

# 卒業論文

視覚と行動の end-to-end 学習により  
経路追従行動を模倣する手法の提案

-経路選択の成功率向上を意図したネットワークの変更と実験的評価-

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

-Modification of network and experimental evaluation aimed at improving  
route selection success rate-

2025 年 1 月 16 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

21C1011 石黒 巧



# 概要

視覚と行動の end-to-end 学習により  
経路追従行動を模倣する手法の提案

-経路選択の成功率向上を意図したネットワークの変更と実験的評価-

本研究室では、移動ロボットのナビゲーション手段を冗長化するために、いくつかの手法を提案している。岡田らはメトリックマップに基づくナビゲーションから生成される経路追従行動を end-to-end 学習を用いて、視覚を入力とする行動にオンラインで模倣する手法を提案してきた。春山らは、岡田らの手法に加えて、カメラ画像から通路の種類を分類、シナリオによって目標方向を決定し、経路を選択する機能を追加している。ここでのシナリオとは、島田らが提案した、「条件」と「行動」に関する単語を組み合わせて構成された 50 例の文章を指す。春山らは、島田ら [?] が作成したシナリオから 7 例を選定し、そのすべてで視覚に基づいて経路を追従して目的地まで到達できることを確認している。選定外のシナリオでは地面の色が変化したり広場などを通過したりするため、視覚に基づいて経路追従するのがより困難な環境の可能性がある。

そのため本論文では、春山らが対象としていないシナリオでも、目的地までカメラ画像のみを入力として経路追従できるか調査する。はじめに、経路追従の成功率を高めるために、ネットワークの変更や新たな学習方法を導入した。シミュレータを用いた実験によって、これらの手法が経路追従の成功率を上昇させるか検証した。さらに、実ロボットを用いた実験によって、春山らが対象としていないシナリオでも目的地まで移動できることを確認した。

キーワード: 自律移動ロボット end-to-end 学習 ナビゲーション

# abstract

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

-Modification of network and experimental evaluation aimed at improving  
route selection success rate

いったん後回し

keywords: autonomouse moblie robot, end-to-end learning, navigation

# 目次

# 図目次

# 表目次

# 第1章

## 序論

### 1.1 背景

移動ロボットにおけるナビゲーションは、目的地までロボットを誘導する制御技術として広く利用されており、物流や、農業、製造業などで活用されている。一般的には、LiDAR や IMU、ホイールエンコーダなどのセンサから得られるデータを用いてオドメトリを計算し、占有格子地図などのメトリックマップに基づいて自己位置推定、経路計画、制御を行うことでロボットを目的地まで誘導する。一方、後述するようなカメラ画像と深層学習に基づくナビゲーション技術の研究も進んでいる。

本研究室の岡田らは、Fig. ??のように、メトリックマップに基づくナビゲーション行動を視覚を入力として模倣することで、視覚に基づいたナビゲーション手法 [?][?] を提案した。この手法では、センサとメトリックマップを入力としたルールベース制御器によって生成されたヨー方向の角速度とロボットに取り付けたカメラから取得した RGB 画像をペアにしてデータセットに加えて学習し、学習後はカメラ画像のみを用いて経路追従行動できることが確認されている。

また、春山らはカメラ画像とシナリオに基づいて、任意の目的地まで自律移動するシステムを提案している。ここでのシナリオとは島田らが提案した、「条件」と「行動」に関する単語を組みわせて構成されている。この手法では、岡田らの視覚に基づいたナビゲーションに加え、カメラ画像から分岐路を認識、シナリオによって目標方向を決定し、経路を選択する機能を追加している。春山らの先行研究では、島田らが提案した 50 例のシナリオの中から対象として

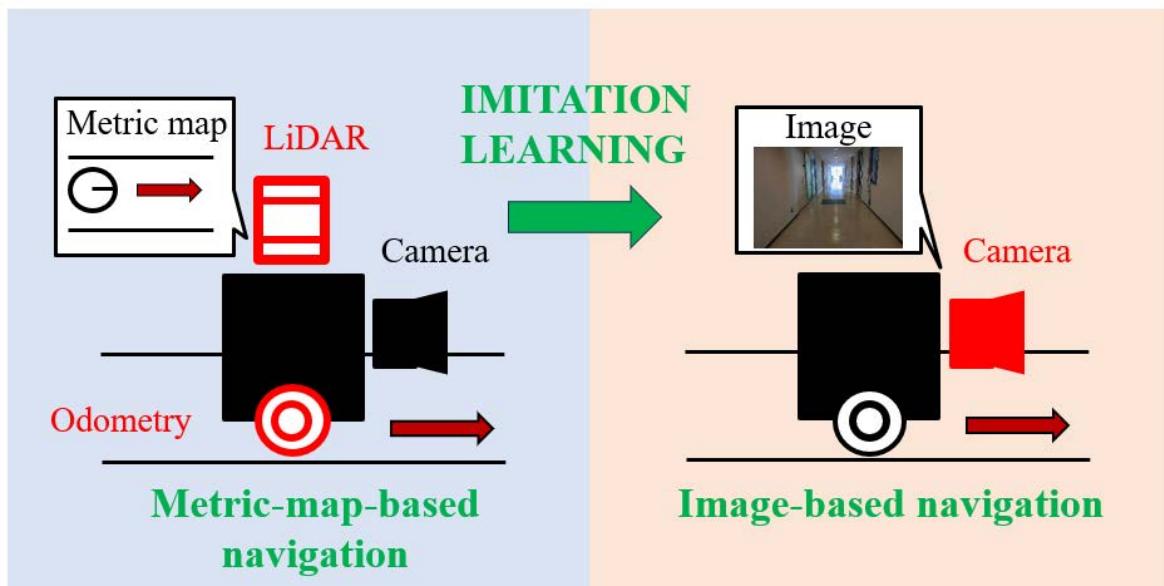


Fig. 1.1: Imitation method of path-tracking behavior

いる 7 例すべてで自律移動が可能であることを確認している。

春山らの先行研究では、島田らが作成したシナリオの中で、以下 Fig. ??の青枠で示すエリアを対象としたシナリオを用いている。この部分はホワイエと呼ばれるスペースを一部を含むものの、壁や床の色が類似しており、一貫性のある環境といえる。一方で Fig. ??の赤色で示すエリアを含むシナリオでは先行研究で走行してきたエリアに加えて、ホワイエを通り抜ける必要があることや、地面の色が異なる区域も対象としていることから、提案する手法で自律走行できないおそれがある。また、春山らの先行研究では対象としたシナリオすべてで目的地まで到達が可能であることが確認されており、失敗の要因は判明していない。

これらの問題点から、新たなエリアを含むシナリオにおいて、目的地までカメラ画像のみで自律移動できるか確認することで、システムの有効性を調査する。

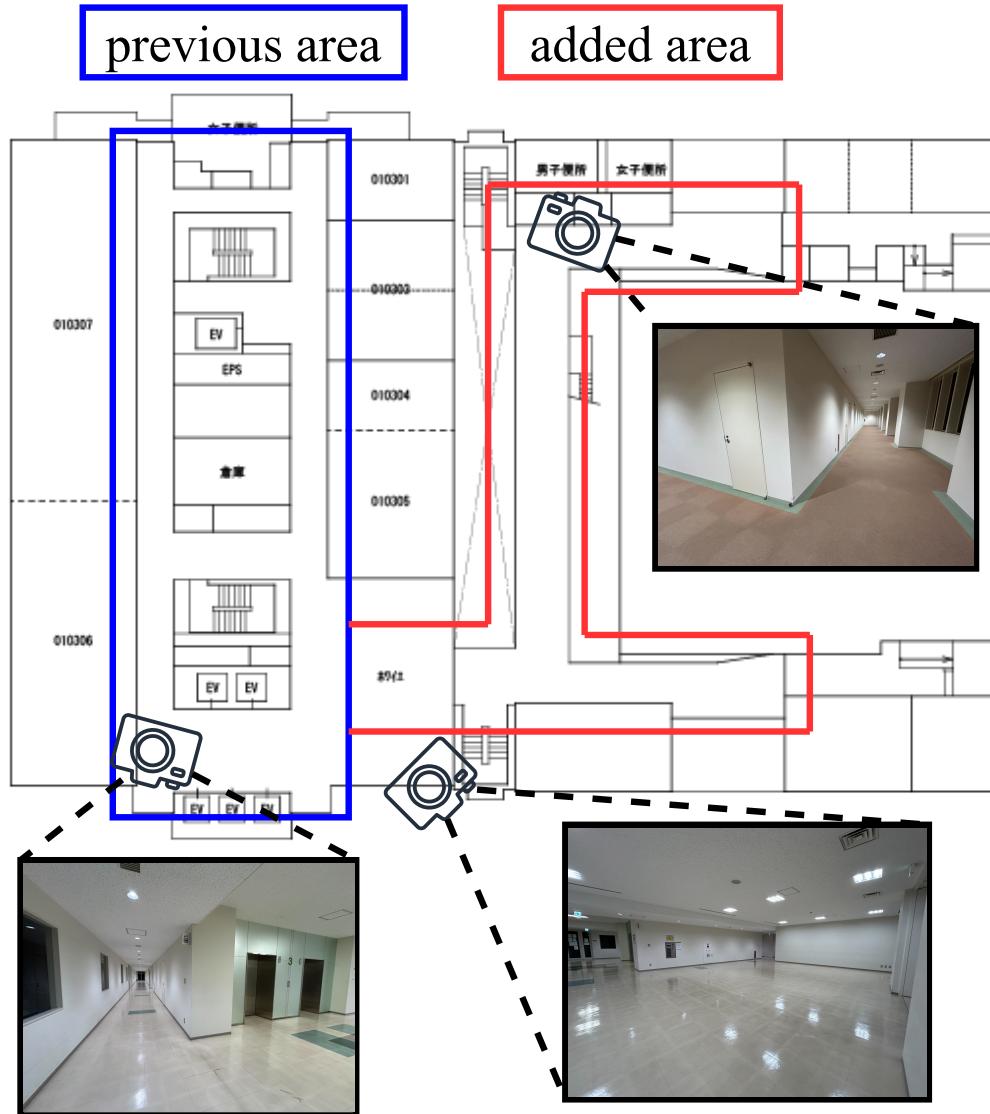


Fig. 1.2: Target area used in the experiment

## 1.2 目的

本論文では、島田らが作成したシナリオにおいて春山らの実験では検証されていないシナリオでも、目的地までカメラ画像のみを入力として経路追従できるかを、実ロボットを用いた実験により確認する。また、経路追従の成功率を高めるためにシステムに改良を加える。

### 1.3 論文構成

第1章では、先行研究や背景、本論文の目的について述べた。第2章では、本研究に関連する技術について述べる。第3章では、先行研究について述べる。第4章では、先行研究からの変更点について述べ、シミュレータを用いた実験によって有効性を調査する。第5章では、実ロボットを用いた実験について述べる。第6章では、本論文について結論を述べる。

## 第2章

# 要素技術

### 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説明する。ナビゲーションを行うためには、LiDAR やオドメトリなどのセンサとメトリックマップを活用し、自己位置推定や経路計画を行うことで、ロボットが目的地まで自律的に移動する仕組みが必要となる。はじめに自己位置推定を行い、ロボットが地図上のどこに位置しているかを特定する。これには、LiDAR やオドメトリデータなどのセンサ情報を利用したアルゴリズムである、AMCL ( Adaptive Monte Carlo Localization )などを活用する。次に、推定した自己位置から目的地までの最適な経路を計画し、経路に基づいてロボットを制御する。メトリックマップに基づくナビゲーションでは、事前に取得した環境情報を活用できるという利点がある。Fig. ??にメトリックマップに基づくナビゲーションの例を示す。本論文ではメトリックマップに基づくナビゲーションにより計算される目標角速度を教師として模倣学習を行う。



Fig. 2.1: Navigation based on metric map

## 第3章

# 関連研究

はじめに，視覚から end-to-end 学習によってナビゲーションを獲得した，岡田らの手法と Felipe らの手法を述べる．次に，島田らが提案したトポロジカルマップの形式，単語の組み合わせによる経路の表現であるシナリオについて述べたのち，春山らの視覚に基づいて目的地まで自律移動するシステムについて述べる．

### 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

岡田らの手法では，メトリックマップに基づく経路追従行動を視覚を入力とした行動へ模倣するために，end-to-end 学習を用いた手法を提案している．ロボットは経路を自律移動するのと同時に学習を行う．手法に基づいて構築されたシステムを Fig. ?? に示す．

訓練時には，ROS の navigation パッケージ [?] を使用して，設定した経路を追従する．ロボットの並進速度は 0.2 m/s に固定する．その際，ロボットに取り付けたカメラから取得した 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にして，0.2 秒の周期でデータセットに追加する．収集には 3 台のカメラを使用することで，データの多様性を高めるとともに過学習を防ぐ効果を狙っている．また，左右のカメラ画像に対するヨー方向の角速度には経路復帰を補助するためのオフセット ( $\pm 0.2\text{rad/s}$ ) を加える．

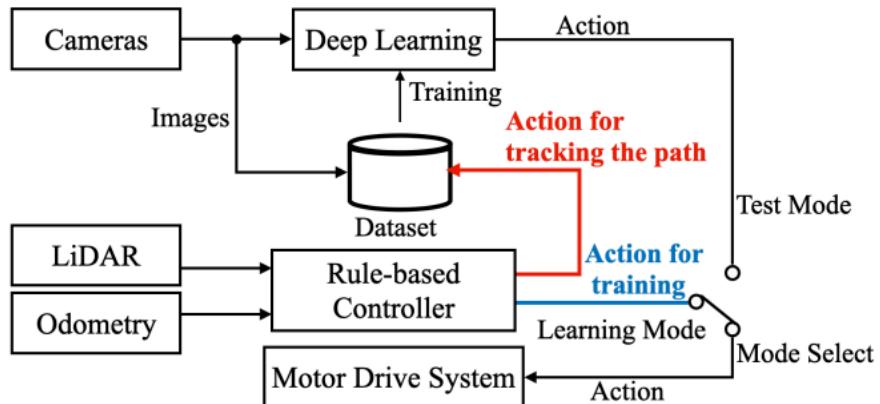


Fig. 3.1: Structure of the Okada and others proposed system ( Quoted from[?] )

Fig. ??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。データセットからはバッチサイズ8で教師データを抽出し，end-to-end 学習を行う。

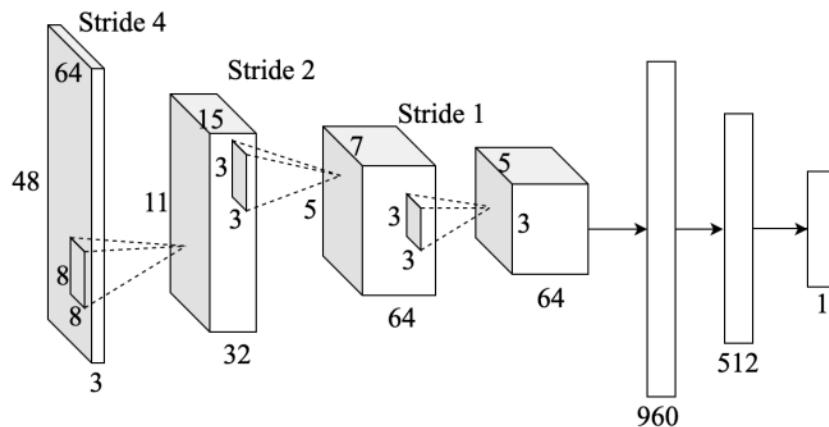


Fig. 3.2: Structure of the network Okada and others used ( Quoted from [?] )

学習器の訓練後は，中央のカメラから得たRGB画像を入力とし，出力されるヨー方向の角速度を用いて経路を追従する。この手法は実ロボットを用いて有効性が調査されており，学習した経路を画像のみを入力とした学習器の出力で経路追従できることが確認されている。

### 3.2 条件付き模倣学習

Felipe ら [?] は視覚を入力とした end-to-end 学習によって自動運転を行う手法において、右折や左折といった行動ごとにネットワークを変更することで、性能が向上することを報告している。具体的には 2 つのネットワークを提案している。1 つ目の Fig. ??に示す (a) のネットワークは画像を処理する CNN、そして CNN の出力と観測、目標方向などのコマンドを入力とする全結合層で構成されている。2 つ目の Fig. ??に示す (b) のネットワークは画像を処理する CNN、そして CNN の出力と観測を目標方向などのコマンドによって全結合層を分ける構造となっている。シミュレータと実環境両方で実験が行われており、どちらも (b) のネットワークがより優れた結果となった。

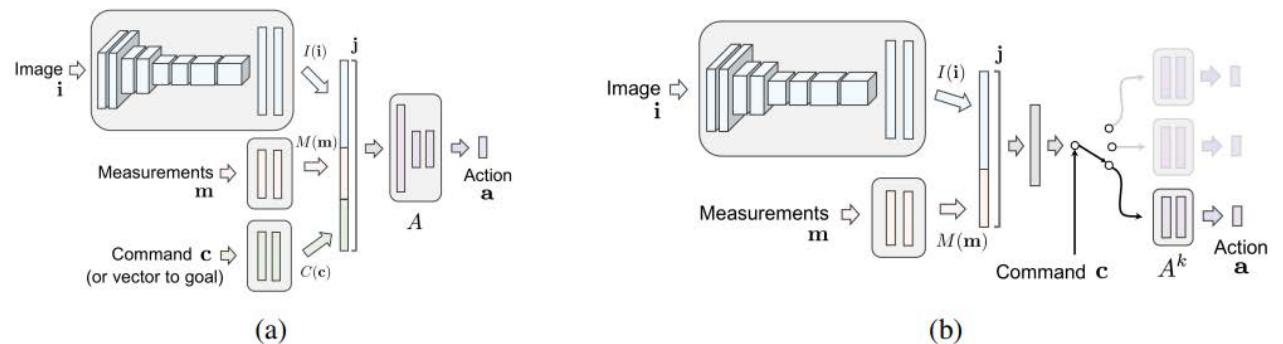


Fig. 3.3: Two network architectures for command-conditional imitation learning  
(Quoted from[?])

### 3.3 トポロジカルマップとシナリオ

#### トポロジカルマップ

トポロジカルマップとは、環境をランドマークや特徴的な箇所をノードとし、その繋がりをエッジで表現した地図である。島田ら [?] が提案したトポロジカルマップを Fig. ??に示す。提案された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、エッジはノード間を接続する。ノードには ID、通路の特徴 (Type)、エッジ ID と相対角度 (Edge) のデータが含まれ、エッジには ID のみが含まれる。この形式は道案内に関するアンケート結果に基づいており、人が道案内で「通路の特徴」や「向いている方向」を重視することが明らかになったことから生成されている。

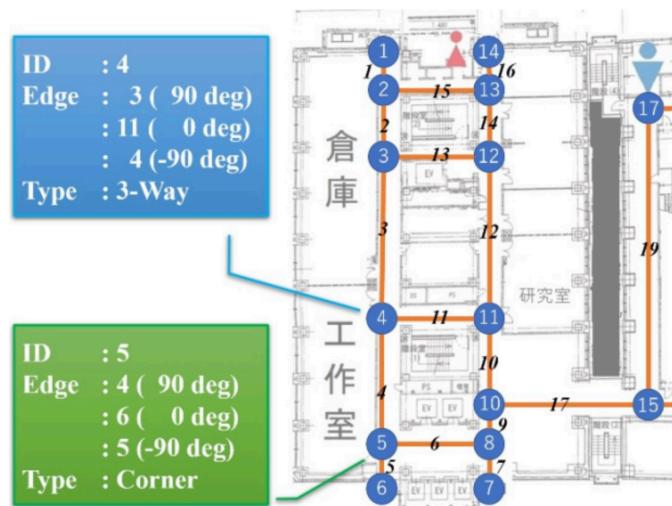


Fig. 3.4: Topological map format proposed by Shimada and others (Quoted from[?])

### シナリオ

シナリオは、トポロジカルマップ上の目的地までの経路を「条件」と「行動」の組み合わせで表現する手法である。「条件」は「次の角」や「突き当たりまで」などを指し、「行動」は「直進」や「右折」などを指す。この形式はトポロジカルマップ同様に道案内に関するアンケート結果に基づいており、人が「条件」と「行動」を組み合わせて道案内をすることが明らかになったことから生成されている。例として、Fig. ??に示す経路は「突き当たりまで直進、左折、突き当たりまで直進、停止」と表現される。

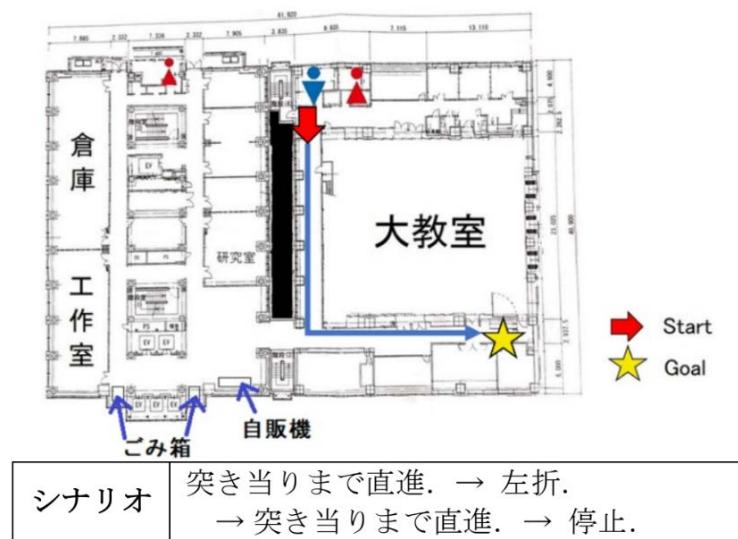


Fig. 3.5: Example scenario proposed by Shimada and others (Quoted from[?])

### 3.4 視覚に基づいて目的地まで自律移動するシステム

春山らは、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している。提案されたシステムの概要図を Fig. ??に示す。

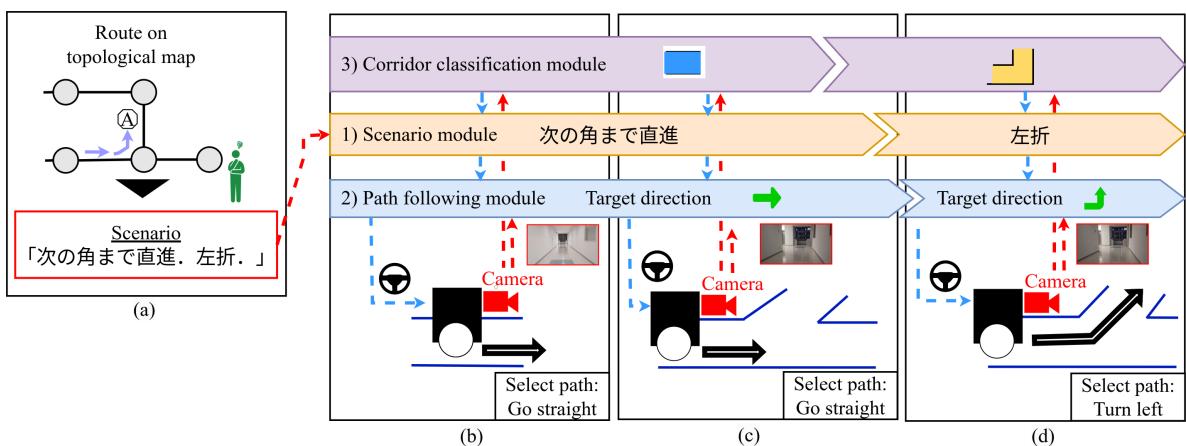


Fig. 3.6: Overview of proposed system by haruyama and others (Quoted from [?])

- 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を与えることで、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) カメラ画像から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）

の3つのモジュールで構成される。ロボットは下記の a から d の一連の流れにより、指示された経路に沿って目的地まで自律移動する。

- (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、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。最初の条件と行動のセットは「次の

角まで」と「直進」である。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、それをシナリオモジュールに与える。これにより、シナリオモジュールは「次の角まで」という条件を満たされたことを確認し、次の行動である「左折」へ遷移する。
- (d) 「左折」に基づいて、経路追従モジュールがロボットを制御し、経路に沿って角を左折する。

### シナリオモジュール

シナリオモジュールは、トポロジカルマップを基に作成されたシナリオから「条件」や「行動」を解釈し、それを分岐路での目標方向に変換して出力する機能を持つ。トポロジカルマップは、特徴的な通路のノード（青）とそれを繋ぐエッジ（緑）で構成され、ノードにはIDや通路の特徴、接続エッジと方向のデータが含まれている。シナリオは目的地までの経路を「条件」と「行動」で表現し、例として「三叉路まで直進。右折。突き当たりまで直進。停止」となる。

シナリオの目標方向への変換では、句点ごとに分解し、「条件」と「行動」を抽出して以下の項目に分類する。

- 1) 通路の特徴（例：「三叉路」「角」）
- 2) 順番（例：「3つ目の」「2番目の」）
- 3) 方向（例：「左手に」「右手に」）
- 4) 行動（例：「右折」「停止」）

Fig. ??に示す例で、「三叉路まで直進」は通路の特徴「三叉路」と行動「直進」に分解される。この処理を経路全体に対して行い、得られた「行動」を分岐路での目標方向として変換し、経路追従モジュールに渡す。また、条件の判定には通路分類モジュールを使用する。

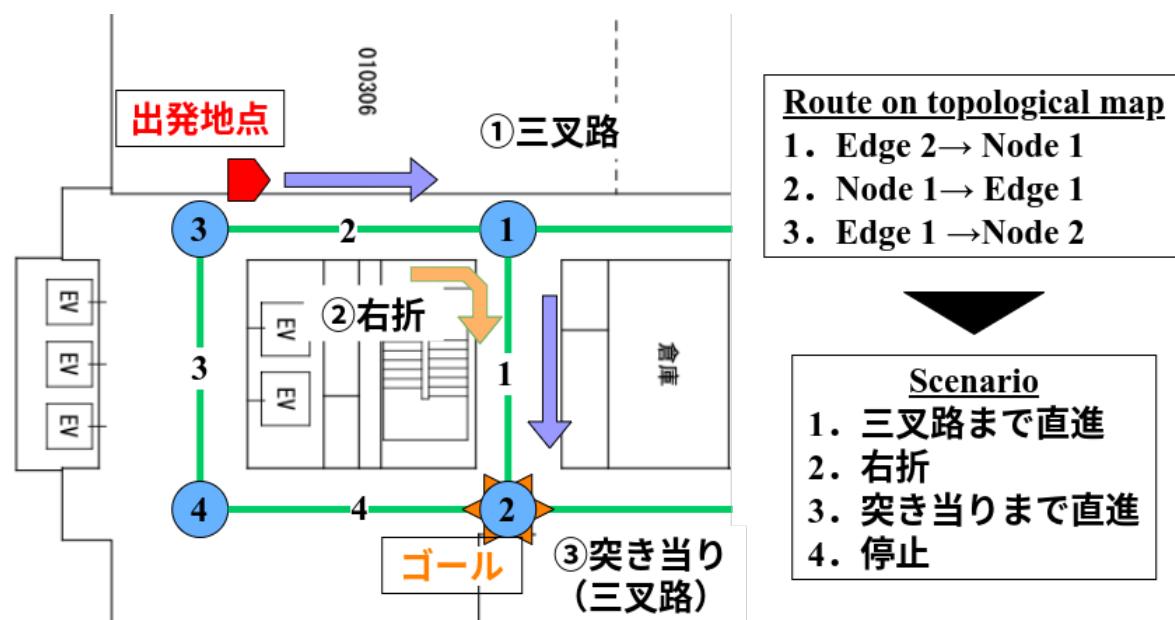


Fig. 3.7: Example of topological map and created scenario (Quoted from [?])

### 経路追従モジュール

このモジュールは、岡田らの手法から目標方向のデータを加えることで、分岐路で経路を選択し、移動する機能を追加したものである。ここで目標方向とは、目標とする進行方向（「直進」や「右折」）を表す。学習時は、カメラ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、目標方向を0.2秒周期でデータセットに加える。データセットから抽出するバッチサイズや、カメラ画像の解像度は岡田ら手法と同様である。データセットの収集には藤原ら[?]が提案した、Fig. ??に示す、目標方向が「左折」、「右折」のデータを7倍にする手法、Fig. ??に示す、学習時に積極的な蛇行する手法を採用する。

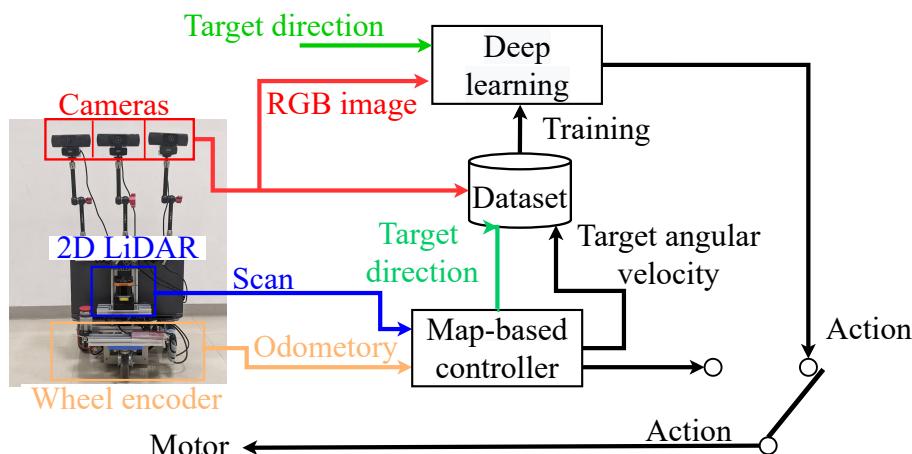


Fig. 3.8: Path-following module system (Quoted from [?])

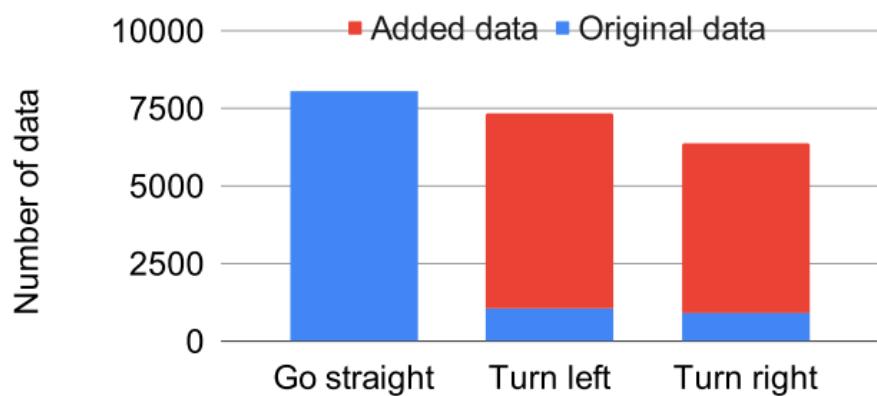


Fig. 3.9: Number of data in the target direction per 10000 steps in the previous experiment(Quoted from [?])

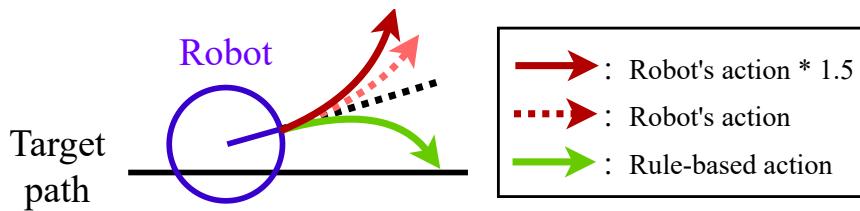


Fig. 3.10: Aggressive meandering(Quoted from [?])

### 通路分類モジュール

このモジュールでは、ニューラルネットワークを用いることで、カメラ画像を入力として、通路の特徴を分類する。データセットの収集のために、ロボットをルールベース制御器に基づいて走行させる。その際に、フレーム数 16、画像サイズ  $64 \times 48$  の連続したカメラ画像と通路の分類ラベルを 1 組として、0.125 秒周期でデータセットに加える。通路の分類ラベルのアノテーションはルールベース制御器から出力されるラベルによって自動的に行う。データセット内の不均衡を改善するために、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を導入している。

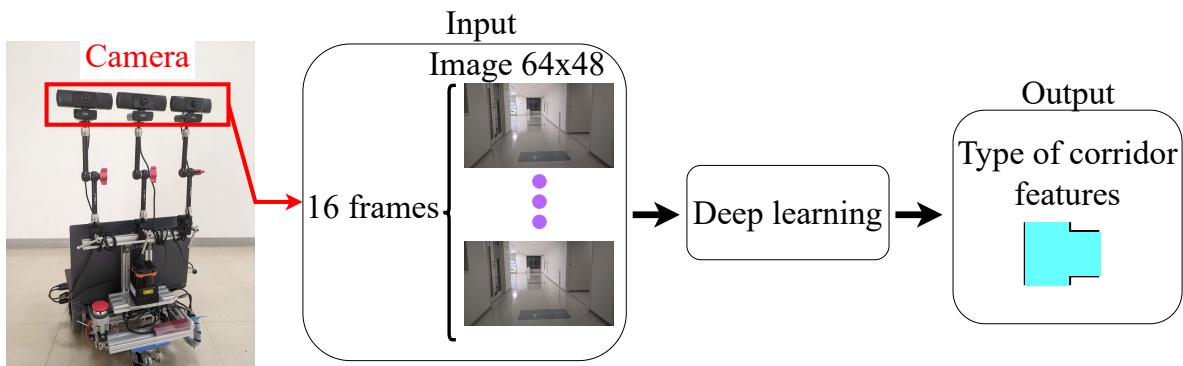


Fig. 3.11: Path-following module system (Quoted from [?])

### 実ロボットを用いた実験

実ロボットを用いた実験により、ロボットを目的地まで到達可能か検証されている。実験では島田ら用いた 50 例のシナリオの中から、7 例が用いられており、そのすべてでロボットが目的地へ到達できることが確認されている。Fig. ??に走行可能なシナリオを示す。

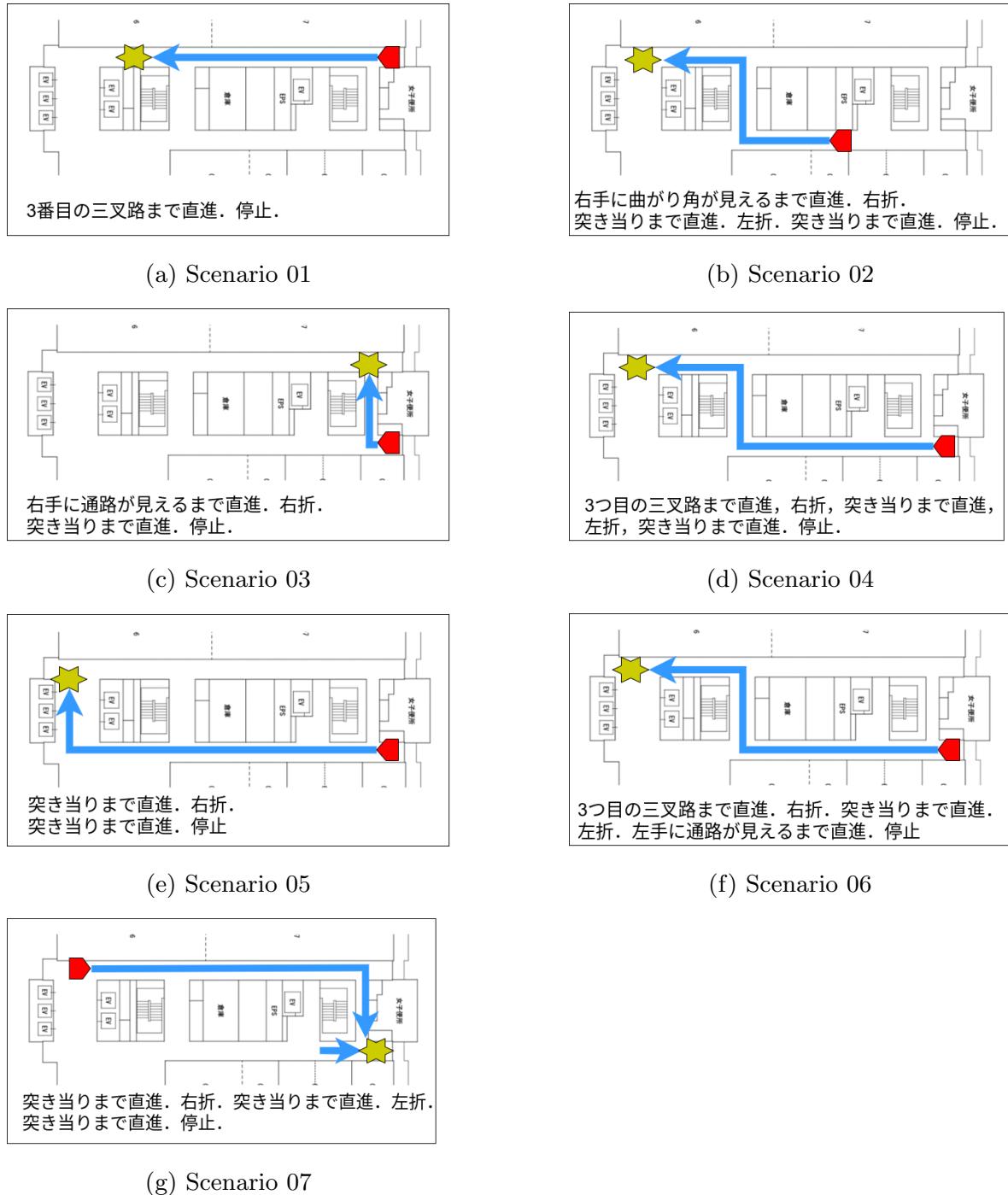


Fig. 3.12: Scenarios used in the experiment (Quoted from [?])

## 第4章

# 機能の改善

経路追従モジュールに関して，経路追従の可能性を向上させるために2点変更を加えた．変更点，理由について以下に述べる．また，シミュレータを用いた実験により，それらの効果を調査する．

### 4.1 ネットワークの変更

春山らの先行研究では，Fig. ??に示すネットワークを使用していた．一方で felipe らの先行研究によると，コマンドによってモデルを分岐する形式のネットワークがより経路追従の成功率が向上すると報告している．そのため，今回の研究では felipe らによって提案されたネットワークを参考に Fig. ??に示す，新たなネットワークを構築した．ネットワークの入力は春山らが作成したものと同様で  $64 \times 48$  の RGB 画像と Table ??に示す，目標方向のワンホップベクトルで，出力はヨー方向の角速度である．また，損失関数や活性化関数などのパラメータは春山らの手法と同様である．

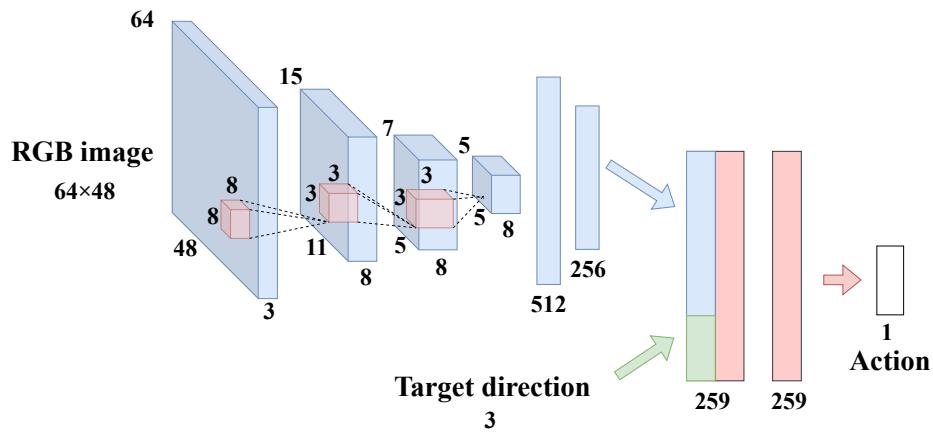


Fig. 4.1: Structure of the network haruyama and others used(Quoted from [?])

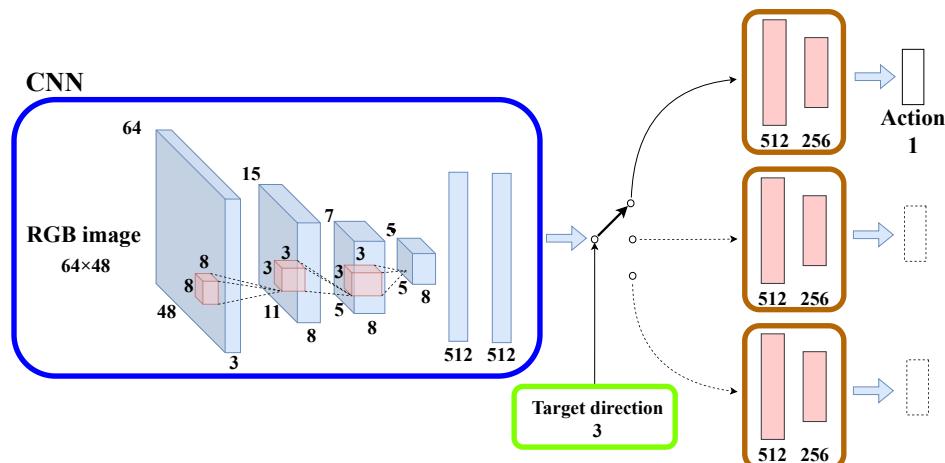


Fig. 4.2: Branched network

Table 4.1: 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]

## 4.2 オフライン学習

春山らの先行研究では学習機の訓練の手法にオンライン学習を用いていた。オンライン学習の欠点として、学習するデータに偏りが発生してしまう点が挙げられる。具体的には、学習の初期に取得したデータは学習される回数が多くなり、学習の後半に取得したデータは学習のされる回数は減少する。これにより、学習が不十分な場合、経路追従できない箇所が存在する可能性がある。この欠点を補うために、オフライン学習を併用して行う。オフライン学習とは一般的に用いられる学習方法で、予め収集したデータを使用して学習する手法を指す。データを予め収集することにより、すべてのデータを同じ回数学習することが可能である。また、オンライン学習では学習するために走行する必要があるが、オフライン学習を併用することで追加の学習を走行せずに実現することができる。

## 4.3 シミュレータを用いた実験

まずは、新たに作成したネットワークで経路を正しく選択できるかシミュレータを用いた実験により調査する。次にオフライン学習を追加で行い、効果を検証する。

### 実験装置

シミュレータに Gazebo[?] を使用する。ロボットには TurtleBot3 Waffle Pi[?] に 3 つのカメラを追加したモデルを用いる。

### 実験環境

実験環境として Fig. ??に示す、千葉工業大学 2 号館 3 階を模した環境を使用する。経路の選択を行う場所として Fig. ?? の A, B の分岐路を対象に行う。Fig. ?? に示すように、A, B の各分岐路で経路に侵入するパターンは 3 つ、また脱出するパターンが 2 つあることから合計 12 回の経路選択を行うことができる。

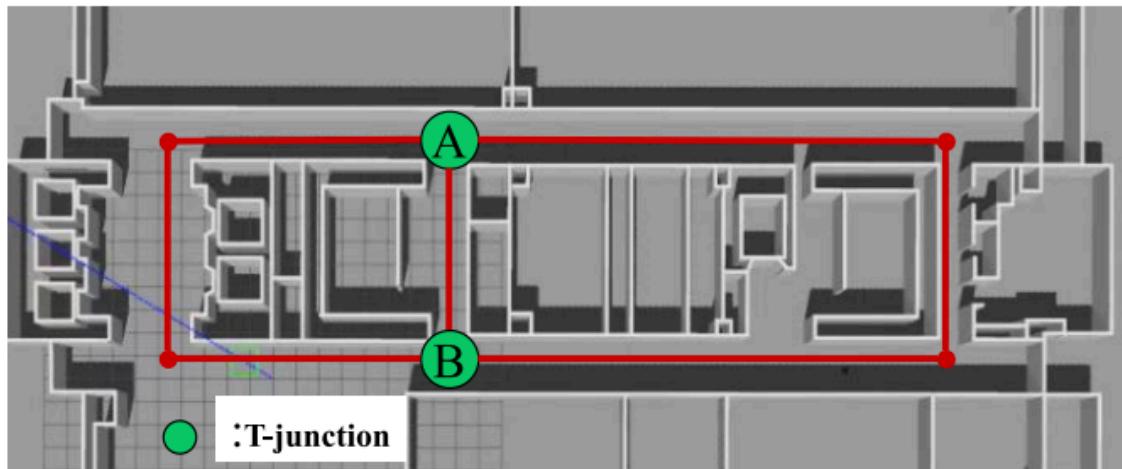
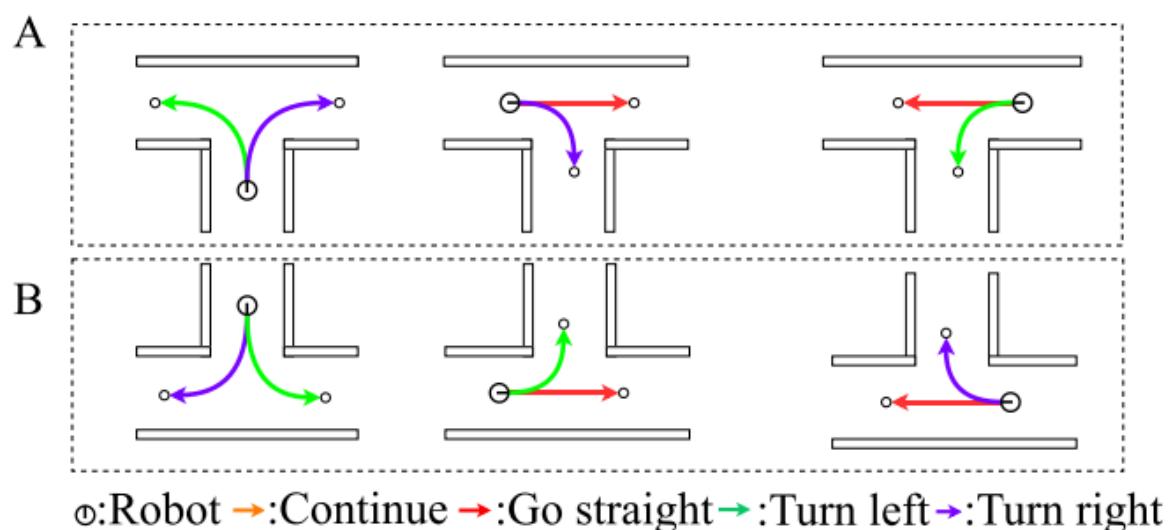


Fig. 4.3: Experimental environment(Quoted from [?])



①:Robot →:Continue →:Go straight→:Turn left→:Turn right

Fig. 4.4: Selecting a path the T-junction(Quoted from [?])

### 実験方法

Fig. ??に示す経路を， a から f の順番で走行しながら，模倣学習を行う。データセットの収集には，藤原ら [?] が提案している，学習データの不均衡を改善する手法，学習時の積極的な蛇行を行う手法を採用する。テスト時は，学習器の出力で壁に衝突することなく，分岐路の先の経由点に到達することができれば成功とする。オフライン学習に使用するデータセットは，a から f の順番で走行しながらオンライン学習を行った際に作成されるものを使用する。オフライン学習のパラメータとして，バッチサイズはオンライン学習と同様の 8 でデータセットからランダムにデータを抽出し，epoch 数は 10 とする。実験では，学習とテストを繰り返し 10 回行う。

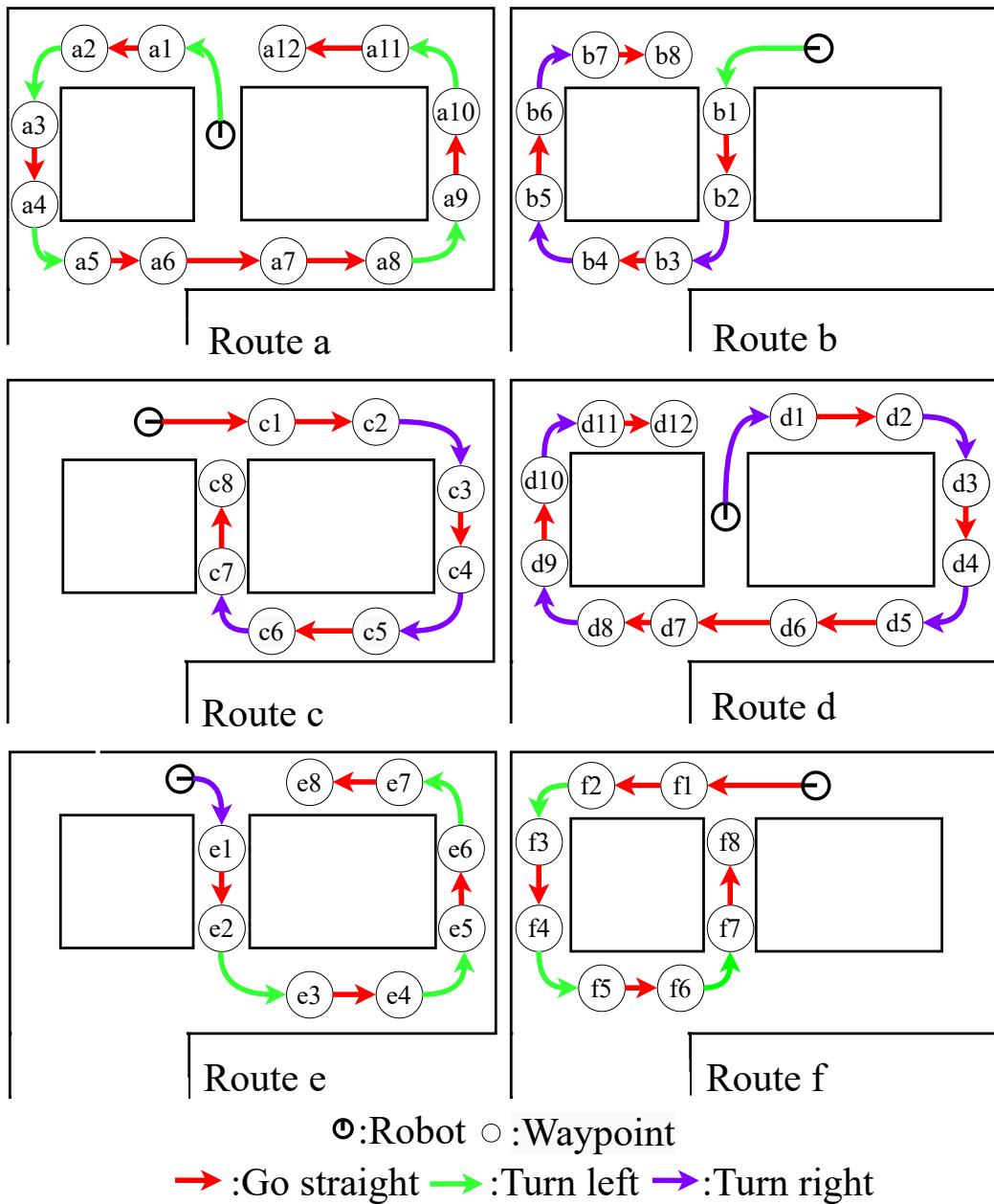


Fig. 4.5: Route for experiment(Quoted from [?])

Table 4.2: Success rate

Experiment	Step + Epoch	Total result
Previous method	10000	109/120 (90.8%)
	20000	114/120 (95.0%)
Branched	10000	113/120 (94.2%)
	20000	115/120 (95.8%)
Branched +	10000 + 10	115/120 (95.8%)
Offline learning		

## 実験結果

Table ??に結果を示す。表では、春山らが使用していたネットワークを Previous method としている。そして新たなネットワークを使用した実験を Branched とした。

新たに作成したネットワークでも、経路を正しく選択することができた。step 数が 10000 の場合は、先行研究の手法と比較すると成功率が上昇した。しかし step 数が 20000 の場合は先行研究と成功率に大きな差は生じなかった。

オフライン学習を併用した場合では、成功率が 20000 step と同等になった。一方で、オンライン学習の弱点であったルートの後半部分での失敗を減少させることができた。具体的には Branched で 10000 step の実験を行った際に一番失敗の多い箇所であった、Fig. ??の Route f の f6 から f7 にかけての経路選択の成功率が向上した。Table ??に、それぞれの実験での f6 から f7 にかけての経路追従の成功率を示す。

これらの結果から、従来の手法では 20000 step 走行しながら学習していたものを、10000 step まで減らすことができたと言える。また、ネットワークの変更やオフライン学習は経路選択の可能性を向上させるには有効であると考えられる。

Table 4.3: Success rate at f6 to f7

Experiment	Step + Epoch	Total result
Branched	10000	5/10
	20000	7/10
Branched +	10000 + 10	8/10
Offline learning		

## 第5章

# 新たなシナリオが走行できるか検証

実ロボットを用いて、新たなエリアを含んだシナリオに関して、ロボットが目的地へ到達可能であるか検証する。

### 5.1 実験装置

Fig. ?? に実験に使用する icart-mini[?] をベースに開発したロボットを示す。ロボットには以下に示すセンサ、PC を搭載している。

- 1) ウェブカメラ (サンワサプライ株式会社 CMS-V43BK)
- 2) 2D-LiDAR (北陽電機 UTM-30LX)
- 3) PC (GALLERIA GCR2070RGF-QC-G)
- 4) エンコーダ

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[?] を使用する。



Fig. 5.1: Experimental setup

## 5.2 実験方法

### 5.2.1 実験環境

実験環境として Fig. ??に示す千葉工業大学 2 号館 3 階を用いる。環境中には、三叉路が 6 つ、角が 3 つ、突き当りが 4 つ含まれている。また、経路追従モジュールと通路分類モジュールの学習データを収集するために、Fig. ??に示すルートを走行する。

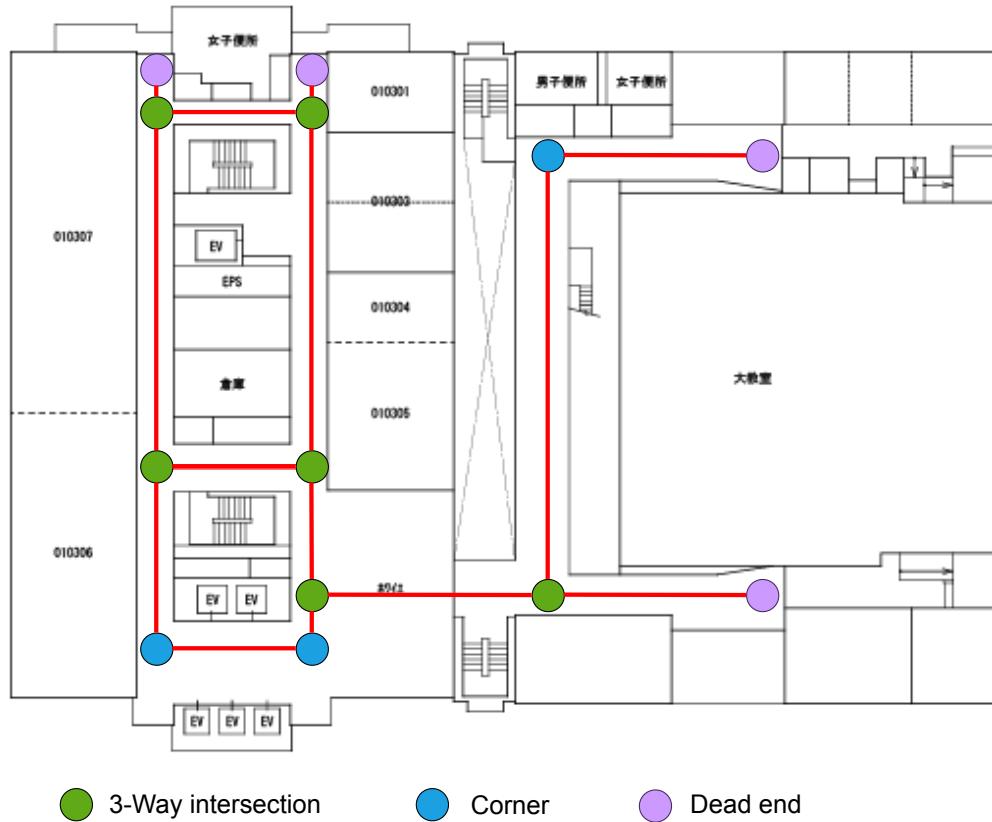
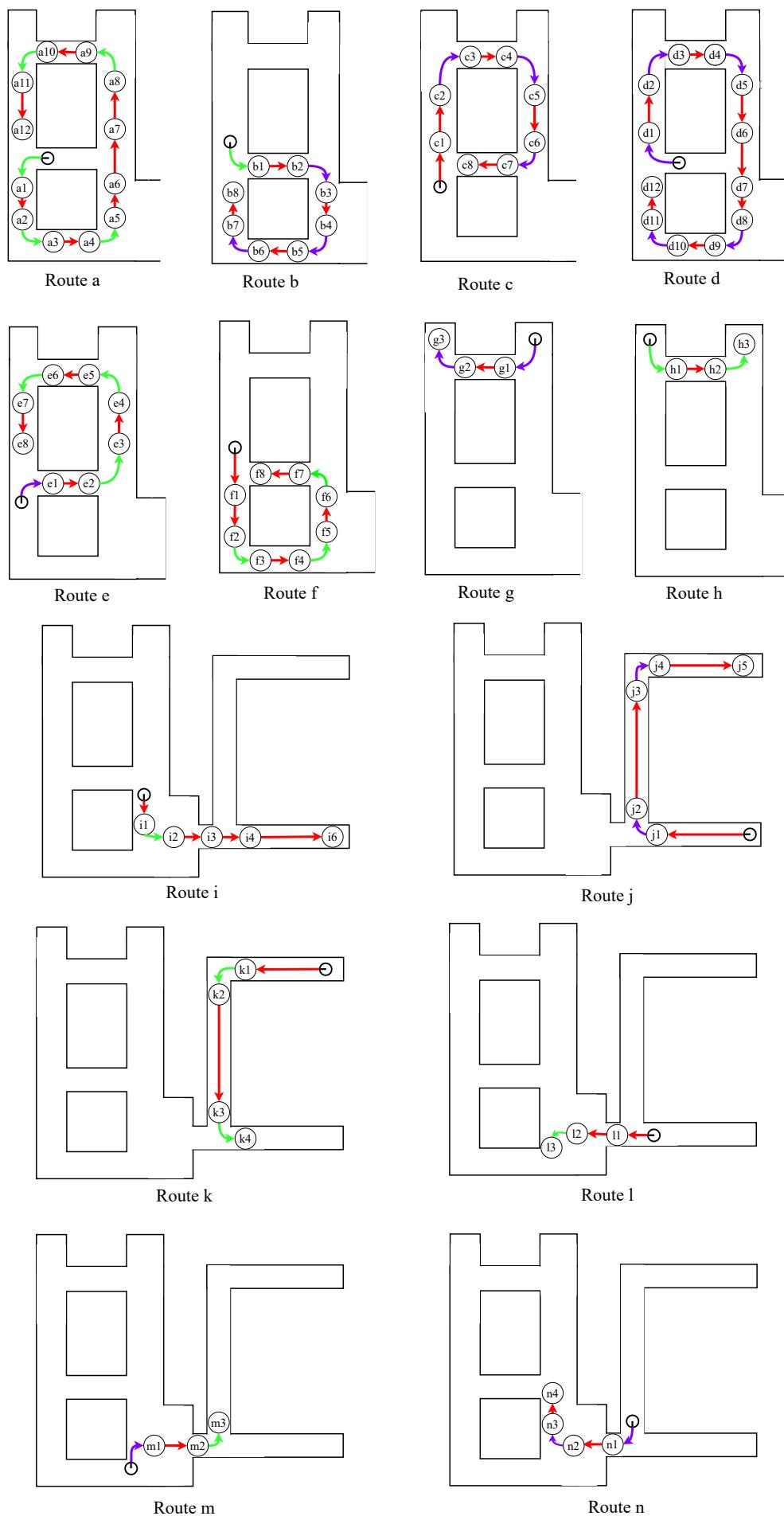


Fig. 5.2: Experimental environment

### 5.2.2 シナリオの選定

実験では島田ら用いた50例の中から、28例を選定した。選定するにあたって、以下の条件を設定した。

- 1) ロボットの移動に安全性が確保できない、Fig. ??上部に示す部分を走行ルートに含まないこと。
- 2) 経路追従モジュールができるない、「後ろを向く」などその場での旋回が含まれていないこと。
- 3) 正面の単眼カメラの画像からでは通路の分類が困難な Fig. ??下部に示す部分がスタート地点では無いこと。



①: Robot →: Go straight →: Turn left →: Turn right

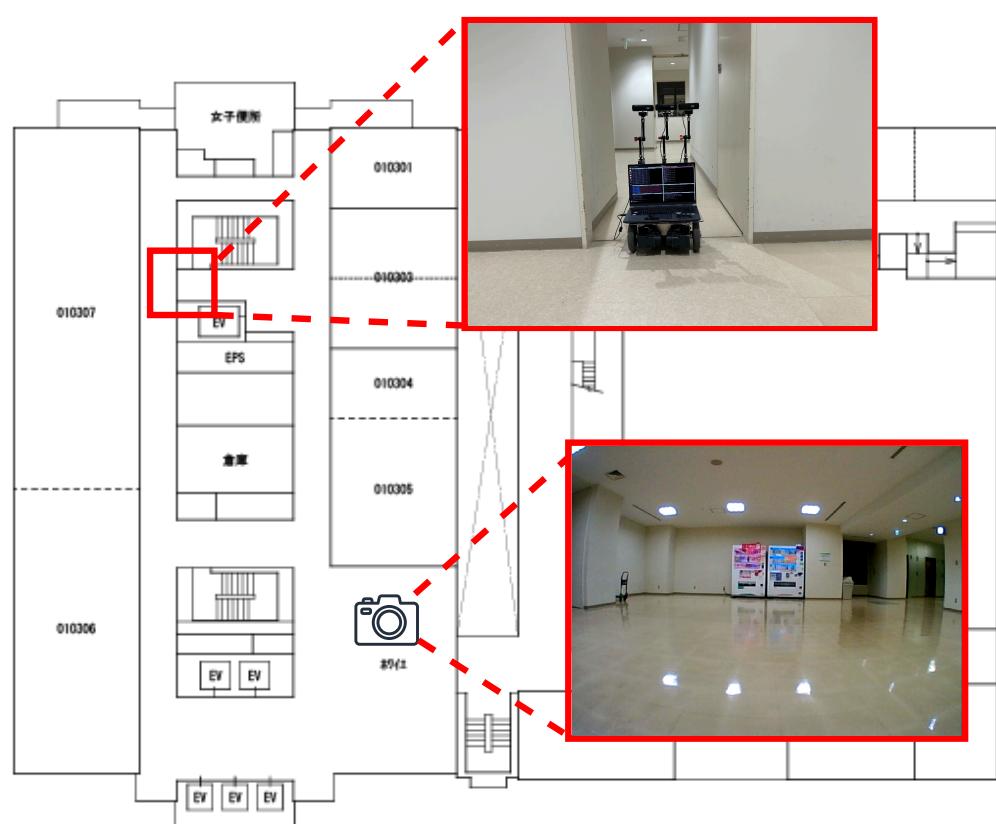


Fig. 5.4: Experimental

### 5.2.3 経路追従モジュールの訓練

Fig. ??に示すルートをオンライン学習させながら1周走行する。データセットの収集には藤原ら [?] が提案する手法を用いる。また、オンライン学習で作成したモデルに追加でオフライン学習を行う。オフライン学習時のデータセットにはオンライン学習の際に作成したルート1周分のデータを用いる。データセットからはオンライン学習と同様のバッチサイズ8でデータをランダムに取得し、epoch数は20とした。

### 5.2.4 通路分類モジュールの訓練

Fig. ??に示すルートをROSのnavigationパッケージを使用して、経路を1周する。その際、3つのカメラからそれぞれ画像データを収集しながら走行する。学習時のパラメータとして、バッチサイズを32、epoch数を30とし、コストアプローチに用いた重みはTable ??に示す。

Table 5.1: The weights assigned to each class in the experiment

Class	Class weights
直進	1
突き当たり	39.5
角(右)	14
角(左)	14.2
十字路	1
三叉路(右)	6.6
三叉路(中央)	7.0
三叉路(左)	6.4

### 5.2.5 シナリオに基づくナビゲーション

2つのモジュールを訓練後，ロボットが目的地まで到達できるか確認する。実験では，ロボットをシナリオのスタート地点，向きに配置し，シナリオを1例ずつ入力する。壁に衝突することなく正しい経路を選択し，目的地で停止した場合に成功とする。

### 5.3 実験結果

選定したシナリオ 28 例中，24 例の成功を確認した。Table ??に全シナリオの走行の成否を示す。また，1 例として Scenario35 を走行時の実験の様子を Fig. ??に示す。

先行研究で経路追従を確認していないエリアでも，経路追従することが確認できた。また，ロボットがシナリオの道順に沿って，分岐路で適切な経路を選択する様子が確認できた。

Table 5.2: Results of the scenario-based navigation

scenario	success or failure
1	
3	✗
4	✗
5	
6	
9	✗
20	
21	
22	
24	
25	
27	
28	
29	
30	
31	
32	
33	
34	✗
35	
36	
37	
38	
39	
40	
41	
42	
50	



Fig. 5.5: An example of the robot applied the proposed system

失敗した4例ではそれぞれ、曲がり角で左折すべきところを直進した。Fig. ??に失敗した箇所を示す。失敗した4例の内訳として、Fig. ??の青枠に示す失敗が3例、緑枠に示す失敗が1例となった。失敗した箇所でも通路の特徴の分類は正しく行われており、経路追従モジュールに与えられる目標方向も正しい値が入力されていた。失敗の原因として、通路分類モジュールの出力に遅れがあることが考えられる。Fig. ??とFig. ??に示す画像は失敗箇所において、学習時とテスト時に「左折」の目標方向を与えたタイミングでのロボットの位置である。学習時は角から1–2 mほど前方で目標方向が変化しているのに対し、テスト時では角から0.2–0.5 mまで近づいてから与えられていることがわかった。経路追従モジュールに目標方向が与えられるタイミングは、通路分類モジュールの出力が変化したタイミングとなる。つまり通路分類が遅ることで目標方向が与えるタイミングが遅くなり、曲がることができないと考えた。確認として、学習時と同じタイミングで経路追従モジュールに目標方向を与えた場合、曲がる様子が確認できた。このことから、失敗した要因として通路分類モジュールの出力に遅れがあることが考えられる。

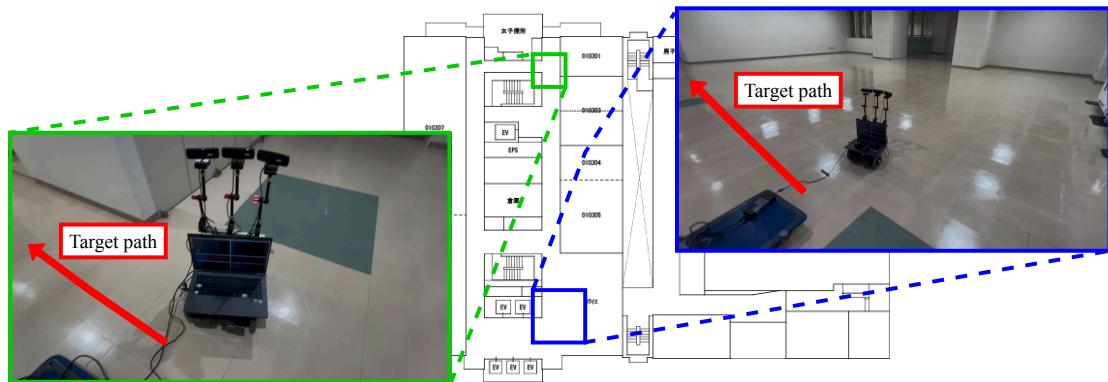
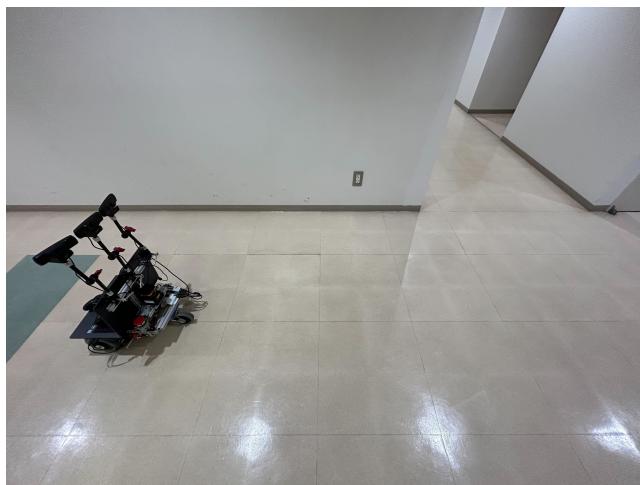


Fig. 5.6: Failed place



(a) Location of label changes during learning

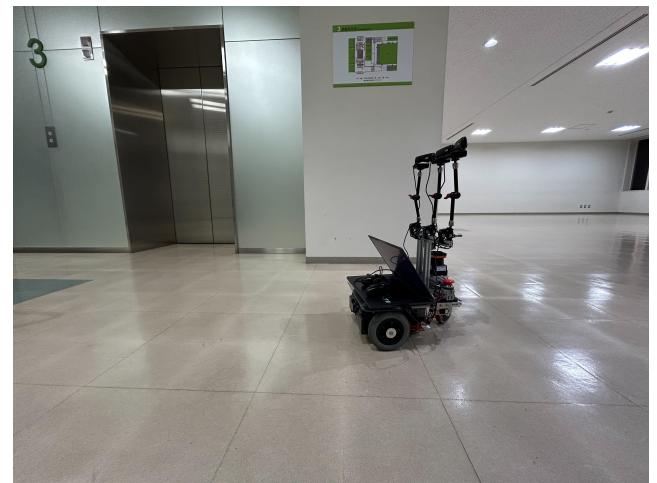


(b) Location of label changes during testing

Fig. 5.7: Failure point 1



(a) Location of label changes during learning



(b) Location of label changes during testing

Fig. 5.8: Failure point 2

## 第6章

# おわりに

### 6.1 結論

本論文では、春山らが提案したシステムを改良し、先行研究では走行が未確認であったシナリオに対しても目的地までカメラ画像のみを入力として自律移動可能か調査した。経路追従の成功率を向上させることを目的として、先行研究からは、行動ごとにモデルを分けるネットワークに変更、オンライン学習による追学習する仕組みを追加した。また、これらの手法が有効であることをシミュレータでの実験で検証した。実ロボットを用いた実験では、先行研究では走行が確認されていないエリアをシナリオを含む場合でも28例中、24例は自律移動が可能であること確認した。失敗した4例に関して、通路分類モジュールの出力が遅れることによって、経路追従に失敗すると考えられる。

## 参考文献

- [1] 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-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.
- [2] 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.
- [3] 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.
- [4] ros planning. 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).
- [5] Felipe Codevilla, Matthias Müller, Antonio López, Vladlen Koltun, Alexey Dosovitskiy. End-to-end driving via conditional imitation learning. 2018.
- [6] 春山健太, 藤原恆, 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.
- [7] 藤原恆, 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-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.
- [8] Gazebo. Gazebo. <https://gazebosim.org/home>. (Accessed on 12/20/2023).
  - [9] ROBOTIS-GIT. turtlebot3. <https://github.com/ROBOTIS-GIT/turtlebot3>. (Accessed on 12/20/2023).
  - [10] 春山健太, 藤原恆, 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-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.
  - [11] T frog Project. icart\_mini. [https://t-frog.com/products/icart\\_mini/](https://t-frog.com/products/icart_mini/). (Accessed on 12/20/2023).
  - [12] open rdc. orne\_navigation. [https://github.com/open-rdc/orne\\_navigation.git](https://github.com/open-rdc/orne_navigation.git). (Accessed on 12/20/2023).

## 謝辞

本研究を進めるにあたり，1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。