

2016 年度 修士論文

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

奈良女子大学大学院 人間文化研究科 博士前期課程 情報科学専攻 新出研究室
15840050 樽井志織

平成 29 年 1 月

目次

1	はじめに	3
1.1	研究背景	3
1.2	使用ロボットについて	4
1.3	NXT Python の拡張	4
2	制御機能	5
2.1	学習機能	5
2.2	拡張ライブラリと ROS によるロボットの制御	5
2.3	BDI エージェントの構築	6
3	実装	6
3.1	画像収集とそれに伴う処理の自動化	6
3.2	目標物認識機能の実現	7
3.3	BDI エージェントによる制御	7
4	実験による検証	9
4.1	学習における画像の認識精度の検証	9
4.2	目標物への到達	9
4.2.1	条件 1	10
4.2.2	条件 2	10
4.3	目標物到達を用いたプランによる行動	10
5	まとめ	11
6	謝辞	11

概要

自律的に目標を達成するロボットの実現のためには、目標を達成するための基本行為をロボットが実行できる必要がある。従来ロボットが基本行為を行うためのライブラリとして、NXT Python+と呼ばれるライブラリの構築とその改良が行われてきた。この過程では、ロボットが実世界で行動するための改良が行われた。例えば元来のNXT Python+ではロボットが90度回転の行為を行う際、その都度パワーを手動で決定する必要があったが、学習機能でモータのパワーを学習することにより、環境が変化しても適切な動作を行うことが可能となった。しかし、自律的に目標を達成するロボットの実現のためには、これまでにNXT Python+で実現されている機能以外にも、目標物への到達など、新たな能力を必要とする。よって本研究では、その能力の1つとして目標物を学習によって認識し到達するという行為を行う機能を構築した。目標物の認識法としては、Haar-Like 特徴量による画像認識をカラーヒストグラムによる判定で補うことで、目標物への到達率を上げた。本論文ではその実現について述べる。

また我々は、我々が実現した機能も含め、このライブラリが提供する機能を、自律的に目的を達成するロボットを構築するのに適していると考えられる、BDI エージェントの基本行為として利用できるようにした。これにより、エージェントがプランの中での基本行為として我々が実現した機能を使用することで、実際に自律的に目標物まで到達することや、さらにそれを副目標として含む多様な行動をとらせることが可能となった。その例をこの論文内で述べる。

1 はじめに

1.1 研究背景

近年、情報科学の発達とともに、人間が持つ能力をロボットに習得させ、実世界にて活用することが求められている。我々はこのようなロボット実現のため、実世界の環境にあわせ、自律的に目標を達成するための能力を提供するライブラリ構築を行っている。

我々が用いているロボットである、LEGO 社の Mindstorms NXT を動かすためのライブラリには、NXT Python と呼ばれるライブラリが広く用いられていた。またこれまでに、このNXT Python に対し、一定量の進行や回転など、より上位の機能を基本行為として追加したライブラリとして、1.3 で述べる NXT Python+ が作成されている。さらに、NXT Python+ で実現された基本行為は、ロボットの重量やまわりの環境が変化した場合、その都度変化した環境に合わせてモータのパワーを手動で調整する必要があったが、これまでに我々は、ライブラリに学習機能を取り入れてこの点を改善した。この学習機能は、強化学習を使用し、NXT Python+ に実装されている行為のうち、ロボットが行動する際に使用頻度が高い90度回転の機能を使用する際のモータのパワーを学習するものである。これにより、手動でモータのパワーの調整を行うことなく、環境が変化しても基本行為の実行が適切にできるようになった。

このように、これまでにはライブラリにロボットの進行や回転などの行為を単位とした機能を持たせることと、それらの機能を強化学習によって動的な環境変化に安定的に対応させることを旨とした拡張が行われてきた。しかし、自律的に目的を達成する小型ロボットの実現を達成するためには、他にもさまざまな行為を行う能力を実現する必要があり、その1つが、目標物を定めてその目標物に接近する能力である。

そこで本研究では、この基本行為として、小型ロボットが与えられた目標物を認識しその目標物まで接近するという行為を、新しい機能として実装した。その際、画像の認識法としてよく用いられる Haar-Like 特徴量による認識法を、カラーヒストグラムを用いた判定で補完すると、目標への到達率が顕著に上がることがわかった。4.2 節でそれについて述べる。

また、自律的に目標を達成する小型ロボットの実現には、人間と同様に目的達成の意図を保持してそれを実行しようとする、BDI エージェントを用いるのが適していると考えられる。そこで我々は、我々が実装した機能も含め、このライブラリを、BDI エージェントのプラン中で基本行為として用いることができるようにした。これにより、ロボットがこれらの機能を用いたプランによって行動し、目標物到達あるいはそれを副目標とするより複雑な行動をとることも可能となった。後に 4 節でその例を示す。

1.2 使用ロボットについて

実験に使用するロボットには教育用 LEGO Mindstorms NXT (以下 NXT と記載) [2] を採用した。NXT は比較的安価なロボットであり、これを使用することで、ライブラリを使用できる機会を増やすとともに、実験の再現を行いやすくした。また、従来研究で得られていた既存ライブラリの開発に NXT を用いていたことからの継続性も考慮している。

NXT は LEGO 社から販売されていたロボットで、頭脳の役割を果たしているインテリジェントブロックにさまざまな形や機能が備わったブロックを組み合わせて、自由にロボットの製作を行うことができる。インテリジェントブロックには出力ポートが 3 つ、入力ポートが 4 つ存在していて、そのポートにセンサーやモーターの機能が備わっているブロックを接続することで、コンピュータから指示を送りロボットの操作を行ったり、ロボットが感知した環境の情報をコンピュータで受け取ったりする。また NXT の制御プログラムは、USB の接続または Bluetooth の通信でコンピュータから転送することが可能であり、本研究での実験では Bluetooth による通信によって NXT の操作を行った。

なお本研究のロボットは、NXT のインテリジェントブロックにロボットの移動のためのモーターを左右 1 つずつで計 2 つ接続している。また、NXT のセンサーとは別にカメラとして ASUS 社の Xtion PRO LIVE をロボットに装着し、それをコンピュータへ接続してロボットの目標物の認識を行っている。

1.3 NXT Python の拡張

この節では、先行研究で得られた NXT Python の拡張である NXT Python+ の機能について述べる。

NXT を制御するためのライブラリはいくつか存在している。我々は、メモリ管理が不要であること、GUI の設計が容易であることなどをはじめ、開発が容易であることを考慮し、開発言語として Python を選択した。このため、ライブラリとしては Python を使用している NXT Python を選択した。しかしこのライブラリは、モーターの回転数を直接制

御するといった下位レベルの制御命令が基になっており、複雑な行動をロボットに行わせようとすると、下位レベルの制御命令をいくつも組み合わせてプログラムを記述する必要があった。このようにプログラムを記述すると、上位レベルの制御命令でプログラムを記述するよりもプログラムが長くなり、記述ミスが増加してしまうといったことや、改良を行い辛いといった欠点となる。

この欠点を改善するため、先行研究 [5] ではライブラリの拡張が行われた。この拡張されたライブラリは NXT Python+ と名前が付けられ、下位レベルの制御命令を組み合わせて作成された上位レベルの制御命令を使用することや、構築された学習機能によって環境やロボット毎にモーターのパワーの大きさを学習することなどが可能である。また、プログラミングに慣れない者であってもロボットの操作が行えるように、スライダやボタンで視覚的にロボットのポートやモーター出力の大きさなどの設定を行うことができる GUI も備えている。

2 制御機能

本節では、我々がロボットの制御を行うために使用した技術について述べる。述べる機能は、目標物を認識するための学習機能、ロボットの制御部を担う ROS、BDI エージェントについてである。

2.1 学習機能

1 節で述べたように、人間の能力に値する基本行為として、目標物を認識しその目標物へと接近するという行為の実装を行う。この行為の目標物を認識するために、学習による画像認識機能を利用した。本研究で使用するロボットとして、実験の再現を行いやすい安価な小型ロボットを想定しているため、利用のしやすさを重視して、機能を作成する際に使用するライブラリには広く用いられている画像処理ライブラリである OpenCV を使用した。また、言語については先行研究に合わせて Python を使用しているので、OpenCV-Python の使用となる。

本研究ではこの OpenCV がサポートしている学習機能の 1 つである traincascade を使用し学習を行った。また、OpenCV には与えられた画像からカラーヒストグラムを作成する機能があり、この機能を認識の補佐に使うことで、目標物到達の成功率を上げることができた。

2.2 拡張ライブラリと ROS によるロボットの制御

ロボットの行動を決定するエージェントの構築の際に、ロボットの制御部の開発を容易にするため、本研究ではロボット開発用のフレームワークである ROS (Robot Operating System) [3] を使用して、NXT Python+ を利用するエージェントの構築に取り組んだ。ROS はオープンソースで提供されているロボット製作のためのオペレーティングシステムであ

る。ROS では、ソフトウェアの構造の一単位であるパッケージが存在し、その中のライブラリの実行ファイルなどをノードと呼んでいて、このノード同士が ROS クライアントライブラリによって通信を行うことによって、異なるノードがお互いにやりとりしながら協調して動作を行う。これにより、ロボットの各機能を独立したプログラムとして作成して、それらを結合することによる開発を可能となること、また、構築したプログラムの再利用が容易となることなどの利点がある。

本研究での ROS 利用の利点の一例として、目標物を検出するプログラムである `haar_like` ノードの構築が挙げられる。このノードは、Python にて作成されたプログラムが元となっており、プログラムにいくつかの加筆を行うことのみでノードとしての使用が可能となった。このように ROS は、ある機能を使用したい場合に、1 からプログラムの構築を行うのではなく、同じ機能のプログラムを再利用する形でロボット開発を行うことができる。

なお、この ROS を使ったエージェント部と NXT Python+ の仲介および実装についての詳細は 3 節にて述べる。

2.3 BDI エージェントの構築

ロボットの制御部に NXT Python+ と ROS を用いたのとは別に、それらで実現される基本行為を用いて自律的に目的を達成するエージェントの構築プラットフォームには、Jason[1] を使用した。これは BDI エージェント構築のためのプログラミング言語である AgentSpeak のインタプリタである。BDI エージェントとは Belief (信念)、Desire (願望)、Intention (意図) の 3 つの心理状態を使って行動の決定を行うエージェントである。エージェントはまず、信念と願望から目標を導きだし、この目標を達成させる手段であるプランを選択してこれを意図とする。次にこの意図つまり手段を実行して、実行後のエージェント外部の変化を知覚し、その知覚をもとにして再び信念と願望から目標を導き出す。このサイクルによって目的を達成するのが BDI エージェントである。Jason はこの AgentSpeak を拡張した言語のインタプリタであり、Prolog に似た形でエージェントが目的を達成するためのプランを記述する。さらに、エージェントが置かれている外部環境を表すプログラムを Java で実装すれば、環境とエージェントが相互作用することで知覚を得たり行為を実行したりすることが可能となる。我々はこれを用いて、エージェントが我々のライブラリで実現される基本行為をプラン内で用いることを可能とし、プランをエージェントが選択することにより、ロボットが環境に応じて柔軟な行動をとることを可能とする。

3 実装

3.1 画像収集とそれに伴う処理の自動化

今回我々は、クマとカエルのぬいぐるみを目標物に用い、これらを区別しながら目標物にたどり着く問題を例題として用いた。学習には目標物、およびそうでないものの画像が大量に必要となるが、既に存在する画像データ集を用いるのでない限り、画像データの収

集には多量の労力を必要とし、収集後の教師データとしての調整（切り出しなど）や分類にも労力がかかる。我々は、後者の効率化のためネットワークで配布されていた既存のソフトウェアを用いたが、より目的に合わせた効率化を図るために、次に述べるようにソフトウェアの改造を行った。

4.1 節で述べる通り、本研究では 2500 枚以上の画像を収集し分類する必要があるだろうと考えられていた。しかし従来のソフトウェアでは、この全てを 1 度に分類しなければならず、また、新しく画像を追加する場合や分類を誤った場合には、初めから分類をやり直す必要があった。この手間を省略するためにソフトウェアに手を加えて、1 度分類を行った画像については再度分類を行わず、途中で作業を中断した場合でもその場所から作業を再開できるように改造した。これにより、収集した画像を逐次追加したり、分類を訂正したりする処理が円滑になった。

3.2 目標物認識機能の実現

目標物認識機能の構築には、2.1 節で述べた通り、学習機能を使用した。目標物の写った画像である正解画像と目標物の写っていない不正解画像を収集し、正解画像内の目標物が存在する座標と個数を traincascade へ与える。学習の方法は 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、学習の速度が速いことと学習の結果で最も高い正解率が得られたことから、今回は Haar-Like 特徴を選択した。そして学習した結果である cascade ファイルを、OpenCV を使用した小型ロボットの制御プログラムで読み込み、目標物の認識を行うことを可能とした。

しかし、本研究でクマの学習を行った場合に、目標物ではない（クマのぬいぐるみではない）物体を目標物と認識してしまう誤認識が多く見られた。このような場合においても目標物を認識し、接近するという目的を達成できるように、目標物の認識機能を補佐する目的で、目標物のカラーヒストグラムの利用も行った。Haar-Like 特徴での目標物の認識は、物体の色情報による識別を行っておらず、類似した形状の物体である場合、上述の検証の通り、類似の物体を目標物と誤認識する場合があった。この誤認識を軽減させるために、学習機能で認識した物体と目標物の色情報を使用して、形状だけでなく色も類似しているものを接近すべき目標物と認識するよう機能の構築を行った。

2.1 節で述べた OpenCV の機能を利用して目標物の画像からカラーヒストグラムを作成する。また、小型ロボットの前方の映像から、目標物と判別された場所の画像を習得し、この画像からもカラーヒストグラムを作成する。この 2 種類のヒストグラムの差違を計算し、差違が大きいものを目標物でないとする事で、形状のみが類似する物体への誤認識を減少させた。

3.3 BDI エージェントによる制御

1.3 節にて述べた小型ロボットを制御する機能や、2.1 節にて述べた学習結果を利用した目標物の認識機能を、2.3 節にて述べた BDI エージェントの基本行為として利用可能にし

た。これにより、構築された基本行為を BDI エージェントがプラン内部で使うことが可能となり、目標物の認識と接近が可能となった。決められた目標物についての学習結果とその名前を与え、さらに目標物の色の情報として画像を与えると、本研究にて用いた目標物だけでなく、他の目標物についても、認識および接近という基本行為を行うことができる。

エージェント側の制御プログラムと小型ロボット側の制御プログラムは、別々のディレクトリに分けて独立に実装した。前者は `jason` ディレクトリ、後者は `scripts` ディレクトリという名であり、これらの名は以降、説明に用いる。また、この 2 つをあわせ、`jason_ros` という名前の ROS パッケージを作成した。なおこの `scripts` ディレクトリは ROS の実行可能スクリプトを格納する `scripts` ディレクトリと同一のものである。`jason` ディレクトリ内には、BDI エージェントの環境設定ファイルである `mas2j` ファイルとゴール・プランのプログラムである `asl` ファイルがあり、これらは Jason によって用いられる。`scripts` ディレクトリには主なプログラムとして、Jason と通信を行うプログラムである `tcp` と、目標物の検出を行うプログラムである `haar_like`、そして NXT の制御・通信を行うプログラムである `nxt` の 3 つが格納されている。これらは ROS でのノードにあたり、パッケージである `jason_ros` を動かせば 3 つのノードが動きだし、2.2 節で述べたような動作を行う。これら 3 つのノードは、上述した通り ROS クライアントライブラリによって通信を行うが、エージェント側のプログラムはクライアントライブラリで通信を行わないため、ロボット側の通信はノードのうちの一つが TCP を介して通信を行っている。(図 1)

ROS 側でこのようにして実現したロボット制御機能を、Jason によるエージェントから基本行為として用いるには、Jason 側で Java 言語により環境を実装し、この中で、所定の行為が呼び出されたら、TCP による通信で ROS 側の `tcp` ノードに呼び出しが伝わるようにする。また、行為の実行結果が `tcp` ノードから Jason 側の環境に伝えられ、エージェントはこれを知覚として取得するようにする。

Jason と ROS が起動されると、まず `nxt` ノードが NXT と ROS の通信を行う。NXT と通信が可能であれば接続した後、この情報を `tcp` ノードへと送信し、これを受け取った `tcp` ノードはエージェント側へと状況を伝える。接続が完了すると、次にエージェント側から到達すべき目標物の情報が `tcp` ノードに渡される。これが `haar_like` に伝わると、`haar_like` はまず目標物の `cascade` ファイルが存在しているかどうかを確認し、これが確認できれば `tcp` ノードを通してエージェントにファイルの有無が伝えられる。ファイルが存在し、目標物の認識が可能な場合、`haar_like` と `nxt` は目標物の認識、できない場合は検索などを行い、目標物に対して何らかの行動を取ることができる。なお本研究では、目標物を認識した後一定距離まで接近するという行動を目標としてプログラムを構築した。NXT が設定された目標を達成し、または達成が不可能であることが認識された場合、この情報がまた `tcp` ノードを介してエージェントに伝えられ、伝えられた情報を基にエージェントの環境が更新され、更新された環境から次の目標が設定される。

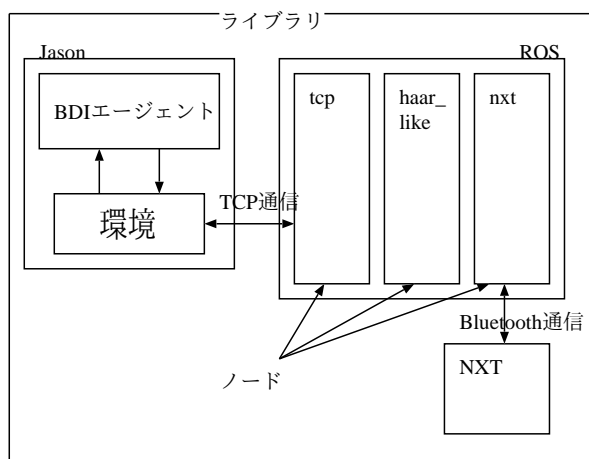


図 1: Jason および ROS による実装

4 実験による検証

4.1 学習における画像の認識精度の検証

集めた画像における目標物の認識精度を確認するために、画像を 10 分割して 9/10 毎に学習を行い、その学習結果で残りの 1/10 の画像をどの程度判別できているのかを算出した。前述の通り学習には Haar-Like 特徴量を使用し、あらかじめ cascade ファイルの作成を行った。判別する画像には正解画像と不正解画像があり、正解画像で目標物が存在していると認識できている状態と不正解画像で目標物を認識していない状態を正解とし、正解画像で目標物を認識できていない状態と不正解画像で目標物が存在していると認識している状態を不正解とした。今回研究で使用した画像の枚数は正解画像 1500 枚と不正解画像 1000 枚で、これらを使用し学習した結果は、正解画像を正解と認識できた確率が 68.8%、不正解画像を不正解であると認識できた確率が 51.6% となった。この結果から、学習によって目標物を認識できる確率は高いが、目標物と目標物でないものを見分ける点において工夫が必要であると判明した。なお、この検証においてはカラーヒストグラムの利用は行っていない。

4.2 目標物への到達

構築したエージェントが、今回基本行為として実装した目標物に対する接近行為を行うことが可能であるか、実際に目標物と、達成すべき目的を設定して検証を行った。目標物は市販の熊のぬいぐるみを設定し、目標物のぬいぐるみが写った正解画像 1500 枚と、目標物のぬいぐるみが写っていない不正解画像を 1000 枚、合計 2500 枚収集した。これは 4.1 で収集した画像と同じものである。また、エージェントの達成すべき目的には、認識させ

たぬいぐるみを探索、認識し、距離が離れている場合には一定距離まで接近するという行動を設定した。

まず、障害物のない空間に NXT と目標物であるぬいぐるみを設置する。条件を 1. 目標物は 1 つで視界に入っている場合、2. 空間に存在しているが視界に入っていない場合、の 2 つの条件に分けて検証を行った。また、検証実験にてカラーヒストグラムを利用して目標物の認識を行った場合と、利用しなかった場合についての到達率を計算し、比較した。

4.2.1 条件 1

NXT と目標物を障害物のない空間に設置する。視界に目標物を認識できる場合、NXT は探索を行うことなく目標物に接近することができた。なお、視界の中心に目標物を認識できるときはそのまま前進し、少し外れている場合は、目標物が視界の中心となるように体の向きを調整してから前進を行う。

また、この条件で検証を行った際の到達回数は、カラーヒストグラムを用いない場合が 10 回中 4 回、用いた場合が 10 回中 8 回であった。

4.2.2 条件 2

NXT と目標物を障害物のない空間に設置する。視界に目標物を認識できない場合、NXT はまず 90 度回転を行って探索を始める。探索にて目標物の認識に成功すれば、NXT は発見した目標物に向かって前進し、接近を行うことができた。

また、この条件で検証を行った際の到達回数は、カラーヒストグラムを用いない場合が 10 回中 6 回、用いた場合が 10 回中 8 回であった。

4.3 目標物到達を用いたプランによる行動

上記の実験では目標物はクマのぬいぐるみのみを設定したが、BDI エージェントとして構築していることから、プランを記述すれば、認識し接近する目標物は複数設定することも可能である。例えば「目標物 2 の所在する地点に到達する」という目標に対して、副目標「目標物 1 に到達する」「目標物 2 に到達する」からなるプランが用意されていた場合、1 番目に指定された目標物に接近した後、そこから 2 番目に指定された目標物を探索して接近することで目的達成となる。例として我々は、1 番目の目標物にクマのぬいぐるみを設定し、2 番目の目標物にカエルのぬいぐるみを設定して実験を行った。まず NXT はクマのぬいぐるみを探索し、接近した後に、次はカエルのぬいぐるみを探索し、クマのぬいぐるみと同じように接近を行った。このように、目標物到達を副目標としてより複雑な行動を行わせることが可能になる。

5 まとめ

本研究では、ロボットが自律的に目的を達成するための能力の1つとして、目標物を認識して到達するという能力を学習機能を利用して実装した。Haar-Like 特徴量による画像認識に、カラーヒストグラムによる判定を援用することで、80%という目標物到達率を実現できた。また、BDI エージェントはプランライブラリから適切なプランとして実装したプランを選択して自律的に目標を達成することができた。

今回は、達成すべき目標として目標物へと接近するという行為を選択した。しかし、このライブラリをより発展させていくために、プランライブラリを改良し基本行為を追加することで、様々な行為を行うエージェントとして活用することが可能となる。また、今回実装した基本行為は、学習機能およびカラーヒストグラムを利用した認識と、単純な目標物への到達である。このため、目標物が見つからなかった場合やその他解決に複雑なプランが必要となる基本行為を実装することにより、目標物に到達するという基本行為をより改良していくことも可能である。他にも、学習機能について他の学習方法を導入することで、より効率よく学習を行うことができる可能性も考えられる。

また今後の課題として、画像の処理速度によって正確に目標物の認識が行われないといった問題や、目標物が正面を向いていない場合は認識率が低下してしまうといった問題などが存在する。より求められる形の目標達成を行えるようにするために、このような問題点や改善点について工夫を行うことで、更なるライブラリの発展を目指していく必要がある。

6 謝辞

本研究の遂行および本論文の執筆にあたり、丁寧にご指導して下さった新出尚之准教授に深く感謝致します。また、新出研究室の皆様にご感謝の意を表します。ありがとうございました。

参考文献

- [1] H.Bordini, R., Hübner, J. F., and Wooldridge, M.: *programming multi-agent systems in AgentSpeak using Jason (Wiley Series in Agent Technology)*, Wiley Interscience, 2007.
- [2] Lego Group: LEGO.com エデュケーションオフィシャルサイト, <http://education.lego.com/>.
- [3] Open Source Robotics Foundation: ROS.org, <http://www.ros.org/>.
- [4] 江川鈴菜: 強化学習による環境別のロボットの基本行為の獲得について, 2014 年度卒業論文, 奈良女子大学理学部情報科学科, 2015.

- [5] 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.
- [6] 樽井志織: 小型ロボット制御のための汎用ライブラリの拡張, 2014 年度卒業論文, 奈良女子大学理学部情報科学科, 2015.

対外発表

樽井，兼松，新出．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．日本ソフトウェア科学会第 33 回大会．2016 年 9 月．