RoboCup サッカーにおける敵位置の予測モデル構築
Constructing Prediction Models of Opponent Positions in RoboCup Soccer

山下雄大 †　中島智晴 †　秋山英久 ‡
Katsuhiro YAMASHITA †　Tomoharu NAKASHIMA †　Hidehisa AKIYAMA ‡
大阪府立大学 †　福岡大学 ‡
Osaka Prefecture University †　Fukuoka University ‡
katsuhiro.yamashita@cs.osakafu-u.ac.jp, tohomaru.nakashima@kis.osakafu-u.ac.jp,
akym@fukuoka-u.ac.jp

Abstract

In this paper, we propose a method that predicts opponent player's positions. This method is used to generate decision making and achieved by using neural networks. Three-layered neural network are used to learn the opponent positions. One neural network learns the mapping from a field status to the position of an opponent player. Thus, 11 neural networks are necessary to learn opponent 11 players. The results by the numerical experiments show that the proposed method predicts opponent positions and helps to generate appropriate decision making.

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 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．RoboCup サッカーでは，競技で勝利することが重要視され，ただ単に勝利するだけではなく，ボールを支配し，確実に勝利することが望まれている．そのためには，チームでの試合運びを行うための戦術が必要である．高度な戦術を取るためには，敵プレイヤの位置を予測することが必要であるが，現状ではそれは困難である．そこで本論文では，敵位置決定のモデル化について調査する．本論文では，RoboCup サッカーシミュレーション 2D リーグを題材とする．敵位置決定のモデルをサッカーフィールドの状況から敵プレイヤ位置へのマッピングである，と定義する．敵位置決定のモデル化にはニューラルネットワークを用いる．数値実験では，ニューラルネットワークの予測精度を調査し，プレイヤに組み込むことで発生するチームへの影響を調査する．

2　行動探索

RoboCup にはサッカー，レスキュー，@ホームの他に，次世代のロボット技術者育成を目的とした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を研究の対象とする．サッカーシミュレーションはモデル化の形式によって 2D リーグと 3D リーグに分けられる．本論文では 2D リーグを扱う．Figure 1 に 2D リーグの試合の様子を示す．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として競技を行う．また，プレイヤ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．試合は 1 サイクル 0.1 秒で離散化され，前後半 3000 サイクルずつ合計 6000 サイクルで試合が行われる．各プレイヤ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意思決定を行う．

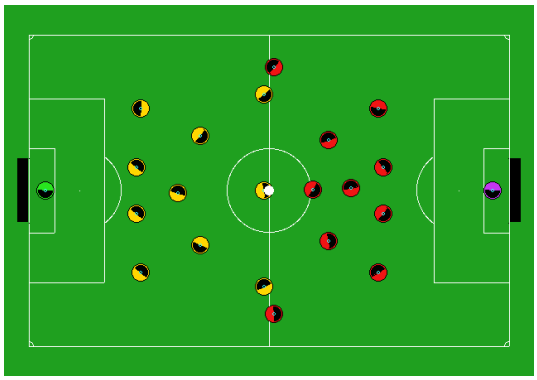

Figure 1: 2D リーグの試合の様子

プレイヤが意思決定を的確にできるかどうかは，チーム戦略における重要な要素である．高レベルの意思決定に関しては，高度な戦略に基づいた試合が行われており，ポジショニング等に関する研究が積極的に行われている．

Luis ら[1] の，Situation Based Strategic Positioning を用いた手法や，Akiyama ら[2]の，Delaunay Triangulation を用いた手法などがある．本論文で使用するプレイヤは，最良優先探索を用いて行動連鎖と呼ばれる木構造を構築することで意思決定を行う[3]．まず，プレイヤの現在の状態をルートノードに入力する．次に，ノードに入力された状態において実行可能な行動の候補（パス，ドリブル，シュートなど）を生成する．生成された行動に対して評価値を計算し，その行動を実行した場合の予測状態と共に子ノードに追加する．ノードが追加されるたびに評価値が最大であるノードを選択し，そのノードにおける予測状態から再び実行可能な候補の行動を生成する．これを繰り返すことで，ノード数があらかじめ設定された最大値に達するまで探索木を成長させる．ただし，木の深さがあらかじめ設定した値を越える場合や，ノードの予測状態から行動が生成できない場合，行動連鎖の終了条件に設定されている行動（シュート）が生成された場合は，その葉ノードの子ノード生成は行わないものとする．構築された木構造の中からノード列をつなげると，行動連鎖が得られる．評価値が最大となるノードから同様に実行可能な行動の候補を生成し，探索木を成長させる．最も評価値の高いノード列を選択することで，戦略上良いと考えられる行動連鎖を実行することが可能となる．

行動連鎖の例を Figure 2 に示す．Figure 2 において，ボールを保持している 10 番のプレイヤは以下のような行動連鎖を生成する．まず，10 番のプレイヤが 7 番のプレイヤにパスを行う，パスを受け取った 7 番のプレイヤはドリブルによりボールを前へ運び，その後 9 番のプレイヤにパスを行う．パスを受け取った 9 番のプレイヤは，相手ゴールに向かってシュートする．10 番のプレイヤは以上の流れを考慮して，7 番のプレイヤへのパスを実行する．行動連鎖を生成することで，プレイヤは数手先の状況を考慮して，より戦略的価値が高い行動を選択することが可能となる．

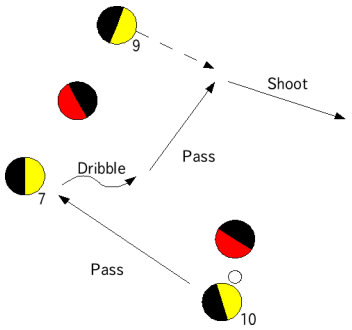

Figure 2: 行動連鎖の例

成功確率を上げるには予測状態の精度向上が必要と予想される．予測の精度を上げるために各プレイヤの位置を正確に予測することが重要である．しかし，プレイヤは制限された情報しか保持していないため，特に敵プレイヤの位置を正確に予測する事は困難である．オープンソースであるサンプルチームの agent2d [4] では，行動連鎖生成時の敵プレイヤの予測位置は，最後に自分が認識した敵プレイヤの位置で固定となっている．しかし，実際には敵プレイヤはサイクル経過と共に動くため，予測状態での敵プレイヤ位置と，実際に行動を実行した後での敵プレイヤの位置に大きな差異が生じてしまい，それによって生成された行動連鎖の質は下がってしまう．そこで本論文では，ニューラルネットワークを用いて敵プレイヤの位置を予測し，予測状態での敵プレイヤの位置を考慮する手法を提案する．

3 提案手法

本論文での提案手法では，ニューラルネットワークを用いた学習を行う．そこで，まずニューラルネットワークの概要について示す．

3.1 ニューラルネットワーク

ニューラルネットワークは，人間の脳の神経回路をシミュレートしたモデルで，高度な学習機能を備えている．ニューラルネットワークには様々な形状のものが存在するが，本論文では三層階層型ニューラルネットワークを用いる．Figure 3 にその概形を示す．三層階層型ニューラルネットワークには入力層，中間層，出力層が存在し，それぞれの層にユニットが存在する．

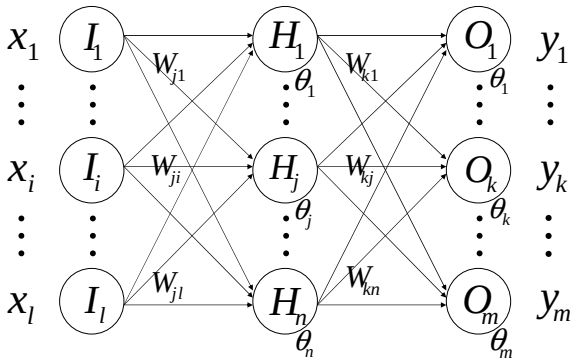

Figure 3: ニューラルネットワークの概形

Figure 3 において，入力層ユニットを $I_i(i = 1, \cdots, l)$，中間層ユニットを $H_j(j = 1, \cdots, n)$，出力層ユニットを $O_k(k = 1, \cdots, m)$ とする．$x_i(i = 1, \cdots, l)$ は入力信号であり，$W_{ji}(i = 1, \cdots, l, j = 1, \cdots, n)$ は I_i と H_j の結合強度，$W_{kj}(j = 1, \cdots, n, k = 1, \cdots, m)$ は H_j と O_k の結合強度である．また，$\theta_j(j = 1, \cdots, n)$ は H_j の閾値，$\theta_k(k = 1, \cdots, m)$ は O_k の閾値を表し，$y_k(k = 1, \cdots, m)$

は出力信号である．三層階層型ニューラルネットワークでは，入力層ユニットに入力信号 x_i が入力される．入力層ユニットの出力を $o_i(i = 1, \cdots, l)$ とすると，

$$o_i = x_i \tag{1}$$

である．中間層ユニットでは入力層の出力 o_i を入力として，以下の式によって計算された $o_j(j = 1, \cdots, n)$ を出力とする．

$$o_j = f(net_j) \tag{2}$$

$$net_j = \sum_{i=1}^{n} W_{ji} o_i + \theta_j \tag{3}$$

$$f(net_j) = \frac{1}{1 + e^{-net_j}} \tag{4}$$

出力層ユニットでは中間層の出力 o_j を入力として，以下の式によって計算された $o_k(k = 1, \cdots, m)$ を出力とする．

$$o_k = f(net_k) \tag{5}$$

$$net_k = \sum_{j=1}^{m} W_{kj} o_j + \theta_k \tag{6}$$

$$f(net_k) = \frac{1}{1 + e^{-net_k}} \tag{7}$$

ニューラルネットワークの出力信号を y_k とすると，

$$y_k = o_k \tag{8}$$

である．ニューラルネットワークの学習には，次に示す誤差逆伝搬学習アルゴリズムを用いる．

3.2 誤差逆伝搬学習アルゴリズム

誤差逆伝搬学習アルゴリズムは，入力信号に対する望ましい出力値を教師信号として与え，出力値との誤差を最小化するように結合強度と閾値を更新するアルゴリズムである．結合強度の更新は学習と呼ばれ，出力値 o_k に対する教師信号 t_k との差を以下の評価関数で表し，これが最小となるように行われる．

$$E = \sum_{k=1}^{m} \frac{1}{2}(t_k - o_k)^2 \tag{9}$$

式 (9) より，結合強度 W_{ji}，W_{kj} および閾値 θ_j，θ_k の修正量は，学習係数を η とすると，それぞれ以下の式で表される．

$$\Delta W_{ji} = \eta \cdot \left(-\frac{\partial E}{\partial W_{ji}} \right) = \eta \delta_j o_i \tag{10}$$

$$\Delta W_{kj} = \eta \cdot \left(-\frac{\partial E}{\partial W_{kj}} \right) = \eta \delta_k o_j \tag{11}$$

$$\Delta \theta_j = \eta \cdot \left(-\frac{\partial E}{\partial \theta_j} \right) = \eta \delta_j \tag{12}$$

$$\Delta \theta_k = \eta \cdot \left(-\frac{\partial E}{\partial \theta_k} \right) = \eta \delta_k \tag{13}$$

$$\delta_j = o_j(1 - o_j) \sum_{k=1}^{m} \delta_k W_{kj} \tag{14}$$

$$\delta_k = (t_k - o_k) o_k (1 - o_k) \tag{15}$$

ここで，学習速度を向上させるために，慣性項係数を α として，修正量を次のようにする．

$$\Delta W_{ji} = \eta \delta_j o_i + \alpha \Delta W_{ji}^{old} \tag{16}$$

$$\Delta W_{kj} = \eta \delta_k o_j + \alpha \Delta W_{kj}^{old} \tag{17}$$

$$\Delta \theta_j = \eta \delta_j + \alpha \Delta \theta_j^{old} \tag{18}$$

$$\Delta \theta_k = \eta \delta_k + \alpha \Delta \theta_k^{old} \tag{19}$$

したがって，結合強度と閾値は次のように更新される．

$$W_{ji}^{new} = W_{ji}^{old} + \Delta W_{ji} \tag{20}$$

$$W_{kj}^{new} = W_{kj}^{old} + \Delta W_{kj} \tag{21}$$

$$\theta_j^{new} = \theta_j^{old} + \Delta \theta_j \tag{22}$$

$$\theta_i^{new} = \theta_i^{old} + \Delta \theta_i \tag{23}$$

3.3 提案手法

試合中における通常プレイが行われている状態を playon と呼ぶ．敵プレイヤの位置予測を行うニューラルネットワークを 1 プレイヤにつき 1 つ用意し，playon 時の敵プレイヤの位置を学習させる．ニューラルネットワークの入力として，

- 現在のボール位置
- ボールの到達予想位置
- 最も近い敵プレイヤがボールの位置に到達するまでに要するサイクル数

を用いる．これらは予測状態を生成するのに必要な情報と考えられる．教師データは，試合の前半における playon 時の敵エージェントの x 座標と y 座標とする．ただし，ボールに最も近い敵プレイヤは，その性質上ボールに向かって移動することが多いため，本来のフォーメーションでの動きとは別の動きをとると見なし，教師データからは除外する．各入力データは正規化された後にニューラルネットワークに入力される．各 x 座標は次のように x_{input} へと正規化される．ここで，$PitchWidth$ はフィールドの縦の幅 105m に，10m の余裕を持たせたものである．

$$x_{input} = \max(\frac{x}{PitchWidth} + 0.5, 1.0) \tag{24}$$

各 y 座標は次のように y_{input} へと正規化される．ここで，$PitchLength$ はフィールドの横の幅 68m に，10m の余裕を持たせたものである．

$$y_{input} = \max(\frac{y}{PitchLength} + 0.5, 1.0) \tag{25}$$

到達サイクル数 $cycle$ は次のように $cycle_{input}$ へと正規化される．

$$cycle_{input} = \max(\frac{cycle}{20}, 1.0) \quad (26)$$

到達サイクル数は 20 サイクルを越えるものはほとんど見られないため，20 サイクル以上は 1.0 とみなす．

ニューラルネットワークの学習は，試合中の各サイクルでのフィールド上の完全な情報を保持できるコーチエージェントによって行われる．コーチエージェントはハーフタイム中に各敵プレイヤに対して順番にニューラルネットワークを学習させる．学習を行ったニューラルネットワークに予測状態時点での各敵プレイヤの位置を出力させ，敵プレイヤのフィールド上での位置を予測する．学習したニューラルネットワークの情報は，コーチから各プレイヤに送られ，プレイヤはその受け取った情報からニューラルネットワークを構築し，行動連鎖生成時の予測状態での敵プレイヤの位置推定に使用する．行動連鎖生成時に使用される行動実行後の予測状態における敵プレイヤの位置を，ニューラルネットワークで得られる敵プレイヤの位置にすることで，より正確な予測状態を作成することを狙う．またそれにより，戦術的価値のより高い行動連鎖を生成する．

3.4 予備実験

提案手法で必要な，中間層ユニット数，学習係数 η，慣性項係数 α，学習のエポック数の各パラメータの値を予備実験により決定した．予備実験では，RoboCup2012 決勝戦の HELIOS2012 のログデータを用いる．ニューラルネットワークの入力は，現在のボールの x 座標と y 座標，出力は，背番号 11 のプレイヤの x 座標と y 座標と設定し学習を行う．学習には誤差逆伝搬学習アルゴリズムを用いる．

3.4.1 中間層ユニット数

エポック数を 10000，$\eta = 0.7$，$\alpha = 0.3$ と固定し，中間層ユニット数を 3，4，5，6，7，8，9，10，11，12，14，20 と変化させて学習を行い，実際の位置との誤差を出力させた．Table 1 に各中間層ユニット数ごとの距離の誤差を示す．Table 1 より，中間層ユニット数は 10 個が最適であると分かる．

3.4.2 学習率

エポック数を 10000，中間層ユニット数を 10 個，$\alpha = 0.3$ と固定し，η の値を 0.1〜0.9 の範囲で変化させて学習を行い，実際の位置との誤差を出力させた．Table 2 に η の値ごとの誤差を示す．Table 2 より，η の値は 0.9 が最適であると分かる．

3.4.3 慣性項係数

エポック数を 10000，中間数ユニット数を 10 個，$\eta = 0.7$ と固定し，α の値を 0.0〜0.3 の範囲で変化させて学習を行い，実際の位置との誤差を出力させた．Table 3 に各中間層ユニット数ごとの誤差を示す．Table 3 より，α の値は 0.3 が最適であると分かる．

3.4.4 エポック数

中間層ユニット数を 10 個，$\eta = 0.7$，$\alpha = 0.3$ と固定し，エポック数を 100，1000，2000，5000，10000，20000，40000，100000，300000 と変化させて学習を行い，実際の位置との誤差を出力させた．Table 4 に各中間層ユニット数ごとの誤差を示す．Table 4 より，エポック数は 2000

Table 1: 中間層ユニット数の変化による誤差

中間層ユニット数	誤差
3	21.80
4	22.19
5	22.50
6	22.76
7	22.97
8	23.15
9	23.30
10	16.40
11	16.48
12	16.55
14	16.66
20	16.89

Table 2: 学習率の変化による誤差

学習率	誤差
0.1	65.51
0.2	45.07
0.3	33.92
0.4	27.00
0.5	22.30
0.6	18.92
0.7	16.40
0.8	14.44
0.9	12.89

Table 3: 慣性項係数の変化による誤差

慣性項係数	誤差
0.0	17.63
0.1	17.05
0.2	16.47
0.3	16.40

で十分学習が行われていると考えられる．エポック数が極端に少ない場合に誤差が小さくなっているように見えるのは，ニューラルネットワークの初期値の影響により，学習が進んでいないためであると考えられる．

Table 4: エポック数の変化による誤差

エポック数	誤差
100	15.72
1000	16.29
2000	16.34
5000	16.38
10000	16.39
20000	16.40
40000	16.41
100000	16.41
300000	16.41

4 数値実験

数値実験では，提案手法を用いた敵プレイヤの位置予測の精度と，提案手法をプレイヤに組み込んだ場合のチーム性能を調査する．ニューラルネットワークの入力層ユニット数，中間層ユニット数，学習係数，慣性項係数の各パラメータは，予備実験の結果から，入力層ユニット数を 5，中間層ユニットを 10，学習係数を 0.3，慣性項係数を 0.9 と設定する．また，入力情報は，現在のボールの x 座標と y 座標，ボールに最も近い敵エージェントがボールに到達すると予測されるサイクル数，そのサイクル数が経過したときのボールの予想到達位置の x 座標と y 座標とする．出力値は敵プレイヤの x 座標と y 座標とする．また，ボールに最も近い敵プレイヤは，ボールを取りにいこうとするためにボールに向かってくることが多く，本来のフォーメーションでの位置から外れた動きをすることが多いので，ニューラルネットワークでの学習要素からは除外する．

4.1 予測精度の評価

提案手法での敵プレイヤの位置予想の精度を調査する．コーチエージェントは前半 3000 サイクルまでの playon 時の各敵プレイヤの位置を記憶しておく．ニューラルネットワークの教師データは，前半 3000 サイクルまでの playon 時に得られた敵プレイヤの位置情報から生成され，ハーフタイム中に各敵プレイヤに対して 2000 エポック学習する．コーチエージェントが試合の後半の playon 時にニューラルネットワークに現在の状態を入力して得られる予測された各敵プレイヤの位置と，実際に予測されたサイクルになった時点での各敵プレイヤの位置の比較を行い，それらの誤差を計測する．

4.2 プレイヤに組み込む場合の評価

各プレイヤが保持しているニューラルネットワークから出力された敵プレイヤの予測位置を，行動連鎖生成時の予測状態に組み込んだチームで試合を実行し，行動連鎖 1 段目での予測状態における各敵プレイヤの位置と，実際に行動連鎖 1 段目の行動を実行した後での各敵プレイヤの位置との誤差を計測する．同様の誤差の比較を，提案手法をプレイヤに組み込んでいないチームでも行い，これらを比較することで，提案手法の効果を確認する．こちらの実験では，試合開始時点で各プレイヤに対して，あらかじめ敵チームのプレイヤの位置を予測するためのニューラルネットワークを与えておき，コーチエージェントがハーフタイムに作成したニューラルネットワークへの更新は行わないものとする．

自チームは，RoboCup2012 に出場した HELIOS2012 [5] とする．RoboCup2012 に出場したチームと agent2d [4] を相手にそれぞれ 10 試合ずつ行う．

4.3 実験結果

提案手法を組み込んだ HELIOS2012 で試合を行い，5.1 章で述べた評価方法を用いて予測精度を評価した．計測された誤差のゴールキーパーを除いたフィールドプレイヤの平均と標準偏差を Table 5 に示す．なお，誤差の平均の単位はメートルとする．

Table 5: 予測の精度

対戦相手	誤差の平均	誤差の標準偏差
agent2d	10.53	7.89
WrightEagle	11.88	8.04
MarliK	11.07	9.01
Ri-one	11.26	6.84
Oxsy	13.62	9.21
YuShan	13.57	9.30
NADCO-2D	10.85	8.00

Table 5 より，どの相手チームでニューラルネットワークを用いて学習を行っても，各敵プレイヤに対して十数メートルの誤差で予測することができることが分かった．ゴールキーパーは，その性質上あまり激しく動き回らずに定位置にいることが多いため，誤差の平均はどのチームも 2 メートルから 5 メートルの範囲で収まっている．また，相手チームやそのポジションによっては，誤差の平均値が 15 メートルを越える場合もあることが分かった．これは，そのプレイヤが，チーム内でボールに積極的にからみにいくことで，フォーメーションから外れた動きをすることが多いために生じると考えられる．

次に，提案手法を組み込んだ場合のチームに対する効

果を，5.2 章で述べた評価方法を用いて評価した．得られたフィールドプレイヤの誤差の平均値を Table 6 に，その標準偏差を Table 7 に示す．

Table 6: 行動連鎖 1 段目における誤差の平均

対戦相手	提案手法なし	提案手法あり
agent2d	15.00	17.80
WrightEagle	15.06	17.56
MarliK	27.55	24.97
Ri-one	12.92	16.89
Oxsy	18.27	21.40
YuShan	10.51	16.65
NADCO-2D	19.55	20.06

Table 7: 行動連鎖 1 段目における誤差の標準偏差

対戦相手	提案手法なし	提案手法あり
agent2d	78.28	27.14
WrightEagle	101.06	25.30
MarliK	100.17	31.25
Ri-one	70.30	16.10
Oxsy	78.86	16.82
YuShan	44.19	18.20
NADCO-2D	84.31	28.61

Table 6 から，提案手法を組み込んだ場合と組み込んでない場合を比較すると，誤差の平均値だけで見ると提案手法はあまり効果がないように思われる．しかし Table 7 から，提案手法を組み込んだ場合と組み込んでない場合の標準偏差を比較すると，提案手法を組み込んだほうが小さくなっていることが分かる．提案手法を組み込んでいないほうの誤差について細かく見てみると，ほぼ一致している場合と，大幅にずれている場合の二極化していることがわかった．これは，提案手法を組み込んでいない場合は，敵プレイヤの位置を最後に認識した場所で固定してあるため，実際に動いていなかった場合は誤差が小さく．動いていた場合は非常に大きな誤差が出てしまうためであると考えられる．提案手法の標準偏差が小さいのは，敵の動きを考慮して予測しているため，大幅なずれが生じていないからであると考えられる．

5 おわりに

本論文では，ニューラルネットの誤差逆伝搬法を用いて敵エージェントのフォーメーションのモデル化を行い，実データから学習することで既知のチームの位置予測が可能であることを示し，行動連鎖への組み込みで予測性能を検証した．今後の課題としては，位置予測の精度の向上や，行動連鎖に組み込んだ際のチームのパフォーマンスについて細かく分析を行うことが挙げられる．

参考文献

[1] Luis Paulo Reis, Nuno Lau and Eugenio Oliveira, “Situation Based Strategic Positioning for Coordinating a Simulated RoboSoccer Team”, Balancing Reactivity and Social Deliberation in MAS, 175–197, 2001

[2] Hidehisa Akiyama and Itsuki Noda, “Multi-Agent Positioning Mechanism in the Dynamic Environment”, RoboCup 2007 : Robot Soccer World Cup XI, 2008.

[3] Hidehisa Akiyama, Tomoharu Nakashima and Shigeto Aramaki, “Online Cooperative Behavior Planning using a Tree Search Method in the RoboCup Soccer Simulation”, Proceedings of 4th IEEE International Conference on Intelligent Network and Collaborative Systems , 2012.

[4] agent2d , http://rctools.sourceforge.jp/pukiwiki/index.php?agent2d

[5] Hidehisa Akiyama, Hiroki Shimora, Tomoharu Nakashima, Yosuke Narimoto, Katsuhiro Yamashita, HELIOS2012 Team Description Paper, $RoboCup$2012, CD-ROM(6 pages), Mexico City, Mexico, 2012.