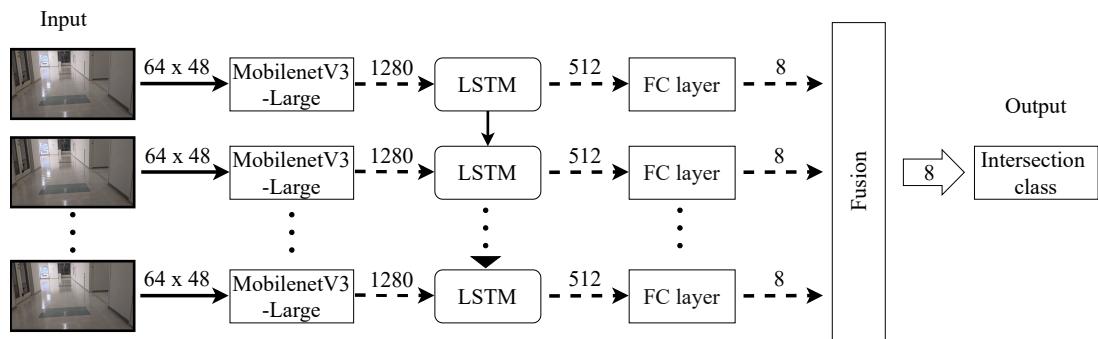


Fig. 5.7: Network of corridor classification module

次に通路分類モジュールのデータセットの作成について述べる。データの作成では、経路追従モジュールの学習と同様に、メトリックマップに基づいたルールベース制御器によって経路を走行する。その際、フレーム数16の連続したカメラ画像と通路の分類ラベルを1組とし、0.125秒周期でデータセットへ加える。分類ラベルのアノテーションは、Fig. 5.8に示すように、通路の特徴を予めメトリックマップに登録しておくことで、自動的に行う。

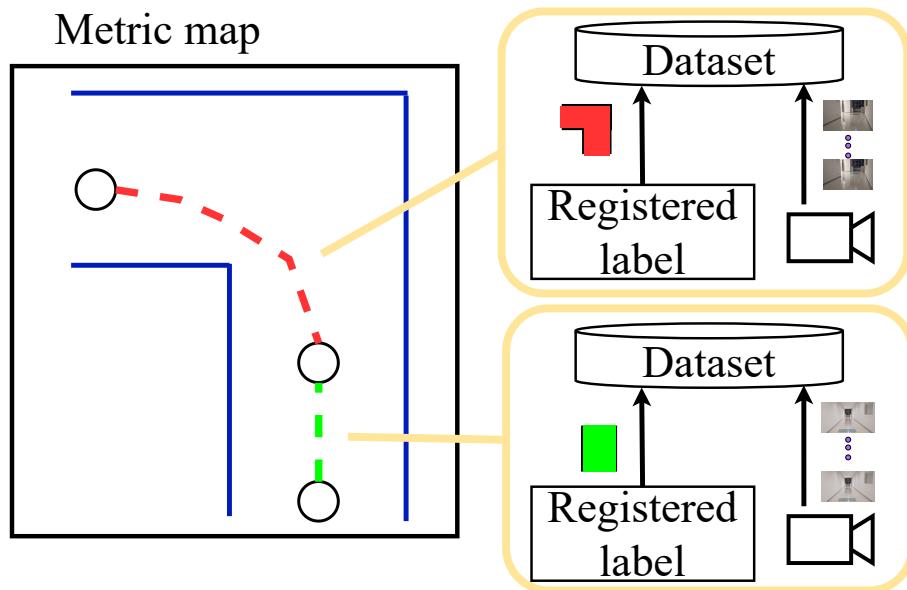


Fig. 5.8: Classification labels registered in the metric map Quoted from [25]

5.2節では不均衡データへの対策として、オーバーサンプリングを行った。この通路文類モジュールでは、不均衡データへの対策として、データセット内のクラス間のデータ数によって重み付けを行うコストアプローチ[20]を行う。具体的には、損失関数(CrossEntropyLoss)で用いるクラスごとの重みをデータ数を基に調整する。

5.4 シナリオモジュール

シナリオモジュールについて述べる。シナリオモジュールはトポロジカルマップから作成されたシナリオから「突き当りまで」という「条件」や「左折」などの「行動」を解釈し、単語で構成された経路を分岐路での目標方向へ変換して出力する。

Fig. 5.9 にトポロジカルマップとそれをもとに作成されるシナリオを示す。図の例では出発地点をエッジ 2、目的地をノード 2 として、その間のエッジとノードを移動する。エッジ 2 からノード 1 は「三叉路まで」という条件と「直進」という行動、ノード 1 からエッジ 1 は「右折」という行動、エッジ 1 からノード 2 は「突き当り（三叉路）まで」という条件と「直進」の行動で表現される。これらを統合すると、最終的に「三叉路まで直進。右折。突き当たりまで直進。停止。」のシナリオが作成される。

次に作成したシナリオを目標方向に変換する処理を述べる。はじめにシナリオを句点ごとに分解し、部分シナリオというものを作成する。この部分シナリオには、次の部分シナリオに遷移するための「条件」とロボットが行う必要がある「行動」が含まれている。この部分シナリオを形態素分析 (MeCab[23]) を用いて単語へ分割する。分割した単語は、「条件」と「行動」を示す以下の項目に分けて登録される。

- 1) 通路の特徴 例えば、「三叉路」「角」など
- 2) 順番 例えば、「3 つ目の」「2 番目の」など
- 3) 方向 例えば、「左手に」「右手に」など
- 4) 行動 例えば、「右折」「停止」など

先に示した例は句点ごとに、三叉路まで直進/ 右折/ 突き当たりまで直進/ 停止/ と分解される。

1 つ目の条件と行動は 1) 通路の特徴 三叉路、4) 行動 直進、

2 つ目の行動は 4) 行動 右折となる。

3 つ目の条件と行動は 1) 通路の特徴 突き当たり 4) 行動 直進

4 つ目の行動は 4) 停止となる。

この 4) 行動を Table 5.1 で示したデータ形式で表現し、分岐路での目標方向として、経路

追従モジュールへ与える。ここで、「三叉路まで」といった条件を達成したかの判定は、5.3節の通路分類モジュールの分類結果を用いて行う。

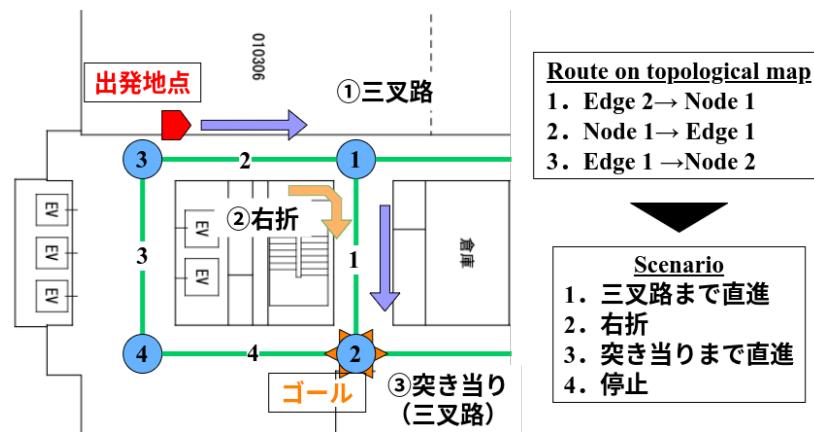


Fig. 5.9: Example of topological map and created scenario (Quoted from [25])

5.5 実ロボットを用いた実験

実ロボットを用いて、構築したシステムにより、ロボットが目的地へ到達可能であるか検証する。

5.5.1 実験装置

実験で用いるロボットを Fig. 5.10 に示す。ロボットは icart-mini[6] をベースに開発したロボットを用いる。センサとして、単眼のウェブカメラ（サンワサプライ株式会社 CMS-V43BK）を 3 つ、2D-LiDAR（北陽電機 UTM-30LX）を 1 つ、左右のモータにそれぞれパルス付きエンコーダを搭載している。制御用の PC には GALLERIA GCR2070RGF-QC-G を使用している。メトリックマップに基づくルールベース制御器には、本学で ROS Navigation stack をもとに開発した orne_navigation[2] を使用する

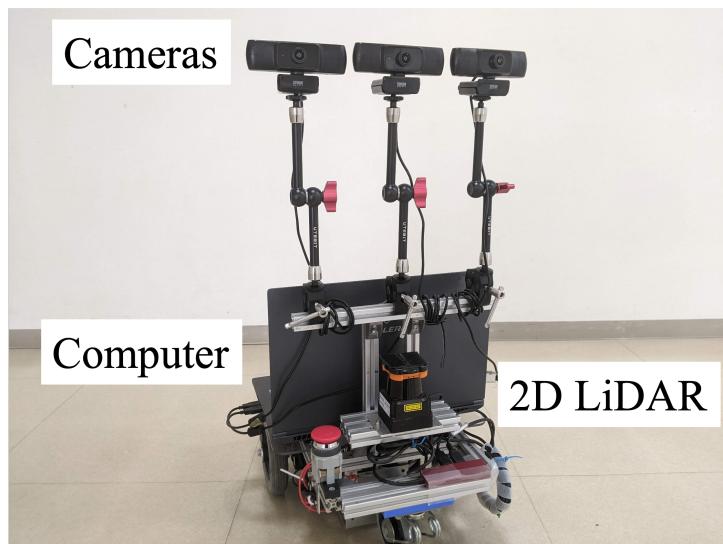


Fig. 5.10: Experimental setup (Quoted from [25])

5.5.2 実験方法

実験環境として Fig. 5.11 に示す千葉工業大学津田沼キャンパス 2 号館 3 階の廊下を用いる。環境中には、三叉路が 4 つ、角が 2 つ、突き当たりが 2 つ含まれている。経路追従モジュールの訓練および通路分類モジュールのデータセット収集では、Fig. 5.12 で示す a から n の経路を順番に走行する。

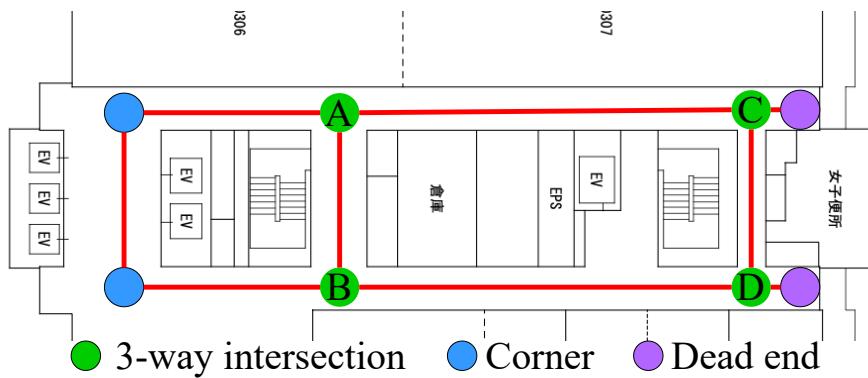


Fig. 5.11: Experimental environment (Quoted from [25])

シナリオの選定

実験では島田らが用いた 50 例のシナリオの中から、Fig. 5.13 に示す 7 例を用いた。このシナリオは以下の 3 つの基準に従って、選定した。

- 1) 2 号館 3 階の廊下において、Fig. 5.11 の場所を対象としていること。
- 2) ロボットが移動困難な Fig. 5.14 の箇所のような狭い通路が含まれていないこと
- 3) その場で「後ろを向く」など経路追従モジュールができない行動が含まれていないこと

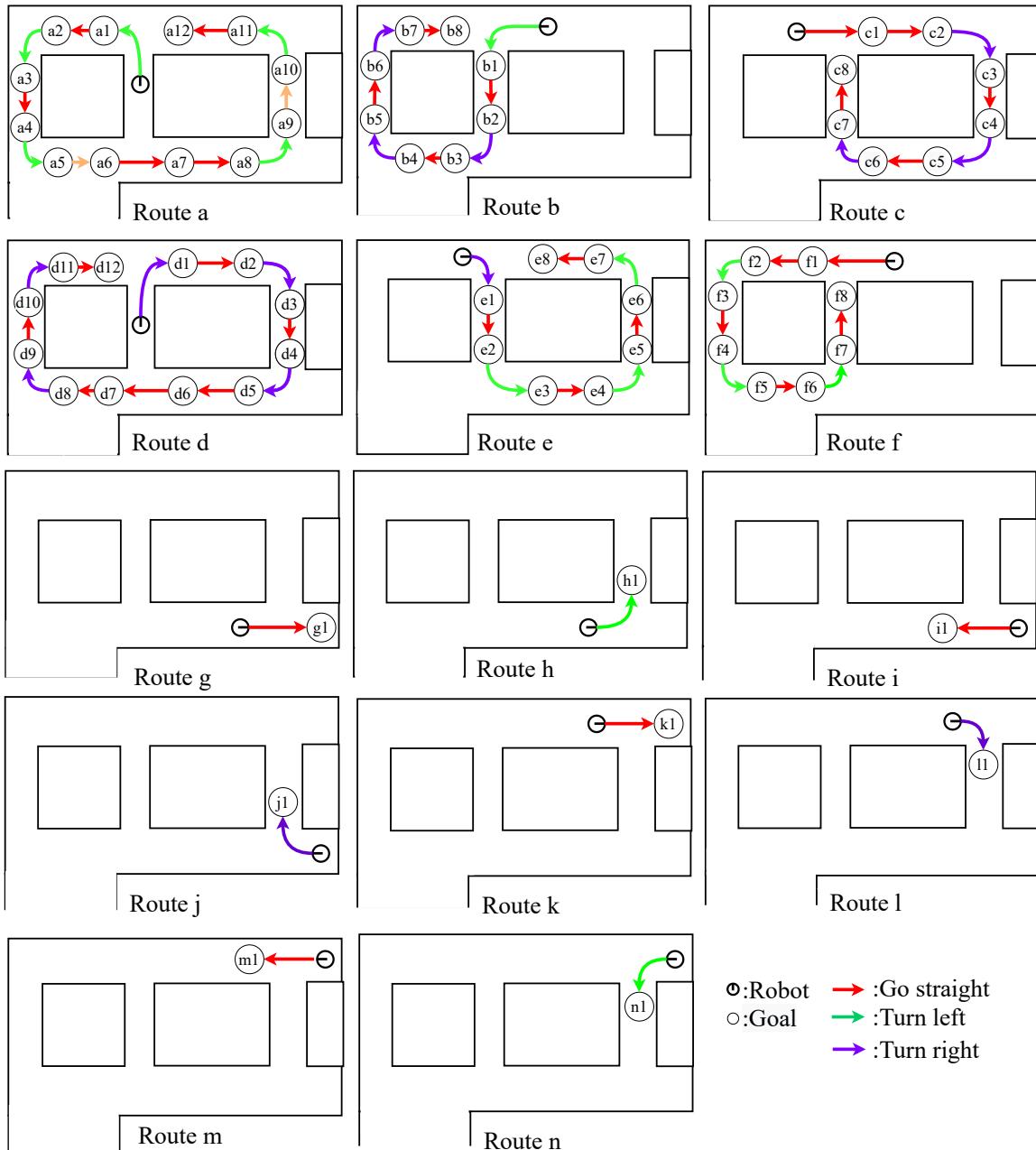


Fig. 5.12: Route used for learning (Quoted from [25])

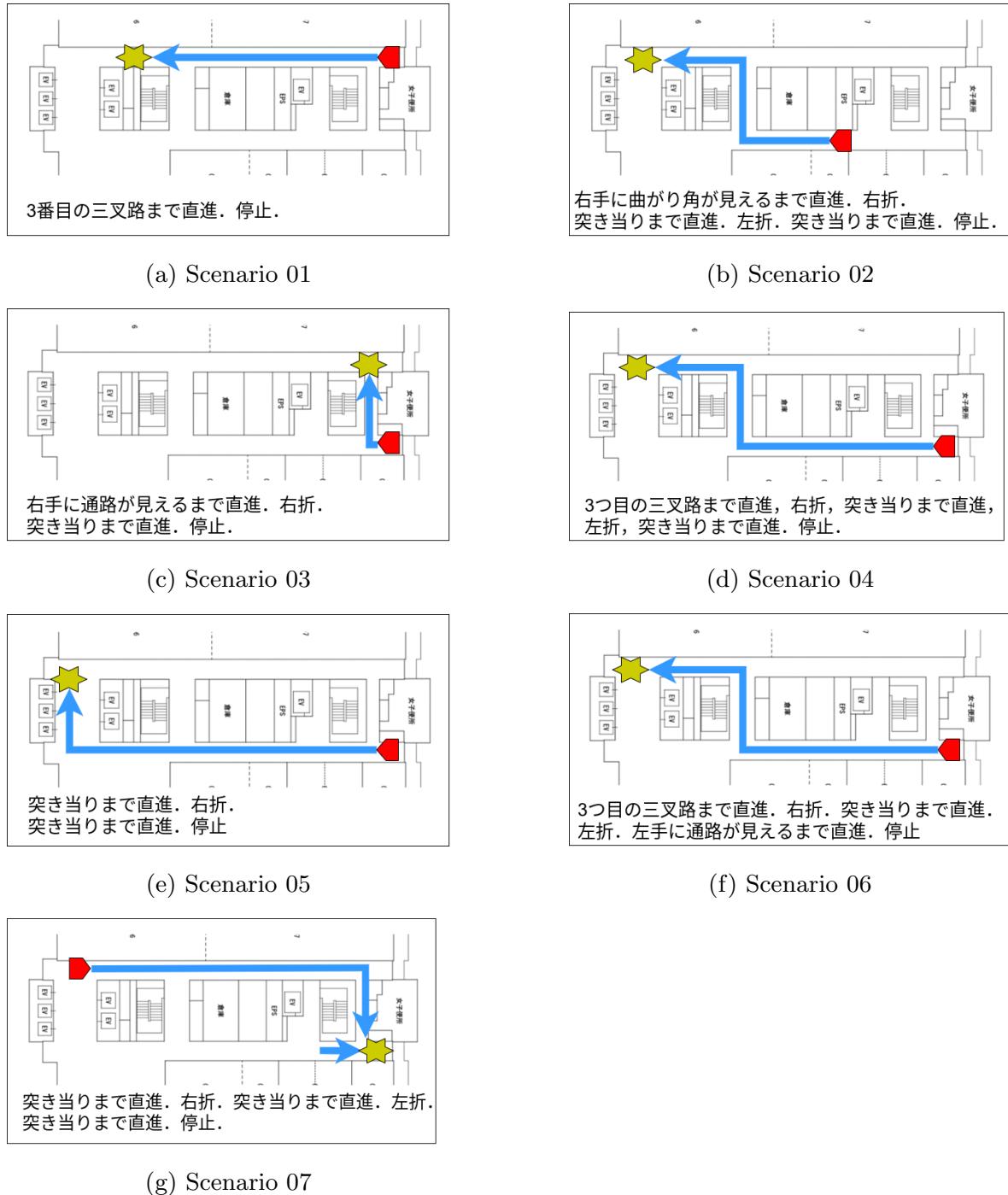


Fig. 5.13: Scenarios used in the experiment (Quoted from [25])

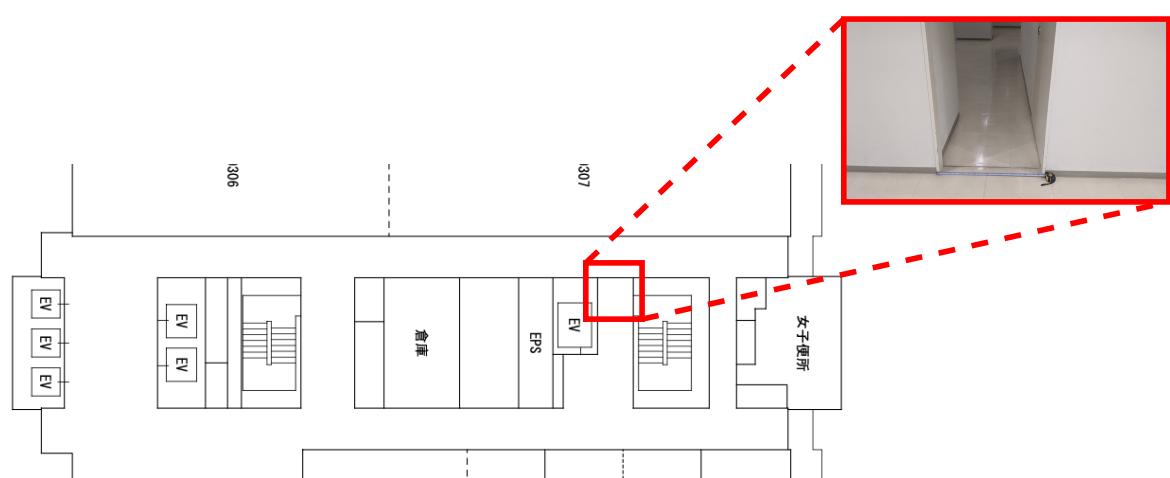


Fig. 5.14: Example of narrow passage in experimental environment

通路追従モジュールの訓練

まずははじめに通路分類モジュールの訓練を行う。前述の経路をメトリックマップに基づいたルールベース制御器の出力を用いて、3周し、データセットを収集する。収集したデータは、1, 2周目を訓練データとし、3周目をテストデータとする。それぞれのデータ数は、訓練データ 5781, テストデータ 2902 である。2つのデータ内のクラス間のデータ数は Table 5.2 である。訓練は損失関数の重みを Table 5.3, バッチサイズを 32 として、30epoch 行った。

訓練後、テストデータに対する通路分類モジュールの正解率 (Accuracy), 適合率 (Precision) を計算する。

Table 5.2: Number of classes in the dataset

Class	Learning data	Test data
直進	2760	1377
突き当たり	90	45
角 (右)	549	279
角 (左)	567	288
十字路	0	0
三叉路 (右)	753	387
三叉路 (中央)	306	153
三叉路 (左)	747	369

経路追従モジュールの訓練

次に経路追従モジュールの訓練を行う。通路分類モジュールの訓練と同様の経路を、オンラインで模倣学習しながら 1周走行する。その際のステップ数は 12000 であった。

2つのモジュールを訓練後、シナリオを 1 例ずつ入力して、ロボットの挙動を観察する。実験では、ロボットをシナリオのスタート地点に移動して、自律移動を開始する。

Table 5.3: The weights assigned to each class in the experiment

Class	Class weights
直進	1
突き当たり	5
角 (右)	5
角 (左)	5
十字路	1
三叉路 (右)	5
三叉路 (中央)	10
三叉路 (左)	5

5.5.3 実験結果

テストデータに対する、通路分類モジュールの正解率と適合率を Table 5.4 に示す。この計算にはモデル評価用ライブラリである torcheval[3] を用いた。

Table 5.4: Evaluation results of the corridor classification module on test data.

Metrics	Value
Accuracy	0.98
Precision	0.98

Fig. 5.15 に Fig. 5.13d のシナリオを入力した実験の様子を示す。キャプションは実行中の部分シナリオと、ロボットが現在位置する通路の特徴である。図に示すように入力されたシナリオの道順に従い、三叉路などの分岐路で適切に経路を選択して自律移動する様子が見られた。結果として、7例すべてでロボットが、目的地へ到達した。以上の結果から、構築したカメラ画像とシナリオに基づいて、経路を追従して目的地まで自律移動するシステムの有効性が確認された。



(a) 3つ目の三叉路まで直進 (First 3-way)



(b) 3つ目の三叉路まで直進 (Second 3-way)



(c) 右折 (Third 3-way)



(d) 突き当たりまで直進 (Straight road)



(e) 左折 (End)



(f) 突き当たりまで直進 (Straight road)



(g) 停止 (End)

Fig. 5.15: An example of the robot applied the constructed system (Quoted from [25])

第6章

おわりに

6.1 結論

本論文では、岡田らの従来手法に対し、動的に経路を選択して走行する機能や視覚に基づいて通路の特徴を検出し、分岐路において目的地に向けた進行方向を提示する機能を追加することで、走行する経路を一定の経路から、設定した任意の目的地に向けた経路へ拡張した。

はじめに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方向を加えた。そして、追加した機能の有効性をシミュレータを用いた実験により検証した。実験では、視覚に基づくナビゲーションにおいて、同一の分岐路であっても目標方向の入力に従い、ロボットが適切に経路を選択して移動する様子が見られた。

次に、視覚から通路の特徴を検出し、分岐路で目標方向を提示する機能を追加した。目標方向を提示する機能には、島田らが提案したトポジカルマップと「条件」や「行動」による経路の表現（シナリオ）を用いた。これにより、視覚に基づいて任意の目的地まで移動するシステムを構築した。そして、実ロボットを用いた実験を行い、構築したシステムにより、カメラ画像とシナリオに基づいて、経路を追従し、ロボットが目的地へ到達できることを確認した。

参考文献

- [1] Gazebo. <https://gazebosim.org/home>. (Accessed on 12/20/2023).
- [2] open-rdc/orne_navigation. https://github.com/open-rdc/orne_navigation.git. (Accessed on 12/20/2023).
- [3] pytorch/torcheval. <https://github.com/pytorch/torcheval.git>. (Accessed on 12/20/2023).
- [4] Robotis-git/turtlebot3: Ros packages for turtlebot3. <https://github.com/ROBOTIS-GIT/turtlebot3>. (Accessed on 12/20/2023).
- [5] ros-planning/navigation: Ros navigation stack. code for finding where the robot is and how it can get somewhere else. <https://github.com/ros-planning/navigation>. (Accessed on 12/20/2023).
- [6] T-frog project/icart_mini. https://t-frog.com/products/icart_mini/. (Accessed on 12/20/2023).
- [7] Dhaivat Bhatt, Danish Sodhi, Arghya Pal, Vineeth Balasubramanian, and Madhava Krishna. Have i reached the intersection: A deep learning-based approach for intersection detection from monocular cameras. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4495–4500, 2017.
- [8] Felipe Codevilla, Matthias Müller, Antonio López, Vladlen Koltun, and Alexey Dosovitskiy. End-to-end driving via conditional imitation learning, 2018.
- [9] Yoshitaka Hara, Tetsuo Tomizawa, Hisashi Date, Yoji Kuroda, and Takashi Tsubouchi. Tsukuba challenge 2019: Task settings and experimental results. *Journal of Robotics and Mechatronics*, Vol. 32, No. 6, pp. 1104–1111, 2020.
- [10] Haibo He and Edwardo A. Garcia. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, Vol. 26, No. 9, pp. 2239–2251, 2014.

- tions on Knowledge and Data Engineering*, Vol. 21, No. 9, pp. 1263–1284, 2009.
- [11] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, Quoc V. Le, and Hartwig Adam. Searching for mobilenetv3, 2019.
 - [12] Alex Kendall, Jeffrey Hawke, David Janz, Przemyslaw Mazur, Daniele Reda, John-Mark Allen, Vinh-Dieu Lam, Alex Bewley, and Amar Shah. Learning to drive in a day, 2018.
 - [13] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.
 - [14] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, Vol. 25. Curran Associates, Inc., 2012.
 - [15] Ryusuke Miyamoto, Yuta Nakamura, Miho Adachi, Takeshi Nakajima, Hiroki Ishida, Kazuya Kojima, Risako Aoki, Takuro Oki, and Shingo Kobayashi. Vision-based road-following using results of semantic segmentation for autonomous navigation. In *2019 IEEE 9th International Conference on Consumer Electronics (ICCE-Berlin)*, pp. 174–179, 2019.
 - [16] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML 2010*, pp. 807–814, 2010.
 - [17] Axel Sauer, Nikolay Savinov, and Andreas Geiger. Conditional affordance learning for driving in urban environments, 2018.
 - [18] Shunya Seiya, Alexander Carballo, Eijiro Takeuchi, Chiyomi Miyajima, and Kazuya Takeda. End-to-end navigation with branch turning support using convolutional neural network. In *2018 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 499–506, 2018.
 - [19] Dhruv Shah, Blazej Osinski, Brian Ichter, and Sergey Levine. Lm-nav: Robotic navigation with large pre-trained models of language, vision, and action, 2022.
 - [20] Jie Xu, Yuxiang Li, Fanjun Meng, Dashun Zhang, Yalan Ye, and Li Lu. Fault

- diagnosis on imbalanced data using an adaptive cost-sensitive multiscale attention network. In *2021 International Conference on Intelligent Technology and Embedded Systems (ICITES)*, pp. 77–82, 2021.
- [21] 岡田眞也, 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.
- [22] 岡田眞也, 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.
- [23] 拓工藤, 薫子山本, 裕治松本. Conditional random fields を用いた日本語形態素解析. 2004.
- [24] 春山健太, 藤原柾, 清岡優祐, 岡田眞也, 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 一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, pp. 2P2-L05, 2022.
- [25] 春山健太, 藤原柾, 馬場琉生, 石黒巧, 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 -トポロジカルマップとシナリオに基づく経路選択機能の追加と検討-. 計測自動制御学会 SI 部門講演会 SICE2023 予稿集, pp. 1B4–03, 2023.
- [26] 島田滉己, 上田隆一, 林原靖男. トポロジカルマップを用いたシナリオによるナビゲーションの提案 一シナリオに基づく実ロボットのナビゲーション-. 計測自動制御学会 SI 部門講演会 SICE2020 予稿集, pp. 1H2–04, 2020.
- [27] 藤原柾, 春山健太, 馬場琉生, 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 一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-. 日本機械学会ロボティクス・メカトロニクス講演会'23 予稿集, pp. 2P2-G06, 2023.

謝辞

本研究を進めるにあたり、熱心にご指導を頂いた林原靖男教授と上田隆一准教授に深く感謝いたします。屋外自律移動ミーティングや研究相談で厳しくご指導いただきました。これにより、研究や技術的、人間的に成長できました。また、ご指導のおかげで2回の学会発表を行うことができました。（この修論の有様のように、文章を書く力は疑問が残りますが）

つくばチャレンジを通して実世界でロボットを動かす経験を積むことができました。この活動を通して、目的に必要な技術をリサーチし、それに基づいて実装する力と、挑戦を恐れず「手を動かして考える」といった物事に取り組む姿勢を得ることができました。また、学部3年で配属されてからの、計4年もの間、研究室で活動しました。その中で、LiDARとIMUを信じられない破壊をしたことについて大変申し訳なく思っています。

研究面での議論やサポート、私生活などで岡田真也先輩、清岡優祐先輩には返しきれぬおんがあります。

また、研究室の同期、後輩（特に屋外自律三人衆）には日々の生活の中で、研究に関する議論や、息抜きをしていただたことに感謝します。そして、学部生の頃より、精神的に辛い時の支えと生活に潤いをくださった彼女へ感謝いたします。最後に、大学院を卒業するまでの24年間、私を育てていただいた両親に謝意を表します。