

# 移動ロボットの行動系列を用いた 自己組織化ネットワークによる部屋の認識

○室田 盛道  
大阪大学 基礎工学部

山田 誠二  
東京工業大学 大学院総合理工学研究科

## Recognizing rooms with a mobile robot's behavior sequence

○Morimichi MUROTA  
Osaka University

Seiji YAMADA  
Tokyo Institute of Technology

### 1 はじめに

従来の移動ロボットによる環境認識の研究は、セルラマップに代表されるように、環境の幾何学的構造を高精度で獲得することを目指したものが多く、しかし、精度のよいセンサーを持たない単純な生物でも環境の認識を行っているし、例えば部屋の識別などの重要ではあるが比較的簡単なタスクのためには、必ずしも精度の高い幾何学的地図が必要ではない。また、工学的にも、精度の低いセンサとアクチュエータのみを用いて、環境を認識するロボットを実現することは、コスト面でも意義がある。

また、当然のことながら、ロボットによる環境認識は、ノイズ等に対し頑健でなければならない。さらに、未知環境における探検を考えると、明示的な教師なしに識別が可能になる学習能力も要求される。

以上のような要求を満たす移動ロボットを実機を用いて構築することが、本研究の目的である。換言すれば、精度の良くないセンサーを用いて、どの程度の環境認識ができるのかを探求することを目的とする。本研究では、頑健な識別能力と教師なし学習を実現するために、行動 (behavior) ベースの移動ロボットにより得られる行動系列から、コホーネンの自己組織化ネットワークを用いて、形状の異なる部屋の識別を行うことを目指す。また、本報告では、ロボットが行動する、壁で囲まれた閉領域を「部屋」と呼んでいる。

Nehmzow と Smithers は、直角のコーナーをもつ単純な囲いの中で、移動ロボットを壁沿い移動させ、得られた情報から、自己組織化ネットワークの学習により、コーナーの同定を行なった<sup>6)</sup>。さらに、Smart と Hallam は、Nehmzow のシステムを用いて、追加実験を行ない、その結果とラットの位置認識との類似性を議論している<sup>7)</sup>。彼らが用いた表現は、過去のコーナーの形状 (凹、凸) と壁の長さからなる方向-期間対というものであり、それが自己組織化ネットワークに入力された。そして、学習の結果、コーナーが同定されるようになった。しかし、センサ入力から方向-期間対への変換は、小さな障害物などのノイズに弱いという欠点がある。これに対し、我々は、まずセンサデータを再現性のある行動系列とし、それをさらにチェーンコーディングにより入力ベクトルに変換することで、ノイズに対して頑健な認識を実現する。

Mataric は、ランドマークをノードとするオートマ

トンにより環境を表現した<sup>5)</sup>。その表現は、幾何学的なものよりも頑健であるが、移動ロボット自身が、センサデータからランドマークに抽出し、さらにその同定を行なう必要がある。しかし、一般に、低い精度のセンサを用いてのランドマークの抽出と同定は困難と考えられること等から、Mataric の手法は、我々の目的にはそぐわないものである。

また、辻と李は、ルートに沿って移動したときに得られるシーンを視覚ベースで記憶する方法を提案し、実装した<sup>8)</sup>。移動ロボットは、全方位の定性的表現を記録し、現在のセンサ情報とのマッチングにより、ルート上での位置決定が可能となる。高度な視覚システムをもつ移動ロボットでは、視覚ベースの環境認識は有効であるが、我々は、視覚よりも情報量のはるかに少ない低精度のセンサだけでの環境認識を目指す。

通常ロボティクスにおいて、自己組織化ネットワークは、センサデータをクラス分け (classification) するためだけに用いられる<sup>2)</sup>。そのような応用では、センサデータがそのまま入力ベクトルとして、自己組織化ネットワークに与えられ、その出力 (勝者ノード) が直接的に何かを意味するわけではない。これに対し、本研究では、出力が直接部屋を指示するため、望ましい結果が得られるように、センサデータをいかに適切に入力ベクトルに変換するかという興味深い問題が生じる。

### 2 全体の処理の流れ

全体の処理の流れを Fig. 1 に示す。まず、識別すべき  $n$  種類の部屋を行動ベースの移動ロボットが、壁沿い移動 (wall following) を行って、部屋内部を一周し、そのとき実行された行動系列 (記号列) を得る。このとき、それぞれの部屋に対し、少なくとも一回は壁沿い移動が行われるとし、合計  $m (\geq n)$  個の行動系列が得られたとする。得られた行動系列を多次元ベクトルに変換し、それを自己組織化ネットワークに入力して、学習が行われる。このとき、どの入力ベクトルがどの部屋に対応するかという情報を明示的に与える教師は存在しない。また、 $m$  個の入力ベクトルが繰り返し与えられて、学習が進んで行く。学習が終了すると、新たにいずれかの部屋の行動系列を自己組織化ネットワークに入力し、その競合層の勝者ノードを調べることで、その部屋が以前にたどったどの部屋であるかが認識される。以上の手続きの詳細を、次節以降で説明し

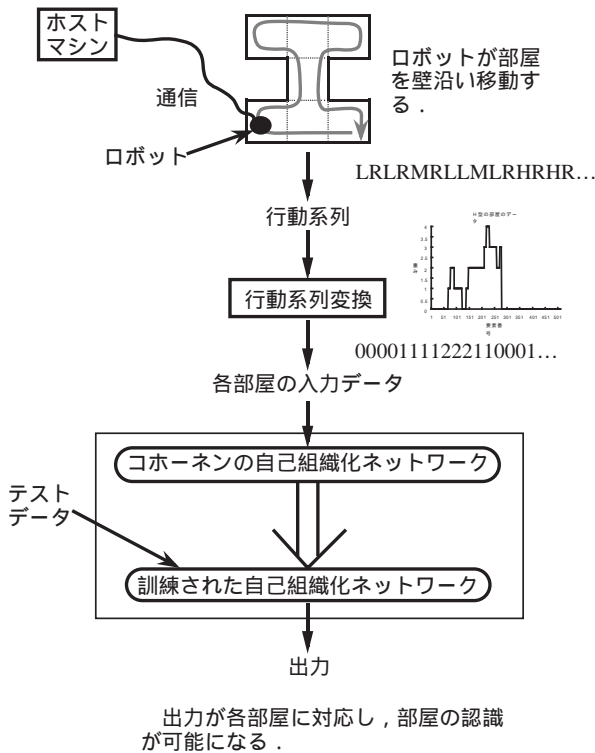


Fig.1 全体の処理の流れ

ていく。

### 3 行動ベース移動ロボットによる壁沿い移動

本研究で用いる移動ロボット ( Fig.2 ) は、マイクロマウス用の小型のもので、左右前方 2 個づつ、正面と斜め前方を向いたの赤外線近接センサを計 4 個もち、さらに慣性型方向センサと移動距離センサも搭載している。赤外線近接センサは壁の認識に、慣性型方向センサはステアリングに、また、移動距離センサは、コーナーを曲がる際の一定距離の移動に用いられる。

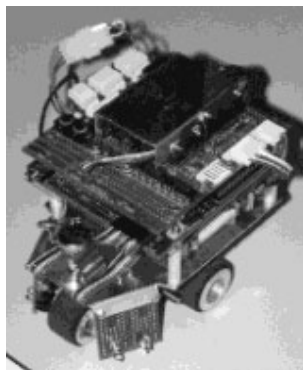


Fig.2 移動ロボット

しかし、赤外線センサの精度は良くなく、またロボットの周囲 20 センチほどの範囲の 4 方向の距離情報という、局所的な情報しか得られない。このような知覚

能力の低い移動ロボットを用いることは、我々の研究目的にそったものである。

また、駆動系は、左右に 2 つの PWM で制御された独立したステップモータによる駆動輪、前後に 2 つの操舵輪がある。操舵輪は、一つの DC サーボモータで駆動される。

移動ロボットは、壁沿い移動という単純な行動を行うが、環境変化への適応性を高めるため、多少の障害物に対してもほぼ同様の移動が可能なが望ましい。そのため、本研究では、Brooks の提案 3) に始まる行動ベース (あるいは、リアクティブプランニング 9) ) による移動ロボットを構築した。行動ベースは、環境変化に対し頑健であり、また精度の低い局所的なセンシングでも、制御が可能である。また、同じ部屋に対する移動軌跡の再現性はそれほどないが、実行された行動系列は再現性が期待できるという好ましい特性がある。

本研究で、移動ロボットは壁沿い移動を行うが、その時に用いられるのが、以下の行動ルールである。このルールは、一種のプロダクションシステムのようなものであるが、条件部がセンサ入力から直接判定可能であり、また結論部がアクチュエータにより直接実行可能であるところに特長がある。よって、行動ルールは、センサ入力に対し迅速に行為を実行できる。なお、ステアリングは、正面方向に向かって絶対的であり、ロボットは時計周りに壁沿い移動をする。

- 行動 A (凹コーナーを曲がる) : *if* 前方の壁が 10cm 以内に近づいて、左 10cm 以内に壁がある *then* 右に 40° スアリングを切る。
- 行動 B (凸コーナーを曲がる) : *if* 左右 5cm、前方 10cm に障害物なし *then* 10cm 前方に進み、左に 40° スアリングを切る。
- 行動 C (壁沿い 1) : *if* 壁に 5cm 以内に近づいた *then* 右に 13.5° ステアリングを切る。
- 行動 D (壁沿い 2) : *if* 壁から 5cm 以上離れた *then* 右に 13.5° ステアリングを切る。

以上の 4 つの行動を適時実行するだけで、かなりスムーズな壁沿い移動が可能となる。

### 4 行動系列の入力ベクトルへの変換

壁沿い移動により各部屋を一周することにより、そのとき実行された前述の行動ルールの系列 ( 行動系列 ) が獲得される。直観的に、この系列は、部屋の形状を反映していると考えられるので、この系列を入力ベクトルに変換して、自己組織化ネットワークで学習されることで、形状の違う部屋の識別が可能になることが期待できる。

行動系列を自己組織化ネットワークに入力するためには、行動系列を多次元ベクトルに変換する必要がある。この際、任意の変換が可能だが、注意すべきは、うまくクラスタリングができるような変換を設定することである。自己組織化ネットワークは、単に入力ベクトル間の ( ユークリッド ) 距離の大小関係を保持した

まま、次元の低い疎な離散ベクトル値に変換するだけなので、元の入力ベクトル空間状で望ましいクラスタリングができていないとよい結果が得られない。つまり、入力の多次元ベクトル空間において1つのグループになって欲しい入力ベクトル値間距離の距離は小さくなるように変換される必要があるし、さらにノイズの影響を吸収できるような変換が望ましい。

そこで、本研究では、部屋の形状が多角形であるという前提の基に、次のような変換を行う。ここでは、入力ベクトルの次元数は、得られた行動系列の長さ  $n$  以上の適当な数  $m$  であり、入力ベクトルは、初期ベクトル値をすべて 0 とする。今、行動系列を  $[r_1, r_2, \dots, r_n](r_i \in \{A, B, C, D\})$ 、入力ベクトル  $I = (v_1, v_2, \dots, v_m)(n \leq m)$  とすると、 $I$  のベクトル値が  $v_1$  から  $v_n$  の順に、以下に示す BI 変換により決定される。なお、この変換は、turning 関数 1)、あるいはチェインコーディングの一種である。

#### < BI 変換 >

1.  $r_i = A$  なら、 $v_i = v_{i-1} + 1$  .
2.  $r_i = B$  なら、 $v_i = v_{i-1} - 1$  .
3.  $r_i = C$  or  $D$  なら、 $v_i = v_{i-1}$  .
4. 上記以外では、 $v_i = 0 (i > n)$  .

例えば、Fig.5(a) は、矩形の部屋の壁沿い移動から得られた行動系列から変換された入力ベクトルである。 $x$  軸と  $y$  軸は、それぞれ入力ベクトルの次元とベクトル値を表す。このグラフにおいて、ゼロでないベクトル値の連続している部分は、部屋全体に対応し、同じベクトル値の連続している部分は、壁に対応し、そして、ベクトル値の変化はコーナーに対応している。また、ベクトル値が 1 増加する変化が凹のコーナーに、1 減少する変化が凸のコーナーに対応する。

移動ロボットによる移動の軌跡とそこで得られた行動系列から BI 変換により変換された入力ベクトルを Fig.3 に示す。Fig.3(a) は、障害物がない部屋であり、Fig.3(b) は、同じ部屋で壁ぎわに障害物をおいた場合である。障害物があるなしの違いはあるが、この 2 つは同じ部屋なので、ロボットにも同一であると認識して欲しい。そのためには、この 2 つの部屋の入力ベクトルの距離が比較的小さくなるように、行動系列から入力ベクトルへの変換を設定する必要がある。BI 変換を用いて得られた入力ベクトルが Fig.3 に示されている。直観的には、Fig.3(a) の影のかかった領域の面積が 2 つの入力ベクトル間の距離を表すが、それが比較的小さいのがわかる。ここで、障害物の存在により、入力ベクトル全体のパターンがシフトされていない点に注意して欲しい。もし 6) で用いられている方向-期間対のような柔軟性のない表現を用いれば、障害物により入力ベクトルパターン全体がシフトしてしまい、2 つの入力ベクトル間の距離が著しく大きくなってしまふ。このように、BI 変換は障害物のようなノイズに

対し頑健であると考えられ、後述する実験によりその有効性が検証される。

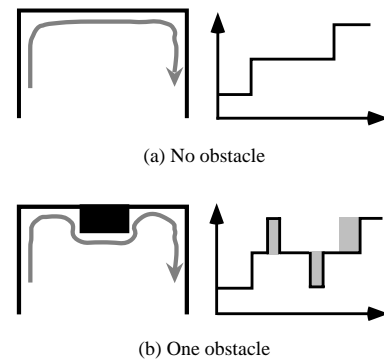


Fig.3 BI 変換の頑健性

## 5 自己組織化ネットワークによる学習

ここでは、コホーネンにより提案された自己組織化ネットワーク 4) について、簡単に触れる。自己組織化ネットワークは、入力層と競合層からなる 2 層ネットワーク (Fig.4) であり、入力層のノード (入力ノード) と競合層のノード (競合ノード) が重みつきリンクで完全結合されている。入力ベクトルが与えられると、入力ノードは入力ベクトルの対応した要素の値をとり、それから競合ノードは入力と重みとの距離を計算し、唯一の勝者ノードを決定する。そして、その勝者ノードの近傍のノードの重みが更新される。

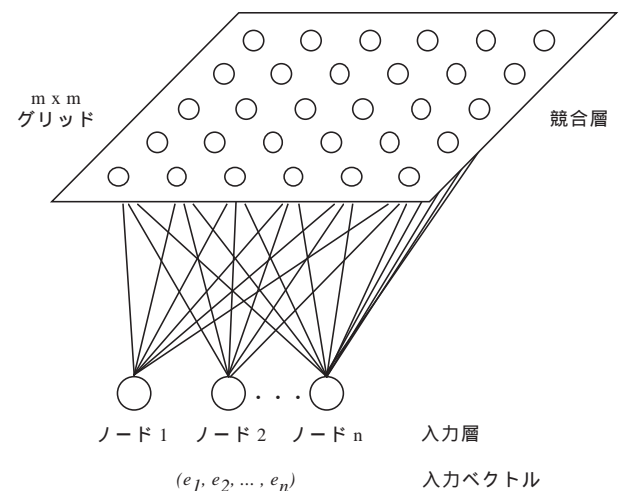


Fig.4 コホーネンの自己組織化ネットワーク

今、入力ベクトルと、入力ノード全体から一つの競合ノード  $u_i$  へのリンクの重みをそれぞれ次の  $E, U_i$  とする。ここで、 $i$  は競合層のノードである。

$$E = [e_1, e_2, \dots, e_n]$$

$$U_i = [u_{i1}, u_{i2}, \dots, u_{in}]$$

自己組織化ネットワークは、まず入力ベクトルに対する各競合層のノードの一致値を計算する。この一致

度は、下式で計算される、重みと入力ベクトルのユークリッド距離である。

$$\|E - U_i\| = \sqrt{\sum_j (e_j - u_{ij})^2}$$

一致値の最も低い(最も一致する)ノードを勝者ノードと呼ぶ。勝者ノードを  $c$  とすると、 $c$  は以下の式を満たす。

$$\|E - U_c\| = \min_i \{\|E - U_i\|\}$$

ここで、最小値はすべての競合ノードから選択される。複数のノードが最小の一致値を持つなら、最小の  $i$  のノードが選ばれる。

勝者ノードが決まると、その近傍の競合ノードの重みが更新される。近傍は、競合層の次元に応じて適当に決められる。例えば、競合層が2次元なら、勝者ノードを重心とする正方形がよく用いられる。近傍のノードの重みの更新式は下式である。なお、 $\alpha$  は、学習率である。

$$u_{ij}^{\text{new}} = u_{ij}^{\text{old}} + \Delta u_{ij}$$

$$\Delta u_{ij} = \begin{cases} \alpha(e_j - u_{ij}) & : \text{ノード } i \text{ が近傍にある} \\ 0 & : \text{それ以外} \end{cases}$$

以上の処理を入力ベクトルが与えられる毎に繰り返す。また、学習率と近傍のサイズは、学習が進むにつれて減少させる場合が多い。

本実験では、入力層は、部屋を壁沿い移動で一周して得られる行動系列の長さより大きな適当なサイズが採られる。そして、競合層のノードは、1次元配列としている。競合ノード数を一定にしておけば、計算量を抑えたままで競合層の次元を上げていくことが可能であるが、後述する実験において、2次元と3次元で試してみた結果、識別精度の向上はみられなかった<sup>1</sup>。

## 6 学習フェーズと試験フェーズ

自己組織化ネットワークの更新がほとんど無くなり、学習が収束してくると、いくつかの限られた競合ノードだけが頻りに勝者になるという状態が生じる。それらの勝者ノードは、部屋に対応していると考えられるので、それらを  $r$  ノードと呼ぶ。更新が無くなると学習は終了し、 $r$  ノードが得られる。得られた  $r$  ノードの個数が部屋数に対応している。

次に、判定すべき部屋の入力ベクトルが、これまで学習された部屋のどれと同じであるかを同定する必要がある。そこで、入力ベクトルと、各  $r$  ノードの重みとのユークリッド距離が計算され、その距離が最小の部屋と同定される。

## 7 実機による実験

前述の移動ロボットを用いて、実機による実験を行った。ホスト計算機は、IBM-PC/AT 互換機 (Intel DX4 100MHz, RAM32M) を用い、Linux 上で C++を用い

<sup>1</sup> というよりも、1次元でも完全に認識できたので、それ以上の精度向上はなかった。また、1次元で端のノードを結ぶトラス状にした場合としない場合も実験を行なったが、有意な差はなかった

て実装をした。また、ホスト計算機と移動ロボットは、RS232C を介して通信する。壁沿い移動を行なうための処理は、移動ロボット上で行ない、得られた行動系列をホスト計算機に送り、自己組織化ネットワークでの学習と部屋の認識の処理は、ホスト計算機上で行なう。

### 7.1 各種設定

白いプラスチックの板で作った Fig.5 に示すような7種類の部屋を、移動ロボットにそれぞれ10周づつ壁沿い移動させ、行動系列を獲得させた。ただし、今回の実験では、部屋を一周したことは人間が教えており、さらに、壁沿い移動開始地点は部屋の種類によって固定している。なお、実装はまだだが一周の判定は、dead reckoning で可能である。

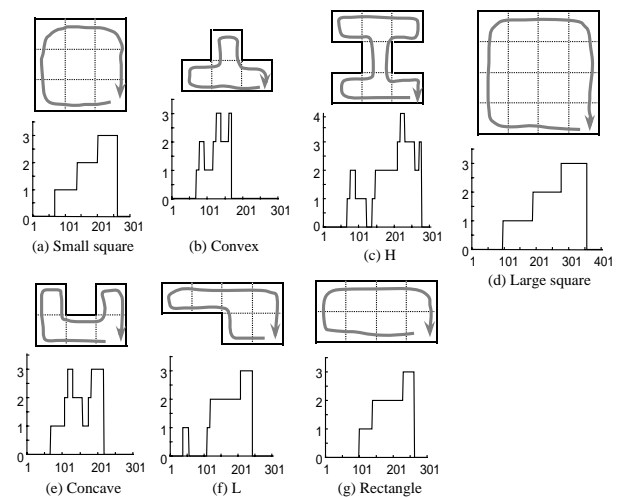


Fig.5 7つの部屋とその入力ベクトル

移動ロボットは、各部屋を1周するのに1400程の行動を実行するが、これをそのまま入力ベクトルとするとサイズが大き過ぎるので、行動  $C, D$  の壁面を移動している系列の比を保持し、かつ行動  $A, B$  を削除しないように、行動系列を  $1/4$  に間引いた。

学習における総訓練回数は、4200である。まず、移動ロボットに、7つの各部屋を10回壁沿い移動させ、70の行動系列を得た。そして、そのうち、各部屋について一つづつ選んだ7個を訓練例として、それぞれ600回づつ順不同で自己組織化ネットワークに与えた。残りの各部屋につき9つの行動系列は、後に試験例として用いられる。

自己組織化ネットワークの入力ノード数を520(次元)、競合層を1次元で32ノードでトラス状とし、BI変換で行動系列を入力ベクトルに変換した。また、 $t$  回目の学習における、学習率  $\alpha$  と近傍の幅  $d$  を以下の式を用いて、学習進行とともに変化させた。また、競合層のノードの重みは、 $1.5 \pm 0.15$  でランダムに設定した。その結果、入力ノードと勝者ノードのユークリッド距離の自乗である  $D$  が、Fig.6のように収束した。

$$\alpha_t = 0.2 \left( 1 - \frac{t}{4200} \right) \quad d_t = 11 \left( 1 - \frac{t}{4200} \right)$$

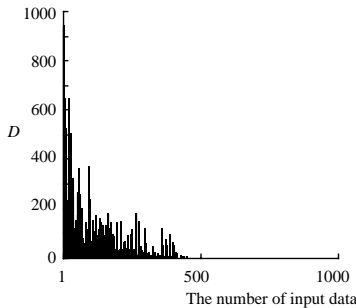


Fig.6 学習の収束

## 7.2 実験結果

先の学習で得られた  $r$  ノードを用いて、以下の実験結果を得た。

### 障害物のない部屋の認識

障害物のない部屋に関しては、前述の試験例 63 個のすべてについて、正しい認識がなされた。これらの行動系列は、障害物はないものの、行動ルールが一部欠けていたりするため、まったく同じものではない。

### 障害物のある部屋の認識

さらに、部屋の壁面に障害物を置いた 9 つの場合につき、同定を行なった。その結果、5 つだけが正しく認識できた。正しく認識できた部屋とできなかった部屋の例を Fig.7 に示す。

Fig.7(a) では、壁に接した、一つの障害物が、2 つの小さな障害物に関しては認識がうまくいっている。しかし、Fig.7(b) のように、2 つの大きな障害物や、壁に対し斜めになっている障害物がある場合は認識に失敗していることがわかる。

障害物が多く存在する場合、それらの部屋を同じ部屋と認識すべきか否かは、ロボットというエージェントがどのような目的のために部屋の認識を行なうのかに依存する。また、本研究では、部屋の輪郭の形状をパターン認識して同定しているわけだが、このような輪郭の形状での識別が、エージェントの目的に適したものか否かはわからない。というより、現在のところエージェントは、学習の目的を持っていないのである。今後は、エージェントの学習の目的に応じた部屋の識別が重要である。

## 8 部分情報による部屋の認識

これまで述べてきた実験では、移動ロボットが部屋を一周した後に、その一周分の行動系列を入力として、部屋の判定を行なっていた。しかし、特徴的な情報が含まれていれば、完全に一周しなくても、部分的に壁沿い移動して得られた行動系列から部屋の認識を行なうことも可能と考えられる。また、前の実験では、壁沿い移動の開始位置を部屋ごとに固定する必要があった。そこで、これらの制約を取り除くべく、部分的な

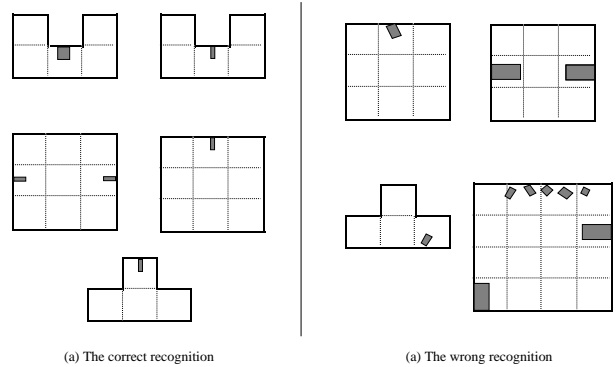


Fig.7 障害物のある環境

情報から部屋の判定を行なう手続きを考案し、予備的な実験を行なった。

判定の手続きを、以下に示す。最も長い壁が先頭になるように入力ベクトルをシフトして比較し、 $r$  ノードとの距離がしきい値以下になったら同定したと見なすという方法である。

### < 部分情報からの部屋の判定手続き >

1. 最も長い壁が先頭になるように行動系列をシフトして、自己組織化ネットワークで学習させる。
2. 移動ロボットから、逐次的に部分情報を受け取り、コーナーの検出毎に行動系列を BI 変換により入力ベクトルに変換し、下記のシフトを行なう。
  - (a) すべての履歴の中で最も長い壁を見つけ、それを先頭にそれから以後に得られたベクトルを残し、他のベクトル値は 0 として、部分入力ベクトルをつくる。
  - (b) もし最も長い壁が複数ある場合は、それぞれについて (a) のシフトを行ない、部分入力ベクトルをつくる。
3. 2. で得られた部分入力ベクトルそれぞれと 1. で得られたそれぞれの  $r$  ノードの重みベクトルとの距離を計算し、それがしきい値  $\zeta$  以下なら、その  $r$  ノードの部屋と判定。

最も長い壁を先頭にするようなシフトを行なうため、壁沿い移動が部屋の壁面のどの位置から開始されてもよい。問題は、しきい値  $\zeta$  の決定であるが、ここでは実験をしながら経験的に決めるとする。

現在、部分情報による判定は、インプリメント中である。本報告の発表時には、結果を報告できる予定である。

## 9 まとめ

精度の低いセンサをもつ行動ベースの移動ロボットが壁沿い移動することによって、得られる行動系列を BI 変換により入力ベクトルに変換し、コホーネンの自己組織化ネットワークによる教師なし学習を行ない、

部屋の認識を行なう方法を提案した。また，マイクロマウスにより実機を用いた実験を行ない，我々の手法の有効性を検証した。さらに，部分的に部屋を壁沿い移動するだけで，判定が可能である手続きを考案した。

また，今後の課題として，オフィス環境での部屋の認識，部屋の文節化，そして壁沿い移動以外の行動による部屋の識別等が考えられる。

## 参考文献

- 1) Esther M. Arkin, L. Paul Chew, Daniel P. Huttenlocher, Klara Kedem, and Joseph S. B. Mitchell. An efficiently computable metric for comparing polygonal shapes. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, Vol. 13, No. 3, pp. 209–216, 1991.
- 2) R. D. Berns and U. Zachmann. Reinforcement learning for the control of an autonomous mobile robot. In *1992 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1808–1815, 1992.
- 3) Rodney A. Brooks. A robust layered control system for a mobile robot. *IEEE Transaction on Robotics and Automation*, Vol. 2, No. 1, pp. 14–23, 1986.
- 4) Teuvo Kohonen. *Self-Organization and Associative Memory*. Springer-Verlag, 1989.
- 5) Maja J. Mataric. Integration of representation into goal-driven behavior-based robot. *IEEE Transaction on Robotics and Automation*, Vol. 8, No. 3, pp. 14–23, 1992.
- 6) Ulrich Nehmzow and Tim Smithers. Map-building using self-organizing networks in really useful robots. In *Proceedings of the First International Conference on Simulation of Adaptive Behavior*, pp. 152–159, 1991.
- 7) Willian D Smart and John Hallam. Location recognition in rats and robots. In *Proceedings of the Third International Conference on Simulation of Adaptive Behavior*, pp. 174–178. The MIT Press, 1994.
- 8) Saburo Tsuji and Shing Li. Memorizing and representing route scenes. In *Proceedings of the Second International Conference on Simulation of Adaptive Behavior*, pp. 225–232, 1992.
- 9) 山田誠二. リアクティブプランニング. 人工知能学会誌, Vol.8 No.6, pp. 729–735. 1993.