

# 状態行動地図のためのベクトル量子化手法の圧縮率向上

Efficiency Enhancement of Vector Quantization Method for State-Action Map Compression

上田 隆一 正 新井 民夫 坂本 浩平 実川 達明 竹下 和考 (東京大学)  
正 梅田 和昇 正 大隅 久 菊地 敏文 小村 正樹 (中央大学)

\*Ryuichi UEDA, Tamio ARAI, Kohei SAKAMOTO, Yoshiaki JITSUKAWA and Kazutaka TAKESHITA,  
The University of Tokyo, all-araibo@prince.pe.u-tokyo.ac.jp  
Kazunori UMEDA, Hishashi OSUMI, Toshifumi KIKUCHI, and Masaki KOMURA,  
Chuo University

We have proposed a method that uses vector quantization to compress control policies that are obtained by dynamic programming. In this paper, we apply vector quantization again to compression of an already vector quantized control policy so as to enhance the compression ratio. This method is evaluated with the puddle world task and with a task for RoboCup four legged robot league. Compression ratio of vector quantized control policies for these tasks can be enhanced by the method.

*Key Words:* vector quantization, dynamic programming, RoboCup, puddle world tasks

## 1. 緒言

近年、動的計画法 (dynamic programming, DP) [1] が様々なロボットの行動設計に利用されている [5, 6]. 筆者らは、ロボットのサッカー行動を設計するために DP を用いてきた [9]. 筆者らの場合、ロボットや他の物体の位置、姿勢を示すパラメータで張られる空間 (状態空間) を格子状に離散化して DP を適用している. 各離散状態に対し、価値反復 (DP の実装法の一つ) によって、その状態でロボットが取るべき行動を求める.

この方法は、確実に解が得られる反面、パラメータ数が多いと離散状態数が非常に多くなる. 最近の計算機では、かなりの状態数 (例えば [11] では 12 億) に対して DP を実行できる. しかし、ロボットのメモリ容量は必ずしも大きくなく、計算結果、すなわち各離散状態で選択すべき行動が記述されたルックアップテーブル (以下、状態行動地図、あるいは地図と呼ぶ) が実装できない場合がある. 関数近似等を用いて状態数を節約して DP を適用する方法もあるが、計算量、解の近似精度、あるいは収束性のいずれかにおいて問題が生じる恐れがある [2].

そこで、筆者らは格子状の離散状態空間で得られた DP の結果を、ベクトル量子化 (vector quantization, VQ) [4] によって圧縮し、ロボットに実装する方法を提案してきた [11, 12]. 本稿では、この手法によって得られた地図を再度ベクトル量子化することで、さらに容量を減らすことを試み、標準問題 (水溜り問題 [7] とロボカップ 4 足ロボットリーグ) で評価を行う.

## 2. 状態行動地図のベクトル量子化圧縮

### 2.1 マルコフ決定過程

本稿では、次の記号で定式化できる問題を扱う.

- 行動セット  $\mathcal{A} = \{a_i | i = 1, 2, \dots, N_a\}$ : エージェント (ロボット) は、このセットから行動を選択し続けてなんらかのタスクを行う.
- 離散状態空間  $\mathcal{S} = \{s_j | j = 1, 2, \dots, N_s\}$ : エージェントはある時点で  $\mathcal{S}$  中のある状態  $s_j$  に属す. 離散状態のいくつかは終端状態と呼ばれる. 終端状態の集合を  $\mathcal{S}_F$  で表す.
- 状態遷移確率  $\mathcal{P}_{ss'}^a$ : エージェントが状態  $s$  で行動  $a$  を選択後、 $s'$  に到達する確率を示す.

- 報酬  $\mathcal{R}_{ss'}^a \in \mathbb{R}$ : 状態  $s$  で行動  $a$  を選択後、 $s'$  に到達した場合の評価. これは、タスクに応じて設計されるものである.

これらの条件で、エージェントが任意の初期値からある終端状態に到達するまでの報酬の合計を大きくするための方策を、なるべく小さい容量のメモリで表現するという問題を扱う.

### 2.2 状態行動地図の作成

$\forall s \in \mathcal{S}$  から  $\exists s_F \in \mathcal{S}_F$  に到達するまでに得られる報酬の合計の期待値を考える. この期待値 (その状態の価値と呼ばれる) の最大値を与える最適状態価値関数  $V^* : \mathcal{S} \rightarrow \mathbb{R}$  は、次のベルマン方程式 [1, 8]:

$$V^*(s) = \max_a \sum_{s'} \mathcal{P}_{ss'}^a [\mathcal{R}_{ss'}^a + V^*(s')] \quad (1)$$

を満たす. 価値反復は、状態価値関数  $V$  を適当に初期化し、

$$V(s) \leftarrow \max_a \sum_{s'} \mathcal{P}_{ss'}^a [\mathcal{R}_{ss'}^a + V(s')] \quad (2)$$

という手続きを繰り返して  $V^*$  を求めるアルゴリズムである.  $V^*$  を求めた後、各状態の最適な行動は、

$$\pi^*(s) \leftarrow \arg \max_a \sum_{s'} \mathcal{P}_{ss'}^a [\mathcal{R}_{ss'}^a + V^*(s')] \quad (3)$$

で求めることができる. 状態空間が格子状に離散化されているとき、 $\pi^*$  の計算結果は、各離散状態に最適な行動が記述されたルックアップテーブル (状態行動地図) としてロボットに実装できる. この地図の容量は、

$$I = N_s \log_2 N_a [\text{bit}] \quad (4)$$

と計算できる. これは、 $N_a$  個の行動を識別するために  $\log_2 N_a [\text{bit}]$  必要で、地図には  $\log_2 N_a [\text{bit}]$  のデータが  $N_s$  個並ぶという計算である.

### 2.3 圧縮の原理

まず、離散状態  $s_j (j = 1, 2, \dots, N_s)$  を、同じ個数ずつ集団  $N_v$  個の集団に分け、各集団に 1 から  $N_v$  まで番号付けする操作を考える. また、各集団内の  $m = N_s/N_v$  個の離散状

態にも 1 から  $m$  まで番号付けする．これらの操作をそれぞれ添え字の変換

$$\varphi(j) = j_\varphi \quad (5)$$

$$\psi(j) = j_\psi \quad (6)$$

で表す．この変換は，状態  $s_j$  を  $j_\varphi$  番目の集団に所属させ， $j_\psi$  番目の要素にすることを意味する．また，逆変換を

$$\tau(j_\varphi, j_\psi) = j \quad (7)$$

と定義する．このとき， $j_\varphi$  番目の集団の順序列  $s_{\tau(j_\varphi, 1)}, s_{\tau(j_\varphi, 2)}, \dots, s_{\tau(j_\varphi, m)}$  に対し， $\tau$  が与える状態番号に対する最適な行動を並べた

$$\mathbf{v}_{j_\varphi} = (\pi^*(s_{\tau(j_\varphi, 1)}), \pi^*(s_{\tau(j_\varphi, 2)}), \dots, \pi^*(s_{\tau(j_\varphi, m)}))$$

をベクトルと呼ぶこととする．また，全ベクトルの集合  $\mathcal{V} = \{\mathbf{v}_{j_\varphi} | j_\varphi = 1, 2, \dots, N_v\}$  を定義する．

次に， $\mathcal{V}$  から「似た」ベクトルを  $N_c$  個の集団（クラス） $K_1, K_2, \dots, K_{N_c}$  に分類（クラスタリング）する．この結果を，ベクトルの添え字  $j_\varphi = 1, 2, \dots, N_v$  からクラスターの添え字（記号は  $j_{\varphi\omega}$  を用いる）への変換

$$\omega(j_\varphi) = \omega \circ \varphi(j) = j_{\varphi\omega} \quad (8)$$

で表す．そして，各クラス  $K_{j_{\varphi\omega}}$  のベクトルを，一つの代表ベクトル  $\mathbf{c}_{j_{\varphi\omega}}$  に置き換える．[11, 12] では，上記の「似た」を数値化するために  $V^*$  を用いている．また，クラスタリングには，一対法による手法 [3] が用いられている．

この操作で，変換  $\varphi, \psi, \omega$  と  $\mathcal{C}$  を用いて， $s_j$  における行動を

$$\pi^*(s_j) = \mathbf{c}_{(j_{\varphi\omega}, j_\psi)} = \mathbf{c}_{(\omega \circ \varphi(j), \psi(j))} \quad (9)$$

で表現できるようになる．また，この式を図示すると Fig.1 のようになる． $\varphi, \psi, \omega$  と  $\mathcal{C}$  による方策表現を，圧縮状態行動地図（圧縮地図）と呼ぶ．圧縮地図の容量は，

$$I_{VQ} = N_v \log_2 N_c + m N_c \log_2 N_a [\text{bit}] \quad (10)$$

で計算される．上式左辺の第 1 項は， $\omega$ （以後，変換表と呼ぶ）の記述に必要な容量である．第 2 項は， $\mathcal{C}$ （符号帳と呼ばれる [4]）の表現に必要な容量である． $\varphi, \psi$  に関しては，表ではなく簡単な数式で表現するとして，容量には加えない．

#### 2.4 圧縮地図のベクトル量子化

前節までの方法では，変換表や符号帳の冗長性を省くことは行われていない．そこで本稿では，圧縮地図の変換表，符号帳をそれぞれ VQ で可逆圧縮し，さらにメモリ容量の小さい圧縮地図（2 段階圧縮地図）を作成することを新たに試みる．可逆圧縮とするのは，不可逆圧縮では圧縮地図をさらに歪ませることになってリスクが大きいためである．

変換表  $\omega$  と符号帳  $\mathcal{C}$  をベクトル量子化すると，それぞれが変換表と符号帳で表されるようになる．変換表  $\omega$  ( $N_v$  個の代表ベクトル番号の並び) を圧縮したときの変換表と符号帳をそれぞれ  $\omega_\omega, \mathcal{C}_\omega$  とする．また，符号帳  $\mathcal{C}$  ( $m N_c$  個の行動番号の並び) に対しても，変換表  $\omega_c$  と符号帳  $\mathcal{C}_c$  を考える．

2 段階圧縮地図は，Fig.2 のように図示できる．変換表  $\omega$  への入力  $j_\varphi$  は，二つの添え字に変換され， $\mathcal{C}_\omega$  から添え字  $j_{\varphi\omega}$  を得るために用いられる．また，符号帳  $\mathcal{C}$  への入力  $j_{\varphi\omega}$  と  $j_\psi$  は，二つの添え字に変換され， $\mathcal{C}_c$  から行動を得るために用いられる．

このとき，変換表について  $m_\omega$  次元の代表ベクトル  $N_{c_\omega}$  個で表現できたとし，符号帳  $\mathcal{C}$  について  $m_c$  次元の代表ベクトル  $N_{c_c}$  個で表現できたとすると，2 段階圧縮地図の容量は次のようになる．

$$I_{DLVQ} = N_v / m_\omega \log_2 N_{c_\omega} + m_\omega N_{c_\omega} \log_2 N_c + m N_c / m_c \log_2 N_{c_c} + m_c N_{c_c} \log_2 N_a [\text{bit}] \quad (11)$$

右辺第 1 項から順に， $\omega_\omega, \mathcal{C}_\omega, \omega_c, \mathcal{C}_c$  の容量である．

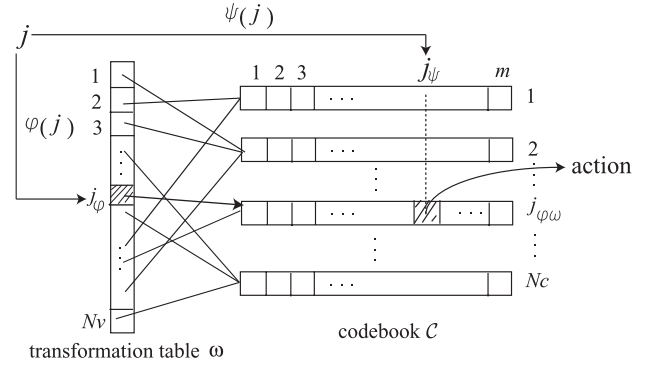


Fig.1 Structure of a compressed map

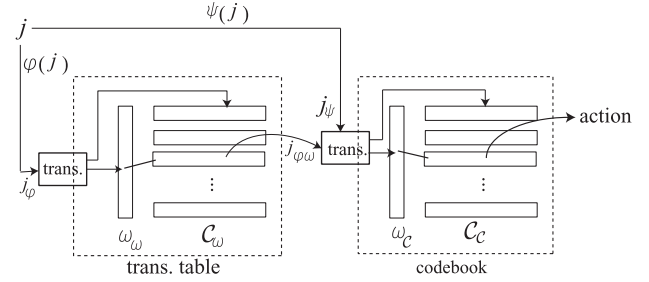


Fig.2 Structure of double layer VQ

### 3. 水たまり問題での評価

#### 3.1 水たまり問題

水たまり問題（puddle world task）は，学習アルゴリズムの評価を行うために用いられる標準問題の一つである [7]．Fig.3(a) は puddle world と呼ばれる環境である．この環境で，エージェントは  $xy$  平面上の点として定義される．エージェントは，右上にあるゴールを目指す．その際，水たまりを通ると罰が課される．評価されるアルゴリズムは，罰と歩数を最小にするような行動則を見つけることを要求される．

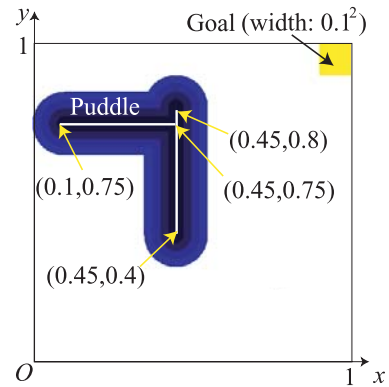


Fig.3 Puddle world

Fig.3 に見られるように，この環境の大きさは  $1 \times 1$  である．水たまりの中心は，二本の線分から成り立っており，水たまりは各線分から幅 0.1 で広がっている．本稿では，エージェントは 4 種類の行動（上下左右）が選択可能とする．各行動一回で，エージェントは平均 0.05 だけ直進する．ただし，行動後に到達する地点は，標準偏差 0.01 でばらつくこととする．この環境の周囲には壁があり，エージェントはその壁を越えようとするとき，壁との衝突点で止まることとする．各行動には一回につき  $-1$  の報酬（罰）が与えられる．また，

水たまりに入ると、 $-400r$  が罰として与えられる。ここで  $r$  は、水たまりの縁のうち最も近い地点からの距離である。

### 3.2 評価の方法

エージェントに行動させてゴールまでの罰の積算を求めることを繰り返し、その平均値で地図の評価を行う。この平均値は、エージェントの初期状態を  $xy$  平面から一様を選んだとき、

$$J[\pi] = \int_{\mathcal{X}} V^{\pi}(x) dx / \int_{\mathcal{X}} dx, \quad (12)$$

の近似となる。本稿では、 $10^6$  個の点  $(x, y) = ((n - 0.5)10^{-3}, (m - 0.5)10^{-3})$  ( $n, m = 1, 2, \dots, 1000$ ) を初期状態として選んだ。

### 3.3 非圧縮地図と圧縮地図、2段階圧縮地図の比較

$x, y$  それぞれについて、区間  $[0, 1]$  を均等に分割して正方形の離散状態を作成し、価値反復を用いて状態行動地図を作成するアルゴリズムを実装した。区間  $[0, 1]$  の分割数を  $\sqrt{N_s}$  (離散状態数は  $N_s$  となる) で表すとき、 $\sqrt{N_s} = 10, 20, 40, 100, 200, 400$  の6種類に対して状態行動地図を作成したところ、Fig.4 のようになった。

これらの地図(非圧縮地図)と圧縮地図の比較を、上記の評価方法を用いて次のように行う。

1. 各非圧縮地図の罰の積算  $J$  を求める。
2. その非圧縮地図よりも  $\sqrt{N_s}$  が10大きい非圧縮地図を作成し、この非圧縮地図から様々な圧縮率の圧縮地図を作成する。
3. 圧縮地図について罰の積算  $J$  と式(10)で計算される容量を求め、非圧縮地図の  $J$  よりも罰の積算が小さく、容量が最小の圧縮地図を選ぶ。
4. 選んだ圧縮地図を2段階圧縮地図にして容量を式(11)で計算する。

この比較の結果は、Table 1 のようになった。また、選んだ圧縮地図を Fig. 5 に示す。これらの圧縮地図の2段階圧縮地図も Fig. 5 と同じものとなる。表のように、評価に用いたどの非圧縮地図に対しても、それより罰の積算が小さく、容量も小さい圧縮地図が存在した。また、それらの圧縮地図を2段階圧縮地図にすると、いずれも容量がさらに小さくなった。

Table 1 Comparison on the puddle world task

$\sqrt{N_s}$	uncomp. maps		comp. maps		
	$I$ [k bit]	$-J$	$I_{VQ}$ [k bit] ( $I_{VQ}/I$ )	$I_{DLVQ}$ [k bit] ( $I_{DLVQ}/I$ )	$I_{DLVQ}$ / $I_{VQ}$
10	0.20	21.29	0.16 (0.80)	0.14 (0.70)	0.88
20	0.80	20.72	0.45 (0.56)	0.37 (0.46)	0.81
40	3.2	20.51	1.1 (0.34)	0.76 (0.24)	0.69
100	20	20.35	7.0 (0.35)	4.1 (0.20)	0.58
200	80	20.30	29 (0.36)	18 (0.22)	0.62
400	320	20.29	65 (0.20)	47 (0.15)	0.73

## 4. ロボカップ環境での評価

本章では [12] の実機実験に使用された圧縮地図を2段階圧縮地図にして、圧縮率を比較する。

### 4.1 ロボカップ4足ロボットリーグ

[12] の実験は、ロボカップ4足ロボットリーグ [10] に基づいて行われた。ロボットは Fig. 6 に見られる自律型4脚ロボット ERS-210 が使用された。また、実験に用いたサッカーフィールドの広さは、Fig.7に見られるように  $4.2[m] \times 2.7[m]$  であった。ERS-210 は各脚と頭部に3自由度を有する多自由度ロボットで、計算資源として MIPS 192 MHz と 32 MB の DRAM が内蔵されている。主要なセンサは頭部の CMOS カメラである。

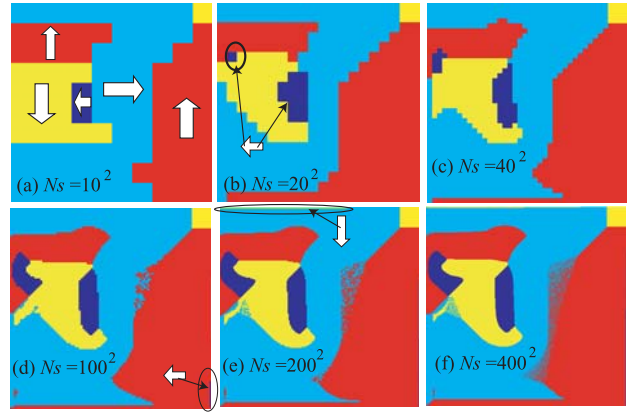


Fig.4 State-action maps

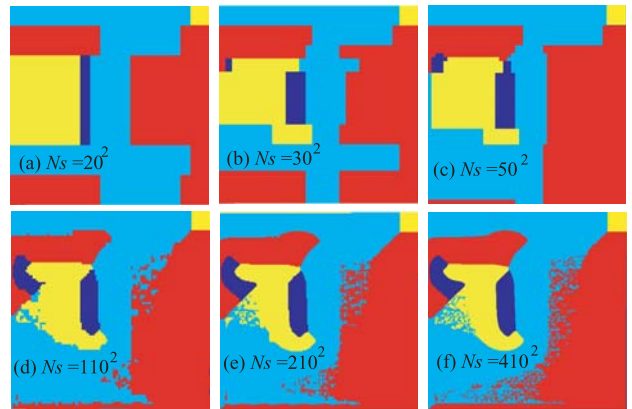


Fig.5 Vector quantized state-action maps

## 4.2 タスク

ロボットが静止したボールに接近するタスクの地図を作成する。ロボットには、ボールをカメラで常に捉えられるよう行動することと、ボールの位置まで来たときに相手ゴールの方向を向いていることが要求された。また、行動の評価は、ロボットが所定の位置に達するまでの歩数の少なさで行われた。ロボットは、移動量と回転量が固定の39種類の行動を用いて移動する。

ロボットの状態は Fig. 7 のように  $(x, y, \theta)$  で表される。 $(x, y)$  はロボットのフィールド上での位置、 $\theta$  は向きである。ボールの位置は  $(x, y)$  を原点とする極座標  $(r, \varphi)$  で表す。 $r$  はロボットとボール間の距離、 $\varphi$  はロボットからのボールの方向で、Fig. 7 のように定義される。したがって、状態は  $(x, y, \theta, r, \varphi)$  で表現される。各変数の定義域と離散化方法は、Table 2 のように設定する。 $r$  は各区間の幅を変えて9つに離散化する。各区間の幅はロボットから近い3区間が  $100[mm]$  で、4区間めからは幅を  $100[mm]$  ずつ増加させる。このとき、離散状態数  $N_s$  は、765,450 となる。

Table 2 Domain and discretization of state space

Variables	domain	# of divisions	width
$x[mm]$	$[-2100, 2100]$	21	200
$y[mm]$	$[-1350, 1350]$	15	180
$\theta[deg]$	$[-180, 180]$	18	20
$r[mm]$	$[150, 3150]$	9	100 - 700
$\varphi[mm]$	$[-75, 75]$	15	10

次の二つの条件を共に満たす状態が含まれる離散状態を終端状態とする。



Fig.6 ERS-210 (made by SONY)

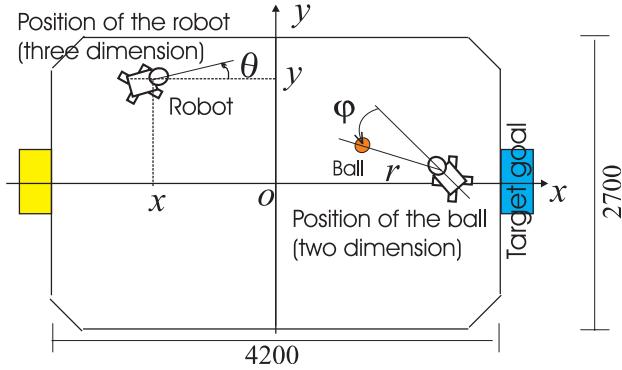


Fig.7 The shape of field and state variables

- $-35 \leq \varphi < 35[\text{deg}]$  かつ  $150 \leq r < 250[\text{mm}]$
- $-40 \leq \theta < 40[\text{deg}]$  または  $-40 \leq \theta_G < 40[\text{deg}]$

ここで、 $\theta_G$  はロボット座標系からみたゴールの方向であり、ロボットがゴール ( $xy$  座標系における点  $(2100, 0)$ ) と正対しているとき  $\theta_G = 0$  である。また、ボールが壁際において上記の条件でロボットがボールに近づけないときは、 $\theta$  の条件を緩和する。

#### 4.3 非圧縮地図と圧縮地図、2段階圧縮地図の比較

前章同様、非圧縮地図と圧縮地図、2段階圧縮地図の比較を行った。[12] の実機実験に用いられた圧縮地図に VQ を適用した。この圧縮地図は、実機に実装では非圧縮地図と同等に機能するものであり、容量が非圧縮地図の 1.4% まで圧縮されたものである。Fig.8 は、この圧縮地図で得られたロボットの行動の例である。

この圧縮地図を 2 段階圧縮地図にしたところ、Table 3 のような結果が得られ、圧縮地図よりも 87%、もとの非圧縮地図の 1.2% まで容量が圧縮できた。水たまり問題と比べて行動の種類が 39 と多かったが、それでも符号表  $C$  には冗長性が残っており、再度 VQ を適用して効果が得られた。

Table 3 Comparison on the puddle world task

	uncomp.	VQ	DPVQ
volume	4,045.7[kbit]	57.0[kbit]	49.7[kbit]
comp. ratio	1	0.014	0.012

## 5. 結言

ベクトル量子化 (VQ) で圧縮された状態行動地図をさらに VQ で圧縮し、圧縮率を向上させることを、水たまり問題とロボカップのタスクの状態行動地図に対して試みた。結果、水たまり問題では、圧縮地図の容量をさらに 60% から 90% 削減可能であった。また、ロボカップのタスクでは、1.4% まで圧縮されている地図の容量をさらに 0.2% 削減し、1.2% まで圧縮することが可能であった。

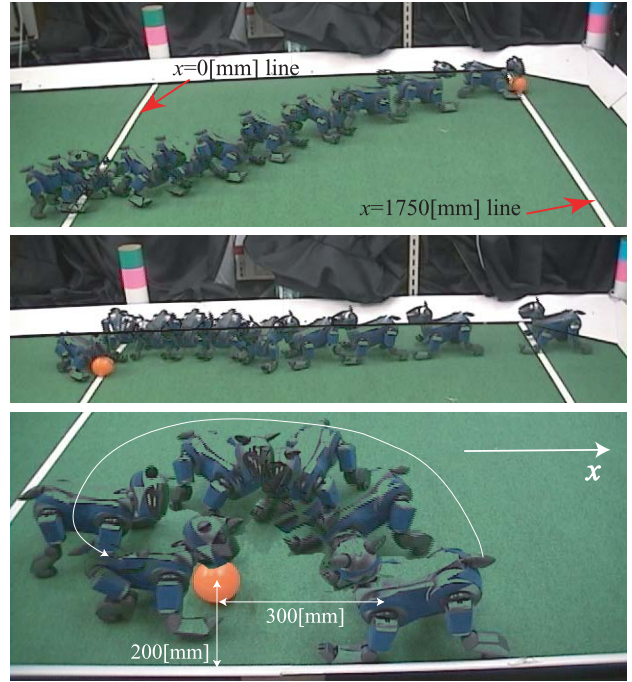


Fig.8 Behavior of the robot with the compressed map in the experiments [12]

## 文献

- [1] Bellman, R.: *Dynamic Programming*, Princeton University Press, Princeton, NJ, 1957.
- [2] Boyan, J. A. and Moore, A. W.: "Generalization in Reinforcement Learning: Safely Approximating the Value Function," in *Advances in Neural Information Processing Systems* 7, pp. 369-376, 1995.
- [3] Equitz, W. H.: "A New Vector Quantization Clustering Algorithm," *IEEE Trans. Acoust. Speech Signal Process*, Vol. 37, No. 10, pp. 1568-1575, 1989.
- [4] Gersho, A. and Gray, R. M.: *Vector Quantization and Signal Compression*, Kluwer Academic Publishers, Boston, MA, 1992.
- [5] Hu, H. and Brady, M.: "Dynamic Global Path Planning with Uncertainty for Mobile Robots in Manufacturing," *IEEE Trans. on Robotics and Automation*, Vol. 13, No. 5, pp. 760-767, 1997.
- [6] Roy, N. and Thrun, S.: "Coastal Navigation with Mobile Robots," in *Advances in Neural Information Processing Systems*, pp. 1043-1049, 1999.
- [7] Sutton, R. S.: "Generalization in Reinforcement Learning: Successful Examples Using Space Coarse Coding," in *Neural Information Processing Systems*, pp. 1038-1044, 1996.
- [8] Sutton, R. S. and Barto, A. G.: *Reinforcement Learning: An Introduction*, The MIT Press, Cambridge, MA, 1998.
- [9] Ueda, R., et al.: "Real-Time Decision Making with State-Value Function under Uncertainty of State Estimation," in *Proc. of ICRA*, 2005.
- [10] 大橋健: "4 足ロボットリーグのとりくみ", 日本ロボット学会誌, Vol. 20, No. 1, pp. 45-46, 2002.
- [11] 小林祐一他: "実時間性と観測コストを考慮した四脚ロボットのサッカー行動設計", 日本ロボット学会誌, Vol. 21, No. 7, pp. 802-810, 2003.
- [12] 上田隆一他: "ベクトル量子化による決定論的方策地図の不可逆圧縮", 日本ロボット学会誌, Vol. 23, No. 1, pp. 104-112, 2005.