

自律ロボットの目標物認識能力向上

奈良女子大学 生活環境学部 情報衣環境学科
生活情報通信科学コース 4 回生 新出研究室
16480309 善木美菜子

2020 年 2 月

目次

1	はじめに	1
2	目標物認識機構	1
2.1	YOLOv3 について	1
2.2	YOLOv3 を用いた目標物の学習	2
3	ロボット制御	5
3.1	Jason	6
3.2	ROS	6
4	実装	7
4.1	実装方針	7
4.2	物体認識機構の実装	8
4.3	基本行為の改良	8
4.4	再探索プランの実装	9
4.5	矩形サイズを用いた目標物接近プラン	9
5	実験	10
5.1	基本行為性能比較実験	11
5.2	逆向き回転による再探索検証実験	12
5.3	矩形サイズによる目標物接近実験結果	14
6	まとめ	14
7	謝辞	15

概要

自らの動作環境を認識しながら目標達成に向けてプランを選択し行動する自律ロボットにおいて、外環境を正しく認識する能力は必要不可欠である。従来研究で実装されていた物体認識機構では、紙に印刷されたものでなければ正確に認識できないという制限があった。本研究ではこれまでと異なるアルゴリズムとしてYOLOv3を採用することで、より高精度な物体認識能力を備えた実用的な自律ロボットを実現した。また、これまでに実装されていた探索行為の改良や、新たな物体探索プランの追加も行った。

1 はじめに

近年、災害の現場から一般家庭に至るまで、様々な場面で人間の立会いを伴わず自らで目標達成に向けたプランを選択しながら行動できるロボットの必要性が高まっている。こうしたロボットは人間から指示や判断が常に与えられるわけではないため、自らの行動する実世界の環境変化を認識し、それに合わせて目標達成のための最適な行動選択を都度行う必要がある。

当研究室では 2011 年から、この要求を達成するための知的機構として人間の行為選択モデルを模した BDI を採用し、ロボットに取り付けたセンサから得る外環境情報と組み合わせて目標物到達行動を行う自律ロボットの研究を行っている。これまでに失敗に対する回復行動 [1][2][3] や地図ベースによる探索 [4] など様々な研究がなされてきたが、その中でもロボットの視覚情報から目標物体を認識する物体認識機構については、2017 年度の研究 [5] で実装された OpenCV による学習器と深層学習を用いたクラス分類器を複合させる方法によって、それまでよりも精度の高い物体認識能力の獲得に成功した。

しかしながら、この物体認識機構によって正確な到達が可能になったのは紙に印刷された目標物のみであり、形状が立体になると認識精度が低下し到達できなくなる課題や、印刷面が隠されると目標物が認識可能圏内にあっても到達できないなど、印刷物しか認識できないことに起因する課題が残されていた。

そこで、本研究では自律ロボットの物体認識機構にこれまでと異なる物体検出アルゴリズムとして YOLOv3 を採用し、これまで不可能であった立体目標物への正確な到達を可能にすることで、より実用的かつ高精度な探索行為が行える自律ロボットの実現を目指した。また、先行研究で実装された探索方法の改良や、新しい物体認識能力を用いた探索プランの実装も行った。

2 目標物認識機構

本節では、物体検出アルゴリズムとして使用した YOLOv3 について記述する。

2.1 YOLOv3 について

YOLO(You Only Look Once) は、米ワシントン大学に在籍する Joseph Chet Redmon 氏によって開発された物体検出アルゴリズムである。物体検出を回帰問題ととらえて処理しており、物体領域とクラス確率の推定を単一のネットワークで行う end-to-end の学習方法によって、高速で高精度な物体認識を実現している。また、画像全体を見て学習や推測を行うため他手法に比べて背景を誤検出することが少ない、学習結果の汎用性に長けているなどの特徴も備えている。

本研究で用いた YOLOv3[6] は YOLO シリーズの中でも最新のバージョン（2020 年 1 月時点）であり、53 層の畳み込み層をもつ特徴抽出器を用いることで前身の YOLO、YOLOv2 よりも高精度な物体認識が可能である。また、クラス予測の際に一般的に使われる softmax 関数ではなく独立したロジスティック分類器を用いることで、検出した物体が複数のクラスに当てはまる場合に複数のラベルを持つ bounding box の生成を可能にするなどの特長もある。

扱う際には同氏の開発した深層学習フレームワークの Darknet を用いることを想定して設計されているため、本研究でも深層学習フレームワークには Darknet を使用した。

2.2 YOLOv3 を用いた目標物の学習

YOLOv3 を自律ロボットの物体認識機構として実装する前に、目標物として使用する物体の画像学習と、完成したモデルの評価実験を行った。なお、YOLOv3 には既存の学習済みモデルが用意されているが、今回は新たに独自データセットによる学習モデルを作成した。これにより、ロボットが目標とする物体に特化したモデルを用いた探索が可能になるほか、今後の研究で新たな物体を認識する必要が出た場合もデータセットに物体の画像を追加して学習させるだけで目的に合ったモデルの作成ができる。

今回は自律ロボットの探索実験で使用する目標物体として、異なる 2 種類の物体 A と B を学習させた。それぞれの実態は図 1 のとおりである。

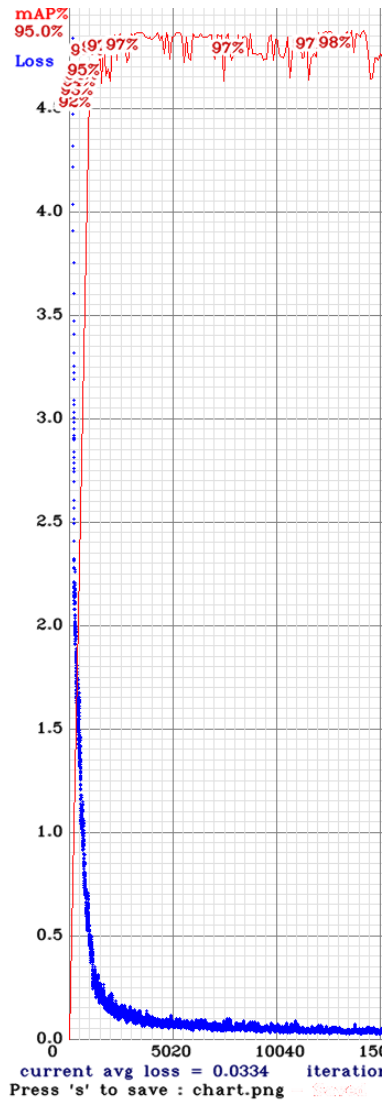
図 1 目標物として使用する 2 種類の物体。左が物体 A、右が物体 B



なお、物体 B については、後ろから見ると顔の模様が確認できず、普通の球体との区別がつかない。そこで、物体 B については顔の模様が確認できた場合のみ認識するように学習を行わせるため、後ろから見た画像は除いてデータセットの作成を行った。

学習を行うにあたり、まず両方の物体について約 100 枚ずつ画像を収集して学習を行った。図 2 はこのデータセットで学習した際の Loss と mAP(mean Average Precision) の遷移を表したグラフである。学習はバッチサイズ=32、サブバッチサイズ=16、画像の大きさは 416×416 をデフォルトとして行い、Loss の数値が 0.03 になったところで十分収束したと判断して学習を終了させた。このとき iteration はおよそ 15000、mAP は 95.0% であった。

図2 最初に作成したデータセットでの学習時 Loss・mAP 遷移グラフ



作成された重みファイルを用いて物体検出実験を行ったところ、形状が単純でポーズや表情などの差分がない B についてはこの段階ですでに誤検知が少なく、高い認識精度を發揮できた。しかし、ポーズや表情に幅があり人型に近い A は B に比べると誤検知が多く、距離が離れると認識精度が下がった。また、両物体に言えることとして、光の当たり方やカメラを設置する場所により画像全体の彩度が変わった場合、検出率が下がることが挙げられた。

そこで、まず認識精度が低かった A については、特に精度の低かった横向きや斜めの角度から見た画像を中心に学習データを増やした。また、距離が離れると認識精度が落ちることについても、データセット内に物体 A を遠方から写した画像が少なかったことが原因であるとわかったため、併せて画像の追加を行った。

さらに、ここまで揃った A と B 両方のデータ画像について Data Augmentation を行い、画像の明度や彩度を調整したり、画像全体にノイズを入れたりしたものを 1 枚の画像につき 8 種類用意した。最終的に物体 A については約 1500 枚、物体 B については約 1000 枚のデータセットが完成した。

このデータセットを学習させた際の Loss と mAP の遷移グラフを図 3 に示す。学習は最初と同様に、バツ

チサイズ=32、サブバッチサイズ=16、画像の大きさは 416 × 416 をデフォルトとしている。Loss の数値が 0.05 前後になったところで十分収束したと判断し、学習を終了させた。iteration は 15000、mAP は 96.6% であった。

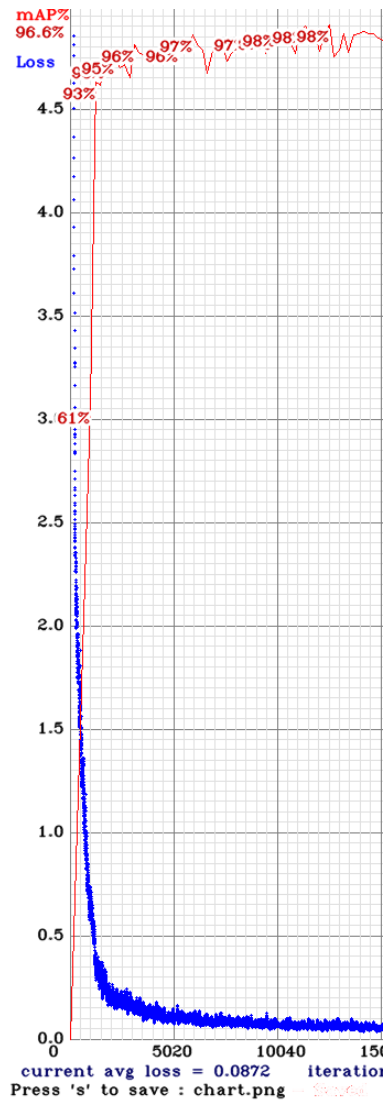


図3 完成したデータセットでの学習時 Loss・mAP 遷移グラフ

この学習によって完成した重みファイルを用いて静止状態での物体検出実験を行った際の結果を表1～表3に示す。実験はAのみが存在する場合、Bのみが存在する場合、両方が存在する場合の3つの状況を想定し、さらにそれぞれの状況について物体とカメラとの距離を変更した静止画像を用意して行った。また、表の確率は物体検出時に示されるクラス確率を表している。

表 1 物体 A のみが存在するとき

	正面	右向き	左向き	後ろ向き
0.3m	100%	100%	100%	100%
0.5m	100%	100%	100%	100%
0.8m	99%	99%	100%	98%
1.0m	99%	99%	100%	90%
1.3m	99%	97%	99%	90%

表 2 物体 B のみが存在するとき

	正面	右向き	左向き	後ろ向き
0.3m	100%	97%	100%	検出せず
0.5m	100%	94%	100%	検出せず
0.8m	99%	100%	99%	検出せず
1.0m	99%	99%	100%	検出せず
1.3m	99%	99%	100%	検出せず

表 3 両方の物体が存在するとき

A \ B	0.7m	1.0m	1.3m
0.7m	100%	100%	99%
1.0m	100%	98%	99%
1.3m	100%	98%	100%

A、B がそれぞれ単独でカメラの視野内に存在する場合も、両方の物体が視野内に存在する場合も、認識精度は高いものとなった。これにより、YOLOv3 による物体認識によって、先行研究の課題であった立体目標物の正確な認識は達成されたといえる。

また、物体 B に関しては、認識しないものとして学習させた後ろからの角度では認識せず、ほかの方向からは認識が可能であることから、ある特徴を有する・有さないの判断も可能であることがわかる。

以上の結果から、YOLOv3 は高精度な物体認識かつ複数物体の同時認識も可能なアルゴリズムであり、様々な状況が想定される自律ロボットの探索行動時に用いる物体認識機構として適していると検証された。

3 ロボット制御

本研究では、先行研究に引き続き自律ロボットの思考部開発に Jason、制御機構開発に ROS を採用した。本節ではこの 2 つについて述べる。

3.1 Jason

Jason[7] は拡張された AgentSpeak のインタプリタであり、BDI エージェントの構築プラットフォームである。

BDI エージェントとは Belief(信念)、Desire(願望)、Intention(意図) の3つの心理状態を使用して論理的な意志選択を行うエージェントであり、信念と願望から導き出した意図の実行後、変化した外環境情報をもとに信念と願望を更新して意図を生成する、というサイクルを繰り返しながら最終目標の達成を図る。

エージェントが外環境変化を知覚するには Java による外部環境の実装が必要になる。外部環境の実装をすることで環境の知覚だけでなく外環境への行為実行も可能となるため、1つの目標に対する達成プランの選択肢拡大や目標の複数保持など、目標達成にむけて柔軟にプラン選択を行う自律ロボットの実現が可能になる。

3.2 ROS

ROS(Robot Operating System)[8] は、オープンソースで提供されているロボット開発用ソフトウェアフレームワークの1つである。

ROS にはプログラムを動かす上で最小単位となる「パッケージ」があり、このパッケージ内部に存在する「ノード」が一般的なプログラムと同等の役割を果たす。ノードはそれぞれ独立で処理を行いつつ、実行されている他のノードと情報をやりとりしながらロボットの制御を行うことができる。

ロボットの制御に関するプログラムをノードとして分散開発・実装できるため、機能の修正や追加を行う際に必要な箇所だけ修正したり、新しくノードを作成することで機能追加の要求に応じることができる特長がある。

本研究では先行研究で用いられたパッケージに加え、YOLOv3 の物体認識結果を ROS ノードとして配信するノードを持つパッケージ darknet_ros を使用する。darknet_ros の詳細は 4.2 で記述する。

4 実装

4.1 実装方針

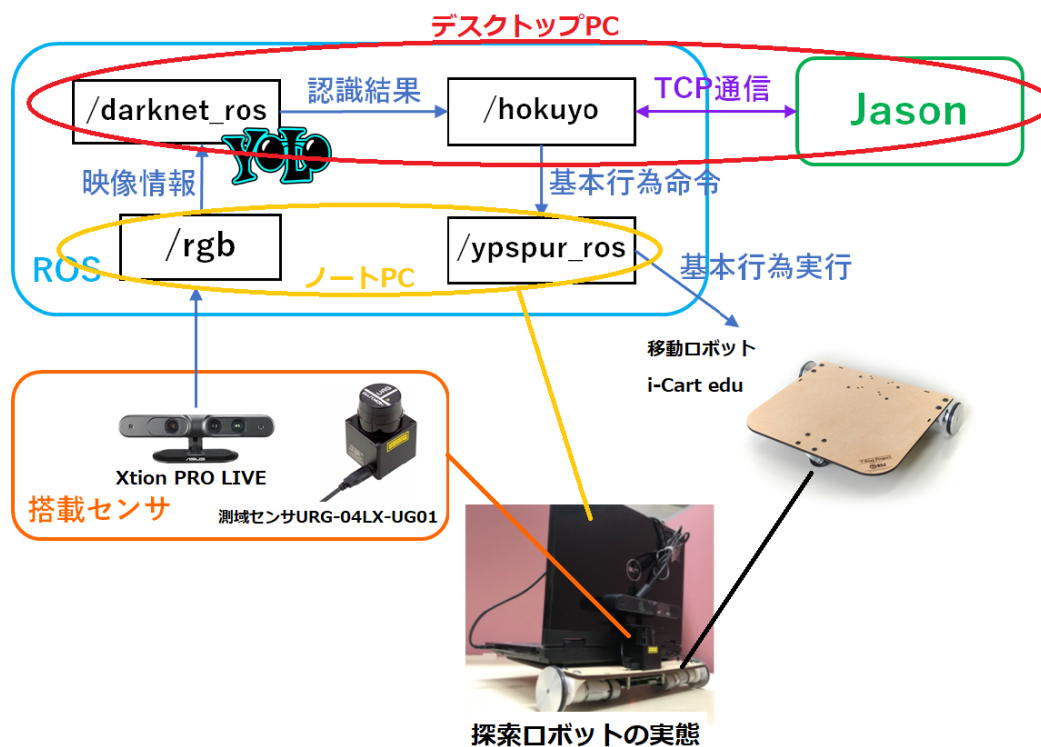


図4 本研究の設計

今回実装した自律探索ロボットは図4の設計で実現した。

Jason および `darknet_ros` と `hokuyo` という2つのROSノードはデスクトップPCで起動し、`rgb` と `yjspur_ros` という2つのROSノードについては移動ロボットに搭載したノートPCで起動している。ROSノード間の通信は2台のPCを無線ネットワークで接続することで可能にしている。

移動ロボットには先行研究で使用された地図ベース走行ロボット i-Cart edu[9] を使用する。また、このロボットには Xtion PRO LIVE[10] と URG-04LX-UG01[11] という2つのセンサーを搭載しており、Xtion PRO LIVE はロボットの視覚情報をROSへ配信するカメラとして、URG-04LX-UG01 はロボットが走行する環境の地図作成を行う測域センサとして使用している。また、センサーはどちらもノートPCへUSB接続することで駆動と取得した情報の伝達を行っている。

先行研究で実装された物体探索方法は次の通りである。

まず URG-04LX-UG01 を用いてロボットが動作する環境の地図を作成する。ロボットは目標物の大まかな位置を信念として持っており、作っておいた地図をもとにその位置まで大域移動をしてから、学習しておいた目標物の情報を使って移動地点の周囲に目標物が存在するか探すというプランで行動する。

この探索方法および「基本行為」と「基本行為を用いた目標物到達プラン」にわけて設計する手法に関して、先行研究 [4] に準拠して本研究でも使用することとする。

4.2 物体認識機構の実装

本研究では、YOLOv3 の検出結果を ROS ノードとして配信することで自律ロボットの探索行動に活用できるようにした。

検出結果を ROS に配信する仕組みには darknet_ros[12] を使用した。darknet_ros は YOLOv3 による物体認識結果を ROS ノードとして配信できるように開発された ROS パッケージであり、これによって検出結果を次のようなメッセージ型のノードで配信できる。

```
Header header
Header image_header
BoundingBox[] bounding_boxes
```

BoundingBox[] の中身である bounding_boxes のメッセージ型は次の通りである。

```
float64 probability
int64 xmin
int64 ymin
int64 xmax
int64 ymax
int16 id
string Class
```

配信されたメッセージは Jason と情報をやりとりする ROS ノード hokuyo で受け取り、メッセージのうち bounding_boxes 内の Class 情報を参照して、Jason が目標としている物体の名前と同じものが検出された中に存在するかどうかを調べることで、目標物を発見できたか判定する。

4.3 基本行為の改良

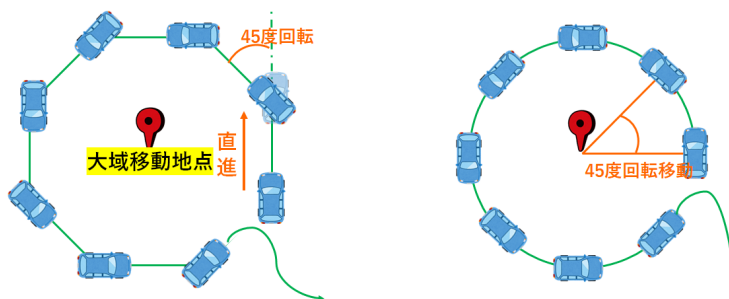


図5 左が従来実験で実装された基本行為、右が今回実装する基本行為

移動ロボットには、BDI エージェントが探索中にプランの中で使用できる手段として、単純に動いたり回転したりといった行動を「基本行為」と称して設計し、実際に使用する際には別プロセスで呼び出して実行するように実装されている。先行研究では、「回転と直進を組み合わせればあらゆる移動が実現可能になる」という観点から、ロボットの体を 45° 回転させる rotate と、あらかじめ設定された距離だけ移動する move という 2 種類の行動が実装されていた。

しかし、現状としてロボットが基本行為プログラムを呼び出すのは大域移動後の探索時のみであり、この時求められる「大域移動地点の周囲を一周する」という動きを実現するのに「回転」→「移動」という 2 つの動作を繰り返す移動方法は時間効率が悪く、実際に探索実験を行った際、目標到達までにかかる時間を引き延ばす原因となっていることがわかった。

そこで、本研究では基本行為を改良し、前方への移動と回転を 1 つのプログラムで実行できるよう新たな基本行為プログラム turn を実装した。turn では図 5 右のように大域移動地点を中心とする半径 0.5m の円を描くように移動する。

4.4 再探索プランの実装

従来の物体認識機構では目標の印刷面が隠されてしまうと到達行動が不可能となっていたが、YOLOv3 によって目標物を様々な角度から認識できるようになったことで、障害物などによって一方向から見ただけでは目標物が認識できなかったとしても、別の方向から探索を行えば目標到達が可能になる状況があると考えられる。

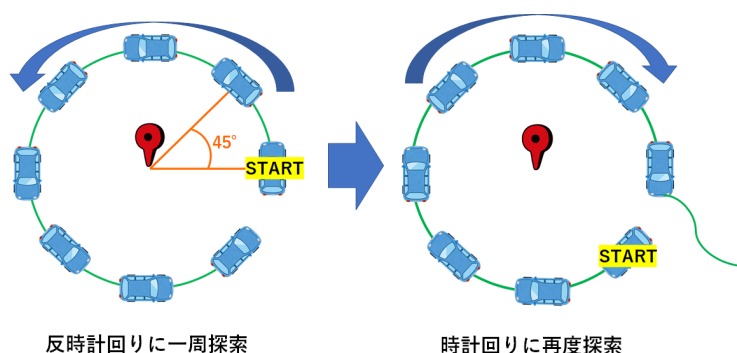


図 6 実装した再探索プラン

そこで今回はこの仮説を検証するために、目標物体が見つけれなかった場合は逆向きの回転で再度範囲内の探索を行うプランを実装した。具体的には、ロボットを 180° その場で回転させる基本行為 reverse と、1 周目とは逆の回転方向で半径 0.5m の円弧を描いて移動する基本行為 return を作成し、図 6 のように 1 周目で目標物体が見つけれなかった場合にはこの 2 つのプログラムで反対方向に回転しながら同じ範囲を再度探索するプランである。

4.5 矩形サイズを用いた目標物接近プラン

先行研究で実装されていた探索プランは、目標物を発見後即座に探索を終了するように設計されていたため、ロボットと目標物の間に距離があっても発見した時点で「到達した」と見なされていた。

本研究では、より目標物体に近づいた状態で到達行動を終了できるように、YOLOv3 から得られる情報を用いた方法として、最初に目標物を認識した際に描画される矩形サイズが一定値より小さければ直進して目標物に接近するプランを実装した。

到達の基準として、今回は目標物体の半径 0.5m 以内にロボットが存在する状態を到達状態と見なすこととする。

本研究で目標物として用いた物体 A、物体 B はどちらもおよそ 15cm 四方の物体であり、0.5m 離れた位置では矩形サイズはおよそ 35000、1.0m 離れた状態ではおよそ 9000 となる。なお、この矩形サイズは YOLOv3 による認識時に ROS へ配信される情報のうち、x 座標と y 座標の最小値と最大値から求まる値である。

このことから、目標物体発見時の矩形サイズが 10000 以下の場合は 1.0m の直進移動、10000 以上 30000 以下の場合は 0.5m の直進移動を行うこととした。

5 実験

今回は自律ロボットの物体認識機構として YOLOv3 を実装した上で、基本行為の性能比較実験、探索失敗時の再探索検証実験、矩形サイズを用いた目標物接近実験の 3 種類を実験した。

これらの実験に際して、今回の実験環境では無線ネットワークの通信量制限により、探索に使用するすべてのプログラムを実行した状態で一連の行動を行うと通信遅延問題が発生する事象が確認された。今回は実装した基本行為と探索プランの性能を測るのが目的であるため、通信遅延緩和の策としてロボットが目標物を発見出来たか判定する時のみ YOLOv3 を手動で起動させ、判定が終わり次第終了させる行程を追加しているが、全てのプログラムを実行しながら一連の行動を実行できるような実装方法を考えることも今後取り組むべき問題の 1 つである。

5.1 基本行為性能比較実験

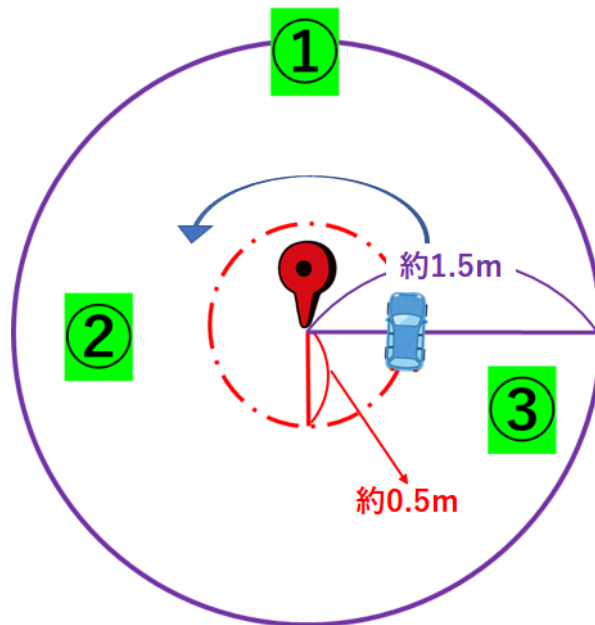


図7 目標物体の位置

ここでは、基本行為の性能比較実験の結果を記述する。

rotate と move の動きを同時に実行するプログラム turn の評価をするにあたり、図7に番号で示した位置に目標物体を設置し、それぞれの場合において探索にかかる時間の計測を1回ずつ行った。なお、この実験では目標物体を隠したり、ロボットの視界を遮ったりするような障害物は探索範囲内に存在しないこととする。実験結果は表4の通りである。

表4 基本行為の性能比較実験

	rotate+move	turn
1	1回/20.22s	1回/9.93s
2	3回/53.50s	3回/24.46s
3	7回/122.78s	7回/55.99s

目標物到達までに実行されるプログラムの回数はどちらの方法でも等しいことから、それぞれのプログラムで実現される行動は同じであることがわかる。しかし、turn を使用した場合には1回の行動達成にかかる時間が従来の方法より約10秒短縮された。また、位置3のように目標物がロボットの探索開始位置の真後ろなど、到達に時間がかかる位置に存在する場合は1分以上の差が出る結果となった。

実験結果より、従来の基本行為を1つのプログラムで実行できるように加えた改良は探索時間の短縮に効果的であったといえる。

5.2 逆向き回転による再探索検証実験

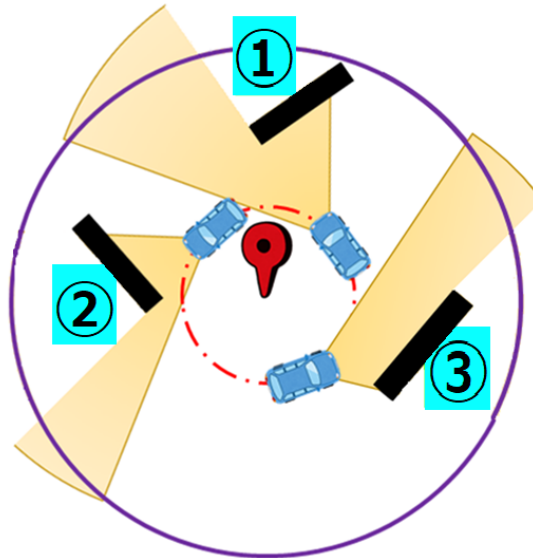


図8 障害物を設置した際の turn による探索図。視界が阻まれ、目標物が認識できない

次に、今回から新たに実装した reverse と return による再探索実験の結果について述べる。実験方法として、図8のように認識可能圏内に存在する目標物が障害物によって隠され、1周目の探索では発見できなかった状況を3パターン想定して目標物と障害物を設置した。ここで、障害物は目標物を一方向から十分隠せる大きさの衝立のような物体を想定している。

図9～図14はそれぞれの場合における探索終了時のロボットの状態とその視界である。なお、写真中の黒い線は本実験とは関係ない。

位置1や位置2のように、ロボットの軌道上に目標物体を視認できる地点が存在するように障害物が置かれた場合は今回設計した再探索方法が有効であった。しかし、位置3のように軌道上の何処からも目標物体が見えないように障害物が設置された場合は再探索でも到達できなかったため、こうした場合にも対処できる別の探索方法を考える必要がある。



図9 位置1の再探索終了状態

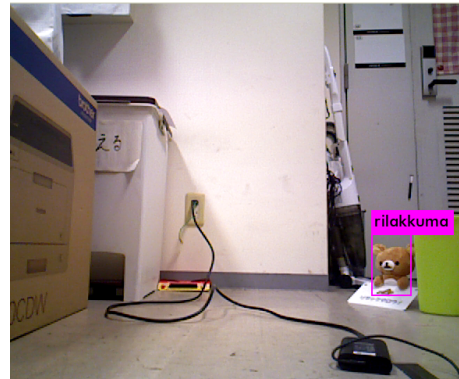


図10 図9でのロボットの視界

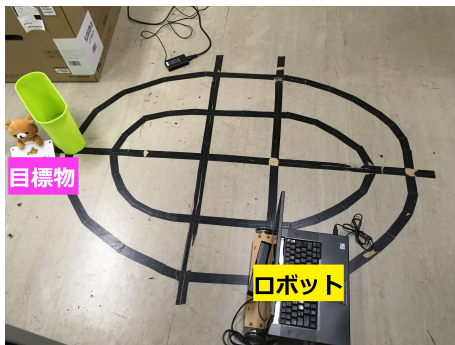


図11 位置2の再探索終了状態



図12 図11でのロボットの視界



図13 位置3の再探索終了状態。目標物は発見出来なかった

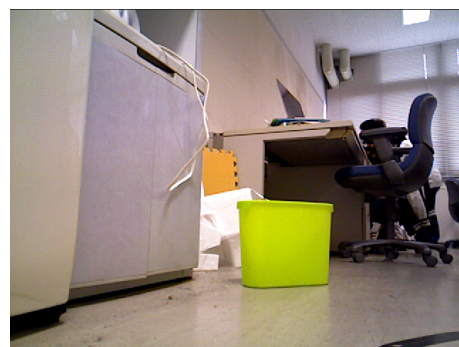


図14 図13でのロボットの視界

5.3 矩形サイズによる目標物接近実験結果

実装した目標物接近プランを検証するために、物体 A、物体 B のそれぞれを目標とした場合において、物体とカメラの距離を 0.3m~1.5m まで 0.3m ずつ変更した実験を各距離につき 3 回行った。

結果は表 5 と表 6 の通りである。なお、各結果に表記した数値は物体発見時の矩形サイズの値である。

表 5 物体 A を目標物としたとき

	1 回目	2 回目	3 回目
0.3m	成功・62700	成功・63333	成功・62472
0.6m	成功・21352	成功・21509	成功・21920
0.9m	成功・13192	成功・14000	成功・13761
1.2m	失敗・11397	成功・7462	失敗・11049
1.5m	成功・6399	成功・6240	成功・5772

表 6 物体 B を目標物としたとき

	1 回目	2 回目	3 回目
0.3m	成功・34830	成功・34314	成功・35370
0.6m	成功・23707	成功・23095	成功・22644
0.9m	成功・11960	成功・10807	成功・10609
1.2m	成功・7221	成功・6800	成功・6715
1.5m	成功・4970	成功・4761	成功・4899

ほとんどの場合において目標物体への接近を正確に実行できたが、物体 A を目標としてカメラとの距離が 1.2m 離れた状態では、矩形サイズが想定されるものより大きく描画され正しい距離の移動が出来ないことがあった。

また、今回実装した目標物接近プランには直進移動のみで目標物体に近づくため、ロボットが目標を常に正面に見ながら移動するような高精度な接近が不可能であることや、目標物体の本来のサイズに応じて、接近が必要か判断する矩形サイズの基準値を設定しなおさなければならないなどの課題も存在する。

より正確な目標物接近行動を行うには、矩形サイズ以外の情報も用いた行動を行う必要があると考えられる。

6 まとめ

YOLOv3 によって、これまで不可能であった立体目標物への正確な到達が可能になった。また、今回改良したロボットの基本行為によって、より効率的な探索行動が実現されたほか、新しく実装した再探索プランによって今回獲得した「目標物をあらゆる方向から認識出来る」能力を活用した探索行動が実現できた。

今後の課題として、目標物体が障害物に完全に隠されてロボットの軌道上からは視認できない場合の探索方法と、より正確な目標物への接近方法を考える必要がある。

前者については現在考えられる方法として、障害物も認識対象として学習させ、再探索でも目標物が見つかる

らない際は障害物の後ろへ回り込んで探索させる手段が挙げられる。これは、人間がものを探するときにも見受けられる行動であり、人間の行動決定を模した BDI モデルによって行動する本研究の自律ロボットにとって、効果的な探索方法であるといえる。ただし、この方法を実現するには障害物にぶつからないよう接近し、後ろに回り込む動作を行う必要があるため、目の前の物体との距離を測る新たなセンサの導入や探索プランの実装が必要になると考えられる。

距離計測センサの導入は目の前の物体との距離が正確に把握できることから、後者の課題解決にも繋がると考えられる。また、到達目標物体だけでなく、ドアや敷居など部屋と部屋との境界線になるものを検出対象として学習したモデルと組み合わせれば、異なる部屋や空間への移動といった新たな環境変化にも対応できる自律ロボットの実現も期待できるため、適切なセンサの選択と実装が望まれる。

また、物体認識機構に関して、他のぬいぐるみなど物体の種類が増えた際にも高精度な認識が行えるかどうかも検討課題である。

7 謝辞

本研究および本論文の執筆にあたり、いつも親身にご指導くださった新出尚之准教授に深く感謝いたします。また、多くのアドバイスを下さった新出研究室の皆様感謝の意を表します。

参考文献

- [1] 石井あすか. 自律ロボットの目標達成に置ける行動の柔軟化について. 2016 年度卒業論文, 奈良女子大学理学部情報科学科, 2017.
- [2] 小谷麻緒. 方位情報を用いた自律ロボットの行為の失敗認識の効率化について. 2016 年度卒業論文, 奈良女子大学理学部情報科学科, 2017.
- [3] 山本千尋. 実世界における行為の失敗の概念を考慮した自律ロボットの实装について. 2016 年度卒業論文, 奈良女子大学理学部情報科学科, 2017.
- [4] 鳥居実桜. 地図ベースの走行を行う自律ロボットの構築. 2018 年度卒業論文, 奈良女子大学生活環境学部情報衣環境学科生活情報通信科学コース, 2019.
- [5] 兼松明未. 自律ロボットの目標物到達機構の精度向上とそれを用いて行動する BDI エージェントの構築. 2017 年度修士論文, 奈良女子大学大学院人間文化研究科, 2018.
- [6] Joseph Redmon and Ali Farhadi. YOLOv3: An Incremental Improvement. *arXiv:1804.02767*, 2018.
- [7] Rafael H. Bordini, Jomi Fred Hübner, and Michael Wooldridge. *Programming multi-agent systems in AgentSpeak using Jason*. WILEY, 2007.
- [8] Open Source Robotics Foundation. ROS. <http://www.ros.org/>.
- [9] T Frog Project. i-Card edu. http://t-frog.com/products/icart_edu/.
- [10] ASUSTeK Computer Inc. Xtion PRO LIVE. https://www.asus.com/jp/3D-Sensor/Xtion_PRO_LIVE/.
- [11] HOKUYO AUTOMATIC CO.LTD. URG-04LX-UG01. <https://www.hokuyo-aut.co.jp/search/single.php?serial=17>.
- [12] Marko Bjelonic. YOLO ROS: Real-time object detection for ROS. https://github.com/leggedrobotics/darknet_ros, 2016–2018.