

## AI チャレンジ研究会 (第 42 回)

*Proceedings of the 42nd Meeting of Special Interest Group on AI Challenges*

### CONTENTS

- RoboCup サッカーにおけるキック分布を用いた勝敗予測…………… 1  
三船 哲史 \*1, 中島 智晴 \*1, Jordan Henrio \*1, 秋山 英久 \*2, 綾井 創一 \*1  
1 大阪府立大学, \*2 福岡大学
- Estimation of Players' Actions in Soccer Matches Based On Deep Autoencoder…………… 7  
Jorge L. Copete, Junichi Suzuki, Quan Wei, Ryo Iwaki, Nobutsuna Endo, Hiroki Mori, Yukie Nagai,  
Minoru Asada (Graduate School of Engineering, Osaka University)
- Implementation of Acoustic Communication in RoboCup SPL…………… 13  
Wen Chun Chen, Jia Jun Tsai, N. M. Mayer (National Chung Cheng University)
- タブレット端末で動作する Arduino 用プログラミング環境 aiBlocks の初心者向け作例集の試作と評価  
…………… 16  
光永 法明 \*1, 吉田 図夢 \*2, 井芹 威晴 \*3  
1 大阪教育大学, \*2 大阪教育大学卒業, \*3 岸和田市立桜台中学校
- 人間動作の観察から動作模倣を行う人型ロボットのためのパーティクルフィルタを用いた関節角度時系列  
推定…………… 21  
高橋 泰岳, 榊原 洸太 (福井大学)
- 渋滞に対する複数ロボットの経路設定に関する一考察…………… 24  
内海 孝亮, 植村 渉 (龍谷大学)
- 全方位移動ロボットに対する移動指示インタフェースの評価…………… 28  
辻 和輝, 植村 渉 (龍谷大学)
- 自律移動ロボットの環境地図作成のための簡易測域センサの利用と評価…………… 34  
津田 諒太, 植村 渉 (龍谷大学)

日時: 2015年5月3日

場所: 福井県生活学習館 3F 学習室303

*Fukui Life-long Learning and women's Center, Fukui, May 3, 2015*

社団法人人工知能学会

Japanese Society for Artificial Intelligence

# RoboCup サッカーにおけるキック分布を用いた勝敗予測

## Predicting Game Results using Kick Distributions in RoboCup

三船 哲史 †, 中島 智晴 †, Jordan Henrio †, 秋山 英久 ‡, 綾井 創一 †

Satoshi Mifune †, Tomoharu Nakashima †, Jordan Henrio †, Hidehisa Akiyama ‡, Soichi Ayai †

大阪府立大学 †, 福岡大学 ‡

Osaka prefecture University †, Fukuoka University ‡

satoshi.mifune@cs.osakafu-u.ac.jp, tomoharu.nakashima@kis.osakafu-u.ac.jp

jordan.henrio@cs.osakafu-u.ac.jp, akym@fukuoka-u.ac.jp, swa01014@edu.osakafu-u.ac.jp

### Abstract

Predicting the game result using kick distributions is studied in this paper. Although it is not possible to exactly know a strategy that a team is taking, that strategy might be well represented by how the players in the team kick the ball during games. Passes and dribbles that are made during a game are extracted to form a kick distribution. It is assumed that the kick distribution represents the strategy of a team. A series of computational experiments are conducted in order to examine the performance of the proposed method.

### 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、それぞれにおいて活発な研究、開発が行われている。Gabelら[2]は、過去の大会の傾向の分析を行い、チームの戦術を定量的に評価している。この研究はこれまでの世界大会に参加したチームを振り返り、戦術がどのように進化しているか調査することを目的としている。また、Abreuら[3]は、ログの情報をを用いてロボットのサッカーと人間のサッカーを比較している。この研究は、ロボットの知能が人間の知能に近いかどうかを調査することを目的としてログファイルを利用している。しかし、ログの情報から相手の戦略を懐石している研究はまだ積極的に行われていない。フィールドの片側に選手を固め攻撃する戦術や守備を偏重する戦術など、様々な戦術が用いられている。戦術間の相性があるため、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、試合中に自チームの戦術と相手チームの戦術との相性を判断し、適切な戦略を選択することで試合を有利に進める

ことが必要である。RoboCup サッカーシミュレーション 2D リーグにおいて、試合ログには、プレイヤーやボールの位置情報、プレイヤーの行動、プレイヤーやコーチ間の情報の伝達などの試合中の全ての情報が含まれている。しかし、試合ログから有益な情報を抽出する効果的な手法は提案されていない。そこで、本論文では試合ログをキック分布として表現することにより、相手チームの戦術を判断する手法を提案する。

本論文では、試合ログを相手チームの戦術や相性の良し悪しで分類する。また、分類結果を用いて前半戦の試合ログから勝敗予測を行う。Earth Mover's Distance (EMD) [4]を類似度として、試合ログをクラスタに分類する。そして、クラスタリング結果を用いて、前半戦の試合ログから勝敗予測を行う。数値実験では、ベースチームの異なる4チームと対戦を行い、得られた試合ログを分類する。クラスタリング結果を用いて、前半戦の試合ログから勝敗を予測し、試合結果と一致しているかどうかにより勝敗予測の精度を調査する。

### 2 RoboCup

RoboCupは、ロボット工学と人工知能の発展が目的の自律移動型ロボットによるサッカーやレスキュー、家庭内作業などを題材とした研究プロジェクトである。RoboCupには「西暦2050年までに、サッカーの世界チャンピオンチームに勝てる自律型ロボットチームを作る」という目標があり、この目標に向けて盛んに研究が行われている。RoboCupにはサッカー以外にも、大規模災害への対応のシミュレーションや災害現場で活躍するロボットの開発を促進するレスキューリーグ、日常生活で人間を支援する自律ロボットによる競技を通じて、人とコミュニケーションしながら役に立つロボットの実現を目指す@ホームリーグの他に、次世代のロボット技術者育成を目的としているジュニアリーグも存在する。本論文では、RoboCup サッ

カーシミュレーションリーグを研究の対象とする。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。サッカーシミュレーションでは、実機を使用せずに、コンピュータ内に用意された仮想フィールド上でサッカー競技を行う。サッカーシミュレーションには2Dリーグと3Dリーグがある。Figure 1, 2に2Dリーグと3Dリーグの試合の様子を示す。2Dリーグでは、基本的な動作(キックやドリブルなど)はコマンドとして実装されている。そのため2Dリーグでは高レベルな意思決定を主な研究対象としている。一方、3Dリーグでは、エージェントはヒューマノイドロボットで形成されているため、基本的な動作を関節から制御する必要があり、基本的な動作が非常に重要である。本論文では2Dリーグを扱う。2Dリーグでは、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のベクトルとして表される。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。各プレイヤー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おり、制限された視覚情報や聴覚情報からドリブルやパス等の行動選択を行う。

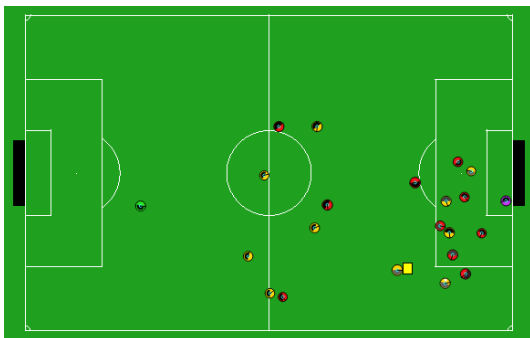


図 1: 2D Simulation League



図 2: 3D Simulation League

### 3 試合ログのクラスタリング

本論文では、試合ログをクラスタリングする。試合ログから特徴量としてキック分布に注目する。キック分布をクラスタリングするための距離尺度としてEarth Mover's Distanceを用いる。本章では、まずキック分布を説明し、次にキック分布間の距離を測るEarth Mover's Distanceの概要を説明する。

#### 3.1 キック分布

本論文では、チーム戦略を表す特徴としてキック分布を考える。キック分布は試合ログから抽出される。キック分布とは試合中にプレイヤーがキックした位置の集合である。キックした位置にボールの移動量を重みとして割り当てる。抽出するキックはパス、ドリブルのみとする。ボールがフィールドの外に出たキックや、相手チームにインターセプトされたパスはキックに含めない。ボールをキックしたプレイヤーの位置( $x$ 座標,  $y$ 座標)をベクトル $p_i$ とし、その重み $w_{p_i}$ をそのキックによってボールが動いた距離とする。Figure 3はopuSCOM対UvA\_Trilearnの試合ログから得られたopuSCOMのキック分布である。Figure 3において、赤い棒の座標はプレイヤーがキックした位置を示し、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。

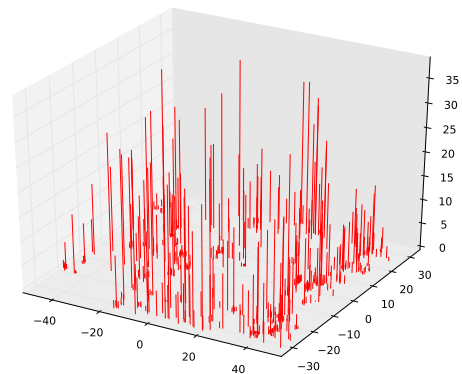


図 3: kick distribution that is obtained from a game between opuSCOM and UvA\_Trilearn

#### 3.2 Earth Mover's Distance

本論文では、キック分布間の距離をEarth Movers's Distance(EMD)を使って表す。EMDは分布間の距離を表すものであり、類似画像検索や類似音楽検索、文書分類[5]などの分野で用いられている。EMDは分布間の距離の計算を輸送問題として定式化し、一方の分布を各場所における供給量、他方の分布を需要量として最小輸送コストを分布間の距離と定義する。輸送問題とは、複数の供給地と需要地があり、需要を満たすように供給地から需要地に

輸送を行うときの最小コストを求める問題である．EMD を求める際，分布は重み付き集合として表現される．一方の分布  $P$  を集合として表現すると， $P = \{(p_1, w_{p_1}), \dots, (p_m, w_{p_m})\}$  となる．分布  $P$  は  $m$  個の特徴量で表現されており， $p_i$  は特徴量ベクトル， $w_{p_i}$  はその特徴量に対する重みである．同様に，もう一方の分布  $Q$  も集合として表現すると， $Q = \{(q_1, w_{q_1}), \dots, (q_m, w_{q_m})\}$  となる．EMD は，2つの分布の特徴量の数が異なっている場合でも計算が可能であるという特徴を持っている． $p_i$  と  $q_j$  の距離を  $d_{ij}$  とし，全特徴間の距離を  $D = [d_{ij}]$  とする． $p_i$  から  $q_j$  への輸送量を  $f_{ij}$  とすると，全輸送量は  $F = [f_{ij}]$  となる．ここで，式 (1) に示すコスト関数を最小とする輸送量  $F^*$  を求め，EMD を計算する．

$$W = \sum_{i=1}^m \sum_{j=1}^n d_{ij} f_{ij} \quad (1)$$

上記のコスト関数を最小化する際，以下の制約条件を満たす必要がある．

$$f_{ij} \geq 0 \quad (1 \leq i \leq m, 1 \leq j \leq n) \quad (2)$$

$$\sum_{j=1}^n f_{ij} \leq w_{p_i} \quad (1 \leq i \leq m) \quad (3)$$

$$\sum_{i=1}^m f_{ij} \leq w_{q_j} \quad (1 \leq j \leq n) \quad (4)$$

$$\sum_{i=1}^m \sum_{j=1}^n f_{ij} = \min\left(\sum_{i=1}^m w_{p_i}, \sum_{j=1}^n w_{q_j}\right) \quad (5)$$

式 (2) は供給地から需要地への輸送量が正であることを示し，一方通行であることを表している．式 (3) は輸送元である  $p_i$  の重み以上に輸送できないことを表す．式 (4) は輸送先である  $q_j$  の重み以上に受け入れることができないことを表す．式 (5) は総輸送量の上限を示し，それは輸送元または輸送先の総和の小さい方に制限されることを表す．

以上の制約条件の下で求められた最適な輸送量  $F^*$  を用いて，分布  $P, Q$  間の EMD を以下のように求める．

$$\text{EMD}(P, Q) = \frac{\sum_{i=1}^m \sum_{j=1}^n d_{ij} f_{ij}^*}{\sum_{i=1}^m \sum_{j=1}^n f_{ij}^*} \quad (6)$$

コスト関数は輸送元もしくは輸送先の重みの総和に依存するため，最適なコスト関数  $W$  を EMD として使用しない．正規化することによってその影響を取り除くことができる．

### 3.3 クラスタリングの手順

キック分布の分類には階層的クラスタリングを用いる．キック分布の分類手順を以下に記す．

- Step 1: 全てのキック分布の組み合わせに対して EMD を計算
  - Step 2: それぞれのキック分布がクラスタであるとする
  - Step 3: 全ての組み合わせに対して，クラスタ間の距離を計算
  - Step 4: 最も距離が小さい2つのクラスタを併合
  - Step 5: クラスタが1つになれば終了
- クラスタが2つ以上ある場合は Step 3 へ

クラスタリング終了後，デンドログラムを作成する．

### 3.4 勝敗予測

3.3 節のクラスタリング結果を用いて，試合の勝敗を予測する．まず，各クラスタごとに勝敗予測対象のチームが勝利した数を調べる．勝利数が過半数を占める場合は「勝利」，同数の場合は「不明」，それ以外を「敗北」とクラスタにラベル付けを行う．次に，勝敗予測対象の試合ログの前半戦のキック分布を抽出する．抽出したキック分布と，クラスタリングにより得られた各クラスタとの距離を計算する．キック分布とクラスタとの距離は群平均法により求める．距離が最も近いクラスタにそのキック分布が属するものとする．キック分布が属するクラスタのラベルを予測結果とする．

## 4 数値実験

提案手法の有効性を確認するため，実際の試合ログを用いて分類実験を行う．また，分類結果を用いて前半戦の試合ログから勝敗予測を行う．

### 4.1 キック分布のクラスタリング

UvA\_Trilearn(2005)[6]，BrainStomers(2009)[7]，HELIOS(2014)[8]，WrightEagle(2014)[9] の4チームを opuSCOM(2014) とそれぞれ10試合ずつ対戦させ，試合ログを作成する．上記4チームは異なるベースであり，戦略がお互いに異なっていると考えられるために選択した．作成した40試合分のログを実験に用いる．opuSCOM は研究室で開発を進めているチームである．試合ログから各キックの  $x$  座標， $y$  座標，キックによってボールが動いた距離を抽出し，キック分布を作成する．全てのキック分布の組み合わせの距離を EMD によって求める．EMD によって求めた分布間の距離を用いて，階層的クラスタリングを行う．

## 4.2 勝敗予測

試合ログから前半戦におけるキック分布を抽出する．抽出したキック分布と，クラスタリングにより得られたクラスタとの距離を計算する．距離が最も近いクラスタにそのキック分布が属するものとする．クラスタリング結果を適切であると思われるクラスタ数に分割し，属している試合ログの勝敗からラベル付けを行う．キック分布が属するクラスタのラベルが試合結果と一致しているかどうかで勝敗予測の精度を調査する．

## 5 実験結果

### 5.1 キック分布

UvA\_Trilearn(2005), BrainStomers(2009), HELIOS(2014), WrightEagle(2014) の 4 チームを opuSCOM(2014) とそれぞれ 10 試合ずつ対戦させ，試合ログを作成する．作成した 40 試合分のログを実験に用いる．クラスタリングに用いた，UvA\_Trilearn, BrainStomers, HELIOS, WrightEagle の 4 チームと opuSCOM との試合の勝敗を Table 1 に示す．

表 1: Game results with UvA\_Trilearn, BrainStomers, HELIOS and WrightEagle

対戦チーム	Win	Draw	Lose
UvA_Trilearn	10	0	0
BrainStomers	7	1	2
WrightEagle	0	0	10
HELIOS	0	0	10

### 5.2 相手チームのキック分布の分類

試合ログからキック分布を作成した．相手チームのキック分布を階層的クラスタリングした結果を Figure 4 に示す．

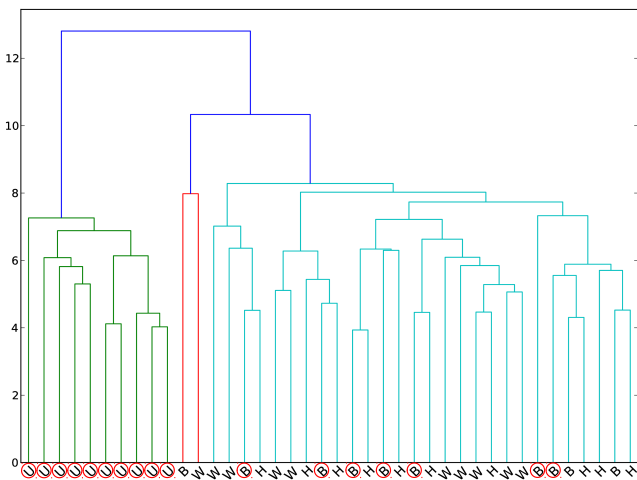


図 4: Clustering result for kick distributions of opponents

Figure 4 において，U が UvA\_Trilearn，B が BrainStomers，W が WrightEagle，H が HELIOS のキック分

布を示している．赤い丸で囲まれたデータは opuSCOM が勝った試合を示している．2 つのクラスタに分類したとき，左側のクラスタが UvA\_Trilearn のデータだけであることが読み取れる．このことから UvA\_Trilearn のデータを分類できたことがわかる．

### 5.3 opuSCOM のキック分布の分類

次に，試合ログから opuSCOM のキック分布を作成した．opuSCOM のキック分布を階層的クラスタリングした結果を Figure 5 に示す．

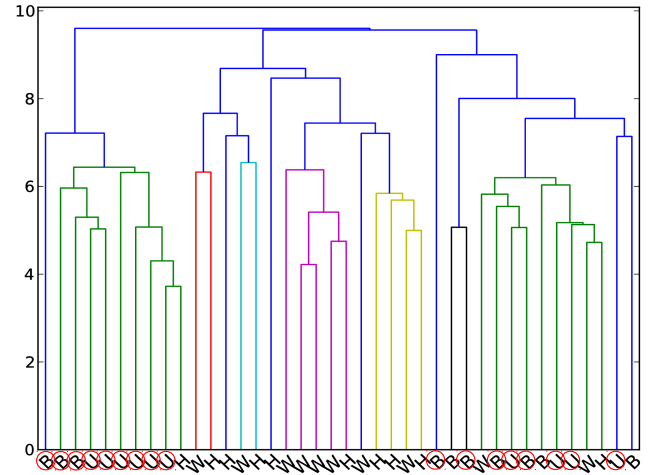


図 5: Clustering result for kick distributions of opuSCOM

Figure 5 において，U が UvA\_Trilearn，B が BrainStomers，W が WrightEagle，H が HELIOS と対戦したときの opuSCOM のキック分布を示している．赤い丸で囲まれたデータは opuSCOM が勝った試合を示している．3 つのクラスタに分類したとき，中央のクラスタに負けた試合，左右のクラスタに勝った試合というように分類できているように読み取れる．

### 5.4 両チームのキック分布の分類

試合のログから 1 試合毎の両チームのキックを区別せずに，キック分布を作成した．両チームのキック分布を階層的クラスタリングした結果を Figure 6 に示す．

Figure 6 において，U が UvA\_Trilearn，B が BrainStomers，W が WrightEagle，H が HELIOS と対戦したときの両チームのキック分布を示している．赤い丸で囲まれたデータは opuSCOM が勝った試合を示している．2 つのクラスタに分類したとき，自チームのキック分布でクラスタリングを行ったときよりも上手く勝敗別に分類できているように読み取れる．

以上の実験結果より，特定のチームや戦術で分類する場合は相手チームのキック分布をクラスタリングすると

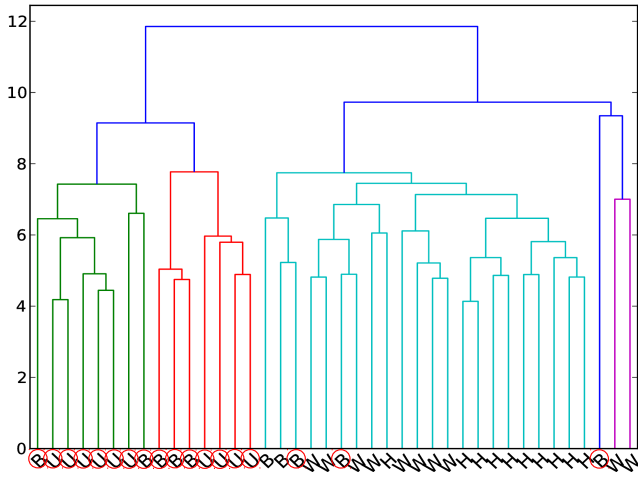


図 6: Clustering result for kick distributions of opuSCOM and opponents

有効であることが分かる。また、自チームとの勝敗で分類する場合は両チームのキック分布をクラスタリングすると有効であることが分かる。

### 5.5 勝敗予測

勝敗予測に用いた、UvA\_Trilearn, BrainStomers, HELIOS, WrightEagle の 4 チームと opuSCOM との試合の勝敗を Table 2 に示す。

表 2: Game results with UvA\_Trilearn, BrainStormers, HELIOS and WrightEagle

対戦チーム	Win	Draw	Lose
UvA_Trilearn	9	0	1
BrainStomers	10	0	0
WrightEagle	1	0	9
HELIOS	0	0	10

次に、クラスタリング結果にラベル付けを行った。相手チームのキック分布を用いたクラスタリングのラベル、opuSCOM のキック分布を用いたクラスタリングのラベル、両チームのキック分布を用いたクラスタリングのラベルをそれぞれ Figure 7, 8, 9 に示す。デンドログラムから適切であると考えられるクラスタ数に分割し、属している試合ログの勝敗からラベル付けを行った。

勝敗予測の結果と実際の試合結果を比較し、それぞれの対戦チームとキック分布についての正答数を Table 3 に示す。

表 3: The number of correct

対戦チーム	opuSCOM	相手チーム	両チーム
UvA_Trilearn	4	10	10
BrainStomers	1	8	4
WrightEagle	9	9	9
HELIOS	10	0	10
平均	6	6.75	8.25

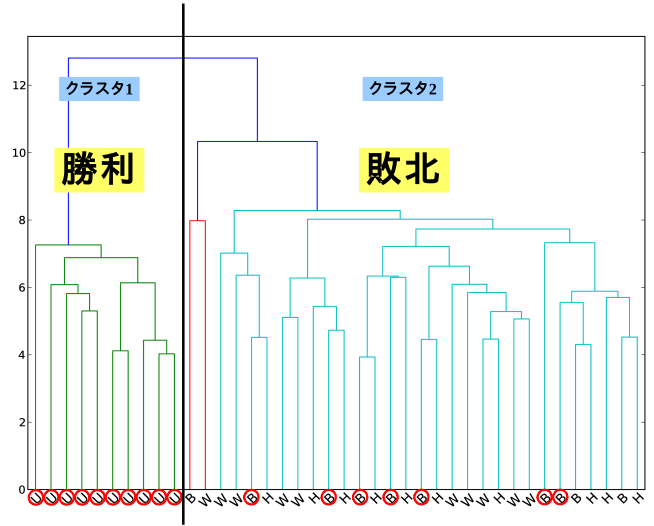


図 7: Labels of clusters : opponent team's kick distribution

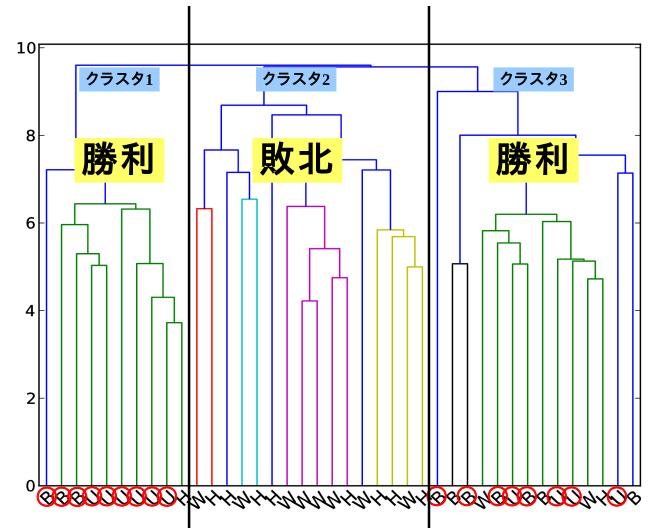


図 8: Labels of clusters : our team's kick distribution

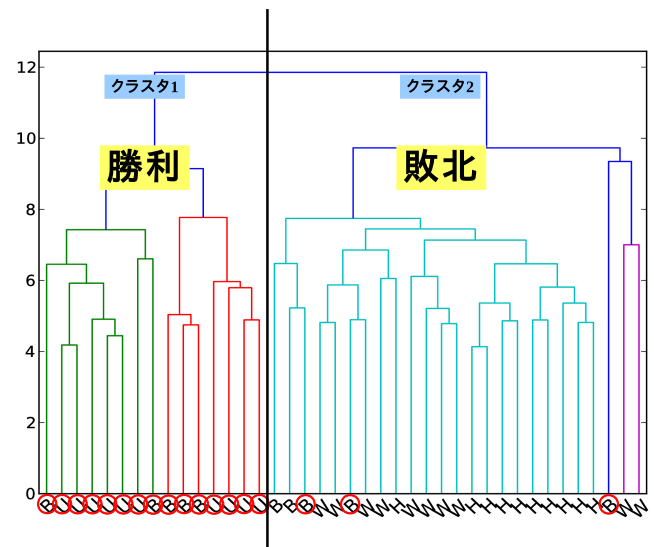


図 9: Labels of clusters : both team's kick distribution

Table 3より,両チームのキック分布から勝敗予測を行ったとき精度が最も良いことがわかる.また,opuSCOMのキック分布,相手チームのキック分布を用いた勝敗予測では,正答数が少ない場合があることが読み取れる.相手チームのキック分布を用いた勝敗予測では,UvA\_Trilearnだと勝利,その他のチームだと敗北というような予測結果となり,正答数が少なくなったのではないかと考えられる.

## 6 おわりに

本論文では,試合ログをキック分布として表現し,EMDを類似度として用いることにより,試合ログをクラスタに分類した.また,クラスタリング結果を用いて前半戦の試合ログから勝敗を予測した.数値実験では,実際に試合ログ間の類似度を計算し,特定のチームや勝敗で分類できることを示した.クラスタリング結果を用いて,前半戦の試合ログから勝敗を予測できることを示した.今後の課題として,分類や勝敗予測の精度を上げることや,試合中に相性の良し悪しを判断できるようにチームに組み込むこと,キック分布を勝敗予測以外に活用することなどが挙げられる.

## 参考文献

- [1] Hiroaki Kitano, Minoru Asada, Yasuo Kuniyoshi, Itsuki Noda, Eiichi Osawa and Hitoshi Matsubara, “ RoboCup: A Challenge Problem for AI ”, *AIMagazine*, Vol.18, No.1, pp.73-85(1997).
- [2] Thomas Gabel, Martin Riedmiller, “ On Progress in RoboCup: The Simulation League Showcase ”, *The 14th RoboCup 2010 Symposium*, pp.36-47, Springer Berlin Heidelberg(2010).
- [3] Pedro Abreu, João Moreira, Israel Costa, Daniel Castelhão, Luis Reis, Júlio Garganta, “ Human Versus Virtual Robotics Soccer: A Technical Analysis ”, *European Journal of Sport Science* 12(1), pp.26-35, Taylor & Francis(2011)
- [4] Y.Rubner, C.Tomasi and L.J.guibas, “ The earth mover’s distance as a metric for image retrieval ”, *International Journal of Computer Vision*, 40(2), pp.99-121(2000)
- [5] 柳本豪一,大松繁, “ Earth Mover’s Distance を用いたテキスト分類 ”, *人工知能学会全国大会論文集* (2007).

- [6] Julle R. Kok and Nikos Vlassis, “ UvA Trilearn2005 Team Description Paper ”, *RoboCup2005*, CD-ROM (5 pages), Osaka, Japan(2005).
- [7] Thomas Gabel, Martin Riedmiller, “ BrainStormers 2D - Team Description 2009 ”, *RoboCup2009*, CD-ROM (6 pages), Graz, Austria(2009).
- [8] Hidehisa Akiyama, Tomoharu Nakashima, Katsuhiko Yamashita, Satoshi Mifune, “ HELIOS2014 Team Description Paper ”, *RoboCup2014*, CD-ROM (6 pages), JoãoPessoa, Brazil(2014).
- [9] Haochong Zhang, Guanghui Lu, Rongya Chen, Xiao Li and Xiaoping Chen, “ WrightEagle 2D Soccer Simulation Team Description 2014 ”, *RoboCup2014*, CD-ROM (6 pages), JoãoPessoa, Brazil(2014).

# Estimation of Players' Actions in Soccer Matches Based On Deep Autoencoder

Jorge L. Copete, Junichi Suzuki, Quan Wei, Ryo Iwaki, Nobutsuna Endo,  
Hiroki Mori, Yukie Nagai, Minoru Asada

Department of Adaptive Machine Systems, Graduate School of Engineering,  
Osaka University, Osaka, Japan  
robocup-spl@ams.eng.osaka-u.ac.jp

## Abstract

The ability to estimate future events is essential for biological systems to adapt themselves to uncertain situations efficiently. Similarly, the abilities to estimate tactical movements of opponents and to perform in an unpredictable manner against opponents in robot soccer competitions are required in the best interest of the robot players. However, there is still a need of approaches that estimate robot players' future actions while dealing with complex dynamic conditions when their behaviors have been altered because of mutual interactions. In this study, we propose a method based on a deep autoencoder to estimate actions of soccer players in an environment with multiple agents and dynamical conditions. We carried out experiments employing data from the RoboCup 2D Soccer Simulation League and showed the validity of the proposed method. Our results suggest that computational models using deep architectures may be key to developing new skills for robots using low-level representations.

## 1 Introduction

In human soccer the ability to understand the tactics of opposing teams is crucial to modifying a team's own actions, thus achieving better performance. In real soccer, for example, while a player who has an offensive role might try to move forward and kick the ball directly into the goal, a midfielder might try to pass the ball to an offensive player who is near the opponent's goal. Human players deal with these situations by estimating the future action of each player on the opposing team. In the context of RoboCup soccer competitions, therefore, learning to estimate the future actions of other players is also required in the best interest of the robot players. One of the main challenges of estimating the actions of robot soccer teams is that the actions of the players during a match depend on several complex factors, including both teams' strategies and the skills and role of each

player on both teams. Therefore, in order to estimate the behavioral patterns of multiple agents in a dynamic environment, we need to employ a learning process that considers all of these factors simultaneously.

Among previous works, Visser et al. [1] proposed classifying the behaviors of the opposing team by inferring the behaviors that players can perform in advance. Bowling et al. [2] proposed adapting to the opponent by observing our own team's effectiveness rather than observing the opponent's behavior. For that purpose, they used multiple team plans that are appropriate for different opponents and situations, and rewarded the plans during the match based on the actual results. Recently Trevizan and Veloso [3] proposed classifying opponents' strategies by studying the similarity of the strategies of two teams by employing feature vectors of the distances between players and the ball. Yasui et al. [4] proposed a dissimilarity function that shows the difference between opponents' deployments at two different times, and extends it to the difference between those of two different time intervals. However, these initial approaches, which classified players' behaviors by finding a strategy from a set of known strategies or learned a strategy assuming that a team's own behaviors are independent of the opposing teams' behaviors, can be regarded as top-down oriented and thus are not sufficiently robust to deal with complex dynamic conditions when the players' behaviors have been altered because of their mutual interactions. Therefore, there is still a need of approaches employing bottom-up procedures to provide low-level behavioral features (e.g., anticipation of players' motions) that are needed at higher-level representations to model teams' strategies.

In this study, we proposed a method to estimate players' actions that does not rely on a priori knowledge of teams' actions or strategies. Specifically, we proposed a method to estimate the trajectory of players during ball motion and the direction of the ball after being kicked. We used dynamical prediction (that is, a mathematical model describing a physical system that changed over time) to estimate the future position of both players and ball. This estimation is desired because both the players



and the ball move; they are dynamic. In our model, we employed deep neural networks to represent the complex dynamics of temporal sequences of player positions and a one-hidden-layer neural network to estimate the orientation of the ball movement. We applied our model to data from the RoboCup 2D Soccer Simulation League to assess our prediction method. The rest of the paper is organized as follows: In section 2, we explain our proposed method. The experimental settings for the RoboCup 2D Soccer simulator are explained in section 3. Section 4 presents the results of our experiments. In section 5, our conclusions are given and future research possibilities are discussed.

## 2 Dynamical estimation of players' actions

In soccer competitions, not only are teams without strategies weak opponents to play against, but also teams using strategies based on repetitive actions are predictable and thus possible to defeat. During soccer matches players perform actions such as running in certain trajectories, passing the ball, keeping the ball as far away as possible from the opponents, and kicking the ball to the opposing goal. Additionally, if the opposing team is in possession of the ball, players are expected to attempt to regain control of the match. Therefore, the ability to estimate the trajectory of other players and the ball is critical for soccer players. In this investigation, we considered two basic actions of soccer players —moving in the field and kicking the ball—and proposed a method to estimate the trajectory of players during ball motion and the direction of the ball after being kicked. To do this, we used the following two assumptions:

- The trajectory of players can be anticipated by learning to estimate future states based on the past states
- The trajectory of a kicked ball can be predicted by learning to estimate future states from the current state

Figure 1 shows a schematic diagram of the method that we proposed. We used a deep autoencoder as a temporal sequence predictor to account for the first assumption, and a feed-forward neural network for the second one. Players are not initially assigned any particular role or strategy; instead their action patterns are learned from raw data consisting of players' positions and ball position. In our implementation, the direction of players' motions and ball's motions are calculated from the estimated positions. We used motion direction but not position because the neural networks were expected to learn to generalize motion patterns but not to calculate exact positions.

### 2.1 Estimation of actions under dynamic conditions based on deep neural networks

In order to account for the dynamic action of players moving in the field, we proposed using deep neural networks (DNNs) which have the ability to represent com-

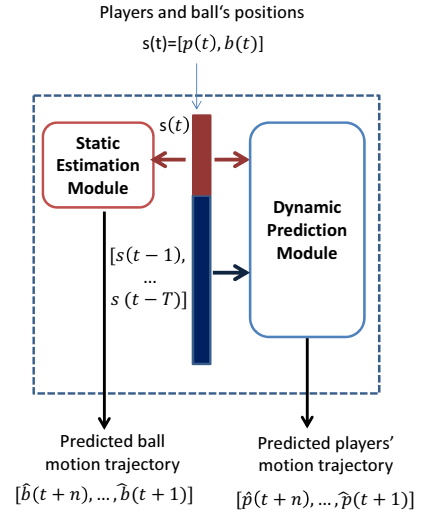


Figure 1: Estimation of static and dynamical actions: the last  $T$  steps for training a deep neural network, and the last step for training a feed-forward neural network.

plex functions. DNNs have been shown to outperform state-of-the-art machine learning algorithms in many applications [5], [6]. In particular, we referred to a previous work by Noda et al. [7], who reported the advantage of employing autoencoders based on deep neural networks for predicting temporal data sequences. In this approach, two deep neural networks are stacked in a mirrored structure, and the output layer of the neural network is connected in a closed-loop manner to the input layer to generate the prediction of future steps, as shown in Figure 2. The input-output mappings of the network are defined as follows:

$$\hat{u}_t = f(r_t), \quad (1)$$

$$\hat{r}_t = f^{-1}(u_t), \quad (2)$$

where  $\mathbf{r}(t)$ ,  $\mathbf{u}(t)$ , and  $\hat{\mathbf{r}}(t)$  are the vectors representing the input data, the corresponding vector feature, and the reconstructed data, respectively.  $\mathbf{f}(\cdot)$  represents the transformation mapping from the input layer to the central hidden layer of the network and  $\mathbf{f}^{-1}(\cdot)$  represents the transformation mapping from the central hidden layer to the output layer.

When employing deep autoencoders to make predictions, the input data is fed into the neural network as a contiguous segment of  $T$  steps. The inputs corresponding to  $T_{in}$  steps ( $T_{in} < T$ ) are filled with previous input data, and the rest of the inputs corresponding to  $T - T_{in}$  steps are filled with the outputs from the closed-loop data, as shown in Figure 3. In this study, the input data segment contains the positions of the players and the ball for  $T$  steps in Cartesian coordinates. Hence, for an input size of  $T$  where  $T_{in}$  corresponds to the size of previous input data, the input to the network at time  $t$  is defined as follows:

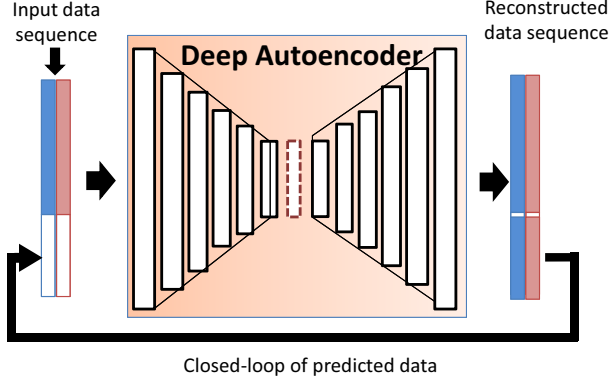


Figure 2: Deep autoencoder for temporal sequence prediction.

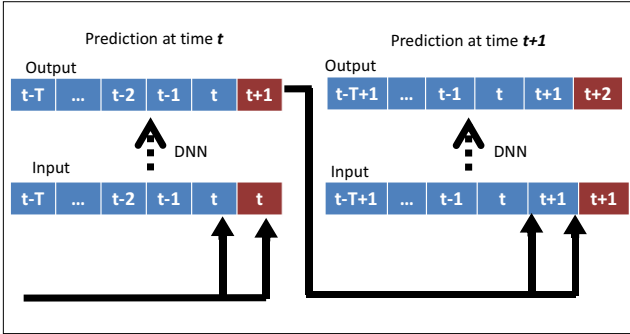


Figure 3: Scheme for recursive input-output loop.

$$s(t) = (b_{t_1}, \hat{p}_{t_1}, b_{t_1}, \hat{p}_{t_1}), \quad (3)$$

$$t - T_{in} + 1 < t_1 < t, \quad (4)$$

$$t + 1 < t_2 < t + (T - T_{in}), \quad (5)$$

where  $\mathbf{b}(t)$  and  $\mathbf{p}(t)$  are the vectors representing the ball's position and the players' positions, respectively.

## 2.2 Estimation of actions under static conditions based on a one-hidden-layer neural network

In this module the current state corresponds to the positions of the players when a player kicks the ball, and the future state is the position of the ball when it stops. We employ a one-hidden-layer neural network that is trained to estimate the direction of the ball, as shown in Figure 4. The inputs of the network are the position of the ball and the players in Cartesian coordinates at time  $t$ , and the output is the estimated position of the ball  $w$  steps ahead:

$$i_t = f(p_t, b_t), \quad (6)$$

$$\hat{b}_{t+1} = f(i_t), \quad (7)$$

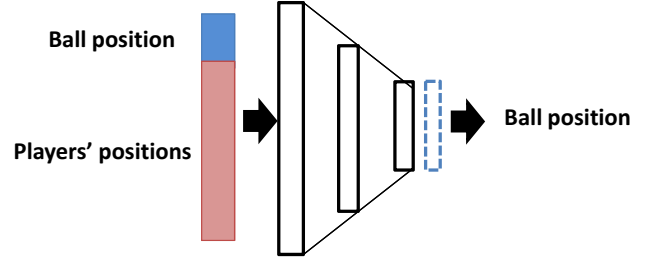


Figure 4: One-hidden-layer neural network for ball direction prediction.

where  $\mathbf{b}(t)$ ,  $\mathbf{p}(t)$ , and  $\mathbf{i}(t)$  are the vectors representing the ball's position, the players' positions, and the input data, respectively.

## 3 Experimental settings

For the static condition, we employed a one-hidden-layer neural network using as input data the position of the ball and the players at a frame before the player kicked the ball, and as output data the position of the ball once it stopped. The size of the input layer was 46 neurons, the hidden layer was 46 neurons, and the output layer was 2 neurons. For the dynamical condition, we employed a deep neural network with 8 hidden layers, composed of an encoder and a decoder whose structures are mirror images of each other. The input and output layers each contain 230 neurons. The number of neurons in the hidden encoder layers is 250, 150, 80, and 30. The decoder layers contain 30, 80, 150, and 250 neurons. The activation functions are linear functions for the hidden layers and logistic functions for the output layer. For this experiment, we implemented the proposed modules in the Python library, Theano, based on available implementations for DNNs [8].

The data we collected from the simulator of the RoboCup 2D Soccer Simulation League included the field position of 22 players (11 players for each team) and the position of the ball. We ran the software using the HELIObase and WEBbase teams. The log data of the matches was recorded and processed to obtain the positions of the ball and the players in Cartesian coordinates  $(x, y)$  at each frame of the match. The length of each match was about 3000 frames. The position data  $(x, y)$  was normalized between 0 and 1 considering the original size of the field, which was  $40 \times 105$  pixels. We recorded the log data of ten matches to be used as training data, and of one match for test data.

To train the deep neural network we used the entire raw data set, and for the one-hidden-layer neural network, we extracted the sets of frames of when players kicked the ball and of when the ball stopped moving. For testing, we extracted from the raw data those segments of the match in which the ball was detected as

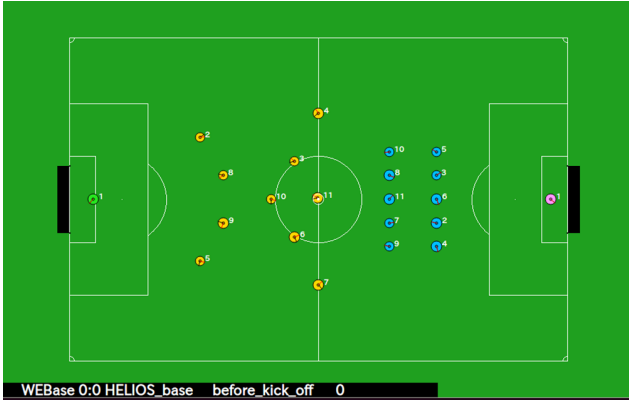


Figure 5: RoboCup 2D Soccer Simulation League.

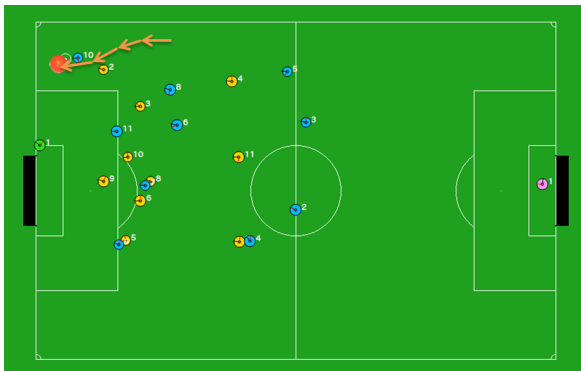


Figure 6: Example of ball motion that meets the requirements of a dynamic condition.

moving. An example of this is shown in Figure 6, where the motion of the ball is represented by arrows. That is, we focused our analysis on dynamical conditions, assuming that the motion of the ball caused players to move to other positions. For our purposes, the ball must have moved at an average speed of greater than 4 pixels per frame (field size is  $40 \times 105$  pixels) to be considered in motion. To calculate the average speed and other results during training and testing, we used a time window of 5 frames, with the target frame included. The deep neural network was trained employing the stochastic gradient descent method, and the one-hidden-layer network was trained using the back-propagation method.

## 4 Results

### 4.1 Evaluations of estimation of ball direction

We conducted the first experiment to estimate ball direction using the extracted data corresponding to the positions of the ball and the players when a player was close to kicking the ball. The results of the experiment are shown in Figure 7 and Figure 8. Figure 7 shows examples of estimation of ball motion direction for several kicking players in different positions in the field. The red and blue arrows indicate the actual and the esti-

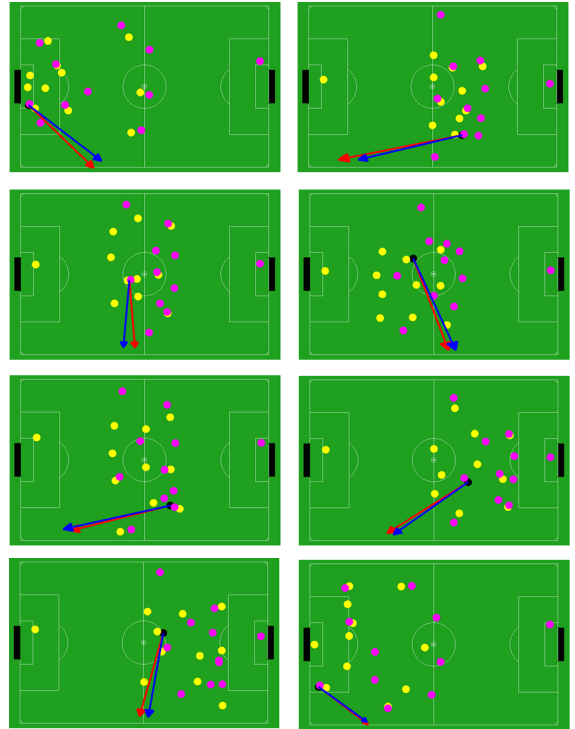


Figure 7: Examples of ball direction estimations for several players in different positions in the field. Red and blue arrows indicate the actual and the estimated directions of the ball, respectively.

imated directions of the ball, respectively. For analysis purposes, we decided that an estimation would be considered successful if the maximum difference between the estimated direction and the actual one was 15 degrees. Figure 8 shows the number of successful and failed estimations of ball direction for each player. For this experiment, the average successful estimation rate was 84.1%. The results demonstrate that our model was effective at estimating the future ball direction under several conditions. These results suggest that the kicking actions of players of both teams of the RoboCup 2D Soccer Simulation League are highly predictable and could be used by both teams to improve their ability to anticipate their opponents' shots.

### 4.2 Evaluation of estimation of direction of players' movements

We conducted the second experiment using the positions of the ball and the players. We considered two cases when analyzing the estimation results: moving players and stationary players. Players were only considered to have moved if the distance between the median of the past positions and the median of the estimated positions was greater than the motion threshold, which was arbitrarily set to 3.7 pixels. We defined eight possible directions that a player could move toward. We set 45 degrees as the maximum allowable difference between the estimated direction and the actual direction to consider an estimation as successful. Otherwise, the estimation was

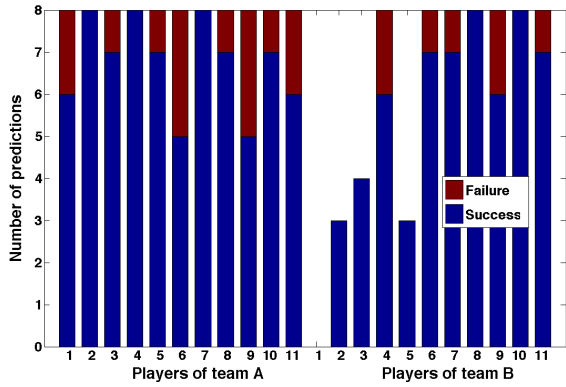


Figure 8: Success rates of ball direction estimation for players kicking the ball.

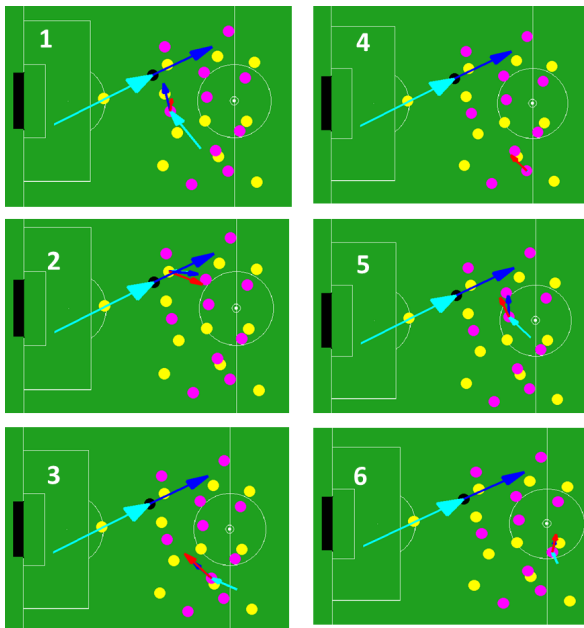


Figure 9: Examples of correct estimations by several players in different positions in the field. The yellow and pink points represent the players' positions for each team, and the black point represents the ball. The cyan arrow pointing to the ball indicates the previous direction of the ball and the cyan arrow pointing to a player indicates the previous direction of that player. The blue arrows originating at the ball and the players indicate the actual future direction of the ball and of the players. The red arrow indicates the estimated direction of the player.

considered a failure.

Figure 9 and Figure 10 show examples of correct estimations and failed estimations of the direction of the motion of several players for the same frame.

In Figure 9 we observe examples of estimation results for six different players using our system, including cases in which the direction before and after the estimation did not change significantly, as well as cases in which play-

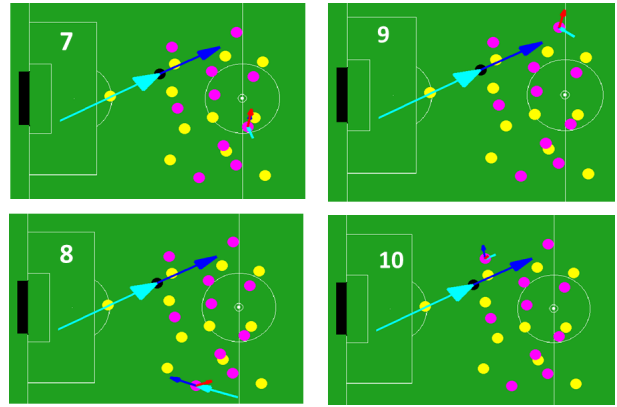


Figure 10: Examples of incorrect estimations by several players in different positions in the field. The interpretation of this figure is identical to that of Figure 9

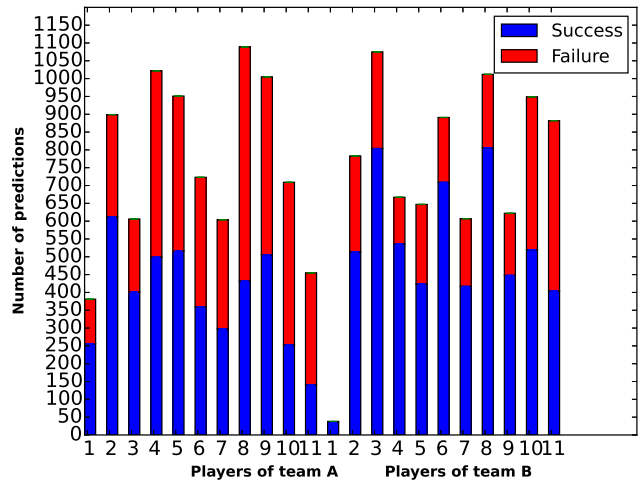


Figure 11: Results of motion direction estimation for each player.

ers were previously stationary and then started to move (image 4), or were moving in one direction and then changed direction (image 2). Figure 10 shows examples of failed estimations. These included cases where the actual motion of the player was not significant enough to be considered moving, although the estimated direction was correct (image 7), where the players stayed practically stationary, so the estimations were considered failures, but the magnitude of the estimation was small (image 9 and 10), or where the estimated direction of the player differed from the actual, but the estimated direction followed the same direction as the ball (image 8). Figure 11 shows that, in general our system was able to effectively infer an average of 50.6% and 69.9% of the future directions of players of team A and B, respectively. This means that our approach was able to learn and generalize motion patterns from the raw data and then apply that knowledge to new scenarios. Addition-

ally, we see the correspondence between the fact that team B (HELIOBase, with 8 goals) had better performance than team A (WEBBase, with 0 goals) during the matches, and the fact that the estimation by the players on team B was significantly higher than the number of correct estimations by the players of team A. It may suggest that the actions of the players on team B include strategic patterns and thus are more predictable. To summarize, these results indicate that our model was able under several conditions to learn motion patterns of the players and then apply that knowledge to infer future states.

## 5 Conclusions

In this study we proposed a method to estimate the action of soccer players and conducted experiments to measure this predictability using data from the RoboCup 2D Soccer Simulation League. The experimental results demonstrate that the future movement directions of the players and of the ball can be estimated successfully to a great extent; consequently, our method was proven to be able to estimate low-level predicted actions that could be adopted to represent teams' strategies. We are considering that task (i.e., adopting low-level actions for strategies' representation) as a future work for this study. Further work is required to classify the roles of the players by exploiting the capability of DNNs to learn high-level representations from raw features, to assess the ability to estimate players' trajectories for different sizes of the time window, and to validate our approach using data from the RoboCup Standard Platform and Middle Size leagues. Finally, we expect that introducing approaches based on DNNs like ours that estimate actions in a bottom-up manner opens the door to developing new skills for robots in RoboCup competitions.

## 6 Acknowledgements

This work was partially supported by the JSPS Grant-in-Aid for Specially Promoted Research (24000012).

## References

- [1] Visser, Ubbo, and Hans-Georg Weland. Using Online Learning to Analyze the Opponents Behavior, RoboCup 2002: Robot Soccer World Cup VI. Springer Berlin Heidelberg, 2003.
- [2] Bowling, Michael, Brett Browning, and Manuela Veloso. Plays as Effective Multiagent Plans Enabling Opponent-Adaptive Play Selection, Proceedings of the Fourteenth International Conference on Automated Planning and Scheduling, 2004.
- [3] Trevizan, Felipe W., and Manuela M. Veloso. Learning Opponents Strategies in the RoboCup Small Size League, International Conference on Autonomous Agents and Multiagent Systems, Springer, 2010.
- [4] Yasui, Kotaro, Kunikazu Kobayashi, Kazuhito Murakami, and Tadashi Naruse. Analyzing and Learning an Opponent's Strategies in the RoboCup Small Size League, RoboCup 2013: Robot World Cup XVII, Lecture Notes in Computer Science, Volume 8371, 2014, pp 159-170.
- [5] Hinton, Geoffrey E., and Ruslan R. Salakhutdinov. Reducing the Dimensionality of Data With Neural Networks, Science 313.5786, 2006, pp 504-507.
- [6] Bengio, Yoshua. Learning Deep Architectures for AI, Foundations and Trends in Machine Learning 2.1, 2009, pp 1-127.
- [7] Noda, Kuniaki, et al. Multimodal Integration Learning of Object Manipulation Behaviors Using Deep Neural Networks, 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems IEEE, 2013.
- [8] Chapelle, Olivier, and Dumitru Erhan. Improved Preconditioner for Hessian Free Optimization, NIPS Workshop on Deep Learning and Unsupervised Feature Learning, 2011.

# Implementation of Acoustic Communication in RoboCup SPL.

Author: Wen Chun Chen, Jia Jun Tsai, Dr. N.M.Mayer

Dept. of Electrical Engineering and AIM-HI, National Chung Cheng University, Taiwan  
b1772508@hotmail.com

## Abstract

This paper discusses how sound may be used to find out the relative position of one Aldebaran Nao robot to another. This functionality can be extended to also transfer data directly from robot to robot by acoustic transmission.

## 1. Introduction

In a robot soccer game, we can see our robots as a team of agents. How these agents cooperate with each other is a big issue because soccer is a cooperative and interactive game. How well the robots cooperate will give a significant advantage in winning the game. Here, we want to use acoustic communication to improve the accuracy of self-localization, exchange the exact relative positions of team mates and some important messages. According to the 2013 RoboCup soccer rule, the robots can use any type communication. The most common type of communication is to use the wireless network. During competition, the wireless network is blocked because the networks of various participants interfere with each other. So we try to use acoustic communication instead. So in this work we discuss probability of transmission data acoustically and at the same time measure the reality angle letter sender and review.



Fig1. Robot Setting

## 2.1. Cross-Correlation

### Function

In digital signal processing, cross-correlation is a measure of similarity of two waveforms as a function of a time-lag applied to one of them. It is commonly used for searching a long signal for a shorter, known feature. It has applications in pattern recognition, single particle analysis, electron tomography, averaging, cryptanalysis, and neurophysiology.[1] For continuous functions  $f$  and  $g$ , the cross-correlation is defined as:

$$(f^* g)(T) = \int_{-\infty}^{+\infty} f^*(t)g(t+T)dt \quad (2.1)$$

where  $f^*$  denotes the complex conjugate of  $f$  and  $T$  is the time lag.

Similarly, for discrete functions, the cross-correlation is defined as:

$$(f^* g)[n] = \sum_{m=-\infty}^{\infty} f^*[m]g[m+n] \quad (2.2)$$

Cross-correlations are useful for determining the time delay between two signals. In the

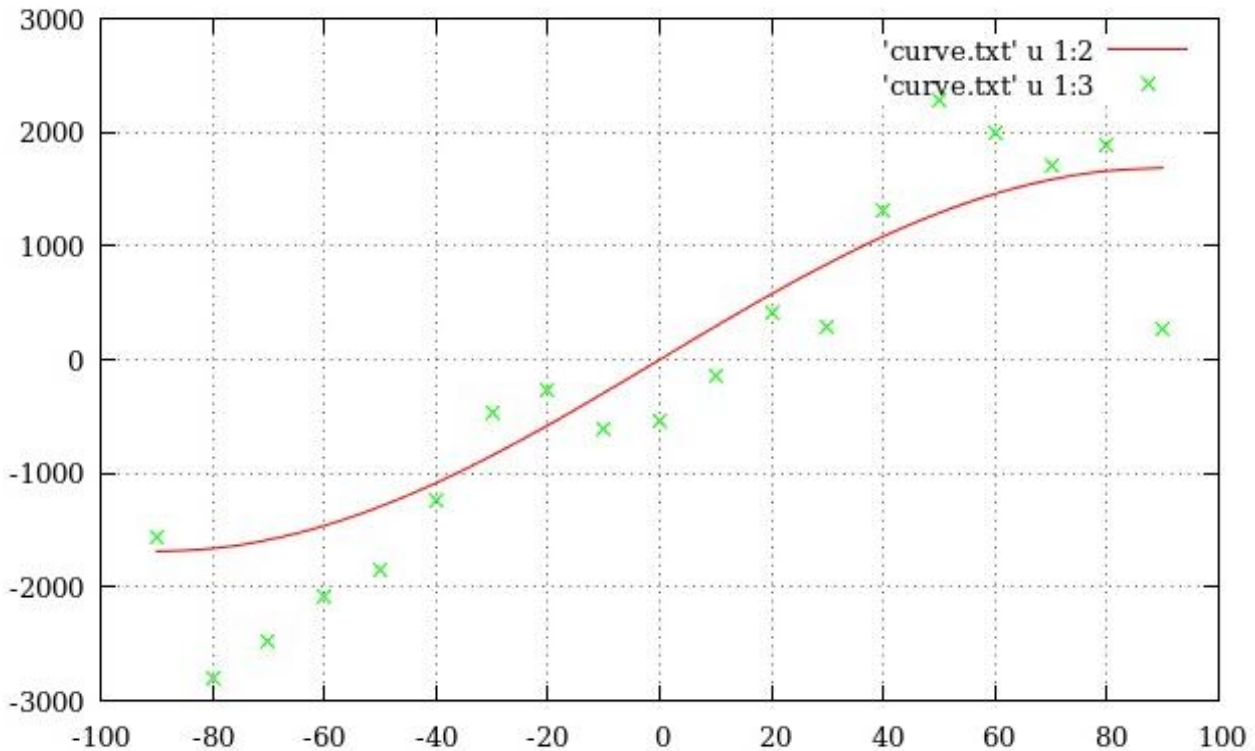


Fig. 2 visualizes the result by using picture. The red curve means the theoretical lags for every angle, and the green points stand for the measurement lags.

following the process of detecting a signal using cross correlation is described: A specific signal pattern is assigned to each sender, and all receivers know these patterns. After a sender "A" has sent his signal pattern, a receiver "B" calculates the cross correlation function of his input signal and A's specific signal pattern by equation(2.2). B now assumes, that A's signal arrived at time:

$$T = \arg \max ((f * g)(t)) \quad (2.3)$$

## 2.2.Goertzel algorithm

The Goertzel algorithm is a Digital Signal Processing technique that provides a mean for efficient evaluation of the Discrete Fourier Transform (DFT). The algorithm was first described by Gerald Goertzel in 1958 [2].

Like the DFT, the Goertzel algorithm

component from a discrete signal[3][4][5]. Unlike direct DFT calculations, the Goertzel algorithm applies a single real-valued coefficient at each iteration, using real-valued arithmetic for real-valued input sequences. For covering a full spectrum, the Goertzel algorithm has a higher order of complexity than Fast Fourier Transform (FFT) algorithms; but for computing a small number of selected frequency components, it is more numerically efficient.

## 3.Result

All the results are recorded in Fig.2. The theoretical lags mean the theoretical time difference of sound arrived to the left and right ear of the robot, and its absolute is maximum equals to 1686 frames, which is equals to 0.03498 second. If the measured lags exceed this value, then it can not be converted into angle. We can see that when

the sound source direction is bigger than  $\pm 40$  degree, our measured lags will exceed the limitation and can't convert into angles. It may be caused by the refraction and diffraction of the sound, since when the source is bigger than  $\pm 40$  degree, the sound wave may be hindered by other part of the head and can not directly hit the microphone at the further side. However, when the sound source direction is below  $\pm 40$  degree, the sound source angle can be correctly detected with an error about  $\pm 12.7$  degree.

## 4. Conclusion

Since the robot can distinguish what his team mate is talking correctly and knows the direction roughly, we can develop more strategies for soccer games. In the future, we should implement more messages in different combinations of the AFSK sequences, for example: the ball is here, this is the wrong goal, and kick fast, use the SSL to fast the process of some vision strategies. Also, this is a plan B to concur the situation of network communication jamming. Furthermore, the ball searching strategy can be improved with this approach since

originally the robot needs 7 seconds to turn his body around to find the ball if he does not know where is it. The robot can find the ball more quickly if he heard his team mate telling him where is the ball. In this case, our NAO robot can play the soccer game more efficiently and smartly.

## 5. References

- [1] The introduction of Cross-Correlation. <http://mathworld.wolfram.com/Cross-Correlation.html/>.
- [2] G. Goertzel. An Algorithm for the Evaluation of Finite Trigonometric Series. *The American Mathematical Monthly*, 65(1):34–35, 1958.
- [3] Chiouguey J. Chen. Modified Goertzel Algorithm in DTMF Detection Using the TMS320C80. 1996.
- [4] Gunter Schmer. DTMF Tone Generation and Detection : An Implementation Using the TMS320C54x. (May):1–19, 2000.
- [5] Semiconductor Systems Engineering, Digital Signal, and Processing Solutions. Add DTMF Generation and Decoding to DSP-  $\mu$  P Designs. 1989.



# タブレット端末で動作する Arduino 用プログラミング環境 aiBlocks の初心者向け作例集の試作と評価

光永 法明 (大阪教育大学), 吉田 凶夢 (大阪教育大学卒業), 井芹 威晴 (岸和田市立桜台中学校)  
Noriaki Mitsunaga (Osaka Kyoiku University),  
Tom Yoshida (Graduated Osaka Kyoiku University), and  
Takeharu Iseri (Sakuradai Junior High School in Kishiwada City)  
mitunaga@cc.osaka-kyoiku.ac.jp

## 概要

本論文では Android OS で動作するタブレット端末用のビジュアルプログラミング環境 aiBlocks の初心者向け作例集を試作したので報告する。aiBlocks は、マイコンボード Arduino 上で動作するプログラムを開発するビジュアルプログラミング環境である。試作した作例集を、aiBlocks を載せたタブレット、マイコンボードや電子部品と共に初心者に渡し、操作の様子を観察し評価した結果、初心者にとって楽しめる環境と作例集であることがわかった。

## 1 はじめに

最近ではスマートフォンやタブレット端末と呼ばれる、機器のほぼ全面がディスプレイとタッチパネルで構成される機器が、従来からあるパソコン以上に普及してきている。それらで動作するアプリケーションソフトは、映像や文章などのコンテンツを表示するものが多かったが、普及するにつれ techBASIC [1], Scratch Jr. [2], Pyonkee [3] といったプログラミング環境も増えている。こういった端末でプログラミング環境を提供することで、あらたにプログラミング用のハードウェア (たとえばパソコン) を用意せずに入門できるという利点がある。また、ビジュアルプログラミング環境を使うと、キーボード操作に不慣れな児童・生徒も、プログラミングを楽しむことができる [4] と言われており、ディスプレイと一体型のタッチパネルによる操作も相性がいいと考えられる。

ところでマイコン (マイクロコントローラ, マイクロコンピュータ) が広く普及し、産業機器, 電化製品だけでなく、趣味の電子工作や学校での学習にも広く部品として使われている。マイコンを使った電子回路の場合には、ハードウェア, ソフトウェアの両方についての知識が必要になる。そのため、ハードウェアのみの電子工作,

ソフトウェアのみで完結するプログラミングと比べて学習のハードルが高くなりやすい。そこで、ある程度ハードウェアを完成させたマイコンボードやモジュールを用意したり、ライブラリを充実させたプログラミング環境を用意するといった工夫がされている。

そういったマイコンボードの中で、最近では Arduino [5] がよく使われており、初心者にも勧められることも多い。Arduino の標準的なプログラミング言語は Arduino 言語 (C++ 言語を基に変更・拡張した言語) であり Windows, Mac OS X, Linux 上のプログラミング環境がある。また環境を拡張すれば、ArduBlock [6] というブロックを並べて記述するビジュアルプログラミング言語が利用できる。Arduino マイコンボード用のタブレット上でのプログラミング環境としては、Arduino 言語の利用できる ArduinoDroid [7], インタプリタ言語 iArduino の利用できる iArduinoTerminal for Android [8] などがある。それらはテキストベースのプログラミング環境であるのに対し、我々は aiBlocks を開発した [9]。aiBlocks は Android OS 用のアプリケーションソフトで、Arduino マイコンボード上で動作するプログラムを、ブロックの操作で作成できるビジュアルプログラミング環境である。完成したプログラムは、aiBlocks を必要とせず、マイコンボードだけで動作する。

本報告では aiBlocks を使った初心者向けの作例集を試作したので報告する。以下では、まず aiBlocks について紹介し、作例集の概略を述べる。そして aiBlocks と作例集を初心者が操作した様子を観察した結果から得た評価を述べる。

## 2 プログラミング環境 aiBlocks

aiBlocks [9] は Android OS 上で動作する Arduino マイコンボード用のビジュアルプログラミング環境である (図 1)。aiBlocks で作成したプログラムはインタプリタ

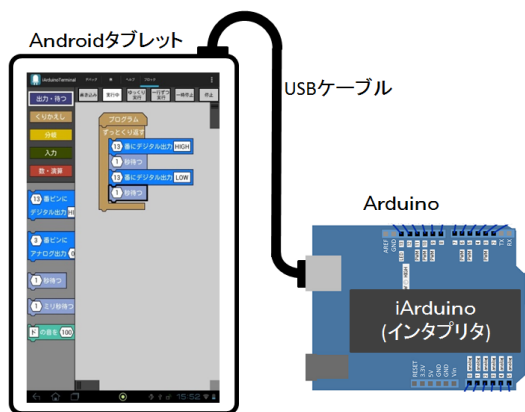


図 1: aiBlocks の動作するタブレットと Arduino は USB ケーブルでつなぐ。Android 上の aiBlocks アプリケーションで作成したプログラムを、Arduino マイコンボード上の iArduino インタプリタが実行する。

言語 iArduino に変換し、Arduino 上のインタプリタに渡し実行させる (図 2)。インタプリタはデバッグ用インタフェースを内蔵しているので、それを利用し aiBlocks は実行中のブロックをハイライト表示し、実行中の入力ピンの値などを表示する。aiBlocks の画面は図 3 となっている。プログラムを構成するブロックは 5 つのカテゴリに分類されており、①カテゴリ選択ボタンで②ブロックパレットを表示させる。ブロックパレットから必要なブロックを③プログラムエリアへドラッグし、プログラムに組み入れる。上部には④実行ツールバーがあり、プログラムの変換・転送 (書き込み)、実行などの操作ボタンが並んでいる。

### 3 作例集の作成

作例集は iArduino 言語の作例集<sup>1</sup>を基に、自身が電子工作とプログラミングの初心者である著者の一人が構成を決めた。まず、作例を真似るのに必要な部品について簡単な紹介 (図 4) をし、ブレッドボードの使い方、カラーコードの読み方、aiBlocks の操作などを説明する (図 5)。つぎに Arduino 上の LED を使うプログラムを紹介した後で、ブレッドボード上での配線例と aiBlocks のプログラムリストを使って作例を紹介する (図 6)。見開き 2 ページを使い、準備する部品、配線例、プログラムを紹介し、トラブルシューティングを載せる。そして部品やプログラムの使い方を理解するために試すとよい内容を「やってみよう」として載せる。

作例の一覧を図 7 に示す。まず、「1.LED を点滅させる」でブレッドボードと aiBlocks の操作に慣れる。「2.LED



```
pinMode(13,OUTPUT);
for(;;){
    digitalWrite(13,HIGH);
    delay(1*1000);
    digitalWrite(13,LOW);
    delay(1*1000);
}
```

図 2: aiBlocks で作成したプログラム (上) を、iArduino 言語のプログラム (下) に変換し、インタプリタへ渡す。プログラムの実行中は、(上) の「1 秒待つ」のように実行中のブロックの枠を太く表示する。

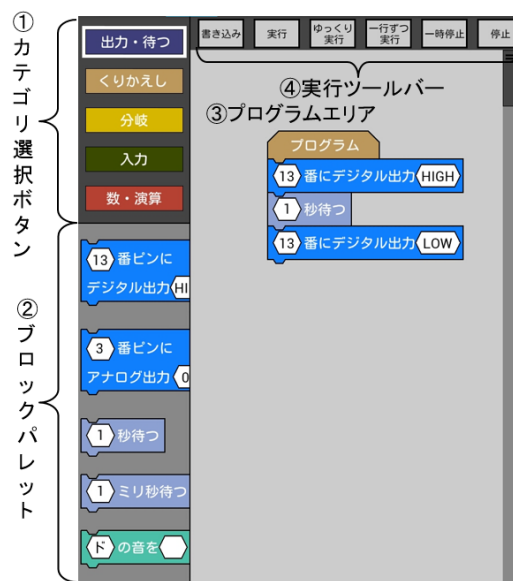


図 3: aiBlocks のプログラミングタブの表示。上部のタブバーを省略している。

<sup>1</sup><http://n.mtng.org/iArduino/ja/>

を2つ点滅させる」「3. 圧電スピーカを使って音を出す」「4. 模型用モータを回してみる」で、デジタル出力、音、待ちブロックを使った、光、音、動きの作り方を知る。そして「5.LEDの明るさを変えてみる」でアナログ出力ブロックを扱う。

「6. スイッチでLEDを点滅させる」「7. スイッチを押したらカッコウがなく」ではデジタル入力とifブロックの使い方を、「8. ボリュームの角度でLEDの点滅の速さを変える」でアナログ入力の使い方を紹介する。「9. 暗くなったらLEDを点滅させる」から「14. 距離センサで何か近づいたらLEDを点滅させる」はセンサの紹介を兼ねている。さいごに「15.LEDをだんだん明るく、だんだん暗くする」「16.LEDを順番に点灯する」「17. 明るさの変化に反応させる」で、少し抽象度の高い話題を取り扱う。

これらの作例の紹介の後で、作例の応用を3つ紹介している。1つは「3. 圧電スピーカを使って音を出す」「9. 暗くなったらLEDを点滅させる」を組み合わせた、照明を消すとハッピーバースデーのメロディーが流れる作品である。つぎに「3. 圧電スピーカを使って音を出す」「10. 温度センサで温度が高くなるとLEDを点灯させる」「17. 明るさの変化に反応させる」を組み合わせた、冷蔵庫のドアが開放されているとメロディーで知らせる作品である。さいごに「4. 模型用モータを回してみる」「7. スイッチを押したらカッコウが鳴く」を組み合わせた、スイッチで動く模型の作品を紹介している。ほかにトラブルシューティングの方法を紹介するページを設けている。

## 4 初心者による評価

aiBlocksと作例集の評価のため、プログラミング未経験の大学生(教育学部)5名の利用の様子を観察しインタビューした。観察に当たっては次のように指示した。まず、配線やプログラムに慣れるため、1~4の作例を作成する。次に12, 15以外の、好きなプログラムを選んでもらいテキストのページだけを見て作成する。そして、配線の最も複雑な「12. 重さを感じたら音を出す」と、プログラムの最も複雑な「15.LEDの明るさをだんだん明るく、だんだん暗くする」を作成する。それ以上の細かい指示はせず、プログラムの改変などは制限しなかった。利用は1人ずつとし、1人の利用時間はおよそ1時間30分である。

利用に当たってaiBlocksと作例集に大きな問題は見られなかったが、Arduinoボード上のLEDに気づきにくい、ブレッドボードの穴の位置の間違いに気づきにくいという問題がみられた。一方、作例集を利用し、配線

## 2. はじめるための準備 (じゅんび)

1. これから使うものを買う

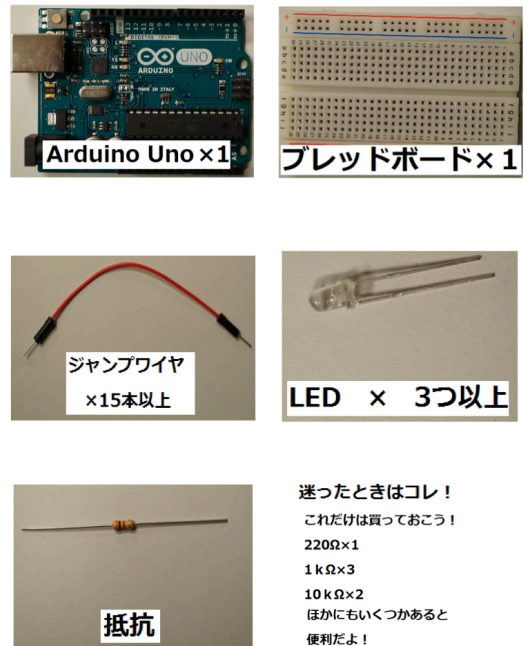


図 4: 作例を真似るのに必要な部品を紹介するページ(部分)

## 3. 知っておこう

1. ブレッドボードの使い方

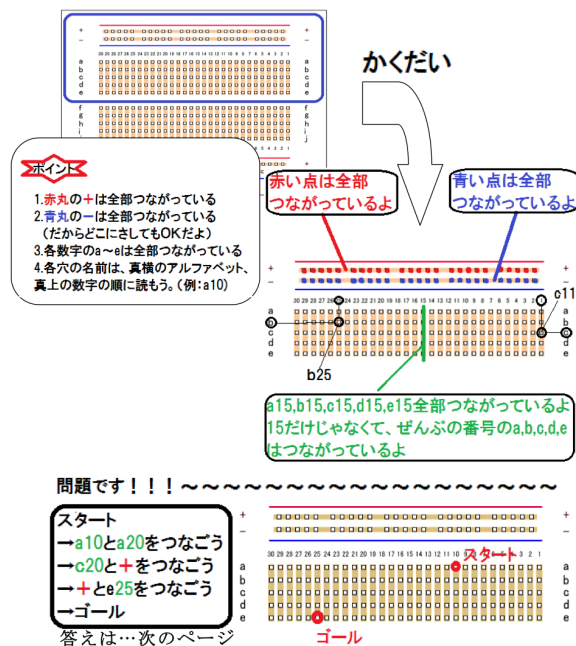
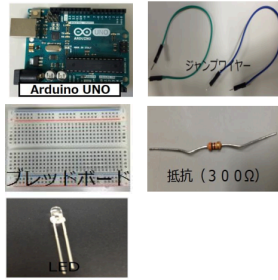


図 5: ブレッドボードの使い方を説明するページ

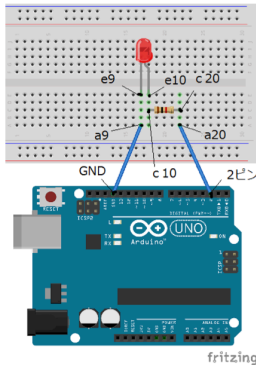
## 1.ブレッドボードに載せたLEDを点滅させよう！！

### 1.準備するもの

- ・Arduino UNO × 1
  - ・ジャンプワイヤ × 2
  - ・ブレッドボード × 1
  - ・LED × 1
  - ・抵抗(300Ω~1kΩ) × 1
- (抵抗どれを使えばいいか迷ったら今回はコレ！！300Ω=カラーコード橙黒茶金を使ってみよう。)



### 2.配線図

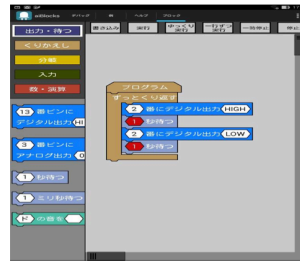


### 3.ここまで完璧！？チェック！

- 抵抗の値はちゃんと300Ω~1kΩの範囲内？(もしそれ以外だとArduinoが壊れちゃうかもよ！)
- タブレットとArduinoはつながってる？
- 繋ぎ方のチェックをするよ！
  - Arduinoのデジタル2番ピンとブレッドボードのa20がジャンプワイヤでつながっている？
  - 抵抗がc20とe10にささっている。(抵抗に向きはないよ。)
  - e10に抵抗の長い足、e9に短い足がささっている。
  - a9とArduinoのデジタルGNDがジャンプワイヤでつながっている。

チェックボックス全部OKかな？プログラムを書いてみよう！！

### 4. aiBlocks で書かれた作例のプログラム



- 全体を「ずっと繰り返す」で囲う。
- 左にある青色の「<13>番ピンにデジタル出力<HIGH>をずっと繰り返す中におこす。」
- 次に「<1>秒待つをその下におこす。」
- 同じ手順で図のようにあと1つつ並べよう。
- 「2つとも<13>をタッチして<2>を選択して、<2>に変えよう。」
- 次に下の「<2>にデジタル出力<HIGH>の<HIGH>をタッチして<LOW>に変えよう。」(すぐには変わらないかも、タッチしたら少し待ってね！)
- ここまでちゃんとできた？OKだったら、左上の「書き込む」を押そう！

ここまで完了したら、[実行]を押してみよう。ちゃんと点滅するかな？

### 5. あれ！？動かない！！そんなときは確認してみよう！

- プログラムに間違いはない？(2番ピンに変えて？ HIGHとLOWになってる？)
- 1度プログラムエリアに「<2>番ピンにデジタル出力<HIGH>」を出して、「<HIGH>」をクリックしてみよう。
  - そしたらもう一度クリックしてみよう。LEDはクリックに合わせて点滅するかな？
- 部品(ジャンプワイヤ、抵抗、LED)をちょっと触ってみよう。
  - 接触悪くない？ちゃんと奥までささってる？
  - もう一度右上の停止を押して、左上の書き込みを押して、やり直してみよう。それでもダメなら...
  - 1回電源を切って、つけないおしてみよう。

### やってみようのコーナー

- ・抵抗の値を変えてみよう。どんな変化があるのかな？(迷った人は、今回はコレ！！1kΩ = カラーコード 茶黒赤金 を使ってみよう)
- ・ブレッドボードの他の部分を使って配線しなおしてみよう。
- ・デジタル2番ピン以外のピンも使ってみよう。(プログラムのどこを書き直せばいいのかな？)

図 6: ブレッドボードに載せたLEDを点滅させる作例のページ。見開き2ページで構成している。

1. LEDを点滅させる
2. LEDを2つつ点滅させる
3. 圧電スピーカーを使って音を出す
4. 模型用モータを回してみる
5. LEDの明るさを変えてみる
6. スイッチでLEDを点滅させる
7. スイッチを押したらカッコウがなくなる
8. ボリュームの角度でLEDの点滅の速さを変える
9. 暗くなったらLEDを点滅させる
10. 温度センサで温度が高くなるとLEDを点灯させる
11. 熱くなったら扇風機を回す
12. 重さを感じたら音を出す
13. 人感センサで人が近づいたらLEDを点滅させる
14. 距離センサで何か近づいたらLEDを点滅させる
15. LEDをだんだん明るく、だんだん暗くする
16. LEDを順番に点灯する
17. 明るさの変化に反応させる

図 7: 作例集で取り扱う作例

図の読み方に慣れてくると、配線ミス(ブレッドボードの穴の位置の間違い)には自然と気づくような様子が観察された。

また、プログラムの作成が楽しくなり、aiBlocksで作成できる以上の長さのプログラムに挑戦した学生がいた。インタビューでは「慣れてきたら楽しかった」、「仕組みはわからないけど、動いたのはうれしかった」という感想が得られた。したがって aiBlocks と作例集は初めてマイコンのプログラムの作成する初心者が楽しめるものであると考える。

## 5 まとめと今後の課題

Androidで動作する、Arduinoマイコンボード用ビジュアルプログラミング環境 aiBlocks の作例集の制作と評価について報告した。今後は aiBlocks, 作例集を公開して、改善を進めていきたい。

## 参考文献

- [1] [http://www.byteworks.us/Byte\\_Works/techBASIC.html](http://www.byteworks.us/Byte_Works/techBASIC.html) (2015/4/19 閲覧)

- [2] <http://www.scratchjr.org/> (2015/4/19 閲覧)
- [3] 合同会社ソフトウメヤ: “Pyonkee”,  
<http://www.softumeya.com/pyonkee/ja/>  
(2015/4/19 閲覧)
- [4] 兼宗, 阿部, 原田: プログラミングが好きになる言語環境. 情報処理, vol.50, no.10, pp.986–995, 2009.
- [5] Arduino: <http://arduino.cc/> (2015/4/19 閲覧)
- [6] ArduBlock: <http://blog.ardublock.com/>  
(2015/4/19 閲覧)
- [7] <http://www.arduino-droid.info/p/introduction.html>  
(2015/4/19 閲覧)
- [8] 光永: タブレット端末で動作する, インタプリタ型言語搭載マイコンのプログラミング環境の開発. 情報処理学会, コンピュータと教育研究会, 第119回研究発表会, 情報処理学会研究報告, Vol.2013-CE-119, No.8, 2013.
- [9] 井芹, 光永: aiBlocks: マイコンにインタプリタを載せて利用するタブレット端末用ビジュアルプログラミング環境. 情報処理学会 コンピュータと教育研究会 128 回研究発表会, 情報処理学会研究報告, VOL.2015-CE-128, No.8, 2015.

# 人間動作の観察から動作模倣を行う人型ロボットのためのパーティクルフィルタを用いた関節角度時系列推定

Real-time joint angle estimation for a humanoid robot to imitate human motion using particle filter

高橋泰岳 榊原洸太

Yasutake TAKAHASHI and Kouta SAKAKIBARA

福井大学

University of Fukui

{yasutake,ksakakibara}@ir.his.u-fukui.ac.jp

## Abstract

An inexpensive camera with depth sensor enables us to observe joint positions of the human demonstrator with reasonable accuracy in a fast and comfortable way. In order to mimic the motion observed with the depth sensor based motion capture system by a humanoid robot, inverse kinematics should be solved to acquire the joint angle of the robot. However, It is tough to solve the inverse kinematics, in fact, to calculate the joint angles of the robot from the position of the joints of the human demonstrator, because of redundancy and difference of body link structure between a human demonstrator and a humanoid robot. This paper proposes a particle-filter-based joint angle estimation method for the humanoid robot to imitate human demonstration. The method provides a realistic solution of the estimated sequence of the joint angle for the humanoid robot imitation in real-time.

## 1 はじめに

近年、人間と同じ環境で活動する自律ロボットの研究や開発が進められている。ロボットが人間と同じ環境で活動するためには、ロボット自らが人間の環境に適応していくことが求められるが、人間の環境においてロボットに必要とされる動作をあらかじめ全て設計しておくことは非常に難しい。そこで、人間の動作をロボットに提示して、ロボットに学習をさせる模倣学習の研究 [Inamura 04] が行われている。

従来の人型の自律ロボットの模倣学習の研究において、人間の動作を模倣させるためのロボットの関節角が既知と

して学習が行われてきた [奥澤 11][Peters 11][谷口 09] が、人間とロボットのリンク構造は一致しないため、一般的にその仮定を満たすのは難しい。近年では安価な深度センサが低価格で入手できるようになり、これを利用した簡易モーションキャプチャシステムを用い、リンク姿勢を簡単に取得出来るようになった。ヒトのリンク姿勢からロボットの関節角を推定することが出来ればロボットの模倣学習に活用することが出来る。

ノイズの多いヒトのリンク姿勢の時系列データからロボットの関節角時系列を一意に求めることは一般的には難しい。例えば、リンク姿勢誤差と関節角の変化が最小となる関節角の時系列を全探索で求める手法 [臼井 13] が考えられるが、ノイズを含めて最適解を求めるため不自然な解を提示したり、探索を全探索で行うためリアルタイムで推定を行えないなどの問題があった。そこで、本報告ではリンク姿勢から関節角の推定に柔軟な計算が行え、リアルタイムに推定を行うことができるパーティクルフィルタを用いる手法を提案し、実験を通してその有効性を検証する。

## 2 パーティクルフィルタによる関節角の推定

Microsoft 社の KINECT センサ (以下、KINECT) を用いてモーションキャプチャを行い、マーカやセンサ類を装着せずに人間の骨格を検出し、15 点の関節位置座標時系列データを記録する (図 1)。記録した教示動作データから関節角を推定するにあたり、ロボット座標系  $\Sigma_r$  を設定し、観測されたカメラ座標系における教示動作の各関節座標をロボット座標系へ変換する (図 2)。リンク姿勢とは関節と関節を結ぶ剛体をリンクとし、リンクの単位方向ベクトルをリンク姿勢と呼ぶ。関節  $l$  と  $m$  を結ぶリンク  $j$  のリンク姿勢を式 (1) で定義する。

$$\varphi^j = \frac{\mathbf{p}_l - \mathbf{p}_m}{\|\mathbf{p}_l - \mathbf{p}_m\|} \quad (1)$$

$p_l$  と  $p_m$  はそれぞれ関節  $l$  と  $m$  のロボット座標系での位置座標で,  $\|\cdot\|$  はユークリッドノルムとする. 本実験に利用するパーティクルフィルタのアルゴリズムを Algorithm 1 に示す. まず, ロボットの関節角のセットをパーティクルとし, 複数用意して初期化する. 3 行目で各パーティクルにおける各関節角の更新を行うが, ここではガウス分布にもとづいてランダムに移動すると仮定している. Algorithm 1 の 4 行目でヒトのリンク姿勢とロボットの関節角にもとづく尤度関数を用いる. 尤度関数はロボットが制御する関節角から得られるロボットのリンク姿勢と観察したヒトのリンク姿勢の類似度とし, 具体的には次式で定義する.

$$h({}^h\varphi_t^1, \dots, {}^h\varphi_t^N | \theta_t) = \exp\left(-\frac{(1-s)^2}{\sigma}\right) \quad (2)$$

$$s = \frac{1}{N} \sum_{j=1}^N {}^h\varphi_t^j \cdot r\varphi^j(\theta_t) \quad (3)$$

${}^h\varphi_t^j$  は KINECT で取得したリンク  $j$  の時刻  $t$  における姿勢,  $r\varphi^j$  はロボットのリンク姿勢を表す. ロボットのリンク構造を図 3 に示す.  $s$  は人間のリンク姿勢とロボットのリンク姿勢の内積であり, 尤度はその内積が 1 に近いほど高い値を示す.  $\sigma$  は類似度関数の広がり決定する係数である. 6 行から 9 行目で尤度の高いパーティクルが優先的に保存され, 尤度が低いパーティクルを排除する.

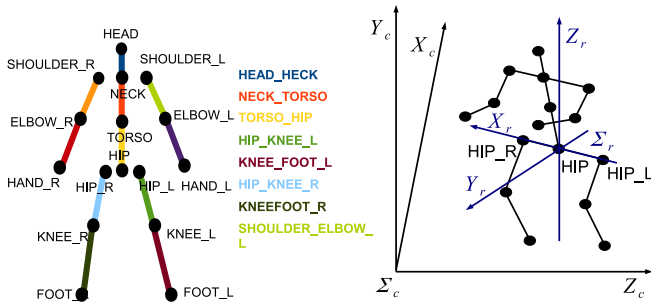


図 1: KINECT で取得する関節とリンク  
図 2: カメラ座標系とロボット座標系

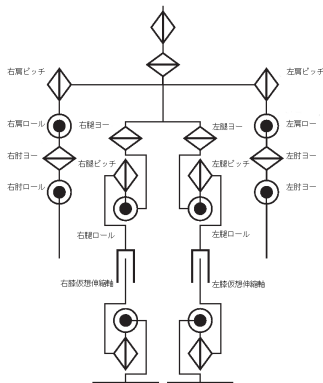


図 3: 人型ロボットのリンク構造

### Algorithm 1 Particle filter

- 1: Initialize particles  $\Theta_t = (\theta_t^{[1]}, \theta_t^{[2]}, \dots, \theta_t^{[M]})$
- 2: **for**  $m = 1$  to  $M$  **do**
- 3: Update particles with the motion model:  
 $\theta_t^{[m]} = \theta_{t-1}^{[m]} + \mathcal{N}(0, \Sigma)\Delta t$
- 4: Calculate the belief of each particle with the measurement model:  
 $w^{[m]} = h({}^h\varphi_t^1, \dots, {}^h\varphi_t^N | \theta_t^{[m]})$
- 5: **end for**
- 6: **for**  $m = 1$  to  $M$  **do**
- 7: draw  $m$  from  $\Theta_t$  with probability  $\propto w^{[m]}$
- 8: add  $\theta_t^{[m]}$  to  $\Theta_{t+1}$
- 9: **end for**
- 10: return  $\Theta_{t+1}$

### 3 検証実験

ここでは提案手法の有効性を示すため 2 つの実験を行った. まず, 人間の動作を KINECT で撮影し, 得られたリンク姿勢からパーティクルフィルタを用いてロボットの関節角の推定を行い, その推定結果を検証した. 図 4 に推定した関節角の時系列をそれぞれ示す. 約 0.2 秒までは推定ができていないが, その後は安定して推定している. 図 4(d) の赤線は右足股関節のピッチ軸を制御する関節の角度を表しているが, ヒトのキック動作に合わせて, 右足が前後に振られる様子が確認できる. また, 他の部位においても同様に関節の動きを推定出来ている. 推定したロボットの関節角を用い, シミュレータ上でロボットを動かす. この一連の作業をリアルタイムで行えることを確認した. この実験より, 人の動作認識の推定においてリアルタイムに処理し, かつ想定するロボットの実行可能な関節角時系列を出力することが確認された.

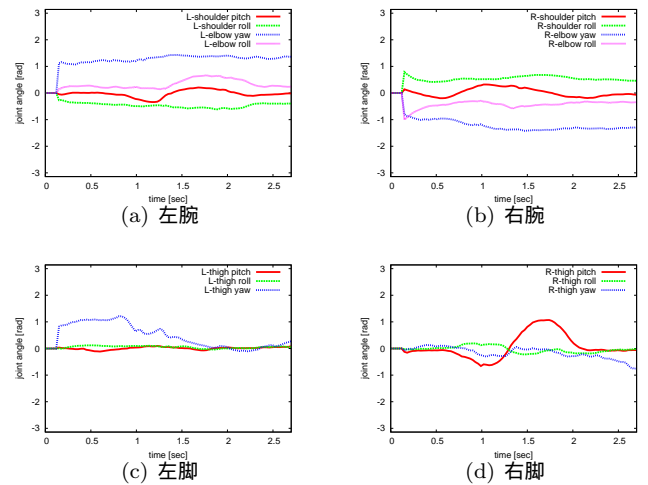


図 4: ヒトのキック動作の観察から推定したロボットの関節角の時系列データ

次に、同一の人型ロボットを2台用意し、片方のロボットのリンクを動かしてそのリンク姿勢を別のロボットに観測させた。観測するロボットはパーティクルフィルタを用いて関節角の推定を行い、推定した関節角がオリジナルのロボットの関節角度とどの程度あっているのかを検証をした。図5と図6はそれぞれ動作を呈示する人型ロボットの関節角と模倣をさせた人型ロボットの関節角を示す。姿勢が一意に決定できる時は正しく推定が行われていることが確認できる。図6(a), 図6(b)では、それぞれのヨー角(青線)が大きく乱れているが、これは動作を呈示するロボットの腕がまっすぐに伸び、ヨー角について一意に求められないからである。腕を曲げるなどしてヨー角が一意に求められた後は、比較的安定して推定できることがわかる。リンク姿勢が冗長自由度により一意に同定できない関節角については誤差が大きいが、時系列上で一意に同定できる時間であればその時点で真値に近づいていることが確認された。

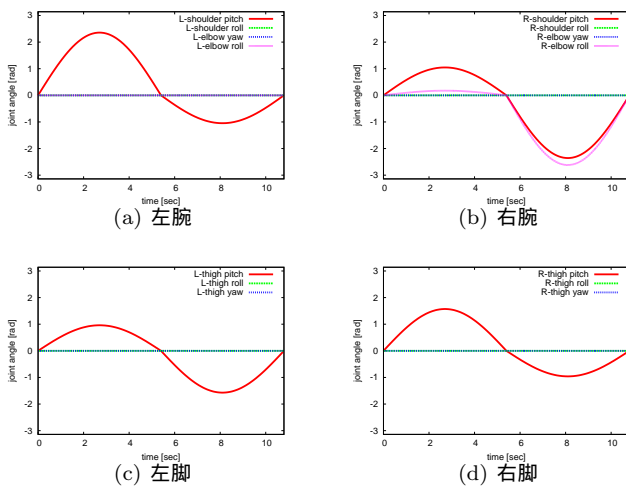


図 5: リンクを動かした人型ロボットの関節角

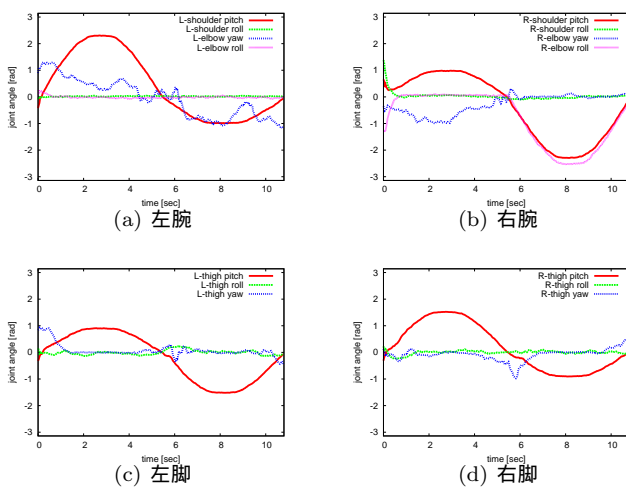


図 6: 模倣をさせた人型ロボットの関節角

#### 4 おわりに

本報告では、KINECT で観測したヒトの動作中の関節の時系列データを用い、人型ロボットがその動作を模倣するためにパーティクルフィルタを用いて関節角の推定を行う手法を提案した。実験より、動作模倣に必要な関節角の推定をリアルタイムに行えることが確認できた。また、リンク構造の冗長性によって一意に関節角が求まらない場合でも、時系列を考慮したアルゴリズムになっているため、妥当な関節角を提示できることを示した。今後はリアルタイムに推定した関節角時系列データを用いた模倣学習に発展させる。

#### 参考文献

- [Inamura 04] Inamura, T., Toshima, I., Tanie, H., and Nakamura, Y.: Embodied symbol emergence based on mimesis theory, *Int. J. Robotics Research*, Vol. 23, No. 4, pp. 363–377 (2004)
- [Peters 11] Peters, J., Mulling, K., Kober, J., Nguyen-Tuong, D., and Kromer, O.: Towards Motor Skill Learning for Robotics, *Robotics Research*, Vol. 131, No. 70, pp. 469–482 (2011)
- [白井 13] 白井 和廉: ヒューマノイドロボットのための対話型強化学習システムの開発, 福井大学 卒業論文 (2013)
- [奥澤 11] 奥澤 悠樹, 加藤 昇平, 加納 政芳, 伊藤英則: 運動の知識化に基づく模倣ロボットの運動認識と応用生成, 電気学会論文誌. C, 電子・情報・システム部門誌, Vol. 131, No. 3, pp. 655–663 (2011)
- [谷口 09] 谷口 忠大, 岩橋 直人: 複数予測モデル遷移の N-gram 統計に基づく非分節運動系列からの模倣学習手法, 日本知能情報ファジィ学会誌, Vol. 21, No. 6, pp. 1143–1154 (2009)



# 渋滞に対する複数ロボットの経路設定に関する一考察

## About routing of multi-robots considering the congestion

内海 孝亮, 植村 渉

Kosuke Utsumi and Wataru Uemura

龍谷大学大学院理工学研究科電子情報学専攻  
Ryukoku University

[utsumi@vega.elec.ryukoku.ac.jp](mailto:utsumi@vega.elec.ryukoku.ac.jp), [wataru@rins.ryukoku.ac.jp](mailto:wataru@rins.ryukoku.ac.jp)

### 概要

近年、生産工場や大型倉庫では作業効率、安全性の向上を図るため産業ロボットを用いたオートメーション化への移行が進んでいる。複数の産業ロボットが同時に動いているので、それぞれの行動によってお互いの進路をふさいでしまい、スムーズに動けなくなることがある。このようなデッドロックが起こらない経路設定方法が必要である。そこで、本研究では複数のロボットが同一環境下で独立して動く際の経路設定方法と渋滞の発生率の関係を調べるためにシミュレータを作成し、ロボット移動の一方通行化に基づくデッドロック回避手法[Takadama 2013]を基に 5 つの経路設定を比較し、渋滞に対して有効な経路設定のルールを検証する。

### 1 はじめに

生産工場や大型倉庫では少量多品種の生産が要求され、作業効率、安全性の向上を図るため、人間による手作業から産業ロボットを用いたオートメーション化への移行が進んでいる。複数の産業ロボットが動いているので、それぞれの行動によってお互いの進路をふさぎ、スムーズに動けなくなることがあり、場合によっては人間の手を介さなければ解決できないことがある。このようなデッドロックが起こらない経路設定が必要である。

複数のロボットが同一環境下で独立して動く際、経路設定方法と渋滞の発生率の関係を調べ、渋滞に対して有効な経路設定のルールを検証する。ただしロボット間では通信を行わず、お互いの経路情報を教えあうことはできない場合を考える。

### 2 渋滞とデッドロックの回避方法

渋滞やデッドロック状態を回避するために様々な方法が提案されている。渋滞回避の交通方法としてラウンドアバウトについて説明する。また、デッドロック回避の先行研究としてロボット移動の一方通行化に基づくデッドロック回避について説明する。

#### 2.1 ラウンドアバウト

ラウンドアバウトとは信号のない交差点で、円形状の一方通行ロータリを交差点に作るものである[Nishinari 2006]。まず車は交差点に来るとロータリに

合流する。ただしロータリを既に走っている車が優先である。ロータリに入ると、その中を回って自分の目的の方向の道へ出ていく仕組みである(Figure 1)。交通量が少ないときは交差点で止まる車はなく、ロータリを回っていけばノンストップで目的の道に進めて信号機で制御する場合よりも効率的に移動ができる。しかし交通量が多くなってくると、ロータリ内は常に車で溢れてロータリ内に車がなかなか入れなくなる。これに対して信号機で制御する場合は、大抵 2 方向からしか車は交差点に進入しないので交差点内が車で溢れることは少ない。また、隣接するラウンドアバウトも同様な状態に陥っている場合は、ロータリ内から出ようとする車も出られなくなりデッドロックが発生する。これをグリッドロックといい、信号で制御する場合でも同様のことが言える。このようにラウンドアバウトは交通量が少ないときは効果的だが、交通量が多いと逆効果になってしまうことがある。

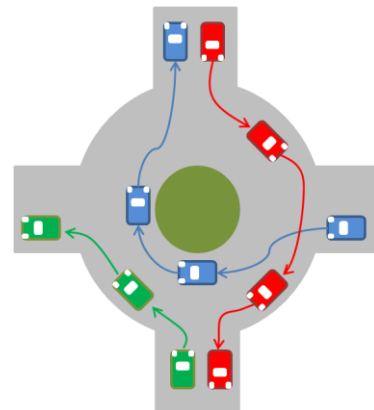


Figure 1 ラウンドアバウトの一例

#### 2.2 ロボット移動の一方通行化に基づくデッドロック回避

複数のロボットが同時に独立して動く際、一箇所にロボットが固まり多数のロボットの経路が重なった場合がある[Takadama 2013]。ここでは六角形状のパネルを敷き詰めたようなグラフの辺上をロボットが移動し、ロボットが移動可能な点が 1 箇所しかないと考えられる。

### 2.2.1 移動経路の一方通行化

移動可能な点が1箇所しかないような超高密度環境において、ロボットは移動する前の元の点に戻ってしまうとデッドロックをいつまでも回避することはできない。したがって、ロボットが移動可能空間に移動したあとは元の位置には戻らず、前進し続けることが必要である。そのためには、全ての辺に対してロボットの移動方向を一方通行化する必要がある。全ての辺における移動方向を一方通行化された構造物は有向グラフになっていると言える。このグラフ上においてある点からある点まで移動できることを示すには強連結の有向グラフであることを示さなければならない。

### 2.2.2 回転パネルと非回転パネル

あるグラフが強連結の有向グラフであることを示すためには、ロビンスの定理[Robin J 1972]より、全ての頂点が少なくとも一つの閉路に含まれている必要がある。ここでは六角形パネルをそれぞれ閉路として見る。しかし、全ての六角形を閉路として見ることはできない。これは、閉路となる六角形を3枚繋げることができないからである。3枚パネルのうち1枚は移動方向が時計回りにも反時計回りにも定められない。したがって、閉路となる六角形と閉路とならない六角形を適切に配置する必要がある。閉路となる六角形を**回転パネル**、閉路とならない六角形を**非回転パネル**と呼ぶ。この2つパネルは1) 回転パネルは逆の回転パネルと接続可能、2) 非回転パネルは回転パネルに隣接可能、3) 非回転パネル同士は隣接不可の条件を満たすように従って配置される。Figure 2はこのルールに従って配置したグラフである。この図では矢印が付いている六角形が回転パネル、矢印が付いていない六角形が非回転パネルを表している。

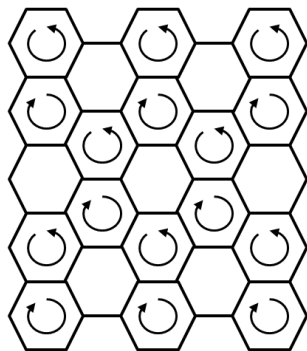


Figure 2 ルールに従って配置されたパネル

### 3 検討事項

2章で述べたラウンドアバウトを連ねていくとFigure 3のように多角形パネルを敷き詰めたような形になり、その辺上を移動しているように見える。そこで、2章で述べたロボット移動の一方通行化に基

づくデッドロック回避に用いられた六角形パネルと比較して多角形パネルの敷き詰めパターンと密度の関係について調べ、渋滞に対して有効な経路設定を導く。また、複数の通行ルールを正方形パネルと六角形パネルを敷き詰めたものに適用する。本研究では2章で述べた回転パネルに基づく一方通行ルールを**回転パネルルール**、2点間をどちらにも移動できるルールを**双方向移動ルール**と呼び正方形パネルと六角形パネルに適用する。また、正方形パネルには一列または一行ごとに一方通行の向きを変える**格子状一方通行ルール**も用いる。本研究で比較していく正方形パネルの経路設定をFigure 4、六角形パネルの経路設定をFigure 5に示す。

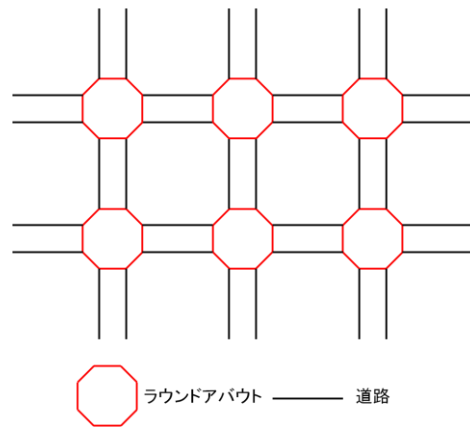


Figure 3 ラウンドアバウトが繋がった様子の略図

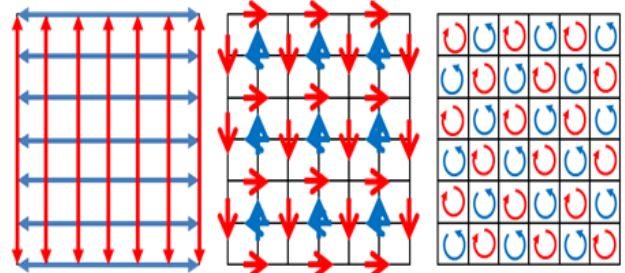


Figure 4 左から双方向、格子状一方通行、回転パネルルールを用いた正方形パネルの経路設定

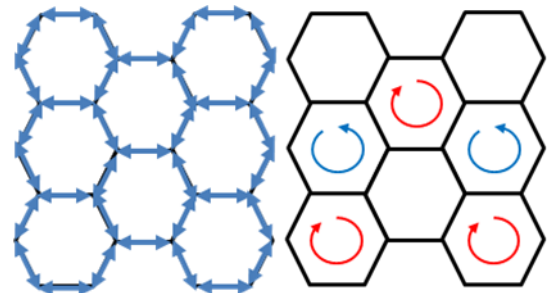


Figure 5 左から双方向、回転パネルルールを用いた六角形パネルの経路設定

## 4 実験

### 4.1 実験方法と評価方法について

本研究では渋滞に対して有効なロボットの経路設定を導出することが目的である。ここで渋滞に対して有効である経路設定とはロボットが任意の2点間をより多く往復できるものとする。各パネルの通行ルールに沿った複数ロボットが移動するシミュレーションを作成し、それぞれの経路設定を用いて実験を行う。ロボットが任意の2点間を移動し続け、一方の点からもう一方の点にたどり着くと1点加算していく。ロボットがある点から隣接する点に移動すると1ステップとして、1000ステップでシミュレーションを終了し各ロボットの点数の合計を比較して評価した。また、ロボットが移動方向を変更する際に生じる回転は30度で1ステップとする。

100台までのロボットを正方形に配置する。正方形の一辺上の座標からスタートし、向かい側の辺上の同じ点まで進み、元の座標に戻る動作を繰り返す。Figure 6に20台のロボットを配置した例を示す。

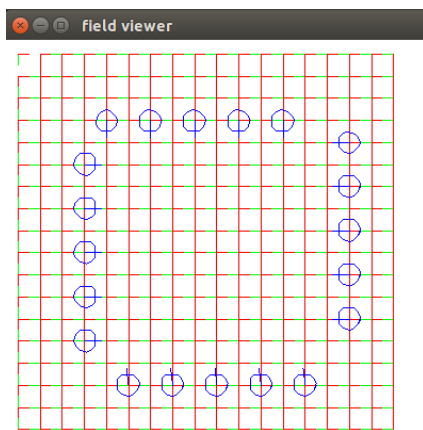


Figure 6 ロボットの台数を20台で配置した例

### 4.2 ロボットが移動するときのルール

各ロボットは移動方向にロボットがいない場合のみ移動する。移動方向にロボットがいる場合は、そのロボットが次の頂点に移るまでは静止する。このとき、あるロボットが1ステップ移動する間に静止しているロボットは1ステップ静止したとみなす。各ロボットは、移動出来ずその場で静止している状態では、自らが静止している間のステップ数を待機時間として計測する。ある頂点で静止していたロボットは隣接する頂点に移動するとき自らの待機時間を0にする。待機時間の初期値は無限大とする。

また分岐地点(Figure 7 青丸)では、隣接しているロボットがいなければ、隣接する2つの頂点の内目的地へ近い方を選択して移動する。もし片方の頂点のみにロボットがいる場合は、もう片方の頂点に移動

し、両方のロボットがいる場合は静止する。合流地点(Figure 7 赤丸)ではその地点に移動しようとしているロボットが他にいないければ、その頂点に移動する。その頂点に隣接しているロボットが同じ頂点に移動を予定している場合は、ロボットの待機時間を利用してどちらが移動するか決定する。待機時間が長い方が優先的に合流地点に移動する。ただし、初期状態や待機時間が全く同じ場合には、固有番号が小さい方が移動する。

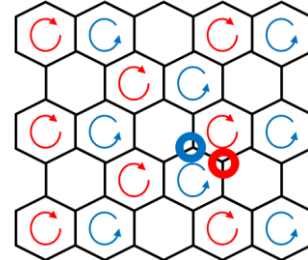


Figure 7 分岐地点と合流地点の様子

### 4.3 シミュレーションについて

各ルールに沿ったシミュレーションのフローチャートを Figure 8 に示す。

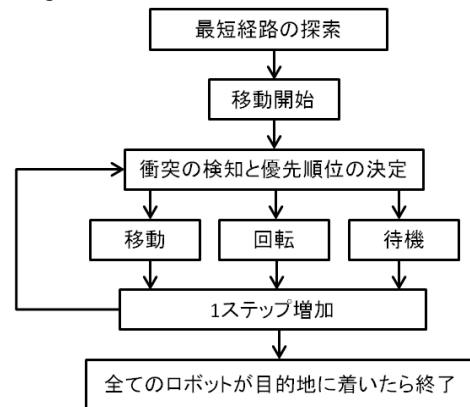


Figure 8 のフローチャート

Figure 9 はシミュレーションを実行した際の途中経過の様子である。

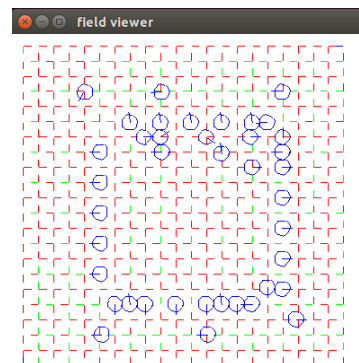


Figure 9 格子状一方通行ルールを適用した正方形パネルのシミュレーション

#### 4.4 結果と考察

ルールのパネルごとのロボットの台数と点数の関係を Figure 10 に示す。

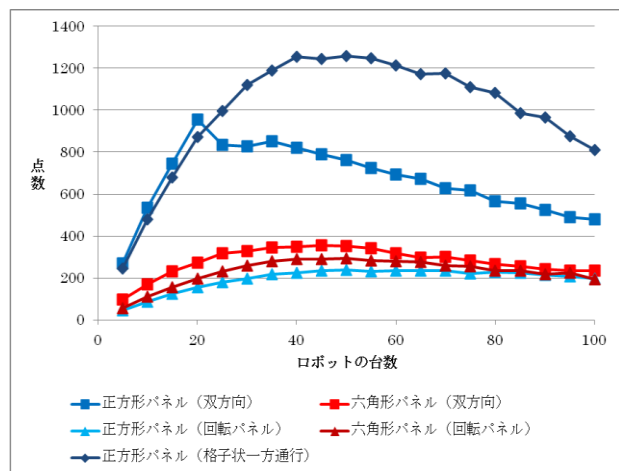


Figure 10 各パネル・各ルールにおけるロボットの台数と点数の関係

Figure 10 から格子状一方通行ルールを用いた正方形パネルが渋滞に対して適した経路設定であることがわかった。一方、回転パネルルールを用いると正方形パネル、六角形パネル共に点数が低くなり渋滞に弱いことがわかった。また双方向移動ルールと格子一方通行ルールを用いた正方形パネルの性能が 20 から 25 の間で交わっている。これは 20 台までは初期位置から直進しても反対側のロボットと衝突しないように配置されているが、20 台を超えて直進すると Figure 11 の赤丸のロボットのように、反対側のロボット衝突する可能性が出てくるためである。そのためそのまま直進できず、横方向に移動して衝突を防ぐので双方向ルールを用いた正方形パネルのグラフの傾きが大きくなり交わったと考えられる。また、格子状一方通行ルールや回転パネルルールではロボットの台数が多くなってくると Figure 12 のように全体的に右側にロボットが固まり一部ではデッドロックに近い状態になっていることが観測できた。双方向移動ルールでは観測できなかったので一方通行化が原因だと考えられる。

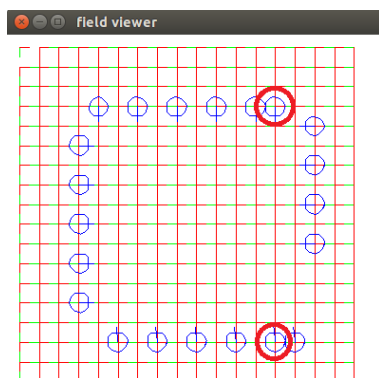


Figure 11 ロボットの台数が 21 台のとき配置図

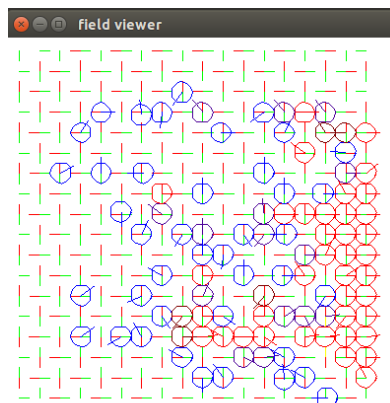


Figure 12 ロボットが右端に偏っている様子

#### 7 結論

本研究では、複数のロボットが同一環境下で動いているときに、それぞれの行動によってお互いの進路をふさいでしまい、渋滞やデッドロックを引き起こしてしまう問題に対して、シミュレーションを作成して渋滞に対して有効な経路設定方法を調べた。実験結果から格子一方通行ルールを用いた正方形パネルが渋滞に対して有効な経路設定であることがわかった。また回転パネルルールは渋滞に対しては有効な経路設定ではないこともわかった。しかし、双方向移動ルール以外はロボットの移動に偏りが出てデッドロックに近い状態になることも分かった。今後は他のパネルを敷き詰めた場合や異なる経路設定のロボットが同環境下で独立して動くときに対して有効な経路設定を提案することが課題である。

#### 謝辞

本研究は JSPS 科研費 15K16313 の助成を受けたものです。

#### 参考文献

- [Takadama 2013] 大谷雅之,佐藤寛之,服部聖彦,高玉圭樹: "超高密度環境下での複数エージェント協調によるデッドロック回避",第 40 回知能システムシンポジウム,P21-24,2013
- [Nishinari 2006] 西成活裕: 渋滞学, 新潮社(2006)
- [Robin J 1972] Robin J.Wilson: Introduction to Graph Theory 4<sup>th</sup> edition (邦訳:グラフ理論入門原書第 4 版, 西関隆夫訳, 西関裕子訳, 近代科学者(2001))
- [Uemura 2012] 植村渉: "マルチエージェントにおける剛直回避のための調整に関する一考察", 計測自動制御学会 システム・情報部門 学術講演会 2012,(1B2--3),(2012)

# 全方位移動ロボットに対する移動指示インタフェースの評価

For omnidirectional mobile robot evaluation of movement instruction interface

辻 和輝, 植村 渉

Kazuki Tsuji and Wataru Uemura

龍谷大学理工学部電子情報学科

Ryukoku University

[tsuji@vega.elec.ryukoku.ac.jp](mailto:tsuji@vega.elec.ryukoku.ac.jp), [wataru@rins.ryukoku.ac.jp](mailto:wataru@rins.ryukoku.ac.jp)

## 概要

現在社会において工場で利用されている産業用ロボットをはじめ、家庭用、医療用等多数のロボットが存在している。産業用ロボットは工場、倉庫等で部品や製品を搬送する役割を担っており、の効率化を図って搬送ロボットの駆動車輪に全方位移動可能な移動機構を持つ車輪を用いたロボットが開発されている。しかし、全方位移動可能な移動機構に対し、従来の操縦用のインタフェースでは操作が対応しておらず、全方位移動のメリットを活かしきれない場合がある。そこで本研究では全方位移動ロボットの自由度に対応したインタフェースに必要な指示方法を検討する。

## 1 はじめに

産業用ロボットには移動の効率化を図って、全方位移動可能な移動機構を持つ駆動車輪を用いたロボットが用いられている。全方位移動ロボットは前後への移動、左右への移動の2自由度に加えて $\pm 180$ 度の旋回運動が可能になっている。3自由度を有しており、狭いスペースを移動する際に切り返しの動作が必要ないなどの利点がある。しかし、自由度が増えたため従来の2自由度の機器向けの操縦用のインタフェースでは、全方位移動のメリットを活かしきれない場合がある。そこで、全方位移動ロボットの自由度に対応したインタフェースに必要な指示方法を検討するために、既存インタフェー

スをいくつか用意し、その評価から必要な指示方法の検討を行う。既存のインタフェースは、それぞれを実際に用意するのではなく、モーションキャプチャシステムであるKinectを用いて、仮想的に用意する。

## 2 自律移動ロボット

### 2.1 全方位移動

全方位移動ロボットとは全方位移動可能な移動機構を持つロボットのことである。全方位移動機構は従来と同じ前後の移動に加え、左右への移動と $\pm 180$ 度の旋回運動が可能であり、3自由度を有している (Figure1)。全方位移動機構の駆動車輪の全方位性ホイールは、ホイールの円周上に設置されている小型の樽型ローラの回転により、左右への移動が可能である。複数の全方位性ホイールを配置し、ホイールのベクトルを進行方向に合わせて足し合わせることで全方位移動を可能にしている。

全方位移動ロボットは狭いスペースを移動する際に切り返しの動作が必要ないなどの利点があり、効率的な移動に適している。現在は工場や倉庫等の限られたスペースでの搬送作業等に利用されている。

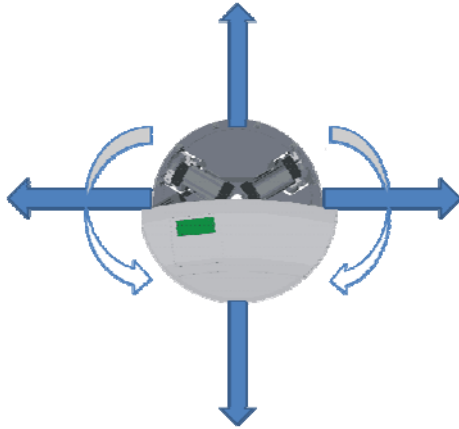


Figure 1 全方位移動ロボットの移動方向

## 2.2 Robotino

RobotinoとはFesto社が販売している移動式ロボットである (Figure2)。Robotinoは距離センサ、光電センサ等のセンサ類を持ち、電動グリッパやプッシュデバイス等を追加できる拡張性を持ち、画像処理、経路制御、プログラミング等のことを行うことができる。Robocup logistics league、ものづくり競技大会や技能五輪大会と言った、Robotinoの制御技術を競う大会が行われている。

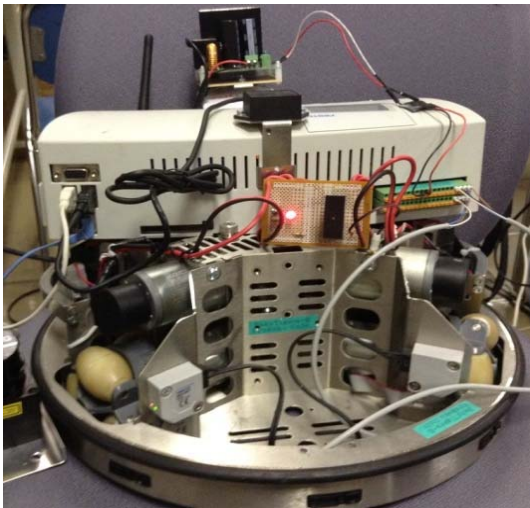


Figure2 Robotinoの外観

## 3 インタフェース

### 3.1 全方位移動可能な機器のインタフェース

全方位移動可能な移動機構を持つ機器として全方位性ホイールやそれと同じ構造を持つ

ホイールを使用した全方位移動車椅子がある。全方位移動車椅子のインタフェースとしてよく用いられているのがジョイスティックであり、ジョイスティックのレバーにより、前後と左右の四方向への方向入力が可能で、レバー部分を握って操作を行う。レーザレンジファインダから得た周辺の環境情報のデータを元に、障害物の方向にジョイスティックが倒れにくくすることで衝突回避を行っている。

### 3.2 二自由度の機器のインタフェース

従来の移動機構に対応したインタフェースには様々なものが開発されている。

自動車のインタフェースには運転操作のために、ステアリングホイールやペダル等が配置されている。左右への進行方向の変更にステアリングホイールを使い、前後の移動の速度調整にペダル類を使用する。

バイクは自動車と同じく、二自由度の機器であるが、インタフェースは自動車とは大きく違っている。左右への進行方向の変更に運転者の体重移動により車体を傾ける動作を用い、前後の速度調整にはアクセルグリップの操作を用いる。

### 3.3 モーションキャプチャデバイス

KinectはMicrosoft社が発売した、Xbox360用のゲームデバイスである (Figure3)。従来のゲームデバイスであったゲームパッドは物理的な接触による操作が必要であるが、Kinectは物理的なコントローラが必要なく、プレイヤーのジェスチャや、音声認識による操作ができる。RGBカメラ、距離カメラ、4つのマルチアレイマイク、チルトモーターを備えている。Kinectは距離カメラで撮影した人物の骨格情報を認識、追跡でき、人物の動きを3次元データで取得することが出来る。Microsoft社からSDK (Software Development Kit) (Kinect for Windows SDK

v1.8,

<https://www.microsoft.com/en-us/download/details.aspx?id=40278>が公開されており、ゲームだけではなく、リハビリテーションや手術中の画面操作デバイス等の分野にも利用されている。

Kinectには4つの大きな機能がある。

#### ・音声認識

音声認識は4箇所に配置されたマルチアレイマイクから行い、それぞれのマイクから音の遅れを解析し、音源位置の特定が可能である。

イタリア語、英語、スペイン語、ドイツ語、ポルトガル語、日本語に対応している。

#### 顔認識

プレイヤーの顔情報を記録する。

#### モーションキャプチャー

Kinectのモーションキャプチャーではマーカーとトラッカーを必要としない

#### 骨格トラッキング

深度センサからKinectの検出範囲にいる人物のKinectまでの距離を計測し、人物の骨格20箇所の位置と動きを検出する。可視化した骨格情報をFigure4に示す。



Figure3 Kinectの外観

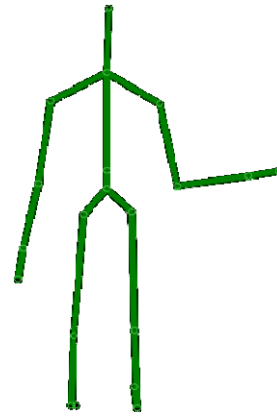


Figure4 可視化した骨格情報

## 4. Kinectを使った全方位移動ロボットの インタフェース

### 4.1 作成するプログラムの概要

本研究では全方向移動ロボットの自由度に対応した移動指示方法抽出のために、Kinectを用いて、仮想的に既存のインタフェースを用意する。Kinectを使ったプログラミングにVisual Studioを使用し、Kinectから取得した画像の表示にOPENCVを使用する。Visual StudioはMicrosoft社がリリースしているソフトウェア開発ツールであり、VisualC++、Visual Basicなど複数のプログラミング言語に対応している。OPENCVはオープンソースのコンピュータ・ビジョンライブラリであり、コンピュータで画像や動画の処理を行うのに必要な機能が実装されている。Kinectと操作ロボットの接続ブロック図をFigure5に示す。

Kinectの検出範囲内(800mm~4000mm)に人物が入り、Kinectがその人物の骨格情報を取得すると骨格トラッキングを開始する。人物を認識したKinectは骨格情報を数値化し、接続しているコンピュータにSDKを通して情報を送る。そしてその骨格トラッキング中の人物が特定の動作をすると、Kinectを介して制御プログラムが情報を受け取り、全方位移動ロボットのモー

タ部分を制御することで、ロボットが移動する。制御プログラムのフローチャートをFigure6に示す。

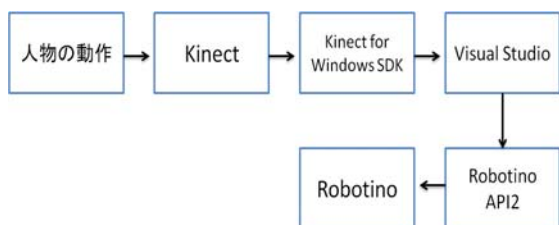


Figure5 Kinectとロボットブロック図

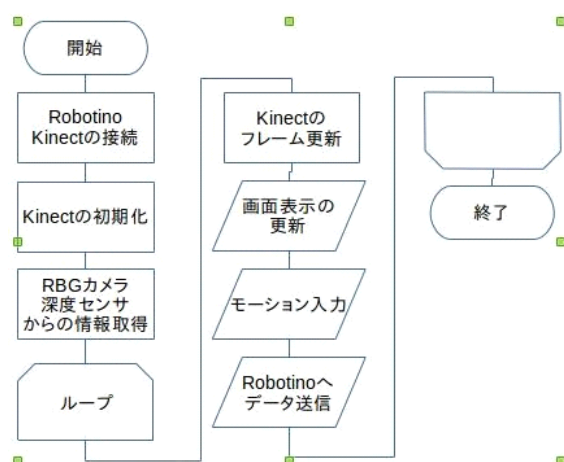


Figure6 フローチャート

## 5. 実験

### 5.1 実験方法

本研究では全方位移動ロボットに適した移動支持方法の抽出が目的である。そのため、従来のインターフェースである、自動車型、バイク型、ジョイスティック型の3つの仮想的インターフェースを作成する。

自動車型(Figure7)

両手の手のひらのX, Y座標をKinectから取得、右の手のひらと左の手のひらの二点間の角度をハンドル操作として判定する。速度調節には右足の前後の位置から判定する。

バイク型(Figure8)

両手手のひらの前後位置で速度調節の判定を

行い、体の傾きから方向変換の判定を行う。

ジョイスティック型(Figure9)

Kinectから右の手のひらのX, Z座標を取得し、XをロボットのX軸移動、ZをロボットのY軸の移動と判定する。

実験では3つの既存インターフェースを用いて全方位移動ロボットへの移動指示を行い、その評価を行う。操作する全方位移動ロボットにはRobotinoを使用する。操縦者には各インターフェースで経路走行を行ってもらおう。

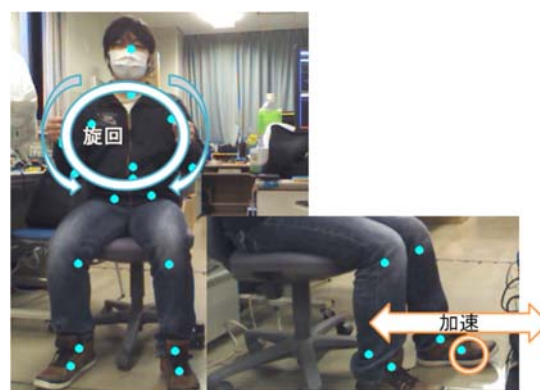


Figure7 自動車型の操作イメージ

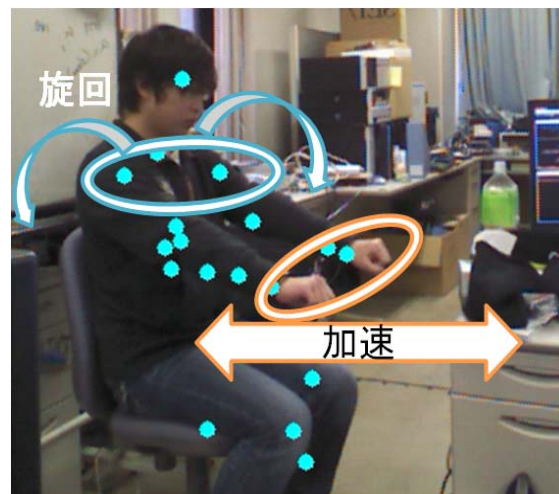


Figure8 バイク型の操作イメージ



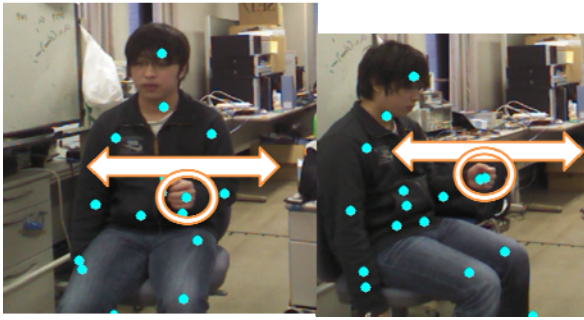


Figure9 ジョイスティック型の操作イメージ

## 5.2 評価方法

仮想的な既存のインタフェースをユーザに操作してもらい、その操作性を評価するためにアンケートを集計する。

アンケートには結果を定量的に扱うために、SD法を用いる(セマンティックディファレンシャル法。大小、冷熱など対形容詞を数段階に分けて評価させ、測定する方法。心理学者チャールズ・オズグッドが開発した、セマンティックディファレンシャル法, デジタル大辞泉, 小学館2008)。評価実験に用いた評価項目の一部をFigure10に示す。

評価項目として加速、旋回性、直進性、疲労感、安定感、総合的な操作性の優劣を測定する。

車	加速がいい					加速が悪い
バイク						
ジョイスティック						

Figure10 評価項目の例

## 5.3 実験結果

集計したアンケート結果をグラフ化したものをFigure11に示す。

結果からバイク型、ジョイスティック型の旋回性、安定性の低さが、自動車型のインタフェースでは全体的に高評価を得られた。それぞれのインタフェースの旋回方法を比較すると、バイ

ク型、ジョイスティック型では操縦者が体を傾けているつもりがない、斜め移動指示を出していないと操縦者が考えていても、僅かな体の傾きや、手の左右位置を取得して方向変更を行っており、このような操縦者の意図していない指示が旋回性、安定性の低評価の原因と考えられる。自動車型のハンドル操作では操縦者からハンドルが正面に位置して、方向変更を行っている両手の位置が見え、意図しない方向変更指示が起こりにくいのだと考える。全方位移動ロボットは従来の移動機構と違い移動動作に加えて旋回運動を行えることから、旋回の操作指示として意図しない操作の誤認識を避けるために、操縦者側から操作状況が把握しやすい操作指示が必要であると考えられる。

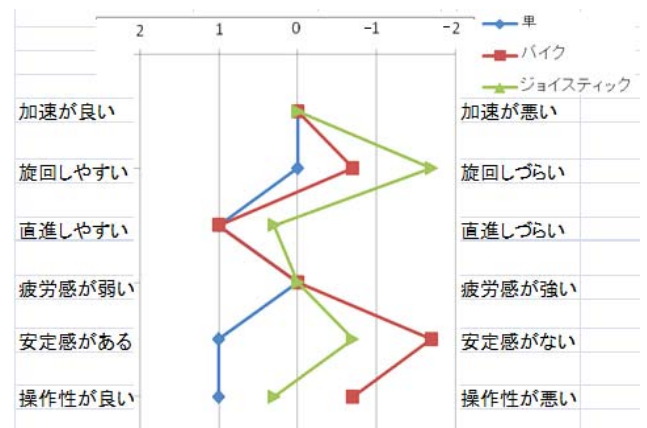


Figure11 グラフ化したアンケート結果

## 6 結論

本研究では、従来の移動機構よりも自由度が増えた全方位移動機構に適した操作方法の検討のため、従来のインタフェースをモーションキャプチャシステムであるKinectを用いて仮想的に作成し、移動方法の抽出を行った。実験結果から自動車のような両手での旋回操作と足での速度調整の操作は直進性、安定感、総合的な操作性と操縦者から良い評価を得られることが分かった。またバイクのような操作の場合は良い評価、悪い評価が別れてしまうことも

分かった。

今後は新たな評価軸を加えることや従来のインタフェースを加えることで、より全方位移動ロボットの自由度に対応した移動指示方法の抽出を進めていくことが課題である。

## 謝辞

本研究はJSPS科研費 15K16313 の助成を受けたものです。

## 参考文献

- [1]中村薫, 齋藤俊太, 宮城英人, KINECT for Windows SDKプログラミング C++編(秀和システム, 2012)
- [2]杉浦司, 中村薫 Kinect for Windows SDK 実践プログラミング(工学者)
- [3]Gary Bradski, Adrian Kaebler, 松田晃一(訳), 詳解 OpenCV —コンピュータビジョンライブラリを使った画像処理・認識 オライリージャパン

# 自律移動ロボットの環境地図作成のための簡易測域センサの利用と評価

distance sensors in order to make the map for the autonomous robots.

津田 諒太, 植村 渉

Ryota Tsuda and Wataru Uemura

龍谷大学理工学部電子情報学科

Ryukoku University

tsuda@vega.elec.ryukoku.ac.jp, wataru@rins.ryukoku.ac.jp

## Abstract

近年、公共施設や家庭で働く自律移動ロボットが注目されている。自律移動には、正確な環境地図の作成が重要になる。環境地図作成とは、壁などの障害物を 2 次元あるいは 3 次元的に測定し、測定点を蓄積することである。この地図により、通行可否の判断が可能になる。2 次元測定には測域センサである LRF(レーザーレンジファインダー) が用いられることが多い。特に、北陽電気社製の URG-04LX-UG01 がロボット研究でよく用いられており、毎秒 5000 点を測定する。しかし、平面が多い環境には性能過多である。

そこで、本研究では測定点の少ない簡易測域センサを作成し、そのセンサの環境地図作成の評価を行う。

## 1 はじめに

近年、自律移動ロボットが公共施設や家庭で働くということが現実味を帯びてきた。病院や介護施設での介護ロボットはもちろん、ルンバに代表される市販化された家庭用お手伝いロボットもある。これらのロボットに必要なことの一つに、経路選択がある。次の作業の目的地までどのようなルートで到達するか、ということをロボットは考え、実行しなければならない。この経路選択を円滑に行うために環境地図作成が重要となる。

環境地図作成とは、そのロボットが活動する環境の地図をロボット自身がセンサを用いて作成することである。平面を移動するロボットの場合、自分からどれだけ離れた場所に通れない場所があるかが分かればよい。測定を一点ではなく 2 次元的に行うことで、自分の位置情報とこれらの情報を組み合わせて地図を作成することができる。



Figure 1: 距離センサと制御部分

## 2 地図作成システム構成

自律移動ロボットの環境地図作成システムは、距離センサと制御部分に分けられる(図1)。

距離センサには赤外線、超音波、LRF などがある。今回の実験では赤外線距離センサで測域センサを作成し、LRF と精度の比較を行う。それぞれのセンサの性能を表1に、制御部分に用いる arduino と Robotino の性能を表2に示す。

## 3 距離センサの測域センサ化の提案

本章では距離センサを用いて測域センサ化する方法を提案する。

測域センサである LRF は回転する鏡でレーザー光を反射することで範囲測定を行っている。今回はサーボモータによって距離センサを往復運動することで LRF の再現を行う。サーボ、距離センサの制御に arduino を用いることで PWM の制御を簡潔にすることが期待できる。また、距離センサの値とサーボモータの角度の値をシリアル通信で出力する。

赤外線距離センサは LRF よりも 1 回の測定時間が長い。サーボの最大速度で 180° 回転すると角度分解能は約 6° となり、粗い測定となる。そこでここでは、距離センサを 2 つ用意し距離センサの測定面を 90 度直交した状態で固定した。これにより時間あたりの測定数が倍になり、さらに往復角度が半分になる。

赤外線距離センサの出力はアナログ値であるため回路

	LRF	赤外線距離センサ
型番	URG-04LX-UG01	GP2Y0A710K
測定可能距離	0.6m 5.6m	1m 5.5m
測定精度	1m : ± 30mm 1.4m : 距離の 3 %以下	1m : ± 2 % 1m : ± 2 %
1 点の測定時間	0.2mm	16ms
測定可能範囲	正面 ± 90	正面 1 点
測定方式	光の飛行時間による位相差方式	PSD(光位置センサ) による三角法
出力形式	デジタル値 (シリアル通信)	アナログ電圧値
コスト	高 (10 万円)	低 (1000 円)

Table 1: LRF と赤外線距離センサの性能比較

	Arduino	Robotino 2
動作電圧	DC5V	DC24V
搭載 CPU	ATmega328	AMD LX800
動作周波数	16MHz	500MHz
入出力ピン	20 本 (内 PWM 出力 6 本:A 入力 8 本)	D 入出力 16 本 A 入力 8 本

Table 2: 制御部の Arduino と Robotino の性能比較

による誤差、ノイズの影響が大きい。さらに赤外線距離センサに用いる PSD は測定時に急な電圧降下を起こす可能性がある。そこで赤外線距離センサの電源にバイパスコンデンサを接続する。また、ノイズの影響を除くためにメディアンフィルタを採用する。メディアンフィルタは複数の測定値の中央値を採用するフィルタである。フィルタの効果を高めるために測定数を増やすと角度分解能が減るため、フィルタ長の調整が重要である。

#### 4 実験、結果

LRF と作成した簡易測域センサを用いて環境地図を作成し、簡易測域センサの往復時間 (s) とメディアンフィルタ長の違い (長さ) による地図作成再現度、及びカバー率を比較する。龍谷大学 1-432 室前の空間 (図 2) で実験を行う。図 5 から図 13 は LRF と赤外線の地図を重ねたものであり、青が LRF、赤が赤外線の作成地図である。LRF の角度分解能 (0.36 度) の範囲ごとにと赤の画素の重心を求め、中心から赤の画素までの距離と中心から青の画素までの距離の 2 つの距離の比  $(1 - (R/(R + \Delta R)))$  を再現率とする (図 3)。また、緑の画素の先に赤の画素が存在する割合を認識率とする。測域部分の往復時間、メディアンフィルタ長ごとにそれぞれ 1 分間で作成した地図の比較をした (図 4)。

図 5 から図 13 より、センサ面と正面且つ距離が近い左右の障害物には精度よく反応した。逆にセンサ面と斜め且つ距離の遠い領域は精度が悪い。認識率はフィルタ長が短いほど良くなったが、フィルタ長が長いことによる再現率の改善はあまり見られなかった。つまり、簡易測域センサの誤差は電圧値 距離に変換する際、電圧値に対する係数が遠距離で一致していない為の誤差と考えられる。よって、今回作成した測域センサは、センサ面の正面か



Figure 2: 龍谷大学瀬田学舎 1 号館 432 室前

つ比較的近距离にのみ障害物が存在するという条件下で LRF と同等であると言える。

#### 5 まとめ

本研究では、LRF では過剰な環境のために赤外線距離センサを用いた簡易測域センサを提案した。実験にて簡易測域センサの往復速度、フィルタ長の違いによる地図作成能力の比較を行った。LRF を基準にパラメータ毎の地図の再現率、認識率を求めた。作成した測域センサは、センサ面の正面且つ比較的近距离にのみ障害物が存在するという条件下で LRF と同等であると言えた。遠距離領域に誤差が生じるのは遠距離での電圧値に対する係数が一致していないためだと分かった。今後は、距離に応じた適切な係数を求め、さらに精度の良い測域センサにしたい。

#### 謝辞

本研究は JSPS 科研費 15K16313 の助成を受けたものです。

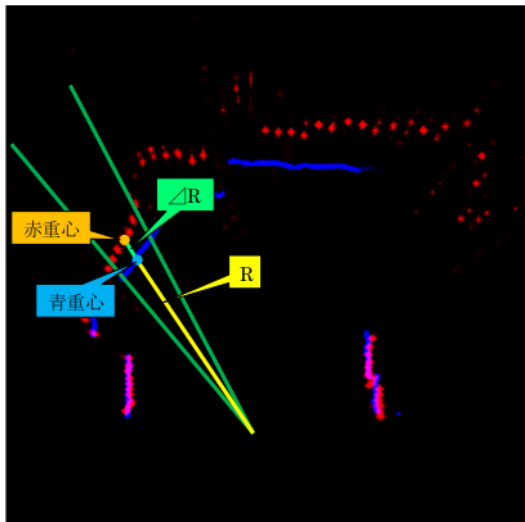


Figure 3: 重心求めイメージ

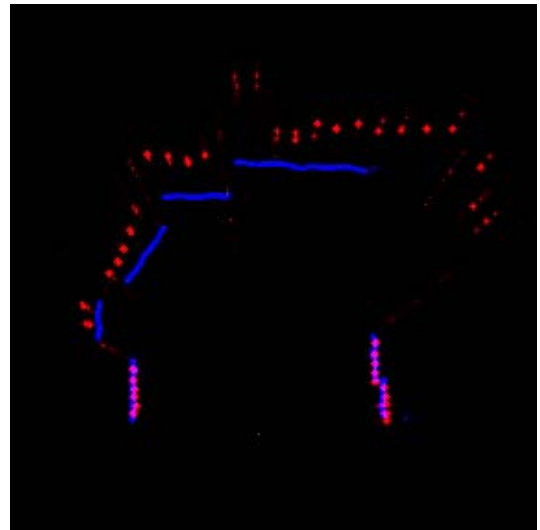


Figure 6: 往復時間 1.0s フィルタ長 4 地図

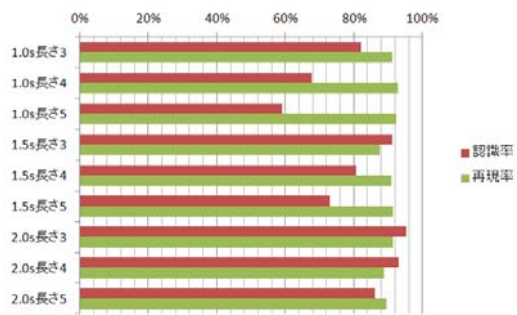


Figure 4: 往復時間、フィルタ長ごとの再現率、認識率

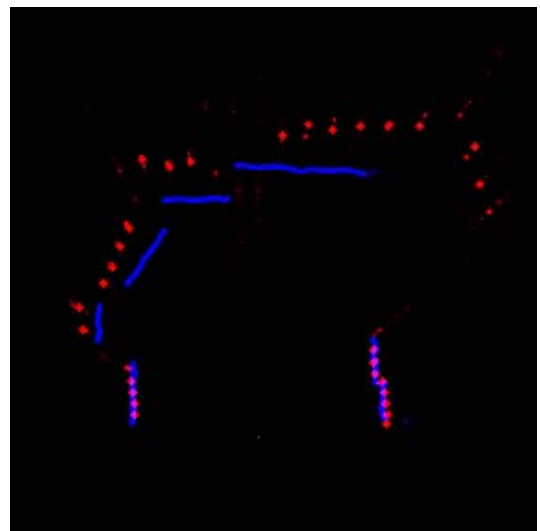


Figure 7: 往復時間 1.0s フィルタ長 5 地図

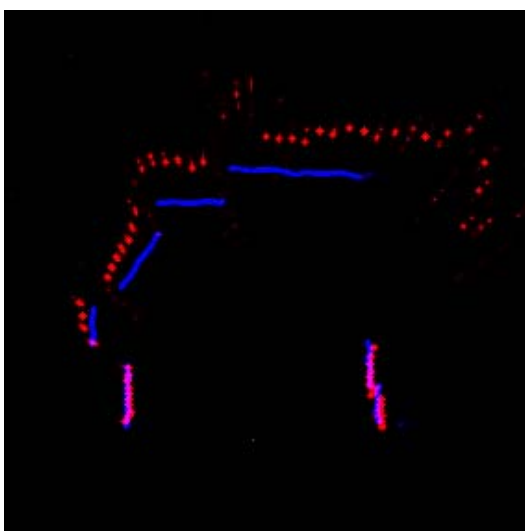


Figure 5: 往復時間 1.0s フィルタ長 3 地図

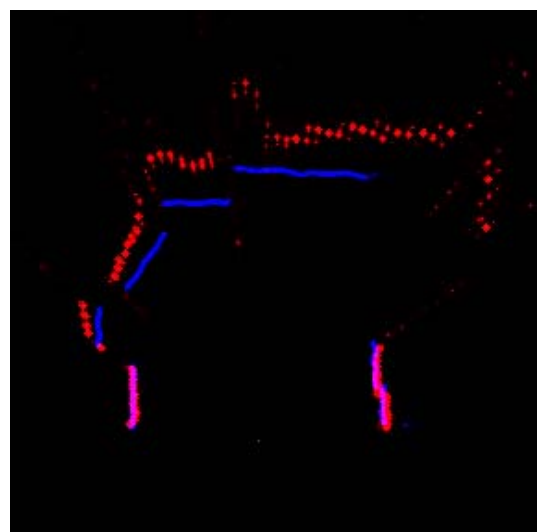


Figure 8: 往復時間 1.5s フィルタ長 3 地図

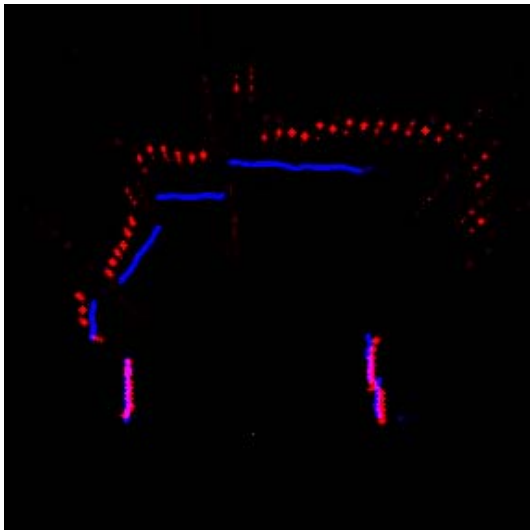


Figure 9: 往復時間 1.5s フィルタ長 4 地図

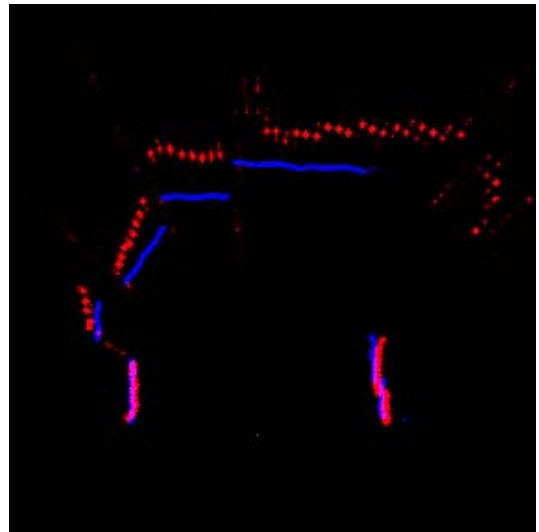


Figure 12: 往復時間 2.0s フィルタ長 4 地図

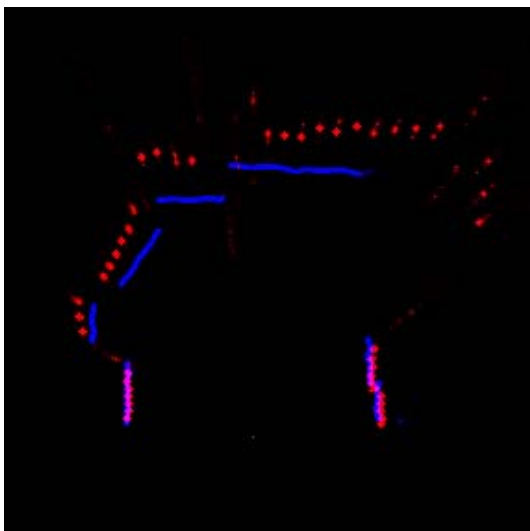


Figure 10: 往復時間 1.5s フィルタ長 5 地図

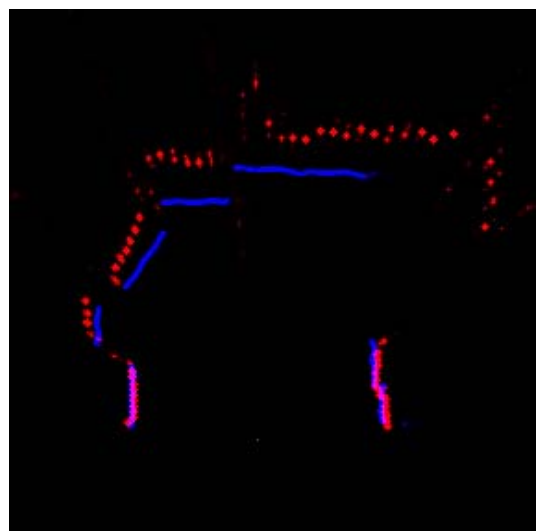


Figure 13: 往復時間 2.0s フィルタ長 5 地図

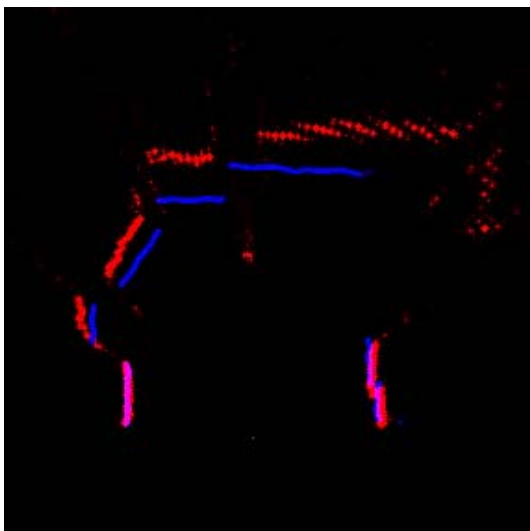


Figure 11: 往復時間 2.0s フィルタ長 3 地図

© 2015 Special Interest Group on AI Challenges  
Japanese Society for Artificial Intelligence  
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**

**光永 法明**

大阪教育大学 教員養成課程 技術教育講座

**主幹事**

**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 大学院  
情報理工学研究科

**担当幹事**

**植村 渉**

龍谷大学 理工学部 電子情報学科

**幹事**

**公文 誠**

熊本大学 大学院 自然科学研究科

**幹事**

**中村 圭佑**

(株) ホンダ・リサーチ・インスティテュート・  
ジャパン

**Executive Committee**

**Chair**

**Noriaki Mitsunaga**

Department of Technology Education  
Osaka Kyoiku University  
mitunaga @ cc.osaka-kyoiku.ac.jp

**Secretary**

**Kazuhiro Nakadai**

Honda Research Institute Japan Co., Ltd.  
/ Graduate School of Information  
Science and Engineering,  
Tokyo Institute of Technology

**Wataru Uemura**

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

**Makoto Kumon**

Graduate School of Science and  
Technology,  
Kumamoto University

**Keisuke Nakamura**

Honda Research Institute Japan Co., Ltd.

---

SIG-AI-Challenges web page; <http://winnie.kuis.kyoto-u.ac.jp/SIG-Challenge/>