

A I チャレンジ研究会(第 54 回)

Proceedings of the 54th Meeting of Special Interest Group on AI Challenges

CONTENTS

- RoboCup SSL Humanoid のためのポイントクラウドサーバの提案 1
平井 雅之, 升谷 保博
大阪電気通信大学
- 順シミュレーションにより生成された予測局面を用いたサッカーの試合状況評価 9
鈴木 雄大, 中島 智晴
大阪府立大学大学院
- RoboCup サッカーにおけるヘテロマッチングが戦術に与える影響 18
福島 卓弥[†], 山口 将貴[‡], 中島 智晴[†], 秋山 英久[#]
[†]大阪府立大学大学院, [‡]大阪府立大学, [#]福岡大学
- アーム型ロボットにおける可動範囲に関する一考察 24
加藤 大登, 鈴木 勇貴, 植村 渉
龍谷大学
- RoboCup Logistics League におけるフィールド内の障害物検知に関する一考察 28
植村 渉
龍谷大学

日時:2019年8月17日

場所: シティホールプラザ「アオーレ長岡」アリーナ 3F 会議室 B

Meeting room B in Nagaoka City Hall Plaza Aô-re Nagaoka, August 17th, 2019

一般社団法人 人工知能学会

Japanese Society for Artificial Intelligence

RoboCup SSL Humanoidのための ポイントクラウドサーバの提案

A Proposal of Point Cloud Server for the RoboCup SSL Humanoid

平井 雅之^{1*} 升谷 保博¹
HIRAI Masayuki¹ MASUTANI Yasuhiro¹

¹ 大阪電気通信大学
¹ Osaka Electro-Communication University

Abstract: 本稿では RoboCup SSL Humanoid の次世代の競技形態のための外部視覚共有システムとして「ポイントクラウドサーバ」を提案する。外部視覚としてフィールドを囲む複数の深度カメラを用いる。深度カメラに1台ごとにサーバプログラムを設け深度カメラから得られた深度画像を3次元の色付き点群に変換し、フィールド座標系に座標変換する。そこからフィルタリング処理を行い不要な点を除去し、さらに、データを圧縮してマルチキャストのUDP通信でネットワークに送出する。このデータを各チームのプログラムが受け取り、そこから先は独自に処理を行う。深度カメラとして Kinect v2 を使って提案方法を実装し検証を行った。

1 はじめに

RoboCup サッカーの小型ロボットリーグ (Small Size Robot League, 以下 SSL)[1] とそのサブリーグである SSL Humanoid[2] のシステムでは、他のリーグと異なり、ロボット自身にはカメラを搭載する必要はなく、外部カメラを用いてフィールド上のロボットやボールの位置を認識しロボットを制御することができる。

現状では、図1のようにフィールド上空のカメラでロボットの頭頂部に取り付けたマーカのパターンを撮影し、2次元的な画像処理を経て、その番号や座標、方向を算出している。しかし、この方式には多くの欠点がある。ロボットの頭頂部のマーカ板は、ロボットを動作させる際に邪魔になったり、ロボットが転倒してマーカが見えなくなると認識されなくなったりする。また、得られるのはロボットの座標と方向のみなので、ロボットの3次元的な状態を知ることはできない。これは、多様なポーズが可能でヒト型ロボットにとって大きな問題である。そこで、SSL Humanoid の提案当時からこの競技の最終段階の外部視覚システムの条件として図2のように3次元・マーカなしを想定している。このようなシステムであれば、複数の外部カメラで撮影したカメラに基づいて、複数のロボットの3次元的な形状を取得し、それに基づきロボットを制御することができ、外部視覚の特徴を活かした新たなアイデアの創出が期待できる。

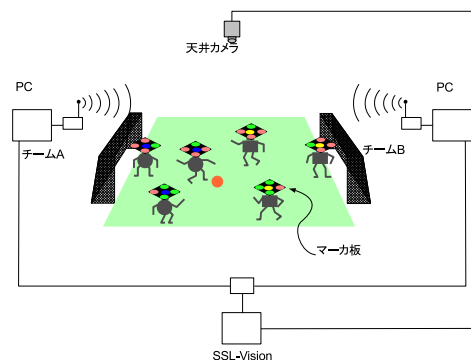


図1: 現在の SSL Humanoid システム (2次元・マーカあり)

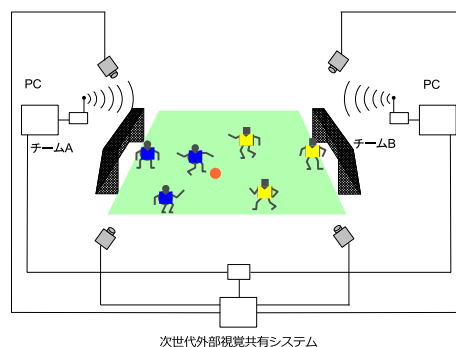


図2: 次世代の SSL Humanoid システム (3次元・マーカなし)

*連絡先: 大阪電気通信大学 総合情報学部 情報学科
〒575-0063 大阪府四條畷市清滝 1130-70
E-mail: odens.oecu@gmail.com

これを受けて、木村らはフィールド上の3次元空間のボクセルデータを共有することを提案し、それを具体化する「ボクセルサーバ」を開発した [3]。このサーバは、8台のカメラを用い、3025[mm]×4050[mm]×500[mm]の領域を10[mm]立方のボクセルで表現し、それをネットワークを介してクライアントへ配信している。また、クライアント側では、ボクセルデータからロボットのポーズ推定を行い、その結果からロボットの行動決定ができることを示している [4]。さらに、植田らはこのボクセルサーバに色の情報を付け加えたカラーボクセルサーバを提案した [5]。これにより、ロボットとボールが接触すると1つのオブジェクトとして認識する問題が解決された。しかし、これらの方式では、十分な3次元情報を得ようとすると少なくとも8台のカメラを必要としており、設置やキャリブレーションに時間と手間がかかるという問題点がある。

そこで、RGBカメラではなく、最近手軽に使えるようになった深度カメラを利用して次世代 SSL Humanoid のための外部視覚システムを実現する「ポイントクラウドサーバ」を提案する。以下の節では、その考え方や実装について述べる。次に、深度カメラとして Kinect v2 を4台用いて、SSL Humanoid のフィールド (4.5[m] × 3[m]) に置かれたオブジェクトを撮影し、実装したシステムの性能を評価し、実際の競技に使えるかどうかを検証する。

2 ポイントクラウドサーバの提案

複数の深度カメラから得られた情報をネットワークを介して参加チームが共有することについて考察する。まず決めなければいけないので、どこまでを共有してサーバ側で処理し、何をクライアント (チーム側) へ提供するかということである。これについて、以下のような三つの選択肢を考える。

1. 深度画像をそのまま提供。
2. 一般的な前処理を行ってから提供
3. 3次元オブジェクトの認識を行ってから提供。

1の場合、サーバ側の処理は非常に簡潔であるが、クライアント側の負担が重く、データの送信量が大きくなり処理速度が下がってしまう。2の場合、データの送信量のある程度削減でき、処理速度も確保しやすいが、カメラのキャリブレーションなどの課題がある。3の場合、一番データの送信量が少ないが、サーバ側の処理が多く、クライアント側での工夫を行う余地が少ない。そこで、本稿では、2を選択し、深度画像から得られた点群データを提供する「ポイントクラウドサーバ」を提案する。

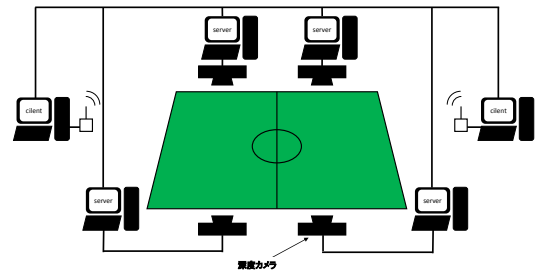


図 3: ポイントクラウドサーバの概念図

ポイントクラウドサーバの概念図を図 3 に示す。複数の深度カメラでフィールドを囲い撮影する。深度カメラ 1 台につきサーバプログラム 1 個を割り付ける。共有するデータはマルチキャストの UDP 通信でネットワークに送出する。

サーバプログラムでは深度画像を 3 次元の色付き点群データに変換し、共通の座標系に変換する。次に、フィールド外や床といった不要な点を除去と点のデータ量の圧縮を行い送信量を削減する。複数のサーバのデータを区別するために、それぞれに別々のマルチキャストアドレスとポート番号を割り当てる。

現在の共有視覚システムである SSL-Vision では、60fps 程度の頻度で処理をしている。一方、一般的な深度カメラでは、30fps 程度の頻度でデータの取得ができるので、提案のシステムでもそれと同じ処理頻度に行えるのが理想的である。しかし、車輪型ロボットに比べてヒト型ロボットは動作が遅いため、10fps 程度の処理頻度でも実用できるのではないかと考えている。

厳密な計測のためには、センサ間の同期ができた方が望ましいが、機材的な制約でそれは難しいかもしれない。そこで、競技としては非同期のデータに基づくことを前提としても良いのではないかと考える。ただし、サーバ間での時刻は同期しておき、点群データにタイムスタンプを付与する。

3 ポイントクラウドサーバの実装

2 節で提案したポイントクラウドサーバを検証するために、実装したシステムについて述べる。

3.1 機材・開発環境

実装した PC はサーバ側は Windows10、クライアント側は Windows8.1 である。開発環境には Visual Stu-

dio 2015, 使用言語は C++ である。

深度カメラは Kinect v2 を用いた。深度画像の解像度は 512×424 画素である。また, Kinect の画角は水平方向に 70 [deg], 垂直方向に 60 [deg], 計測範囲は 0.5 [m] から 8 [m] までである。

本研究では, 点群を表示させ操作するため, およびキャリブレーションに使用した点群内での平面検出や不要な点群のフィルタリングに, Point Cloud Library を使用している。実装に利用したバージョンは, 1.8.1 である。

3.2 サーバの実装

まず, Kinect の深度画像は KinectGrabber[6] を使用して色付き点群に変換する。

Kinect の深度画像から得られた点群の座標値は, カメラ座標系で表現されている。クライアント側での点群のマージが容易になるように, サーバ側で点群を共通のフィールド座標系へ変換する。図 4 に示すように, フィールド座標系は, フィールドの平面上の中心を原点とし, フィールド平面の法線方向を z 軸, ゴールとゴールを結ぶ方向を x 軸とする。

ある点 p を, カメラ座標系で表現したものを ${}^c p$, フィールド座標系で表したものを ${}^f p$ とすると, 両者の関係は, 式 (1) のように表現できる。

$${}^f p = {}^f R_c {}^c p - {}^f p_c \quad (1)$$

ここで, ${}^f R_c$ はフィールド座標系に対するカメラ座標系の回転を表す行列, ${}^f p_c$ はフィールド座標系に対するカメラ座標系の並進を表すベクトルである。本稿の実装では, 別のキャリブレーションプログラムで, この回転行列と並進ベクトルを推定し, その結果をファイルを介してサーバプログラムに受け取って処理に利用する。

Kinect v2 の深度画像から得られる点群の点の数は最大で $512 \times 424 = 217088$ 個である。このデータをそのまま送るとデータの送信量が多く処理速度の低下に繋がるためデータの送信量を削減する必要がある。

点群の中には不要な点が多く存在するためフィルタリングを行う。まずフィールドから大きく外れた箇所の点群を除去する。SSL Humanoid のフィールドは 3025×4050 [mm] であるので, フィールドの原点を中心とし, 3600×5000 [mm] に含まれない点を除去する。次に床面は不要なので z 軸方向に 5 [mm] 以下の点群を除去する。これにより, フィールド上のオブジェクトの点群だけが残る。

さらに, 点群の点 1 個当たりのデータ量も減らす。PCL における色付き点群の 1 点のデータは, 座標値 (X,Y,Z) がそれぞれ [m] 単位の浮動小数点数で 4byte,

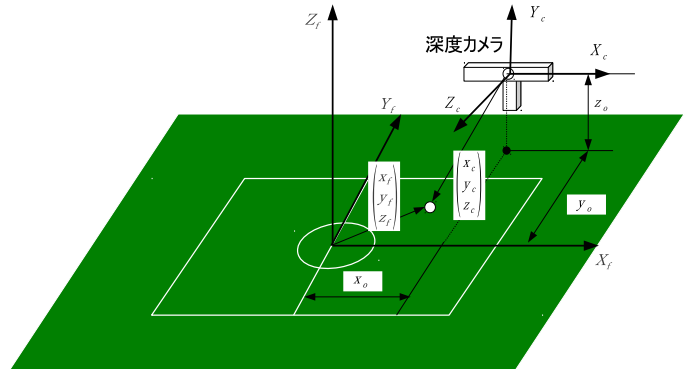


図 4: フィールド座標系とカメラ座標系

色情報 (R,G,B, α) がそれぞれ 1 byte, 合計で 16byte である。これを, 座標値はそれぞれ [mm] 単位の整数 2byte に, 色情報はそれぞれ 5bit に変換し合計で 8byte に圧縮する。

3.3 テストクライアント

テスト用のクライアントプログラムを作成する。全てのサーバから新しいデータが届いた場合, 点群をマージし表示する機能のみが実装されている。今後は, マージした点群に対して様々な処理を行う処理の実装が必要である。

3.4 キャリブレーション

キャリブレーションによって求めるべきものはカメラ座標系からフィールド座標系に変換するための回転行列と並進ベクトルである。この手法はフィールド座標系の xy 面と xz 面と yz 面に一致する平板を設置し, それを撮影した点群データから, 回転行列と並進ベクトルを求める。

xy 面は床を使用し, xz 面と yz 面に相当する平板として衝立などで壁を作る。その 2 つの壁と床を Kinect で撮影し, その点群を使い PCL で平面検出を行う。その際の深度カメラと平面と座標系のおおよその位置関係を図 5 に示す。この例ではフィールドの第 4 象限に Kinect を設置している。

3.4.1 キャリブレーションの実装

点群から平面を取得する方法は PCL のモデルタイプは SACMODEL_PLANE で, 検出方法は SAC_RANSAC

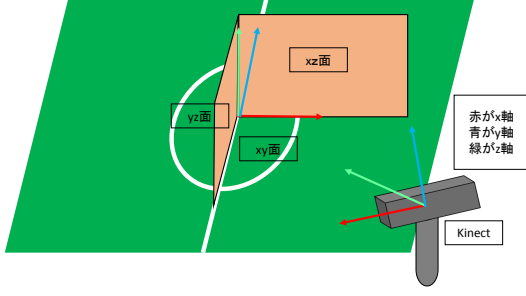


図 5: 深度カメラ, 平面, 座標系の位置関係

を用いて検出する. 閾値は 0.05[m] で試行回数は 50 回である. これにより点群内の平面の式の係数が得られる.

この方法で平面を検出すると点群内で一番大きな平面が求められる. その後元の点群から求めた平面に含まれる点群を取り除き再度平面検出を行うことで複数の平面を検出する. これを 3 つの平面が検出されるまで繰り返す.

検出された 3 平面のパラメータから, 座標変換に必要な回転行列と並進ベクトルを推定し結果をファイルに保存する.

3.4.2 キャリブレーションの原理

xy 平面, xz 平面, yz 平面の係数をそれぞれ $a_z, b_z, c_z, d_z, a_y, b_y, c_y, d_y, a_x, b_x, c_x, d_x$ とする. ただし以下の様な関係を満足しているとする.

$$\sqrt{a_i^2 + b_i^2 + c_i^2} = 1 (i = x, y, z) \quad (2)$$

xy 平面の式を式 (4), xz 平面の式を式 (5), yz 平面の式を式 (6) と表す.

$$a_z x + b_z y + c_z z + d_z = 0 \quad (3)$$

$$a_y x + b_y y + c_y z + d_y = 0 \quad (4)$$

$$a_x x + b_x y + c_x z + d_x = 0 \quad (5)$$

3 つの平面の式を x, y, z に対する連立方程式とする. これを解くと, その解の x, y, z はカメラ座標系におけるフィールド座標系の原点である. この点を以下の様に表す.

$${}^c \mathbf{p}_f = \begin{pmatrix} x_o \\ y_o \\ z_o \end{pmatrix} \quad (6)$$

平面の式の a, b, c はそのまま平面の法線ベクトルを表しており, フィールド座標系の座標軸と平行である. これを以下の様を書くことにする.

$$\mathbf{n}_i = \begin{pmatrix} a_i \\ b_i \\ c_i \end{pmatrix} (i = x, y, z) \quad (7)$$

ただし, \mathbf{n}_i の向きに任意性があるが, Kinect がフィールド座標系の原点の方向を向いているという前提で, Kinect がフィールド座標系のどの象限にあるかによって適切に設定する. 推定した平面のパラメータは誤差を含んでいるため, 各法線ベクトルは直交しているとは限らない. そこで, 床面の法線を基準に補正する.

$$\mathbf{n}'_x = \mathbf{n}_x - (\mathbf{n}_x \cdot \mathbf{n}_z) \mathbf{n}_z \quad (8)$$

$$\mathbf{n}''_x = \frac{\mathbf{n}'_x}{|\mathbf{n}'_x|} \quad (9)$$

$$\mathbf{n}'_y = \mathbf{n}_z \times \mathbf{n}''_x \quad (10)$$

ここで $\mathbf{x} \cdot \mathbf{y}$ はベクトル \mathbf{x} と \mathbf{y} の内積, $\mathbf{x} \times \mathbf{y}$ はベクトル \mathbf{x} と \mathbf{y} の外積を表す. カメラ座標系におけるフィールド座標系の姿勢は回転行列を使って式 (11) のように表される.

$${}^c R_f = (\mathbf{n}''_x \mathbf{n}'_y \mathbf{n}_z) \quad (11)$$

これらを使うと, フィールド座標系からカメラ座標系の変換は, 式 (12) のように表される.

$${}^c \mathbf{p} = {}^c R_f {}^f \mathbf{p} + {}^c \mathbf{p}_f \quad (12)$$

この式を ${}^f \mathbf{p}$ について解くと式 (13) になる.

$${}^f \mathbf{p} = ({}^c R_f)^T ({}^c \mathbf{p} - {}^c \mathbf{p}_f) = {}^f R_c {}^c \mathbf{p} - {}^f \mathbf{p}_c \quad (13)$$

ここで,

$${}^f R_c = ({}^c R_f)^T \quad (14)$$

$${}^f \mathbf{p}_c = {}^f R_c {}^c \mathbf{p}_f \quad (15)$$

であり, これらがキャリブレーション結果そのものである.

4 評価実験

この節では, 本稿で提案したシステムや手法の評価実験について述べる. SSL Humanoid のルールで定められた 3025×4050 [mm] のフィールドに対して 4 台の Kinect を設置した. Kinect を設置したフィールド座標系での位置を表 1 に示す. 単位は [mm] である. 取得した点群データはテストクライアントで表示し, 評価に使用する. 実験に使用したサーバ PC とクライアント PC の仕様を表 2 に示す.

表 1: Kinect の配置座標 [mm]

	x[mm]	y[mm]	z[mm]
第 1 象限	2800	2000	1000
第 2 象限	-2900	1700	1100
第 3 象限	-2800	-2100	1000
第 4 象限	2800	-2000	1100

表 2: 使用した PC の仕様

	CPU	RAM
server	Intel Celeron@1.70GHz	4GB
cilent	Intel Core i7@2.30GHz	8GB



図 6: キャリブレーションの様子

4.1 複数カメラの統合の確認

本稿で提案したキャリブレーションを用いて Kinect の位置姿勢を推定し、その結果を用いて各 Kinect から取得した点群をマージする。3つの平面を用意するために使用した衝立は 1000×600 [mm] と 1100×600 [mm] の 2枚のプラスチックダンボールを固定したものである。これをコートに印に合わせて立てることで床を合わせ、互いに直交する三つの平面を用意する。キャリブレーションを行う際の様子を図 6 に示す。統合確認実験の際のフィールド上のオブジェクトの配置を図 7 に示す。フィールドには大きさ約 $260 \times 140 \times 170$ [mm] のロボットが 3台と大きさ約 $350 \times 160 \times 190$ [mm] のロボットが 3台と直径 60 [mm] のボールを 1つ設置してある。マージ後の点群を図 8 に示す。図 8 より、キャリブレーションにより 4つの Kinect から送られてくる点群データがマージされ、SSL Humanoid のフィールド全域の点群を取得できていることがわかる。しかし Kinect 間の座標に少なからずずれが見られ、キャリブレーションの際の平面の設置方法の改善や基となる原理の見直しが必要であることがわかる。

フィールドの平面とフィールド上のオブジェクトでキャリブレーションやマージの処理の良し悪しを判断するために、フィールド外の点を削除した結果を図 9 から図 11 に示す。図 9 よりフィールド内の白線にずれがあることがわかり、図 10 と図 11 より、床の点群が一致していることがわかる。これらの結果からキャリブレーションの際に使用した平面として、床はかなりの精度で平面が検出できている。これにより、平面を大きくするなどの設置方法を改善すると他の平面での精度も向上すると考えられる。



図 7: 点群取得時のフィールド



図 8: マージされた点群を全て表示したもの

4.2 フィルタリングの効果の確認

本稿で提案したフィルタリングを行い、実際のデータの削減量、処理速度の変化を確認する。床面のフィルタリングを行った点群を図 12 に示す。フィルタリン

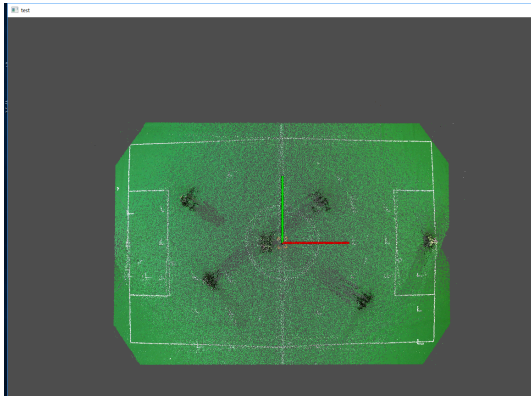


図 9: フィールド外の点を削除した点群 (XY 面)



図 10: フィールド外の点を削除した点群 (XZ 面)



図 11: フィールド外の点を削除した点群 (YZ 面)

表 3: フィルタリング前後のデータ

	フィルタリング前	フィルタリング後
点数	217088 × 4	約 5500
データ量	1736704[byte] × 4	約 43200[byte]
処理速度	1~3[fps]	6~9[fps]

グを行う前後での点数, データの大きさ, 処理速度を表 3 に示す.

図 12 より, フィルタリングにより床面がほとんど除去されていることがわかる. 表 3 より, フィルタリン



図 12: フィルタリングで床面を削除した後の点群

表 4: 使用したロボットと点数

使用ロボット	RIC-30
大きさ	260 × 140 × 170[mm]
中心での点群	約 800
四隅での点群	約 2200
ボールの大きさ	直径 60mm
中心での点群	約 100
四隅での点群	約 300

グを行う前と後で点数, データの大きさが大きく削減され, 処理速度も向上していることがわかる. 目標としていた 10fps には届いていないため, さらに改善が必要であるがこのフィルタリングが十分に有効であることがわかる.

4.3 得られたオブジェクトの点群の確認

本稿で提案したキャリブレーションとフィルタリングを行い得られた点群データはオブジェクトを認識するのに使用できるかを確認し, ロボットに複数の姿勢を取らせ点群での変化を確認する. 検証に使用したロボットとボールの写真を図 13 に示す. オブジェクトをフィールド座標系の (0,0) 付近に設置し, 写真とほぼ同じ角度から見た点群を図 14 に示す. 真上から見た点群を図 15 に示す. オブジェクトをフィールド座標系の (1800,1300) 付近に設置し, 写真とほぼ同じ角度から見た点群を図 16 に示す. 使用したロボットとボールのそれぞれについての大きさと得られた点の数を表 4 に示す. なお, 点数は Kinect4 台分の合計である.

図 14 と図 15 より, 立っているロボットと倒れているロボットに明らかな違いがあることがわかる. 重心

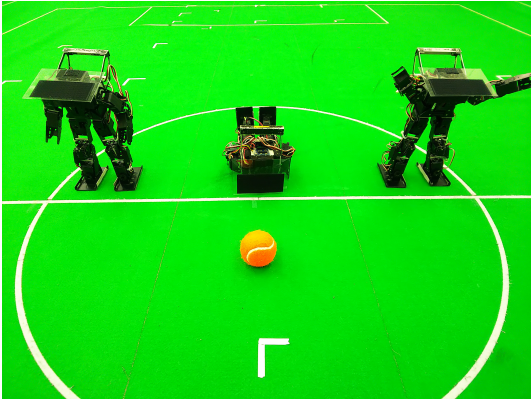


図 13: オブジェクトの点群の確認実験の対象

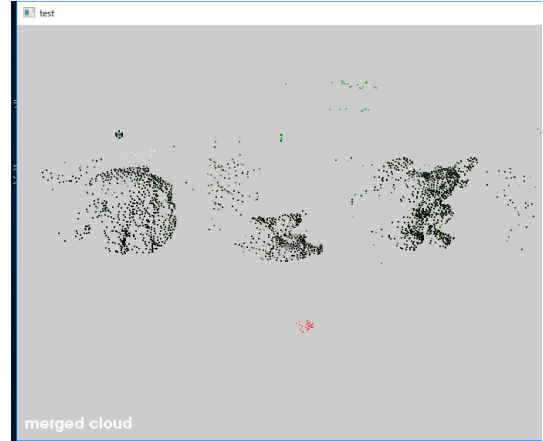


図 16: オブジェクトの点群（フィールドの隅，斜め上から）



図 14: オブジェクトの点群（フィールドの中心，斜め上から）



図 15: オブジェクトの点群（フィールドの中心，真上から）

の位置が低くなっており，点群が前後に広がっているのがわかる．これを利用すればロボットが立っているのか倒れているかの判断が可能である．しかし，点数が少ないため，左のロボットと右のロボットに大きな違いは見られない．大きな変化は現状でも判断が可能であるが細かな姿勢の違いを判断するにはカメラの解像度を上げるなどを行い点数を増やす必要がある．

図 16 では図 14 と違い，左のロボットと右のロボットの姿勢の違いが見て取れる．フィールドの隅だと深度カメラから近く，オブジェクトの点数が増えるため，ロボットの状態が詳細にわかる．

図 14 と図 15 より，ロボットの点群を取得した際は問題にはならなかったがボールの点群を取得すると座標変換の結果の不一致がはっきりと確認できる．ボールはロボットに比べ小さなオブジェクトであるためずれが大きくなり現れてしまう，これではボールの認識に影響があるため精度の改善が必要である．

表 4 より，オブジェクトの点数がフィールドの位置によって約 3 倍程違うことがわかる．フィールドの中心はどの Kinect からとも遠いからである．

5 クライアント側の処理の検討

この節では評価実験の結果を経て，クライアント側の処理として求められるものを検討する．

5.1 クラスタリング

まずロボットやボールを認識するにはクラスタリングが必要である．個々のオブジェクトが孤立しているならクラスタリングするのは難しくないがオブジェクト同士が接触した際のクラスタリングが課題である．ロ

ロボットとボールの場合は色などで判別できるがロボット同士の接触は判別がつきにくいと考えられる

5.2 ロボットの位置と向き

ロボットの認識ができたと仮定すると次の問題はロボットの位置と向きの認識である。クラスタリングした点群データの重心を求めれば位置はわかる。クラスタリングした点群データから主軸方向を認識すれば縦の向きがわかる。多くのロボットは上から見ると横に長いことが殆どであるのでそれも利用できるが、それだけでは前後の判断が難しい。ロボットの胴体の前面にのみ色の違うものを張るなど何かしらの対処が必要である。

5.3 ロボットのチームと号機の区別

各ロボットに指令を送る際にロボットの識別は必須である。敵味方の区別も必要である。マーカ板を撤廃したがチームの認識のため、色のついたラベルなどを張ることが必要だろう。

6 おわりに

本稿では、RoboCup SSL Humanoid の共有視覚システムとして利用することを想定した「ポイントクラウドサーバ」を提案した。

4台の Kinect でフィールドを全域を撮影することができた。フィルタリングなどの処理でデータを削減すると処理頻度が向上した。もう少し処理速度の向上ができればヒト型ロボット用としては十分実用が可能である。点群のデータはオブジェクトを認識するのに十分利用できる。

今後の課題は、キャリブレーションの精度の向上、ポイントクラウドサーバシステムの処理周期の向上、クライアント側のプログラム作成である。

参考文献

- [1] 長坂保典: ロボカップの道しるべ: 第3回 小型ロボットリーグ, 情報処理, vol.52 No.1, pp.95-110, 2011.
- [2] 升谷保博, 成瀬正: 外部カメラを用いたヒト型ロボットによるサッカー競技 RoboCup Soccer SSL Humanoid, 人工知能学会誌, Vol.25, No.2, pp.213-219, 2010.
- [3] 木村堯海, 升谷保博: 複数の外部カメラ画像に基づくヒト型ロボットの3次元形状の実時間取得 RoboCup Soccer SSL Humanoid の3次元情報サーバをを目指して, 第31回人工知能学会 AI チャレンジ研究会, pp.33-38, 2010.
- [4] 植田康生, 木村堯海, 勢川友樹, 升谷保博: 外部カメラ画像から得られた複数のヒト型ロボットのボクセルデータに基づく行動決定, 第33回人工知能学会 AI チャレンジ研究会, pp.7-12, 2011.
- [5] 植田康生, 升谷保博: RoboCup SSL Humanoid のためのカラーボクセルサーバの提案と応用, 第37回人工知能学会 AI チャレンジ研究会, pp.9-14, 2013
- [6] 杉浦司: GitHub UnaNancyOwen/KinectGrabber <https://github.com/UnaNancyOwen/KinectGrabber/tree/Kinect2Grabber>

順シミュレーションにより生成された予測局面を用いた サッカーの試合状況評価

Situation Evaluation for a Soccer Game Using Predicted Situations Generated by Forward Simulation

鈴木 雄大 中島 智晴*
Yudai Suzuki Tomoharu Nakashima

大阪府立大学大学院 人間社会システム科学研究科
Graduate School of Humanities and Sustainable System Sciences
Osaka Prefecture University

Abstract: In this paper, we propose an evaluating method for the RoboCup soccer game situation. The idea is to evaluate a game situation with multiple situations by using a machine learning method. Through preliminary computational experiments, we found that multiple future situations are necessary for evaluating a current game situation. However, as a matter of course, future information is not available during an ongoing game. For this problem, we propose a FOrward Simulation for Situation Evaluation (FOSSE) approach. This approach consists of two parts. The first part is Forward Simulation part that predicts the future situations by using past and the current situations. The other part is Situation Evaluation part that evaluates the current situation by using the current and predicted future situations. We employ deep learning models such as an LSTM in both parts of the FOSSE approach. First, we show that the evaluation performance can be increased by using successive multiple situations in time. Especially, the effectiveness of using future information rather than past information is shown. Then, we present the FOSSE approach where both current and predicted future information of game situations are used to evaluate current game situation. Computational experiments are conducted to investigate the effectiveness of the proposed method.

1 はじめに

スポーツにおいて、試合中に優勢や劣勢を把握することは重要である。試合状況を定量的に評価することができる。これにより、チームの優勢度合を正確に把握することができる。これにより、チームの戦略切替の指針や、スポーツの自動実況分野への応用が期待できる。しかし、動的に状況が変化する試合を定量的に評価するのは困難である。この問題に対して、機械学習手法を用いることで、定量的な試合状況の評価を実現する。

本論文における実験環境として、RoboCup サッカーシミュレーション 2D リーグ [1] を採用する。Nakashima and Pomas [2] は、*SituationScore* と呼ばれるサッカーの試合状況における優勢度合の指標を定義した。本論文においても、*SituationScore* をいくつか変更を加え

た上で試合状況の評価指標として用いる。

試合状況を評価するもっとも単純な方法は、現状の試合状況のみの単一局面を使用することである。しかしながら、特にサッカーでは試合が動的に展開し、試合状況は刻々と変化する。そのため、選手やボールの動きを捉えることができない単一局面を用いた状況評価は難しいと言える。そこで、本論文では、複数局面を用いた試合状況評価の有用性を調査する。

未来の試合状況を用いることができれば、現在までの情報を用いるよりも高い精度の試合状況評価ができると考えられる。しかし、当然のことではあるが、試合中に未来の情報は取得不可能である。この問題に対し、本論文では、FOrward Simulation for Situation Evaluation (FOSSE) アプローチを提案する。本提案手法は、シミュレーションによって未来の試合状況を生成し、生成された未来の試合状況と現在の試合状況を使用して現在の試合状況を評価する。

FOSSE アプローチは二段階で構成される。一段階

*連絡先：大阪府立大学大学院 人間社会システム科学研究科
〒599-8531 大阪府堺市中区学園町 1-1
E-mail: tomoharu.nakashima@kis.osakafu-u.ac.jp

目は、過去と現在の試合状況から未来の試合状況を予測するシミュレーションモデルを用いた順シミュレーションである。二段階目は、順シミュレーションで予測した複数局面と現在の局面を入力とし現在の試合状況の *SituationScore* を出力する局面評価モデルを用いた状況評価である。シミュレーションモデルとして、Recurrent Neural Network (RNN), そして、局面評価モデルとして、Deep Neural Network (DNN) を用いる。

RoboCup サッカーシミュレーションでは、プレイヤーは自身の視野に入るオブジェクトの情報のみが入手可能である。聴覚情報から近隣のプレイヤーより補完できる情報が送られてくることもあるが、それでもフィールド全体の情報を把握することは事実上不可能である。そこで本論文では、プレイヤーエージェントの視点ではなく、全体の試合状況を俯瞰して見ることができるコーチエージェント、もしくは、観客の視点からの状況評価に着目する。また、本論文における“状況評価”や“局面評価”は、現在攻撃しているチームの優勢度合や得点の可能性などの試合状況を理解するためのものを意味することに留意する。他の研究として、Fukushimaら [3] のように個々のプレイヤーエージェントが次の行動決定をする際の状態または行動の評価値を意味している研究もある。本論文における状況評価は将来的にそのような目的での使用も考えられるが、本論文では議論の対象外である。

以下の章では、まず最初に2章でRoboCupと局面評価のための指標について説明する。次に、3章では、局面評価モデルが複数局面を用いることで単一局面のみを用いるよりも高い精度を示すことを調査する。また、実データを用いて未来の情報が過去の情報よりも試合状況を評価する上で有効であることを検証する。最後に、4章で、数値実験から提案手法であるFOSSEアプローチが有効であることを示す。

2 RoboCup サッカーにおける試合状況の定量的評価

本論文では、RoboCup サッカーシミュレーション2Dリーグ [1] を研究対象とする。一般的に、試合中のチームの優勢劣勢を判断する際、どちらのチームがボールを保持しているか、または、サッカーフィールド上のボールの位置座標が指標として用いられる。しかしながら、そのような簡潔な指標では試合中のチームの優勢度合を正確に把握することはできない。そのため、試合状況を定量的に評価することができる指標が求められる。

Nakashima and Pomas [2] は、次のゴールするまでの時間を用いて試合状況の評価指標である *SituationScore* を定義した。そして、*SituationScore* を教師として教師

あり学習をさせた機械学習モデルにより、*SituationScore* を予測させることで試合状況の定量的な評価を実現した。本論文においても、*SituationScore* を採用する。ただし、論文 [2] の定義を修正して、試合状況の評価指標として使用する。

以下では、まず、RoboCup サッカーシミュレーション2Dリーグについて説明し、次に、変更を加えた *SituationScore* について述べる。

2.1 RoboCup サッカーシミュレーション2Dリーグ

RoboCup [1] は、ロボット工学と人工知能の開発の発展を目指した研究プロジェクトである。RoboCupは、様々なリーグから構成される。RoboCup サッカーシミュレーション2Dリーグは、そのうちの一つのリーグで、実際のロボットではなく、シミュレーションによる仮想のプレイヤーエージェントを用いる。プレイヤーエージェントは、図1に示すように円で表現され、コンピュータ上の2次元の仮想サッカーフィールドでサッカーを行なう。プレイヤーエージェントとボールの位置は、2次元ベクトルで表現される。一つのCPUが全てのプレイヤーエージェントを制御する一般的な娯楽向けのサッカーゲームとは異なり、各プレイヤーエージェントは独立したエージェントとしてプログラムされているマルチエージェント環境である。1試合は6,000サイクルで構成され、1サイクルは0.1秒で離散化されている。試合が終了すると、各サイクルにおけるプレイヤーエージェントやボールの位置座標などの全ての試合情報が含まれた試合ログが生成される。

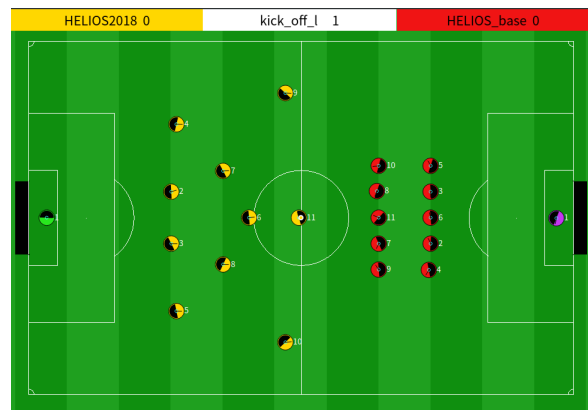


図1: RoboCup Soccer Simulation 2D League.

2.2 *SituationScore*

NakashimaとPomas [2] は、*SituationScore* と呼ばれる試合状況の評価指標を定義した。*SituationScore*

の値は、試合状況がゴールする状況に近づくに連れて上昇する。元々の定義では、 $SituationScore$ の最大値は 100（左チームがゴールしたとき）、最小値は -100 （右チームがゴールしたとき）として、全サイクルにおけるチームの優勢度合を考慮していた。しかしながら、チームの優勢と劣勢の境界に当たる $SituationScore$ が 0 に近い試合状況の評価は難しい。この問題に対して、本論文では定義に変更を加えることで解決する。まず、 $SituationScore$ の下限を 0 と定義し、両チームではなく 1 チームの優勢度合のみを考慮する。また、ゴールした状況から遠い状況ほど正確な状況評価が難しいので、ゴールしたサイクルから 50 サイクル前までの状況を考慮する。本論文で用いる変更を加えた $SituationScore$ を次式で示す：

$$SituationScore(t) = 50 - n, \quad (1)$$

ここで n は、次のゴールまでかかるサイクル数を表し、 $0 < n \leq 50$ とする。右チームに対する $SituationScore$ は、符号を入れ替えることで使用することができる。図 2 は、左チームがゴールする 9 サイクル前の状況を示す。このときの $SituationScore$ は $+41$ である。

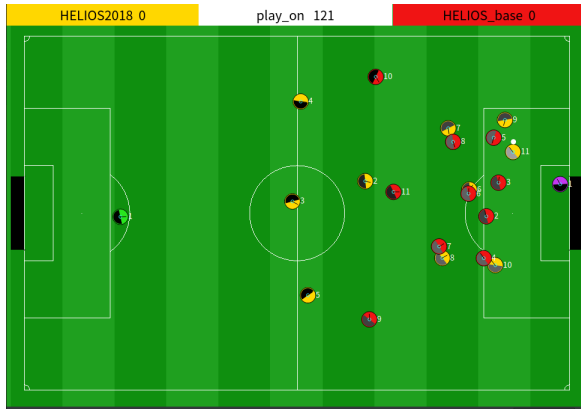


図 2: A situation where the left team scores in nine time cycles.

2.3 データセット

本論文の数値実験で用いるデータセットの作成手順を以下に示す：

1. HELIOS2018 [4]（左側）と agent2d [5]（右）で試合を実施。
2. 試合ログより、左チーム (HELIOS2018 [4]) がゴールしたサイクルを抽出。
3. 各ゴールの前 50 サイクル分の数値情報をそれぞれの $SituationScore$ と共に保存。数値情報は、

22 体のプレイヤーエージェントとボールの位置座標。 $SituationScore$ は式 (1) に基づいて計算され、 $SituationScore$ の値は、状況評価の真値として用いる。

データセットは、1,000 試合から抽出した 394,350 サイクルの数値情報から生成した。また、データセットは以下のように三分割して使用した:学習用データ 274,500 cycles (5,490 goals \times 50 cycles), 検証用データ 39,400 cycles (788 goals \times 50 cycles), テスト用データ 80,450 cycles (1,609 goals \times 50 cycles)。以下に示すすべての実験において、上記により作成されたデータセットを用いる。

3 複数局面を用いた状況評価

3.1 局面評価モデル

本章では、試合状況評価において複数局面を用いることの有効性を検証する。局面評価モデルとしてシンプルな DNN を用いる。局面評価モデルは、サイクル t における $SituationScore$ の値を出力する。図 3 に DNN モデルの概要を示す。図中の \mathbf{X} は、プレイヤーエージェントやボールの位置座標などの試合状況を示す。 \mathbf{X}_t は、現在の試合状況（サイクル t ）の情報を示す。 \mathbf{X}_{t-n_p} は、過去の試合状況の情報を示す（現在のサイクル t から n_p サイクル前）。 \mathbf{X}_{t+n_f} は、未来の試合状況の情報を示す（現在のサイクル t から n_f サイクル後）。

次節の数値実験では、様々な入力を与えた時の局面評価モデルの性能を比較し調査する。

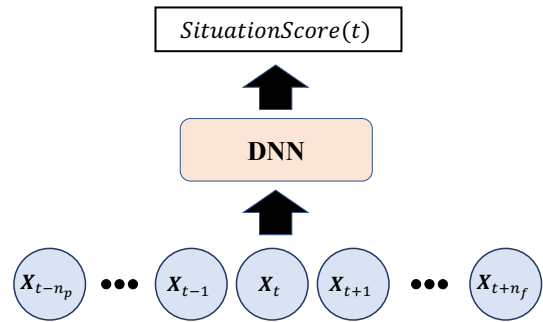


図 3: The overview of Deep Neural Network.

3.2 数値実験

3.2.1 実験設定

前節で説明した局面評価モデルを、2.3 説の手続きにより作られたデータセットで学習させ、局面評価の性

能を調査する．以下のように DNN モデルの入力を変え，四つのモデルを比較する．

Model 1: 単一局面（現在の試合状況のみ）

Model 2: 複数局面（現在，過去，未来の試合状況）

Model 3: 複数局面（現在，過去の試合状況）

Model 4: 複数局面（現在，未来の試合状況）

図 4～7 に各モデルの構造を示す．各モデルはユニット数 16 の中間層が 20 ある．DNN モデルの学習のために，バッチサイズを 64 とし，オプティマイザには Adam [6] を $\alpha = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ で用いた．表 1 に実験設定を示す．入力として与える試合状況における 1 サイクルの数値情報の次元数は，以下の 3 種類で実施する：2 次元（ボールの位置座標），24 次元（ボールと左チームのプレイヤーエージェントの位置座標），46 次元（ボールと全プレイヤーエージェントの位置座標）．Model 2, 3, 4 には，5 サイクル分の過去と未来の試合状況の情報を用いる．

モデルの精度を測定する指標として，平均絶対値誤差（MAE）を用いる．

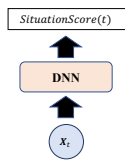


図 4: Model 1.

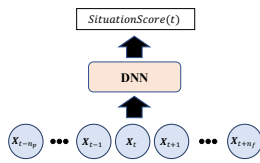


図 5: Model 2.

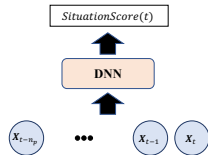


図 6: Model 3.

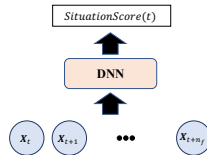


図 7: Model 4.

3.2.2 実験結果

表 2 に実験結果を示す．結果から複数局面を使うモデルが単一局面を使うモデルよりも予測精度が高いことが分かる．また，単一局面を入力として用いる Model 1 の誤差が最も大きく，複数局面を用いるその他のモデル（Model 2, 3, 4）では動的な試合の中におけるチームの優勢度合を捉えることができたためと考えられる．

さらに，未来の情報を用いることの有効性も示唆されている．Model 3 よりも，Model 2 や Model 4 の

方が誤差が小さい．状況評価の際に，過去の情報よりも未来の情報を用いることが有効であることが分かる．これより，現在の試合状況が過去どのように形成されたかよりも，現在の試合状況からどのように展開していくかが現在の状況評価にとって重要であると言える．

ここまでの実験で，複数の未来の情報を用いることの有用性が示された．しかしながら，試合中に未来の情報を得ることができないため，Model 2 や Model 4 は現実的ではないという問題がある．次の章では，この問題に対する提案手法について述べる．

4 FOSSE アプローチ

前章で，過去や未来の情報を用いることで局面評価モデルの性能を高めることが示された．特に，未来の情報を用いることは比較対象の他のモデルよりも有効であることが分かった．しかしながら，実応用を考えた際に未来の情報を得ることができない問題がある．この問題に対して，本論文では FOSSE (FOward Simulation for Situation Evaluation) アプローチを提案する．

図 8 に FOSSE アプローチの概要を示す．提案手法は，順シミュレーションと状況評価の二段階で構成される．一段階目の順シミュレーションでは，過去と現在の試合状況を用いて未来の試合状況を予測する．二段階目の状況評価では，予測された未来の試合状況と現在の試合状況の情報を用いて，サイクル t における $SituationScore$ を出力する．

以下の節では，FOSSE アプローチの各段階について説明する．まず，4.1 節で，一段階目の順シミュレーションについて詳細に説明する．次に，4.2 節で，二段階目に当たる FOSSE アプローチをによる状況評価について述べる．最後に，4.3 節で数値実験により提案手法の有効性を検証する．

4.1 順シミュレーション

図 9 に順シミュレーションの概要を示す．順シミュレーションは，過去と現在の試合状況を入力とし，未来の試合状況を予測して出力する．

4.1.1 Recurrent Neural Network

RNN は，時系列データを扱うことができるニューラルネットワークである．時刻 t の RNN ブロックは，時刻 $t-1$ の RNN ブロックの出力ベクトルを入力として使用する．そして，RNN ブロックからの出力ベクトルを次の時刻 $t+1$ の RNN ブロックの入力として用いる．未来の試合状況はこの手順によって予測される．未来の状況予測のためのこの手順を順シミュレー

表 1: Experimental settings.

Input information	Single situation (current)
	Multiple situations (current, past, and future)
	Multiple situations (current and past)
	Multiple situations (current and future)
Dimensionality for one situation	2 inputs (Ball position)
	24 inputs (Ball position, Left team's player positions)
	46 inputs (Ball position, All player positions)

表 2: Experimental results.

Dimensionality of one situation	Model	MAE
2 inputs	1	3.94
	2	3.38
	3	3.76
	4	3.31
24 inputs	1	3.84
	2	3.36
	3	3.57
	4	3.11
46 inputs	1	3.51
	2	3.32
	3	3.45
	4	3.07

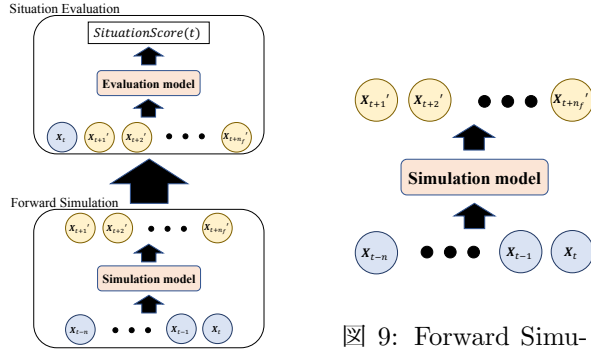


図 8: FOSSE approach.

シオンと呼ぶ。図 10 にこの手順を示す。この図では、 $t-n$ サイクルから現在の t サイクルまでの試合状況を用いて、未来の $t+1$ サイクルの試合状況を予測している。時系列 $\{X_{t-n}, \dots, X_t\}$ の各情報は同じブロックで再帰的に処理される。ブロックは一般的な RNN の中間層を表している。時系列の最後の部分がブロックによって処理された後、予測された次の試合状況が全結合層 (FC) によって生成される。

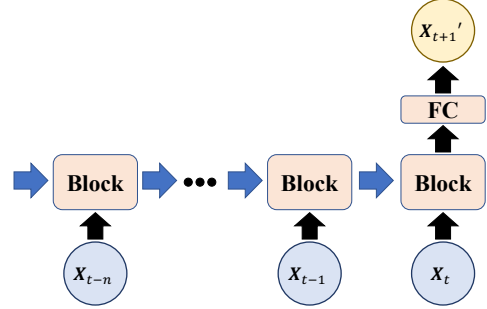


図 10: Recurrent Neural Network for Forward Simulation.

4.1.2 関連研究

RNN を用いた順シミュレーションに関する研究はいくつか見受けられる。Aida ら [7] は、RNN により周辺の車両の軌道予測を実現した。彼らは、軌道予測を自動車の自動運転に応用した。また、ヘルスケアの分野では、Edward ら [8] が電子医療記録から得られた膨大な時系列データを使用して、医師の診断と患者への薬の投与順序を RNN によって予測した。

Zhiyuan ら [9] は、RNN を拡張した Long Short-Term Memory (LSTM) [10] を用いて飛行機の軌道予測に応用した。飛行機の軌道予測は単純な軌道予測よりも困難であるにもかかわらず、高い予測精度を示した。Alexandre ら [11] は、周りの人々の情報を共有する Social pooling を導入した LSTM を用いて群衆における人々の軌道予測に取り組んだ。Wu ら [12] は、LSTM を用いて動画からリアルタイムに未来の人間の姿勢を推定する手法を提案した。

上記の関連研究から、RNN は時系列データを用いて未来の状態を予測する有効な手法であることが示された。さらに、難しいタスクにおいても LSTM が有効であることが分かった。これらの関連研究に基づいて、本論文では、シミュレーションモデルの RNN の構造として LSTM を採用する。

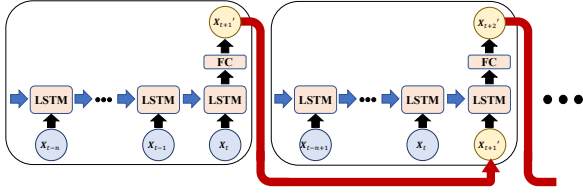


図 11: The architecture of forward simulation using LSTM.

4.1.3 順シミュレーションの予備数値実験

本節の数値実験では、LSTM を用いた順シミュレーションの精度を調査する。LSTM の学習のために、バッチサイズは 512 に、オプティマイザは Adam を $\alpha = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ として使用した。LSTM は、単一局面の情報と、前の LSTM ブロックの出力ベクトルを用いて 512 次元の出力ベクトルを生成する。出力ベクトルは次の LSTM ブロックの入力として扱われる。

順シミュレーションにより予測する未来の試合状況数は、入力に使用する過去の試合状況数によって決まる。図 11 に手順を示す。例えば、モデルが現在までの 5 サイクル分の試合状況（過去 4 サイクル分の試合状況

\mathbf{X}_{t-4} , \mathbf{X}_{t-3} , \mathbf{X}_{t-2} , \mathbf{X}_{t-1} , 現在の試合状況 \mathbf{X}_t) を入力として、予測した次のサイクル $t+1$ の試合状況 \mathbf{X}'_{t+1} を出力する場合を挙げる。全結合層 (FC) は、入力として与えた時系列データの最後の処理の後に、予測した次のサイクル $t+1$ の試合状況 \mathbf{X}'_{t+1} を出力するために扱われる。そして、 \mathbf{X}'_{t+2} は、予測した \mathbf{X}'_{t+1} を含む次の 5 サイクル分の試合状況 (\mathbf{X}_{t-3} , \mathbf{X}_{t-2} , \mathbf{X}_{t-1} , \mathbf{X}_t , \mathbf{X}'_{t+1}) を入力として出力される。この処理手順を 4 回繰り返すことで、現在から 4 サイクル後の未来の状況 \mathbf{X}'_{t+4} を出力する。予測された 4 サイクル後の未来の状況 \mathbf{X}'_{t+4} と実際の状況 \mathbf{X}_{t+4} の差異を誤差として順シミュレーションにおける学習モデルの評価指標とする。具体的には、予測した試合状況と実際の試合状況における各プレイヤーエージェントとボールの位置座標の MAE を用いる。

表 3 に 実験の結果を示す。実験結果から、3 サイクル後の予測よりも 5 サイクル後の予測の方が誤差が大きいことが分かる。容易に予想できることではあるが、実際の値の代わりに予測した結果を積み重ねて予測を繰り返していくために、遠い未来であればあるほど予測が難しいと言える。

表 3: Experimental results of forward simulation using LSTM.

# of situations	Dimensionality of one field information	MAE
Three	2 inputs	1.27
	24 inputs	0.67
	46 inputs	0.62
Five	2 inputs	1.89
	24 inputs	1.12
	46 inputs	1.34

4.2 FOSSE アプローチを用いた試合状況の評価

3 章では、DNN モデルに未来の情報を用いることがより正確な状況評価に有効であることが示された。この実験結果に基づいて、未来の試合状況を用いる DNN を局面評価モデルとして用いる。未来の情報は試合中に扱うことができないため、4.1 節で述べた順シミュレーションによって未来の状況を予測する。本研究では、LSTM を FOSSE アプローチの順シミュレーション部分に用いる。提案する FOSSE アプローチの構造を図 12 に示す。提案手法では、順シミュレーションによって生成された未来の試合状況 ($\mathbf{X}'_{t+1}, \dots, \mathbf{X}'_{t+n_f}$) を使用して現在の試合状況を評価する。DNN モデルと RNN モデルは別々に構築される。図 12 の下部に示すように、順シミュレーションで LSTM モデルは次のサイクルの試合状況を予測する。状況評価部分では、順シミュレーションで予測された未来と現在の試合状況を用いて現在の試合状況の *SituationScore* を推定する。

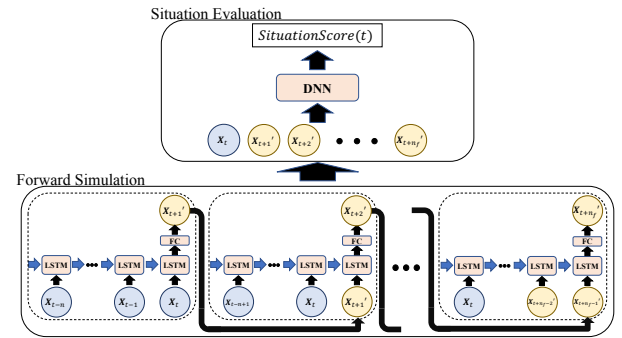


図 12: The overview of our FOSSE architecture with DNN and LSTM.

4.3 数値実験

4.3.1 実験設定

表 4 に、本章の数値実験の実験設定を示す。単一局局面（現在）、複数過去局面（現在+過去）、複数未来局面（現在+未来）、提案手法（現在+予測未来）の四つの DNN モデルの精度の比較を行なう。

4.3.2 実験結果

表 5 に実験結果を示す。結果から、提案手法が単一局局面を用いるよりも高い精度を示した。また、3 サイクル分の試合状況の情報を用いる場合では、提案手法の精度は複数過去局面を用いた場合よりも高かった。したがって、順シミュレーションにより予測された未来の試合状況を用いることが、過去の情報を使用して状況評価するよりも有用であることが分かった。複数未来局面を用いた場合の精度は最も高かったが、この場合では本来試合中に取得できない未来の状況を使用している。現実的な実時間アプリケーションを想定した際には、未来の値は使用することができないため、このモデルを実際に用いることはできない。一方で、提案手法は順シミュレーションによって未来の試合状況を予測するため、試合中にリアルタイムで使うことができる。

また、5 サイクル分の試合状況が使用される場合では、提案手法は単一局局面を使用したときよりも高い精度を示した。一方で、複数過去局面を用いたモデルと比べ有効性を示すことができなかった。この原因として、4.1 節でも挙げたように、シミュレーションモデルの出力の誤差は予測する未来が先であればあるほど大きくなることが考えられる。本論文において、シンプルなシミュレーションモデルを使用したため、より精巧なシミュレーションモデルを使用することで向上する可能性がある。シミュレーションモデルの精度向上を今後の課題とする。

数値実験によって、順シミュレーションと状況評価を組み合わせた提案手法の有効性が示された。試合の状況を正確に把握することは、サッカーだけではなく、多くのスポーツでの勝利のために重要である。本論文で提案した FOSSE アプローチは、他のスポーツでの状況評価に対しても大きく貢献できる。

5 おわりに

本論文では、RoboCup サッカーシミュレーション 2D リーグの試合状況を評価するための FOSSE アプローチを提案した。試合状況を評価する上で三つの貢献を述べた。まず初めに、単一局局面よりも複数局面を用い

ることの有効性を示した。次に、未来の試合状況の情報が過去の情報よりも有用であることを示した。最後に、主な貢献である、順シミュレーションによって生成された未来の情報を使用した状況評価を行なう FOSSE アプローチを提案した。FOSSE アプローチは、順シミュレーションと状況評価の二段階で構成される。

本論文において、FOSSE アプローチは複数未来局面を入力とした DNN モデルを状況評価の局面評価モデルとして、LSTM を順シミュレーションのシミュレーションモデルとして使用した。数値実験から、提案手法の有効性が示された。本提案手法は、進行中の試合をリアルタイムで評価することが可能である。FOSSE アプローチは、サッカーのみならずバスケットボールやラグビーなどのスポーツにおいても適用可能であることが期待できる。

6 今後の課題

本論文の数値実験で用いたデータセットは、二つのチームのみを使用した試合から生成した。様々なチームが参加している RoboCup の大会など実用的な応用までを考慮した際に、提案手法が他のどのチームに対しても機能する汎用性を示す必要がある。今後の課題として、提案手法の汎用性についての検証を挙げる。そのために、学習用データセットに含まれていない未知のチームに対して学習済みのモデルの性能の調査を行なう必要がある。

さらに、4.3 節の数値実験で既に述べたように、シミュレーションモデルの予測精度を向上させることも課題として挙げられる。例えば、中間層の層数や、入力と与える情報を変えるなど、異なるシミュレーションモデルの構造を検討する。他の案として、画像データを用いることが試合状況の評価において良い予測精度を示すことが Nakashima and Pomas [2] によって示されているため、本論文で提案した FOSSE アプローチを画像データに適応した構造で使用することで、高い精度での状況評価が期待できる。また、人間の思考プロセスを計算機で再現する機械学習手法を検討する。人間の思考プロセスを機械学習手法に取り入れることは、人工知能分野の発展に貢献する可能性がある。

最終的には、提案手法を RoboCup サッカーのチームに実装し、試合中の戦略切替の指針として適用することを目的とする。また、チームへの実装だけではなく、観客のエンターテインメント性の向上のためのアプリケーションへの応用も今後の課題とする。

表 4: Experimental settings of evaluating situation.

Input	Single Past+current (Multiple-Past) Future+current (Multiple-Future) Predicted future+current (Multiple-Predict)
Dimensionality per situation	2 inputs (Ball position) 24 inputs (Ball position and Left team's player position) 46 inputs (Ball position and All player position)
# of input situations	1, 3, 5

表 5: Experimental results.

Dimensionality	Input	MAE (# of time cycles)		
		One	Three	Five
2 inputs	Single	3.94	-	-
	Multiple-Past	-	3.70	3.76
	Multiple-Future	-	3.33	3.31
	Multiple-Predict (proposed)	-	3.63	3.77
24 inputs	Single	3.84	-	-
	Multiple-Past	-	3.62	3.57
	Multiple-Future	-	3.35	3.11
	Multiple-Predict (proposed)	-	3.55	3.59
46 inputs	Single	3.51	-	-
	Multiple-Past	-	3.53	3.45
	Multiple-Future	-	3.28	3.07
	Multiple-Predict (proposed)	-	3.44	4.09

参考文献

- [1] Hitoaki Kitano, Minoru Asada, Yasuo Kuniyoshi, Itsuki Noda, Eiichi Osawa, and Hitoshi Matsubara, "RoboCup: A Challenge Problem for AI", *AI Magazine*, Vol.18, No.1, pp.73–85, 1997.
- [2] Tomoharu Nakashima, and Tanguy Pomas, "Evaluation of Situation in RoboCup 2D Simulations using Soccer Field Images", *Proc. of the 22th annual RoboCup International Symposium*, 12 pages, 2018.
- [3] Takuya Fukushima, Tomoharu Nakashima, and Hidehisa Akiyama, "Mimicking an Expert Team through the Learning of Evaluation Functions from Action Sequences", *Proc. of the 22th annual RoboCup International Symposium*, 12 pages, 2018.
- [4] Hidehisa Akiyama, Tomoharu Nakashima, Yudai Suzuki, An Ohori, and Takuya Fukushima, "HELIOS2018: Team Description Paper", *RoboCup2018 Montreal*, 6 pages, 2018.
- [5] Hidehisa Akiyama, and Tomoharu Nakashima, "HELIOS Base: An Open Source Package for the RoboCup Soccer 2D Simulation", in *RoboCup 2013: Robot Soccer World Cup XVII*, Springer, Berlin, Heidelberg, pp.528–535, 2014.
- [6] Diederik P Kingma, and Jimmy Ba, "Adam: A method for stochastic optimization", *Proc. of International Conference on Learning Representations*, 2015.
- [7] Aida Khosroshahi, Eshed Ohn-Bar, and Mohan Manubhai Trivedi, "Surround Vehicles Trajectory Analysis with Recurrent Neural Networks", *Proc. of the IEEE 19th Conference on Intelligent Transportation Systems (ITSC)*, pp.2267–2271, 2016.
- [8] Edward Choi, Mohammad Taha Bahadori, Andy Schuetz, Walter F. Stewart, and Jimeng Sun, "Doctor AI: Predicting Clinical Events via Recurrent Neural Networks", *Proc. of the Machine Learning for Healthcare 2016*, pp.301-318, 2016.

- [9] Zhiyuan Shi, Min Xu, Quan Pan, Bing Yan, and Haimin Zhang, “LSTM-based Flight Trajectory Prediction”, *Proc. of the 2018 International Joint Conference on Neural Networks (IJCNN)*, pp.1–8, 2018.
- [10] Sepp Hochreiter, and Jrgen Schmidhuber, “Long Short-Term Memory”, *Journal of Neural Computation*, Vol.9, No.8, pp.1735–1780, 1997.
- [11] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese, “Social LSTM : Human Trajectory Prediction in Crowded Space”, *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pp.961–971, 2016.
- [12] Wu Erwin, and Hideki Koike “FuturePose-Mixed Reality Martial Arts Training Using Real-Time 3D Human Pose Forecasting With a RGB Camera”, *Proc. of 2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp.1384–1392, 2019.

RoboCupサッカーにおける ヘテロマッチングが戦術に与える影響

Influence of Hetero Matching on Tactics in RoboCup Soccer

福島 卓弥^{1*} 山口 将貴¹ 中島 智晴¹ 秋山 英久²
Takuya Fukushima¹ Masaki Yamaguchi¹ Tomoharu Nakashima¹ Hidehisa Akiyama²

¹ 大阪府立大学大学院
¹ Osaka Prefecture University
² 福岡大学
² Fukuoka University

Abstract: This paper discusses the effective use of the hetero-genious players in RoboCup soccer simulation 2D league. The RoboCup server provides a set of 15 player types with different abilities. The team can pick up 11 player types according to its team strategies and tactics. This paper considers the idea of so-called hetero matching where several players switch their roles during the game. For example, from an offensive point of view, a attacker who is able to dash faster should be assigned to a slower defender. On the other hand, from the defensive perspective, a faster defender should be marked to a faster attacker. In the computational experiments, we switch side attacker to their other side in order to achieve better match-up against opponent defenders. Moreover, in the same way, the role of defenders are changed so that the difference in the abilities against the opponent forward players gets bigger. Through the experiments, we evaluate the effectiveness of using the hetero matching by examining the team performance through the analysis with kick distributions.

1 はじめに

多くの対戦型スポーツには戦術があり、個々のスキルと同等に勝敗を決める重要な要因である。一般的に、スポーツにおける戦術には相手（チーム）との相性が重要である。しかし、戦術は複雑であり、人間が暗黙知として戦術を認識することはできても、数値情報として目視してもはっきりと判断できない場合が多い。

一方で、スポーツのデータ分析に関する研究が盛んに行われている。例えば、野球においては、アメリカ野球学会（SABR）は野球の記録や人物について調査、研究を行っている。その名称を用いたセイバーメトリクスという測定基準を用いた戦術分析が重視されている [1]。また、Yamamotoら [2] はサッカーにおける特定の状況に対して特徴を可視化することで、戦術分析を行っている。Fewellら [3] はバスケットボールにおけるプレイヤー間の戦術的なネットワークを解析している。その他にも、様々なスポーツにおいて可視化や機械学習を用いたデータ分析が行われている [4][5]。

RoboCupサッカーシミュレーション2Dリーグにおいても、度重なる研究開発の成果により、多種多様な戦術を組み立てることが可能になってきている。近年の世界大会では、敵に応じた戦術を実装するチームが多く見られる。戦術間には相性があるため、より有利な試合運びを行い確実な勝利を実現するためには、現在の敵に適した戦術に切り替えることが必要である。これらの背景から、サッカーのチーム戦術に注目した研究は、人間のサッカーだけでなく計算機によるシミュレーションサッカーにおいても需要が高まっている。たとえば、Abreuら [6] は、セットプレイ後の動きとして最も敵戦術に適した手法を機械学習手法を用いて推定している。また、Henrioら [7] はコーナーキック時におけるフォーメーションを敵に応じて切り替える仕組みを提案している。

チーム全体の戦術を分析する手法として、Nakashimaら [8] は、キック分布に注目したクラスタリング手法を提案している。プレイヤーのキック情報を分布として捉え、キック分布間距離によって戦術を解析する手法であり、各チームの戦術的特徴を表現可能な手法の一つである。また、著者らはキック分布をキック確率分布

*連絡先：大阪府立大学大学院人間社会システム科学研究科
〒599-8531 大阪府堺市中区学園町 1-1
E-mail: takuya.fukushima@kis.osakafu-u.ac.jp

に変換し、確率分布間の距離測度を用いることで計算量を削減し、人間の主観との比較を行うことで、その手法の妥当性を示している [9]。これらの手法は、どちらもチーム全体の戦術を分析する手法であるが、個体間の相性を評価したものではない。

一方で、試合状況を大きく変える行動、例えばスルーパスなどは、パスレシーバとディフェンダの個体間の身体性やスキル完成度といった観点がある。スキル成功の要因となる。身体性の相性では、走力の高いプレイヤーを敵方のうち走力が低いプレイヤーとマッチアップさせることで、スルーパスの成功可能性を高めることができる。本論文では、個体間の身体性の相性に注目し、相性の良いマッチアップによりチーム性能を向上させることを目標とする。また、これにより得られる戦術やチーム性能の変化を確認することで、相性の良いプレイヤーを割り当てることの有効性を確認する。

2 戦術表現

RoboCup サッカーシミュレーション 2D リーグにおいて、近年多様なチーム戦術が実装されている。チーム戦術を表現する手法として、フォーメーションに注目したものと、行動選択に注目したものがある。例えばフォーメーションに注目した研究として、Faria ら [10] は、機械学習手法を用いた敵のフォーメーション推定を行っている。また、フォーメーションを戦術単位で分類し、クラスタリングを行う手法も提案されている [11]。

一方で、行動選択に注目した研究として、たとえば Bombini ら [12] は、行動パターンを用いてプレイヤーの行動を分類する手法を提案している。また、著者らはこれまでに行動軌跡に注目した評価関数モデリングにより強豪チームの戦術を模倣する研究を行った [13]。行動軌跡の例を図 1 に示す。さらにキック分布やキック確率分布を用いた行動軌跡間の類似度分析により、人間の主観に合致した分析を可能にしている [8],[9]。

これらの文献は全てチーム戦術に注目した手法であり、敵チームの戦術分析や自チームの戦術作成に応用され、戦術の相性問題に取り組んでいる。しかし、個体間の相性に注目したアプローチについては十分な議論がされていない。本論文では、個体間の相性に注目した手法を用いて、その個体間相性によって得られる戦術の変化を検証する。

3 ヘテロマッチング

RoboCup サッカーシミュレーション 2D リーグでは、試合開始時にサッカーサーバがヘテロロニアスなプレイヤーエージェントを 18 体生成する。生成されたプレ

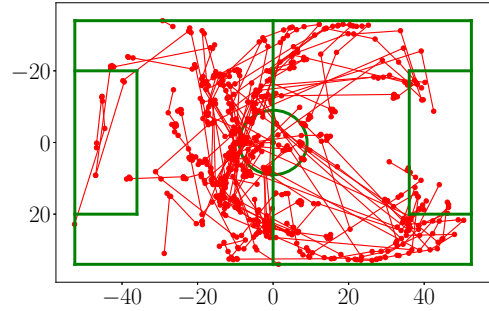


図 1: An example of action trajectories

イヤーは走力やキッカブルエリアの半径といった特性が異なる。各チームのコーチエージェントは、生成された 18 体のプレイヤーから 11 体を選択し、チームが採用しているフォーメーションのポジションに割り当てる。例えば、多くのチームのベースチームとして用いられる Agent2D [14] では、センターフォワードやセンターバックに足の速いプレイヤーを割り当てている。スルーパスを狙う戦術であれば、サイドフォワードに足の速いプレイヤーを割り当てる。このとき、マッチアップする敵のディフェンダとの走力の差が大きければスルーパスの成功確率が高くなることが期待される。

RoboCup サッカーシミュレーション 2D リーグにおいて、ほとんどのチームは敵のプレイヤー割当に依らず、自チームの戦術に応じた割当案を採用し、プレイヤーのスタミナがなくなるまではそのプレイヤー割当が変更することはない。しかし、より敵の戦術に応じた戦術を確立していくためには、敵のプレイヤー割当に応じて自チームのプレイヤー割当を変更する必要がある。本論文では、これをヘテロマッチングと呼ぶことにする。ヘテロマッチングを用いることによる自チームの戦術の変化とその影響を調査する。具体的なヘテロマッチングの方法については、数値実験の章で説明する。

4 類似度分析

戦術の変化を確認するためには、キック分布やキック確率分布間距離を用いた類似度分析が有効であることが示されている [8],[9]。本論文においても、ヘテロマッチングによる戦術の変化をキック確率分布間距離を用いて確認する。

4.1 キック分布・キック確率分布

キック分布は、試合中にプレイヤーがキックした位置とボールの移動量との組の集合である。キック位置にボールの移動量を重みとして割り当てていると解釈することもできる。プレイヤーにキックされたボールの位

置 (x, y) をベクトル \vec{b} とし、そのキックによってボールが動いた距離をベクトルの重み $w_{\vec{b}}$ とする。図 2 に図 1 から変換されたキック分布の例を示す。図 2 において、縦棒の座標はボールがキックされた位置を示し、高さはベクトルの重み、すなわちそのキックによってボールが動いた距離を表している。

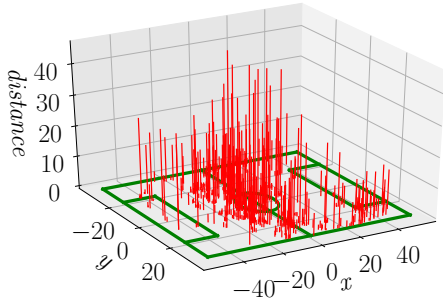


図 2: An example of a kick distribution

キック分布は有限数の 3 次元ベクトルからなる集合であるため、二つのキック分布間の距離を計算すると分布内の要素と要素の組み合わせ爆発により計算コストが膨大になる可能性がある。そこで、キック分布を連続的に表現可能な確率分布に変換することで、距離計算のコスト削減を図る。本研究では、カーネル密度推定 (KDE) を用いてキック分布をキック確率分布に変換する。KDE は有限の標本から母集団の確率密度関数を推定するノンパラメトリック手法である。カーネル関数は d 次元のガウシアンカーネルとし、平滑化のためのパラメータであるバンド幅 h は以下の式で定義する。

$$h = n^{-\frac{1}{d+4}}$$

ここで、 n はデータ数、 d は次元数を示す。キック分布において、ベクトル \vec{b} (2 次元) とその座標での重み $w_{\vec{b}}$ が入力として扱われるため、キック確率分布の次元数も $d = 3$ となる。カーネル密度推定を用いて得られるキック確率分布を $p(\vec{b}, w_{\vec{b}})$ とする。 $p(\vec{b}, w_{\vec{b}})$ は、座標 \vec{b} で長さ $w_{\vec{b}}$ のキックが行われる確率を意味する。図 2 のキック分布をキック確率分布に変換した例を図 3 に示す。図 3 では、座標 $\vec{b} = (0.0, 0.0)$ で長さ $w_{\vec{b}}$ が行われる確率分布を表している。図 3 のように、キック確率分布では分布が連続的に表現される。

4.2 分布間距離

カーネル密度推定により得られたキック確率密度分布について、二つのキック確率密度分布の類似度を求めるため、以下で説明する分布間の距離尺度を用いる。

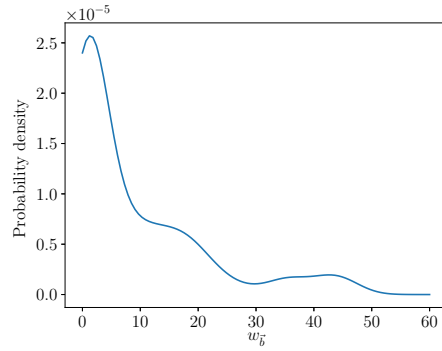


図 3: An example of a kick probability distribution when $\vec{b} = (0.0, 0.0)$

分布間距離尺度の一種として、分布間の密度差推定を用いた L^2 距離が知られている。確率密度関数 $p(\mathbf{x})$ と $q(\mathbf{x})$ の距離を以下の式で定義される L^2 距離により表す。

$$L^2(p, q) = \int (p(\mathbf{x}) - q(\mathbf{x}))^2 d\mathbf{x}$$

ここで、確率密度関数 $p(\mathbf{x})$ と $q(\mathbf{x})$ は 4.1 節で説明したキック確率分布とする。 L^2 距離は、非負性、非退化性、対称性、三角不等式といった距離の公理を満たすことが知られている。厳密に正確な L^2 距離を求めることは困難であるが、確率密度関数を用いることで推定することができる。

5 数値実験

5.1 実験設定

数値実験では、ヘテロマッチングを用いることで戦術に変化がoccurかどうかを、キック確率分布間距離を用いて調査する。自チームとして HELIOS2019 を用いる。チーム HELIOS は世界大会で優勝経験のある強豪チームである。敵チームとして

- HELIOS2019 (ヘテロマッチングなし) [15]
- MT2018 [16]
- Oxsy2018 [17]
- WrightEagle [18]

の 4 チームを用いる。敵チームはそれぞれ、世界大会で上位の成績を残す強豪チームである。敵チームのプレイヤー割当に応じて、自チームのプレイヤーの割当を変更する。本論文では自チームのスループス数増加と敵チームのスループス数減少を目的として、以下の二つポジションにおいてヘテロマッチングを行う。

- サイドフォワードの2プレイヤーの左右を入れ替え
- ディフェンダ（サイドバックとセンターバック）の4プレイヤーのポジションを入れ替え

また、ヘテロマッチングの指標として、以下二つの走力指標を用いる。

- 5m 走るのに必要なサイクル数 (5m)
- 30m 走るのに必要なサイクル数 (30m)

入れ替え対象である、サイドフォワードの2プレイヤーとディフェンダの4プレイヤーを図4に示す。赤い円で囲まれたプレイヤーはサイドフォワード、黄色い円で囲まれたプレイヤーはディフェンダを意味する。ヘテロマッチングを行わない設定 (Without) とヘテロマッチングを行う設定を比較し、ヘテロマッチングの性能を評価する。

サイドフォワードに対するヘテロマッチングについては、対応する敵のディフェンダとの指標の差が大きくなるように、ディフェンダに対するヘテロマッチングについては、対応する敵フォワードとの指標の差が小さくなるように入れ替える。つまり、サイドフォワードにおいては、敵ディフェンダとの走力指標の差が大きくなるようにヘテロマッチングを行う。一方、ディフェンダにおいては、敵フォワードとの走力指標の差が小さくなるようにヘテロマッチングを行う。敵チームは試合中ポジション入れ替えを行わないものと仮定する。

各設定において100試合ずつ行い、性能評価を行う。性能評価指標として、得点 (Goals for), 失点 (Goals against), スルーパス回数 (# of t-passes), 被スルーパス回数 (# of opp t-passes), 勝率 (Win rate) を用いる。

5.2 実験結果

各実験設定における試合結果を表1-4に示す。まず、得点とスルーパス回数に注目する。ほぼ全ての実験設定において、得点とスルーパス回数に改善は見られなかった。本実験では、オフENSに関してサイドフォワードのヘテロマッチングを行ったが、従来の割当方法として、どちらも足の速いプレイヤーを起用していることから、ヘテロの入れ替えを行うことによる影響が小さかったことが原因であると考えられる。

また、表1, 3から、失点と被スルーパス回数に関して、チーム HELIOS2019 とチーム Oxy2018 に対して減少傾向にあることが確認できる。これらのチームはサイドフォワードへのスルーパスを頻繁に用いてサイド突破を試みる傾向の強いチームである。ディフェンダと敵フォワードとの相性を合わせることで、スルー

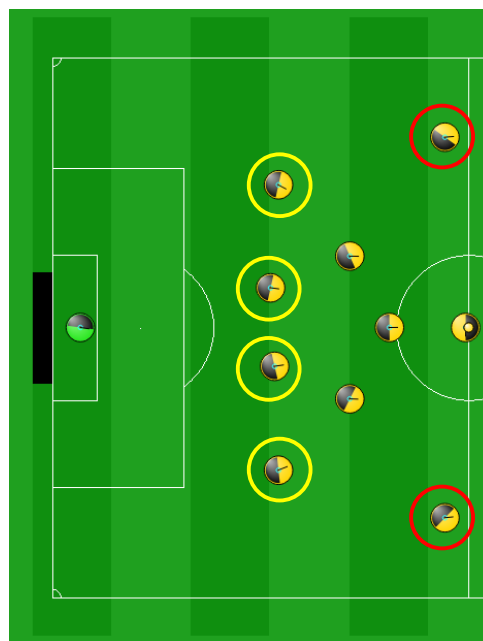


図4: Target players of hetero matching method. Red circles mean Side-Forward players, and yellow circles are Defenders.

パスや失点を抑えることがわかる。一方で、表2から、チーム MT2018 については、5m 指標でのヘテロマッチングを行った場合においては、スルーパス回数が減らず、30m でのヘテロマッチングにおいては、減少することが確認できた。表4から、チーム WrightEagle に対しては、ヘテロマッチングにより失点を抑えることができたが、被スルーパス回数は減少しなかった。チーム MT2018 やチーム WrightEagle はフォワードが2人の2トップフォーメーションであり、中央突破を狙うチームである。中央からドリブルで突破してくるチームにはヘテロマッチングを行うことは有効でないことがわかる。これは、中央からのドリブル突破を防ぐには、足の速さといったパラメータ的要因ではなく立ち回りなどの技術的な要因が大きいことが原因と考えられる。以上のことからディフェンス面でのヘテロマッチングはサイド突破を狙うチームに対して有効であることが分かった。

表5にチーム HELIOS2019 に対するのキック確率分布間距離の平均と分散を示す。実験設定30mのとき、平均値、分散ともに小さくなる傾向が確認できた。ヘテロマッチングを行うことで、距離 (非類似度) の平均や分散を小さくすることは、似た行動軌跡が得られる傾向にあることを示している。これは、敵のプレイヤー割当方法が固定であるため、片側サイドが常に苦手である状況にあることから、ヘテロマッチングによってそのサイドから攻める頻度が高いと推測される。こ

これらの結果から、行動軌跡の類似度分析から、ヘテロマッチングによって敵にとって苦手なサイドを狙って行動選択する確率が上がったと考えられる。

初期配置の時点でプレイヤーの相性がマッチしている場合においては、ヘテロマッチングの機能がなくても片側に偏った行動軌跡になる場合が存在した。ヘテロマッチングを行わない場合においても、ヘテロマッチングの際と同じような状況が生まれる可能性があり、これにより本実験の実験結果を確認しがたいことがわかった。ヘテロマッチングをするべきプレイヤーパラメータが与えられるような乱数のシードに固定した状態で実験を行うことで、本手法の効果はさらに明確に確認することができると考えられる。

表 1: Team performance against HELIOS2019

Setting	Goals for	Goals against	# of t-passes	# of opp t-passes	Win rate
30m	1.09	1.06	6.19	5.67	0.36
5m	1.35	1.15	5.84	6.11	0.44
Without	1.29	1.34	6.2	6.35	0.34

表 2: Team performance against MT2018

Setting	Goals for	Goals against	# of t-passes	# of opp t-passes	Win rate
30m	3.94	0.43	10.01	4.77	0.96
5m	4.16	0.48	9.86	4.99	0.94
Without	4.22	0.4	10.04	4.9	0.97

表 3: Team performance against Oxy2018

Setting	Goals for	Goals against	# of t-passes	# of opp t-passes	Win rate
30m	2.0	0.86	9.07	4.26	0.68
5m	2.21	1.06	9.53	4.57	0.65
Without	1.93	1.11	9.08	4.77	0.53

表 4: Team performance against WrightEagle

Setting	Goals for	Goals against	# of t-passes	# of opp t-passes	Win rate
30m	3.46	1.36	8.52	3.68	0.79
5m	3.28	1.57	8.61	3.84	0.73
Without	3.13	1.68	8.08	3.67	0.70

表 5: L^2 distance

	Without	5m	30m
Mean	9.550×10^{-7}	9.380×10^{-7}	9.320×10^{-7}
Variance	2.569×10^{-13}	3.677×10^{-13}	6.780×10^{-14}

6 おわりに

本論文では、プレイヤー間の相性に注目したヘテロマッチングを提案した。チーム性能と行動軌跡の類似度から、ヘテロマッチングの効果を検証した。ヘテロマッチングにより、得点率やスルーパス回数といったオフENSEの性能は上がらなかったが、失点率や被スルーパス回数といったディフェンスの性能が向上することが

分かった。今後の展望として、ヘテロマッチングの効果をより確かなものにするために、乱数のシードを固定した状態での実験や、全プレイヤーをヘテロマッチング対象にすることなどが挙げられる。

参考文献

- [1] メジャーリーグの数理科学〈上・下〉. シュプリンガー数学リーディングス, 2004.
- [2] Risa Yamamoto and Yohei Nakada. Improvement of Visualization Method for Attack Patterns in Team Sports. In *2018 Joint 10th International Conference on Soft Computing and Intelligent Systems and 19th International Symposium on Advanced Intelligent Systems*, pp. 461–466, 2018.
- [3] Jennifer H. Fewell, Dieter Armbruster, John Ingham, Alexander Petersen, and James S. Waters. Basketball teams as strategic networks. *PloS one*, Vol. 7, No. 11, 2012.
- [4] C. Perin, R. Vuillemot, C. D. Stolper, J. T. Stasko, J. Wood, and S. Carpendale. State of the Art of Sports Data Visualization. *Computer Graphics Forum*, Vol. 37, No. 3, pp. 663–686, 2018.
- [5] サッカーマティクス～数学が解明する強豪チーム～「勝利の方程式」～. 光文社, 2017.
- [6] Pedro Henriques Abreu, Daniel Castro Silva, João Portela, João Mendes-Moreira, and Luís Paulo Reis. Using model-based collaborative filtering techniques to recommend the expected best strategy to defeat a simulated soccer opponent. *Intelligent Data Analysis*, Vol. 18, No. 5, pp. 973–991, oct 2018.
- [7] Jordan Henrio, Thomas Henn, Tomoharu Nakashima, and Hidehisa Akiyama. Selecting the Best Player Formation for Corner-Kick Situations Based on Bayes’ Estimation. In Sven Behnke, Raymond Sheh, Sanem Sariel, and Daniel D Lee, editors, *RoboCup 2016: Robot World Cup XX*, pp. 428–439, Cham, 2017. Springer International Publishing.
- [8] Tomoharu Nakashima, Satoshi Mifune, Jordan Henrio, Oliver Obst, Peter Wang, and Mikhail

- Prokopenko. Kick extraction for reducing uncertainty in RoboCup logs. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2015.
- [9] Takuya Fukushima, Tomoharu Nakashima, and Hidehisa Akiyama. Similarity Analysis of Action Trajectories based on Kick Distributions. In *RoboCup International Symposium 2019*, 2019. 12 pages.
- [10] Brígida Mónica Faria, Luís Paulo Reis, Nuno Lau, and Gladys Castillo. Machine Learning algorithms applied to the classification of robotic soccer formations and opponent teams. In *2010 IEEE Conference on Cybernetics and Intelligent Systems, CIS 2010*, pp. 344–349, 2010.
- [11] Takuya Fukushima, Tomoharu Nakashima, and Hidehisa Akiyama. Online opponent formation identification based on position information. In Hidehisa Akiyama, Oliver Obst, Claude Sammut, and Flavio Tonidandel, editors, *RoboCup 2017: Robot World Cup XXI*, pp. 241–251, Cham, 2018. Springer International Publishing.
- [12] Grazia Bombini, Nicola Di Mauro, Stefano Ferilli, and Floriana Esposito. Classifying agent behaviour through relational sequential patterns. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Vol. 6070 LNAI, pp. 273–282, 2010.
- [13] Takuya Fukushima, Tomoharu Nakashima, and Hidehisa Akiyama. Mimicking an Expert Team Through the Learning of Evaluation Functions. *RoboCup 2018: Robot World Cup XXII*, Vol. 11374, pp. 1–11, 2019.
- [14] Hidehisa Akiyama and Tomoharu Nakashima. HELIOS base: An open source package for the RoboCup soccer 2D simulation. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Vol. 8371 LNAI, pp. 528–535, 2014.
- [15] Hidehisa Akiyama, Tomoharu Nakashima, Takuya Fukushima, Yudai Suzuki, and An Ohori. HELIOS2019 : Team Description Paper. In *RoboCup 2019 Sydney*, 2019. 6 pages.
- [16] Zheng Yang, Ziqiang Liu, Xiaorui Wang, Ning Dong, Xiangben Hu, JingLi Chen, Shengbing, and Lv Gang. MT2018 : Team Description Paper. In *RoboCup 2018 Montreal*, 2018. 5 pages.
- [17] Sebastian Marian, Dorin Luca, Bogdan Sarac, and Ovidiu Cotarlea. OXSYS 2018 Team Description. In *RoboCup 2018 Montreal*, 2018. 6 pages.
- [18] Xiao Li, Rongya Chen, and Xiaoping Chen. WrightEagle 2D Soccer Simulation Team Description 2015. In *RoboCup 2015 Hefei*, 2015. 5 pages.

アーム型ロボットにおける可動範囲に関する一考察

About the range of movement for the arm typed robot.

加藤 大登¹ 鈴木 勇貴¹ 植村 渉^{1*}

¹ 龍谷大学
¹ Ryukoku University

Abstract: 第四次産業革命 (Industrie 4.0) では、工場のオートメーション化が注目されており、柔軟な生産体系が求められる。本研究では、そのような場における、自動搬送ロボット (Automatic Guided Vehicle:AGV) に注目する。従来、AGV が物を取ったり置いたりする時は、その作業対象の前に移動して、静止してから作業を行っていた。しかし、効率化を考えると、作業できる領域に入った段階で作業を開始した方が望ましく、また、他の移動ロボットや人間等の近接を考えると、作業中に移動できた方が望ましい。そこで、本研究では、ロボットアームを搭載した移動式ロボットが、作業をしながら数 cm 程度の移動を行う場合を目標とする。その際に、ロボットの移動を考慮してロボットの先端座標を扱うことで、ロボットがどのように移動しても安定して物体を扱うことができる。本研究では、そのような動きをするために、ロボットの先端座標を固定したときの状況を検討する。

1 はじめに

第四次産業革命 (Industrie 4.0) として、工場のオートメーション化が注目されている。そこでは、大量生産を目的とする今までの工場と異なり、変種変量生産が求められている。自動化の究極は、段取り替えなしで、一品物を次々と作るラインが目標となる [1]。

そのようなラインでは、ロボットアームによる柔軟な組み立て作業が必要となり、その加工機器に対して効率よく部品を提供したり、できあがった加工品を適切に搬送することが必要となる。前者は、World Robot Summit [2] のものづくり分野 [3] として世界大会の開催を予定しており、後者は、RoboCup[4] Logistics League[5] として世界大会が開催されている。本研究では、自動搬送ロボット (Automatic Guided Vehicle:AGV) に注目する。従来、AGV が物を取ったり置いたりする時は、その作業対象の前に移動して、静止してから作業を行っていた。しかし、効率化を考えると、作業できる領域に入った段階で作業を開始した方が望ましく、また、他の移動ロボットや人間等の近接を考えると、作業中に移動できた方が望ましい。そこで、本研究では、ロボットアームを搭載した移動式ロボットが、作業をしながら数 cm 程度の移動を行う場合を検討する。その際に、ロボットの移動を考慮してロボットの先端座標を扱うことで、ロボットがどのように移動しても安定して物体を扱うことができる。本研究では、そのようなアームの制御を行うために、まずはアームの先端を固定し

た時に土台のロボットが移動できる範囲を検討する。

2 移動式ロボットとロボットアーム

本章では、本研究で扱う移動式ロボット Robotino と、ロボットアーム Cobotta について紹介する。

2.1 移動式ロボット: Robotino

RoboCup Logistics League では、ドイツの FESTO 社製の全方位移動型ロボットである Robotino をチーム標準のロボットとして用いている。このロボットには、オムニホイールが 3 個ついており、そのモーターを制御することでどの方位にも進むことが可能である。実際にプログラムする際は、各プログラミング言語からハードウェアを制御する API を呼び出す。各モーターの回転速度を指示する形だけでなく、ロボットの座標系に対する速度を指定することで各モーターの回転速度を計算する関数も用意されている。図 1 に弊研究室のチーム (BabyTigers-R) が 2019 年の世界大会で用いた Robotino を示す。ロボット下部の駆動部分は、各チーム共通であるが、CPU の入っているヘッド部から上の部分は、チーム独自のものとなっている。Robotino には、モーターが 3 個ついており、それぞれの速度 $v_n(t)$ ($n = 1, 2, 3$) は式 1 による PID 制御を行っている [6]。

$$u_n(t) = K_p \left(e(t) + \frac{1}{T_N} \int_0^t e(t') dt' \right) + K_d \dot{e}(t) \quad (1)$$

*連絡先: 龍谷大学理工学部電子情報学科
〒520-2194 滋賀県大津市瀬田大江町横谷 1-5
E-mail: wataru@rins.ryukoku.ac.jp



図 1: BabyTigers-R の 2019 年度 RoboCup 世界大会にて用いた移動式ロボット

ここで, $u_n(t)$ は操作量, K_p は PID 制御の比例項 p , K_i を積分項とし $K_i = 1/T_N$ とする. そして, K_d は微分項である. Robotino では, $k_p = k_i = k_d = 25$ を利用し, 式 2 の値を用いている.

$$K_p = k_p/2, K_i = k_i/1024, K_d = k_d/2 \quad (2)$$

図 2 に, Robotino の座標の計算方法を示す.

2.2 多関節アーム型ロボット: Cobotta

Cobotta は, デンソーウエーブ社製のアーム型ロボットである. 工業用ロボットと仕組みは同じであるが, サイズが小さく, 重さも 4.5kg と軽量である. アームの制御方法として, 直接アームを動かしてモータ角を覚えさせるダイレクトティーチングが手軽にできるアームである. RoboCup Logistics League では, 加工機器に材料であるワークを提供したり, 加工されたワークを受け取る際にグリッパーやハンドが必要となる. グリッパーやハンドに対する制約はなく, それを含むロボット全体のサイズに対する制約を満たせば競技に出ることができる.

2019 年の BabyTigers-R は, 専用の把持装置を作成せず, RT-SHOP 社製の CRANE+[7] を利用している.

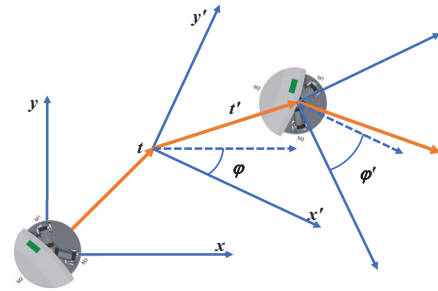


図 2: Robotino の座標の計算方法 (Odometry)

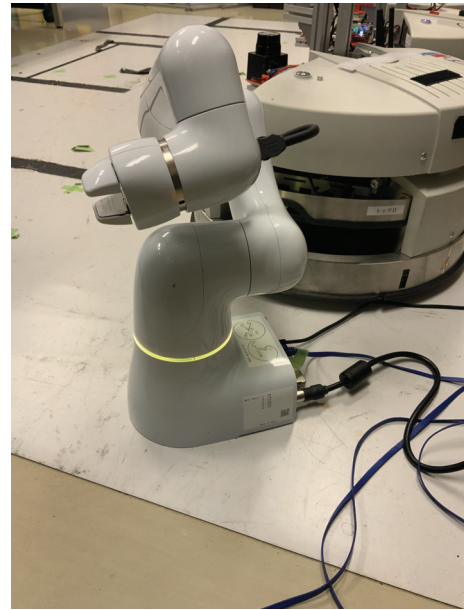


図 3: 6 自由度を持つ小型アーム型ロボット Cobotta

2020 年の大会に対しては, Cobotta の利用を検討している. CRANE+は, モーターが 5 つあり, ハンド部のモーターを除くと自由度は 4 である. 一方, Cobotta はハンド部を除くと自由度は 6 であり, CRANE+よりも複雑な動きが可能となるが, 同一先端座標に対する各モーターの角度の組が一意に定まらず, 制御が複雑になる.

図 3 に Cobotta の全容を示す. また, 表 1 に各モーターの設定可動範囲を示す. ソフトウェアとしての設定可能値と, Soft Axis Limiting (SAL) の値である. ただし, 組み合わせによっては物理的に設定できない角度も含まれる. SAL は, 値を超えるとモーターの電源を OFF にする機能である.

3 先端座標の計算

本章では, グリッパーの制御方法について検討する. 各モーターの値から, グリッパーの先端座標 (x, y) を計

表 1: cobotta の各モータ角のソフトウェアリミットと SAL の角度制限値 [deg]

項目	ソフトウェアリミット		SAL の角度制限値の範囲	
	-方向	+方向	最小値	最大値
J1	-150	150	-160	160
J2	-60	100	-70	110
J3	18	140	8	150
J4	170	170	-180	180
J5	-95	135	-105	145
J6	-170	170	-180	180

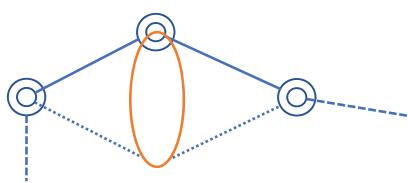


図 4: 3 自由度のアームにおける 2 軸目の可動範囲

算する方法を運動学と言い、一つずつ順番に計算すれば求まる。アームの根元を基準座標 (x_1, y_1) とし、そこから各モータの角度 ϕ_i とモータ間の長さ l_i を基に、次の関節点の座標 (x_{i+1}, y_{i+1}) を計算する ($1 \leq i < n$: n はモータの数)。

一方、アームの先端座標から各モータの角度を計算する方法は逆運動学と言い、自由度が高いアームの場合、複数の解が存在する場合が生じるため、そこから一つを選ぶ仕組みが必要である。図 4 に 3 軸アームの両端を固定した場合を示す。先端座標が、両方のアームを伸ばした状態である場合は、各モータの角度は一意に決まる。しかし、そうでない場合は、2 軸目の関節は、図のように円上のいずれの点でも存在でき、円運動することができる。

ここで、アームを固定している土台が動く場合を考える。このとき、先ほどとは逆の状況になる。つまり、従来は土台が固定されていて先端のアームの軌跡を計算するのに対して、ここでは先端のアームを固定して土台の固定点が軌跡を描くことになる。そのため、計算の対象となるモータの順番を逆に行うことで、アームの先端を空間に固定したまま土台部分の移動が可能になる。ただし、土台の部分はロボットに固定されているため、 z 軸方向を向いている条件が必要となる。



図 5: Cobotta のアームの構成

4 可動範囲の確認と比較

各モータの可動範囲と長さを基に、土台部分を固定したときの先端部分の可動範囲と先端部分を固定したときの土台部分の可動範囲を調べた。図 5 が、cobotta のアームを真上に伸ばしたときの状態で、それぞれの関節部を示す。なお、今回は簡単のため、真横から見た可動範囲を検討する。そのため、土台部分のモータの回転は考慮していない。

先端部分の可動範囲は、図 6 となった。先端部分の可動範囲に関しては、デンソーウェーブの仕様書にも同様の図があり、それとの一致も確認している。一方、先端部分を固定したときの土台部分の可動範囲は、図 7 となった。先端部分稼働の場合と比べ、土台部分の軸が z 軸を向いているという条件があるため、先端部分稼働の場合よりも面積が狭い。また、実際に利用する場合には、土台部分が上下に動くことは困難であるため、この図の y 軸方向の直線上を移動するのみとなる。そのため、可動幅が長い部分に合わせて、ロボットの高さを設定した方が良いことが分かる。例えば、最初の J1 のリンクの位置 (y 軸の値が最小値となる部分) に対して、 z 軸方向に約 8cm 分の幅があるため、4cm ぐらい高い場所に設置するのが望ましいことになる。つまり、アームの上端に対して、真下 48cm ほどの部分に土台が来ようようにロボットに搭載すれば、先端部分を固定した時に動く範囲を広く取ることができると考えられる。

5 まとめ

第四次産業革命 (Industrie 4.0) では、工場のオートメーション化が注目されており、柔軟な生産体系が求められる。本研究では、そのような場における、加工用アーム型ロボットを搭載した移動式ロボットに注目した。通常のアーム型ロボットの制御は、アームの先端座標を指定することで行うが、本研究では、アームの先端座標を空間に固定し、それを載せている移動式

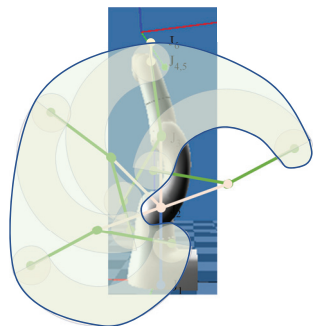


図 6: 土台を固定したときの先端部分の可動範囲

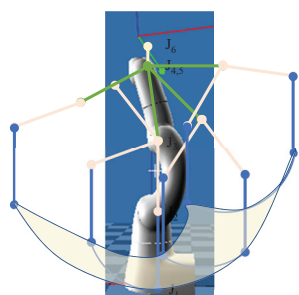


図 7: 先端部分を固定したときの土台の可動範囲

ロボットを動かすことを目指した。特に，従来のアーム型ロボットはアームの先端の可動範囲が性能を決める重要な項目であったが，土台側の可動範囲は考慮されていなかった。本研究にて，土台の可動範囲は，先端の可動範囲と異なり，狭い範囲であることが分かった。今回は土台部分の可動範囲のみを調べたが，範囲内の各点に対する評価は行っていない。今後の課題として，可動範囲内の各点に対して，逆運動学で求まる解の数を元に，可動範囲内の稼働しやすい部分とにくい部分を明確にする必要がある。

謝辞

本研究成果は，科学技術振興機構事業研究成果最適展開支援プログラム A-STEP 機能検証フェーズの支援を受けた。また，人協働ロボットとして株式会社デンソーウェーブより Cobotta をお借りして研究を行った。

参考文献

- [1] The Industrial Robotics Competition Committee, “Industrial Robotics Category - Assembly Challenge - Rules and Regulations 2018”, https://worldrobotsummit.org/download/rulebook-en/rulebook-Assembly_Challenge.pdf (閲覧日，2019年07月31日)
- [2] World Robot Summit, <https://worldrobotsummit.org/> (閲覧日，2019年07月31日)
- [3] World Robot Summit - Industrial assembly challenge, <https://worldrobotsummit.org/wrs2020/challenge/industrial/assembly.html> (閲覧日，2019年07月31日)
- [4] RoboCup Federation, <https://www.robocup.org/> (閲覧日，2019年07月31日)
- [5] RoboCup Logistics League (RCLL), <https://ll.robocup.org/home/> (閲覧日，2019年07月31日)
- [6] Festo, “Robotino View Help”, <http://doc.openrobotino.org/download/RobotinoView/en/index.html>, http://doc.openrobotino.org/download/RobotinoView/en/robotino_motor_dialog.htm, http://doc.openrobotino.org/download/RobotinoView/en/robotino_odometry.htm (閲覧日，2019年07月31日)
- [7] RT ROBOT SHOP, “ロボットアーム CRANE+ V2 完成品”, JANコード：4582221098197, https://www.rt-shop.jp/index.php?main_page=product_info&cPath=1324&products_id=3626 (閲覧日，2019年07月31日)

RoboCup Logistics League における フィールド内の障害物検知に関する一考察

About the detection of the obstacles on the field of RoboCup Logistics League

植村 渉^{1*}

¹ 龍谷大学

¹ Ryukoku University

Abstract: 工場のオートメーション化を想定した自律移動ロボットの競技大会として、RoboCup Logistics League[1]がある。本リーグでは、フィールド上に14台の工作機器が設置され、さらに相手チームを含めて最大6台のロボットとフィールドに対する審判などの数名の人間が動き回っている。この環境下で自由に動き回るためには、工作機器や壁、そして障害物となる他ロボットや人間の検知が重要である。本研究では、工作機器と壁の識別を目的とする。工作機器は立方体の形をしており、長辺側は75cmとなっている。一方、壁は、長さに規定がなく、大会ごとに変わる可能性があるため、単純に長さだけでMPSと壁の識別をすることは望ましくない。すなわち、レーザーレンジファインダーのみでは、識別が難しい。そこで、本研究では、カメラ画像からの物体検知を併用することで、MPSと壁との識別を行う可能性について検討する。

1 はじめに

工場のオートメーション化を想定した自律移動ロボットの競技大会として、RoboCup Logistics League[1]がある。本リーグでは、壁で囲われたフィールド内に、Modular Production System(MPS)と呼ばれる14台の工作機器が設置され、さらに相手チームを含めて最大6台のロボットとフィールドに対する審判などの数名の人間が動き回っている。この環境下で自由に動き回るためには、障害物となる他ロボットや人間の検知が重要である。

また、フィールドの周囲の壁は、すべてを覆っているのではなく、一部は壁のない部分が存在するため、フィールドの外が見える環境となっている。そのため、ロボットの自己位置の推定精度が悪いと、センサー等で検知している対象物がフィールドの中の物か外の物かの判断が難しい。著者が所属している龍谷大学のチーム(BabyTigers - R)は、残念ながら試合開始時の自己位置の推定精度があまり良くなく、自己位置を用いたフィールド内外の判定が難しい状態になっている。この状態でも適切に競技を進めるためには、MPSと壁の認識が重要になる。MPSは立方体の形をしており、長辺側は75cmとなっている。一方、壁は、長さに規定がないため、単純に長さだけではMPSと壁の識別を

することは難しい。すなわち、レーザーレンジファインダーのみでは、識別が難しい。そこで、本研究では、カメラ画像からの物体検知を併用することで、MPSと壁との識別を行う可能性について検討する。

2 RoboCup Logistics League

RoboCup Logistics League (RCLL)では、工場のオートメーション化を想定し、無人搬送ロボット(Automatic Guided Vehicle: AGV)の技術を競う競技である。チームは、最大3台のフェスト社製移動式ロボットRobotinoを用いて、競技に参加する。フィールドは14m×8mの大きさで、周囲は50%以上70%以下の割合で高さ約50cmの壁に囲われている。すなわち、外枠のうち半分弱は壁のない部分があり、外が見えている状態である。図1に2019年世界大会のフィールドを示す。図の右側がチームブースで、左側が競技フィールドである。長方形の形をしているが、上辺側にはほとんど壁がないことが確認できる。

RCLLでは、競技のフェーズとして、準備フェーズ(setup phase)、探索フェーズ(exploration phase)、そして生産フェーズ(production phase)の3つに分かれている。本研究では探索フェーズを主として扱うが、他のフェーズ、特に生産フェーズにおいても有効である。

*連絡先：龍谷大学理工学部電子情報学科
〒520-2194 滋賀県大津市瀬田大江町横谷1-5
E-mail: wataru@rins.ryukoku.ac.jp

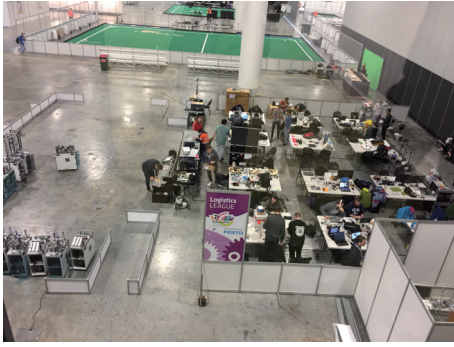


図 1: 2019 年度 RoboCup 世界大会の Logistics League のフィールド

準備フェーズは、各チームがロボットをフィールドに持って行き、ネットワークに接続したりプログラムを起動したりする時間である。同時に、審判の役割を果たすプログラムである RefBox は、フィールド内の加工マシン (modular production system: MPS) のレイアウトをランダムに決定し、人間の審判は、その決定に従って MPS をフィールドに配置する。それら試合の準備が終わると、探索フェーズになる。

探索フェーズは、工場のラインの頻繁な変更を想定しており、加工マシンである MPS がフィールド上にランダムに配置される。現場において、ラインの変更のたびに、加工マシンの位置情報を手動で作成するのは手間であるため、ロボットに自動で調べさせるのがこのフェーズである。本競技では、5 種類の加工マシンを使い、そのマシンの種類も本来はロボットが自動で識別すべきであるが、現段階ではその技術はテクニカルチャレンジとして扱い、MPS の側面に貼った二次元バーコードである ALVAR のタグから種別を読み取る。そして、MPS の置いてある場所 (ゾーン名)、角度、そして種別を RefBox に報告することで得点となる。

生産フェーズは、審判プログラムの RefBox からいくつかの注文が提示され、作るべき製品の種類 (オプション等の有無)、配送する時間、そして必要とする個数が示される。各チームのロボットは、その情報を基に、各ロボットの動きを決定し、ロボットは製品の素材を扱うために MPS 間を移動する。なお、生産フェーズは探索フェーズに続いて実施されるが、探索フェーズの成果とは分離して扱うため、生産フェーズに入ったと同時に RefBox から加工マシンの位置情報等は告知される。そのため、生産フェーズにおいては、フィールド情報は既知となり、最短経路の算出が可能となる。しかし、フィールド上には、相手チームのロボットや審判、そして補助作業によってチームメンバーが入る場合があり、障害物検知と経路再設定の機能は必要である。

探索フェーズでは、フィールド上の MPS を見つける

必要がある。MPS の形は、幅 70cm、奥行き 35cm、高さ約 90cm の直方体であり、レーザーレンジファインダーで 70cm の直線を見つることができれば、MPS の長辺であることがわかる。なお、短辺側はアクリル板の構造となっており、MPS の内部の機器 (バッテリーやエアコンプレッサーなど) が見えるため、短辺をレーザーレンジファインダーで検知するのは難しい。ここで、フィールドの外周の一部に壁のない部分が存在することに気をつけなければならない。場合によっては、70cm 前後の長さの壁が存在することもあり、それを MPS と認識する場合もある。また、フィールドの外が見える環境となっていることにも気をつけなければならない。ロボットの自己位置の推定精度が悪いと、センサー等で検知している対象物がフィールドの中の物か外の物かの判断が難しく、フィールドの外に置いている MPS に反応する場合もある。

著者が所属している龍谷大学のチーム (BabyTigers - R) は、残念ながら試合開始時の自己位置の推定精度があまり良くなく、自己位置を用いたフィールド内外の判定が難しい状態になっている。この状態でも適切に競技を進めるためには、MPS と壁の識別が重要になる。レーザーレンジファインダー単体での識別が困難であるため、本研究では、カメラ画像による物体検知を併用することで、MPS と壁との識別を行う可能性について検討する。

3 レーザーレンジファインダーによる MPS の識別

2019 年度の BabyTigers-R では、FESTO 社製のプログラミング言語である RobView を用いてロボットを制御した。RobView は、DIN EN 61131 に基づくブロック形プログラミング言語である。HOKUYO 社製のレーザーレンジファインダーである URG-04LX-UG01 に対応しており、レーザーのファンクションブロックがある。このブロックには 2 つの引数があり、検知する範囲 (角度) と、検知開始角度を指定する。その指定した範囲内における最近接点までの距離が出力される (図 2)。

探索フェーズでは、フィールド内の MPS を見つける必要があるが、もし、MPS がこのレーザーレンジファインダーの範囲内にある場合、それに対する一番近い点は、MPS の角の場合と、面の場合が考えられる。ここで、その見つけた最近接点から左右を確認すると、どちらも、その確認点と最近接点は直線になるはずである。そして、さらに左右に進めたときに、その直線と、左右にある点との距離が 0 (実際には誤差を考慮して $\pm\alpha$ の幅を持たせる) である限り、最近接点から直線が続いていると考えられる。また、距離が一定以上離

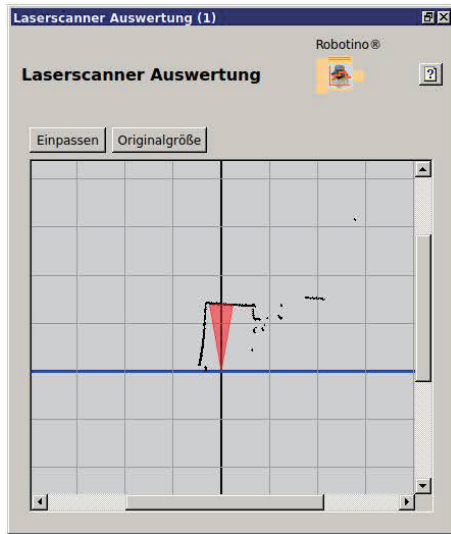


図 2: robView における LRF のファンクションブロック。

れた場合、直線から点であることがわかり、その一つ前の点と最近接点との間に面が存在することがわかる。MPS の角の場合は、最近接点の左側と右側の面が直交しているはずであり、面が最近接点の場合は、左側も右側も同一直線上に存在するはずである。実際には、MPS の短辺側はアクリルの板を使っているため、MPS の内部が見えてしまい、面を構成していない場合がある。その場合は、長辺側の長さを確認し、MPS の長辺 (70cm) と近いかどうかで MPS を判断する。フィールド上には、類似した長さの壁も存在するため、できる限り壁が見えないようにフィールドの内側を向いて走行することで、誤検知を避けている。

4 画像処理による MPS の識別

本研究では、壁と MPS を識別するために、GPU として Jetson nano を用いる。また、リアルタイムオブジェクト検出で代表的な yolo[2] の適用を目指す。レーザーレンジファインダーによる距離情報や、MPS 検知と組み合わせるためには、レーザーレンジファインダーの検知範囲とカメラの撮影範囲とを合わせる必要がある。そこで、今回はそれらを同時に扱えるステレオカメラを利用することも検討した。ステレオカメラの実行結果を図 3 に示す。

5 まとめ

RoboCup Logistics League において、フィールド上の機器と壁との識別方法について検討した。従来のレーザーレンジファインダーによる識別では、誤検知の可

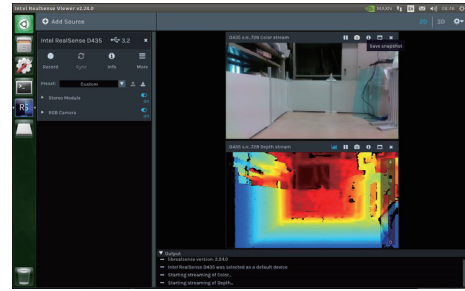


図 3: Jetson Nano 上で realSense で取得した距離情報

能性があったため、画像処理による識別を併用した。次回以降の大会にて、性能が向上することを期待したい。

謝辞

本研究成果は、科学技術振興機構事業研究成果最適展開支援プログラム A-STEP 機能検証フェーズの支援を受けた。

参考文献

- [1] RoboCup Logistics League (RCLL), <https://ll.robocup.org/home/> (閲覧日, 2019 年 07 月 31 日)
- [2] <https://pjreddie.com/darknet/yolo/> (閲覧日, 2019 年 07 月 31 日)

© 2019 Special Interest Group on AI Challenges
Japanese Society for Artificial Intelligence
一般社団法人 人工知能学会 AI チャレンジ研究会

〒162 東京都新宿区津久戸町 4-7 OS ビル 402 号室 03-5261-3401 Fax: 03-5261-3402

(本研究会についてのお問い合わせは下記をお願いします。)

AI チャレンジ研究会

主査 / 担当幹事

光永 法明

大阪教育大学 教員養成課程 技術教育講座

Executive Committee Chair

Noriaki Mitsunaga

Department of Technology Education,
Osaka Kyoiku University

主幹事

鈴木 麗璽

名古屋大学 大学院情報学研究科 複雑系科学専攻

Secretary

Reiji Suzuki

Department of Complex Systems Science,
Graduate School of Informatics,
Nagoya University

担当幹事

植村 涉

龍谷大学 理工学部 電子情報学科

Wataru Uemura

Department of Electronics and Informat-
ics, Faculty of Science and Technology,
Ryukoku University

幹事

干場 功太郎

神奈川大学 工学部 電気電子情報工学科

Kotaro Hoshiba

Department of Electrical, Electronics and
Information Engineering, Faculty of Engi-
neering, Kanagawa University

中臺 一博

(株) ホンダ・リサーチ・インスティテュート・

ジャパン / 東京工業大学 工学院

システム制御系

Kazuhiro Nakadai

Honda Research Institute Japan Co., Ltd.
/ Department of Systems and Control
Engineering, School of Engineering,
Tokyo Institute of Technology

SIG-AI-Challenges web page; <http://www.osaka-kyoiku.ac.jp/~challeng/>