

公立はこだて未来大学 2020 年度 システム情報科学実習
グループ報告書

Future University-Hakodate 2020 System Information Science Practice
Group Report

プロジェクト名

複雑系知能ロボットへの挑戦

Project Name

Challenge to Complex Intelligent Robot

グループ名

グループ A

Group Name

Group A

プロジェクト番号/Project No.

21-A

プロジェクトリーダー/Project Leader

安藤直輝 Naoki Ando

グループリーダー/Group Leader

古里優典 Masanori Furusato

グループメンバ/Group Member

赤坂駿斗 Hayato Akasaka

垂石哲平 Teppei Taruishi

湊宏太郎 Koutaro Minato

古里優典 Masanori Furusato

指導教員

松原仁 鈴木恵二

Advisor

Hitoshi Matsubara Keiji Suzuki

提出日

2020 年 1 月 31 日

Date of Submission

January 31, 2020

概要

“複雑系知能ロボットへの挑戦”プロジェクトは、人工知能とロボットの融合による知能ロボットの開発に取り組む。坂道や段差といった環境のなかでロボットが二足歩行するのは困難である。そのような場所でも活躍できる人型ロボットを作りたい。本プロジェクトの目標は次世代知能ロボットとして、様々な路面を進む人型ロボットの開発である。ロボットを開発するロボット班と画像認識技術や強化学習のプログラムを開発する AI 班の 2 つのグループに分かれて活動を行った。私たち AI 班はロボットの行動を選択するプログラムを作成することで目標の達成を目指す。

キーワード 人工知能, ロボット, 画像認識, 強化学習

(※文責：赤坂駿斗)

Abstract

The project “Challenge to Complex Intelligent Robot” is to tackle creating intelligent robot by fusion between Artificial intelligence and robot. It is difficult for the robot to walk on slopes and steps. We want to create a humanoid robot that can operate in such places. This project goal is to develop humanoid robot moving on various road as Next-generation intelligent robot. Now, we work in dividing into two groups of robot group and AI group to develop the image recognition and reinforcement learning program. The AI team to achieve the goal by creating a program that selects the behavior of the robot.

Keyword Artificial intelligence, Robot, Image recognition, Reinforcement learning

(※文責：赤坂駿斗)

目次

第 1 章	背景	1
第 2 章	到達目標	3
2.1	本プロジェクトにおける目的	3
2.1.1	通常の授業ではなく、プロジェクト学習で行う利点	3
2.2	具体的な手順・課題設定	3
2.3	課題の割り当て	4
第 3 章	課題解決のプロセスの概要	7
第 4 章	課題解決のプロセスの詳細	9
4.1	各人の課題の概要とプロジェクト内における位置づけ	9
4.2	担当課題解決過程の詳細	10
4.2.1	赤坂駿斗	10
4.2.2	垂石哲平	11
4.2.3	湊宏太郎	13
4.2.4	古里優典	14
4.3	担当課題と他の課題の連携内容	15
4.3.1	赤坂駿斗	15
4.3.2	垂石哲平	15
4.3.3	湊宏太郎	15
4.3.4	古里優典	16
第 5 章	結果	17
5.1	プロジェクトの結果	17
5.2	成果の評価	19
5.3	担当分担課題の評価	22
5.3.1	赤坂駿斗	22
5.3.2	垂石哲平	23
5.3.3	湊宏太郎	23
5.3.4	古里優典	24
第 6 章	今後の課題と展望	25
付録 A	中間発表の集計結果	27
付録 B	最終発表の集計結果	29
付録 C	新規習得技術	31
付録 D	活用した講義	33

付録 E その他製作物	35
参考文献	41

第 1 章 背景

近年、ロボット工学の分野は成長産業であり研究対象として注目を集めている。特に人型ロボットは、連日メディアを通して世界中の多くの研究を見聞きする。公開されている実験やプロモーションの映像では、人型ロボットは走ったり跳ねたりするものからドアを開ける動作に入る前に崩れ落ちるものまで多種多様である。そこで本プロジェクトでは、人間が足環境に対応し、自律して様々な路面を進む人型ロボットの開発を目指す。また、ここでは自律を「人工知能による制御で各モーションを繰り返すこと」として、開発を進める。私たち AI 班は本プロジェクトで強化学習と画像認識によるロボットの制御プログラム開発に取り組む。

現状、人型ロボットが人間のように行動することはできていない。これが実現できれば、私たちの生活の利便さ等に大きな影響を与えることだろう。これを達成するために必須のスキルは、大きく分けて「人間のように関節を動かして行動できること」と「人間のように自律すること」の2つだと考える。この2つのスキルを合わせ持つロボットの開発を目指すことが私たちの活動意義であると考えている。私たち AI 班は前期の活動で、障害物を色で判別するプログラムとカメラで撮影した画像や映像をソケット通信によって別のコンピュータに送信するプログラムを作成した。また、後期の活動では前期に作成したカメラのプログラムの改良やロボットを動かすプログラムとロボットが適切なモーションを再生するための強化学習のプログラムを作成した。

(※文責：赤坂駿斗)

第 2 章 到達目標

2.1 本プロジェクトにおける目的

様々な道を自律歩行できる人型ロボットの開発をすることが本プロジェクトの課題である。最終的には、公立はこだて未来大学の二階購買前から出発して一階に降り、食堂横の出入り口から屋外に出て最寄りのバス停まで自律歩行するロボットを作成することが目標である。そのためにロボットの周囲の障害物や地形情報を検知するプログラムや検知した情報をもとにロボットがどのような状態にあるかを判断し、どのモーションを採用するかについて強化学習を用いて決定するプログラムを作成することが必要だと考えた。

(※文責：垂石哲平)

2.1.1 通常の授業ではなく、プロジェクト学習で行う利点

本課題ではロボットの組み立て、機材の設定、実験といった時間をかけることが必要となる作業が多いため、複数人で作業を分担して取り組むことにより効率よく作業を行うことができる。通常の授業では教員による講義が中心なうえ、学生の個人作業が多いため時間のかかる作業は適していない。また、実験を行う際に実験日を跨ぐなどして途中で環境が変わってしまうと意図せぬ要因がはたらいてしまい実験結果が正しく求められなくなる可能性がある。そのことから、1時間30分で行われる通常の講義では取り組める実験に制限がかかってしまうが、プロジェクト学習では3時間続けて実験を行えるため、通常の講義よりも正確な実験結果を得ることができる。さらに、プロジェクト学習では学生同士が主体的かつ共同で作業を行っていくことで各々がサポートし合いながら作業を行うことが可能になり、成果物の質を高めることができるほか、コミュニケーション能力やプログラミング能力といった各々の能力の向上が見込める。

(※文責：垂石哲平)

2.2 具体的な手順・課題設定

1. テーマを決める

課題：ロボットを使って何をするかテーマを会議で決める。様々な障害を通り目的地に到達する2足歩行の人型ロボットを作ることを前提とした。

2. 人型ロボットについての話し合い

課題：ロボットやセンサは何を用いるか決める。また、どのような技術を使うと良いか決める。

3. 作業分担

課題：円滑に進めるため2つのチームに分ける。ロボット班とAI班の2チームに分かれて、お互いの作業分担を確認する。

4. 事前学習

Challenge to Complex Intelligent Robot

課題：ロボットがどのように動くかを学ぶためにロボットのモーションを学習する。画像認識への理解を深める。強化学習について学ぶ。理想のロボットを完成させるためにどのような機材が必要か調べる。

5. 客観カメラの設定

課題：スタートとゴールの位置やモーションを再生する毎に移動するロボットの位置を把握して管理する方法を決める。

6. 主観カメラの設定

課題：どのように画像を撮るか、撮った画像をどのように処理するかを決める。ロボットへの搭載方法をロボット班と相談する。

7. コースの決定

課題：具体的にどのようなコースでロボットを歩かせるかを決定する。ロボット班が調べた校内の地形を考慮して決める。

8. 強化学習の導入

課題：強化学習をどのように行うかを学ぶ。プログラミング言語と作りたい AI に適切な強化学習モデルの選定をする。

9. ロボットをプログラムで動かす

課題：強化学習をするにあたってロボットを Python のプログラムで動くようにする。

10. プログラムの統合

課題：全てのプログラムを統合する。前期に作成したカメラを用いた画像認識のプログラムとロボットを動かすプログラムを強化学習のプログラムと統合し、動作するように調整する。

11. 実験準備

課題：実験する環境を準備する。画像認識でロボットやゴールの位置を把握できるようにする。

12. 実験

課題：強化学習を用い、道にあったモーションを選択できるようにする。実験結果を参考に強化学習プログラム内の評価値を調整しスタートからゴールまでをスムーズに動くようにする。

13. 実践

課題：公立はこだて未来大学の二階購買前から出発し、一階に降り、食堂横の出入り口から屋外に出て最寄りのバス停までを実際に動かす。

(※文責：赤坂駿斗)

2.3 課題の割り当て

各人の希望に合わせて以下のように割り当てた。

- ロボット班と AI 班に分けた。
ロボット班 安藤直輝、田中澁大、加藤大地、小山晃弘、西谷和将
AI 班 赤坂駿斗、垂石哲平、古里優典、湊宏太郎
- ロボットのモーション班、画像認識班、強化学習班に別れた。
ロボットのモーション班 加藤大地、西谷和将、垂石哲平

Challenge to Complex Intelligent Robot

画像認識班 田中滉大、小山晃弘、古里優典

強化学習班 安藤直輝、赤坂駿斗、湊宏太郎

- 画像認識、動画認識
古里優典
- 主観カメラの処理
垂石哲平、湊宏太郎
- コースの決定
安藤直輝、田中滉大、加藤大地、小山晃弘、西谷和将
赤坂駿斗、垂石哲平、古里優典、湊宏太郎
- 強化学習の導入
垂石哲平、湊宏太郎
- ロボットのプログラミング
赤坂駿斗、垂石哲平、古里優典、湊宏太郎
- プログラムの統合
赤坂駿斗、垂石哲平、古里優典、湊宏太郎
- 実験
赤坂駿斗、垂石哲平、古里優典、湊宏太郎
- 実践
赤坂駿斗、垂石哲平、古里優典、湊宏太郎

(※文責：湊宏太郎)

第 3 章 課題解決のプロセスの概要

1. テーマを決める

解決過程：プロジェクト学習を進めていくにあたり、どういったものを作りたいのか、どのような技術を学びたいのかといったことを指導教員の意見を参考にしながらプロジェクトメンバー全体で話し合った。その結果、強化学習を用いてロボットを二足歩行させることをテーマとして決定した。

2. 人型ロボットについての話し合い

解決過程：プロジェクトの目標について話し合い決定した後、その目標を達成するために必要となるハードウェアやソフトウェアについて指導教員の意見を取り入れながら話し合った。ハードウェアであるロボットは一から作るのか、購入するのかについて話し合い、近藤化学が販売している人型ロボットを購入することにした。また、ロボットの状態を検知するためにどのようなセンサを搭載すべきか、プログラムを作成するにあたりどのような技術を応用し、実装すべきかについて決めた。

3. 作業分担

解決過程：効率的に作業を進めるため、プロジェクトメンバーをプログラムの作成などソフトウェアを扱う A グループとロボットの組み立てなどハードウェアを扱う B グループの二つのグループに分けた。また、プロジェクト全体をまとめるプロジェクトリーダーと各グループをまとめるグループリーダーについても話し合っ決めて、各グループで交流できる場を Slack に設けることで今後の活動を円滑に進めていく準備を整えた。

4. 事前学習

解決過程：購入した物品が届くまでの時間でロボットのモーション、画像認識、強化学習の三つについて事前学習を行った。その際、A グループと B グループのメンバーがそれぞれ最低一人は含まれるようにして三つのグループに分かれて学習を行った。学習教材としては指導教員に提示された Web サイトと情報ライブラリーで借りた書籍を用いた。定期的にグループ間で学習状況を確認し、レビューすることで情報を共有すると同時に学習意欲を高めた。

5. 客観カメラの設定

解決過程：ロボットや障害物を含めたコース全体を俯瞰的に撮影する客観カメラとして web カメラを使用した。また、OpenCV ライブラリーを導入し、カメラで撮影した映像に色認識を行うことで障害物とロボットの位置関係を座標の数値として検出するプログラムを作成した。

6. 主観カメラの設定

解決過程：ロボットに搭載する主観カメラとして Raspberry Pi のカメラモジュールを使用した。Raspberry Pi 内にあらかじめ用意されていた Python 環境を利用してカメラを起動し、画像を撮影するプログラムと撮影した画像を送受信するプログラムを作成した。実装にあたり、客観カメラと同様に OpenCV ライブラリーを導入した。

7. コースの決定

解決過程：ロボットがプロジェクトの目標を十分に達成したと判断するにはどのようなコー

スをロボットに歩かせればよいのかについて話し合った。階段や点字、坂道といった様々な状態の道を歩けることから、公立はこだて未来大学の二階購買前から最寄りのバス停までをコースとして決定した。

8. 強化学習の導入

解決過程：関連する書籍や教員から提示された web サイトを読み、強化学習に関する基礎知識を学んだ。また、サンプルプログラムを動かすなどして実装方法や応用について学んだ後、ロボットに歩行を学習させるためにはどのように行動や報酬を設定するかについてグループ全体で話し合い決めた。学習段階で使用していた参考書のサンプルプログラムをもとに強化学習を行うプログラムを作成した。

9. ロボットをプログラムで動かす

解決過程：専用のソフトウェアや付属のコントローラを用いずにロボットをプログラム言語である Python を用いて動かせるようにした。そのために、使用するロボットの販売元である近藤科のホームページから Python 用のライブラリをダウンロードした。ポート番号やボーレートがサンプルのものとは異なり、通信が行えないといった問題があったため、マニュアルを読んだり、教員のアドバイスをもらいながら実際に動かすことで解決した。

10. プログラムの統合

解決過程：画像認識を行うプログラム、強化学習を行うプログラム、ロボットを動かすプログラムの三つを統合した。強化学習を行うプログラムとロボットを動かすプログラムは一つのプログラムにまとめ、必要なタイミングで画像認識を行うプログラムを参照し、ゴールやロボットの座標を得られるようにした。それぞれのプログラムから得られる値を関連付けることにより、ロボットに強化学習を用いて歩行させるプログラムを作成した。

11. 実験準備

解決過程：上から撮影するカメラを設置するための備品を用意した。ゴールやロボットの位置、ロボットが転倒しているかどうかを画像認識で判断するために赤い布を床に置いてゴール地点とし、ロボットの頭や足裏などに黄色い布を貼り付けた。また、ロボットの胸部に青い布を貼り付けることで転倒時に仰向けかうつ伏せかを判断できるようにした。これはロボットが起き上がるためのモーションが仰向けの場合とうつ伏せの場合で異なるためである。

12. 実験

解決過程：スタートとゴールの位置を決め、作成したプログラムを用いて実際にロボットに歩行させた。50 回試行させることを目標として実験を行い、各試行終了時のロボットとゴールの距離を記録した。実験中、転んでも立ち上がれない、学習を重ねてもゴールに近づかないといった問題点が見つかった場合には実験を中断してモーションや評価値を見直し、調整を行うなどして対処した。

13. 実践

解決過程：実践を行う段階で階段や坂に対応することができなかつたため、コースの一部である食堂前から出入り口までの平坦な道を実際に歩行させることで平坦な道であれば目的地まで歩行が可能であることを確かめた。

(※文責：垂石哲平)

第 4 章 課題解決のプロセスの詳細

4.1 各人の課題の概要とプロジェクト内における位置づけ

赤坂駿斗の担当課題は以下のとおりである。

- 5月 Python を使った強化学習を学習。
- 6月 Raspberry Pi の設定。
- 7月 socket 通信の導入。
- 9月 Python を使ったロボットのモーション再生プログラム作成。
- 10月 強化学習プログラムの考案。
- 11月 強化学習プログラムの作成、実験。
- 12月 最終発表に向けての準備。
- 1月 最終報告書の作成。

(※文責：赤坂駿斗)

垂石哲平の担当課題は以下のとおりである。

- 5月 ロボットの挙動の学習。
- 6月 Raspberry Pi の設定。
- 7月 socket 通信の導入。
- 9月 強化学習についての学習。
- 10月 強化学習についての学習、実装方法の考案。
- 11月 プログラムの作成、実験。
- 12月 最終発表に向けての準備。
- 1月 最終提出物の作成。

(※文責：垂石哲平)

湊宏太郎の担当課題は以下のとおりである。

- 5月 unity のシミュレーション、Python を使い強化学習を学習。
- 6月 Raspberry Pi の設定。
- 7月 socket 通信の導入。
- 9月 強化学習についての学習。
- 10月 強化学習についての学習。
- 11月 プログラミングの作成、実験。
- 12月 最終発表に向けての準備。
- 1月 最終報告書の作成。

(※文責：湊宏太郎)

古里優典の担当課題は以下のとおりである。

Challenge to Complex Intelligent Robot

- 5月 OpenCV の学習。
- 6月 Raspberry Pi の設定。
- 7月 動画上で色認識プログラムの作成。
- 9月 Python を使用してロボットを動かすプログラムの作成。
- 10月 画像認識プログラムの作成。
- 11月 強化学習の実験。
- 12月 最終発表に向けての準備。
- 1月 最終提出物の作成。

(※文責：古里優典)

4.2 担当課題解決過程の詳細

4.2.1 赤坂駿斗

- 5月 自律するロボットを開発する上で、強化学習によるモーションの制御がアイデアとして上がった。それを実装するために、Python の OpenAI Gym ライブラリを用いて学習した。この際、元々の A グループ、B グループではなく、プロジェクトのメンバーを三つのグループに分けて学習を行った。強化学習を学習する私たちのグループは、機械学習の複数のプログラムを紐解き、学習した。
- 6月 ロボットに搭載する予定の Raspberry Pi が届いたため説明書の通り組み立てた。セットアップに関しては初期設定とライブラリの導入をして外部の PC から操作できるようにした。さらに、カメラモジュールで写真や動画を撮れるようにした。加えて、Raspberry Pi がフリーズすることが多いため、フリーズしたときに外部の PC から再起動を行えるようにした。
- 7月 Raspberry Pi と PC で無線通信を行うにあたり指導教員に話を聞いたところ、socket 通信技術を勧められたため、Web サイトを参考にしながら socket ライブラリを導入するなどして学習した。
- 9月 前期の時点で、自律するロボットを開発する上で、強化学習によるモーションの制御がアイデアとして上がっていた。強化学習を行う前の大前提として、ロボットを動かすプログラムが必要となった。そのためプログラム言語は、モーション再生のサンプルプログラムが存在し、前期に作成した画像認識のプログラム言語である Python を使用した。プログラムは、HeartToHeart4 というモーション作成ソフトによって割り振ったモーション番号を指定して再生することができる。ロボットを動作させるプログラムを作成することで、Python の基本的な使い方を再度学ぶことができた。当初はこのプログラムを Raspberry Pi というシングルボードコンピュータを利用して無線化をしようとしたが、強化学習の実験段階において、ロボットを USB 接続した状態で実験すると効率がいいと判断し、有線接続で動かせるプログラムを作成した。
- 10月 9月時点でロボットを PC との有線接続でプログラムによるモーション再生を可能とした。そのため、このプログラムを利用した強化学習プログラムを考案した。考案にあたって、グループの中で9月に強化学習について勉強したメンバーから強化学習のモデルをアイデアとしていくつか出してもらった。その中から一つのモデルに絞り、実装するプログラムの内容について会議した。具体的には、強化学習の1つのモデルを9月に作ったプログラムを利用

して Python で強化学習をしつつ、ロボットを動かせるような仕様にした。複数人での会議では度々議題から外れたような会話になることがあることが分かったため、自分が意識して議題から外れた質問やアイデアが出たときは修正した。

- 11月 10月時点でロボットを使った Python による強化学習プログラムの基礎部分ができていたので、前期で作成した画像認識のプログラムと統合して、カメラを用いた強化学習のシステムを構築した。また、実験を重ねることでプログラムの精度を上げていった。この学習段階では、Python の実践的なコードの書き方とプログラムを作成する上での心構えを学ぶことができた。また、画像認識はカメラや光源、目標物の素材によって精度が変わることがあり、どのような素材に変更すればよいか適切なアイデアを出すことができた。
- 12月 ポスターと発表用原稿を作成した。また、発表をよりわかりやすいものとするために、ロボットが強化学習によって動いている実験のシーンの動画を見せるために撮影をした。発表資料を作成した後にグループで発表練習をした。発表用原稿で自分たちの伝えたい事が分かりやすく伝わっているかという部分に重点を置き、改善することができた。発表者の分担はグループ全員で行い、リハーサルを繰り返し行うことで本番でも自分の役割を全うすることができた。
- 1月 最終報告書と学習フィードバックシートを作成した。最終報告書の作業分担に関しては、メンバーと話し合いで決め、期日に自分が文責である部分の文章を持ち寄り全員で誤字脱字等のチェックをした。前期よりも期日までの時間を長く取ったことでグループメンバーの書いた文章を読んでお互いに修正することができたので、よりよい文章を書けた。また、自らやグループメンバーの活動を評価することで今までやってきた活動の意味や価値を認識することができた。

(※文責：赤坂駿斗)

4.2.2 垂石哲平

- 5月 指導教員によって提示された Go Simulation! Demo というソフトの「モーション設計」機能を用いて学習を行った。この際、元々の A グループ、B グループではなく、プロジェクトのメンバーを三つのグループに分けて学習を行った。まず、同じグループのメンバーと共にどのような挙動を誰が作るのかについて話し合っただけで決めた。また、定期的に他のグループと経過報告を行い、習得した技術や知識についてプロジェクト全体で共有すると同時にフィードバックを得た。ロボットの各関節の動き方や役割、重心の挙動を把握し、ロボットの基礎的なモーションを作成する技術を習得した。これ以降のグループ活動は元の A グループ、B グループに分かれて行った。
- 6月 同じグループのメンバーと共に、購入した Raspberry Pi について付属していた取り扱い説明書を読みながら組み立てた後、Web サイトを参考にしながら初期設定や OpenCV ライブラリの導入を行った。プログラミング言語の Python を扱うにあたり「情報処理演習 I」で学んだ Python の取り扱い方法を活用した。また、Raspberry Pi 本体に専用のカメラモジュールを取り付け、プログラム上でコマンドを用いて実際に写真や動画を撮影することで正常に稼働するかどうかを確かめた。Raspberry Pi 内で Python で書かれたプログラムを動かす方法とカメラモジュールの取り扱い方法、OpenCV ライブラリの使用方法を習得した。
- 7月 Raspberry Pi と PC で無線通信を行うにあたり指導教員に話を聞いたところ、socket 通信

Challenge to Complex Intelligent Robot

技術を勧められたため、Webサイトを参考にしながら socket ライブラリを導入するなどして学習した。また、通信プロトコルである TCP/IP を採用するにあたって講義「情報ネットワーク」で学んだ通信プロトコルに関する知識を用いた。socket 通信技術の実装にあたり、メンバー同士の PC と web カメラを用いて文章や画像の送受信が行えることを確認した後、Raspberry Pi に socket ライブラリを導入し、実装した。socket 通信を介して文章や画像といったデータの送受信に関する技術を習得した。

- 9月** web サイトや情報ライブラリーの書籍を用いて強化学習に関する学習を行った。強化学習の基礎的な理論や位置づけといった知識を学んだ。また、書籍内のサンプルプログラムについて自身の PC 上で同じものを実装することで Q ラーニングやニューラルネットワークの働きを直感的に学び、応用方法を身につけた。強化学習に関する理解を深めるにあたって講義「人工知能基礎」で学んだ機械学習の知識やプログラミング技術を参考にした。
- 10月** 9月に引き続き学習を進めた。また、自身が得た知識をグループ内で共有し、自分たちのプロジェクトにおいてどのような形で強化学習を取り入れるべきかの意見を述べ、話し合いを行った。話し合いやプログラムを作成していく中でメンバーから出た質問についてもできるだけ素早く正確に答えることで強化学習に関して全員がある程度の理解ができるように意識した。自身の知識を他者に正しく伝えるための能力や、他者からの意見を聞き、できるだけ取り込む客観性と柔軟性を養った。
- 11月** 学習を行う際に使用していたサンプルプログラムをもとに強化学習を行うプログラムを作成した。また、前期で作成した画像認識のプログラムとロボットを操作するためのプログラムを含めた3つのプログラムを統合し、ロボットに強化学習を用いてモーションを選択させるプログラムを作成した。具体的には、強化学習の評価値や状態遷移に画像認識から得られた値を用いるようにし、強化学習で選択する「行動」としてロボットに設定されているモーションを再生するようにした。その後、作成したプログラムを用いて実際にロボットを動かし、実験を行った。Python 上でプログラムを扱うための実践的知識を学ぶことができた。また、光度など実験を行う上での注意点について体感することで、実験の障害となる可能性のあるものを考えるための予想の立て方や対処法の考え方を身につけることができた。
- 12月** ポスターや発表用原稿を作成し、最終発表に備えた。その際、グループ内で分担して作業を行うことで効率的に作業を進めていくことができた。制作していく中で、中間発表で使ったものを参考にしたり、メンバーと確認を取り合いながら作成することで正しい内容になるように工夫した。その後、完成した原稿を読み込み、自分たちの伝えたいことが過不足なく聴衆に伝わるかを確かめた。また、グループ間で発表を評価しあい、フィードバックすることで発表の完成度を高めた。自分の伝えたい意見をまとめる能力や必要に応じて説明を付け足したり省いたりする能力を養うことができた。
- 1月** 報告書、学習フィードバックシートといった最終提出物を作成した。前期ではスケジュールの確認を怠っており期限の直前に取り組むことになってしまったが、後期ではその反省を活かし、スケジュールを細かく確認しながらプロジェクトを進めることで余裕を持って提出物作成に取り組めるようにした。スケジュール管理の重要性や自らの活動を評価する観点を学んだ。

(※文責：垂石哲平)

4.2.3 湊宏太郎

- 5月 自律するロボットを作成する上で強化学習を学習することが必要であると考えた。この際、元々の A グループ、B グループではなく、プロジェクトのメンバーを三つのグループに分けて学習を行った。まず始めに unity による強化学習を学んだ。unity を使いゲームを通して強化学習を学ぶことで、より理解しやすく、また、ロボットを動かす上で役に立つと考えたためである。unity の基礎知識から学び ML-Agents を使って実際にゲームを動かすことで、強化学習について学んだ。その後 Python を使い、Q 学習などの知識について学んだ。
- 6月 同じグループのメンバーと共に、購入した Raspberry Pi について、付属していた取り扱い説明書を読みながら組み立てた。その後、web サイトを参考にしながら Raspberry Pi の各種設定を行った。また、Raspberry Pi 本体にカメラモジュールを取り付け、プログラムを用いて写真や動画を撮影した。
- 7月 Raspberry Pi の処理速度が遅いことや、一つの PC でプログラムを処理する必要があったため、Raspberry Pi から PC に画像を送る必要があった。どのような技術を使うことが出来るかを調べたところ socket 通信という技術が見つかったため、socket 通信を学習した。Python の socket 通信を使い、初めはメンバー同士の PC 間で動画や画像などを送ることが出来るように学習し、その後実際に Raspberry Pi で撮った写真を Raspberry Pi から PC に送ることが出来るよう学習した。
- 9月 AI 班で今後どのような技術を学び、実際にロボットを動かしていくかをメンバーと話し合った。その後、長期休みの間に学習した強化学習の勉強を引き続き行った。そこでまだ勉強が足りない部分などをピックアップし、どのような技術を使えばロボットに強化学習を適応出来るかを話し合い、今後学ぶ物を決定した。
- 10月 9月に引き続き、強化学習の勉強をした。強化学習に対しての理解をさらに深めることや、強化学習をロボットに対して使えるようにプログラミングに落とし込む方法について本を参考にし考えた。また、メンバーとわからないところを話し合い、お互いに理解出来るように話し合った。
- 11月 引き続き強化学習をプログラムで動かす方法について学んだ。ロボットを強化学習するプログラムが完成し、他のメンバーが作ったロボットを動かすプログラム、画像処理をするプログラムを統合し強化学習の実験を行った。実験を行いプログラムの不具合や実験環境の不具合、ロボットの調整などを行った。強化学習には多くのデータが必要であるため、実際にロボットを動かしデータをとった。
- 12月 実験のデータが集まって行き、強化学習で使う報酬の調整も順調に進んで来たため、実際に想定していたコースや、色認識しやすい明るさの部屋でロボットを動かした。また、最終発表に向けポスターや動画の作成などを行った。発表では強化学習という難しいトピックを扱っているため、傍聴者にもよく理解してもらえるようにポスターや発表の原稿などを工夫して作成した。また、ロボットの強化学習の実験の動画や、ロボットが実際に動いている動画を傍聴者に見せる事で、よりわかりやすく伝えることが出来た。
- 1月 最終報告書を作成するために、グループ間で活動内容や成果物の認識にズレなどが無い話し合った。前期では報告書を書く時期が遅く、メンバーと話し合う時間をあまりとる事が出来なかったため、その反省を踏まえ早いうちから報告書に取り掛かる事にした。これにより余裕を持って文章を完成することが出来、自分達がやってきた活動の反省や意味などを再確認することが出来た。その後それらをまとめ、報告書を完成させた。

4.2.4 古里優典

- 5月 指導教員に OpenCV というライブラリの存在を教えてもらい、OpenCV の学習を行った。この際、元々の A グループ、B グループではなく、プロジェクトのメンバーを三つのグループに分けて学習を行った。まず初めに、使い慣れている Java 環境で OpenCV について学ぶことで OpenCV に慣れることができた。続いて、ロボットや Raspberry Pi で使用できる Python 環境で OpenCV を使用することを学んだ。そのおかげで、Java 環境と Python 環境で顔認識を動画上で行うことができた。また、顔認識以外の OpenCV の機能をサンプルプログラムで実行することができた。これ以降のグループ活動は元の A グループ、B グループに分かれて行った。
- 6月 同じグループのメンバーと共に、購入した Raspberry Pi について付属していた取り扱い説明書を読みながら組み立てた。その後、Web サイトを参考にしながら初期設定や OpenCV ライブラリの導入を行った。また、Raspberry Pi 本体にカメラモジュールを取り付け、プログラムを用いて写真や動画を撮影することで正常に稼働するかどうか確かめた。Raspberry Pi 内でプログラムを動かす方法と カメラモジュールの取り扱い方法を習得した。
- 7月 Python 環境で動画上の色を認識するプログラムを作成した。しかし、座標を求めることができなかったので枠を作成し、枠の座標を表示するようにした。また、複数の色を認識するために、マスクデータを 3 つ作成し 3 つの色を同時に認識するようにした。続いて、画像を認識し任意の色の大きさを表示するプログラムを作成した。そのほかにも、中間発表に向けてスライドを作成したり発表準備を行った。中間報告書の作成を行い、グループ内での後期への課題などを決めた。
- 9月 今後の最終発表までに解決すべき課題や実装すべき機能についてまとめた。Python を使用し、専用のアプリケーションなしでもロボットが動作を行うようにプログラムを作成した。ロボットに登録されている通信速度を変更することで Python でもロボットと通信を行うことが可能となった。また、パソコンを使用せずに Raspberry Pi でプログラムを動かせるようにした。
- 10月 7月に作成した画像認識プログラムを強化学習の実験で使用できるように改良を行った。今まで別画面で認識していた 3 つの色を 1 つの画面でも表示できるようにした。ロボットとゴールの色を決め、別々のものとして認識できるようにした。枠の座標を使用し、認識した物体の中心の座標を取ることができるようになった。上カメラを使用することで、座標からロボットとゴールの距離を求めることができるようにした。ロボットが転んだ際に仰向けかうつ伏せ化を判断することができるようにした。横カメラを使用することで、ロボットが転んでいるかどうか判断することができるようになった。
- 11月 強化学習の実験ではロボットを使用して行った。実際に実験を行うと、光の反射によってこちらの想定とは違う色を認識してしまい誤った学習を行ってしまうことがあった。そのため、なるべく色を反射しない素材を使用することで光の反射を抑えることに成功した。また、実験によって見つかった問題点の修正と改善を行った。画像認識を使用して実験を行うにあたり、周囲の環境に大きく影響されてしまうということを学んだ。
- 12月 最終発表に向けて、ポスターや発表練習などの準備を行った。また、最終発表で使用する動画の作成を行った。画像認識や強化学習について知らない人にでも上手く説明することが

Challenge to Complex Intelligent Robot

できるように気を付けて練習を行った。実際の発表では動画やポスターを有効活用し、多くの人に分かりやすいと言ってもらえる発表を行うことができた。

- 1月 最終提出物の作成を行った。前期での反省を活かし、役割分担や作業計画を早めに立てることができたため、時間に余裕をもって最終提出物の作成を行うことができた。また、前期に比べてかなりスムーズに話し合いなどが進むようになり、前期よりもグループの団結力が高まったと感じることができた。

(※文責：古里優典)

4.3 担当課題と他の課題の連携内容

4.3.1 赤坂駿斗

Aグループ内での連携としては、それぞれのメンバーが作成したプログラムを統合したことが挙げられる。また、統合後は改善に関してアイデアを出したり細かい修正に関わった。Bグループとの連携としては、1体のロボットを相談して交互に使った。Bグループが主に使用していたソフトウェアをこちらが使う際にモーションの設定などを委託することで効率化した。

(※文責：赤坂駿斗)

4.3.2 垂石哲平

Aグループ内での連携としては、強化学習のプログラムに合わせて画像認識のプログラムに改善案を提示したり、場合によっては自ら画像認識のプログラムを調整するなどした。Bグループとの連携としては、Raspberry Pi 及びカメラモジュールをロボットに搭載するかどうかについてサーボの出力やロボットの重心の不安定さといった問題点があることなどをBグループの意見を参考にしながら話し合い、決めた。また、Bグループがロボットに実装するモーションをどのようなものに設計するのかについて、Aグループが想定している強化学習の内容に沿うように要望を出した。実験中に明らかになったモーションの問題点についてはすぐに報告し、その場でモーションの修正を行ってもらったようにした。グループ間での話し合いの際、事前学習で得たモーション作成の知識が役に立った。お互いの作業に関して知識があるのとないのとでは出せる要望の正確さに大きく影響が出ることを学んだ。

(※文責：垂石哲平)

4.3.3 湊宏太郎

Aグループ内での連携として、Aグループ全員で色認識をするプログラム、ロボットを動かすプログラム、強化学習を行うプログラムを統合して一つのプログラムとして動くよう完成させた。実験を行う中で、色認識が上手くいかない事や、強化学習の報酬の設定などが上手くいかないことがあったので、それらを調整しプログラムを納得出来るものに仕上げる事が出来た。これらのプログラムを理解し、最終発表に向けてまとめる必要があったためメンバーと話し合い、どのようにプログラムが動いているかを理解し、発表するためにポスターや動画としてまとめた。Bグループとの連携では強化学習する際、想定したコースを歩くことが出来るようにどのようなモーションを必

Challenge to Complex Intelligent Robot

要とするかを話し合い、Bグループの意見も参考に提案した。また、Raspberry Piなどを搭載するのにどのようにロボットに装着するかについても話し合った。

(※文責：湊宏太郎)

4.3.4 古里優典

Aグループ内での連携として、画像認識のプログラムを作成する際に、画像認識に必要なシステムのアイデアを出し合った。また、強化学習のプログラムと統合する際に、不必要な情報や必要な情報を分けプログラムの改善を行った。Bグループとの連携では、Heart to Heart4に関する情報の分析やシミュレーションソフトを使用してモーション作成の手助けを行った。また、実際に実験で使いたいモーションについてや実験を行った際に生じた問題点などを確認しあった。ロボットにRaspberry Piを搭載する際に、Bグループと話し合いを行った。

(※文責：古里優典)

第 5 章 結果

5.1 プロジェクトの結果

ロボットにカメラを搭載し、前方にある障害物や道を認識するために主観カメラのプログラムの作成をした。主観カメラには Raspberry Pi のカメラモジュールを用いた。カメラで撮った画像を Raspberry Pi から別の PC に送るプログラムを Python の socket 通信を用いて作成した。撮った画像を別の PC に送るのは Raspberry Pi の処理速度が遅いためと、一つの PC で全てのプログラムを管理したかったためである。撮った画像に画像認識を適用し、障害物や道に色テープや色付きの布を貼ることで障害物や道を判別出来るようにした。

ロボットがいる位置とゴールの位置を把握するために、客観カメラのプログラムを作成した。客観カメラでは 3 色の色（赤、青、黄色）を認識し、それぞれの座標を表示するプログラムを作成した。客観カメラは上に取り付け、ロボットとゴールの座標を管理した。客観カメラはロボットの実験にも用いた。実験する時にはロボットとゴールの位置を把握する事、また、ロボットが転倒しているか否かを判断する必要がある。そのため、上カメラだけではなく、横カメラも実装した。ロボットの肩や頭の部分、足の裏、胸部などに色テープや色付きの布などを貼り、色認識を行った。色付きの布を使うのは、色テープの場合、光が反射してカメラで色を判別することが難しくなるためである。色認識する時に色を多く使うと、各色の認識出来る色の範囲が狭くなってしまいうため、色の判別が難しくなってしまう、異なる色でも同じ色と認識してしまう可能性があった。そのため色認識しやすい 3 色を用いるようにした。また、カメラが床や背景に対して誤作動を起こすことがあった為、ある程度の大きさより大きい物だけ色認識できるようにした。上カメラからはロボットの頭の部分につけた布と、ゴールに見立てた布を色認識し X 座標、Y 座標で管理出来るようにした。横カメラからはロボットが転倒したか否かを判断するために、ロボットの肩と足の裏に色付きの布を貼り、肩についている色付きの布の座標と、転倒した時に見える足の裏の布を座標で管理出来るようにした。座標で管理する事により、ある数値以下の時は転倒していると判断出来るようにした。また、横カメラで転倒を判断した際に、上カメラではロボットの胸部に貼ってある青色の布を認識すると、ロボットが仰向けに転倒したと判断し、そうでない場合はうつ伏せに転倒したと判断出来るようにした。

ロボットをリモコンなどの人による操作なしで動けるようにするため、ロボットを動かすプログラムを作成した。ロボットのモーション作成ソフトの HeartToHeart4 で定めたモーションを再生することが出来るようにロボットに搭載されているマイコンボードへ信号を送る事のできる Python のプログラムを作成した。マイコンボードに特定の数字を送るとその数字に合わせたモーションをロボットが実行出来るよう作成した。

前期の最後に想定していたコースが変わり、目標とするルートとして、はこだて未来大学の 2 階の購買から階段を降り、食堂横の出入口から屋外に出て、最寄りのバス停をゴールとするルートとなった。これを解決するためにロボット班がゴールにたどり着くために必要と考えられるモーションを作り、AI 班がロボット班の作ったモーションを状況に合わせてロボットが選択することが出来る強化学習のプログラムを作成することとなった。強化学習には Q 学習と ϵ -greedy 法を用いた。 ϵ -greedy 法を採用した理由は、行動を選択する際、ランダムな行動を採用することで初めの方に採用した Q 値の影響を受けすぎないようにするためである。この方法を採用することで、初

Challenge to Complex Intelligent Robot

めに選択したモーションの影響を受けすぎないように、効率的に学習を行えるようになった。Q 値は今回の実験ではロボットが6回行動したものを1セットとして強化学習を行った。Q 値に使う値を、ロボットの転倒に関する報酬を2つ、ロボットの歩行に用いる報酬を3つ、ロボットがゴールのそばに近づいたときに用いる報酬を2つ、計7つ用意した。転倒に関する報酬では、ロボットが転倒した時にすぐ起き上がることが出来るようにする事と、ロボットが転倒しないようにするために、転倒している時はマイナスの報酬を与え、転倒から起き上がるとプラスの報酬を与えるよう設定した。ロボットの歩行に用いる報酬では、ロボットがゴールまで真っ直ぐ進むよう、直線状から外れるとマイナスの報酬を、直線状から外れた状態から直線状に戻った時はプラスの報酬を与えるよう設定した。ロボットがゴールに近づく時、実践で使う際はそこから別の動作をしようと考えていたため、ゴールにたどり着くと止まることが出来るよう報酬を設定した。そのためゴールから接近している状態から離れた状態になった時負の報酬を与えた。報酬を与える時、大きすぎる報酬を与えてしまうとその結果がQ 値に大きく影響してしまい上手く強化学習することが出来ないため、距離などで報酬を与える時は小さくしてから与えた。また負の報酬のみを与えて実験をすると外的要因やロボットの不具合によって動作が失敗した時に、意図せぬ報酬が与えられてしまうことがあったため、正の報酬も与えるよう設定した。また、この実験で実装した行動一覧を表 5.1 に、実装した状態一覧を表 5.2 に示した。この表 5.1 はロボットのマイコンボード内に表中の Action の項の番号に右の行動が設定されている事を表している。また図 5.1 として学習の試行回数とゴールの残りの距離のグラフを示した。このグラフは強化学習を 50 回行った結果で、実験でロボットを1回行動させ終わった時のロボットとゴールの距離を縦軸に、試行回数を横軸として設定した図である。

実験にはロボットを動かすプログラミング、客観カメラのプログラミング、強化学習のプログラミングを統合したプログラミングを用いた。上からカメラを撮るためにカメラを取り付ける機材なども用意し、照明の明るさなどが実験に適している環境や、実際に想定したコース上で実験を行った。

表 5.1 実装した行動一覧

Action	各ロボットの動作
0	低速歩行
1	高速歩行
2	左サイドステップ
3	右サイドステップ
4	うつ伏せからの起き上がり
5	仰向けからの起き上がり
6	挨拶

表 5.2 実装した状態一覧

State	各状態のロボットの状況
0	ゴールとロボットが一直線上にある状態
1	ロボットが左にずれている状態
2	ロボットが右にずれている状態
3	ロボットがうつ伏せで倒れている状態
4	ロボットが仰向けで倒れている状態
5	ロボットがゴールに接近した状態

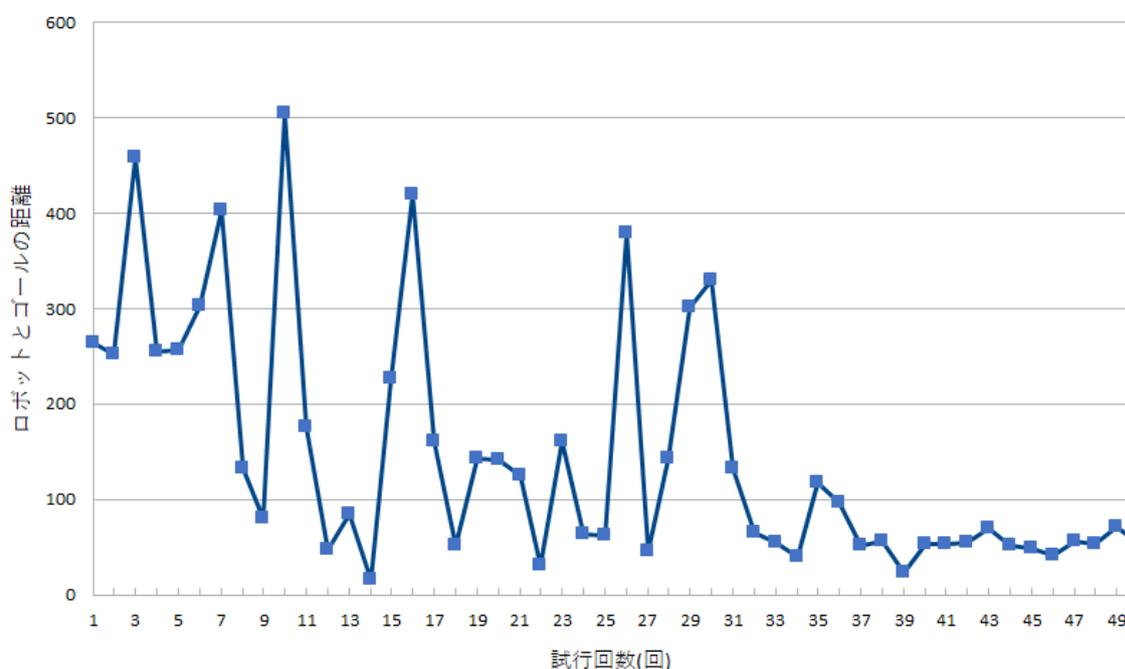


図 5.1 50 回試行した結果

(※文責：湊宏太郎)

5.2 成果の評価

今回はロボットに Raspberry Pi を実際に搭載することをしなかった。理由として、ロボットに取り付ける Raspberry Pi 自体がとても重くロボットが起き上がるなどのモーションを行う時に起き上がることができない点や、Raspberry Pi の画像処理速度などが遅く適切に動かないことが挙げられたためである。これらの問題はロボットのサーボをより強いものにする事や、より良い性能の小型コンピューターを使用する事により解決できると考えた。

客観カメラは実験に使う時も、実践に使う時もうまく動いた。実験に使う時、色が蛍光灯の光で反射してカメラが色を認識しなかったり、想定していない物をその色と認識してしまうなど調整すべき課題は出てきたが、光を反射しにくい布を使用することで概ね上手くいった。実践の時も人が上から撮るカメラや横から撮るカメラ、ゴールを動かす事でロボットがスタートからゴールまでを移動することが可能になった。

Challenge to Complex Intelligent Robot

ロボットを動かすプログラミングでは、ロボットに信号を送りロボット班が作ったモーションを再生することができた。しかし、ロボットのサーボモーター一つ一つを直接動かしロボットにモーションを行わせるプログラミングを作成することはできなかった。この課題を解決できる方法はまだ見つかっていないが、マイコンに送るコマンドの信号を解析することができれば、サーボモーター一つ一つを直接動かす信号が見つかる可能性がある。そうすることで、サーボモーターの問題は解決できるのではないかと考えた。この方法を使うと強化学習もまた新たな手法で行うことが考えられ、より自律して動くロボットが完成すると考えた。

強化学習のプログラムは最低限実験を行うことができるものに仕上がったと考えた。実験を始めた当初は、思い描いた結果が得られなかったが、報酬の与え方を工夫した事や、正の報酬も同時に与えた事、色付きの布を使い色認識をさせた事でうまく色を認識することができるようになった事で、徐々に到達点に近づいてきた実感を得られた。今ではロボットが外的要因や不具合によって予期せぬ動作をしても思考を重ねていくうちに上手く想定通りの結果にたどり着くようになった。

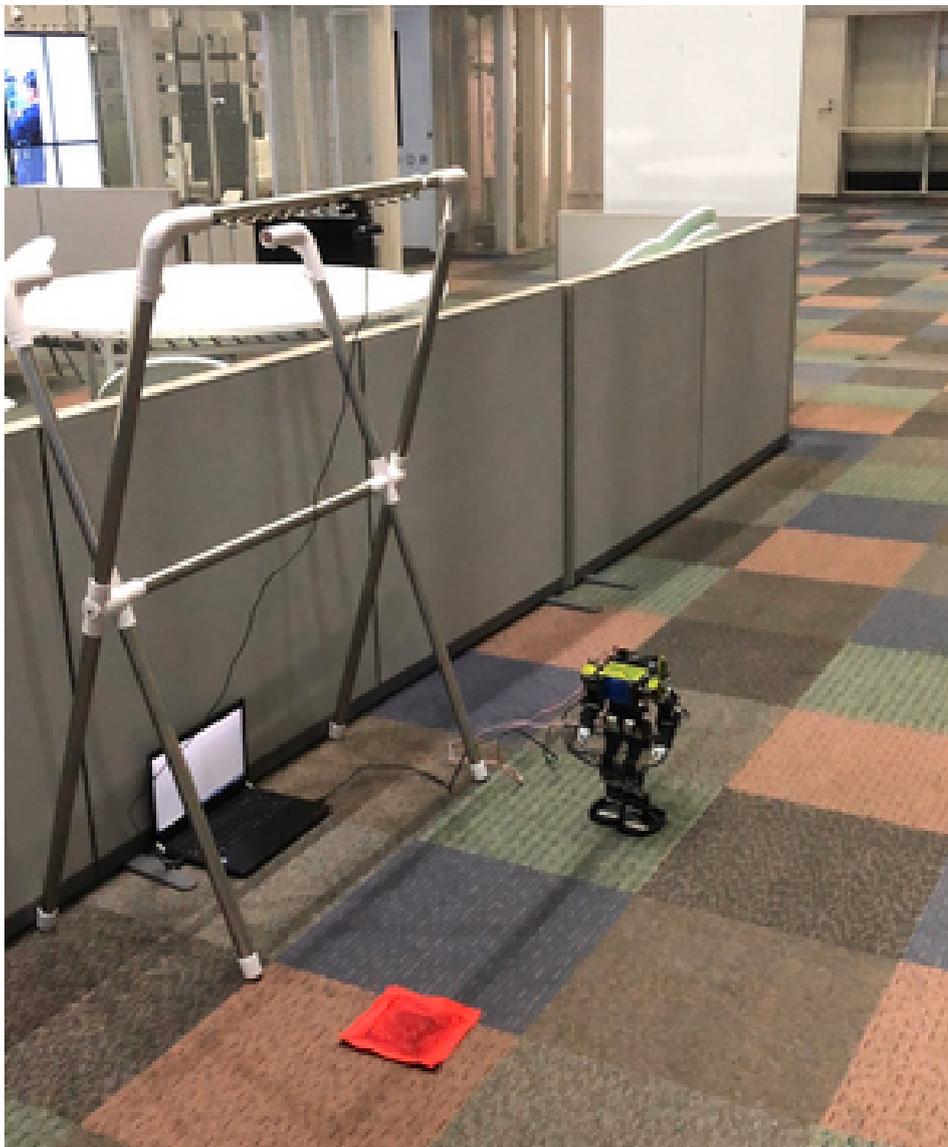


図 5.2 実験風景



図 5.3 上からの様子

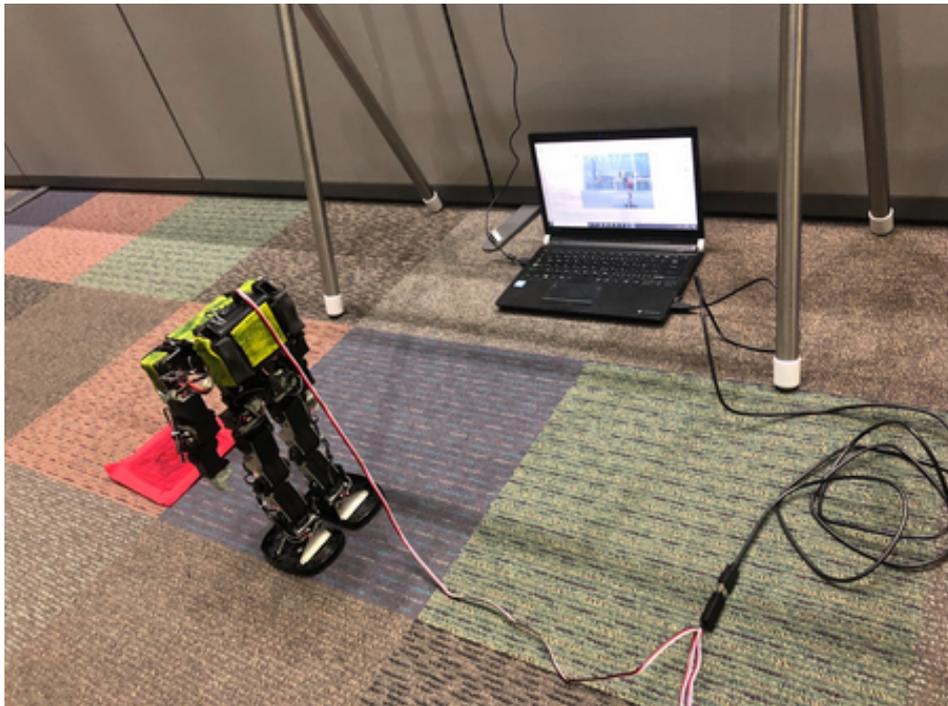


図 5.4 横からの様子

実践では平面でのロボットの歩行は、人間が、上カメラや横カメラ、ゴールを持ち、動かす事により達成することができた。人間の力を借りないでスタートからゴールまでをロボットが進むのは、今とは違った客観カメラの実装などが必要であると考えられ達成する事はできなかったが自分達では納得いく物に仕上がった。しかし、当初想定していた階段の上り下りや、坂道の歩行などは、モーションが完成していない事や、ロボットのサーボの出力が足りない事、また実践を試みた時期に道に雪が積もっていたなどの環境の問題もあり達成する事はできなかった。これを達成するためには、よりパワーや耐久性のあるロボットを用いる事や、別のアプローチからのモーションの作成、環境が整った時期に行うなどの方法で解決できると考えた。

(※文責：古里優典)

5.3 担当分担課題の評価

5.3.1 赤坂駿斗

私は主に、前期の前半は強化学習の実装に向けて学習を進めた。実装できるまでの知識と技術を蓄えることができなかったため、後期では教員の協力を得ながら実装を目指したい。前期の後半は、Raspberry Pi の初期設定とカメラモジュールの調整を主に担当した。別の PC からネットワークを介し、Raspberry Pi の操作をする部分に関しての知識が不足しているため、さらなる操作をできるようにしたい。後期では主にロボットをプログラミングで動かせるようにするプログラムを作成し、そのプログラムを用いてロボットを強化学習で最適な行動をとるようにすることに取り組んだ。前半はロボットを動かすプログラムの作成をした。エラーの除去に大きく時間をとられてグループに迷惑をかけてしまった。しかし、教員と相談しながら取り組めたため、Python のプログラムに関して、自分では気づけなかったことや知らなかった知識を身に着けることができたと感じる。予想より時間はかかってしまったが、ロボットをプログラミングで動かすことは強化学習

Challenge to Complex Intelligent Robot

プログラムに必須であるため、完成させることができた点は評価できると考える。後半は、ロボットの強化学習プログラムの作成に取り組んだ。自分は先のプログラムを作成している間に強化学習について勉強していたメンバーと相談してどの強化学習のモデルがいいかアイデアを出し合い決めた。自分も前期に強化学習の勉強をしていたがこの会議がより理解を深めるきっかけとなった。その後、会議によって決まった強化学習のモデルに沿うようにロボットがモーションを選択するようなプログラムを簡易的に作成した。期日は無かったが初期モデルとして早急にプログラムを組めたのが良かった。それ以降のロボットを用いた実験とプログラムの改善は主に別のメンバーが行っていたため、自分は問題点を取り上げ、改善に関するアイデアを出す等してプロジェクトに関わっていた。最終発表はグループメンバーと話し合ってポスターの作成を分担して作成した。前後期合わせての総評として、グループでの会議での役割を大きく果たせたのがよかった。元々、議題に対するアイデアを出すことが得意だったので多くのアイデアを出すことができた。また、本プロジェクトの活動を通して会議をスムーズに進行させる役回りをする事で、よりグループメンバーのアイデアを引き出すことができ、そのような力をつけられた。

(※文責：赤坂駿斗)

5.3.2 垂石哲平

前期の活動の中でも前半には GoSimulationDemo1.3.4 を用いてロボットのモーションの学習をすることに力を入れた。重心や関節の動きなどロボットの挙動についての知識と技術を身につけることができた。後半は主観カメラのプログラム作成に力を入れた。カメラモジュールを取り扱うにあたり、Raspberry Pi 内に OpenCV ライブラリをインストールし、写真や動画を撮影できるようにした。また、socket ライブラリをインストールし、Raspberry Pi と PC 間で socket 通信を行えるようにした。それにより、カメラモジュールで撮影した画像を送信することが可能になり、Raspberry Pi の負荷を減らすと同時に別の PC で色認識などの画像処理を行うことを可能にした。後期の活動では、前期の活動結果を踏まえて先をしっかりと見据えて活動すべきだと考えた。そのため、プロジェクト内でも特に重要だと考えられる強化学習の実装について積極的に取り組むように心がけ、プロジェクトを進めた。その甲斐もあってか、プログラムを作成するにあたって必要な強化学習に関する知識と技術をしっかりと身につけることができたように思う。グループ内、グループ間共に強化学習に関する話し合いでは自分が中心となり率先して意見を出し、全体を引っ張ることができた。これらの点から、全体を通してプロジェクトに良く貢献できたと考えるが、現状では対応できる道に制限がある点や評価値を手動で調整する必要がある点など改善すべき点が多い。

(※文責：垂石哲平)

5.3.3 湊宏太郎

前期の前半には強化学習について学習を進めた。強化学習は、本を用いる事や、実際に強化学習を使ったゲームをプログラミングで動かす事により学習した。強化学習については少しだけ理解することが出来たが、実際にプログラミングとして他のものに应用することは出来なかった。前期の後半には Raspberry Pi で写真を撮る事や、撮った写真を他の PC に送るために socket 通信のプログラムを開発した。初めのうちは画質が低いものしか送れなかったり、画像が文字列に直された物

Challenge to Complex Intelligent Robot

を送るなど、上手くいかないことが多かったが、最終的に Raspberry Pi で撮った写真を他の PC に送る事に成功した。後期には主に強化学習の学習に取り組んだ。前期で学んだ事や、長期休みの間に学んだことをまとめ、より強化学習についての理解を深めた。また、強化学習をプログラムで動かす方法も学んだ。実際に強化学習をロボットが行うプログラムを作成することは出来なかったが、強化学習について深く理解することが出来、さらに、グループのメンバーが作成した強化学習のプログラムを読むことで、どのようにプログラムを作るかや、評価値に関する報酬の設定の方法なども学ぶことが出来た。またこれまでに学んでいた強化学習のアイデアを出す事で強化学習の実験を効率的に進めることが出来たと考える。

(※文責：湊宏太郎)

5.3.4 古里優典

前期では、Python の OpenCV ライブラリを使用して、動画上でゴールとロボットを認識するプログラムを作成するために、色認識を活用した。それにより、ゴールとロボットとの座標や距離を求めることができた。しかし、ゴールまで歩く際に転んだ場合などの判別ができないため、判別方法を考えていきたい。また、Raspberry Pi を使用してカメラで動画や写真を撮ることができるようになった。後期では、強化学習の実験に向けたプログラムの作成を行った。例えば、Python を使用してロボットを動かせるプログラムや上カメラや横カメラの画像認識プログラムを作成した。画像認識プログラムは、3 色の色を認識することができる。またそのほかにも、認識した色を枠で囲む、認識した色の距離を測定する、ロボットが転倒したかを判断する、ロボットが仰向けに転倒したかうつぶせに転倒したかを判断することができる。そのため、強化学習の実験での報酬を決める指標や、実際にどのようにカメラが認識しているかを調べることができた。そのほかにも、Raspberry Pi の設定や実験環境の作成など成果物以外の行動も行った。また、前期の活動で判明した問題は、上カメラと横カメラを使用することで解決することができた。しかし新たな問題として、色認識自体がその場の環境に大きく影響されてしまうことが分かった。そのため、色認識だけでなくセンサーや GPS などを使用し、色認識以外での周囲の状況の認識が必要である。

(※文責：古里優典)

第 6 章 今後の課題と展望

今後の展望として、課題が3つある。1つ目はロボットの完全な自律化である。現在は、パソコンから直接コードでつないでロボットを動かしている。目標である購買前からバス停までは距離が長いのでロボットにコードなどをつなぐことができない。したがって、ロボットが自律しなければならない。解決策の1つとして、ロボットに Raspberry pi を搭載することでロボットを Raspberry Pi から動かすことができる。そうすると、ロボットにパソコンなどからコードを伸ばす必要がなくなり、自律化することができるようになる。2つ目は位置情報の認識方法である。現在ロボットは色認識を使用してロボットの位置とゴールの位置を認識している。しかし、色認識はカメラの性能や周りの環境に大きく依存するため、想定している認識をすることが困難である。また、カメラの視野でしか行動ができないため長距離の移動が困難である。そのために、色認識以外の位置情報を認識することが必要である。3つ目はロボットの状態の管理である。現在は色認識を使用して、仰向けに転んでいるかうつぶせに転んでいるかを判断している。しかし、どれくらいロボットが傾いているかなどを感知することができていない状態である。ロボットがどれくらい傾いているかを知ることができれば、転倒前に対策を立てたり、坂道をスムーズに歩けるようになる。自分の状態を知ることができる解決策の1つとしてロボットにセンサーを取り付けその値を入力することが必要である。これら3つの課題を達成することが今後の展望である。

(※文責：古里優典)

付録 A 中間発表の集計結果

回収したシートの枚数：56 枚

発表技術について

平均点：7.3

今後の課題

発表技術についての課題は3つある。1つ目は、スライドについてである。今回のスライドの問題点は、フォントやアニメーション、余白が統一されていない点があげられる。今後は、見やすいスライドを意識するために、統一感を出していきたい。2つ目は、発表者についてである。問題点は、発表する際に傍聴者を見ず、スライドやカンペなどを見ていた点があげられる。そのため、声が傍聴者に届いておらず聞き取りづらいという意見をいただいた。今後は、発表者は傍聴者のほうを向き、声を届ける必要がある。3つ目は時間配分についてである。問題点は、発表する時間が長く、質問が次の発表時間まで食い込んでしまっていた点があげられる。今回は質問時間を5分と定めて行っていたが、発表時間が予定よりも長くなってしまい、質問時間が5分よりも短くなってしまった。今後は、時間配分に気を配り、発表時間を厳守していきたい。以上の3つが発表技術についての課題である。

自分のグループの評価

評価点:3 (5段階評価の内)

理由

理由として、上記の3つの課題が存在するため評価点を下げた。しかし、声が大きく聞き取りやすい発表だという評価ももらえたため評価点を3とした。

発表内容について

平均点：7.7

今後の課題

発表内容についての課題は2つある。1つ目は、強化学習についてである。問題点は、強化学習についてほとんど発表することができなかった点があげられる。実際に前期ではほとんど強化学習について学ぶことができなかった。したがって、強化学習について学ぶことが今後の課題である。2つ目は、プログラムの統合についてである。問題点は、どのようにプログラムを統合するか未定である点があげられる。今後は、どのようにプログラムを統合するか具体的な案を出していきたい。以上の2つが発表内容についての課題である。

自分のグループの評価

評価点:3 (5段階評価の内)

理由

理由として、上記の2つの今後の課題が存在するため評価点を下げた。しかし、目標が分かりやすく、目標までの道筋も明確だという評価ももらえたため評価点を3とした。

(※文責：古里優典)

付録 B 最終発表の集計結果

回収したシートの枚数：60 枚

発表技術について

平均点：8.3

残った課題

発表技術についての課題は3つある。1つ目は、映像についてである。映像があることで内容が理解しやすいという意見を多数いただいたが、手振れなどにより映像が見にくいという意見もいただいた。改善案として、手振れ補正などの編集をすることがあげられる。2つ目は、AI班とロボット班の内容を両方とも見せることができなかつた点である。両グループとも伝える内容が多く、同時に発表するしかなかつた。改善案として、内容をスライドに簡潔にまとめて発表することが必要だつた。3つ目は、話し方についてである。おおむね好評であつたが、1部の人からポスターを見ながら発表をしているのでセリフを読んでいるみたいなどの意見があつた。改善案として、傍聴者を見てしゃべることで改善されると考えている。

自分のグループの評価

評価点:4 (5段階評価の内)

理由

理由として、上記の3つの課題が存在するため評価点を下げた。しかし、発表が聞きやすかつたや、熱意が伝わってきたなど肯定的な意見も多くいただいた。したがつて評価点を4点とした。

(※文責：古里優典)

発表内容について

平均点：8.2

残った課題

発表内容についての課題は2つある。1つ目は、ポスターの内容だ。アンケートでは、ポスターの文字量が多すぎて内容が伝わつてこないという意見が挙げられた。これについて、図解を増やすべきというアドバイスを得た。2つ目は、オリジナリティや何が新しい事なのか伝わらなかつたということだ。より意識的に原稿に組み込むことで伝えられるとよかつた。以上の2つが発表内容についての課題である。

自分のグループの評価

評価点:4 (5段階評価の内)

理由

「目的から課題を見つけ、それに対するアプローチが的確であつた。」というような意見を複数頂き、私たちの伝えたいことが伝わっていると感じた。また、前期の発表に比べて話したい内容が増えたため、内容を厳選して伝えることに重きをおいて原稿を作成した。しかし、先に残つた課題として挙げたように一部課題となるような意見も挙げられたため4点とした。

(※文責：赤坂駿斗)

付録 C 新規習得技術

OpenCV による画像工学

Python のプログラミング

Raspberry Pi の取り扱い全般

socket 通信による通信方法

Q ラーニングを用いた強化学習

付録 D 活用した講義

画像工学・画像の認識について

情報処理演習 I・Python の扱いについて

情報ネットワーク・通信プロトコルに関する知識と技術

人工知能基礎・強化学習について

付録 E その他製作物

強化学習を行い、ロボットにモーションを選択させるプログラムをソースコード E.1 に、カメラを起動し画像認識を行うプログラムをソースコード E.2 に記載する。

ソースコード E.1 kyouka.py

```
# coding:utf-8
import numpy as np
import sys
sys.path.append('../Rcb4Lib') #Rcb4Lib の検索パスを追加
from Rcb4BaseLib import Rcb4BaseLib #Rcb4BaseLib が使えるように設定
import time #time が使えるように宣言
import color2 #色認識プログラムのインポート
import cv2

rcb4 = Rcb4BaseLib() #rcb4 をインスタンス(定義)
rcb4.open('COM3',1250000,1.3) #(portName,bundrate,timeout(s))

def random_action():
    return np.random.choice([0, 6])

def get_action(next_state, episode):
    epsilon = 0.5 * (1 / (episode + 1)) #徐々に最適行動のみをとる  $\epsilon$ -greedy 法
    if epsilon <= np.random.uniform(0, 1):
        a = np.where(q_table[next_state]==q_table[next_state].max())[0]
        next_action = np.random.choice(a)
        #print("USE EPSILON!!")
    else:
        next_action = random_action()
    return next_action, epsilon

def step(state, action, away, fall, y, z, fd, blue, yellow):
    reward = 0
    fr = -50 #転倒した時に与える報酬。fall reward の略
    adjust = 30 #報酬調整用
    ho = z/adjust

    if fd > fall and yellow >10:
        if blue == 1:
            state = 4
            reward = fr
        else:
            state = 3
            reward = fr

    elif z <= 70:
        state = 5
        reward = 5/ho

    elif state==0:
        if y > away:
            state = 1
            reward = -ho
        elif y < -away:
            state = 2
```

```

        reward = -ho
    else:
        state = 0
        reward = -ho/3

    elif state==1:
        if y > away:
            state = 1
            reward= -ho
        else:
            state = 0
            reward = ho/3

    elif state==2:
        if y < -away:
            state = 2
            reward= -ho
        else:
            state = 0
            reward = ho/3

    elif state==3:
        if y > away:
            state = 1
            reward = -fr/2
        elif y < -away:
            state = 2
            reward = -fr/2
        else:
            state = 0
            reward = -fr/2

    elif state==4:
        if y > away:
            state = 1
            reward = -fr/2
        elif y < -away:
            state = 2
            reward = -fr/2
        else:
            state = 0
            reward = -fr/2

    else:
        if y > away:
            state = 1
            reward = -ho*5
        elif y < -away:
            state = 2
            reward = -ho*5
        else:
            state = 0
            reward = -ho*5
    return state, reward

def updata_Qtable(q_table, state, action, reward, next_state):
    gamma = 0.9
    alpha = 0.5
    next_maxQ = max(q_table[next_state])
    q_table[state, action] = (1 - alpha) * q_table[state, action] + \
        alpha * (reward + gamma * next_maxQ)

```

```

    return q_table

max_number_of_steps = 10 #1試行のstep数
count = open('count.txt', "r") #エピソードの回数をテキストから引っ張ってくる
st = count.read()
count.close()
co = int(st) #エピソード数が文字列なので int 型にする

if co == 1: #最初は q_table を設定
    q_table = np.zeros((6, 7)) #(state, action)
else: #1回目以降は Qvalue.txt を使用
    q_table = np.loadtxt('Qvalue.txt')

print(q_table) #q_table 確認用
if co % 10 == 1:
    np.savetxt('Q テーブルの記録_' + st + '.txt', q_table, fmt = '%.2f')

state = 0
episode_reward = 0
away = 20 #y の値がどれだけズレてもよいかの値
fall = 330 #転倒したかどうかの判断のための値

y, z, fd, blue, yellow = color2.dimension(fall) #座標獲得
if fd > fall and yellow > 10:
    if blue == 1:
        state = 4
    else:
        state = 3
elif z <= 70:
    state = 5
elif y > away:
    state = 1
elif y < -away:
    state = 2
else:
    state = 0

for t in range(max_number_of_steps): #1試行のループ
    action, epsilon = get_action(state, co-1) #アクションの決定
    rcb4.motionPlay(action + 1) #決定したアクションに対応した番号のモーションの再生
    y, z, fd, blue, yellow = color2.dimension(fall)
    next_state, reward = step(state, action, away, fall, y, z, fd, blue, yellow)
    print('s:%d_a:%d_r:%.3f_n_s:%d' % (state, action, reward, next_state))
    episode_reward += reward #報酬を追加
    q_table = update_Qtable(q_table, state, action, reward, next_state) #Q テーブルの更新
    state = next_state
    time.sleep(2)

rcb4.motionPlay(20) #姿勢を元に戻す
print('episode:%d_total_reward:%.4f_epsilon:%.4f' % (co, episode_reward, epsilon))

file = open('count.txt', 'w') #エピソード数を管理している txt ファイルを開く
co += 1
file.write(str(co)) #エピソード数を1つ増やす
print('エピソード数を更新しました。')

file.close()
print('%d 回目のロボットとゴールの最終的な距離は %.2f です。' % (co-1, z))
cv2.destroyAllWindows()
np.savetxt('Qvalue.txt', q_table, fmt = '%.4f')

```

```

import cv2
import numpy as np
import math

def dimension():
    def find_rect_of_target_color(image):
        hsv = cv2.cvtColor(image, cv2.COLOR_BGR2HSV_FULL)
        h = hsv[:, :, 0]
        s = hsv[:, :, 1]
        mask = np.zeros(h.shape, dtype=np.uint8)
        mask[((h < 10) | (h > 240)) & (s > 128)] = 255
        contours, _ = cv2.findContours(mask, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE
        )
        rects = []
        for contour in contours:
            approx = cv2.convexHull(contour)
            rect = cv2.boundingRect(approx)
            rects.append(np.array(rect))
        return rects

    def find_rect_of_target_color2(image):
        hsv = cv2.cvtColor(image, cv2.COLOR_BGR2HSV_FULL)
        h = hsv[:, :, 0]
        s = hsv[:, :, 1]
        mask = np.zeros(h.shape, dtype=np.uint8)
        mask[((h>31)&(h <70)) & (s > 128)] = 255
        contours, _ = cv2.findContours(mask, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE
        )
        rects = []
        for contour in contours:
            approx = cv2.convexHull(contour)
            rect = cv2.boundingRect(approx)
            rects.append(np.array(rect))
        return rects

    def find_rect_of_target_color3(image):
        hsv = cv2.cvtColor(image, cv2.COLOR_BGR2HSV_FULL)
        h = hsv[:, :, 0]
        s = hsv[:, :, 1]
        mask = np.zeros(h.shape, dtype=np.uint8)
        mask[((h>130)&(h <190)) & (s > 128)] = 255
        contours, _ = cv2.findContours(mask, cv2.RETR_TREE, cv2.
            CHAIN_APPROX_SIMPLE)
        rects3 = []
        for contour in contours:
            approx = cv2.convexHull(contour)
            rect3 = cv2.boundingRect(approx)
            rects3.append(np.array(rect3))
        return rects3

    capture = cv2.VideoCapture(1)
    capture2 = cv2.VideoCapture(0)
    _, frame = capture.read()
    _, frame2 = capture2.read()
    rects = find_rect_of_target_color(frame)
    rects2 = find_rect_of_target_color2(frame)
    rects3 = find_rect_of_target_color(frame2)
    rects4 = find_rect_of_target_color2(frame2)
    rects_blue = find_rect_of_target_color3(frame2)
    rect=[0,0,0,0]

```

Challenge to Complex Intelligent Robot

```
rect2=[0,0,0,0]
rect3=[0,0,0,0]
rect4=[0,0,0,0]
rect_blue=[0,0,0,0]

#横カメラ
if len(rects) > 0:
    rect = max(rects, key=(lambda x: x[2] * x[3]))
    cv2.rectangle(frame, tuple(rect[0:2]), tuple(rect[0:2] + rect[2:4]), (0, 0, 255), thickness=2)
    print('\033[31m{}\033[0m'.format(rect))

if len(rects2) > 0:
    rect2 = max(rects2, key=(lambda x: x[2] * x[3]))
    cv2.rectangle(frame, tuple(rect2[0:2]), tuple(rect2[0:2] + rect2[2:4]), (0, 255,255),
    thickness=2)
    print('\033[32m{}\033[0m'.format(rect2))

x=rect[0]-rect2[0]
y=rect[1]-rect2[1]
z=math.sqrt(x*x+y*y)

text=str(round(z))

font = cv2.FONT_HERSHEY_PLAIN
cv2.putText(frame,text,(rect[0],rect[1]-10),font, 2, (255, 255, 255), 2, cv2.LINE_AA)
cv2.line(frame,(rect[0],rect[1]),(rect2[0],rect2[1]),255,2)
cv2.line(frame,(0,400),(1000,400),(0,0,0),2)

cv2.imshow('Horizontal', frame)

#縦カメラ
if len(rects3) > 0:
    rect3 = max(rects3, key=(lambda x: x[2] * x[3]))
    cv2.rectangle(frame2, tuple(rect3[0:2]), tuple(rect3[0:2] + rect3[2:4]), (0, 0, 255), thickness
    =2)
    print('\033[31m{}\033[0m'.format(rect3))

if len(rects4) > 0:
    rect4 = max(rects4, key=(lambda x: x[2] * x[3]))
    cv2.rectangle(frame2, tuple(rect4[0:2]), tuple(rect4[0:2] + rect4[2:4]), (0, 255,255),
    thickness=2)
    print('\033[32m{}\033[0m'.format(rect4))

x2=rect3[0]+rect3[2]//2-(rect4[0]+rect4[2]//2)
y2=rect3[1]+rect3[3]//2-(rect4[1]+rect4[3]//2)
z2=math.sqrt(x2*x2+y2*y2)
fd=rect2[1]
text=str(round(z2))
blue = 0

font = cv2.FONT_HERSHEY_PLAIN
cv2.putText(frame2,text,(rect3[0],rect3[1]-10),font, 2, (255, 255, 255), 2, cv2.LINE_AA)
cv2.line(frame2,(rect3[0],rect3[1]),(rect4[0],rect4[1]),255,2)

if rect2[1] >300:
    if len(rects_blue) > 0:
        rect_blue = max(rects_blue, key=(lambda x: x[2] * x[3]))
        cv2.rectangle(frame2, tuple(rect_blue[0:2]), tuple(rect_blue[0:2] + rect_blue[2:4]), (0,
        255,0), thickness=2)
        print('\033[34m{}\033[0m'.format(rect_blue))
        if rect_blue[2] >6:
```

Challenge to Complex Intelligent Robot

```
        print('blue!')
        blue =1
    print('fall_down')

cv2.imshow('Vertical', frame2)

return y2, z2, fd, blue
```

参考文献

- [1] 北山直洋・北山洋幸. Java で始める OpenCV3 プログラミング. 株式会社カットシステム, 2016.
- [2] 露木誠・小田切篤. 15 時間でわかる Python 集中講座. 株式会社技術評論社, 2016.
- [3] 牧野浩二・西崎博光. Python による深層強化学習入門 Chainer と OpenAI Gym ではじめる強化学習. オーム社, 2018.