2017年度 修士論文

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

奈良女子大学 人間文化研究科
博士前期課程 情報科学専攻 新出研究室
16840045 兼松明未

平成30年1月

目 次

概 要

我々は、人間の行為選択を模したエージェントモデルである BDI エージェントを用いて、実世界において自律的に目的を達成するロボットの実現を目指している．そのようなロボットに必要な能力の一つとして、目標物を認識してそこに到達する行動が挙げられる．従来研究では、目標物に関する大量の学習データの収集の困難さなどの制約のもとで、認識率を上げることが課題であった．そこで我々は、学習方法の検討により認識率を上げることを目指し、ディープラーニングを用いて物体を認識する能力の獲得を行うとともに、この能力を BDI エージェントが用いて行為決定を行うためのライブラリの構築を行った．これにより、ロボットが複数の経由地を経て目標物に到達するなどのプランで行動することが可能となることを示す．

1 はじめに

1.1 研究背景

近年、自動お掃除ロボットから災害救助に至るまで、さまざまな分野で自律型のロボットが普及しつつある中で、我々はそのようなロボットの研究を行っている．そのようなロボットに求められるのは、人間同様に複雑かつさまざまな行為を獲得し、周囲の状況に応じて都度最適な行動をとる能力である．それらの中でも「目標物を認識してそこに到達する」という能力は、最も汎用的な基本行為のひとつであると考えられる．そのため、我々はこの能力の獲得を目指してきた. また、ロボットが動的な環境のもとで行為を行うには、周囲の環境を知覚し、自ら思考し、目的を決め、それを達成するための手段を選択し、行動するという過程が必要となる．そこで我々は、ロボットの思考を BDI エージェントを用いて構築し、目標物到達機構の実装を行ってきた．なお、本研究は同研究室 4 回生の小松芙美子との共同研究である．

1.2 本論文の構成

本研究では、従来研究における課題の解決に取り組むことで、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た BDI エージェントの構築を行った．本節では、本論文の構成について述べる．

2 章では、ロボットの目標物認識能力獲得のために行った画像認識について述べる．従来研究では、目標物認識能力が低いことが課題であった．従来の画像認識方法だけでは、しばしば目標物を誤って認識してしまい、途中で目標を見失うことがあった．それにより、目標にたどり着くまでに遠回りをして時間を費やすことや、別の地点に到達してしまうことさえあった．そこで、認識率を上げるために、ディープラーニングによる物体認識法を利用し、画像認識能力の向上を試みた．

3 章では、ロボットエージェントの意思決定を行うために用いた BDI エージェントの構築について述べる．BDI エージェントを利用することによって、論理的な思考を行えるようになることを示す．従来研究から継続して使用しているが、従来研究ではエージェントの柔軟な行為選択の部分が十分実現できていなかった．

4 章では、本研究の開発環境について述べる．特に重要であるエージェント構築のための Jason, ロボット制御のための ROS, 使用ロボットの EV3 については詳細を述べる．

5 章では、目標物到達機構の実装方法について述べる．従来研究の仕様と課題を踏まえて、向上した画像認識能力や新しく導入したロボットを組み込んで新たに構築した目標物到達機構について述べる．

6 章では、実際に行った検証実験について述べる．また、その結果による考察を行う．

最後に、7 章では、まとめと今後の課題について述べる．

2　画像認識法

本節では、ロボットの目標物認識能力を獲得するために行った画像認識の手法について述べる．本研究における物体認識能力は、物体検出器と画像分類器を複合することによって複数物体の同時認識を可能としている．ゆえに、それぞれの学習器について述べたあと、物体認識能力の評価について述べる．

2.1　OpenCV による物体検出

画像認識に用いられる機械学習のアルゴリズムにはいくつかの種類がある [1]．従来研究 [2] では、目標物認識の足がかりとしてオープンソースの画像処理ライブラリである OpenCV を用いて機械学習を行う方法 [3] を採用した．開発言語には、従来研究では Python を用いており、本研究でも継続して Python を使用している． OpenCV による物体検出器の学習方法は、識別する画像の特徴をとらえる特徴量と呼ばれる識別器フィルタと学習サンプルを Adaboost に投入して学習を行うことで、どの識別器を選択すればよいかを決定し、検出物体が正解であるか不正解であるかを振り分けるというものである．学習サンプルとしては、識別する物体の画像（正解画像）とそれを含まない画像（不正解画像）が必要となり、それらを大量に収集しなければならない．特徴量は、Haar-like 特徴・LBP 特徴・HOG 特徴の 3 つから選択することが可能で、本研究では、従来研究から継続して、顔認識などにも広く用いられる Haar-like 特徴量を選択して学習を行っている．Haar-like 特徴量とは、画像の一部分を切り出して、二値画像として処理することで局所的

な明暗差を算出し、それらを組み合わせることによって画像を判別するアルゴリズムである.

2.1.1 従来研究における利用

従来研究 [2] では、2 種類の物体を用意し、それぞれの物体の単体検出器を作成した. サンプル画像として収集した画像の枚数は、正解画像が各物体につき約 1000〜2000 枚、不正解画像が約 1000 枚である. しかし、この物体検出器はどちらも学習した物体以外を認識してしまう誤認識が多く見られ、正解率は約 68.8%程度にとどまったため、認識性能の向上が求められた.

学習器の性能を向上させるために最も単純な方法は、学習に用いるデータ量を増やすことである. しかし、手作業では大量の学習データを収集することが困難な上に、学習に用いるデータ量が増大するにともなって、学習にかかる時間や、学習結果を利用する際の処理時間も大きくなってしまう. そのため、本研究のように動的にロボットを動作させるような利用方法の場合、この方法は現実的ではなかった.

そこで、 認識の際には二値画像として扱われて失った色情報を、認識後に補完することを行った [2]. まず、物体検出器で検出を行うと、図 1 のように、複数の検出結果が得られる. 以前は、これらの中からどれを選択すればよいかが分からず、最大の検出枠を最適な結果としていた. そこで、目標物体の一部を切り出した画像をあらかじめ用意し、検出枠のそれぞれについてカラーヒストグラムを計算して色情報を補完する. これにより、図 2 のように、最も目標物体に近似した色情報を持つ検出枠をしぼりこむことが可能となり、目標物認識能力はある程度向上した. (ここで示した図はイメージであり、実際の実験では検出対象として版権のあるキャラクターなどを使用している. 以後の図でも同様.)

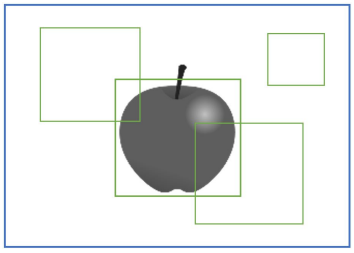

図 1: 物体検出器による検出結果

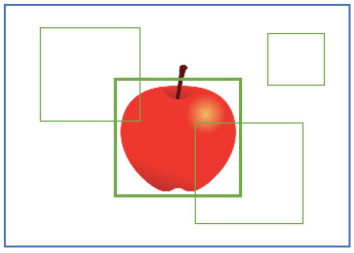

図 2: 色情報の補完後

しかし、いずれにせよ物体認識能力は精度が高いといえるものではなく、さらなる認識能力の向上が求められた. また、先の学習器は単体の物体のみを検出する物体検出器であり、複数物体を同時に認識する能力を獲得すること

も課題であった．さらに、ロボットの動作をよりスムーズにするために、物体認識をより高速に行う必要があった．

2.1.2　本研究における利用

本研究では、複数物体の同時検出を目的とするため、従来研究と同様の手法によって、6 種類の物体を一度に検出可能な学習器を作成した．学習器は、サンプルに使用する画像とその合計枚数を変え、12 種類作成した．その中から成績のよかった 3 つの学習器 A, B, C について、学習に使用したサンプル画像の枚数や学習所要時間、性能を表 1 にまとめた．なお、この表における正解の検出率とは、6 物体いずれかの領域を候補領域として検出した場合の比率を示している．また、誤認識の検出率とは、6 物体以外の領域を候補領域として検出するか、１つの物体に対して 2ヶ所以上の重複した候補領域を検出する場合の比率を示している．

学習器名	A	B	C
正解サンプル数 (枚)	3000	3000	1200
負正解サンプル数 (枚)	4000	2000	1200
学習所要時間	70 時間 8 分	24 時間 35 分	5 時間 45 分
学習器の処理速度 (fps)	2.106	2.623	2.6576
正解の検出率 (%)	97.573	100	99.029
誤認識の検出率 (%)	63.254	64.483	66.667

表 1: 6 物体検出器の学習と利用に関するデータ

表 1 より、サンプル数の合計が多いほど、学習所要時間や学習器の処理速度が大きくなることが分かる．また、いずれの学習器においても正解の検出率は非常に高い値を示している．一方で、誤認識の検出率も相当高いものとなった．これらの学習器による物体検出結果の表示は、図 3 のようになる．

誤認識の検出率が高い要因は、6 物体の検出を同時に学習したため、検出する物体に多様性が生まれたからだと考えられる．しかし、正解の検出率が高いことから、誤認識さえ取り除くことができれば、これらの学習器は有用であるといえるだろう．

2.2　ディープラーニングによる画像分類

目標物認識能力の向上、ならびに、複数物体の同時認識能力の獲得を目指して、我々は新たな学習方法を取り入れることを検討した．その学習方法は、画像認識などさまざまな分野において非常に高精度の性能を発揮し、現在注目が高まっているディープラーニングである [4]．本研究では、画像認識の分

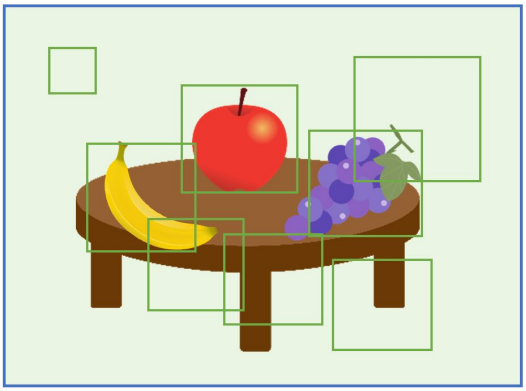

図 3: 6 物体検出器による検出結果の例

野で広く使われる畳み込みニューラルネットワークによる多クラス分類器の学習を行った．

2.2.1 Caffe の利用

Caffe はオープンソースで提供されているディープラーニングのフレームワークである．特徴としては、柔軟なアーキテクチャを保持していることで簡単に動作させることができること．CPU のみならず GPU を利用した高速な計算処理が可能であること．従来研究で使用してきた開発言語の Python を使用して、容易にコードを拡張することができること．などが挙げられる．なお、本研究でも GPU を導入している．NVIDIA 社の GeForce GT730 を使用し、学習速度が CPU のみ (Intel 社の Celeron E300 2.50GHz) での処理の場合に比べ約 4.2 倍となった．

本研究では、Caffe がサポートしている CNN による多クラス分類を利用して、画像の分類を行った．ネットワークの構成は、層の数、ノードの数、ネットワーク構成の記述方法などを Caffe に付属の CIFAR-10 と呼ばれるデータセットを参考にしている．CIFAR-10 は、10 クラスの画像分類を行うデータセットで、2 層構造の畳み込みニューラルネットワークである．ニューラルネットワークは、データを入力するための入力層、特徴を抽出するための中間層、結果を出力するための出力層で構成される．中間層は自由な構成が可能で、畳み込みニューラルネットワークでは、局所的な特徴抽出を行うための畳み込み層、局所的な最大値を計算するためのプーリング層、フィルタ処理のための全結合層が組み合わされている．

学習を行うには、ニューラルネットワークを構成した後、画像データ（学習データとテストデータ）を用意する必要がある．また、学習の結果をふまえて、学習率などの初期のパラメータや層の構成を調整することで、学習器の性能を高めることができる．

2.2.2　6 クラス画像分類器の作成

本研究では、まず物体検出器で検出した領域が 6 種類の物体のどれであるかを分類するために、6 クラス画像分類器を作成した．各クラスオブジェクトは、物体検出器で学習したものと同一ある．画像データは、クラスごとに約 600 枚ずつ収集した．そして、集めた画像からクラスオブジェクトに該当する領域を切り出すと、画像サンプルの枚数は約 900 枚ずつとなった．ここで、画像データの枚数に対してサンプル数が増えるのは、1 枚の画像につき複数のオブジェクトを切り出せる場合があるからだ．さらに、認識精度を高める手法として有効であることが知られている「データ拡張」を行った．データ拡張とは、学習データに反転、回転、拡大・縮小、輝度値の調整などさまざまな加工を施すことによって学習データ量を簡単に増やす手法である．本研究では、1 つのサンプルにつきオリジナル、回転、左右反転、拡大、コントラスト調整、平滑化の 6 パターンのデータ拡張を行った．そのため、全クラスオブジェクトの画像データの合計は 34026 枚となった．さらに、本研究ではこれらの画像サンプルを学習用とテスト用で 6:1 の比率で分けることとした．ゆえに、学習用データは 29170 枚、テストデータは 4856 枚となった．そして、この学習データを用いて画像分類器の学習を行った．なお、各クラスのサンプル画像枚数の内訳は表 2 に示す．

学習器の精度を検証するには、作成したテストデータを用いて画像の分類テストを行う．より高精度の学習器を目指して、学習時のパラメータの調整を行いながら、表 2 の画像サンプルを用いた学習器の作成を繰り返した．それらの中で最も高い精度を上げた学習器の学習と利用に関するデータを表 3 に示す．表 3 における正確性とは、テストデータを用いた分類の正答率を表している．

No.	元データ	切り出し後	拡張後
0	1000	1000	6000
1	605	861	5166
2	600	929	5574
3	800	1025	6150
4	600	913	5478
5	780	943	5658
合計	4385	5671	34026

表 2: クラスごとのサンプル数内訳

学習データ (枚)	29170
テストデータ (枚)	4856
画像サイズ（px）	32×32
学習時間（時間）	4.5
処理速度 (fps)	132.377
正確性 (%)	99.733

表 3: 画像分類器のデータ

この学習器は 132fps の処理速度で、99.7%の正確性を達成したことから、高速かつ高精度の画像分類を実現することができたといえるだろう．

2.3　物体認識への応用

物体を検出する能力と画像を分類する能力を獲得したところで、物体認識への応用を試みた．物体認識は、与えられた画像から物体の位置を特定し、クラス分類を行うことで可能となる．そこでまず、入力画像から候補領域の抽出を行う必要がある．これにディープラーニングを用いる方法もあるが、本研究では 2.1.2 節で作成した OpevCV による 6 物体検出器を利用する．つづいて、この物体検出器から検出される複数の候補領域をクラス分類する必要がある．これには、2.2.2 節の 6 クラス分類器を利用する．そして、これらを組み合わせて物体認識を行うと認識結果の表示は、図 4 のようになる．

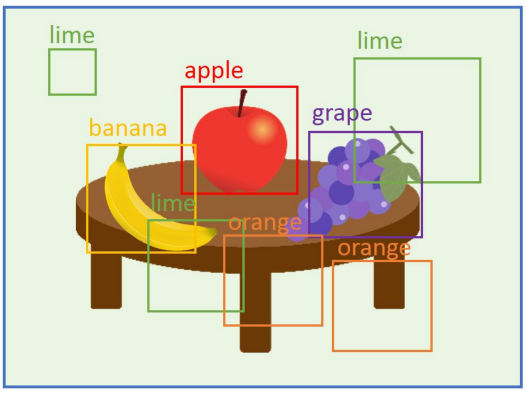

図 4: 6 物体検出と 6 クラス分類

図のように、物体検出器が目標物体として正しく検出した候補領域については画像分類器でも正しくクラス分類が行われるが、一方で、物体検出器が誤って検出した候補領域については、画像分類器でそれを 6 クラスのいずれかに分類してしまうという問題があった．これを解決するには、物体検出器での誤検出を減少させるか、あるいは、画像分類器で誤検出を取り除くといった処理が必要となる．我々は、後者の方法を選択して誤認識の減少に取り組んだ．

2.3.1　7 クラス分類器による誤検出除去

物体検出器の誤った検出結果を取り除くために、我々はこれまでの 6 クラス分類器に、新たなクラスとして誤検出を分類するためのクラスを追加する必要があると考えた．そこで、7 つ目のクラスとして、6 つのクラスオブジェクトのいずれでもない不正解オブジェクトを分類するためのクラスを追加した．

7 つ目のクラスのサンプル画像には、物体検出器で検出された誤認識領域を画像データにして収集した 5400 枚の画像を用いた．2.2.1 節、表 2 の 6 ク

ラスのサンプル画像にこの 7 つ目のクラスのサンプル画像を追加して得られたサンプル画像の合計は、39426 枚となった．これらのサンプル画像を 6 クラス分類器と同様に、学習データ 33799 枚とテストデータ 5627 枚に分けた．そして、ニューラルネットワークの構成も同じにして学習を行った．なお、7 クラス目の画像は、収集した枚数が他のクラスのサンプルにデータ拡張を行った後の枚数と同程度となったため、データ拡張を行っていない．

作成した 7 クラス分類器のクラスごとのサンプル数の内訳と、学習機の学習と利用に関するデータを表 4 と表 5 にまとめた．

No	元データ	切り出し後	拡張数
0	1000	1000	6000
1	605	861	5166
2	600	929	5574
3	800	1025	6150
4	600	913	5478
5	780	943	5658
6	5400	5400	5400
合計	9785	11071	39426

表 4: クラスごとの分類のサンプル数内訳

学習データ (枚)	33799
テストデータ (枚)	5627
画像サイズ（px）	32×32
学習時間（時間）	4.5
処理速度 (fps)	160.090
正確性 (%)	98.503

表 5: 画像分類器のデータ

この 7 クラス画像分類器の分類精度は、98.5%となった．また、6 クラス分類器の場合と比較してみると、分類クラス数および学習データ数が増加しても学習所要時間や処理速度への影響はほとんどみられないことがわかる．これは、ニューラルネットワークの構成が同じであることに起因する．

2.3.2 データ拡張の有用性

これまでの画像分類器の学習では、既に述べたように、学習においてデータ数を増やすために、1 つのサンプルを 6 パターンに増やすデータ拡張を行っている．ここでは、データ拡張の有用性を示す．また、これまではデータ拡張後に画像サンプルを学習用とテスト用に分けており、両者が類似していたことで分類精度が高まったとも考えられる．そこで、ここでは拡張を行った学習データとは別にテストデータ約 210 枚用意して、7 クラス分類の検証を行う．

検証に用いるのは、A, B, C の 3 つの 7 クラス分類器である．１つのサンプルにつき、A の学習器では、データ拡張なしの 1 パターンのサンプルのみを使用する．B の学習器では、回転（回転角度は± 10〜90 度をランダムに選択する）の拡張を行った 2 パターンのサンプルを使用する．また、C の学習器では、これまでと同じ 6 パターンの拡張を行ったサンプルを使用する．表

6 に、クラスごとのデータ数の内訳と学習器の正確性を示す．この表における正確性は、テストデータが正しく分類された比率を表す．

No	オリジナル	A: 拡張なし	B: 1 種類拡張	C: 6 種類拡張
0	1000	1000	2000	6000
1	861	861	1722	5166
2	929	929	1858	5574
3	1025	1025	2050	6150
4	913	913	1826	5478
5	943	943	1886	5658
6	5400	5400	5400	5400
合計	11071	11071	16742	39426
正確性 (%)	ー	81.9	90.7	92.2

表 6: クラスごとのサンプル数内訳と学習器精度

これらの学習器の正確性は、A が 81.9%, B が 90.7%, C が 92.2%となり、拡張パターンが多いほ精度が向上している．以上のことから、データ拡張の有効性が示されたといえる．また、学習データとは別に用意したテストデータに対しても高い正確性を保っていることから、過学習による正解率の上昇ではないと判断できる．

2.3.3 6 物体認識能力の獲得

7 クラス分類器の精度を確認したところで、先の 6 物体検出器とこの 7 クラス画像分類器を組み合わせて物体認識を行った．実行結果画面には、6 クラス目までに分類された場合には認識結果を表示し、7 クラス目に分類された場合には認識結果を表示しないようになっている．また、1 つのオブジェクトに対する重複検出を避けるために、1 クラスにつき複数の認識結果が得られる場合には最も確率の高い結果のみを表示する処理を施た．この実行結果の表示は、5 のようになり、誤認識領域を取り除いて物体を認識できていることが分かる．

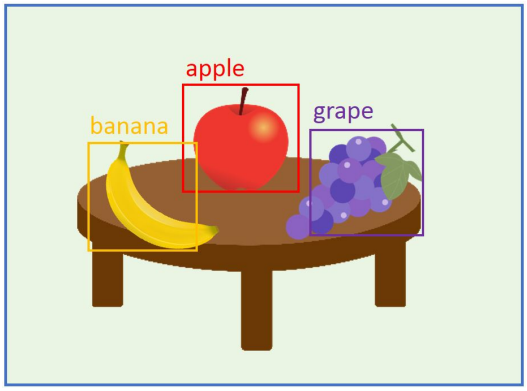

図 5: 6 物体検出と 7 クラス分類

つづいて、物体認識能力の精度と処理速度の検証を行う．この物体認識能力は自律ロボットの視覚情報として機能し、ロボットはこの情報をもとに行動する．そのため、ロボットに搭載したカメラ映像を高速に処理しつづける必要がある．物体認識の高速化を図るには、最も時間を要する処理である物体検出を高速化すればよい．物体検出の高速化には、検出器に入力する画像を小さくすることで実現できる．本研究で使用するロボットに搭載したカメラの解像度は 640×480 であり、入力画像を 1.0 倍, 0.5 倍, 0.4 倍, 0.3 倍と変えて物体認識を行った際の処理速度と精度を表 7 にまとめる．

入力画像倍率	処理速度 (fps)	正確性 (%)
1.0	2.198	84.306
0.5	6.185	86.921
0.4	8.247	85,312
0.3	10.657	77.465

表 7: 物体認識の入力画像倍率の変更による処理速度と正確性

直感的な考えでは、入力画像の倍率が小さいほど処理速度は高速になる一方、正確性は下がるように思われる．しかし、実際には 1.0 倍より 0.5 倍や 0.4 倍の方が正確性が高まった．これは、入力画像が大きいほど物体検出器の誤検出数が多くなるためであると考えられる．つまり、0.5 倍程度の画像を入力して物体検出をする方がよいという結果が得られた．

以上のことから、低倍率画像を利用した 6 物体検出器と、誤検出除去機能を持つ 7 クラス画像分類器を併用した、高速かつ高性能な物体認識能力を獲得することができたといえるだろう．

3　BDI エージェント

BDI エージェントとは、信念 (Belief)、願望 (Desire)、意図 (Intention) の 3 つの心的状態を用いて熟考し、目標達成のためのプランを選択し、意志決定を行うエージェントモデルである．人間の行動決定方式を模倣したモデルであるため、人間同様の柔軟な思考を可能とする．

BDI エージェントでは、信念や願望から最終的に達成したい目標を持つと、いくつかの途中に達成する副目標が生成され、これをサブゴールと呼ぶ．このサブゴールを達成するための手段としてプランを用意しておく．そして、周囲の環境から知覚を得ると、これらのプランの中から状況に応じた最適なプランを選択して行動する．このとき、意図が形成される．選択したプランが失敗した場合も、失敗を回復するためのプランを選び直すことが可能である．こうしてサブゴールを順に達成していくことによって最終目標の達成を目指すものである．

自律型のロボットエージェントの複雑な論理思考を構築するには、BDI エージェントを用いて意思決定を行うことが適していると考えられる．

4　開発環境

開発には、以下のものを使用している．

ソフト	エージェント	Jason	BDI 構築プラットフォーム
	OS	Ubuntu14.04	Debian Linux ベースの OS
		ROS Indigo	ロボット制御用フレームワーク
ハード	ロボット	EV3	角度, 距離などのセンサ搭載
	カメラ	Xtion	RGB, 深度, 赤外線センサ搭載

4.1　Jason

Jason[5] は、BDI エージェントの構築プラットフォームのひとつで、AgentSpeak のインタプリタである．Jason で記述するプランはトリガリングイベント、コンテキスト、ボディで構成されている．論理型プログラミング言語である Prolog と同様の記述方式を用いて、条件を組み合わせて複雑なプランを生成することができる．そのため、さまざまなプランを用意しておくことで、動的な環境に適応したプランを選択することができるようになり、目標達成に向けた幾多のアプローチが可能となる．また、選択したプランに失敗した際も、失敗回復のためのプランを用意しておけば、それを選択して行動することで、目標達成に復帰することができる．

本研究において実装した「目標物に到達する」という目標は、BDI エージェントのサブゴールのひとつに該当する．5 章では、これを実現するために行ったプラン設計について述べる．

4.2 ROS

ROS(Robot Operating System)[6][7] とは、オープンソースで提供されているロボット用ソフトウェアフレームワークのひとつである．複雑なロボットのふるまいを簡単にプログラミングするために開発され、幅広いロボット開発用プラットフォームを備えている．ハードウェア抽象化やメッセージ通信、パッケージ管理、可視化ツールなどの機能が含まれる．また、ROS のコミュニティは世界中に広がっており、あまたのユーザが開発したソフトウェアがフリーで公開されている．これらを利用することで、容易に自身のロボット開発を発展させることができる．

本研究では、従来研究より継続して ROS を用いた制御を行っている．

4.2.1 基本用語

ここでは、ROS で用いられるいくつかの用語について簡単な説明を行う．

- ノード … ROS の最小の構成要素．実行プログラムに相当する．
- パッケージ … ソフトウェアをまとまりとして管理するための単位．
- メッセージ … ノード同士でやりとりされる情報で、特有の型を持つ．
- トピック … メッセージをやりとりするための通信網の役割を持つ．
- 配信・配信者 … メッセージをトピックに送ることを配信という．また、メッセージを配信するノードを配信者と呼ぶ．
- 購読・購読者 … メッセージをトピックから受け取ることを購読という．また、メッセージを行動するノードを購読者と呼ぶ．
- サービス … ノード同士でリクエストに対してレスポンスを返すという方法で情報をやりとりする際に用いられる．特有の型を持つ．

4.2.2 通信方法

ROS でノード同士が通信する方法には 2 通りある．1 つは、トピックを介したメッセージ通信である．この方法は非同期通信であるため、継続的な通信に向いている．配信者と購読者が同時にトピックにアクセスしてメッセージをやりとりすることで、シームレスに情報を受け渡すことができる．ひと

つのトピックに対して配信者と購読者は複数存在することも可能である．また、どちらかしかない場合も問題ではない．もう 1 つは、サービスによる通信である．この方法は同期通信であるため、一方が要求しもう一方が応答するような通信に向いている．図 6, 7 に、ROS の通信モデルを示す．

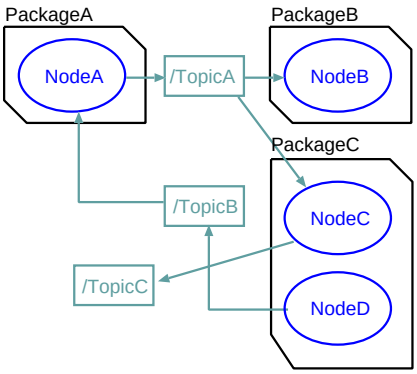

図 6: ROS メッセージ通信モデル

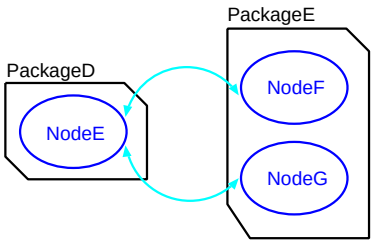

図 7: ROS サービス通信モデル

メッセージとサービスはそれぞれ ROS 独自の型を有しており、ROS ノードでやりとりされる情報は一定の型で処理される．これはつまり、異なる言語同士のプログラムであったり、他者が開発したプログラムを利用する場合であっても、ROS 通信を利用して、簡単に相互で情報をやりとりすることができるということである．

4.3 EV3

本研究で使用するロボットは、LEGO 社の LEGO Mindstorms EV3[8]（以下 EV3 とする）である（図 8）．従来研究では、同社の LEGO Mindstorms NXT[9]（以下 NXT とする）を使用してきた．しかし、NXT が ROS に直接対応していないこともあって、開発において下位のロボット制御と上位の行為決定との間を結ぶ機構に柔軟性を欠いていた．そこで、本研究では ROS に直接対応しており、より高精度の EV3 を導入した．

従来研究では、NXT 制御のライブラリとして NXT Python+の開発が行われており [10]、本研究では EV3 制御のライブラリの構築を行った．なお、このライブラリについては共同研究者である小松の 2017 年度卒業論文 [11] で述べる．

図 8: 使用ロボット

5　実装

目標物到達機構は、獲得した目標物認識能力などの知覚情報をもとに、エージェントが思考し、ロボットが目標物へ向かって探索行動や接近行動を繰り返すことによって実現されている．構成要素は、3 部門に分けられる．1 つ目は、ロボットエージェントの知能をつかさどる Jason 部である．2 つ目は、カメラやロボットのセンサ情報の受け取りや、指令を受けて動作するロボット部である．3 つ目は、Jason 部の指令とロボット部の知覚を相互に伝達することでロボット制御を行う ROS 部である．ここでは、従来研究 [2][12][13][14] と本研究の設計についてそれぞれ述べ、本研究で行った改良点を示す.

5.1　従来研究における設計

2.1.1 節でも述べたように、従来研究では、目標物到達機構に用いるロボットの目標物認識能力が低かった．それによって、ロボットが目標地点まで辿り着けないことがしばしばあった．これを補うために、効率的に行為を遂行する仕組みが形成された．行為に失敗する場合の定義を行うことで、エージェントが行為の失敗を認識することができるようになる．また、そこで目的の達成を失敗とするわけではなく、可能なら何らかの回復行動をとって引き続き目標達成のために行為を行うことができる．さらに、Xtion カメラや NXT に搭載されたセンサの情報を利用して、より多くの知覚をもとに、複雑な行動を行えるようになった.

ロボットが目標物到達行為に失敗する場合としては、ロボット自体の不具合やロボットと目標物との間に障害物が発生する場合などさまざまな状況が想定できる．従来研究では、「方位」と「所要時間」の側面から行為の失敗を認識することとした．従来研究における設計を、図 9 に示す．なお、この図は図 6・7 と同様、青丸は ROS ノード、緑は ROS メッセージの流れ、水色の矢印は ROS サービスによる通信を表し、黒枠は ROS パッケージを表す.

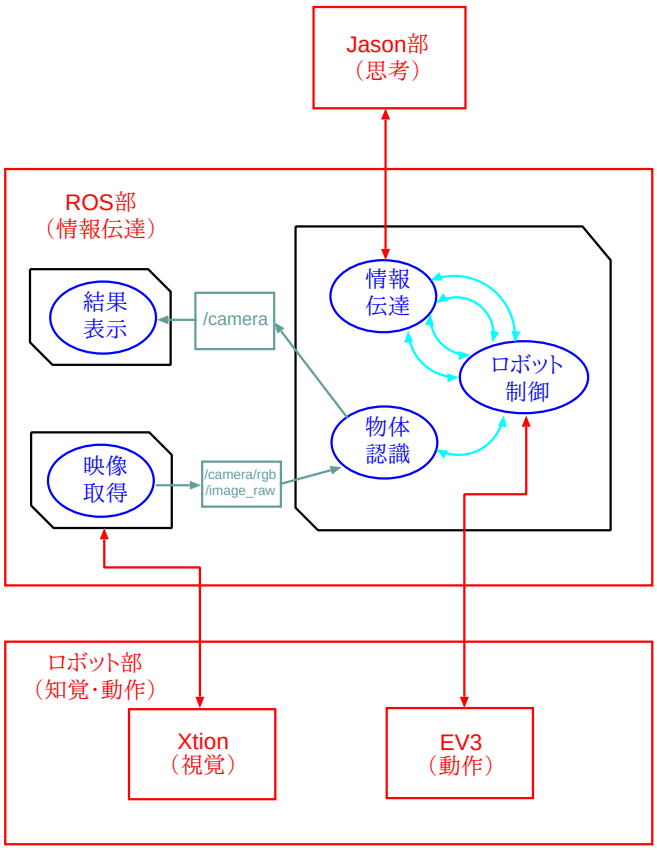

図 9: 従来の設計

Jason 部

目標物到達のプランは、Jason 部から目標物を引数に渡して呼び出される．すると、ROS 部に指令が伝達され、ロボットは到達行動を開始する．しばらくすると、ROS 部から到達が完了したかの状況を示すステータスが返る．ステータスは、成功、継続中、失敗の 3 パターンがあり、次のプランはこのステータスに応じて選択される．成功のステータスの場合は、目標物に到達したことを示すので、そこで処理を終了する．継続中のステータスの場合は、到達行動開始からの経過時間をみて、一定時間以内であればもう一度同じ目標物を指定して目標物到達プランを継続するが、一定時間以上であれば時間超過で到達失敗とみなす．失敗のステータスは、ロボットが到達行動を開始してから 360 度の全方位を探索し終えると返ってくる．つまり、このステータスはロボットがその地点で目標物に到達することができなかったことを示しており、次に回復行動のプランを選択する．回復行動には 2 パターンあり、ロボットが到達行動中に目標物を認識せずに失敗した場合は地点を変えて再

探索を行い、認識して失敗する場合は途中に障害が発生したと考えられるため、一定時間待機した後、もう一度その地点で再探索を行う.

ロボット部 (NXT)
ここでは、NXT に搭載した Xtion カメラからの RGB 映像、深度映像と共に、NXT に付属の方位を検出するためにのコンパスセンサの情報を受け取る．これを、外界からの知覚とし、ROS 部へ情報を伝達する．また、ROS 部から指令が来れば、モータを稼働して直進行動や回転行動を行い、目標物に向けて行動する．この動作は、モータのパワーと動作時間を指定することを繰り返して実現される.

ROS 部
Jason 部とロボット部の仲介の役割を担う情報伝達部である．ロボット部から届く RGB 映像をもとにした目標物認識による位置推定、深度映像をもとにした目標物までの距離推定、コンパスセンサによるロボットが向いている方位情報の取得を行う．さらに、この情報をもとに、ロボットに目標物体に対して接近・探索行動を実行させ、到達状況のステータスを Jason 部に返す.

以上の実装により、失敗が起こっても到達完了に向けてアプローチすることができるようになった．しかし、実世界の環境にはよりさまざまな状況が存在している．そのため、ロボットのセンサを増築して得られる知覚を増やすことなどで、ロボットエージェントのプランをより複雑に構成することが望まれた．また、ロボットが得た知覚情報などは、必要な時に適宜呼び出す仕組みになっていたが、ROS メッセージを活用して常時利用可能にすることが望まれた.

5.2 本研究における設計

本研究では、新たに獲得した高精度の物体認識能力と、新しいロボットを用いて、ロボットエージェントの拡張を行った．物体認識能力は、複数の物体を同時に認識できるようになったため、目標物までの経路設定機能を追加した．また、誤認識率が著しく低下し、信頼できる認識機能となったため、目標物の見失いを防ぐ機能も追加した．さらに、ROS との親和性が高い EV3 の利用と搭載するセンサの増築によって、柔軟なプランの構築を行った．図 10 に、本研究の設計を示す．なお、この図は図 9 と同様、青丸は ROS ノード、緑は ROS メッセージの流れ、水色の矢印は ROS サービスによる通信を表し、黒枠は ROS パッケージを表す.

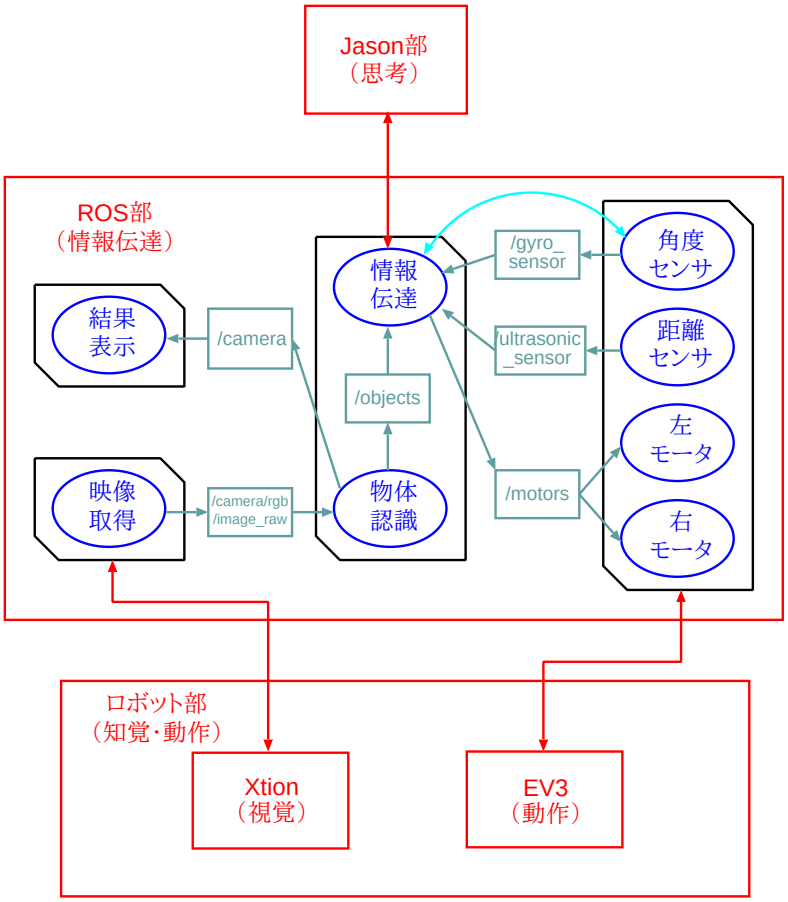

図 10: 本研究の設計

Jason 部

従来研究では、単体の物体のみ検出可能であったため、目標物到達プランは、指定した物体に到達することしかできなかった．しかし、本研究では複数の物体の同時認識が可能となったため、最終到達目標に対して、そこに到達するまでの経路を指定し、それらを順に経由することで最終目標への到達を果たすことが可能となった．基本的に、エージェントは指定の経路を順に経由するが、より後部に設定された経由地を途中で見つけた場合は、そちらに向かう方が効率がよい．そこで、目標に到達したかどうかのステータスをひとつ追加し、経由地をショートカットできるようにした．また、ロボット部で得られる EV3 のセンサ情報をもとに、プランが選択できるようになった．

ロボット部 (EV3)

従来研究で得られる知覚情報には、不正確なものがあった．1 つは、Xtion

カメラの深度センサである．これは、正面物体との距離測定に用いていたが、おおよそ近いか遠いかの情報しか得られず、距離が約 15cm 以内の場合は一定の値を示すなどの問題があった．正面物体との距離情報は、障害物の検知を行うときに重要な役割を果たすため、本研究では EV3 に搭載した超音波センサを用いて正確な距離計測を行った．超音波センサによって、0〜255cm の正確な距離を計測できるようになった．もう一つは、方位の検出に用いたコンパスセンサである．このセンサは動的な環境で使用するには値の振れ幅が大きい．また、方位は 0〜360 度の値で得ることになっており、回転角度を求めるには 0 度地点を左右に越える場合を考慮しなければならないなどの欠点があった．そこで、本研究では EV3 に搭載したジャイロセンサを用いて回転角度の計測を行った．ジャイロセンサでは、ロボットの向きが 1 回転を超えると方位として 360 度超や 0 度未満の値が得られ、$-32768 \sim 32767$ 度の範囲で方位を返すので、この範囲を超えない限り、0 度の向きを左右に超えるような回転を行っても単純な減算で回転した角度が得られる．また、動的な環境でもぶれずに動作可能である．本研究では新たに搭載したセンサおよび、ロボットの動作に用いるモータを ROS メッセージで制御する．これによって、ROS による統合的な処理が可能となった．

ROS 部

先述の通り、ロボットのセンサやモータの制御を ROS メッセージ化し、さらに EV3 制御用のパッケージとしてまとめたことにより、ノード同士のつながりが明確になり、ROS で利用しやすいようになった．また、以前は物体認識の不正確さなどが原因で、認識結果をもとに、ROS サービスを用いて行動選択を行うという仕様になっていた．つまり、「認識→行動→停止→認識→行動→停止→…」という順に処理が行われるため、一動作に所要する時間が大きかった．本研究では、高速かつ高精度化した物体認識能力を利用することで、認識結果を常に ROS メッセージとして配信することができるようになり、ロボットのスムーズな動作を実現した．

以上の実装により、従来研究に比べてプランが汎用化され、Jason や ROS のつながりも明確化された．

6 検証実験

本研究で実装した新しい目標物到達機構を用いた実験について述べる．従来研究では目標物への到達達成率は 10 回中 8 回となっていた．

6.1 実験環境

実験は、EV3 の Bluetooth の通信範囲、および Xtion カメラのコードの長さの制約などから、当研究室内の約 3m 四方の環境で行った．また、目標地点までの経由地のリストはあらかじめエージェントに与えておくものとする．目標物の認識可能範囲は物体によって異なるが、約 1〜2m であった．そのため、地点ごとの間隔を場合分けして、それぞれの到達達成回数を調べることにする．加えて、乗り越え困難な障害物などは設置しない．

ここで、目標物を約 10cm 四方とした場合の認識可能圏を表 8 に示す．

目標物体	A	B	C	D	E	F
認識可能距離 (cm)	119	160	140	133	113	133

表 8: 6 物体の認識可能距離一覧

6.2 実験結果

実験は、経由地ごとの間隔を以下の条件に分けて配置して行う．

条件 1: 1.0m 間隔で経由地を配置する

回数	1	2	3	4	5	6	7	8	9	10
経由地の数	1	1	1	2	2	2	3	3	4	4
到達完了	○	○	○	○	○	○	○	○	○	○
到達までの時間（秒）	19	18	29	70	53	57	88	96	123	139

表 9: 1.0m の実験結果

条件 2: 1.3m 間隔で経由地を配置する

回数	1	2	3	4	5	6	7	8	9	10
経由地の数	1	1	1	2	2	2	3	3	4	4
到達完了	○	○	○	○	○	○	○	○	○	○
到達までの時間（秒）	25	69	47	114	55	57	199	104	245	332

表 10: 1.3m の実験結果

6.3　考察

今回の実験では、すべてのパターンで最終目標地点への到達が可能であった．また、1.0m 間隔ではすべての物体で認識可能圏に目標地点が存在するためスムーズな到達が可能であったが、1.3m 間隔では認識可能圏内に目標地点が存在しない場合があるため、失敗の回復行動を伴って目標に到達することになり、到達時間を要する結果となった．以上のことから、経由地のリストには、次に見えるであろう目的地を指定するべきだが、仮に見えていない目的地を指定してしまった場合にも到達することが可能であると言える．問題としては、鋭角的に目標物に接近する場合、距離センサの照射範囲が狭いため、物体までの位置を正確に測れず衝突してしまう場合もみられた．

7　まとめと今後の課題

本研究によって、高精度な画像認識能力を獲得した．また、この能力を利用することによって、より汎用的なロボットエージェントを構築することができた．これらにより、高精度の目標物到達機構を実現することができた．なお、本研究での開発では、GPU の利用に NVIDIA 社のプロプライエタリなコンピューティングプラットフォーム (CUDA) とドライバを用いている他は、全てフリーなソフトウェアを用いている。

今後の課題としては、GPS センサを搭載することによって、物体の位置関係を把握し、経路の組換えを行うことや、障害物を回避したり、路面の状況を考慮して行動するというような、より汎用的なプランを構築することができるようになることが望まれる．また、認識可能圏内に目標物が存在しない場合の探索行動をより効率的なものにする必要がある．

謝辞

本論文の執筆にあたって、熱心にご指導いただいた新出尚之准教授、ならびに、共に研究に勤しんだ小松さん、そして、新出研究室の皆様に深く感謝の意を表す．

参考文献

[1] 藤吉弘亘, 堀修, 三田雄志, 山内悠嗣他．ディジタル画像処理 [改訂新版]．公益財団法人 画像情報教育振興協会, 2015.

[2] 樽井志織．目標物到達機構を持つ自律的な小型ロボットの制御を行う bdi エージェントの構築．2016 年度修士論文, 奈良女子大学人間文化研究科博士前期課程情報科学専攻, 2017.

[3] 桑井博之, 豊沢聡, 永田雅人. 実践 OpenCV2.4 for Python 映像処理&解析. 株式会社カットシステム, 2014.

[4] 斎藤康毅. ゼロから作る Deep Learning Python で学ぶディープラーニングの理論と現実. オライリー・ジャパン, 2016.

[5] Rafael H. Bordini, Jomi Fred Hübner, and Michael Wooldridge. *programming multi-agent systems in AgentSpeak using Jason*. Wiley, 2007.

[6] Ros.org. `http://www.ros.org/`.

[7] Aaron Martinez and Enrique Fernandez. *Learning ROS for Robotics Programming*. Packt Publishing, 2013.

[8] The LEGO Group. mindstorms EV3. `https://www.lego.com/en-us/mindstorms/products/mindstorms-ev3-31313`.

[9] The LEGO Group. レゴマインドストーム公式サイト. `http://www.legoeducation.jp/mindstorms/index.html`.

[10] 小島侑子. 小型ロボット制御のための汎用ライブラリの構築. 2011 年度修士論文, 奈良女子大学大学院人間文化研究科, 2012.

[11] 小松芙美子. 自律ロボット行動制御ライブラリの拡張. 2017 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, 2018（発表予定）.

[12] 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実装について. 2016 年度卒業論文, 奈良女子大学理学部情報科学科, 2017.

[13] 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.

[14] 石井あすか. 自律ロボットの目的達成にお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.