

# 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-

○春山健太（千葉工大），藤原恆（千葉工大）馬場琉生（千葉工大）

石黒巧（千葉工大）上田隆一（千葉工大）林原靖男（千葉工大）

## A proposal for an online imitation method of path-tracking behavior by end-to-end learning of vision and action - Adding and consideration of path selection function based on a topological map and scenario -

○ Kenta HARUYAMA (CIT), Masaki FUJIWARA (CIT), Ryusei BABA (CIT),

Takumi ISHIGURO (CIT), Ryuichi UEDA (CIT), Yasuo HAYASHIBARA (CIT)

**Abstract:** This manuscript describes a method for preparing a manuscript for the annual conference of the SICE SI division. This manuscript describes a method for preparing a manuscript for the annual conference of the SICE SI division. This manuscript describes a method for preparing a manuscript for the annual conference of the SICE SI division. This manuscript describes a method for preparing a manuscript for the annual conference of the SICE SI division.

### 1. 緒言

本研究グループでは、end-to-end 学習により、カメラ画像と目標とする経路の情報を入力として、経路を追従する行動をオンラインで模倣する手法を提案してきた[1][2]。また、実験により手法の有効性を検証してきた。

これまで提案した手法では、LiDAR やオドメトリ、事前に作成した幾何学的な地図を入力とする ROS Navigation Stack[3]などの地図に基づいたルールベース制御器によって経路追従する。その際、この行動をカメラ画像と目標とする進行方向の情報（以後、目標方向と呼ぶ）を入力としてオンラインで模倣するように学習する。これにより、Fig.1 に示すカメラ画像と目標方向に基づき、経路を選択して追従する行動を獲得できる。

この手法により、地図に基づく経路追従とカメラ画像を入力とする経路追従の 2 つのナビゲーション手段が得られる。この 2 つの手段を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。

前報では[1][2]、カメラ画像を入力とする経路追従における、目標方向の出力方法を議論の対象としておらず、目標方向の出力を地図に基づいたルールベース制御器に依存していた。そのため、カメラ画像を入力とする経路追従においても、事前に作成した幾何学的な地図を必要とする問題があった。この問題の解決には、地図に基づいたルールベース制御器に依存しない、つまり幾何学的な地図を必要としない目標方向の出力方が必要だと考えられる。解決に向けて、幾何学的な

地図を用いずに、目標方向を出力する手段を検討する。本稿では、カメラ画像を入力とする経路追従に対して、島田らが提案した「突き当りまで」という「条件」や「直進」などの「行動」による経路の表現（以後、シナリオと呼ぶ）[4]を目標方向へ変換し、出力する仕組みを追加する。そして、事前に作成した幾何学的な地図を用いず、カメラ画像とシナリオに基づいて経路を追従し、目的地まで自律移動する手法を提案する。また、実口ボットを用いた実験を通して、提案した手法の有効性を検証する。

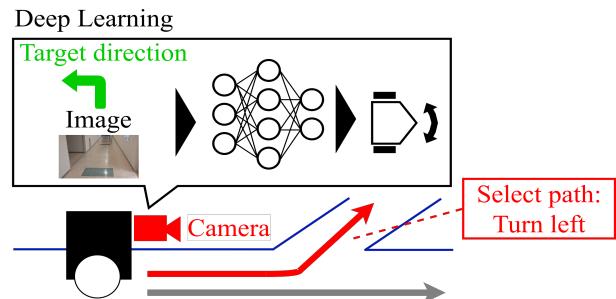


Fig.1 Path selection and following behavior based on camera images and target direction by imitation learning.

### 2. 提案手法

提案する手法では、カメラ画像と「三叉路まで直進」といったシナリオを入力として、経路を追従することで目的地まで自律移動する。この手法の特徴として、目的地までの自律移動に、LiDAR などを用いて事前に

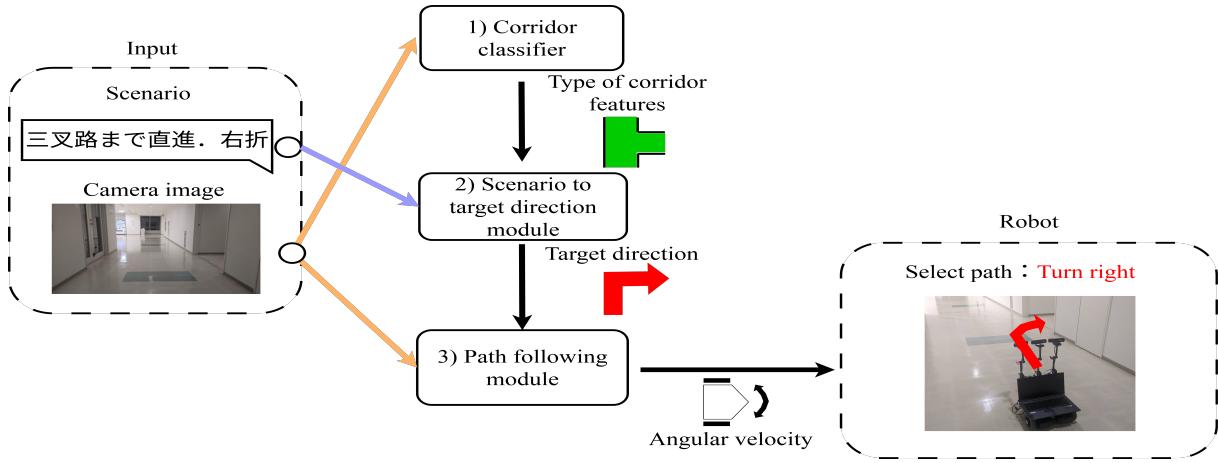


Fig.2 Overview of the system used in the method

作成した幾何学的な地図を必要としないことが挙げられる。ここでは全体の流れについて述べた後、それぞれの詳細を述べる。

手法のシステム概要と動作例を Fig.2 に示す。システムは、ロボットに取り付けたカメラから得た画像データと、人が作成したシナリオを入力とし、

- 1) シナリオの「条件」の判定に用いる、通路の特徴を分類する通路分類器
- 2) シナリオを解釈し、目標方向へ変換して出力するモジュール
- 3) カメラ画像と目標方向に基づく経路追従モジュール

の3つのモジュールで構成されている。Fig.2 の動作例では、人間によって作成された「三叉路まで直進。右折」というシナリオがシステムへ入力されている。この入力されたシナリオを 2) のモジュールが「条件」と「行動」に分解し、この行動を目標方向へ変換して出力する。具体的には、1) の通路分類器から「三叉路」という分類結果が得られるまで、「直進」の目標方向を出力し、3) の経路追従モジュールへ与える。3) の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をもとに、「直進」の経路を追従する。1) の通路分類器から「三叉路」の分類結果が得られた場合、2) のモジュールは次に「右折」の目標方向を出力し、3) のモジュールは「右折」の経路を追従する。

## 2.1 カメラ画像と目標方向に基づく経路追従

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について述べる。このモジュールは、地図を用いたルールベース制御器の出力を模倣するように学習して、カメラ画像に基づいた経路追従行動を獲得する。学習後、外界センサを RGB カメラのみとして、経路追従が可能という特長がある。

Fig.3 に経路追従モジュールのシステムを示す。学習時は、2D-LiDAR やオドメトリ、事前に作成した幾何学的な地図に基づいたルールベース制御器 (ROS Navigation Stack[3]) によって、設定した経路を走行する。その際、入力をカメラ画像、目標方向、出力をヨー方向の角速度 (地図を用いたルールベース制御器が出力) として、0.2 秒周期でデータセットに加える。さらに、バッチサイズを 8 として教師データをデータセットから抽出し、0.2 秒の周期でオンラインで学習する。

このデータセットにデータを追加、データの抽出、学習の 1 連の流れを 1 ステップとする。ここで目標方向は地図に基づいたルールベースの制御器から出力される信号を用いる。データセットの収集には、藤原ら [2] が提案したデータセットの不均衡の改善、学習時における積極的な蛇行といった最も成功率の高い手法を用いる。

学習後、カメラ画像と後述するシナリオを解釈し、変換するモジュールから目標方向を与え、出力されるヨー方向の角速度によりロボットを制御する。

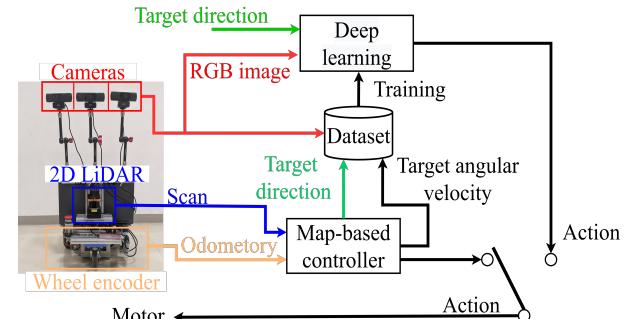


Fig.3 Imitation learning system with target direction

## 2.2 シナリオから目標方向への変換

目的地まで到達するために必要な目標方向を出力するモジュールについて述べる。このモジュールは入力されたシナリオから「突き当たりまで」という「条件」や「直進」などの「行動」を解釈し、単語で構成された経路を目標方向へ変換して出力する。後述する分類器と組み合わせることで、幾何学的な地図を用いずに、目的地までの自律移動に必要な目標方向を、経路追従モジュールへ与えることを可能とする。

「3つ目の三叉路まで直進。右折。突き当たりまで直進。停止」というシナリオを目標方向として変換する例を述べる。このモジュールでは、シナリオを句点ごとに分解後、「条件」と「行動」を示す言葉を抽出し、以下の項目に分けて登録する。

- 1) 通路の特徴
- 2) 順番
- 3) 方向
- 4) 行動

シナリオの例は句点ごとに3つ目の三叉路まで直進/右折/ 突き当たりまで直進/ 停止/ と区切られる。1つ目の区切りでは

1) 通路の特徴：三叉路

2) 順番：3

4) 行動：直進

2つ目の区切りでは

4) 行動：右折

が登録される。この一連の作業を末尾の区切りである「停止」が登録されるまで行う。ここで登録される「行動」をTable 1に示すワンホットベクトルで表現し、目標方向として、前述の経路追従モジュールへ与える。

Table 1 Target direction and data for imitation learning

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

### 2.3 通路の特徴を分類する通路分類器

シナリオで用いる「条件」の達成を判別するために必要な通路の特徴をカメラ画像と深層学習によって分類する通路分類器について述べる。このモジュールは通路の特徴をRGBカメラのみで検出可能という特長を持っている。通路分類器の概要をFig.4に示す。通路分類器はフレーム数16の一連の連続した画像データ（画像サイズは $64 \times 48$ ）を入力とし、通路の特徴の分類を出力する。通路の特徴の分類は島田らの手法[4]に倣い、Fig.5に示した8つとする。通路分類器のネットワークアーキテクチャはDhaiyatらが提案するCNNとLSTMを組み合わせたLRCN[5]を参考として構築した。なおシステムでは、CNNアーキテクチャにMobileNetV3-Large[6]を用いる。

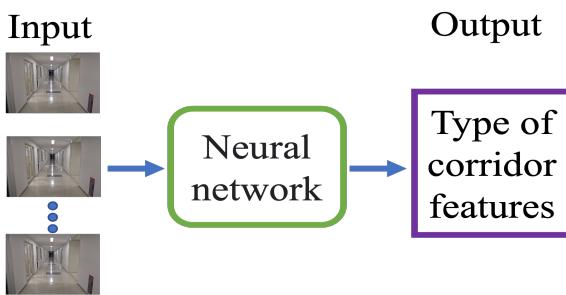


Fig.4 Types of corridor features classifier overview

次に分類器のデータセットの作成について述べる。地図に基づいたルールベース制御器によって経路を走行し、その際、フレーム数16の一連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。その際、分類ラベルのアニメーションは、地図上で対応する分類ラベルを生成する機能を追加した地図に基づいたルールベース制御器の出力を用いて、自動的に行う。

学習に用いるデータセット内で、各クラスのデータ数が大きく異なる不均衡データは、分類に大きな影響

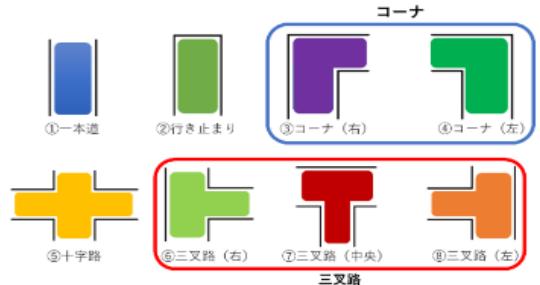


Fig.5 Types of corridor features

を与える[7]とされている。そのため、本稿では学習する際に、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[8]を導入している。

## 3. 実験

実ロボットを用いて、提案するカメラ画像とシナリオに基づく経路追従手法により、ロボットが目的地へ到達可能であるか検証する。

### 3.1 実験装置

実験にはFig.6で示すカメラを3つ搭載したロボットを使用する。

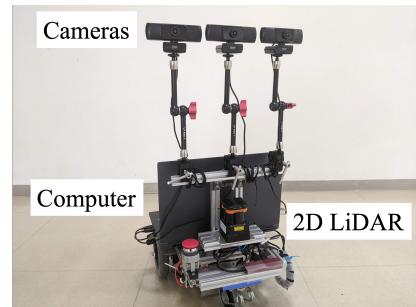


Fig.6 Experimental setup

### 3.2 実験方法

実験環境としてFig.7で示した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。経路はFig.8で示したaからnを順番に走行する。

実験では島田らが用いた50例のシナリオの中から、Fig.7に示したエリアを対象として抽出した7例を用いる。シナリオの抽出において、1. 地図を用いたルールベース制御器で通行が困難な場所が含まれるもの。2. その場で「右を向く」といった経路追従モジュールが達成困難なものを除外している。

まずははじめに通路分類器の訓練を行う。前述の経路を地図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。それぞれのデータセットを構成する一連の連続した画像データの組数は訓練データでは5781組、テストデータでは2902組である。訓練はバッチサイズを32として、30epoch行った。訓練の結果、テストデータに対するAccuracyは0.98となった。

次に経路追従モジュールの訓練を行う。通路分類器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となつた。

2つのモジュールを訓練後、抽出した7例のうちの1例のシナリオをシステムに入力する。この入力したシナリオのスタート地点へシナリオに基づいた向きでロボットを配置し、実験を開始する。なお、経路から外れるといった要因で走行の継続が困難になった場合でも即時失敗とせず、失敗箇所を記録しながら、人が介入し、実験を継続する。Fig.9に実験で用いた7例のシナリオを示す。

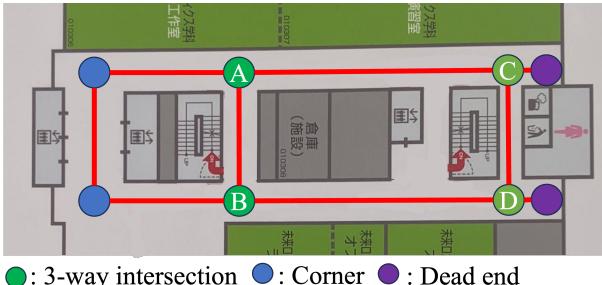


Fig.7 Experimental environment

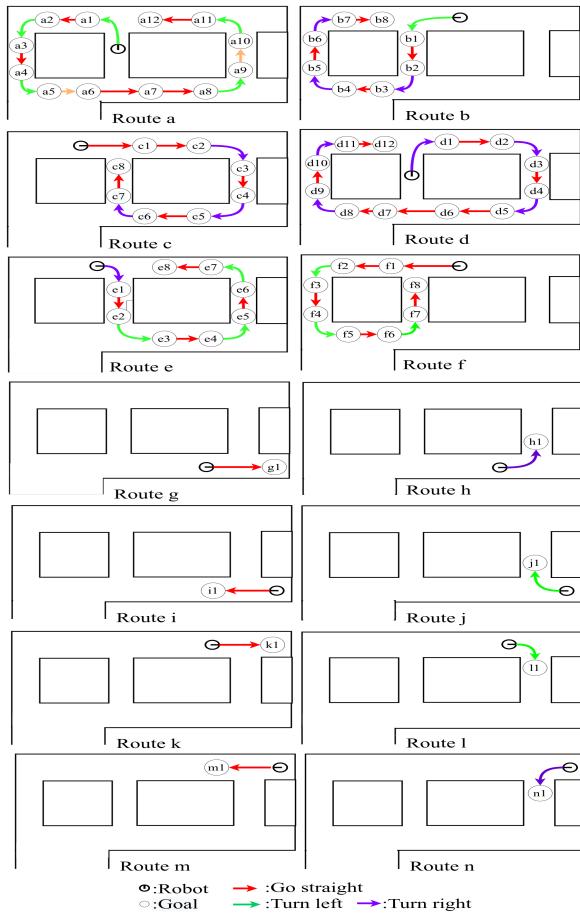


Fig.8 Route for the experiment

### 3.3 実験結果

Fig.10にFig.9(f)における経路追従の様子を示す。結果として、抽出した7例のうち、すべての例でロボットが人間の介入なしで、指定された経路を追従して目的地へ到達した。以上の結果から、提案するカメラ画

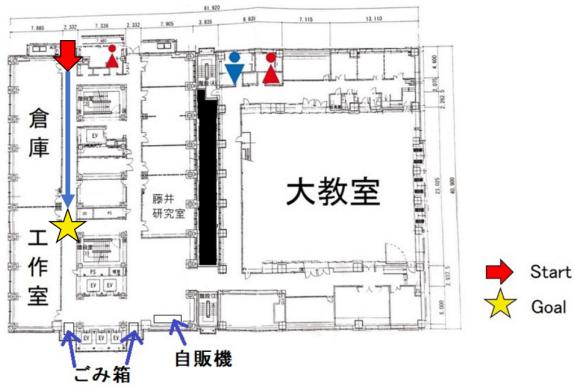
像とシナリオに基づいて、経路を追従して目的地まで自律移動する手法の有効性が確認された。

### 4. 結言

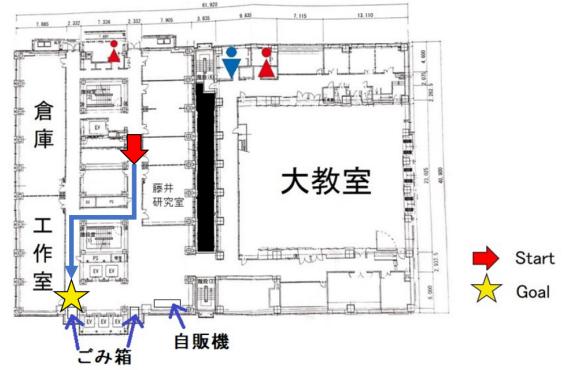
本稿では、事前に作成した幾何学的な地図を用いず、カメラ画像とシナリオに基づいて経路を追従して目的地まで自律移動する手法を提案した。そして、実ロボットを用いた実験により提案手法の有効性を検証した。

### 参考文献

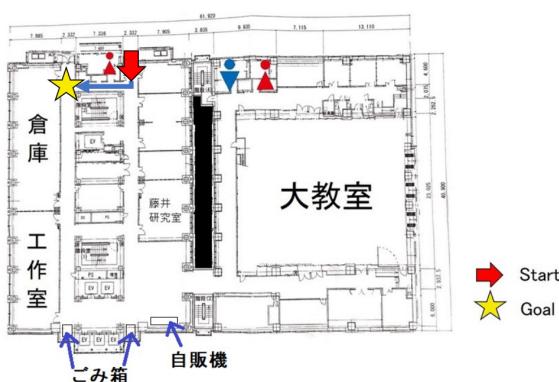
- [1] 春山 健太 et al.: “視覚と行動のend-to-end学習により経路追従行動をオンラインで模倣する手法の提案”. ロボティクス・メカトロニクス講演会講演概要集 2022, 2P2-L05, (2022). doi: 10.1299/jsmermd.2022.2P2-L05.
- [2] 藤原征, 春山健太, 馬場琉生, 上田隆一, 林原靖男: “視覚と行動のend-to-end学習により経路追従行動をオンラインで模倣する手法の提案 - 実環境における経路選択機能の検証と学習時間の短縮化の検討 - ”. 日本機械学会ロボティクス・メカトロニクス講演会'23 予稿集, 2P2-G06, (2023).
- [3] ROS Navigation Stack: *ROS Navigation Stack*. Accessed on 25.10.2023. URL: <https://github.com/ros-planning/navigation>.
- [4] 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, 1H2-04, (2020).
- [5] Dhavat Bhatt et al.: “Have i reached the intersection: A deep learning-based approach for intersection detection from monocular cameras”. *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2017, pp. 4495–4500. doi: 10.1109/IROS.2017.8206317.
- [6] Andrew Howard et al.: *Searching for MobileNetV3*. (2019). arXiv: 1905.02244 [cs.CV].
- [7] Harsurinder Kaur, Husanbir Singh Pannu, and Avleen Kaur Malhi: “A Systematic Review on Imbalanced Data Challenges in Machine Learning: Applications and Solutions”. *ACM Comput. Surv.* 52.4, (Aug. 2019). ISSN: 0360-0300. doi: 10.1145/3343440. URL: <https://doi.org/10.1145/3343440>.
- [8] Jie Xu et al.: “Fault Diagnosis on Imbalanced Data Using an Adaptive Cost-sensitive Multiscale Attention Network”. *2021 International Conference on Intelligent Technology and Embedded Systems (ICITES)*. 2021, pp. 77–82. doi: 10.1109/ICITES53477.2021.9637101.



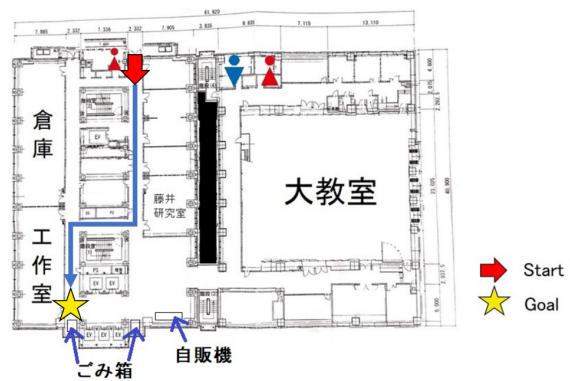
(a) Scenario 01



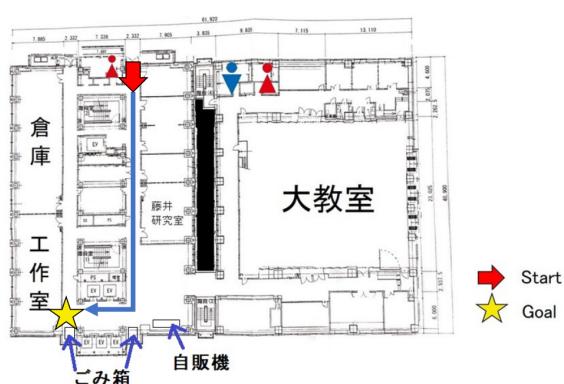
(b) Scenario 02



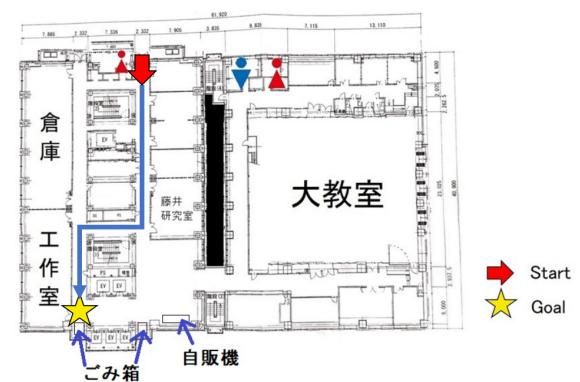
(c) Scenario 03



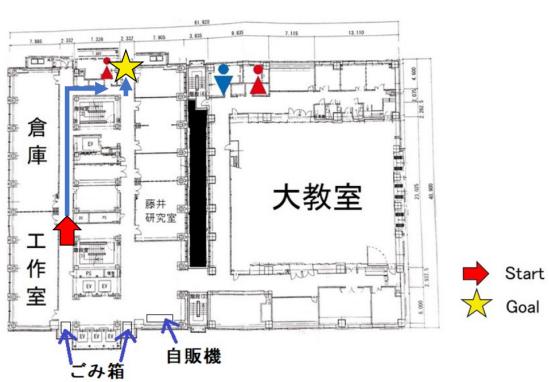
(d) Scenario 04



(e) Scenario 05



(f) Scenario 06



(g) Scenario 07

Fig.9 Example of the scenario used in the experiment



(a) First 3-way : Go straight



(b) Second 3-way : Go straight



(c) Third 3-way : Turn right



(d) End : Turn left



(e) End : Stop

Fig.10 Behavior of path following in scenario