

AI チャレンジ研究会(第48回)

Proceedings of the 48th Meeting of Special Interest Group on AI Challenges

CONTENTS

- 食品単語のベクトル空間の構築と評価 (第2報) 1
矢野達也, 林豊洋, 大橋健
九州工業大学
- RoboCup サッカーにおける敵位置情報に基づいた敵フォーメーション同定 7
福島 卓弥†, 中島 智晴†, 秋山 英久‡
†大阪府立大学, ‡福岡大学
- Improvement of Multiple Robots' Self-localization by Using Perspective Positional Information 13
會澤 要, 鈴木 拓央, 小林 邦和
愛知県立大学

日時:2017年5月6日

場所: 愛知県立大学 長久手キャンパス S棟 1F S101

Aichi Prefectural University Nagakute Campus, May 6th, 2017

一般社団法人 人工知能学会

Japanese Society for Artificial Intelligence

食品単語のベクトル空間の構築とその評価(第2報)

Construction of a dense vector space for food category words and its evaluation (2nd report)

○矢野 達也 林 豊洋 大橋 健
Tatsuya Yano Toyohiro Hayashi Takeshi Ohashi
九州工業大学
Kyushu Institute of Technology
n238070t@mail.kyutech.jp

Abstract

自然言語処理分野の1つのアプローチとして word2vec が注目されている。word2vec は、ニューラルネットワークの学習に基づき、与えられた文章から単語のベクトル表現を生成する。本研究では、類似する食品単語に対して密なベクトル空間の構築とその評価を目的とした。word2vec の入力コーパスは、各食品のカテゴリや原材料情報を web 上で取得し、ジェネレータにより作成した。食品のカテゴリ、原材料を学習させたコーパス1、原材料のみを学習させたコーパス2を作成し、それぞれのベクトル空間を構築した。構築したベクトル空間は k-means 法により 30 のクラスターに分類し、評価を行った。具体的には、各クラスター間のデータの分散値、含有率の比較、各クラスター内でのユークリッド距離、コサイン類似度による類似語検索、PCA 法による低次元圧縮上での可視化により評価を行う。結果、どちらのコーパスも単語の類似性に対して優れたベクトル空間の構築が確認できた。コーパス1で構築したベクトル空間では、カテゴリを学習させているため、菓子、飲料の明確な区別の中で、類似度分類が可能であった。コーパス2では、原材料のみ学習させたため、カテゴリの区別なく類似語の分類ができた。自作の文章をベクトル空間構築のコーパスとして使用することの有用性が検証できた。

1. はじめに

近年、家庭用サービスロボットの開発が活発である。国際的ロボット競技大会である RoboCup@Home では、家庭環境において、人間とロボットのインタラクションを評価するテストを行っている。ロボットには、柔軟な言語処理の能力が求められる。例えば、ロボットに「コーヒーを持って来て」と指示し、そこにコーヒーが無かった場合、解決策の1つとしてコーヒーの代替品を持ってこることが挙げられる。こういった際、類似語を機械的に取得することができれば、より柔軟な対応が期待できる。本研究では、word2vec を用いて、自作した食品単語の文章データからベクトル空間を構築し、この課題への対応を試みる。

2. 関連研究

word2vec は Tomas Mikolov らによって提唱された、単語をベクトル表現にする手法である[1]。入力として文章を与え、文章中の単語の共起関係にもとづき、予め設定していた次元数のベクトル(分散表現)を学習する。通常、学習コーパスには数十万から数百万の語彙数を持つテキストデータを使用する。文章データに含まれる語彙数が多いほど幅広い表現が可能なベクトル空間を構築できる反面、特定の分野に関しては、疎なベクトル空間になる場合が考えられる。根本らは、雑談ができる知識表現の獲得を目指して、青空文庫の文学作品を著者別に学習し、雑談対話システムへの検討を行っている[2]。そこでは、「人間」の類似単語をみると、夏目漱石の作品では「価値」、「学問」、太宰治の作品データでは、「事態」、「思想」といった単語であった。著者の思想や性格にもとづき広義の類似語結果が取得できていることが確認できる。このように、word2vec では構築したいベクトル空間に合わせた入力コーパスを用意する必要がある。

本研究では、自作の文章ジェネレータを用いて文章データを作成することで、類似食品単語の検索に適したベクトル空間の構築を行う。また、構築したベクトル空間にクラスタリング、主成分分析を施しその評価を行う。本研究の第1報では、菓子、飲料それぞれベクトル空間の構築とその評価を行った[3]。第2報では、菓子、飲料を1つのベクトル空間に集約し、その評価を行う。

3. 食品単語のベクトル空間の構築

3.1 提案手法の概要

本研究では、家庭用サービスロボットでの運用に焦点を絞り、構築するベクトル空間は一般家庭を想定した菓子、飲料に限定する。文章は Web から取得した食品情報をもとに、自作の文章ジェネレータにより生成する。生成された文章を word2vec のコーパスとして用いて、食品単語のベクトル空間を構築する。

3.2 データ収集

まず、文章生成に必要なデータを Web から取得

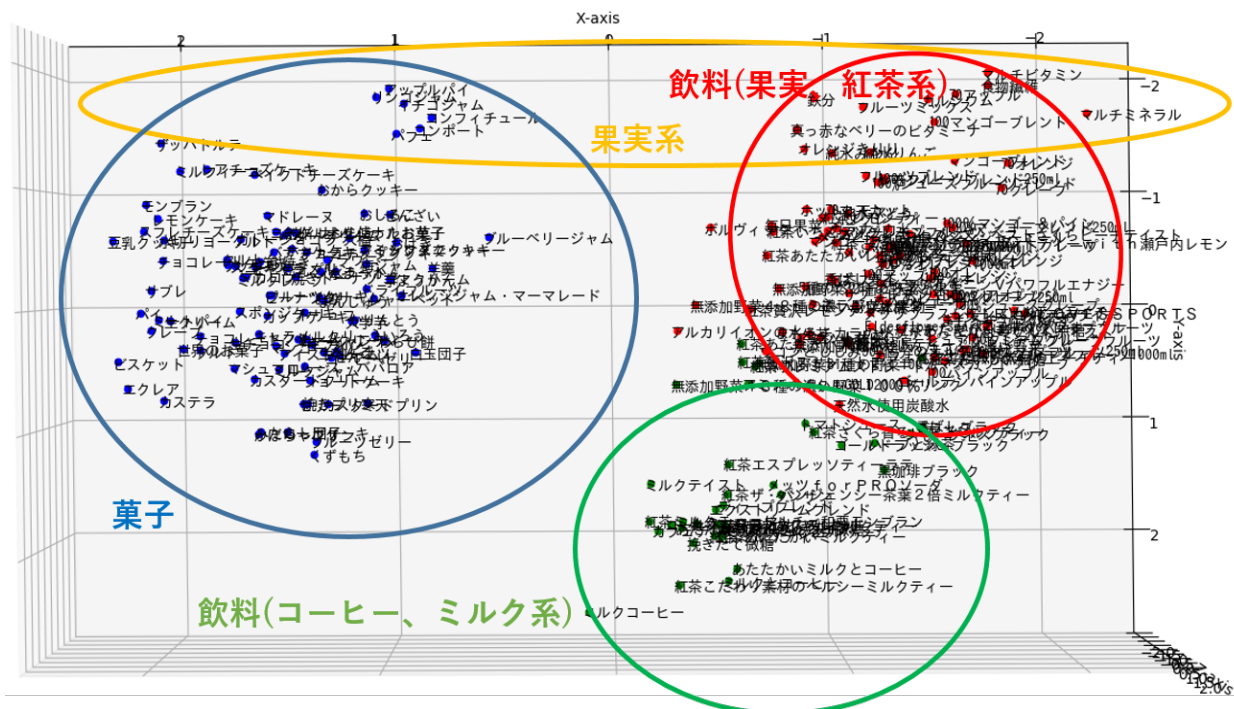


図 1. 食品単語ベクトル空間の可視化 (コーパス 1)

する。菓子は楽天株式会社が運用する楽天レシピ(注1)のレシピページから 100 品、飲料はキリン株式会社の製品一覧ページ(注2)から 140 品を抽出し、それぞれ名前、カテゴリ (飲料または菓子)、原材料の情報を取得する。取得例として、チョコチップクッキーではカテゴリ (菓子)、原材料 (薄力粉、バター、チョコ) となる。

3.3 文章生成

取得した食品情報を文章ジェネレータのテンプレート文に当てはめ、1 食品ごとに 1200 文の短文を生成する。チョコチップクッキーでの生成例を以下に示す。

1. [チョコチップクッキー] は [菓子]
2. [チョコチップクッキー] の 原料 は [バター]

第 1 報では、菓子、飲料の文章を別々にコーパスとして使用し、それぞれ独立したベクトル空間を作成した。第 2 報では菓子、飲料の 2 つを統合した文章をコーパスとして使用し、1 つのベクトル空間を構築する。テンプレート 1,2 (カテゴリ、原材料) を用いて生成するコーパス 1 と、テンプレート 2 (原材料) のみで生成するコーパス 2 を、word2vec の入力コーパスとして、ベクトル空間を構築する。

3.4 Skip-gram による学習

word2vec には学習モデルとして Skip-gram と Continuous Bag-of-Words があり、本研究では Skip-gram を採用した。Skip-gram は、文章中の単語を入力とし、その前後の単語を推定する学習を行う。前後何単語を関係性のある単語とするかは window パラメータで設定する。出力層では、ソフトマックス関数を用いて window パラメータで設定した前後の単語の出現確率を出力する。中間層では、出力層の周辺単語の出現確率のエラー率が最小となるように学習を行う。word2vec では、こ

の中間層を単語の特徴ベクトルとし、ベクトル空間として使用する。今回は、文章ジェネレータで生成した文章を word2vec のコーパスとして使用する。window パラメータは初期値の 5 として学習を行う。また、word2vec は初期値として 100 次元の単語ベクトルを生成するが、今回は菓子の語彙数が 282 個、飲料の語彙数が 384 個と少数であるため、生成する単語ベクトルの次元数は 30 とした。

4. ベクトル空間の可視化

word2vec により構築した 30 次元の食品単語ベクトル空間に主成分分析を施し、3 次元上に可視化を行う。また、分布を確認するためクラスタ数を 3 に設定し、k-means 法を用いたクラスタリングを行う。コーパス 1 で生成したベクトル空間を図 1、コーパス 2 で生成したベクトル空間を図 2 に示す。

図 1 を見ると、コーパス 1 では、テンプレート 1 を用いて菓子か飲料かを明示的に学習させているため、菓子が左側に、飲料が右側に大きく 2 つに分離している。さらに飲料の中でも、野菜、果物、紅茶系飲料は右上に、ミルク、コーヒー系飲料が左下の 2 つに分かれて分布していることが確認できる。図 1 の上部では、原材料として果物が含まれる菓子、飲料が同じ軸に並んでいる。これらの結果より、コーパス 1 で構成したベクトル空間では、菓子か飲料のカテゴリ軸と原材料の軸によるベクトル空間が構築できていることが確認できる。

図 2 は、原材料のみを学習させたベクトル空間である。カテゴリ分類を陽に学習させていないため、図 1 のように分布が 2 極化することはないが、

(注 1) : <https://recipe.rakuten.co.jp/>

(注 2) : <http://www.kirin.co.jp/products/list/nutrition/softdrink/>

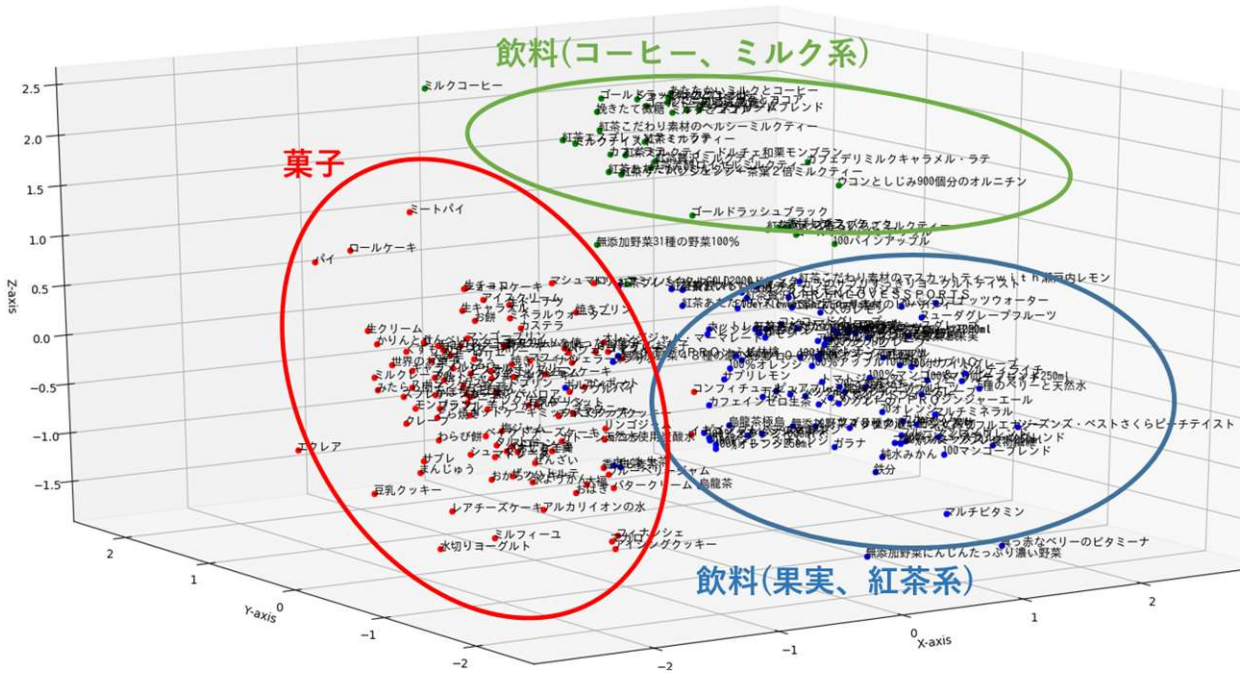


図2. 食品単語ベクトル空間の可視化 (コーパス 2)

原材料の観点から、菓子と飲料が分かれていることが確認できる。左側に菓子、右側に飲料が分離し、飲料の中でも原材料の観点から2つに分離している。また水系の飲料は、菓子の飲料の原材料として菓子、飲料どちらにも含まれるため中央付近に分布している。これらの結果より、コーパス2で構成したベクトル空間では、原材料のみでベクトル空間を構築していることから菓子、飲料の区別を考慮しない分類ができていたことが確認できた。

5. ベクトル空間の評価

構築したベクトル空間をクラスタリングした後、各クラスター内で類似語検索を行いベクトル空間の評価をする。クラスタリングには k-means を用いる。クラスター数は菓子(14分類)、飲料(7分類)のカテゴリ数を考慮し30とする。

5.1 食品単語ベクトル空間のクラスタリング

ベクトル空間にクラスター数 2~30 で k-means を適用し、以下の条件でその変化を確認する。

- ①クラスタリングの結果、和菓子が最も多く含まれるクラスターを和菓子のクラスター、果実飲料が最も多く含まれるクラスターを果物飲料のクラスターとする。
- ②2つのクラスター間の級間分散、級内分散を算出、加えて各クラスターに含まれる和菓子、果実飲料のデータ数から含有率を算出し各クラスターでの変化を評価の指標とする。級間分散/級内分散、クラスター内の含有率結果はクラスター数ごとに20回のクラスタリングを施しその平均とする。

コーパス1ベクトル空間でのクラスタリング結果を図3に示す。また、実際の分布を2次元上に可視化した図を図4,5,6に示す。それぞれの図には果実飲料、和菓子が多く含まれる上位3つのクラスターの果実飲料、和菓子を表示している。果実飲料が含まれる上位3クラスターには赤系の3色、和菓子系が含まれる上位3クラスターには青系の3色、それ以外は灰色でプロットしている。同様にコーパス2ベクトル空間でのクラスタリング結果を図7に、2次元上に可視化した図を図8,9,10に示す。

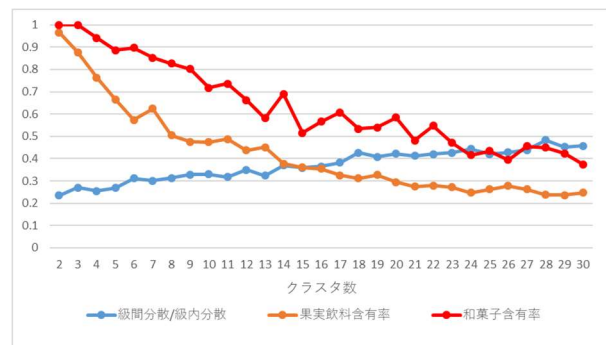


図3. コーパス1ベクトル空間クラスタリング

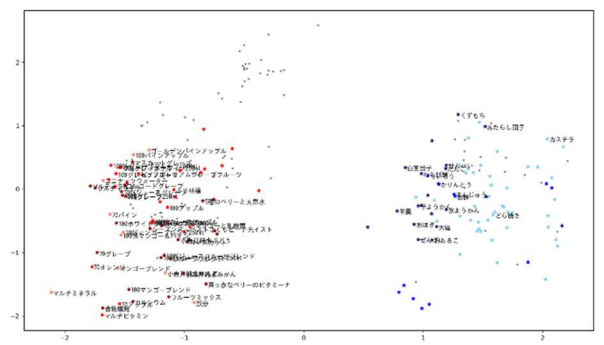


図4. コーパス1ベクトル空間のクラスタリング可視化(k=10)

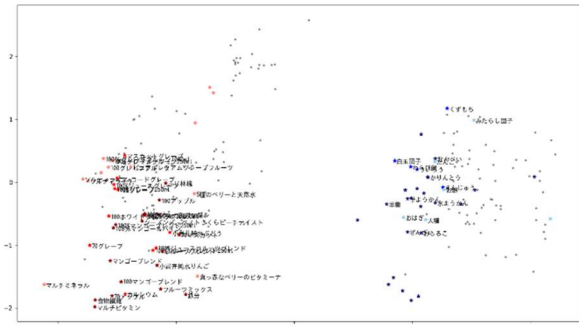


図 5. コーパス 1 ベクトル空間のクラスタリング可視化(k=20)

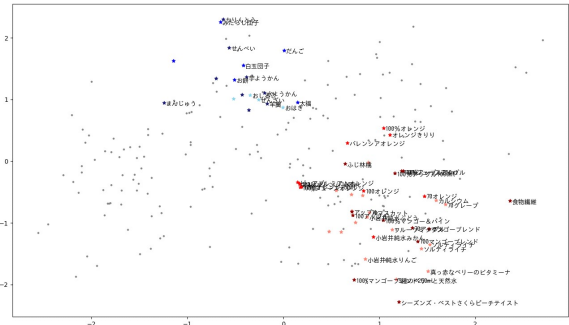


図 10. コーパス 2 ベクトル空間のクラスタリング可視化(k=10)

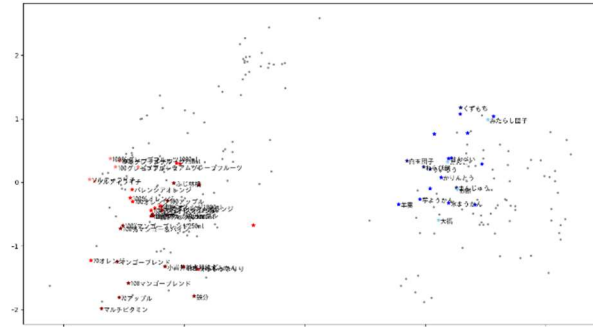


図 6. コーパス 1 ベクトル空間のクラスタリング可視化(k=30)

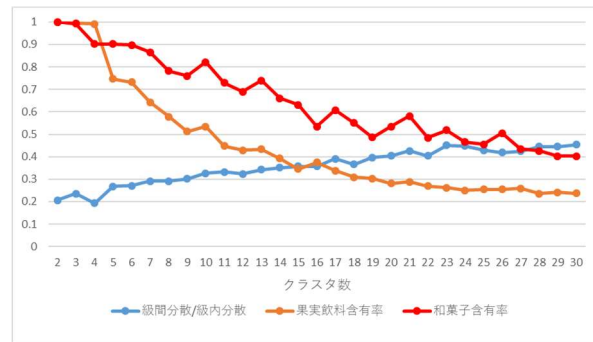


図 7. コーパス 2 ベクトル空間クラスタリング

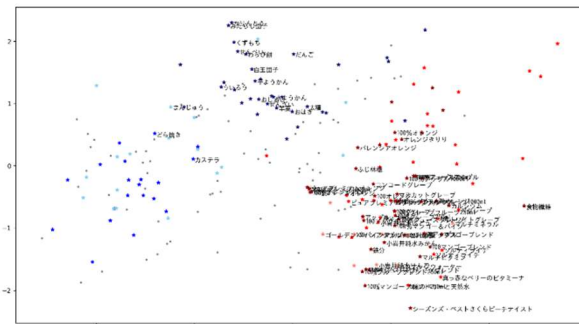


図 8. コーパス 2 ベクトル空間のクラスタリング可視化(k=10)

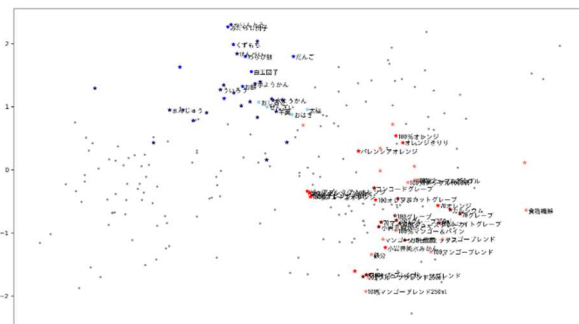


図 9. コーパス 2 ベクトル空間のクラスタリング可視化(k=20)

図 3、図 7 の結果を見ると、クラスタ数を増やすごとに果実飲料クラスタの果実飲料含有率、和菓子クラスタの和菓子含有率が減少し、級間分散/級内分散が増加している。クラスタ数を増やすことで、より細かい基準で分類されてクラスタが細分化されるため妥当な結果だといえる。図 3 ではカテゴリを学習させているため、クラスタ数が少ない段階において分類度が高いが、クラスタ数を増やすと図 4 の各値に大きな差異はなかった。2次元プロット図では、クラスタ数の増加に伴いデータの細分化が視覚的に確認できる。また、コーパス 1 は菓子、飲料を区別しているため、コーパス 2 に比べてデータの分散が小さく、クラスタがコンパクトにまとまっている。

5.2 クラスタ内類似語検索

構築したベクトル空間に $k=30$ で k -means を施し、各クラスタ内での類似語検索を行う。類似度の評価にはユークリッド距離(E-dist)とコサイン類似度(C-dist)を用いる。表 1,2 にコーパス 1,2、ベクトル空間における「羊羹」が属するクラスタ内での「羊羹」の類似語検索結果を示す。同様に、表 3,4 ではコーパス 1,2 ベクトル空間における「トロピカーナ 100%ジュースグレープ」の類似語検索結果を示す。表は、コサイン類似度を軸に類似度が高い順に並べている。ユークリッド距離と相違がある順位には赤で記している。表 4 を除くこれらの結果では、両距離計において、類似度順に違いはなかった。

食品(コーパス 1)では、羊羹の類似度が高いものとして、水羊羹や寒天といった菓子以外に、梅ジャムや、シャーベットといった菓子も取得していることが確認できる。上位 5 件以外にも同じクラスタには和菓子やプリン菓子が属していた。食品(コーパス 2)では、羊羹と同じクラスタには 4 つのみの単語が属し、原材料から類似度が高いものを取得できている。

飲料(コーパス 1)、100%ジュースグレープの類似度が高いものとして、同じグレープ系飲料が取得できている。上位 5 件以外には、ブレンド系の果実飲料や、果実飲料以外にもトマトジュースなどが属していた。飲料(コーパス 2)では、類似度の上位 5 件以外にもグレープ系の果実飲料が同じクラスタに属していた。

表 1. 「羊羹」類似語結果 (コーパス 1)

単語名	C-dist		E-dist	
水ようかん	1	0.985	1	1.308
寒天	2	0.893	2	1.974
梅ジャム	3	0.849	3	2.314
シャーベット	4	0.779	4	2.749
コーヒーゼリー	5	0.758	5	3.144

(C-dist: コサイン類似度, E-dist: ユークリッド距離, 距離値の左側にある番号は類似度順を表す, 以下同)

表 2. 「羊羹」類似語結果 (コーパス 2)

単語名	C-dist		E-dist	
水ようかん	1	0.960	1	1.268
寒天	2	0.901	2	1.905
芋羊羹	3	0.895	3	2.040
コーヒーゼリー	4	0.787	4	3.013

表 3. 「100%ジュースグレープ」類似語結果 (コーパス 1)

単語名	C-dist		E-dist	
トロピカル 100 グレープ	1	0.996	1	0.317
ハイパー100 グレープ	2	0.995	2	0.347
ホワイトグレープ	3	0.914	3	1.564
コンコードグレープ	4	0.866	4	2.428
マスカットグレープ	5	0.861	5	2.436

表 4. 「100%ジュースグレープ」類似語結果 (コーパス 2)

単語名	C-dist		E-dist	
トロピカーナ 100 グレープ	1	0.996	1	0.306
ハイパー100 グレープ	2	0.995	2	0.346
ホワイトグレープ	3	0.914	3	1.536
コンコードグレープ	4	0.859	6	2.535
マスカットグレープ	5	0.898	7	2.584

5.3 食品単語ベクトル演算

word2vec ではそれぞれの単語を分散表現としてベクトル化しているため、単語間でのベクトル演算が可能となる。ベクトル演算の例としては、「King」 - 「Man」 + 「Woman」 = 「Queen」のようになる。単語のベクトル演算から、ベクトル空間の評価を行う。ここでは、3つの単語を入力し、コサイン類似度、ユークリッド距離により4単語目を推測する比較演算を行う。比較演算結果を表 5, 6 に示す。表 5 は、「ファイアブラック」「ファイアカフェラテ」「午後の紅茶おいしい無糖」を入力した結果である。表 6 は「生クリーム」「ロールケーキ」「生チョコ」を入力した結果である。比較演算にはコーパス 1 で構成したベクトル空間を用いた。前節と同様に、コサイン距離を軸に類似度が高い順に並べている、

表 5 では、飲料単語での比較演算を行っている。無糖のコーヒーと加糖のコーヒー、無糖の紅茶を入力したため出力には加糖の紅茶が期待できる。コサイン類似度の結果を見ると、ミルクティー、エスプレッソティーといった加糖の紅茶が取得できた。同様にユークリッド距離の結果においても、加糖の紅茶が取得できた。また、両距離計を用い

た結果、上位 5 単語中、4 単語は同じ単語が取得できた。

表 6 では、菓子単語での比較演算を行っている。こちらは材料とそれから作れる菓子の関係を入力としたため、出力としてチョコ系の菓子が期待できる。結果を見ると生チョコを使った菓子が取得できた。また、両距離計での結果上位 5 単語中、3 単語は同じ単語が取得できた。前節のクラスター内類似語検索結果と合わせて、30 次元ベクトル空間において、ユークリッド距離系の有用性が確認できた。

表 5, 6 のように、カテゴリ内でのベクトル演算は期待した結果を得ることができたが、菓子、飲料単語を跨いだベクトル演算を行うと期待した結果は得られなかった。これには、菓子、飲料に使用させる原材料の表記が異なるためである。例えばレモンという材料は、飲料ではレモンで表記されるが、菓子ではレモン果汁で表記されることが多い。カテゴリ間でのベクトル演算を可能にするには、お互いの表記のゆれをなくし、統一する必要があると考えられる。

「ファイアブラック」: 「ファイアカフェラテ」
= 「午後の紅茶おいしい無糖」 : ?

表 5. ベクトル演算結果 1

単語名	C-dist		E-dist	
午後の紅茶 ミルクティー	1	0.739	1	3.797
午後の紅茶 茶葉 2 倍ミルクティー	2	0.724	8	4.240
午後の紅茶 エスプレッソティー	3	0.719	4	4.003
午後の紅茶 あたたかいミルクティー	4	0.718	3	3.985
午後の紅茶 ストレートティー	5	0.715	2	3.893

「生クリーム」: 「ロールケーキ」 = 「生チョコ」 : ?

表 6. ベクトル演算結果 2

単語名	C-dist		E-dist	
トリュフ	1	0.684	1	5.515
ガトーショコラ	2	0.673	5	5.581
ジェラート	3	0.661	20	5.951
カップケーキ	4	0.659	6	5.597
チョコレートケーキ	5	0.658	2	5.524

次に、表 5 における比較演算の位置関係について 2 次元上に可視化した図を図 11 に示す。名前は入力した 3 単語と類似度が最も高いものを表示しプロットには、入力単語と、類似度上位 5 単語を赤で表示した。30 次元のデータを 2 次元まで圧縮したため、綺麗な平行線とはならないが、コーヒーとカフェオレの関係性が視覚的に確認できる。

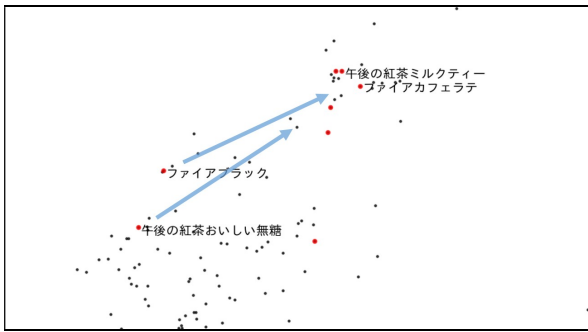


図 11. 比較演算 2次元可視化

6. まとめ

本研究では、食品単語の類似性に関して密なベクトル空間の構築を目的とし、ジェネレータから文章を生成し、word2vecを用いたベクトル空間の構築を行った。また、構築した2つのベクトル空間にクラスタリング、主成分分析を用いてその評価を行った。カテゴリを明示的に学習させたコーパス1では、菓子、飲料が大きく分離するため各カテゴリ内での類似語分類となり、コーパス2で構築したベクトル空間では、カテゴリの分類なく食品単語全体を原材料から分類できた。また、様々なベクトル空間評価方法により、ジェネレータで生成する文章、学習させる内容を変えることで期待するベクトル空間の構築が可能であると分かった。

今後の課題として、菓子、飲料間でもベクトル演算が可能となるようなベクトル空間の構築が挙げられる。

参考文献

- [1]Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean
"Efficient Estimation of Word Representations in Vector Space," ICLR, 12pages, (2013)
- [2]根本太晴,岡田浩之「word2vecによる雑談対話システムの検討」,第2回 iHR 研究会,2頁,(2015)
- [3]矢野達也,林 豊洋,大橋 健「食品単語ベクトル空間とその評価」,第6回 iHR 研究会,(2017)

RoboCup サッカーにおける敵位置情報に基づいた敵フォーメーション同定

Opponent's Formation Identification based on Position Information for RoboCup Soccer

福島 卓弥 †, 中島 智晴 †, 秋山 英久 ‡

Takuya FUKUSHIMA†, Tomoharu NAKASHIMA†, Hidehisa AKIYAMA‡

大阪府立大学 †, 福岡大学 ‡

Osaka Prefecture University†, Fukuoka University‡

takuya.fukushima@edu.osakafu-u.ac.jp

tomoharu.nakashima@cs.osakafu-u.ac.jp

akym@fukuoka-u.ac.jp

Abstract

The aim of this paper is to propose a method for identifying the opponent formation type in an online manner during a game. To do so, opponent teams were clustered according to the position of their players. Each cluster is investigated to determine the difficulty for our team to defeat such a strategy. Then, an identification model is used online to determine if the opponent team adopts such a strategy or not. Furthermore, we also investigate how quickly the opponent formation can be identified. Through a series of computational experiments, it is shown that the model can identify opponent formation type quickly and accurately. Therefore, we show the effectiveness of the identification model to switch our strategy.

1 はじめに

ロボット工学と人工知能の領域横断型研究プロジェクトとしてRoboCup [1] が知られている。RoboCup には様々なリーグが存在しており、それぞれのリーグにおいて活発に研究、開発が行われている。その中の一つであるRoboCup サッカーリーグでは、ただ単に勝利するだけでなく、賢く安定して勝利することが望まれている。ランダムや思いつきで作られた戦術を使って勝利するよりも、緻密なデータ分析や機械学習によるモデル化を活用して勝利につながる戦術を生成することがRoboCup の理念と合致する。パスやドリブル等の行動を洗練する [2, 3], 相手の行動を予測して意思決定を行う [4], 相手チームに合わせた戦術をとる [5, 6], といった研究が様々な観点から進められている [7].

現在、サッカーシミュレーション 2D リーグでは、毎年各チームが変化を加えることにより、多種多様な戦術が存在している。戦術には相性があるため、1つの戦術で全てのチームに勝つことは困難である。そのため、安定した勝利には、相手チームに対して相性の良い戦術をとることが有効である。有効な戦術を選択するためには、自チームが複数の戦術を用意し、相手チームの戦術を見極め、それに応じて自チームの戦術を切り替える必要がある。相手チームの戦術を素早く見極め戦術を切り替えることで、試合を有利に進めることが可能になる。

本論文では、相手チームの戦術を見極める手法に焦点をあてる。本研究の目的は、自チームの戦術が相手チームと相性が悪いかどうかを判断することである。相手チームとの相性を調べるためには、相手チームを分析しなければならない。相手チームの戦術や行動をモデル化し識別する研究は進められている [8] が、オフラインでの識別に用いられている。相手チームの戦術は試合が始まるまで分からないため、試合中にオンラインで判断することが求められる。そこで本論文では、オンラインで戦術を切り替えるために、戦術はプレイヤーのポジショニングによって表現されるものと仮定し、敵位置情報に着目する。まず、敵位置情報に基づいて相手チームのクラスタリングを行い、自チームが不得意とするフォーメーションを調査する。次に、相手チームが不得意なフォーメーションであるかどうかをオンラインで判断するため、識別モデルを構築する。識別モデルの実際の試合における有効性を示すため、経過時間と識別率の関係を検証する。

2 RoboCup

2.1 RoboCup サッカーシミュレーション 2D リーグ

本研究では、RoboCup サッカーシミュレーション 2D リーグを対象とする。シミュレーションリーグはRoboCup 創設当初から存在する最も古いリーグの1つである。2D リー

グでは実機を使用せず、コンピュータ内に用意された二次元平面を仮想サッカーフィールドとし、円形のエージェントをプレイヤーとして競技を行う。また、プレイヤーやボールの位置と速度は全て二次元ベクトルとして表される。2Dリーグでは、kick, dash, turn などの抽象化されたコマンドを基本行動とする。図 1 に 2D リーグの試合の様子を示す。試合は前後半 3000 サイクルずつ合計 6000 サイクルからなる。1 サイクルは 0.1 秒で離散化されている。

プレイヤーやコーチはそれぞれ独立したエージェントとしてプログラムされている。各プレイヤーには実際の人間と同様に視野が設定されており、自身の視野内で認識できた情報に基づき、視覚情報が形成される。また、他のプレイヤーやコーチがメッセージとして発している情報を聴覚情報として使用できる。これらの視覚情報や聴覚情報からフィールド情報を形成し、これに基づきドリブルやパスなどの意思決定を下す。しかし、視覚情報にはノイズが含まれ、正確な情報を獲得することができない。聴覚情報についても確実に受信できるわけではなく、コーチのメッセージは通常プレイ時には到達までに遅延も発生する。そのため、プレイヤーはフィールド上の正確な情報を保持することはできない。一方で、コーチはフィールド上のすべての情報をノイズなしで取得することができるため、正確なフィールド情報を保持することができる。通常プレイ時におけるプレイヤーへの意思伝達には遅延が発生し、回数も制限されているが、ハーフタイム時においては、プレイヤーに即時に情報を伝達することが可能である。

また、試合毎に異なる能力を持つプレイヤーのセットが与えられ、各チームがポジションの割当を行う。このようにランダムな要素が多く、プレイヤーが獲得する情報や物体の移動にノイズがかかることから、同一の対戦相手であっても、試合毎に結果や内容が異なる。

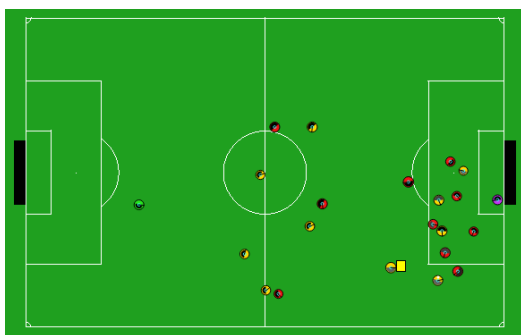


図 1: Soccer simulation 2D league

2.2 試合ログ

試合ログは、RoboCup サッカーシミュレーション 2D リーグにおいて、試合終了後にサーバから出力されるファイルである。試合ログには、各プレイヤーの最高速度やキックできる範囲等といったパラメータ、ゲームの状態、サイク

```
(player_type (id 17)(player_speed_max 1.05)(stamina_inc_max 51.6181)
(player_decay 0.459447)(inertia_moment 6.48617)(dash_power_rate 0.00489698)
(player_size 0.3)(kickable_margin 0.643989)(kick_rand 0.0439887)(extra_stamina
85.5322)(effort_max 0.857871)(effort_min 0.457871)(kick_power_rate 0.027)
(foul_detect_probability 0.5)(catchable_area_1_stretch 1.03085))
(playmode 1 kick_off_)
(team 1 opuSCOM NEO_FS 0 0)
(show 1 ((b) 0 0 0 0) ((l 1) 0 0x9 -49.1109 0.0076 -0.0444 0.003 -92.301 90 (v h
180) (s 8000 1 1 130555) (c 0 1 69 0 1 71 1 1 0 0 1)) ((l 2) 11 0x1 -18 -5 0 0 7.368
-3 (v h 180) (s 8000 0.941673 1 130600) (f 1 11) (c 0 0 50 0 1 51 1 0 0 0 1)) ((l 3) 8
0x1 -18 5 0 0 -38.621 26 (v h 180) (s 8000 0.805717 1 130600) (f 1 11) (c 0 0 50 0
1 51 1 0 0 0 1)) ((l 4) 7 0x1 -18 -14 0 0 -5.5 7 (v h 180) (s 8000 0.944233 1 130600)
(f 1 11) (c 0 0 50 0 1 51 1 0 0 0 1)) ((l 5) 16 0x1 -18 14 0 0 7.083 2 (v h 180) (s 8000
0.876201 1 130600) (f 1 11) (c 0 0 50 0 1 51 1 0 0 0 1)) ((l 6) 4 0x1 -15 0 0 0 29.599
1 (v h 180) (s 8000 0.996398 1 130600) (f 1 11) (c 0 0 50 0 1 51 1 0 0 0 1))
```

図 2: Game log

ル毎のプレイヤーとボールの位置や速度の情報、プレイヤーの行動、プレイヤーやコーチ間の情報の伝達等といった試合中の全ての情報が含まれている。そのため、試合ログを用いることで終了した試合を再生することができる。実際の試合ログを図 2 に示す。図 2 のように、試合ログは試合中の情報が文字列で表現されている。そのため、分析に用いる際は必要な情報のみを抽出する。

2.3 関連研究

RoboCup サッカーシミュレーション 2D リーグでは、戦術やポジショニング、相手チームの分析に関する様々な研究が行われている [8]。相手チームの分析に関する研究は、フィールド上のすべての情報をノイズなしで取得できるコーチエージェントを用いて行われる場合が一般的である。その一例として、Gregory ら [9] は決定木を用いて、プレイヤーに攻撃や守備のアドバイスをを行うコーチを開発した。また Ramin ら [10] はルールを用いたエキスパートシステムと決定木を用いたコーチ開発の研究を行った。Mazda ら [11] もまた、敵の行動予測のモデル化を研究した。戦術を表現するために、ポジショニングに着目した手法がいくつか提案されている。例えば、Luis ら [12] は自チームの戦術に応じてポジショニングを変更する手法を提案している。また Akiyama ら [13] は、Delaunay Triangulation を用いたボール位置に基づくプレイヤーのポジショニング手法を提案している。相手チームの戦術分析に関して、例えば Riley ら [14] は、敵プレイヤーのポジションやパス、ドリブルを記録することにより、相手チームを識別する手法を提案した。また、三船ら [5] は、キック情報に着目して戦術切り替えを行う手法を提案した。Visser ら [15] はフォーメーションを識別する研究を行った。しかしながら、相手チームの戦術を素早く確実に判断することはできていない。そのため、試合中に戦術を切り替えるためには、精度が高く、高速で判断可能な手法が必要である。

3 フォーメーション同定

本研究では、戦術はプレイヤーのポジショニングにより表現されるものと仮定する。相手チームの戦術を見極めるに

は、フォーメーションの特徴を正しく捉える必要がある。本章では、まず敵フォーメーション同定の概要を説明し、次に識別に用いる手法について以下に説明する。

3.1 敵位置情報に基づいたデータ抽出

敵のフォーメーションを数値表現するため、図 3 のようにサッカーフィールドをグリッド状に分割する。1 サイクル毎に各グリッド内に存在する敵プレイヤーの人数を計算する。各グリッド毎に計算した値を積算する。この積算値を、計算に用いたサイクル数により割ることで平均値を求める。この平均値を本実験での入力データとして扱う。例えば 6×4 に分割した際は、24 次元ベクトルのデータが計算される。

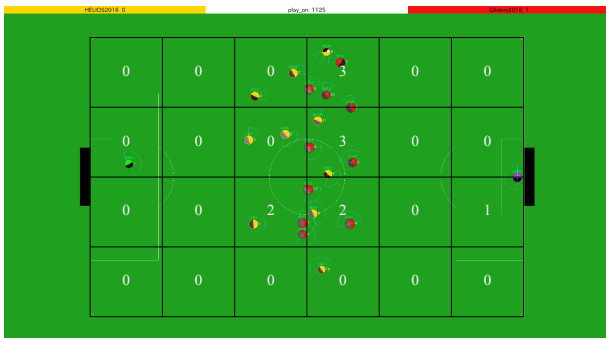


図 3: Discretization of the soccer field by a grid of size 6×4 .

3.2 フォーメーションの識別モデル

フォーメーションを識別するためのモデルを構築する。識別モデルによる有効性を調査するため、複数の機械学習手法によりモデルを構築する。モデルの構築には、ニューラルネットワーク (NN)、サポートベクターマシン (SVM)、ランダムフォレスト (RF) を用いて、教師あり学習を行う。それぞれのデータに対するラベル付けは、相手チームのフォーメーションタイプによって決定する。ラベル付けの方法については??章で、教師データの作成については 4 章で後述する。

本論文では、ボールの支配時間が長いにもかかわらず得点に結びつかない場合、戦術の相性が悪いと定義し、戦術を切り替えるべきであると判断する。相手チームが戦術切り替えの対象かどうかを識別するために、試合ログから自チームのボール支配率を計算する。キックを行ったプレイヤーと、その次にキックを行ったプレイヤーが同一チームである場合、その間のサイクルはキックを行ったチームがボールを支配しているものとする。不得意度を表す指標を、式 (1) と定義する。ここで、 k 試合目における自チームのボール支配率を $p(k)$ 、得点を $g(k)$ とおく。

$$score(k) = d(g(k)) \cdot p(k), \quad (1)$$

$$d(x) = \begin{cases} 1 & (x = 0) \\ 0 & (x \geq 1). \end{cases} \quad (2)$$

複数の試合ログから式 (1) を計算し、その平均値が閾値以上のチームを戦術切り替えの対象チームとして扱う。

3.3 敵位置情報に基づくチーム分類とラベル付け

本項では、位置情報に基づく相手チームの分類手法について説明する。相性の悪いチーム戦術における特徴的なフォーメーションタイプを調査するため、3.2 節で戦術切り替えが必要であると判断されたチームを対象に、クラスタリングを行う。本論文では、クラスタリングのために混合ガウス分布を用いる。混合ガウス分布のハイパーパラメータは EM アルゴリズムを用いて最適化する。最適クラスタ数は Calinski-Harabasz 基準値 [16] によって決定する。Calinski-Harabasz 基準値ではクラスタ内分散とクラスタ間分散の比を用いる。

3.3 節の結果から、3.1 節で先述したデータにラベル付けを行う。これを識別モデルの教師データとして扱う。また、ラベルの数は最適クラスタ数に準拠する。

4 数値実験

本論文では、実際の試合ログを用いて数値実験を行う。まず、戦術切り替えの対象であるチームのうち、特徴的なフォーメーションを分類する。次に、適切な戦術に切り替えるため、敵位置情報を用いてフォーメーションを識別する。この際、敵フォーメーションの識別に必要なサイクル数を検証するため、サイクル数による識別率の変化を調査する。また、グリッド数による識別率の変化を調査する。

4.1 フォーメーションの分類

本節では、戦術の切り替えが必要であるチームのフォーメーションを分類する。最適クラスタ数を決定することで、特徴的なフォーメーションの数を調査する。

4.1.1 実験設定

試合ログを作成するために、HELIOS [17] と CYRUS2014, InfoGraphics, HERMES2015, Gliders2016, FURY, HERMES2016, MarliK2016, Ziziphus, FRA-UNited, WrightEagle, Ri-one の 11 チームを 200 試合ずつ対戦させる。本実験では前半のみの試合ログを使用する。HELIOS は本研究室と福岡大学が共同で開発しているチームである。他のチームは RoboCup 2014-2016 に出場したチームのうち、3.2 節の手法から戦術切り替えが必要と判断したチームである。CYRUS2014, InfoGraphics は RoboCup 2014 に、HERMES2015, WrightEagle は RoboCup 2015 に出場したチームである。Gliders2016, FURY, HERMES2016, MarliK2016, Ziziphus, FRA-UNited, Ri-one は RoboCup 2016 に出

場した最新のチームである。クラスタ数を 2-10 個の間で変化させ、Calinski-Harabasz 基準値が高いクラスタ数を最適クラスタ数の候補とする。入力データとして、フィールドを 30×20 のグリッド状に分割したものを利用して、データを作成する。

4.1.2 実験結果

実験結果を図 4 に示す。図 4 から最適クラスタ数の候補として 3 または 4 が考えられる。クラスタ数を 3 としたとき、図 5 上図のようにゴール前に全てのプレイヤーを集める守備を偏重したチーム (CYRUS2014, HERMES2015, FURY, Ziziphus) と、図 5 下図のように一直線状に並びスルーパスを防ぐようなフォーメーションのチーム (InfoGraphics), その他のチームという 3 種類に分類された。クラスタ数を 4 としたときは、クラスタ数 3 の結果に加えて、同様に一直線状に並ぶチームが新たなクラスタとして選出された (Gliders2016)。InfoGraphics と Gliders2016 の違いとして、Gliders2016 はボールを保持すると攻撃的になるのに対し、InfoGraphics はボールを保持しても攻めてこないことが考えられる。そのため、InfoGraphics と Gliders2016 を同じフォーメーションとして扱い、最適クラスタ数は 3 とする。以降、図 5 上図のようなフォーメーションのチームを “wall”, 図 5 下図のようなフォーメーションのチームを “line”, そうでないチームを “normal” と呼ぶことにする。

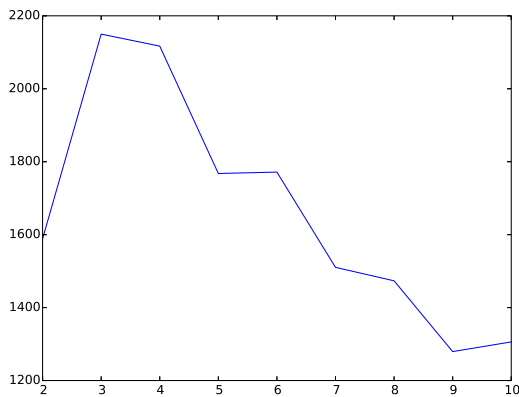


図 4: Variation of the Calinsky-Harabasz index according to the number of clusters.

4.2 フォーメーション識別

本節では、実験 4.1 の結果を踏まえ、敵位置情報に基づきフォーメーション識別を行う。オンラインでの判断への有効性を示すため、経過時間と識別率の関係性を検証する。また、グリッド数による識別率の変化を調査する。

4.2.1 実験設定

実験 4.1 の結果を踏まえ、wall, line, normal の 3 クラス識別を行う。フィールドを 6×4 , 12×8 , 15×10 ,

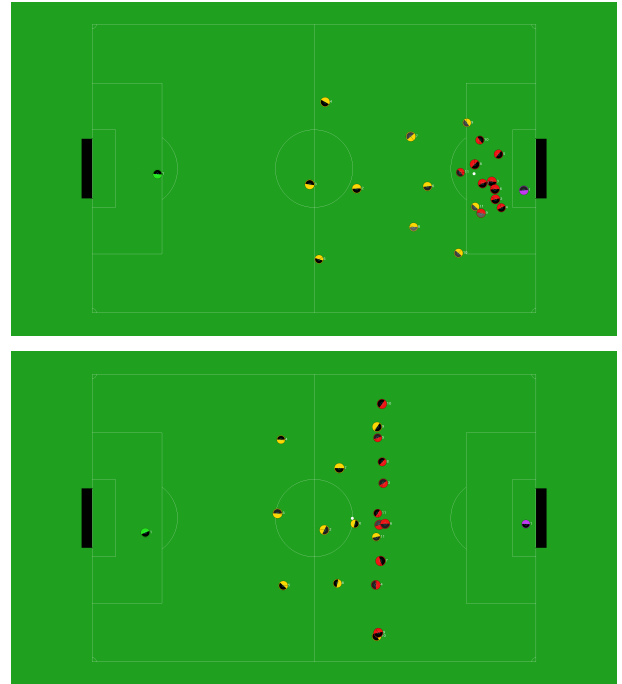


図 5: Typical defensive formations. Top: “wall”, bottom: “line”.

24×16 , 30×20 の 5 種類のグリッド状に分割し、教師データを作成する。HELIOS と RoboCup 2016 に出場した全てのチーム, CYRUS2014, InfoGraphics, HERMES2015, WrightEagle の 19 チームを対戦させ、前半のみの試合ログを作成する。4.1.1 項の際と同様に、CYRUS2014, InfoGraphics は RoboCup 2014 に出場したチームであり、HERMES2015, WrightEagle は RoboCup2015 に出場したチームである。wall のチームのうち、CYRUS2014 と FURY は得失点差によって戦術を変更する。そのため、この 2 チームに対しては自チームが無得点である場合のみのデータを用いる。それぞれのクラスに対して、約 3000 個のログから位置情報を抽出し、教師データとして扱う。データのラベル付けは 4.1 節の結果に従う。ラベル付けの一覧を表 1 に示す。識別器として NN, SVM, RF を用いる。識別器のパラメータは予備実験の結果から表 2 のように設定する。

表 1: Opponent teams’ formation labels

Opponent Team	label
CYRUS2014	wall
HERMES2015	wall
FURY	wall
Ziziphus	wall
InfoGraphics	line
Gliders2016	line
others	normal

表 2: Hyper-parameters used for the classifiers

Classifier	Parameter	Setting
NN	Activation function	Logistic function
	Optimization algorithm	L-BFGS method
	Structure	3 layers
	The number of neurons in Input layer	The number of grid
	The number of neurons in Output layer	3 neurons
	L2 penalty	0.0001
	Tolerance	0.0001
SVM	Kernel	Linear
	Penalty	1.0
	Tolerance	0.0001
RF	Criterion	Gini index
	The number of trees	10
	Sampling	Bootstrap

4.2.2 実験結果

実験結果を図 6-10 に示す。図 6-10 それぞれは、フィールドを 6×4 , 12×8 , 15×10 , 24×16 , 30×20 に分割した場合の実験結果を表している。どのグリッド数においても 1500 サイクル付近から、収束していることが分かる。サイクル数が十分多いとき、識別器による識別率の違いはあまり見られない。NN と SVM については、グリッド数に比例し、少ないサイクル数での識別率が高い。このことから、グリッド数が多ければ、より早い段階で識別が可能であることが分かる。また、RF と比較して NN と SVM は収束に必要なサイクル数が少ないことが分かる。一方、RF ではサイクル数が少ない場合、NN, SVM と比較して識別能力が低い。このことから、RF を使用するにはある程度のデータ量が必要であると考えられる。

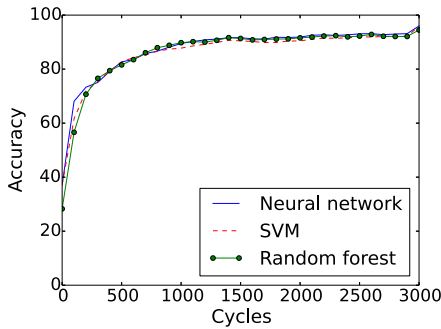


図 6: Accuracy rates of the three models according to field discretized with a grid of size 6×4

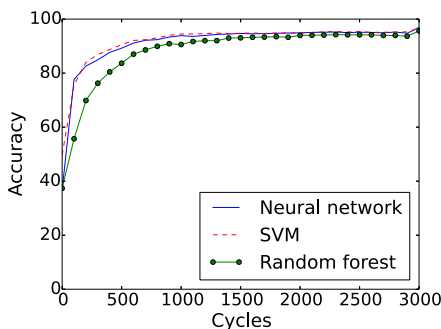


図 7: Accuracy rates of the three models according to field discretized with a grid of size 12×8

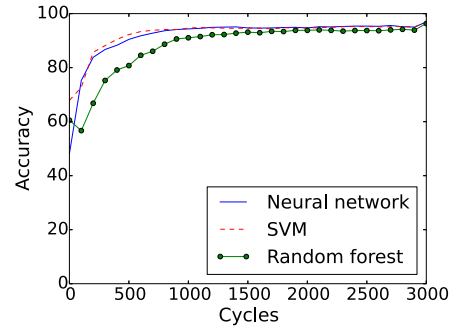


図 8: Accuracy rates of the three models according to field discretized with a grid of size 15×10

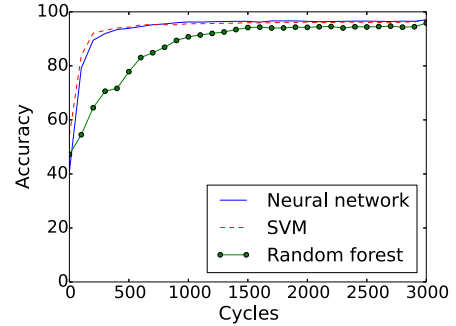


図 9: Accuracy rates of the three models according to field discretized with a grid of size 24×16

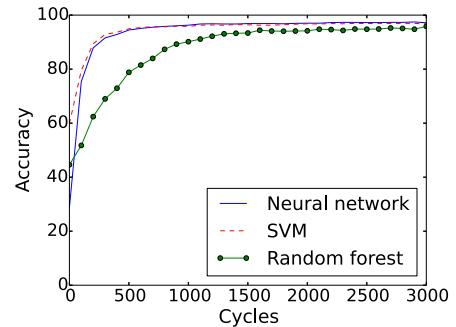


図 10: Accuracy rates of the three models according to field discretized with a grid of size 30×20

5 おわりに

本論文では、敵の戦術を見極めるために、位置情報を用いてオンラインで戦術を判断する手法を提案した。敵の戦略を見極めることができれば、それに応じて自チームの戦術を切り替えることができる。そのために、敵フォーメーションを同定しラベル付けを行い、機械学習手法を用いて識別モデルを構築した。今後の課題として、未知のチームに対する識別率の調査や、オンライン識別器の実装などがあげられる。

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] Luiz A. Celiberto Jr., Carlos H. C. Ribeiro, Anna Helena Reali Costa, and Reinaldo A. C. Bianchi. “Heuristic Reinforcement Learning applied to RoboCup Simulation Agents,” *Proc. of the 11th RoboCup Symposium*, pp. 220-227, 2007.
- [3] Aijun Bai, Feng Wu, and Xiaoping Chen. “Towards a Principled Solution to Simulated Robot Soccer,” *Proc. of the 16th RoboCup Symposium*, pp. 141-153, 2012.
- [4] Thomas Gabel, Martin Riedmiller and Florian Trost. “A Case Study on Improving Defense Behavior in Soccer Simulation 2D: The NeuroHassle Approach,” *Proc. of the 12th RoboCup Symposium*, pp 61-72, 2008.
- [5] 三船 哲史, 中島 智晴, 秋山 英久, “RoboCup サッカーにおけるキック分布を用いたゲーム戦術の切り替え,” *Proc. of the 44th Meeting of Special Interest Group on AI Challenges*, pp. 1-6, 2016.
- [6] Jordan Henrio, Thomas Henn, Tomoharu Nakashima, and Hidehisa Akiyama, “Selecting the Best Player Formation for Corner-Kick Situations Based on Bayes’ Estimation.” *Proc. of the 20th RoboCup Symposium*, 12 pages, 2016.
- [7] 秋山 英久, “ロボカップサッカー シミュレーション 2D リーグ必勝ガイド,” 秀和システム, 2006.
- [8] Shokkofeh Pourmehr, Chitra Dadkhah, “An Overview on Opponent Modeling in RoboCup Soccer Simulation 2D,” *Proc. of the 15th RoboCup Symposium*, pp. 402-414, 2011.
- [9] Gregory Kuhlmann, Peter Stone, and Justin Lallinger, “The UT Austin Villa 2003 Champion Simulator Coach: A Machine Learning Approach,” *RoboCup 2004: RoboCup 2004: Robot Soccer World Cup VIII* pp. 636-644, 2005.
- [10] Ramin Fathzadeh, Vahid Mokhtari, Morteza Mousakhani, and Alizera Mohammad Shahri, “Coaching with Expert System Towards RoboCup Soccer Coach Simulation,” *Proc. of the 10th RoboCup Symposium*, pp. 488-495, 2006.
- [11] Mazda Ahmadi, Abolfazl Keighobadi Lamjiri, Mayssam M. Nevisi, Jafar Habibi, and Kambiz Badie, “Using a Two-Layered Case-Based Reasoning for Prediction in Soccer Coach,” *Proc. of the International Conference on Machine Learning; Models, Technologies and Applications. MLMTA ’03*, pp. 181-185, 2003.
- [12] Luis Paulo Reis, Nuno Lau and Eugnio Oliveira, “Situation Based Strategic Positioning for Coordinating a Simulated RoboSoccer Team,” *Balancing Reactivity and Social Deliberation in MAS*, Vol. 2103, pp. 175-197. 2001.
- [13] Hidehisa Akiyama, Itsuki Noda, “Multi-Agent Positioning Mechanism in the Dynamic Environment,” *Proc. of the 12th RoboCup Symposium* , pp. 377-384, 2008.
- [14] Patrick Riley, Manuela Veloso, “On Behavior Classification in Adversarial Environments,” *Proc. of the 5th Distributed Autonomous Robotic Systems (DARS 2000)*, pp. 371-380, 2000.
- [15] Ubbo Visser, Christian Drücker, Sebastian Hübner, Esko Schmidt, and Hans-Georg Weland, “Recognizing formations in opponent teams,” *RoboCup 2000: Robot Soccer World Cup IV*, pp. 391-396, 2001.
- [16] Calinski, T., and J. Harabasz, “A dendrite method for cluster analysis,” *Communications in Statistics*, Vol. 3, No. 1, pp. 1-27, 1974.
- [17] Hidehisa Akiyama, Tomoharu Nakashima, Jordan Henrio, Thomas Henn, Sho Tanaka, Tomonari Nakade, Takuya Fukushima, “HELIOS2016: Team Description Paper,” *RoboCup2016 Leipzig, Germany*, 6 pages, 2016.

Improvement of Multiple Robots' Self-localization by Using Perspective Positional Information

Yo Aizawa^{*1}, Takuo Suzuki^{*2}, and Kunikazu Kobayashi^{*3}

^{*1,2,3}Department of Information Science and Technology, Aichi Prefectural University, Aichi, Japan

^{*1}im172001@cis.aichi-pu.ac.jp, ^{*2}takuo.suzuki@ist.aichi-pu.ac.jp, ^{*3}kobayashi@ist.aichi-pu.ac.jp

Abstract

This study ¹ aimed to improve the precision of multiple robots' self-localization in the standard platform league of RoboCup, i.e. a robotic soccer competition. For improving the precision of the self-localization, we proposed a new technique that uses an external camera out of the field for assistance. Robots in the field use the unscented particle filter that estimates their position from some landmarks. When a robot equipped with the filter cannot recognize any landmarks exactly, particles spread and the precision of the self-localization decreases. Therefore, the overlooking camera out of the field observes each robot's position. When particles spread, the external camera estimates the foot position of the robot, and then the robot sprinkles particles on the neighborhood again. In this way, even if a robot cannot recognize landmarks exactly, assists of the external camera revise the position of particles and improve the precision.

1 Introduction

The RoboCup (Robot Soccer World Cup) project sets a goal that a fully autonomous robot team shall win against the most recent winning team of FIFA World Cup in soccer by 2050.

The RoboCup Soccer Standard Platform League (SPL) is a league that all teams compete with the same standard humanoid robot called NAO developed by Softbank Robotics[1]. The robot operates fully autonomously, that is with no external control, neither by humans nor by computers. In RoboCup Soccer SPL, the robot must process all the calculations on vision processing and decision making using low-end CPU (Intel Atom 1.6GHz). In addition, the robot must devote a lot of computation resource to percept a white goal and a mostly white ball in vision processing. Each team has

five player robots and optionally has one coaching robot that can send instructions at a perspective view from outside the field. An example of the positional relationship between the field and the coaching robot is shown in Figure 1.

In RoboCup Soccer SPL, a self-localization mechanism that estimates player own position and orientation is required. We use the unscented particle filter (UPF)[3] which is currently a mainstream method[4] for self-localization. However, a robot cannot accurately grasp any landmarks, then particles do not converge, so the estimation error of self-localization becomes large.

In addition to the conventional method, by using the coaching robot as the observer, an area where a player is likely to exist is specified. We propose a method to promote convergence of particles by correcting the coordinates of scattering particles based on the information from the coaching robot. From this method, estimation error of the self-location is assumed to be suppressed when the player cannot accurately recognize landmarks.

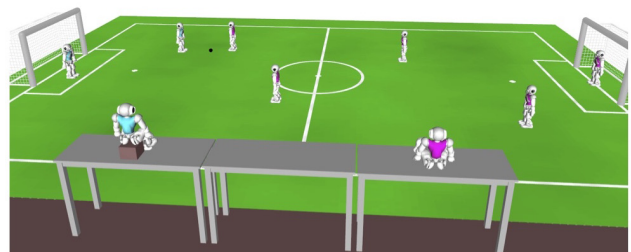


Figure 1: Coaching robots can observe the almost whole field[2]

2 Unscented Particle Filter (UPF)

The UPF is a combination of the unscented Kalman filter (UKF)[5] and a particle filter (PF)[6]. The difference between UPF and PF is that UPF is used the UKF for updating each particle.

The UPF estimates the position of the robot by using

¹This paper was submitted to SICE Annual Conference 2017.

a finite number of particles assumed to be the robot. The first step is motion update step. In this step position of each particles is updated the by using robot motion information. The second one is measurement update step. The robot calculates the weight of each particles based on observation information. The third one is resampling step. It sprinkles the particles according to the weights.

3 Proposed method

When UPF cannot accurately grasp landmarks, particles may not converge. When such a situation occurs, the coaching robot behaves as an observer, assists to estimate the self-localization of the player from the outside, and encourages the convergence of the particles. The flow of the proposed method is shown in Figure 2.

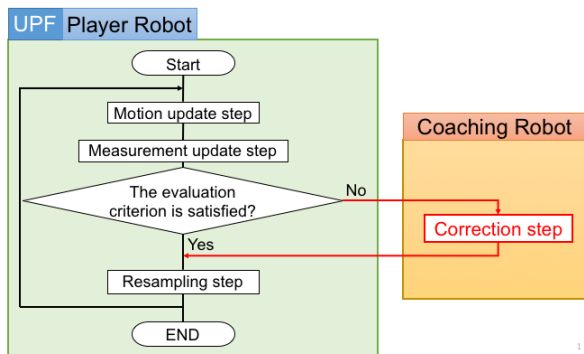


Figure 2: Outline of proposed method

3.1 True perspective image

At first, the coaching robot gets a perspective image as shown in Figure 3. Then it is transformed to a true perspective image by using homography transform[7] (see Figure 4). Since the homography transform requires more than four coordinates on an image, the coaching robot will select more than four points out of 17 candidates, i.e. four corners of the field, eight corners of the penalty areas, two penalty marks, two intersections of the center line and the side lines, and a point of the center mark. In Figure 3, we use eight points by indicating red circles.



Figure 3: Original image with known positions.

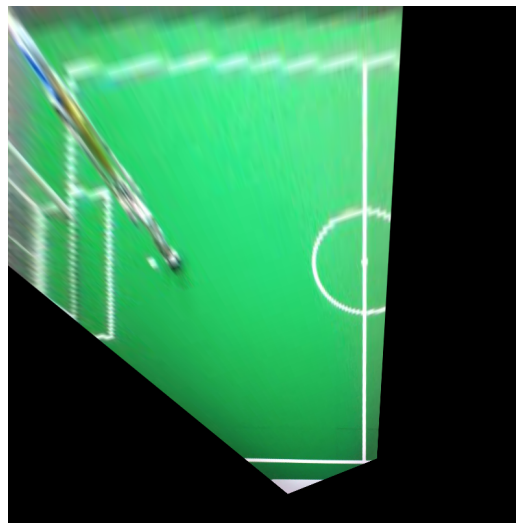


Figure 4: Homography transformation

3.2 Estimation of a player's position

We estimate straight lines with a high possibility that a robot exists. Only the jersey regions are extracted from the transformed image. Then, the regions are denoising by opening processing[8] (see Figure 5) and Increasing connectivity by closing processing[8]. After that, we extract regions of the own team's jersey, they are certain that the player robot will be on the line calculated by simple linear regression analysis (see Figure 6).

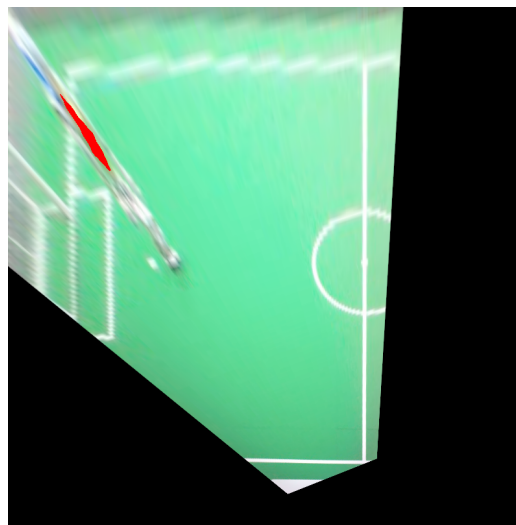


Figure 5: Extraction of uniform

3.3 Estimation of a player's foot position

We estimate the foot of the player robots and use it as the reference of the position at which particles are resampled. The foot position of the player robot is estimated as the bottom point of regions excluding the field on the line(Figure 6). We transform the color space of the perspective image into $L^*a^*b^*$ to detect the color of the green field. L^* stands for lightness and a^* and b^* are chromaticness index equivalent to hue and sat-

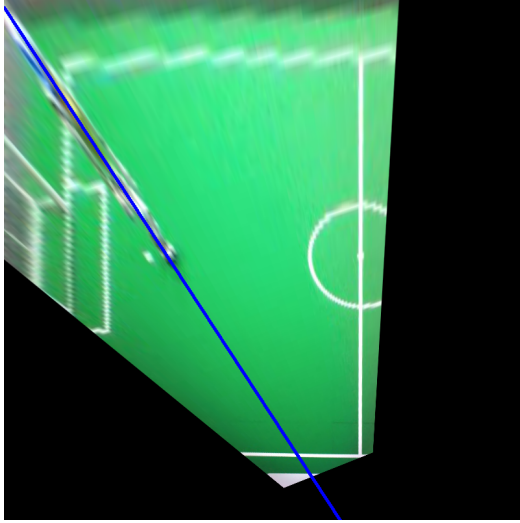


Figure 6: Straight line expressing rough robot position



Figure 8: Background subtraction

uration. The color approaches red as the value of a^* becomes high and green as it becomes low, and yellow as the value of b^* becomes high and blue as it becomes low. We binarize the image of a^* by Otsu's thresholding method[9]. By doing so, we extract regions other than the green color of the field. As applying the homography transform to the image, the true perspective image is shown in Figure 7. There is a possibility that the estimated position of the feet may be displaced by the line of the field in Figure 7. Therefore, the region of the moving object is extracted using the background difference and the position of the robot is specified. The region of the moving object obtained from the background subtraction is shown in Figure 8.

The region with the most continuous region of Figure 7 on the straight line is extracted. The lowest point of the region of Figure 8 included in this region is regarded as the foot. The estimated foot position is illustrated in Figure 9 as a red circle.

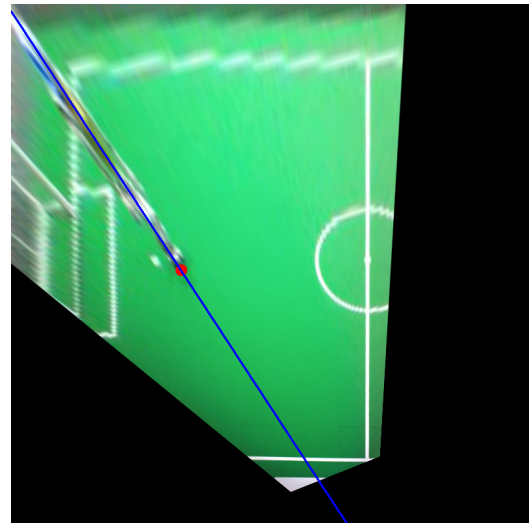


Figure 9: Estimated position of robot's foot

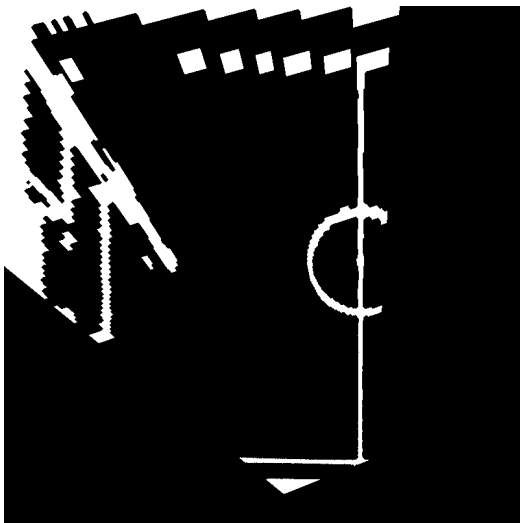


Figure 7: Image except green field region

4 Experiment

We verify whether the player can be assisted self-localization of using images acquired by the coaching robot. Firstly, it is evaluated how the estimated foot position is closer to the true one by comparing the proposed and the conventional methods. Secondly, after correcting the position of the particle by the proposed method, it is verified whether it is close to the true position as compared with the conventional method.

The experiments were conducted and used two players under an LED uniform lighting environment with natural light. We use the OpenCV 3.1 library as a tool for image processing. The value of α in the normal distribution in Section 3.4 is empirically set to 8 in order to prevent particles from spreading. The number of particles is 12.

3.4 Determination of resampling position

Based on the estimated foot position, the locations where particles are scattered are determined. Taking into account the error of the estimated foot position, the positions of particles are determined according to the normal distribution as given by Eq. (1).

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \quad (1)$$

where μ is the mean and σ^2 is the variance. In this paper, the value of μ is defined by the foot position x , and the value of σ is set to $1/\alpha$. The particles are gathered into the foot estimated by increasing the value of α . Based on the above, the positions of particles are indicated by yellow circles in Figure 10.

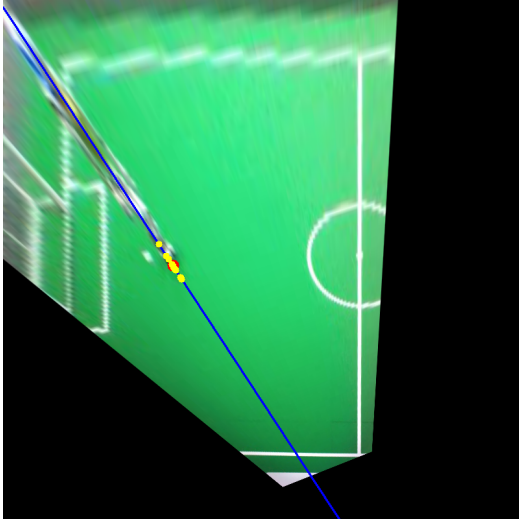


Figure 10: Resampled position of particles

4.1 Experiment 1: Verification of the accuracy of the estimated foot position

When distributing particles using the proposed method, the estimated position accuracy of the estimated foot position of Section 3.3 is important. Because the coordinates of particle resampled are highly based on the foot position. Therefore, we verify the accuracy of the estimated foot position using the proposed method by measuring the actual foot position. In addition, we compare the estimation error of the self-localization with the conventional method.

In this experiment, two conditions are set in order to see the change in error according to the distance between the coaching and the player robots. Therefore, we estimate the foot positions of two robots simultaneously. As shown in Figure 11, the robot is placed, the robot A is closer to the coaching robot, and the robot B is far one.

In the experiment, the robots follow a path as shown in Figure 11 where landmarks such as lines and goals are difficult to recognize and self-position estimation becomes difficult. We set the player robots in the red circles as the initial state and walk to the blue circles according to the red arrows. At that time, self-position estimation

is performed using UPF. When both the robots reached the blue circle, the coaching robot estimates the foot position of the player robots. Experiments were carried out three times and the errors against the true position are averaged to compare the accuracy.

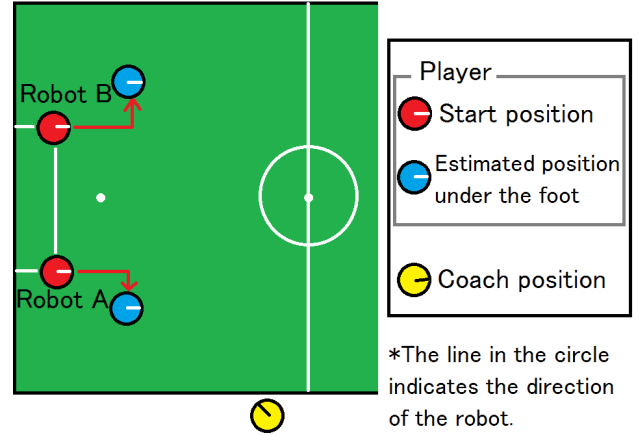


Figure 11: Routes in experiment 1

4.2 Result of Experiment 1

The experimental results in Experiment 1 are shown in Table 1. Improved rate in Table 1 is obtained from Eq. (2). In Eq. (2), R is the Improve rate, E_c is the error average of the conventional method, and E_p is the error average of the proposed method.

$$R = \frac{E_c - E_p}{E_c} * 100 \quad (2)$$

From Table 1, Both the robots A and B are more accurate than the conventional method, so it can be applied even in situations where there is a difference in observation distance between the coach robot and the player robot. In addition, the total improvement rate of the accuracy of the estimated position at the total of 6 times by the two robots A and B three times is 74%. Therefore, on the basis of the estimated foot position, resampling particles is expected to improve the accuracy. Moreover, we could confirm that it is possible to estimate not only one robot but also multiple robots.

Table 1: Average error of the estimated foot position (Experiment 1)

Method \ Robot	Average error [mm]	
	Robot A	Robot B
Conventional method	630	450
Proposed method	199	80
(Improved rate [%])	(68)	(82)

4.3 Experiment 2: Verification of the accuracy of self-localization after resampling

After resampling the particles using the proposed method, the robot moves again and the self-position es-

timization accuracy at the last position is verified. We also compare the estimation error of the self-localization with the conventional method.

As seen in Section 4.1, the player robots moves by two kinds of routes as illustrated in Figure 12 . Experiments were carried out three times and the errors against the true position are averaged to compare the accuracy.

As in Experiment 1, robots walk to the blue circles in Figure 12, then correct the particle position only once using the proposed method at the position of the blue circles. After that, when it reaches the gray circle along the red solid arrow shown in Figure 12, it estimates its own position. We compare the accuracy of self-position estimation with the normal UPF and that with the proposed UPF that corrected particles only once using the proposed method.

When resampling is performed using the proposed method, the direction of the particles is determined according to the normal distribution based on the estimated direction. The normal distribution is given by Eq. (1). In self-localization, the direction is corrected by recognizing landmarks. Therefore, the value of μ is set to the previous estimated direction and the value of σ is empirically set to $\pi/8$.

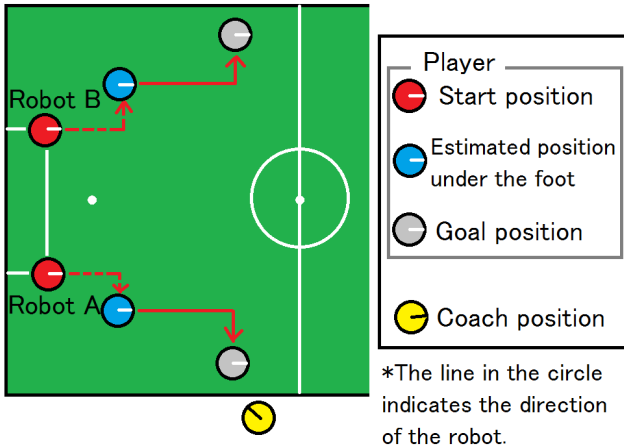


Figure 12: Routes in experiment 2

4.4 Result of Experiment 2

The results in Experiment 2 are shown in Table 2. Improved rate in Table 2 is obtained from Eq. (2).

From Table 2, both the robots A and B using the proposed method are more accurate than those using the conventional method., so it can be applied even in situations where there is a difference in observation distance between the coach robot and the player robot. In addition, the total improvement rate of the accuracy of the estimated position at the total of 6 times by the two robots A and B three times is 72%. Moreover, we could confirm that it is possible to estimate not only one robot but also multiple robots.

The coaching robot can not always estimate the feet of the player robot at all times. However, once using

the proposed method from this experiment, it was confirmed that the estimation accuracy was improved after that. Therefore, under the situation where the player’s foot can be estimated, it is expected that the estimation accuracy after that can be improved by using the proposed method.

Table 2: Average error of self-localization (Experiment 2)

Coach \ Robot	Average error [mm]	
	Robot A	Robot B
without coach	987	606
with coach	169	274
(Improved rate [%])	(83)	(55)

5 Conclusion

In this paper, we proposed a method for improving the accuracy of the self-position estimation method, the UPF, in the RoboCup soccer standard platform league. In the proposed method, the position of the particle is corrected by using the observer (coach robot) assisted the subjects (player robots) who performs self-position estimation. As a result, by using the proposed method, the estimation accuracy of the self position is improved by 72% compared with the conventional method.

In addition, since improvement of estimation accuracy after that can be confirmed by correcting the position of the particle once using the proposed method, it is expected that estimation accuracy will improve only by using the proposed method when the coaching robot can estimate player’s foot.

As future work, when there are two or more player robots as in a normal game, it is necessary for the player robot to discriminate from the position information of the player robot estimated by the coaching robot which information about ourselves.

Acknowledgements

This work was partly supported by Aichi Prefectural University, Japan.

References

- [1] Aldebaran “NAO H25”, Aldebaran documentation “<http://doc.aldebaran.com/2-1/index.html>”
- [2] RoboCup Technical Committee: “RoboCup Standard Platform League (NAO) Rule Book”, June 9, (2016).
- [3] R. van der Merwe, A. Doucet, N. de Freitas, and E. Wan: “The Unscented Particle Filter”, Proc. of NIPS, pp.584-590 (2000).
- [4] T. Röfer, T. Laue, J. Richter-Klug, M. Schünemann, J. Stiensmeier, A. Stolpmann, A. Stöwing, F. Thielke, B-Human Team Report and Code Release 2015 (2015)

- [5] R.E.Kalman: “ A New Approach to Linear Filtering and Prediction Problems ” , Transactions of the ASME-Journal of Basic Engineering, vol.82 (Series D): pp.35-45 (1960).
- [6] S. Thrun, W. Burgard, and D. Fox, Probabilistic Robotics, The MIT Press (2005).
- [7] R. Hertley and A. Zisserman, “ Multiple View Geometry in computer vision”, 2nd Edition, Cambridge University Press, pp.32-36, (2003).
- [8] R. Szeliski, Computer Vision: Algorithms and Applications, Springer (2011).
- [9] N. Otsu, “ A Threshold Selection Method from Gray-Level Histograms”, IEEE Trans. Systems, Man, and Cybernetics, vol.9, no.1, pp.62-66, (1979).

© 2017 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にお願いします.)

AI チャレンジ研究会

主査

公文 誠

熊本大学 大学院先端科学研究部

Executive Committee Chair

Makoto Kumon

Faculty of Advanced Science and
Technology, Kumamoto University
kumon @ gpo.kumamoto-u.ac.jp

主幹事 / 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

Secretary

Noriaki Mitsunaga

Department of Technology Education,
Osaka Kyoiku University

担当幹事

植村 渉

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

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

Reiji Suzuki

Department of Complex Systems Science,
Graduate School of Informatics,
Nagoya 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/>