機械学習における特徴抽出と分類に関する研究

宮本，行庸

(Degree)
博士（工学）

(Date of Degree)
2001-03-31

(Date of Publication)
2009-02-19

(Resource Type)
doctoral thesis

(Report Number)
甲2228

(URL)
https://hdl.handle.net/20.500.14094/D1002228

博 士 論 文

機械学習における特徴抽出と

分類に関する研究

平成 13 年 1 月

神戸大学大学院自然科学研究科

宮 本　行 庸

目 次

図 目 次

表 目 次

第 1 章

緒論

近年の急速な情報化社会の発展にともない，現実世界に多くの情報が氾濫している．これらの情報の中から，ある事例と別の事例が類似しており，概念的に同じであると判断することは非常に困難であり，その必要性が叫ばれている．また，これらの要求は，なるべく人間の手を介さないで，自動的に行われることが望ましい．人工知能の研究はこれらの要求を実現可能であり，とりわけ機械学習は事例をもとに複雑な情報を体系化することを目的とした研究分野である．

機械学習においては，事例を表現するための特徴がその学習精度を大きく左右する．また，機械学習の技術は計算機上に実装されるため，現実世界の事例を計算機上で表現可能な特徴に変換しなければならない．このとき，表現のために用いられる特徴が学習にとって常に有用であるとは限らず，場合によっては妨げとなることもありうる．しかしながら，事例を適切に表現する特徴を最初から決定することは極めて困難であり，特徴を決定しなければ事例を表現できないという事態に陥ってしまう．このため，事例を表現する特徴の選択には慎重であるべきだが，過度な選択は事例の特徴を失いかねないという矛盾を解消する必要がある．

特徴の抽出は，事例を表現する低レベルの情報から二次的に行うことも可能である．本論文では，事例を表現する原始的な情報をもとに，それらの情報に対して変換や組み合わせ等の操作を施して，学習に有効な特徴へと到達させることが可能であると主張する．また，これらの抽出された特徴を用いて，それぞれの対象領域に固有の学習手法を適用し，これらの学習手法が有効であることを実験によって示す．対象領域として採用したのは，人工離散環境，画像分類，知能ロボットであり，これら異なる 3 領域のいずれにおいても特徴を有効に利用した学習が可能であることを示す．

次に，本論文での構成を述べる．本研究は大きく 3 つの研究から構成される．第 2 章では，本研究に関連する機械学習の主要な学習手法と，それらの概要について述べる．また，機械学習における特徴の重要性を説き，本研究の位置づけについて述べる．

第 3 章では，第 1 の研究として，離散環境における特徴の抽出手法と，それらの特徴を強化学習に導入するための方法論について述べる．強化学習は，教師つき学習とは異なり，学習者自らが試行錯誤を繰り返しながら，次第に行動を洗練していく学習手法であり，本研究では強化学習の中でも，特に注目されている手法の一つである Q 学習の効率化を図る．

Q 学習は，有限離散マルコフ環境下において，充分な試行の後では最適解への収束が保証されている一方，学習完了までに必要とする状態が膨大になるという欠点を持っている．これは，Q 学習が過去の状態を参照する際に完全一致を要求するため，環境の全状態を探索するまで学習が収束しないことが原因である．また，Q 学習が収束するための条件として，状態を識別するのに充分な属性が与えられなければならないという仮定がある．このため，最初に決定する属性を慎重に選ぶ必要がある．逆に，状態を記述する属性を詳細にとりすぎると，状態の一致条件が厳しくなり，さらに収束が遅れるという問題が生じる．このような問題の解決策として，得られた状態から新たな特徴を作り出し，それらの特徴を用いて状態を評価する機能を Q 学習に持たせることが考えられる．構成的帰納学習で提案された特徴構成法は，対象領域に適切な特徴を新たに作り出し，状態の記述を更新する手法である．特徴構成法は，分類などの概念学習において蓄積される状態数の削減と学習精度の向上などの成果が報告されており，Q 学習との統合により上記の機能の実現が可能であると考えられる．

本研究では，特徴構成の機能を持つ強化学習システム FCQL（Feature Constructive Q-Learning）を提案する． FCQL は Q 学習の枠組に特徴構成法を統合したシステムで，報酬を用いて状態を分類し，クラスごとに共通の空間的な特徴を構成して，最適解への収束に必要な状態数の削減と，評価関数の早期収束を目的としている．

第 4 章では，第 2 の研究として，現実的な事例である静止画像を扱い，それらの画像から特徴を抽出して，適切な分類が行える学習手法について述べる．テクスチャは画像の特徴を知る最も有力な手がかりの一つであり，テクスチャ解析の技術は航空写真や医療画像の物体認識などに適用されている．画像認識の分野ではこれまで多くの研究がなされているが，テクスチャを表現する適切な特徴量の決定が難しいため，テクスチャ認識は現在でも困難な問題と考えられている．

テクスチャ認識の難しさとして，異なるスケールのテクスチャを特徴づける適切な手法がなかったことがあげられる．近年，画像解析を行う手段として，ウェーブレット変換に代表される手法がこの問題を解消すると期待されている．ウェーブレット変換は，画像の空間周波数帯域ごとの特徴量を抽出する手法であり，中でも木構造ウェーブレット変換（Tree-Structured Wavelet Transform: TSWT）は特に注目されている．TSWT は，画像を重要な空間周波数帯域に選択的に分割し，帯域分割構造とエネルギーで特徴づける手法である．TSWT は一定の周波数帯域のみを分割しているのではないため，その結果得られる構造は一般に一意に定まらない．このことは，TSWT によって生成された構造はテクスチャごとに固有のものとなり，ウェーブレット特徴量と並んでテクスチャを表現する特徴量となりうることを示している．TSWT で得られた固有の構造はテクスチャ分類に役立つと期待でき，この構造の有効利用が考えられる．

本研究では， TSWT によって得られた構造がテクスチャ分類に重要な特徴量であると主張する．TSWT をテクスチャ分類に適用する際には，ウェーブレット特徴量を適切に抽出するために，構造全体から重要な帯域を選択するボトムアップ的構造化手法を採用する．このとき，構造化されたテクスチャの分布を示すために，新たな指標としてテクスチャエントロピーと呼ばれる尺度を提案する．また，分類精度の向上のため，特徴量としてモーメント特徴量を導入する．

第 5 章では，第 3 の研究として，実環境における多関節型ロボットを対象とした，ロボットの作業のための空間構成法および構成された空間を用いた学習の方法論について述べる．現在，実環境において自律的に行動を獲得できるロボットシステムの開発が求められている．機械学習の分野において研究が進められている強化学習は，環境に対する先見的知識を前提としない，漸次性に優れた学習手法として，このようなシステムへの有効性が注目されている．

従来の強化学習の枠組みは，状態観測の完全性を仮定したマルコフ決定過程（Markov Decision Process: MDP）としてモデル化され，シミュレーション環境下において有効である．しかしながら，実環境では状態観測に不完全性や不確実性が存在し，行動に非決定性やノイズが含まれるため，その有効性が必ずしも保証されないという問題がある．また，実環境におけるロボットの行動獲得では，変動する環境に順応する能力が求められる． MDP 環境を前提とする従来の強化学習では，環境が変動する場合，以前強化されたルールを新しい環境においてそのまま用いることはできない．これらの問題を解決するために，実環境での行動獲得を目的とした状態空間の構成が求められる．

本研究では，すべて対等で曖昧な情報源である光センサを用い，固定点に拘束された多関節型ロボットアームを対象とする．モデル構築手法として，センサからの情報をもとに，概念学習を用いて適切な入力要素を自律的に選択しながら行動モデルを生成し，強化学習によってモデルを拡張する手法を提案する．また，モデルに対する信頼度を設定し，生成されたモデルを切り替える学習手法を提案する．本手法の有効性を示すために，ロボットアームを用いた実環境での実験を行う．実験により，実環境における行動モデルの自律的獲得と，変動する環境への順応という 2 点を検証する．

最後に，第 6 章において，本研究で得られた研究成果をまとめ，今後の課題について述べる．

第 2 章

機械学習における特徴

2.1 緒言

本章では，機械学習における主要な学習手法について概観する．また，機械学習における特徴の重要性について述べる．

2.2 強化学習

2.2.1 強化学習の概要

強化学習（Reinforcement Learning: RL）[1] は，本来動物心理学あるいは動物行動学の分野で用いられた用語である．ある動物個体に特定の行動を起こした時にだけ餌などの報酬を与えるという操作を繰り返すと，その行動パターンが次第に**強化**され，最終的には実際には報酬が与えられなくても，同様の状況下に置かれるとその行動を起こすようになる．このように，罰による行動の抑制も含め，**条件づけ**と言われる一連の適応現象を実現する学習を**強化学習**と呼ぶ．

2.2.2 強化学習の問題領域

強化学習とは，ある種の学習問題のクラスを指す言葉である．学習者としてある環境の中で行動を起こすエージェントを想定する．学習者は各時間ステップにおいて得られる感覚入力から行動を決定する．このとき，実際にとった行動に対して環境から報酬あるいは罰が与えられるが，報酬の大きさは過去数ステップの行動系列に対して決定される．強化学習の基本構造を図 2.1 に示す．

学習の目的は，ある時間の長さにわたる報酬の重み和を最大化することにある．この手法を形式的に記述すれば以下のようになる．時刻 t における報酬の大きさを r_t とすると，

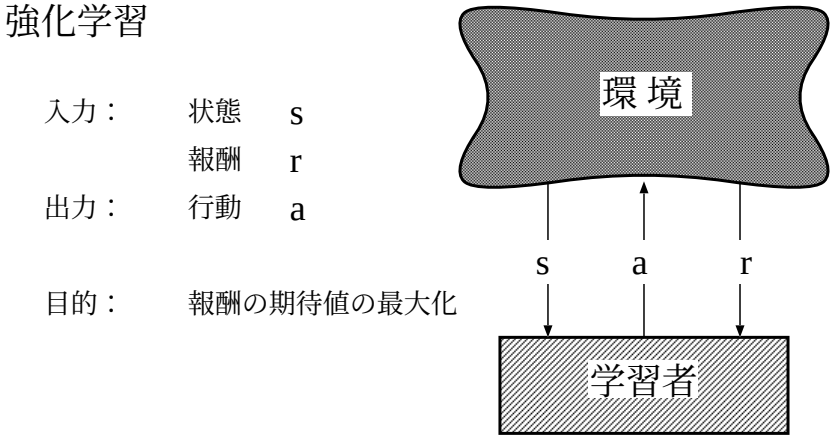


図 2.1: 強化学習の基本構造

学習者の目的は，現在から未来にわたる報酬の重み和

$$v_t = \sum_{i=t}^{\infty} \gamma^{i-t} \cdot r_i \tag{2.1}$$

を最大化することである（γ:割引率，$0 \leq \gamma \leq 1$）．ここで γ は，次の時刻で最適と思われる行動を選択したときに得られる報酬の見積り値を一段階だけ割り引くのに用いる．しかし，現実には未来の報酬は観測できないため，一般には過去から現在までの報酬の重み和

$$\hat{v}_t = \sum_{i=0}^{t} \gamma^{t-i} \cdot r_i \tag{2.2}$$

を v_t の近似として学習性能の評価に用いる．

強化学習における問題の特徴は，

1. システムが出力すべきデータが環境からは与えられず，システムが実際に行った出力に対する評価（報酬）という形で与えられる．
2. システムの出力に対する評価が即座に与えられず，行動の系列に対する評価が遅れて与えられる．

という 2 点にまとめられる．この特徴より，学習者自らの行動の系列が学習に際して最も重要であると言える．

強化学習システムと呼ばれるものは，一般に時刻 t での状態 s_t に対する評価を決定する**学習部**と，状態から次にとるべき行動 a_t を決定する**実行部**に分けることが出来る．実行部では，学習によって得られた状態に対する評価見積りに基づいて行動を決定するが，その時点での評価見積りを最大とするような行動が必ずしも最適であるとは限らない．なぜなら，強化学習では学習者自身の経験が学習者自身の行動に強く依存するからである．経験の内容によって学習結果が異なることは当然だが，強化学習では特に学習者の行動選択が経験の内容を左右する．

2.2.3 Q 学習

Q 学習 [2][3] では，状態と行動の組に対する評価値を見積もる．この評価値を Q 値と呼び，Q 値を導く関数を Q 関数と呼ぶ．時刻 t において状態 s_t にあり，行動 a_t を選択した結果，状態 s_{t+1} へ遷移し，報酬 r_t が得られたとすると，更新すべき Q 値の幅は以下の式で表される．

$$\Delta Q(s_t, a_t) = \alpha(r_t + \gamma \max_b Q(s_{t+1}, b) - Q(s_t, a_t)) \tag{2.3}$$

上式において，α は学習率であり，$0 < \alpha \leq 1$ なる定数である．また，γ は割引率であり，次のステップで最適と思われる行動 b を選択した時に得られる報酬の見積り値 $\max_b Q(x_{t+1}, b)$ を一段階だけ割り引いた値と，時刻 t で直接得られた報酬値 r_t の和に $Q(x_t, a_t)$ を近づける．最も単純な Q 関数の実現方法は，すべての (s, a) 組に対する表を作成することである．有限マルコフ過程では，充分に大きな回数の試行後には，以下のように Q 値を最大にする行動が選択される．この手法を**貪欲な決定戦略（greedy policy）**と呼ぶ．

$$a = arg \max_b Q(s, b) \tag{2.4}$$

Q 学習は動的な環境に安定して強い手法とされている．しかしながら，行動決定を司る Q 関数の実現には最終的に学習目的に直接関連がないデータをも記憶しておく必要があり，充分な試行を行った後にも冗長な部分が多く含まれる．また，あらかじめ実際に有効な特徴を要素の判別することが困難となっている．

2.3 構成的帰納学習

2.3.1 構成的帰納学習の概要

構成的帰納学習（Constructive Induction: CI）[4] は，特徴構成の帰納を有する帰納学習の一手法である．帰納学習は，獲得すべき概念をその具体例から生成する学習である．学習したい概念を求めるには，目的とする概念の例（正例）とその概念に属さない例（負例）が必要である．

このような学習を行う以前に準備する情報として，目的とする概念を表現するための事前知識や知識表現が必要である．これはバイアスと呼ばれている．バイアスの例として，例に含まれる属性が挙げられる．

このように，帰納学習では学習する概念や基準がバイアスに含まれる属性から例に従って選ばれることから，選択的帰納学習（Selective Inductive Learning）と呼ばれることもある．つまり，バイアスにない候補は選べず，学習できないことになる．

帰納学習では，バイアスが学習の範囲・能力を規定している．より柔軟な学習を行うには，広いバイアスを持つか，バイアスを必要に応じて生成する能力が必要とされる．これらの能力を帰納学習に持たせたのが構成的帰納学習である．

2.3.2 構成的帰納学習の問題領域

従来の帰納学習の枠組では学習が困難である問題領域として，選言概念があげられる．選言概念とは，概念記述が選言形式で表現される概念を指している．特徴とは，ある属性が特定の値をとることである．選言概念の中で，特徴間に何らかの従属関係がある概念を特に選言標準形（Disjunctive Normal Form: DNF）概念 [5] と呼ぶ．

DNF 概念の例として，三目並べ問題が挙げられる．この問題は「三目並べゲーム終了時点における×の勝ち」という概念を学習するものである．各事例はゲーム終了時の盤面の状態を表しており，与えられた属性のうち×が一列に三連を成しているものを正例，それ以外を負例とする．各属性は盤面上の特定の一区画に対応しており，それぞれの属性は “×”，“○”，“(空白)” のいずれかの値を持つものとする．

三目並べ問題における学習の困難さは，各特徴が正例中にも負例中にも一様に存在していることにある．つまり，特徴 A_1 =“×” に着目して事例集合を分割してみても，分割されたどちらの集合にも正例と負例が同程度に含まれていることになるので，正例と負例の分割は困難である．以上のような時，(A_1 =“×” $\cap$ A_2 =“×” $\cap$ A_3 =“×”) のような特徴の連言を用いることによって，初めて正例と負例を部分的に分割することが出来る．構成的帰納学習では，特徴の連言を新たなバイアスとして採用し，学習対象概念を再描写している．

以上のように，DNF 概念は従来の帰納学習アルゴリズムでは学習が困難であるが，構成的帰納学習を用いることによって学習が容易となる概念の典型的な例であると言える．

2.3.3 構成的帰納学習の基本概念

構成的帰納学習は，学習以前または学習中にバイアスを変更・生成する．つまり，構成的帰納学習には選択的帰納学習の機能に加えて,「バイアスの生成機能」と「生成されたバイアスの選択機能」が必要となる．バイアス生成機能により生成された新たな概念属性がバイアス選択機能により適切に選択され，選択的帰納学習システムで利用するバイアスを変更する．バイアスを特徴と見なすと，本研究における構成的帰納学習の概要は，図 2.2 のように表わせる．

2.4 事例に基づく学習と分類

2.4.1 事例に基づく学習

事例に基づく学習（Case-Based Learning: CBL）[6] は，事例集合から一般的な概念記述を帰納することなく，事例そのものによって概念を表現する学習手法である．事例を分類する際には，事例集合内から最も類似している事例を検索し，この類似事例が属するクラスを分類対象とする．

最も単純な事例に基づく学習では，すべての事例を事例集合に記憶するため，事例の記憶量と類似事例の検索にかかる計算コストが問題となっている．このため，冗長な事例を

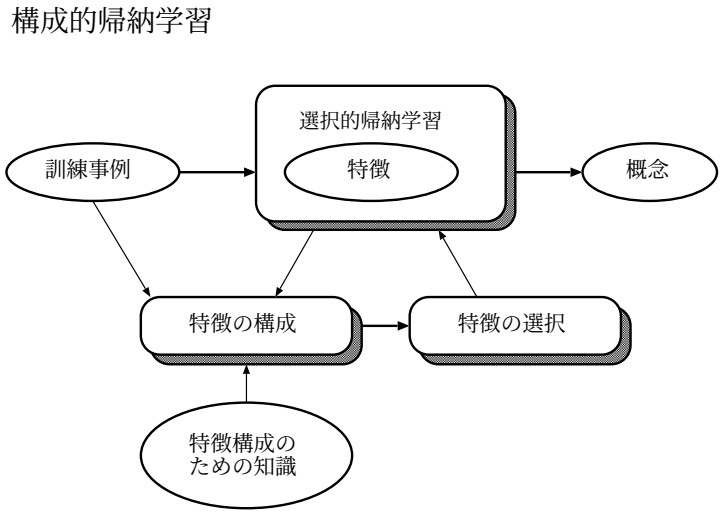


図 2.2: 本研究における構成的帰納学習の概要

排除し，記憶すべき事例を減らすことは，重要かつ有益な技術であると考えられている．

2.4.2 距離関数を用いた分類

事例間の類似度は，ユークリッド距離などの距離関数を用いて定義されることが多い．この距離関数を用いて事例集合と分類したい事例の距離を計算し，距離が最小のクラスに分類する．一般に，特徴の数 n が同じ 2 つの事例 $x = (x_1, x_2, ..., x_n)$, $y = (y_1, y_2, ..., y_n)$ 間のユークリッド距離 $D(x, y)$ は，以下の式によって定義される．

$$D(x, y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \tag{2.5}$$

事例集合との距離は，各事例集合ごとの事例との間で計算する．しかしながら，前述のようにすべての事例を記憶しておくのでは非効率なため，各事例集合ごとに代表的な事例を残しておいたり，平均的な事例を計算したりする手法がとられている．

2.5 特徴の重要性

前節までに述べたいずれの学習手法においても，事例を表現するための特徴が重要であることが明らかになった．しかしながら，事例を適切に表現する特徴を事前に決定することは容易ではない．このため，すでに何らかの特徴を用いて記述された事例を，それらの特徴から得られた二次的な情報を用いて表現し，それを新たな特徴として事例を再描写することが重要である．これらの新たな特徴が事例を適切に表現することに成功していれば，精度の高い学習や分類を行うことができる．

本研究では，事例を表現する特徴を，学習対象ごとに適切に抽出する手法を適用し，それらの特徴を用いて学習・分類を行い，より知的な振舞いをする学習機構を検証し，実験によって評価する．

2.6 結言

本章では，機械学習における主要な学習手法について概要を解説した．また，機械学習における特徴の重要性について述べ，本研究の位置づけを明らかにした．

第 3 章

離散環境における特徴抽出

3.1 緒言

本章では，まず記述の容易な対象領域として人工的な離散環境を取扱い，この領域を表現するのに適している特徴の抽出を試みる．特徴の抽出手法として特徴構成法を採用し，抽出された特徴に基づいて事例を評価して，環境に対する学習を行う．学習手法としては強化学習を採用し，領域内での認識と行動の繰り返しによるシミュレーションを行って，学習状況を評価する．

3.2 研究背景

近年，実環境における移動ロボットのように，未知なる環境下での学習機能を持ったシステムに関心が高まりつつある．このような環境下では，学習の対象やタスク，あるいは環境との相互作用によって環境に変化が生じ，それらの変化に対するシステムの即応性が求められる．このようなシステムの設計において，動物の学習形態を模倣した**強化学習** [1] が注目されている．強化学習は，教師つき学習とは異なり，学習者自らが試行錯誤を繰り返しながら，次第に行動を洗練していく学習手法である．強化学習の中でも，特に注目されている手法の一つに Q **学習** [2][3] があげられる．

Q 学習は，有限離散マルコフ環境下において，充分な試行の後では最適解への収束が保証されている一方，学習完了までに必要とする状態が膨大になるという欠点を持っている．これは，Q 学習が過去の状態を参照する際に完全一致を要求するため，環境の全状態を探索するまで学習が収束しないことが原因である．また，Q 学習が収束するための条件として，状態を識別するのに充分な属性が与えられなければならないという仮定がある．このため，最初に決定する属性を慎重に選ぶ必要がある．逆に，状態を記述する属性を詳細にとりすぎると，状態の一致条件が厳しくなり，さらに収束が遅れるという問題が生じる．このような問題の解決策として，得られた状態から新たな特徴を作り出し，それらの特徴を用いて状態を評価する機能を Q 学習に持たせることが考えられる．

得られた状態を分割し，状態空間を構成する手法に，以下のような研究がある．G アル

ゴリズム [7] では，状態を記述する属性をノードとして，木構造による Q 関数の実現で状態空間を分割しているが，与えられた属性をそのまま用いて分割を行うため，この手法ではDNF 問題を解くことができない．浅田 [8] らによる状態空間の自律的構成法では，ゴールまでの行動数に基づいて状態の集合を構成しているが，時間的な抽象化を行っており，状態に基づく空間的な分類は行われていない．前者は与えられた属性による状態空間の分割，後者は得られた状態を時間別に分類して状態空間の再構成を試みる手法であるが，いずれの手法も得られた状態から対象領域の特徴を構成するものではない．また，EOPs[9] では，クラスを識別する関数を定義して分類を行っているが，この手法もまた得られた状態を分類するにとどまり，同じクラスに属する状態から空間的な特徴を構成するものではない．

構成的帰納学習 [4][5][10] で提案された**特徴構成法**は，対象領域に適切な特徴を新たに作り出し，状態の記述を更新する手法である．特徴構成法は，分類などの概念学習において蓄積される状態数の削減と学習精度の向上などの成果が報告されている．本章では，特徴構成の機能を持つ強化学習システム FCQL （Feature Constructive Q-Learning）を提案する [11][12][13]． FCQL は Q 学習の枠組に特徴構成法を統合したシステムで，報酬を用いて状態を分類し，各クラスごとに共通の空間的な特徴を構成して，最適解への収束に必要な状態数の削減と，評価関数の早期収束を達成している．

3.3 特徴構成法

3.3.1 構成的帰納学習における特徴構成法

対象とする領域の状態集合を分類し，同じクラスに属する状態に共通の概念を獲得する手法に，帰納学習がある．一般的な帰納学習は，概念記述があらかじめ与えられた属性から選ばれるために，選択的帰納学習とも呼ばれる．選択的帰納学習で学習できない問題の例に DNF 問題がある． DNF 問題とは，学習したい概念が DNF （選言標準形： Disjunctive Normal Form）で記述される問題領域である．目標概念となる DNF は，属性の連言の選言形式で記述される．

DNF 問題の代表例に三目並べ問題（図 3.1）がある．この問題は,「三目並べの終了時における X の勝ち」という概念を学習する問題である．以下では，学習したい概念に含まれる例を正例，それ以外を負例と呼ぶ．

ここで，正例を記述する概念が $(x_1{=}\mathrm{X})$ であると仮定して，選択的帰納学習を用いて正例と負例への分割を考える．このような記述は正例中にも負例中にも一様に存在しているため，学習は失敗する．これは，三目並べ問題で学習したい概念が $(x_1{=}\mathrm{X}) \wedge (x_2{=}\mathrm{X}) \wedge (x_3{=}\mathrm{X}) \vee (x_4{=}\mathrm{X}) \wedge (x_5{=}\mathrm{X}) \wedge (x_6{=}\mathrm{X}) \vee \cdots$ のような DNF で記述され，単項の属性のみでは分類できないことが原因である．このように，DNF 問題のような複雑な問題をを学習するには多くの例を保持しておかなければならず，例からの学習を行う Q 学習において非効率的な問題の代表例であると言える [14].

構成的帰納学習では，得られた状態から新たに特徴を構成し，その特徴が一定の基準を満たせば選択される．つまり，構成的帰納学習には，選択的帰納学習の機能に加えて,「特

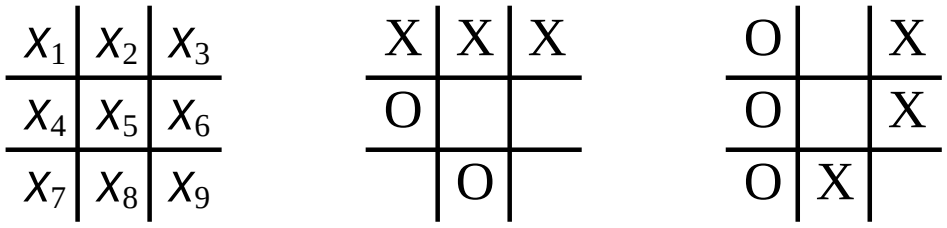


図 3.1: 三目並べ問題

徴の構成」と「特徴の選択」の機能が追加されている．一般に，構成的帰納学習における特徴とは，対象領域の概念を表現するための要素を指す．特徴の構成は，学習した概念に矛盾が発生した場合 [1]，選択できる概念の候補がなくなった場合などに行われる．特徴を構成するためには操作子が必要となるが，与えられた属性から特徴を構成する操作子の例として，数値属性に対しては比較，統合，細分化など，非数値属性に対しては分割，統合，論理演算などがあげられ，三目並べの例では論理演算子の一つである論理積を用いている．

特徴構成法において議論される点は，大きく分けて二つある．まず第一に，特徴構成の手続きが組み込み型か前処理型かという点である．組み込み型は，学習アルゴリズム本体に特徴構成法が組み込まれており，学習中に特徴構成を逐次行うことができるが，各アルゴリズムごとに特化した実装をする必要がある．一方，前処理型は学習を行う前に特徴構成を行うため，他の学習アルゴリズムのフィルタとして用いることができるが，学習中には特徴を構成できないことが欠点である．

第二に，特徴構成が状態に基づくか知識に基づくかという点である．状態に基づく手法は，特徴構成の際にあらかじめ領域依存な知識を必要としないため，汎用性の高い手法であるが，構成された特徴の精度は低い．一方，知識に基づく手法は，領域に固有の特徴が適切な形で構成されやすいが，知識が必要になるため，汎用性に欠ける．

代表的な構成的帰納学習システムに IB3-CI[5] ，GALA[15] などがある．IB3-CI では，帰納学習システム IB3 に特徴構成法 STAGGER を組み込み，分類精度の向上を達成している．STAGGER は組み込み型であるが，評価は静的な状態集合を対象としているので，逐次的に得られる状態に基づいて構成された特徴には，冗長なものが多く含まれる．また，特徴構成のための知識を与えることも可能であるが，IB3-CI は知識に基づく特徴構成のみでしかよい学習結果は得られていない．

一方，GALA は状態に基づく前処理型の特徴構成法で，多くの帰納学習アルゴリズムのフィルタとして適用できる．GALA は特徴選択の評価関数として**利得比基準** [16] を用いており，精度の高い選択ができる．しかしながら，前処理型であるため，対象領域の状態がすべて既知である必要があり，学習中には特徴を更新できないことが問題となっている．

Q 学習では逐次的に状態に遭遇するため，特徴は学習中に適宜更新されることが望まし

[1] 正例ばかりで過剰一般化となる場合など．

い．このため，FCQL では特徴構成法をアルゴリズム中に組み込んでいる．さらに，特徴に基づいた行動決定と，特徴の評価値の更新のために，FCQL では各特徴と行動の対に評価値を与える関数を新たに定義している．本章では，Q 学習では非効率な問題の例として DNF 問題を取り上げる．また，特徴を構成する操作子に論理積を採用し，以下では属性—値対の連言のことを**特徴**と呼ぶ．

3.3.2 FCQL における特徴構成法

特徴の構成

帰納学習では，与えられる状態は複数の属性—値対と，その状態の属するクラスで記述されている．FCQL では，このクラスに相当する部分を，状態が得られた直後の報酬を用いて定義している．具体的には，時刻 t で得られた状態を s_t ，報酬を r_t とすると，s_t は複数の属性—値対で記述されている部分で， r_t がクラスに相当する部分となる．各クラスごとの報酬値が既知であるとすると，s_t は 報酬値が r_t に最も近いクラスに分類される．

また，FCQL で扱う特徴は，属性—値対の連言で記述され， 2 値をとる．たとえば，第 3.3.1 項の三目並べの例では，状態 s_t に対し，特徴 $f_1 = (x_1{=}\mathrm{X}) \wedge (x_2{=}\mathrm{X}) = \mathrm{True}$ となることは，s_t に $(x_1{=}\mathrm{X})$ と $(x_2{=}\mathrm{X})$ という属性—値対が含まれていることを意味している．

FCQL の特徴構成では，まず，得られた状態の属性—値対について，すべての 2 項連言が構成される．このとき，すでに構成された特徴が存在すれば，それらの特徴と属性との連言についても，可能な組合せがすべて構成される．三目並べの例では，すでに特徴 f_1 が存在しているとき，特徴 f_2

$$\begin{aligned} f_2 &= (f_1 = \mathrm{True}) \wedge (x_3 = \mathrm{X}) \\ &= (x_1 = \mathrm{X}) \wedge (x_2 = \mathrm{X}) \wedge (x_3 = \mathrm{X}) \end{aligned}$$

という組み合せも可能となり，属性の 3 項以上の連言も特徴の候補として構成される．状態を記述する属性数を n とすると，特徴を構成する項数は最大で n となる．また，一つの状態から構成される特徴数は，最大で $\sum_{i=1}^{n} {}_n\mathrm{C}_i$ となり，全特徴数はこの状態数倍が上限となる．

特徴の選択

構成された特徴の候補は，何の評価も受けていないため，その中から適切な特徴を選択する必要がある．たとえば，第 3.3.2 項の三目並べでの特徴 f_1 を用いて，特徴 f_3

$$\begin{aligned} f_3 &= (f_1 = \mathrm{True}) \wedge (x_4 = \mathrm{O}) \\ &= (x_1 = \mathrm{X}) \wedge (x_2 = \mathrm{X}) \wedge (x_4 = \mathrm{O}) \end{aligned}$$

といった特徴も構成できるが，この特徴は三目並べにおける正例の条件[2] を満たしていないので，学習には不適切な特徴である．このような候補すべてを採用すると特徴が増えす

[2] X が一列に並んでいる状態.

ぎてしまうため，その中から適切な特徴を選択しなければならない．FCQL では，特徴の採用に評価指標を設け，特徴の候補の中から最も適切な特徴を選択して解決している．特徴選択のための評価指標として，FCQL では GALA に導入された利得比基準を採用する．

利得比基準は，ある状態集合を分割するときに，**利得比**（$Gain\ Ratio$）が最大の属性を選択する基準である．たとえば，状態集合 T を部分集合 $T_1, T_2, ..., T_n$ に分割する n 値の属性 X と，いくつかのクラス C_i を考える． T 中の状態数を $|T|$ ， T 中で C_i に属する状態数を $freq(C_i, T)$ と表すと， k 個のクラスに関する**平均情報量**（$Info$）は， T 内のある状態が属するクラスを同定するのに必要な情報量の平均値を表し，

$$Info(T) = -\sum_{i=1}^{k} \frac{freq(C_i, T)}{|T|} \times \log_2(\frac{freq(C_i, T)}{|T|}) \tag{3.1}$$

となる．X によって T が n 通りに分割された後，クラスを同定するのに必要な**情報量の期待値**（$Info_X$）は，

$$Info_X(T) = \sum_{i=1}^{n} \frac{|T_i|}{|T|} \times Info(T_i) \tag{3.2}$$

となる．これらの差である**利得**（$Gain$）

$$Gain(X) = Info(T) - Info_X(T) \tag{3.3}$$

は， X で T を分割したとき，クラス分けに役立つ部分の情報量を表している．一方，**分割情報量**（$Split\ Info$）は，X による分割自体の情報量を表し，

$$Split\ Info(X) = -\sum_{i=1}^{n} \frac{|T_i|}{|T|} \times \log_2(\frac{|T_i|}{|T|}) \tag{3.4}$$

となる．したがって，利得比

$$Gain\ Ratio(X) = \frac{Gain(X)}{Split\ Info(X)} \tag{3.5}$$

は，分割によって得られる情報量のうち，クラス分類に役立つ部分の割合を表している．

利得比基準は，分類すべきクラス数が状態数と比べて充分に小さいときにも，分類に役立つ特徴を分割された部分集合の情報量に基づいて定量的な評価ができるという利点がある．また，離散属性では属性ごとにとり得る属性値の数に隔たりが生じやすいが，この場合でも利得比基準は特定の属性を不利に扱うことがない．さらに，分割後の部分集合の大きさが不均等な場合にも安定した結果が得られるという性質がある．したがって， FCQL が対象とする問題領域の状態は離散属性で記述されており，特徴のとる値が 2 値であるために不均等な分割になりやすいが，利得比基準はこのような問題領域に頑健であるため，FCQL への導入に適していると考えられる．

FCQL では，構成された特徴の候補の中で利得比が最大の特徴を 1 つ選択し，新しい特徴としている．これは，利得比が大きな特徴は，その特徴を用いて状態集合を分割したときに，同じクラスの状態が集まりやすいことを意味している．利得比が同じ特徴が複数個

存在する場合は，連言を構成する項数が少ない特徴を選択している．最後に，この特徴がすでに採用された特徴と重複しなければ，新たな特徴として採用する．採用された特徴は特徴を記憶しておく領域に追加され，この領域を**特徴集合**と呼ぶ．以上の手続きで採用された，特徴集合中にある j 番目の特徴 f_j に対し，状態 s_t において $f_j = \mathrm{True}$ となるとき，特徴 f_j は**状態 s_t に含まれる** といい，

$$f_j \in s_t \tag{3.6}$$

と表記される．このときに採用された特徴を f_j ，特徴構成が行われる直前にとった行動を a_t ，得られた報酬を r_t とすると，f_j と a_i の対に対して評価値を与える関数 F を **特徴関数**と呼び， $F(f_j, a_i)$ の初期値を

$$F(f_j, a_i) = \begin{cases} r_t & (\ i = t\ ) \\ 0 & (\ i \neq t\ ) \end{cases} \tag{3.7}$$

と定義する．特徴関数は，特徴の評価値の更新，および特徴の淘汰に用いられる．

特徴関数の更新

第 3.3.2 項の手法で得られた特徴を追加していくのみでは，特徴が増えすぎてしまう恐れがある．また，構成の時点ではよいと判断された特徴も，学習が進むにつれ不適切になっていく可能性もある．選択された特徴が不適切なとき，それらの特徴に基づいた行動決定は最適であるとは言えないばかりか，不適切な行動を招く恐れもある．この現象を解消するために，選択された特徴の評価値を更新し，不要な特徴を淘汰する手続きが必要となる．FCQL では，この手続きを行う評価基準に特徴関数を用いている．たとえば，ある特徴 f_j の特徴関数の値 $F(f_j, a_i)$ は，以下の式に基づいて更新される．

$$F(f_j, a_i) \leftarrow F(f_j, a_i) + \alpha(r_t + \gamma \max_{g \in s_{t+1}, b \in A} F(g, b) - F(f_j, a_i)) \tag{3.8}$$

α は学習率で，$0 < \alpha \leq 1$ なる定数である．α の値が大きいほど，直前の報酬を重要視するように関数が更新され，$\alpha = 1$ のときには過去の関数値を一切利用しない．γ は割引率で，$0 \leq \gamma \leq 1$ なる定数である．$\gamma = 0$ の場合は，現在の報酬のみに着目し，将来得られる報酬を無視することになる．逆に，γ の値が大きいほど将来の報酬を重視し，$\gamma = 1$ の場合は時刻 t と $t+1$ で得られる報酬を等価に扱うことになる．つまり，式（3.8）は，次の時刻で最適と思われる行動を選択したときに得られる報酬の見積もりを一段階だけ割り引いた値と，直前に得られた報酬の和に，学習率に従って $F(f_j, a_i)$ の値を近づけることを示している．

構成されたすべての特徴は， 0 でない報酬値が得られた状態から構成されているので，試行が進むにしたがって F 値は実際に得られる報酬値に近付いていく．誤って構成された特徴は，以降の試行で報酬を得られることが少なく， F 値は 0 に近づいていく．この更新手続きの結果，すべての行動について評価値のクラスが 0 になった特徴は特徴集合から削除される．

強化学習から特徴構成へのフィードバック

第 3.3.2 項の操作で更新された特徴関数を用いて，特徴構成の際に採択されなかった特徴の評価値の更新に反映させることを考える．第3.3.2 項で構成された特徴のうち，すでに特徴集合中に存在するために棄却された特徴については，式（3.8）に従ってその評価値が更新される．以上の操作を強化学習側から特徴構成へのフィードバックとし，強化された特徴関数を更新に用いた結果，既存の特徴の評価値をより早く正確な値に近づけることができる．

状態集合の大きさに関する考察

利得比基準による特徴の評価は，対象とする状態数の大きさに依存する．充分な大きさの状態集合が得られたとき，利得比基準によって対象領域を表現するのに充分な数の特徴を得ることができる．しかしながら，学習初期においては，経験した状態数が少なく，Q 表の規模も小さい．利得比基準による評価は小さな状態集合に対しては安定せず [16]，必ずしも正当な評価がなされるとは限らない．不適切に選択された特徴が学習初期において不適切な行動を招き，適切な状態の蓄積による Q 関数拡張の妨げになっていると考えられる．この現象は，充分な量の状態を蓄積しないうちに特徴を構成し始めた点に問題があり，蓄積された状態の量に基づいて特徴の構成を始めるタイミングを考慮する必要がある．

以上のような問題点の解決策として，特徴を構成するのに充分な量の状態を得てから特徴構成を始めることが考えられる．しかしながら,「充分な」量を定量的に示す指標を作ることは困難である．また，充分な量の状態を得るまで待つと，試行全体としての収束速度に影響が出る可能性もある．

FCQL では，最初に特徴構成をしない通常の Q 学習を行い，一定数の状態を蓄積し，その後一度だけ特徴構成を行う手続きに切り替える．このとき，学習アルゴリズムを Q 学習から FCQL に切替えるまでの状態を蓄積する過程を**ディレイ**と呼ぶ．

3.4 特徴構成法を用いた Q 学習

3.4.1 対象とする学習領域

FCQL が対象とする学習領域は，外界となる環境が有限離散マルコフ決定過程によってモデル化される．時刻 t におけるシステムの入出力は，以下のように定式化される．

- 与えられる入力：
  - 状態 s_t
  - 報酬 r_t
- 決定すべき結果：
  - 行動 a_t

状態 s_t は離散属性で記述される．報酬 r_t はあらかじめ定義されたクラスのうち，いずれかのクラスに属するものとする．また，行動 a_t は，あらかじめ定義された行動集合 A のうち， s_t において実行可能な行動の集合から選択される．選択された a_t を実行し，このとき生じた環境の変化に応じて報酬 r_t が環境から得られる．学習の目的は，長期にわたる割引報酬和の最大化にあり，単位行動あたりの報酬値で評価される．

また，FCQL が対象とする問題領域は，行動に伴う報酬が即時に得られることを仮定している．このため，即時的な報酬に無関係な状態は学習の対象とされず，即時に報酬が得られる行動からの学習が行われることとなる．このような行動の獲得は**反射的行動獲得**と呼ばれ，FCQL では反射的行動獲得を行う．

3.4.2 FCQL アルゴリズム

Q 学習では，状態と行動の組に対して評価値を設定し，この評価値を手がかりに学習が進行する．この評価値を Q 値と呼び，Q 値を導く関数を Q 関数と呼ぶ．最も単純な Q 関数の実現方法は，すべての状態と行動の組について表を作り，内容として各組の Q 値を記録しておき， Q 値を直接更新しながら学習を進めていく手法である．この実現方法は**Table Lookup 法**と呼ばれる．

FCQL では，現在の状態の入力，行動選択，行動の実行と報酬の獲得，学習と特徴構成による内部状態の更新までの一連の手続きを一周期とする．また，Q 関数の実現には，離散的な環境での実装が容易な Table Lookup 法を用いている．図 3.2 に FCQL アルゴリズムを示す．

行動選択

行動選択手続き **Policy** では，まず s_t に含まれる特徴を判定する．特徴集合中にある j 番目の特徴について，$f_j = \mathrm{True}$ となる特徴を検出し，この操作を特徴集合中のすべての特徴について行う．

s_t に含まれる特徴が存在する場合は，過去に経験した状態と同様の特徴をもつ状態に遭遇したと判断され，各行動の評価値によって Boltzmann 分布に基づいた確率選択が行われる．s_t に含まれる特徴が複数存在するときは，各特徴における報酬の和によって， Boltzmann 分布に基づいた確率選択が行われる．すなわち，状態 s_t に含まれる特徴 f_j に基づく行動 a_i を選択する確率 $p(a_i|f_j \cap f_j|s_t)$ は，

$$p(a_i|f_j \cap f_j|s_t) = \frac{e^{F(f_j,a_i)/T}}{\sum_{f_l \in s_t} \sum_{a_k \in A} e^{F(f_l,a_k)/T}} \tag{3.9}$$

となる．ただし， T は温度定数で，値が大きいほど行動はよりランダムになり，それぞれの行動を選択する確率の差が小さくなる．逆に， T を 0 に近づけると，わずかな Q 値の差が行動選択に大きく影響し，極限では Q 値を最大にする行動が選ばれる．つまり，

$$a_t = \arg\max_b Q(s_t, b) \tag{3.10}$$

MAIN LOOP:

1. 現在の状態 s_t を入力する.
2. 行動 a_t を選択する.
   $a_t \leftarrow \mathbf{Policy}(s_t)$.
3. a_t を実行し，次の状態 s_{t+1} に遷移し，報酬 r_t を獲得する.
4. 内部状態を更新する.
   $\mathbf{Learn}(s_t, a_t, r_t, s_{t+1})$.
5. 1. に戻る.

Policy(s_t):
以下に示す優先順位で，いずれか一つの処理が選択される.

1. s_t に含まれる特徴が存在すれば，特徴に基づく行動選択を行う.
2. s_t と同一の状態が Q 表に存在すれば，状態に基づく行動選択を行う.
3. ランダムに行動を選択する.

Learn(s_t, a_t, r_t, s_{t+1}):
以下に示す優先順位で，いずれか一つの処理が選択される.

1. 特徴に基づく行動選択を行った場合，特徴関数を更新する.
2. 状態に基づく行動選択を行った場合，Q 関数を更新する.
3. $r_t \neq 0$ の場合，新たに特徴を構成する.
4. 内部状態の更新を行わない.

図 3.2: FCQL アルゴリズム

となる．学習当初より報酬が最大の行動を選択すると，環境の探索が不十分となるため，確率的な行動選択を採用する [17]． Boltzmann 分布による行動選択は，各行動ごとの報酬を重みとした確率選択であり，正の報酬が得られる行動を重視し，負の報酬が得られる行動を小さな正の値で評価している．報酬の値を選択確率に直接用いるのではなく，すべて正の値に変換するために，式（3.9）の分母が 0 になることを回避できる．

以上のような行動選択方式を**特徴に基づく行動選択**と呼ぶ．特徴に基づく行動選択は，以下に示す**状態に基づく行動選択**よりも優先的に処理される．

s_t に含まれる特徴が存在しない場合は，学習者は s_t を Q 表と照合し，最適と判断される行動 a_t を決定する．このとき， s_t と一致する状態が Q 表内に存在すれば，Boltzmann 分布に基づく確率選択を行い，行動選択手続きを終了する．すなわち，状態 s_t で行動 a_i を選択する確率 $p(a_i|s_t)$ は，

$$p(a_i|s_t) = \frac{e^{Q(s_t,a_i)/T}}{\sum_{a_k \in A} e^{Q(s_t,a_k)/T}} \tag{3.11}$$

となる．この行動選択方式を状態に基づく行動選択と呼ぶ．

上記のいずれにも該当しない場合，s_t において可能な行動の中からランダムに行動が選択される．

学習と特徴構成

学習手続き **Learn** では，行動 a_t の実行により遷移した状態 s_{t+1} と，得られた報酬 r_t を用いて，システムの内部状態を更新する．このとき，行動選択手続きでとった戦略により，以下のように処理が分かれる．

行動選択手続きで特徴に基づく行動選択が行われた場合は，同じ特徴をもつ状態を過去に経験したと判断できるので，式（3.8）に基づいて特徴関数を更新し，学習手続きを終了する．

状態に基づく行動選択が行われた場合は，すでに経験済の状態であると判断できるので，以下の式に基づいて Q 関数を更新し，学習手続きを終了する（α は学習率， γ は割引率）．

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha(r_t + \gamma \max_{b \in A} Q(s_{t+1}, b) - Q(s_t, a_t)) \tag{3.12}$$

上記のいずれにも該当しない場合は，r_t によってさらに処理が分かれる．$r_t \neq 0$ のときは，報酬が得られる特徴が存在すると判断できるため，第 3.3.2 項で述べた手法で特徴の構成と選択を行い，特徴集合に追加する．また，s_t を新たに Q 表に追加し，それぞれの行動に対する Q 値の初期値を

$$Q(s_t, a_i) = \begin{cases} r_t & (\ i = t\ ) \\ 0 & (\ i \neq t\ ) \end{cases} \tag{3.13}$$

と定義する． $r_t = 0$ のときには，Q 関数の更新，特徴関数の更新，および特徴構成のいずれの処理も行われない．

3.5 FCQL による学習例

3.5.1 問題設定

本節では，人工的な迷路問題 [18] に対し FCQL を用いてシミュレーションを行い，その結果について述べる．対象となる問題領域を採用した理由として，

1. 有限離散マルコフ環境である．
2. 構成したい特徴が DNF で記述される．

といった点があげられる．なお，対象とする問題領域は，図 3.3 に示すような 2 次元の迷路を想定する．迷路は広さが 7 × 7 ブロック の格子状の環境で，この環境内には 5 種類の物体が存在する．

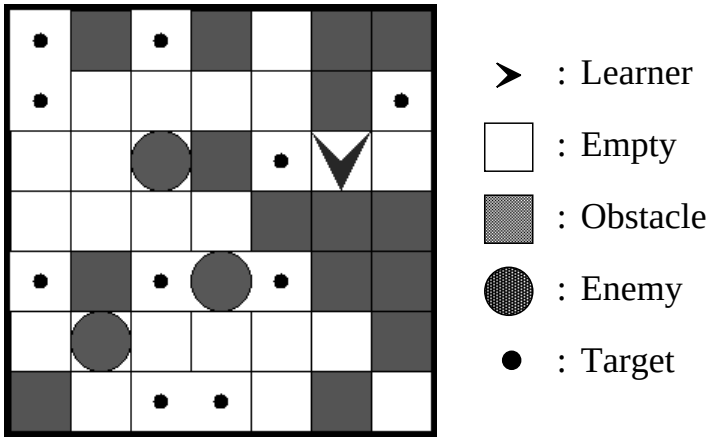


図 3.3: シミュレーション環境

図 3.3 で矢頭型に描かれている物体が学習者の位置を表している．小さな点で示されているのが餌で，学習者が餌を捕らえると +0.5 ポイントの報酬を得る．大きな円で描かれているのが敵で，出現地で静止しており，学習者が重なると −1.0 ポイントの報酬を得て，その試行は終了する．矩形状に黒く塗りつぶされている物体は障害物で，障害物の向うにさらに障害物がある場合は，学習者は障害物のある方向へ進むことができない．それ以外の場合は，障害物の向うにある物体を押し潰しながら障害物とともに 1 ブロック進む．このとき，敵を押し潰すと +1.0 ポイントの報酬を得る．また，白い矩形で表される領域は空白になっており，学習者はこの領域に進むことができる．学習者が入力として得られる属性は，学習者自身の位置，および学習者の周囲 4 方向 × 距離 2 ブロック，合計で 9 属性である．各試行開始時における学習者とすべての敵，障害物，および餌の出現地点は，それぞれ順に空白の中からランダムに選ばれるものとする．また，各物体の数をそれぞれ敵が 3 ，壁が 15 ，餌が 10 と定める．

学習者が行う状態の入力，行動選択，行動の実行に伴う報酬の獲得，および内部状態の更新までの一連の手続きを**1 ステップ**と定義する．また，学習者が敵に重なるか，あるいは 100 ステップが経過するまでの一連のステップ群を**1 試行**とする．シミュレーションは最大 100 ステップの試行を 2000 回繰り返し，以上の作業 5 回の平均を取り，単位試行あたりの報酬値，および蓄積されている状態数を評価する．この問題領域では反射的行動獲得を行うため，常に最新の関数値を重視するように α と γ の値を高く，T の値を中央よりやや低くして，アルゴリズム中の各パラメータを $\alpha = 0.9$ ，$\gamma = 0.9$ ， $T = 0.4$ と設定する [17][19].

この問題領域が持つ特徴は，学習者の前後左右いずれかの 1 ブロックの距離に餌が存在するという単項の属性で記述される特徴が 4 種類，同様に 1 ブロック先に敵が存在する特徴が 4 種類，学習者の前後左右いずれかが壁で，その 1 ブロック先に敵が存在するという 2 項連言の特徴が 4 種類，計 12 種類である．図 3.4 に，この問題領域における特徴の正解を示す.

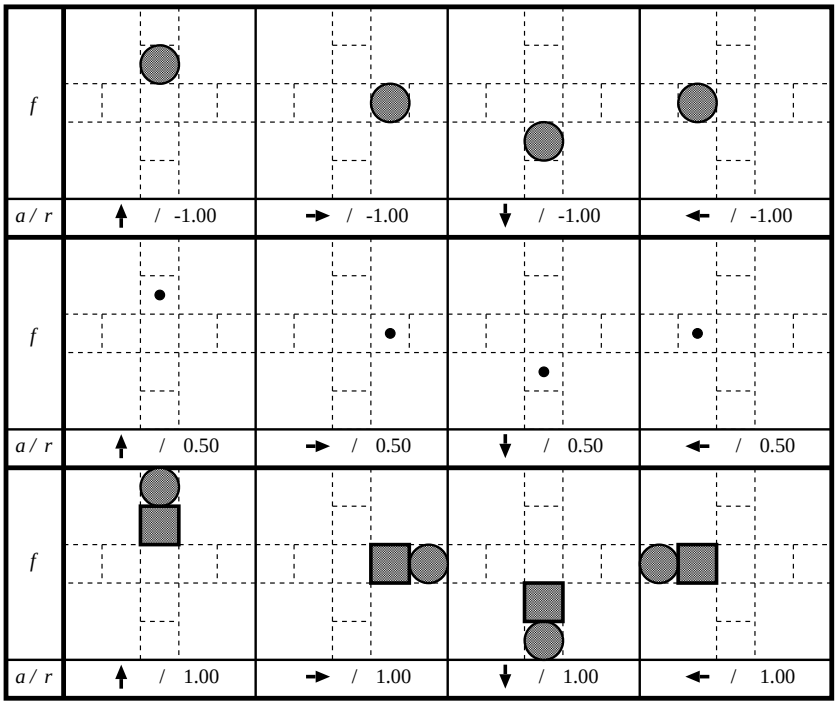


図 3.4: 問題領域に固有の特徴

上段の f で示す図が対象領域の特徴で，学習者の周囲 4 方向の空間にある物体を示している．下段の a/r で示している部分が，それぞれ 特徴 f に基づく行動とその評価値で，行動は矢印の方向へ進むことを表している．得られる 3 種類の報酬ごとに一つのクラスとすると，これらの特徴は各クラスごとの DNF で記述できる.

3.5.2 シミュレーション結果

本項では FCQL を用いたシミュレーション結果を示す．比較対象として，構成したい特徴が最初から与えた FCQL ，および従来の Table Lookup 型 Q 学習を取り上げ，理論的な収束値との比較，および改良による学習効率の変化を評価する．本シミュレーションでは，構成したい特徴を最初から与えた FCQL が収束する値を理論値と見なしている．評価対象は，学習中に蓄積された状態数，及び単位試行あたりの平均報酬値の推移とする．本シミュレーションでは特徴構成を始める状態数を 400 個と定める．なお，ディレイをかける状態数については，第 3.7 節の問題点で再び検討する．図 3.5，図3.6 は，それぞれ学習中に蓄積された状態数の推移，および平均報酬値の推移を示している．

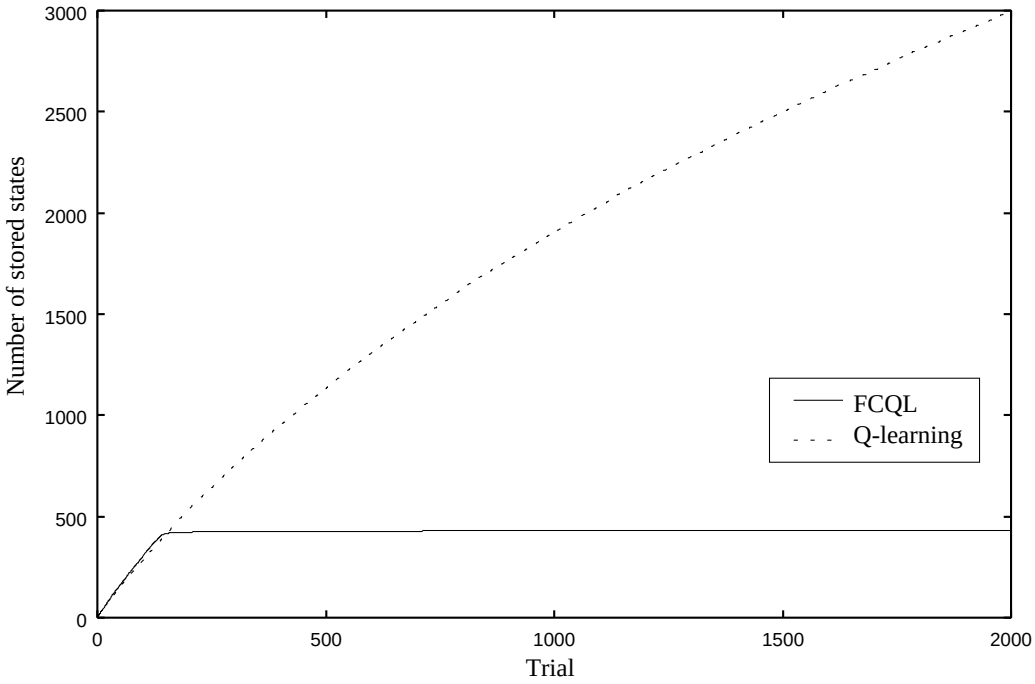


図 3.5: 蓄積状態数の推移

図 3.5より，状態数 400 のディレイをかけたにもかかわらず，蓄積状態数の大幅な増加はなく，従来の Q 学習よりかなり少ない状態数で安定している．これは，一括して最初に蓄積した 400 個の状態が，対象とする領域の性質をよく反映したものであると考えられる．よって，ディレイをかけている間に蓄積された状態をもとに特徴を構成すると，以降の学習効率の向上に多大な貢献があったものと推測される．

図 3.6より，ディレイを考慮した結果， FCQL は学習初期に急激な傾斜を描いて最適値へと収束し，その後も安定状態を続けている．改良前の状態と比較すると，学習初期での学習速度，および収束ポイントの双方で改善が見られる．また，ディレイによる遅れは，結果として学習全体に影響を及ぼすことはなかったと考えられ，学習効率の向上が見られた．

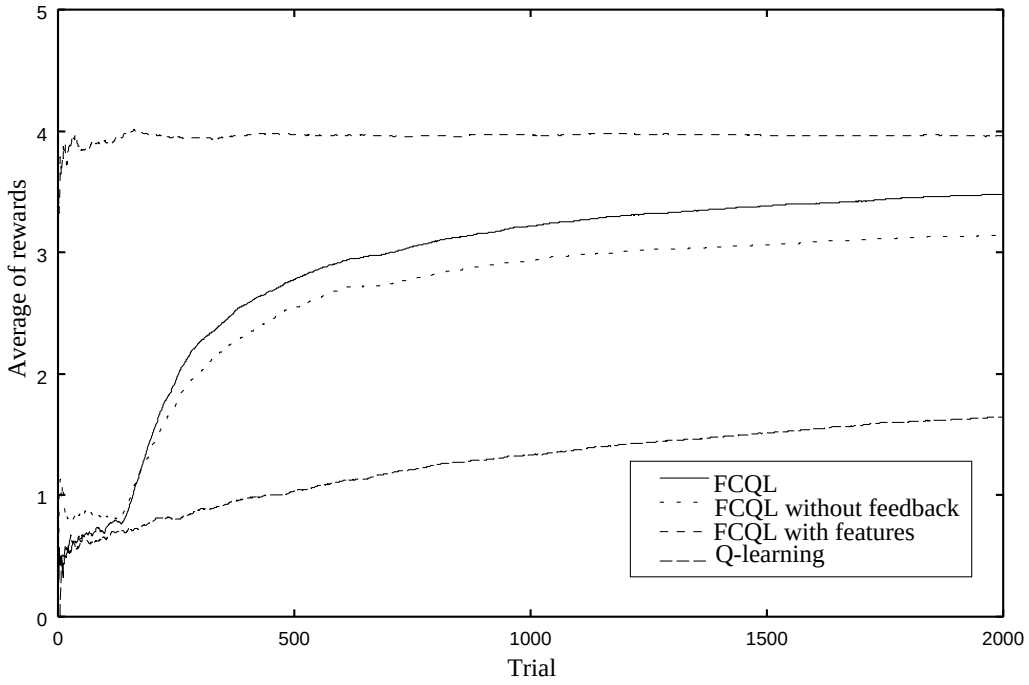


図 3.6: 平均報酬値の推移

また，強化学習からのフィードバックを除いた場合，特徴関数の収束が遅れるため，収束値が若干劣る結果が表れている．

次に，学習中に構成された特徴の例を 図 3.7 に示す．この例では，実際に構成された特徴は 28 個で，そのうち 12 個を構成された順に示している．

正解となる特徴のうち，2 項連言で記述される特徴については，4 つすべてが構成されていた．単項で記述される特徴は，他の不要な属性が混在し，冗長な記述となっているが，報酬と無関係な特徴は一つしか構成されておらず（図 3.7の斜線部），蓄積された状態を分類するのには充分であったと考えられる．また，構成された特徴の連言の項数はほとんどが 2 項までで，最長で3 項のものがごくまれに見られる程度であった．

最後に，構成された特徴が実際に有効に利用されていることを検証する．図 3.8 は，各試行において，それぞれ特徴，および状態に基づいて行動が決定された回数の平均を表している．

FCQL では，状態に含まれる特徴があれば，優先してその特徴に基づく行動決定を行うが，図 3.8より，特徴に基づいて行動が決定された回数の方が試行全体として多いと判断できる．これは，学習初期で構成された特徴が，同じクラスに属する状態の性質をよく表しているため，未経験の状態に対しても同じ特徴を持つ経験済みの状態と同じ行動を選択し，成功したためであると考えられる．したがって，正の報酬が得られる特徴を持つ状態に対しては過去の行動を積極的に追従し，負の報酬が得られる特徴を持つ状態に対しては過去の行動を避ける様に判断され，単位時間あたりの報酬の向上に大きく貢献している．

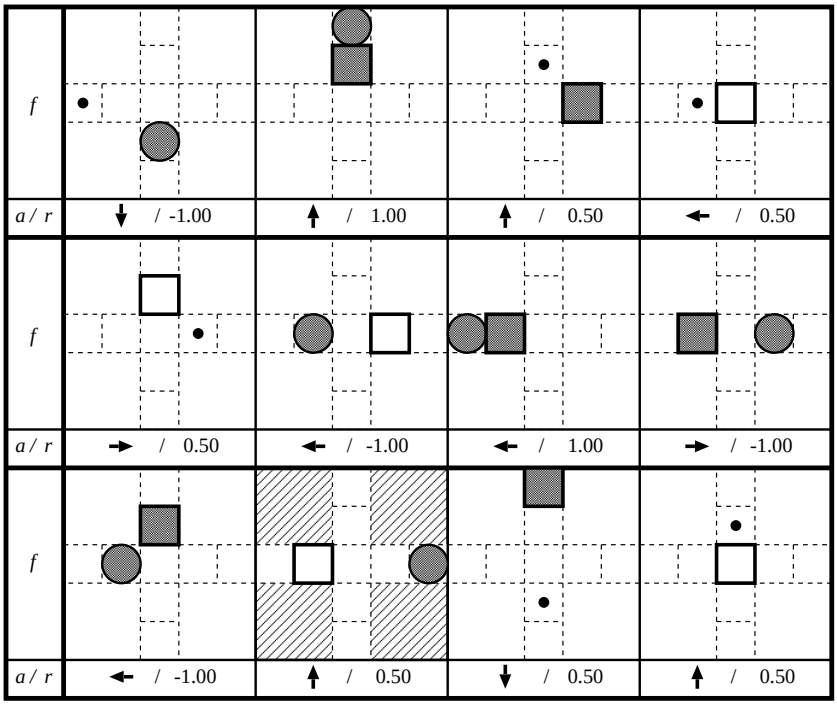


図 3.7: 構成された特徴の一部

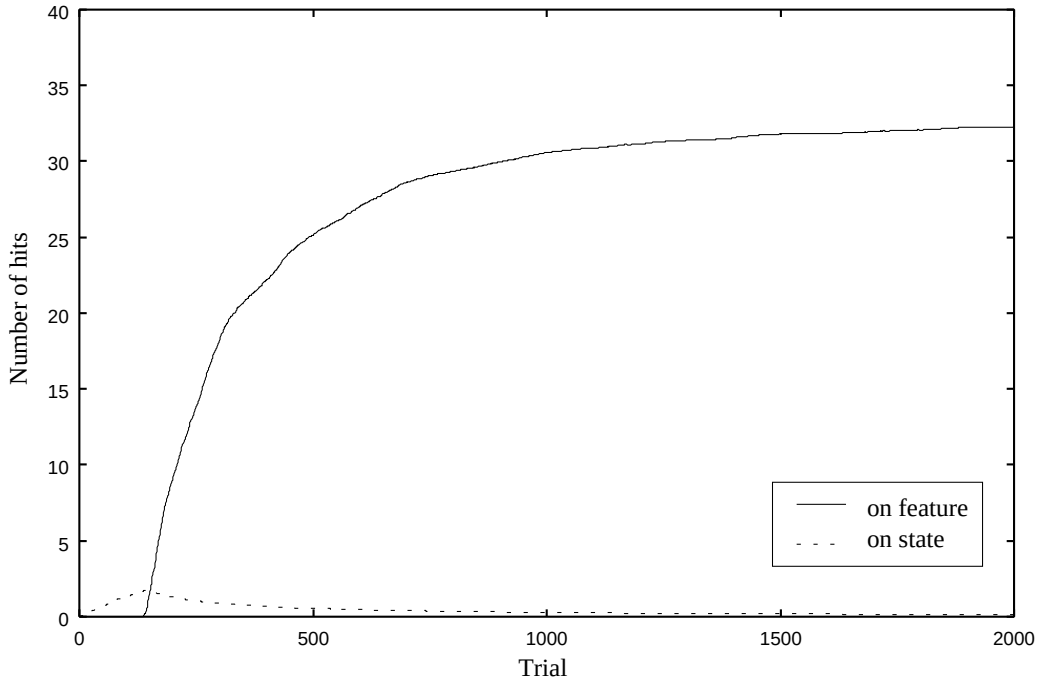


図 3.8: 特徴および状態に基づく行動決定回数

3.6 ディレイに関する検証

FCQL では，ディレイをかける状態数をあらかじめ 400 と設定した．この状態数は，特徴を抽出するために充分な量の状態集合であることを以下に検証する．図3.9 は，ディレイをかける状態数をそれぞれ 0，100，200，300，400，500，700 に設定した場合の報酬の推移を示している．

状態数が少ない場合は，構成される特徴の精度が低いため，不安定な振る舞いを示している．逆に，状態数を増やすと，学習初期において特徴構成を行わないため，報酬の向上が遅くなる．2000 試行終了時において，ディレイをかける状態数を 400 に設定した場合の収束値が最も高いので，本シミュレーションでは特徴構成を始める状態数を 400 個と決定した．

本研究では，従来の Q 学習を行う局面と，FCQL を行う局面の 2 つの局面を設定した．このことは，状態の蓄積と，蓄積された状態からの特徴抽出という 2 つの局面が存在し，それらの局面を切り替えていることを表わしている．シミュレーションでは，最初に Q 学習を行い，その後一度 FCQL に切り替えて効率改善に成功している．これら 2 つの局面を適宜切り替えていくことにより，不適切な特徴が構成された場合でも新たに状態の蓄積の局面に移行することができ，誤った特徴からの学習を回避することができる．シミュレーションでは切り替えを一度としたが，複数回の局面の切り替えを検討すれば，学習領域によって適切な特徴構成を行うことが可能であるといえる．

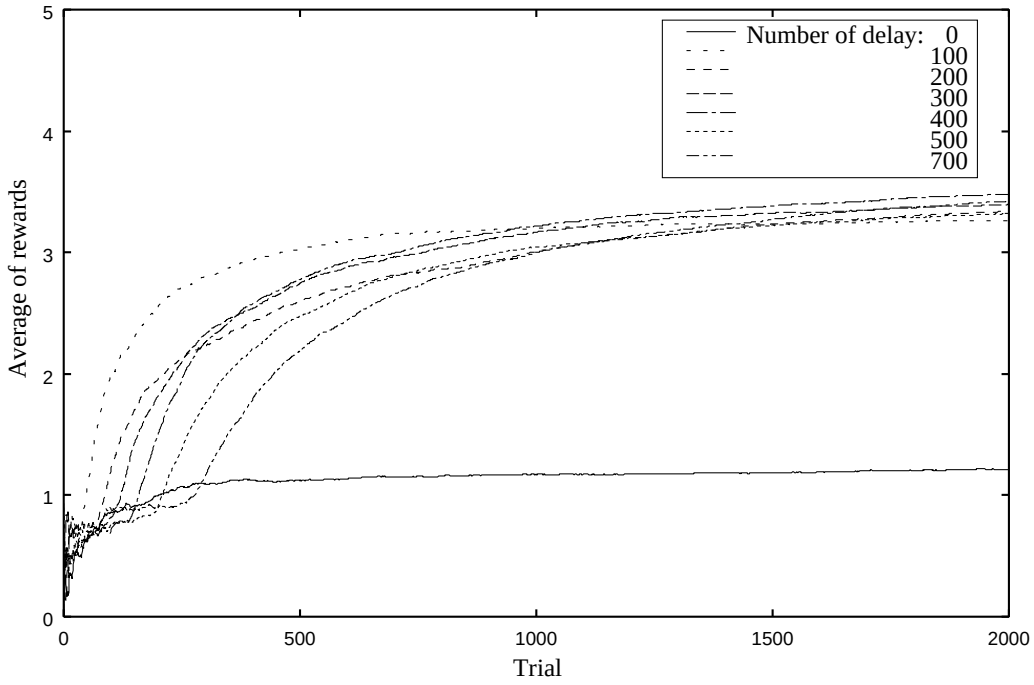


図 3.9: ディレイをかける状態数を変化させた場合の報酬の推移

3.7 結言

本章では，FCQL を人工的な迷路問題に適用し，報酬の推移についてシミュレーションを行った．環境の例として取り上げた迷路問題は，ロボット学習の範疇で用いられる代表的な問題の一つである．しかしながら，実際のロボットを実験に用いる場合には，センサからの情報も連続値であり，FCQL の枠組に組み込むには適切な方法で離散化する必要がある．本章での迷路問題はすでに離散化された状態にあり，一般的な実数値の連続空間を対象とする問題に対しては，適切な離散化を行う操作子の適用によって対応が可能であると考えられる．

また，FCQL では，特徴を構成する手法として論理演算を採用したが，他にも特徴を構成する手法にに発見的な手法の採用が考えられ，この点についても検討の余地がある．対象領域とした DNF 問題は，Q 学習のみでは非効率な問題の例として取り上げたが，最終的には入力属性の組合わせで特徴を構成できるという前提に立っている．状態を識別する要素が，センサからの情報に直接現れてこない問題では，論理演算では得られた状態からの特徴構成ができないため，他の特徴抽出の手法が有利となる場合がある．このような他の領域ての特徴抽出手法について次章で検討する．

第 4 章

画像分類における特徴とその利用

4.1 緒言

第 3 章の結果より，機械学習における適切な特徴の抽出および選択が，学習結果に大きな影響を及ぼすことが示された．また，特徴の抽出には領域ごとに充分な量の事例が必要となることも分かった．第 3 章の対象領域は，人工的な離散環境であったが，現実的な事例についても検証する必要がある．

本章では，現実的な事例としてパターン画像であるテクスチャをとりあげ，その特徴を空間的および周波数的に抽出して，それらの特徴から事例をよりよく表現する特徴に変換することを試みる．また，これらの特徴を用いて，機械学習の一手法である事例に基づく学習による分類実験を行う．

4.2 研究背景

テクスチャは画像の特徴を知る最も有力な手がかりの一つであり，テクスチャ解析の技術は航空写真や医療画像の物体認識などに適用されている．画像認識の分野ではこれまで多くの研究がなされているが，テクスチャを表現する適切な特徴量の決定が難しいため，テクスチャ認識は現在でも困難な問題と考えられている [20] [21] [22] [23] [24] [25] [26] .

テクスチャ認識の難しさとして，異なるスケールのテクスチャを特徴づける適切な手法がなかったことがあげられる．近年，画像解析を行う手段として，**ウェーブレット変換** [27] に代表される特徴抽出手法がこの問題を解消すると期待されている．

り，低周波領域を再帰的に分割する**ピラミッド構造ウェーブレット変換**（Pyramid-Structured Wavelet Transform: PSWT）[28] が有名である．人物や風景などの自然画における物体認識では，対象物に形状が存在する．画像認識のためには変化の緩やかな帯域である低周波帯域が重要であると知られており，変化の激しい中高周波帯域はノイズとして除去される．このため，自然画の解析には PSWT が適していると考えられている．しかしながら，テクスチャには形状が存在せず，画素値の変化が激しいテクスチャでは，そのテクスチャの特徴を表わす重要な帯域は低周波帯域以外に現れることもしばしばある．これらの帯域を

分割しない PSWT はテクスチャ分類には不適切であり，重要な帯域を見逃すという問題が発生する．

このような問題を解決するために，テクスチャの特徴づけには中高周波帯域をも分割する構造化手法が適切であると考えられ，中でも**木構造ウェーブレット変換**（Tree-Structured Wavelet Transform: TSWT）[29] [30] は特に注目されている．TSWT は，画像を重要な空間周波数帯域に選択的に分割し，帯域分割構造（以下，**構造**と呼ぶ）とウェーブレット特徴量で特徴づける手法である．

TSWT をテクスチャ分類に利用した研究に，Chang [31] の研究がある．Chang は，テクスチャの特徴づけには重要な帯域の「位置」と「値」が役立つと考え，それらの帯域のウェーブレット特徴量を選択的に分類に適用した．このときに利用する情報は構造化されたテクスチャの「一部」であり，構造全体は分類に貢献しないという立場をとっている．

TSWT は一定の周波数帯域のみを分割しているのではないため，その結果得られる構造は一般に一意に定まらない．このことは，TSWT によって生成された構造はテクスチャごとに固有のものとなり，ウェーブレット特徴量と並んでテクスチャを表現する特徴量となりうることを示している．PSWT のように常に一意の構造をとる分割手法では，構造自体がテクスチャを表現する手段にはなり得ないが，TSWT で得られた固有の構造はテクスチャ分類に役立つと期待でき，この構造の有効利用が考えられる．

本章では，Chang の手法の誤りを指摘し，TSWT によって得られた構造がテクスチャ分類に重要な特徴量であると主張する．TSWT をテクスチャ分類に適用する際には，ウェーブレット特徴量を適切に抽出するために，構造全体から重要な帯域を選択するボトムアップ的構造化手法を採用する．このとき，構造化されたテクスチャの分布を示すために，新たな指標としてテクスチャエントロピーと呼ばれる尺度を提案する．実験では，様々な条件下でテクスチャエントロピーと分類精度を従来法と比較し，本手法の優位性を示す [32] [33][34][35]．さらに，提案手法が現実的な画像の変化に有力であることを示すため，テクスチャ分類にモーメント特徴量を導入して，実験によりその評価を行う [36]．

以下，第 4.3 節では，ウェーブレット変換の概要と周波数帯域分割手法について述べる．第 4.4 節では，ボトムアップ的構造化に基づくテクスチャ分類手法を提案し，その詳細について述べる．第 4.5 節では，様々な条件下でテクスチャエントロピーと分類精度の比較を行って，実験により本手法の優位性を示す．第 4.6 節では，提案手法の優位性を示すため，テクスチャ分類にモーメント特徴量を導入して，実験によりその評価を行う．最後に，第 4.7 節で本章の結論について述べる．

4.3 ウェーブレット変換と構造化

ウェーブレット変換とは，フィルタ関数を用いて信号を周波数帯域ごとに分割する手法であり，フィルタとして L（ローパスフィルタ）および，H（ハイパスフィルタ）が用いられる．2 次元のウェーブレット変換では，原画像に対して L および H を水平方向と垂直方向に適用し，4 つの領域（LL，HL，LH，HH）に分割する．このとき，低周波帯域のみを再帰的に分割する手法をピラミッド構造ウェーブレット変換，それ以外の帯域につ

いても選択的に分割する手法を TSWT と呼ぶ．図 4.1 は，レベル 3 の TSWT をテクスチャに適用した例である．

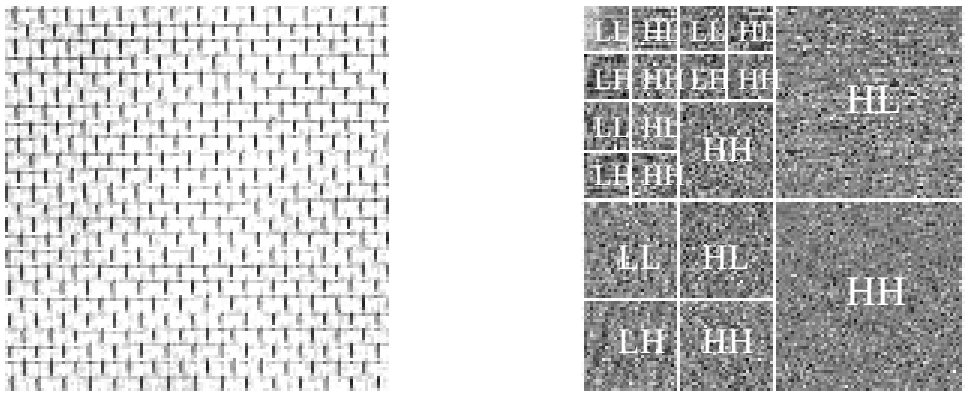


図 4.1: レベル 3 のウェーブレット変換の適用例

図 4.2 は，図 4.1 を構造として表現したものである，

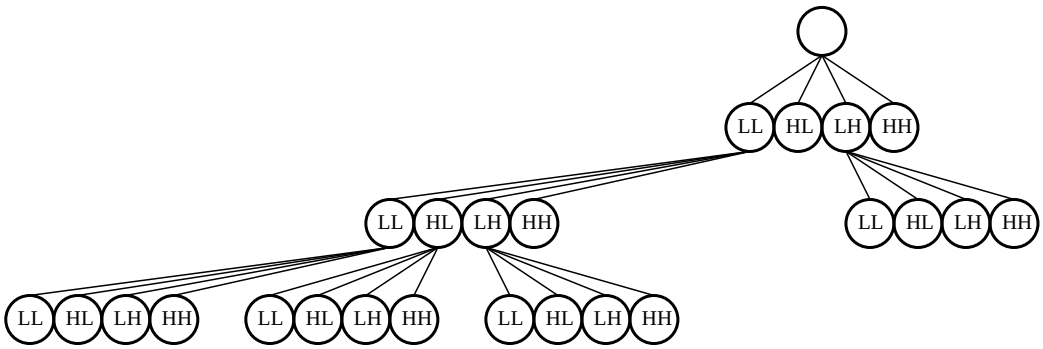


図 4.2: 構造化されたテクスチャの例

本稿におけるテクスチャの構造とは，原画像をウェーブレット変換によって分割し，生成された空間周波数帯域を各ノードとした木構造を持つグラフ構造を意味している．これは，分割前のサイズの画像を親，分割後の画像を子とすると，それらの親子関係を最小単位とし，各ノードを再帰的，選択的に分割して生成できる木構造である．また，原画像より構造を生成する手続きを構造化と呼ぶ．

TSWT による画像の構造化手法は，大別して 2 つある．一つは，**トップダウン的構造化**（$\mathrm{TSWT}_{\mathrm{TD}}$）で，もう一つは**ボトムアップ的構造化**（$\mathrm{TSWT}_{\mathrm{BU}}$）である．$\mathrm{TSWT}_{\mathrm{TD}}$ は，原

画像を指定レベルまで上から順に分割していく手法であり，重要と見込まれる帯域のみを選択しているため，一回あたりの計算量は少なくてすむ．逆に， $\mathrm{TSWT}_{\mathrm{BU}}$ は，一旦原画像を指定レベルまで全分割し，後に末端ノードより重要でない帯域を削除していく手法である．一回の計算量は $\mathrm{TSWT}_{\mathrm{TD}}$ より少なくなることはないが，重要な帯域を見逃すことがない．

このような考え方に基づいて， Chang は，テクスチャの構造化に $\mathrm{TSWT}_{\mathrm{TD}}$ を適用し，原画像を分割する際に分類に重要と見込まれる帯域のみを分割する手法を提案している．図 4.3 に，Chang の $\mathrm{TSWT}_{\mathrm{TD}}$ アルゴリズムを示す．

---

1. 入力画像を TSWT で 4 分割し，これらを木構造の子ノードとする．
2. 各子ノードのエネルギーを計算する．
3. あるノードのエネルギー e が他より特に小さければ，そのノードの重要性は低いと判断し，分割を停止する．この判断は同じスケールで最大のエネルギーを持つノードの値 e_{max} と比較して行われる．すなわち，$e < Ce_{max}$ $(C < 1)$ ならば分割を停止する．
4. 3. の条件を満たさないノードについて，指定レベルを越えない限りは，1. に戻ってさらに分割を行う．

図 4.3: $\mathrm{TSWT}_{\mathrm{TD}}$ アルゴリズム

Chang の $\mathrm{TSWT}_{\mathrm{TD}}$ アルゴリズムでは，ある帯域を分割する際に，それ以下の階層について重要な帯域が出現するかどうかの判断を，見込みをつけて行っている．また，分類時においては，構造の一部のみを利用して構造の近似として扱っている．この手法をテクスチャに適用すると，以下のような 3 つの問題が起こる．

まず第一に，PSWT 同様，テクスチャの特徴となる重要な帯域をうまく取り出せないことである．テクスチャにおける重要なノードは低周波帯域とは限らず，その位置が一定していない． Chang の手法における見込は，重要なノードはその親ノードもまた重要であるという仮定に基づいて行われているが，実際にはこの仮定は必ずしも成り立たない．分割途中で見込みが外れたノードはその子孫ノードの可能性が見逃されてしまい，テクスチャの特徴づけがうまく行われないことが充分に考えられる．図 4.4 に，この問題が起こる例を示す．

図 4.4 で，レベル 1 において 1 番目のノードの値 e_1 が最大であり，他の 3 ノードはすべて Ce_1 より小さい値であるとする．ここで， C は 図 4.3 に示された，見込のためのパラメータである．このとき，レベル 1 の階層を $\mathrm{TSWT}_{\mathrm{TD}}$ で分割すると，ノード 1 以外の

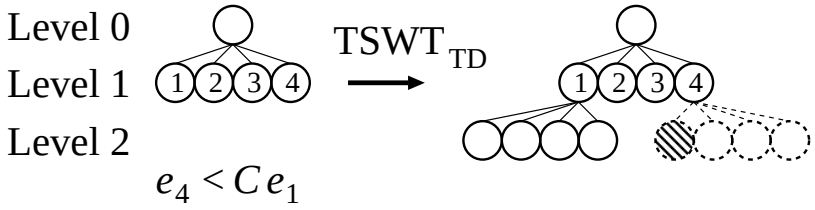


図 4.4: $\mathrm{TSWT_{TD}}$ で問題が起こる例

ノードのさらに下のレベルに重要なノードがあった場合，これを見逃してしまうことになる．たとえば，図 4.4 において，ノード 4 の下のレベルにある斜線部の重要なノードは，$\mathrm{TSWT_{TD}}$ によって見逃されてしまい，特徴として現れてこないという問題が起こる．

第二に，テクスチャを特徴づけるエネルギーが構造ごとに分けられていない点にある．Chang は同じテクスチャ内ならば異なる構造間であってもノードのエネルギーを一緒に平均し，そのテクスチャを特徴づけるエネルギーとしていた．しかしながら，これらのエネルギーは構造ごとに分けなければテクスチャの特徴をうまく表現しているとは言えず，正確な分類が行えないことになる．

第三に，分類時に構造の一部分しか用いない点である．Chang は，各構造についてエネルギーの大きな順に 5 個のノードを取り出し，それらの位置を照合して構造の近似として扱い，構造の出現頻度を計算していた．しかしながら，この手法では構造全体に対する照合を行っていないため，テクスチャ内の構造の分布を正しく表しているとはいえない．たとえば，異なる構造であっても，上位 5 個のノードのみは一致しているという場合が想定され，これらの構造は同じであると扱われる．この一致をもって同一の構造であると判断するのは不適切であり，構造の分布がより一意に近いという方向へと歪曲されてしまう．以下の節では，テクスチャの構造化に $\mathrm{TSWT_{BU}}$ を導入して，$\mathrm{TSWT_{TD}}$ の問題点に対する解消方法について述べる．

4.4 テクスチャのボトムアップ的構造化と分類

4.4.1 分割と構造化

Chang の構造化アルゴリズムでは，あるノードを分割するかどうかの判断が，その直下の子ノードのエネルギーと比較して行われている．この場合に想定しうる不具合は，その子ノードよりさらに下の階層に重要なノードがあってもその重要なノードは出現せず，構造に反映されないことである．このことは，テクスチャごとの特徴が失われ，異なるテクスチャであっても構造が類似するという可能性を引き起こす．また，図 4.3 中の見込みを行うためのパラメータ C についても，対象依存にしか決定できず，C の決定には多くの実験を必要とする．

本稿で提案する $\mathrm{TSWT_{BU}}$ に基づく構造化手法では，あらかじめ全分割された帯域から

重要な帯域を選択するので，Chang の手法のように重要な帯域を見逃すことがない．また，実験結果に左右される見込みのためのパラメータを必要としない．これらの解消法により，テクスチャごとの特徴を表す重要な帯域が失われることなく，また，恣意的なパラメータを用いない汎用的な構造化が期待できる．図 4.5 に， $\mathrm{TSWT}_{\mathrm{BU}}$ に基づく構造化アルゴリズムを示す．このアルゴリズムによって，テクスチャの構造が決定される．

---

1. 入力画像を TSWT で全分割する．
2. 各ノードのエネルギーを計算する．
3. ノード i のエネルギーが，i の子ノードの平均エネルギーよりも大きければ，i の子ノードを削除する．
4. 3. の操作を最下位ノードの一層上から最上位ノードまで順に行う．

図 4.5: $\mathrm{TSWT}_{\mathrm{BU}}$ アルゴリズム

手順としては，まず $\mathrm{TSWT}_{\mathrm{BU}}$ でテクスチャを分割し，構造化されたテクスチャについて各ノードのエネルギー（l_1 ノルム）を計算する．Chang は，l_1 ノルムと l_2 ノルムは識別能力に大差はなく，計算量の点で l_1 ノルムが有用であるため採用したと記している．本研究においても，計算量および Chang の手法との比較という観点から，l_1 ノルムをエネルギーとして採用する．

図 4.5 に示すように，本手法では一旦全分割した後にノードのエネルギーを比較し，これを枝刈りの基準としている．各ノードのエネルギーは，レベルごとの画素数で正規化される．分割された画像の特徴量を $x(m,n)$ （ $1 \leq m \leq M$，$1 \leq n \leq N$，M，N はそれぞれ画像の水平方向と垂直方向の画素数）とすると，エネルギー e は以下の式で計算される．

$$e = \frac{1}{MN} \sum_{m=1}^{M} \sum_{n=1}^{N} |x(m,n)| \tag{4.1}$$

ここで，あるノードについてそのエネルギーが子ノードの平均エネルギーよりも大きい場合は，それらの子ノードは重要でないと判断し，削除する．この操作を最下層より一階層上から順に最上位階層まで行い，枝刈りによる構造化とする．生成された構造は，テクスチャの特徴を表現するのに重要なノードが失われることなく構造化された結果を表している．

4.4.2 テンプレートの生成

構造化されたテクスチャは，その構造により何種類かに分けることができる．理想的には，各テクスチャの構造は一意で表現でき，テクスチャ間の構造はすべて異なるのが望ま

しい．しかしながら，実際には，各テクスチャには複数の構造が存在しうる．このため，テクスチャごとに一括してエネルギーを平均し，そのテクスチャを代表する構造とすることはできない．

Chang の手法では，同じテクスチャであれば，構造が異なってもすべて一括してエネルギーを平均し，テクスチャごとに代表的なエネルギーを作成していた．このため，複数の構造が存在するテクスチャにおいて，それぞれの構造を構成するノードの数や位置が一致しない場合には，異なる構造のエネルギーが混ざってしまう問題があった．このことは，ある構造では重要な値を持つが，別の構造では重要でない値を持つノードを一緒に平均してしまい，Chang の主張する重要なノードの順番が変化する場合も考えられ，代表的なエネルギーを求める手法としては正しくない．

本研究では，あるテクスチャに構造が複数存在すれば，存在する構造ごとにエネルギーを平均し，これらをそのテクスチャを代表するエネルギーを持つ構造として扱う．この平均された構造を**テンプレート**として定義する．テンプレートは，同一テクスチャであっても異なった構造の場合はエネルギーを一緒に平均することはない．各テクスチャに複数のテンプレートを準備すれば，構造ごとに独立した特徴量を保持することができる．この手法により，同一テクスチャで異なる構造間のエネルギーが混ざってしまう問題を解消している．図 4.6 にテンプレート生成アルゴリズムを示す．

---

1. テクスチャ T_i について，存在する構造の種類（k_i 種類）を計上する．
2. T_i 中の構造 j について，その出現数で同じノードのエネルギーの和を平均し，構造 j のテンプレートとする．
3. T_i 中の k_i 種類すべての構造について，2. の操作を行う．
4. すべてのテクスチャについて，1. – 3. の操作を行う．

図 4.6: テンプレート生成アルゴリズム

まず最初に，各テクスチャについて，存在する構造の種類を数え上げる．次に，それぞれの構造について同じノード同士のエネルギーの和を求め，その出現数で各ノードのエネルギーを正規化する．この操作をすべての構造，およびすべてのテクスチャについて行う．これらの操作により，各テクスチャごとに存在する構造について，それぞれの平均的なエネルギーを他の構造から独立に求めることができる．

4.4.3 構造の分布

構造化されたテクスチャの分布は，各構造の出現頻度から計算できる． Chang の手法では，同一テクスチャ内の分布のみに注目し，しかもそれらがなるべく一意に近くなるように見せかけるため，恣意的に構造の一部のみを照合して分布を表現していた．このことは，同一テクスチャ内で唯一の構造を持つことのみが分類に貢献するという考えに基づくものである．

本稿ではこの点を誤りであると指摘し，テクスチャ間の構造の分布についても調査すべきであると主張する．たとえテクスチャ内で一意の構造であっても，他のテクスチャに同じ構造が存在すれば，分類は困難になる．たとえば，すべてのテクスチャが同じ構造しか持たない PSWT では，構造を利用した分類は全く行えない．同様に， Chang の手法では構造の一部しか参照しないため，異なるテクスチャ間で同じ構造が存在する場合が想定される．つまり，テクスチャ内の構造の一意性は，他のテクスチャとの区別ができて初めて活用できると考えられる．しかも，テクスチャ内の構造の分布についても，構造全体を照合してその種類を特定すべきである．

これらの考えに基づいて，本稿では構造全体を参照した分布を，テクスチャ内とテクスチャ間の双方について計算する手法を提案する．この指標を**テクスチャエントロピー**と呼ぶ．テクスチャエントロピーは，テクスチャ内およびテクスチャ間の構造の分布をそれらの出現頻度をもとに計算している．これらテクスチャ内およびテクスチャ間のエントロピーについて，具体的な操作手順を図 4.7 に示し，以下に詳細を述べる．

---

1. テクスチャ T_i について，存在する構造の種類（k_i 種類）を計上する．
2. 各構造について， T_i 内での出現確率 p_j を計算する．
3. T_i 内のテクスチャエントロピー H_{local} を計算する．
4. すべてのテクスチャについて,（1）–（3）の操作を行う．
5. すべてのサンプルについて，存在する構造の種類（n 種類）を計上する．
6. 各構造について，すべてのサンプル内での出現確率 q_l を計算する．
7. テクスチャ間のテクスチャエントロピー H_{global} を計算する．

図 4.7: テクスチャエントロピー計算アルゴリズム

まず，テクスチャごとに存在する構造の種類を数え上げ，構造の分布を計算する．テクスチャ T_i 内のテクスチャエントロピー $H_{local}(T_i)$ は，各構造の出現確率 p_j を用いて以下

の式で計算できる．

$$H_{local}(T_i) = -\sum_{j=1}^{k_i} p_j \log(p_j) \tag{4.2}$$

ここで， k_i は 図 4.6 中に示された構造の種類を表す．式(4.2) は，テクスチャごとの構造が一意に近いほど小さな値となり，構造が一意の場合は最小値 0 をとる．構造が複数種類存在するときは，その頻度に偏りがある場合は 0 に近く，均等な割合で出現する場合は大きな値となる．

次に，すべてのサンプルについて構造の種類を数え上げ，構造の分布を計算する．このとき，異なるテクスチャであっても，構造が一致すれば同じ種類とする．たとえば，PSWT の場合にはすべてのテクスチャについて 1 種類の構造しか存在しないことになる．各構造の出現率を q_l とすると，テクスチャ間のテクスチャエントロピー H_{global} は，各構造の出現率 q_l を用いて以下の式で計算できる．

$$H_{global} = -\sum_{l=1}^{n} q_l \log(q_l) \tag{4.3}$$

ここで， n は 図 4.7 中に示された構造の種類を表す．H_{global} は，テクスチャ間の構造が異なるときほど大きな値となり，このときに分類に有利であると考えられる．

4.4.4 分類

Chang の分類では，一部のノードのみを照合しているため，構造が異なる別のテクスチャについて，それらの一部のノードが一致すれば誤って同じテクスチャと認識してしまう恐れがある．理想的には，テクスチャ間の構造がすべて異なれば，テクスチャごとの特徴が構造によって表現できると考えられ，構造のみを利用した分類が行える．しかしながら， Chang の手法では，テクスチャ内の構造の一意性を高めるために求めた平均的な構造が，他のテクスチャの構造とも似てしまうという結果を引き起こし，構造を利用した分類に高い精度は期待できない．

より分類精度を上げるためには，構造自体もテクスチャを表現する要素の一つととらえ，分類基準に反映させることが重要であると考えられる．この考えに基づき，本稿では，第 4.4.1 節で生成された正確な構造を用いて，構造全体のエネルギーを用いた分類を行う．このことは，エネルギー全体を用いることと，構造自体を利用することの二つの条件を同時に満足する基準となっている．

用いるものである．それぞれのテクスチャを 2 種類のベクトルで表現し，末端に至るまでより正確に構造の照合を行う．この操作により，一部のノードのみが似ている異なるテクスチャに誤分類する可能性を防ぐことができる．図 4.8 に，分類アルゴリズムを示す．

まず最初に，訓練フェイズで訓練サンプルをウェーブレット変換によって分割，構造化し，構造ごとにテンプレートを生成する．この手続きによって各ノードのエネルギーが計算され，これらの値より**エネルギーベクトル**を求めることができる．エネルギーベクトルは，各ノードのエネルギーを要素とし，上位階層より順に並べたベクトルである．一回の

- 訓練フェイズ
  1. 訓練サンプルを TSWT によって分割し，エネルギーを計算する．
  2. すべてのサンプルについて，1. の操作を行う．
  3. 構造ごとにテンプレートを生成し，エネルギーベクトルおよび構造ベクトルを求める．
- 分類フェイズ
  1. テストサンプルを TSWT によって分割し，エネルギーベクトルおよび構造ベクトルを求める．
  2. 各テンプレートからエネルギーベクトルおよび構造ベクトルを取り出す．
  3. 指定された距離関数でテストサンプルとテンプレートの距離を計算する．
  4. 距離が最小のテンプレートと同じテクスチャとしてテストサンプルを分類する．

図 4.8: 分類のための学習アルゴリズム

分割につき 4 つのノードが生成されるため，分割レベルを L とすると，エネルギーベクトルの次元 d_e は以下の式で表せる．

$$d_e = \sum_{i=0}^{L} 4^i \tag{4.4}$$

したがって，エネルギーベクトル $\mathbf{e}$ は，各ノードの値より以下のように表現できる．

$$\mathbf{e} = (e_1, e_2, ..., e_{d_e}) \tag{4.5}$$

構造化の結果削除されたノードのエネルギーはその値を 0 とする．このことによって，エネルギーの値で各ノードの存在を判断でき，エネルギーベクトルは構造全体を表わすベクトルとなる．

また，得られたエネルギーベクトルより**構造ベクトル**を求めることができる．構造ベクトルとは，生成された構造に対し，あるノード i が子ノードを持つ場合には 1，ない場合は 0 の 2 値で各要素 s_i が表現されるベクトルである．最下層には子ノードが存在しないため，分割レベルよりも一つ少ない階層で構造ベクトルを表現できる（図 4.9）．

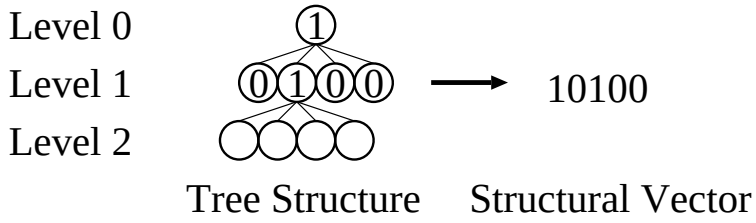


図 4.9: 構造ベクトルの抽出

ここで，分割レベルを L とすると，構造ベクトルの次元 d_s は，以下の式で表せる．

$$d_s = \sum_{i=0}^{L-1} 4^i \tag{4.6}$$

したがって，構造ベクトル $\mathbf{s}$ は，各ノードの値より以下のように表現できる．

$$\mathbf{s} = (s_1, s_2, ..., s_{d_s}) \tag{4.7}$$

分類フェイズでは，テンプレートとテストサンプルの類似度を用いた分類を行う．類似度を測るための距離関数は，構造とエネルギーの双方を利用した尺度を用いる．ここで，サンプル x, y が存在するとき，$D_e(\mathbf{e}_x, \mathbf{e}_y)$ をそれぞれのサンプルのエネルギーベクトル間のユークリッド距離とし，$D_s(\mathbf{s}_x, \mathbf{s}_y)$ をそれぞれの構造ベクトル間のユークリッド距離とすると，x, y 間の各距離はそれぞれ以下の式で定義される．

$$D_e(\mathbf{e}_x, \mathbf{e}_y) = \sqrt{\sum_{i=1}^{d_e} (e_{x,i} - e_{y,i})} \tag{4.8}$$

$$D_s(\mathbf{s}_x, \mathbf{s}_y) = \sqrt{\sum_{i=1}^{d_s}(e_{x,i} - e_{y,i})} \tag{4.9}$$

ここで，$e_{x,i}$ は $\mathbf{e}_x$ の各要素を表わし，他のベクトルについても同様とする．分類時においては，式（4.8）および式（4.9）で計算された距離が最小のテンプレートと同一のテクスチャと判断し，テストサンプルをそのテクスチャとして分類する．

4.4.5 計算量

$\mathrm{TSWT_{BU}}$ の計算量は，結果がどのような構造となっても常に一定である．分割の回数，および比較の回数とも，子ノードを持つノードの個数と等しくなるため，その回数はともに式（4.6）で表わされる．また，$\mathrm{TSWT_{TD}}$ の計算量は構造の結果によって異なる．最悪の場合は，同じ階層の 4 つのノードの値にほとんど差がない場合で，このときの分割回数は全分割に等しく，$\mathrm{TSWT_{BU}}$ の場合と同じく式（4.6）で表わされ，比較回数はその 3 倍となる．逆に，最も計算量が少ない場合は，ある一つのノードの値のみが他の 3 つのノードよりも著しく大きい場合で，このときの分割回数は L 回，比較回数は $3L$ 回となる．

以上のように，$\mathrm{TSWT_{BU}}$ は画像を一旦全分割した後に枝刈を行う手法であるため，一回あたりの計算量は $\mathrm{TSWT_{TD}}$ に比べて多くなることが予測される．しかしながら，$\mathrm{TSWT_{TD}}$ では見込のためのパラメータ C をあらかじめ決定することができないため，適切な C の値は繰り返し検証して決定され，その分の計算量の増加が見込まれる．

4.5 実験

4.5.1 実験データ

実験には Brodatz [37] のテクスチャアルバムより得られた 25 種類のテクスチャを用いる（表 4.1）．各テクスチャは解像度 100 dpi で取り込まれ，256 階調のグレイスケール形式の画像となっている．付録 A に，実験に用いた画像データを示す．

実験では，$\mathrm{TSWT_{BU}}$ および，比較対象として $\mathrm{TSWT_{TD}}$，PSWT のそれぞれのアルゴリズムについて評価を行う．分類では，距離関数をエネルギーのみの距離，構造のみの距離，および双方を用いた場合の 3 通りに変化させる．また，訓練サンプルとテストサンプルに重複を許す場合と許さない場合を設定し，これらすべての組み合わせについて実験を行う．

経験的に，分割を停止するのに適切な最小の画像サイズが知られている．小さく分割しすぎると，サンプル間のばらつきが出て，不安定になる．これらの報告 [38] [31] より，実験では分割の最小サイズを 32×32 画素とする．また，入力画像のサイズを 256×256 画素とし，レベル 3 のウェーブレット変換を行う．

各サンプルの生成方法は，まずサイズ 512×512 の画像を左右二つに分け，それぞれの領域から 256 個の部分画像を所定のサイズで切り出す．次に，各テクスチャごとに訓練サンプルを左側から，テストサンプルを重複を許す場合は左側，許さない場合は右側からラ

表 4.1: 実験に用いたテクスチャ

ID	Texture Description	ID	Texture Description
D3	Reptile skin	D68	Wood grain cloth
D6	Woven aluminum wire	D74	Cotton
D9	Glass lawn	D77	Coffee beans
D11	Homespun woolen cloth	D79	Oriental grass fiber
D16	Herringbone weave	D82	Oriental straw cloth
D19	Woolen cloth	D83	Woven matting
D21	French canvas	D84	Raffia looped to a high pile
D29	Beach sand	D92	Pigskin
D34	Netting	D95	Brick wall
D53	Oriental straw cloth	D102	Cane
D55	Straw matting	D103	Loose burlap
D57	Handmade paper canvas	D105	Cheesecloth
D65	Handwoven Oriental rattan		

ンダムにそれぞれ 100 個選択する．この結果，訓練サンプル，テストサンプルのそれぞれについて， 25 種類のテクスチャごとに 100 サンプル，合計 2500 個のサンプルが生成されたことになる．また，図 4.3 のアルゴリズム中のパラメータは $C = 0.3$ とし，ウェーブレット変換のフィルタには Daubechies 20 を用いる．図 4.10 に，実験の概要を示す．

4.5.2 実験結果：構造の分布

本項では，構造化されたテクスチャの構造の分布を 第 4.4.3 項の手法で解析した結果を示す． 表 4.2 は， $\mathrm{TSWT_{BU}}$ ， $\mathrm{TSWT_{TD}}$ および PSWT に基づいて構造化されたテンプレート数とテクスチャエントロピーを各テクスチャごとに計算し，その後 25 種類のテクスチャについて平均した値を示している．テンプレート数は，各テクスチャの構造の分布を知る簡易な指標として活用できる．また，テクスチャエントロピーは，構造の一意性が高いほど 0 に近い値が得られ，テクスチャ内ではその値が小さいほど，テクスチャ間ではその値が大きいほど分類に貢献すると考えられる．

PSWT ではすべてのテクスチャの構造が同じであるため，これらテクスチャエントロピーの値はテクスチャの特徴を示さない．$\mathrm{TSWT_{TD}}$ では，大胆な見込みをつけて分割を行うため，構造は多様化せず，H_{local} の値が小さいことがテクスチャ 1 種類あたりの構造数が少ないことを示している．しかしながら，H_{global} の値が小さく，分布にばらつきがないと推測されるため，テクスチャ間の構造に関しても似た構造をとるものが多いという結論が導かれ，分類時に区別しにくいということが予想される．

逆に，$\mathrm{TSWT_{BU}}$ では，H_{local} の値が $\mathrm{TSWT_{TD}}$ と比較して大きく，テクスチャ 1 種類

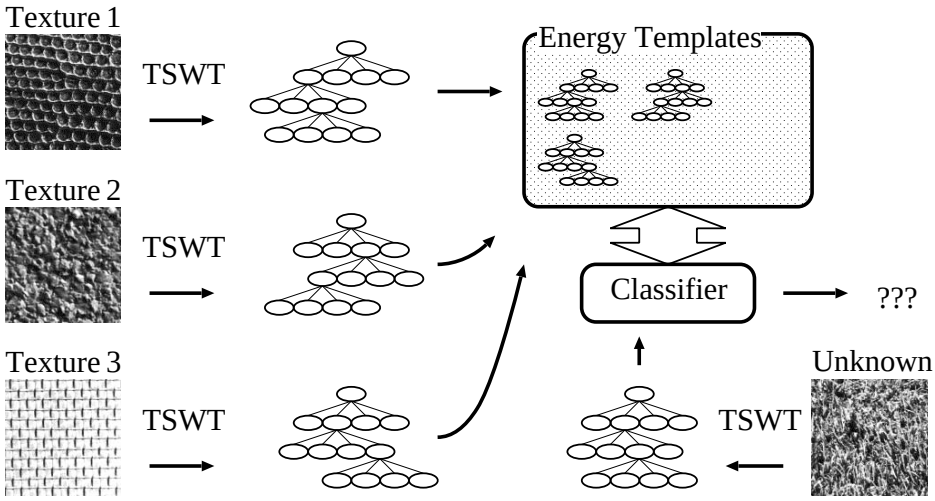


図 4.10: 実験の概要

表 4.2: 各アルゴリズムの構造の分布

構造化手法	テンプレート数		テクスチャエントロピー	
	k_i の平均	n	H_{local}	H_{global}
$\mathrm{TSWT_{BU}}$	2.44	29	0.47	2.61
$\mathrm{TSWT_{TD}}$	1.32	14	0.12	1.33
PSWT	1.00	1	0.00	0.00

あたりの構造数は $\mathrm{TSWT_{TD}}$ よりも多いといえる．しかしながら，テクスチャ間では互いに異なる構造をとっており，テクスチャごとの特徴が構造にはっきりと現れていることが H_{global} の値より読み取れる．このことは，$\mathrm{TSWT_{BU}}$ による構造化が冗長な構造を生成するわけではなく，むしろ各テクスチャごとの特徴を複数の構造で表現し，他のテクスチャとの区別ができていることを示している．H_{local} は，その値が小さいほど一意に近い構造をとるが，現実のテクスチャでは分布が単純であるとは限らず，このときに高い分類精度を示すとは一概には言えない．このような構造の分布と分類精度との関連について次項で述べる．

4.5.3 実験結果：分類精度

本項では，構造化されたテクスチャの分類精度を第 4.4.4 項の手法で求めた結果を示す．表 4.3 は，各手法に基づいて構造化されたテクスチャの分類精度の平均を示している．表 4.3 中の D_e は距離関数として式（4.8）を，D_s は式（4.9）を用いた場合を示している．また，複数の候補が最後まで残った場合は分類失敗とした．

表 4.3: 各アルゴリズムの分類精度

構造化手法	サンプルの重複の可否	分類精度 (%)	
		D_e	D_s
$\mathrm{TSWT_{BU}}$	重複あり	99.8	41.6
$\mathrm{TSWT_{TD}}$	重複あり	99.1	26.9
PSWT	重複あり	98.6	0.0
$\mathrm{TSWT_{BU}}$	重複なし	94.7	45.7
$\mathrm{TSWT_{TD}}$	重複なし	82.4	23.7
PSWT	重複なし	82.0	0.0

表 4.3 より，全体に $\mathrm{TSWT_{BU}}$ による分類精度が高いことから，PSWT ，$\mathrm{TSWT_{TD}}$ に対する $\mathrm{TSWT_{BU}}$ の優位性が示されている．特に，訓練サンプルとテストサンプルに重複がない場合は，$\mathrm{TSWT_{BU}}$ が他の二者に対してかなり高い値を示している．このことは，すべての帯域を考慮した構造化手法が，テクスチャごとの特徴をよく表す帯域の抽出に適しており，未知のテクスチャの分類に有効であるといえる．

さらに，構造自体もテクスチャの特徴をよく表していることは，構造ベクトルのみを距離関数に用いた場合の分類精度から判断できる．絶対的な分類精度自体は低いため，構造のみで分類するのは現実的ではないが，すべて同一の構造となる PSWT では不可能な構造のみでの分類について，$\mathrm{TSWT_{BU}}$ ではその可能性を示したといえる．$\mathrm{TSWT_{BU}}$ における D_s 欄の値が $\mathrm{TSWT_{TD}}$ に対して高いことは，$\mathrm{TSWT_{BU}}$ による構造化が各テクスチャに固有の構造を生成している可能性が高いと言える．

4.5.4 考察

表 4.2 および表 4.3 より，現実的なテクスチャ分類では構造が一意に定まりにくいため，$TSWT_{BU}$ を用いて構造化し，テンプレートを各テクスチャあたり 2 〜 3 種類生成して各テクスチャの特徴を表現するのが最もよい分類精度を導くことが分かる．

以上の結果より，$TSWT_{BU}$ がテクスチャ分類に優れていることが分かった．特に，訓練サンプルとテストサンプルに重複がない場合の精度の劣化を最小限におさえているが，さらなる分類精度の向上が望まれる．現実には，訓練で用いたサンプルと全く同一のサンプルを扱うような状況は想定しがたいため，訓練サンプルに含まれないサンプルを分類する能力が求められる．

本稿におけるサンプルの重複とは，訓練に用いたサンプルをそのまま分類テストに用いることを意味している．つまり，サンプルの切り出し位置や向きが変化した場合において，仮にサンプル間に重複領域が存在したとしてもそれらを異なるサンプルとして扱う．このような微妙なずれは，現実問題として充分に想定しうることであり，それらを同一のテクスチャと見なす能力が必要であると考えられる．

4.6 モーメント特徴量の導入

4.6.1 変化に対する識別能力

本節では，未知のサンプルに対する分類精度の向上のために，テクスチャを特徴づける要素として**モーメント特徴量**を導入する．モーメント特徴量は，パターン認識の分野で広く用いられている特徴量である．

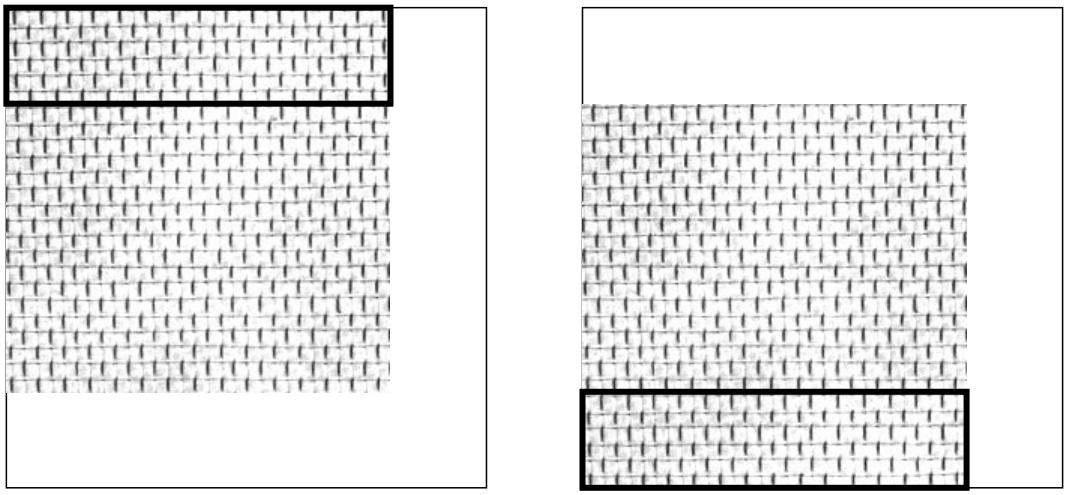
一般に，分類したい画像が常に訓練サンプルと同じ向きや縮尺であるとは限らない．このような状況を画像が変化したといい，モーメント特徴量は画像の変化に対して識別能力が高いという事実が知られている [39].

画像の変化には，移動，回転，縮尺の 3 種類がある．以下では，1 次モーメント特徴量を例に，モーメント特徴量が各変化について不変量となることを説明する．

まず移動であるが，テクスチャでは，物体の位置というものを想定していないので，移動についての考え方が物体認識の場合と異なる．図 4.11 に，1 次モーメント特徴量に対する移動の変化についての考え方を示す．テクスチャは，周期性のあるパターン画像であるため，サンプルの切り出し位置を移動させて失われた画素の部分（図 4.11 の太枠部分）は，新たに現れた画素の部分とグレイレベルの分布で比較するとほとんど差がないことが予測される．

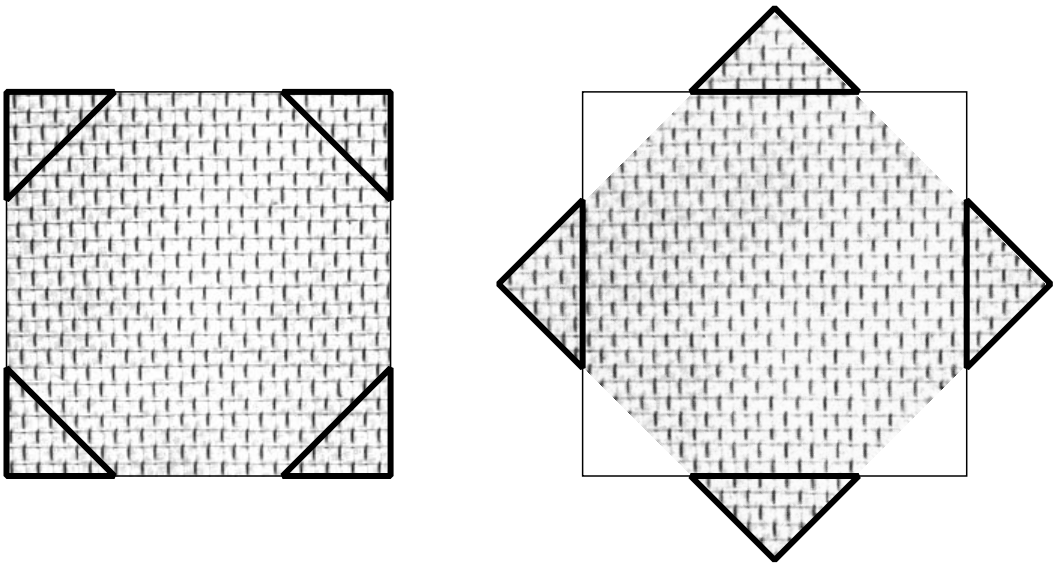
第 4.5 章の実験では，サンプルの切り出しの始点を変化させているため，これを移動の変化ととらえることができる．

次に，回転の変化に対する画素値の分布を図 4.12 に示す．図 4.12 の回転画像における太枠の領域は，原画像から欠けた領域の画素値の分布と，ほとんど差異がないことが予測される．また，回転については，90 度単位の回転では変化後の画素がすべて変化前と同じとなるため，90 度単位以外の回転で検証する必要がある．



Original Image　　Transitted Image

図 4.11: 切り出し位置の移動による変化



Original Image　　Rotated Image

図 4.12: 回転による変化

また，縮尺については，本稿では拡大の変化を想定する．自然画では，拡大の変化を加えると物体が枠よりはみ出し，縮小の変化を加えると背景の領域が増えるため，それぞれ画素値の分布は大きく変化してしまう．テクスチャは，周期性を持つ一様なパターン画像であるとの定義から，これらの拡大，縮小の変化を加えても，画素のグレイレベルの分布に大きな差異はないと予想できる．

図 4.13 に，表 4.1 におけるサンプル ID: D6 の原画像および上記三変化を加えた画像について，256 階調ごとの出現確率をグラフにして示す．図 4.13 より，4 つの画像についての画素値の分布が，変化を加えてもほぼ不変であると言える．また，この画素値の分布はテクスチャに固有のものであり，他のテクスチャとは異なる．図 4.14 に，表 4.1 におけるサンプル ID: D3 の原画像の画素値の分布を示す．図 4.13 および図 4.14 より，これら 2 つの画像の画素値の分布が異なることから，画素値の分布はテクスチャに固有であると推測できる．

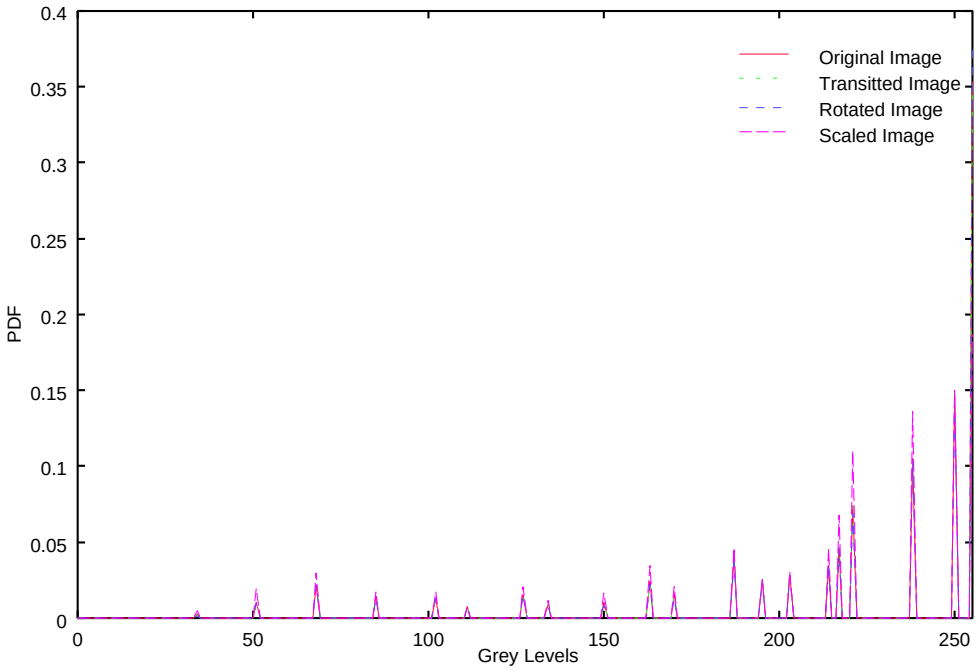


図 4.13: 画素値の分布：D6 の 4 つの画像の比較

以上，検証してきたように，テクスチャに対して移動，回転，縮尺の変化を加えても画素値の分布にはほとんど差はなく，モーメント特徴量は画像の変化に対して不変な特徴量であると言える．

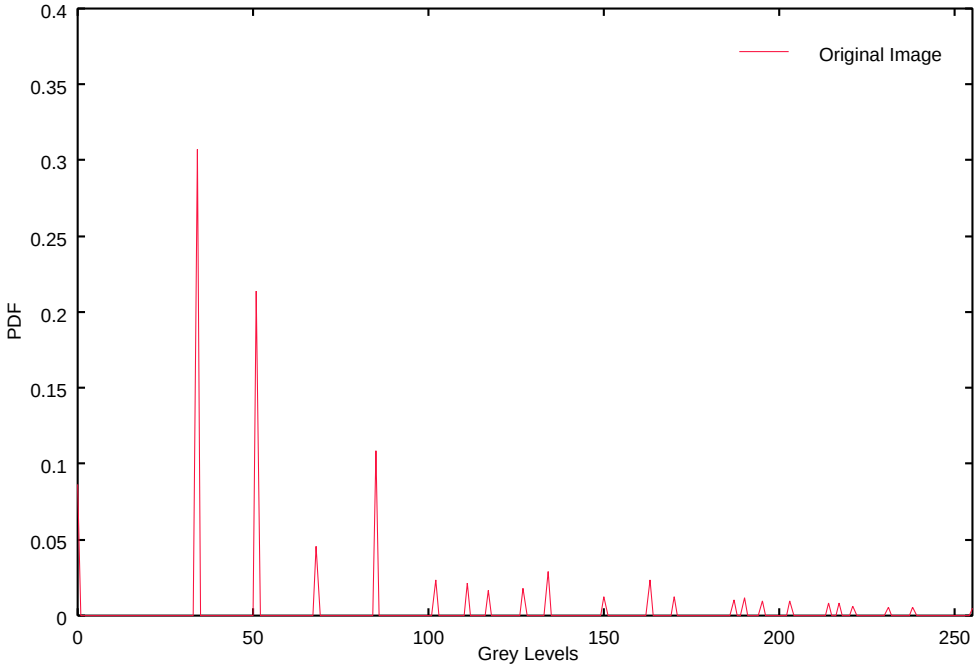


図 4.14: 画素値の分布：D3

4.6.2 モーメント特徴量の概要

モーメントを画像の特徴量として利用する研究には，Terrillion [40]，Milanese [39]，Li [41] の研究などがある．これらの研究は，前処理として対象画像の画素値を二値化し，物体認識用に輪郭や形状を抽出している．しかしながら，テクスチャには形というものが存在せず，輪郭を抽出することはできない．また，多段階のグレイスケールの画素で表現されているため，画素を二値化すればテクスチャとしての情報が失われてしまう．したがって，これらの手法をそのまま適用し，二値化した画像を用いて分類を行うことはできない．そこで，本研究では Mandal [42] の論文にあげられているヒストグラムに基づくモーメントを用いる．

ヒストグラムは，各画像について画素値ごとにそれらの頻度を算出して求めることができ，これを画素のグレイレベルの確率密度関数（Probability Density Function: PDF）$f(x)$ の近似とする．ヒストグラムに基づくモーメント特徴量は，関数 $f(x)$ をもとに計算できる．ウェーブレット変換された分割画像については，各ウェーブレット係数を画素に対応させて，モーメント特徴量を計算する．ここで，関数 $f(x)$ の k 次のモーメント特徴量 M_k は，以下の式で計算できる．

$$M_k = \int_{-\infty}^{\infty} x^k f(x) dx \quad (k \geq 1,\ k \in Z) \tag{4.10}$$

式 (4.10) に基づいて，1 次モーメント M_1 から最大 K 次モーメント M_K までについて，ノードごとにすべてのモーメント特徴量を計算する．1 次モーメント特徴量は画素値の総

和となるため，画素値の平均をとったエネルギーと同じものと見なせる．これら 1 次から K 次までのモーメント特徴量を要素とするベクトルを各ノードについて求める．さらにこのベクトルを要素とし，構造を構成するすべてのノードについて求めたベクトルを，**モーメントベクトル**とする．

ここで，$D_m(\mathbf{m}_x, \mathbf{m}_y)$ をサンプル x, y のモーメントベクトル $\mathbf{m}_x, \mathbf{m}_y$ 間のユークリッド距離とすると，モーメントを用いた x, y 間の距離は，式（4.8）になぞらえ，以下の式で定義される．

$$D_m(\mathbf{m}_x, \mathbf{m}_y) = \sqrt{\sum_{i=1}^{d_m} (m_{x,i} - m_{y,i})} \tag{4.11}$$

$m_{x,i}$, $m_{y,i}$ はそれぞれ $\mathbf{m}_x$ ， $\mathbf{m}_y$ の各要素を表わす．また，$\mathbf{m}$ の次元 d_m は d_e の K 倍に等しい．図 4.8 で示されるアルゴリズムに基づき，距離関数として式（4.11）で計算された距離が最小のテンプレートと同じテクスチャとして分類する．

4.6.3 テクスチャに変化を加えた分類実験

本項では，モーメント特徴量の導入による各変化ごとの分類精度の変量を実験によって評価する．実験では，分割手法として $\mathrm{TSWT_{BU}}$ を用い，訓練サンプルとテストサンプルに重複を許さない場合を想定し，モーメント数を $K = 8$ とする．また，回転の実験における回転角を 45 度，縮尺の実験における拡大率を 2 倍とする．図 4.15 に，これらの変化を加えたサンプルの例を示す．

表 4.4 に，モーメント特徴量の導入前後の変化ごとの分類精度を示す．表 4.4 中の D_m, D_e は距離関数としてそれぞれ式（4.11），式（4.8）を用いた場合の分類精度を表している．

表 4.4: モーメント特徴量を用いた各変化の分類精度

構造化手法	変化	分類精度 (%)	
		D-1	D-4
$\mathrm{TSWT_{BU}}$	移動	95.3	94.8
$\mathrm{TSWT_{BU}}$	回転	75.0	66.8
$\mathrm{TSWT_{BU}}$	縮尺	32.9	31.4

表 4.4 より，まず，移動の変化を加えた場合において分類精度が向上していることが分かる．回転の変化を加えた場合，全体的な分類精度は高くないが，この条件下においてもモーメント特徴量を用いた距離関数による分類結果が分類精度の向上を示している．また，縮尺の変化を加えた場合は，その分類精度の低さから，ほぼ異なるテクスチャであると認識されているが，この場合においてもモーメント特徴量の導入により分類精度の向上が見られた．これらの結果より，ウェーブレット変換自体は回転・縮尺に対して不変ではない

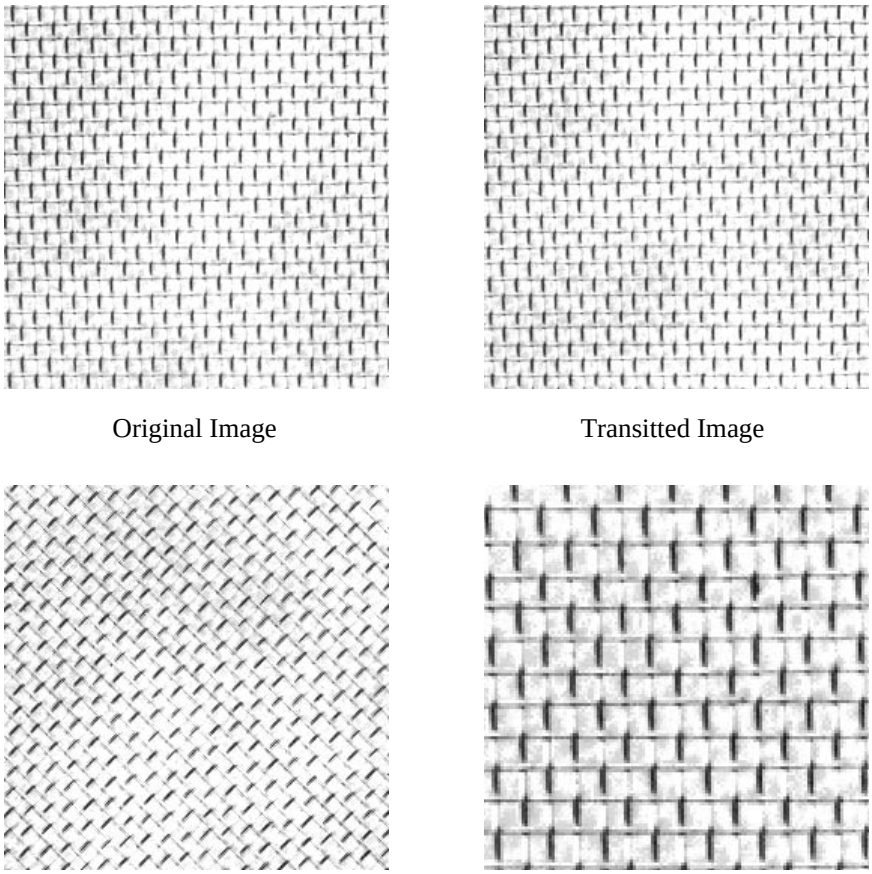


図 4.15: 変化を加えたサンプル

が，モーメント特徴量の不変性を利用して，モーメント特徴量の導入により分類精度が向上することが示された．

4.7 結言

本章では，テクスチャ分類において特にその構造に着目し，構造の分布を表わす新たな指標としてテクスチャエントロピーを提案した．また，各テクスチャごとにテンプレートを作成する手法を提案し，様々な条件と構造化手法においてテンプレートを用いた分類精度の比較を行った．実験の結果，構造化されたテクスチャの構造の分布が構造化アルゴリズムに強く依存していることを示し，定量的な評価を行った．分類においては，PSWT および $\mathrm{TSWT}_{\mathrm{TD}}$ に対する $\mathrm{TSWT}_{\mathrm{BU}}$ の優位性を示した．さらに，モーメント特徴量の導入による分類精度の向上と，変化に対するパフォーマンスの評価を行った．

以上，本章の結果より，現実的な事例においても，それらの事例をよく表わす特徴が存在し，特徴を用いた適切な学習が行えることを示した．

第 5 章

知能ロボットのための特徴空間の構成

5.1 緒言

前章までの研究により，複数の対象領域において事例から特徴を抽出し，特徴を用いた有効な学習が行えることを示すことができた．これらの領域はいずれも静的であり，一度構成された特徴が常に有効となる場合を想定している．現実的な例では，対象領域が動的に変化し，今まで有効であった特徴が学習に貢献しなくなる場合が想定される．

本章では，学習中に環境が変動するような領域を対象とし，特徴の評価値を環境に適合させて学習局面を切り替える手法を検証する．本手法は特徴を用いた状態空間を行動モデルとして自律的に獲得し，現在の環境が属するモデルを判別して適切な学習を進行させていく手法である．

5.2 研究背景

現在，実環境において自律的に行動を獲得できるロボットシステムの開発が求められている．一方，機械学習の分野において研究が進められている強化学習 [1] は，環境に対する先見的知識を前提としない，漸次性に優れた学習手法として，このようなシステムへの有効性が注目されている．

従来の強化学習の枠組みは，状態観測の完全性を仮定したマルコフ決定過程（Markov Decision Process: MDP）としてモデル化され，シミュレーション環境下において有効である．しかしながら，実環境では状態観測に不完全性や不確実性が存在し，行動に非決定性やノイズが含まれるため，その有効性が必ずしも保証されないという問題がある．また，実環境におけるロボットの行動獲得では，変動する環境に順応する能力が求められる． MDP 環境を前提とする従来の強化学習では，環境が変動する場合，以前強化されたルールを新しい環境においてそのまま用いることはできない．これらの問題を解決するために，実環境での行動獲得を目的とした状態空間の構成が求められる．

ロボットの行動獲得のための状態空間の構成法として，浅田ら [43] [44] の研究がある．これらの研究では，ロボットが獲得した状態を選択した行動ごとに分別し，状態空間を楕

円体モデルや局所モデルで近似している．しかしながら，これらの手法における入力要素は画像処理によってあらかじめ必要な要素が選択されており,「どの要素が有効であるか」という，状態分類における最も重要な部分が既知であるため，距離指標のみで状態を判別できるようになっている．また，扱うロボットの自由度が高く，状態空間と行動が密接に関連しているため，ロボットの立場に立った論理的な状態空間を構成しているとは言いがたい．

本章では，すべて対等で曖昧な情報源である光センサを用い，固定点に拘束された多関節型ロボットアームを対象とする．モデル構築手法として，センサからの情報をもとに，概念学習を用いて適切な入力要素を自律的に選択しながら行動モデルを生成し，強化学習によってモデルを拡張する手法を提案する．また，モデルに対する信頼度を設定し，生成されたモデルを切り替える学習手法を提案する [45][46]．本手法の有効性を示すために，ロボットアームを用いた実環境での実験を行う．実験により検証するポイントは以下の 2 点である．

- 実環境における行動モデルの自律的獲得
- 変動する環境への順応

以下，第 5.3 節では，ロボット学習と学習モデルに関する研究動向を提示し，それらに対する本研究の位置づけを行う．第 5.4 節では，ロボット学習システムに強化学習を適用する際に，本研究で対象とする環境の特性について述べる．第 5.5 節では，状態観測の不確実性，不完全性と，広大で連続な状態空間に対処するために，概念学習を用いて行動モデルを生成・拡張する手法について述べる．さらに，生成された行動モデルのモデル信頼度を強化し，変動する環境に順応する手法について述べる．第 5.6 節では，第 5.5 節で説明した行動モデルの生成と拡張に対する実験を行い，その有効性を示す．また，移動する光源を追従する学習問題に本手法を適用する実験を行い，結果を考察する．最後に，第 5.7 節でまとめと今後の課題について述べる．

5.3 ロボット学習と学習モデル

5.3.1 ロボットの動作計画問題

ロボットの形状，動作空間中の物体の形状とその位置・姿勢が与えられたとき，ロボットの初期位置から目標位置に到る連続な動作系列を生成する問題を，ロボットの動作計画問題と呼ぶ． Latombe [47] の研究では，ロボットの動作計画問題に対して，ロードマップ法，セル分解法，ポテンシャル法などの枠組みが示されている．このうち，ポテンシャル法は環境からの入力情報に基づいて状態空間を局所的に探索することを用いた動作計画法であり，本章における位置情報を与えられないロボットアームの動作計画に対応する．

本手法における行動モデルは，状態空間の局所的な特徴を一般化して生成される状態遷移モデルであり，ポテンシャル法の枠組みに入るものである．

5.3.2 経由点と教示を用いた強化学習

変動する環境におけるロボットの動作獲得の研究として，経由点表現を用いた強化学習法 [48] [49] がある．この手法において，ロボットの行動は，人間が教示したタスク遂行に基づく教師付き学習によって学習される．また，タスクの達成を学習する過程において，教示された運動軌道を強化学習によって修正するために物理法則に基づいた数式モデルを使用している．

これに対して，本手法はロボットが事前に数式モデルや知識を持ち得ない状況での，教師なし学習による行動獲得を前提としている．教師なし学習は，学習の効率などの点において教師付学習に劣るが，教師によって教示された行動の精度にとらわれず，学習の最適性が保持されるという利点を持つ．

5.3.3 説明に基づく強化学習

状態空間の一般化に基づく学習手法として，説明に基づく強化学習（Explanation-Based Reinforcement Learning: EBRL）[50] がある．EBRL は，説明に基づく学習（Explanation-Based Learning: EBL）[51] と強化学習を融合した学習手法で，行動の効果を表す完全なモデル（状態遷移則）を背景知識としてあらかじめ学習システムに与え，このモデルを利用して状態を汎化している．状態の汎化を行わない通常の強化学習では，単一の状態における報酬値が獲得されるのみであるが，EBRL では領域における期待報酬値が一度に獲得されるため，広大で連続な状態空間を持つ実環境においても学習の収束が早く，また簡潔に状態空間を記述することができるので，必要な記憶領域も抑えられるという利点がある．

しかしながら，獲得すべき領域は行動者にとって未知であることが多く，このような場合に環境の完全なモデルをあらかじめ準備することは不可能であるといえる．本手法における行動モデルは，未知の環境でロボットが実際にセンサを介して獲得した遷移経験をもとに状態遷移の規則を生成して，この問題に対応している．

5.4 強化学習と環境の観測性

5.4.1 MDP 環境下における強化学習

強化学習とは，ある状態に対する望ましい行動出力を，学習者に行動の評価値である**強化信号**を与えて強化する学習手法である．最も著名な強化学習である Q 学習 [3] は，状態遷移確率などの環境に対する事前知識を必要とせず，また MDP 環境下においては多数の試行のあとに Q 値が収束することが保証されているため，現在，多くの問題に適用されている．

しかしながら，本章の実験環境において，状態観測はアーム先端に取りつけられた 3 方向の光センサのみによって行われる．このため，環境を部分的にしか観測できず，観測値にはノイズがともなうため，同一の状態において異なる観測値が得られたり，異なる状態において同一の観測値が得られたりする可能性がある．つまり，ロボットは自身の状態を

完全には認識できず，MDP 環境を前提としている従来の強化学習をそのまま実環境に適用することは非常に困難である．

5.4.2 POMDP 環境における強化学習

部分観測マルコフ決定過程（Partially Observable MDP: POMDP）環境は，MDP 環境に状態観測の不確実性を付加して拡張した環境であり，完全な状態認識が不可能であるという点に関して実環境と関係が深い．POMDP 環境では，**信念**と呼ばれる状態の確率分布を求め，ロボットが環境のどの状態にいるのかを確率的に表して，強化学習による問題解決を行う手法が提案されている [52].

また，POMDP 環境における問題は，行動主体の内部に環境のモデルを生成し，信念の状態空間を生成して， MDP 環境の問題へと帰着することが可能である [53] [54]．信念は，状態数が n であるとき，

$$\vec{\pi}(t) = \langle \pi_1(t), \pi_2(t), \ldots, \pi_n(t) \rangle \tag{5.1}$$

のように確率ベクトルの形式で表現される．ここで $\pi_i(t)$ は時刻 t におけるロボットの状態が i である確率を表す．ロボットが時刻 t に状態 i において行動 a をとり，時刻 $t+1$ に状態 j に遷移し観測値 x を観測する状態遷移確率を $P^a_{i,j}$ ，状態 j において x が観測される確率を $b_j(x)$ と表すとき，時刻 $t+1$ における信念は，時刻 t における信念 $\pi_i(t)$ と $P^a_{i,j}$，$b_j(x)$ を用いて以下の式（5.2）により求められる．

$$\pi_j(t+1) = k \cdot b_j(x) \sum_i P^a_{i,j} \pi_i(t) \tag{5.2}$$

式（5.2）において， k は $\vec{\pi}(t+1)$ の個々のベクトルの和を 1 にするための正規化定数である．

信念が求められれば，POMDP 環境下での問題は MDP 環境下と同様に解決することができる．ここで，状態と行動の組に対する期待報酬の見積りである Q 値を $Q(\vec{\pi}, a)$ と表現し，信念状態空間上の各状態 i における行動a に対する期待報酬を V 値と呼び， $V[i,a]$ と表す．$Q(\vec{\pi}, a)$ は以下の式（5.3）で近似される．

$$Q(\vec{\pi}, a) \approx \sum_{i=1}^{n} \pi_i V[i,a] \tag{5.3}$$

また，Q 値の学習は V 値を強化するだけで可能となる．V 値の強化はQ 学習における強化の規則に，それぞれの状態の確率分布である信念を考慮することによって，以下の式（5.4）に示す規則に従う．

$$V[i,a] = (1 - \alpha\pi_i(t))V[i,a] + \alpha\pi_i(t)(r + \gamma \max_{c \in A} Q(\vec{\pi}(t+1), c)) \tag{5.4}$$

ここで，r は行動者が受けとる報酬を表し， α $(0 < \alpha \leq 1)$ は学習率，γ $(0 \leq \gamma \leq 1)$ は割引率と呼ばれ，割引率は将来の報酬に対する評価を決める定数である．

信念を用いて内部に環境のモデルを生成するこれらの手法はモデルベース手法と呼ばれ，POMDP 環境下の学習においてその有効性が示されている．しかしながら，環境のモデルを生成するためには，すべての経験を記憶しておく必要があり，状態空間が広大かつ連続である実環境では，経験を積むにしたがって環境のモデルが拡張を続けるため，必要な記憶領域は学習時間に比例して増加する．このため，膨大な状態数を持つ実環境では，報酬の伝播が終了せず，学習が収束しない可能性がある．

以上より，本研究で対象とする環境の特性をまとめると，以下のようになる．

- 環境が部分的にしか観測できず，観測に不確実性が存在する．
- 環境が連続かつ広大であり，すべての経験を保持しておくのは現実的ではない．

第 5.5 節で説明する行動モデルは，状態空間を自律的に分割し，各状態に対する情報を領域を対象とした情報と見なして，これらの問題の解決を目指すものである．

5.5 行動モデルの生成と拡張

5.5.1 行動モデルの意義

実環境の状態空間において，不完全性や不確実性を含む観測状態すべてに対して行動のルールを保持することは，実ロボットを扱う上での収束性を損う可能性がある．この問題を解消するために，EBRL のように状態空間を分割し，領域ごとに遷移ルールを割り当てることが考えられる．しかしながら， EBRL のような環境に対する先見的知識が必要な手法は実環境に対して不向きである．また，システムの設計者が事前に状態空間を分割する手法では，分割がロボットにとって最適である保証がなく，状態空間が不適切に分割されてしまうと，ロボットは最適な行動を獲得できない．

これに対し，ロボットの遷移履歴から，概念学習により一般的な遷移規則を生成することを目指したのが**行動モデル**である．情報を汎化すれば，広大で連続な空間における点としての情報を，論理空間における領域の情報として扱うことができる．このことは，見かけ上の距離や方向などの物理空間と行動が密接に関連していない場合，ロボットの立場から見た空間を構成して，作業領域を分割することができる．また，汎化された情報は自らの経験によって得られたものであり，環境に対する先見的知識を必要としないため，よりロボット自身の行動に密接に関連したモデルの生成が期待できる．

以上のような考えに基づき，Christiansen[55] は実ロボットを対象とした不確実な環境に対する行動モデルの生成手法を提案した．ここでいう不確実性とは，行動後の状態が一意に定まらないという意味であり，これら行動後の状態を集合として扱い，確率的な遷移モデルを生成して解決している．行動前の状態についても同様に集合として扱っているが，モデル生成の過程で複数の状態集合が重複する問題について考慮していない．このことは，ロボットが現在属している位置を特定できないことになり，次の行動を決定できない．

この問題は，ある状態が属する複数の状態集合に対し，行動後の評価値を見積もる関数を設定し，学習の進行に沿って誤った状態集合を淘汰していけばよい．状態集合を生成する過

程で完全に重複を避けることは困難であるため，行動後の状態への遷移確率をもとに，各行動の評価値を見積もることが可能となる．このような考えに基づき，本節では Christiansen のモデル化手法をもとに，状態集合評価値を見積もり，更新する拡張を強化学習を用いて行う．

5.5.2 行動モデルの生成

行動モデルの構成

行動モデルは，連続的な状態・行動空間のためのオペレータの一種である funnel オペレータ [55] と同様の記述形式をもって表現される．行動モデルの保持するルールを $[M, S]$ と記述する．ここで，M，S はそれぞれ mouth，spout と呼ばれ，以下の意味を持つ．

$$\forall [x, a] \in M \quad x' \in S \tag{5.5}$$

式（5.5）において，$[x, a]$ はある状態 x と x において選択された行動 a の組であり，状態 x' は $[x, a]$ の結果観測された状態の観測値である．以降，この遷移を $[x, a, x']$ と記述する．spout S は遷移後の状態の集合を，mouth M は spout S からの後向き予測により獲得される状態集合と行動の組を示す．つまり，行動モデルとは，ある行動遷移に関する行動前の状態集合（mouth）から行動後の状態集合（spout）への写像ととらえることができる．行動モデルを視覚化したものを図 5.1に示す．

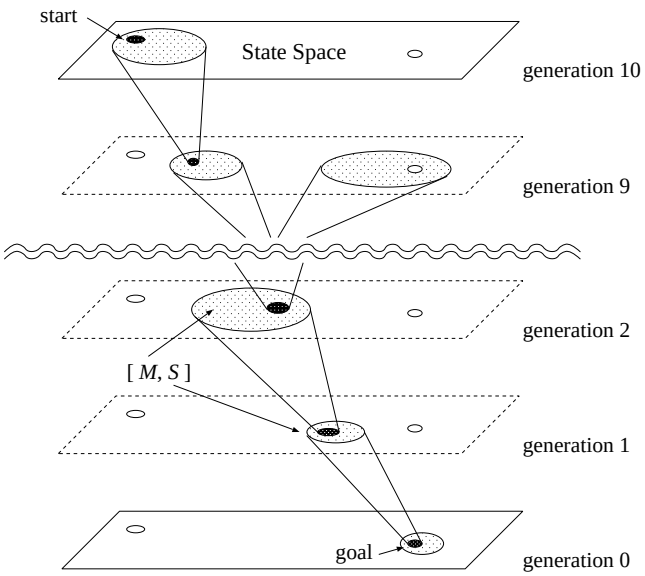


図 5.1: 行動モデル

行動モデルの各ルールは，$[M, S]$ 以外にいくつかのパラメータを保持する．**遷移確率パラメータ**は，そのルールの mouth に $[x, a]$ が包括される訓練事例 $[x, a, x']$ のうち，同ルー

ルの spout に状態 x' が包括される確率を表す．つまり，遷移確率パラメータは，あるルールが訓練事例と照らし合わせてどの程度信用できるかを示す指標となる．**generation** は図 5.1 で示されるように，ゴール状態を generation 0 として，後向き予測によって獲得された順にラベル付けされたルールの生成順位である．つまり，generation が若いほど，ゴール状態に近い領域で生成されたルールであるといえるので，generation はゴールへの遷移系列を求める上での指標となりうる．

行動モデルの生成手法

行動モデルの生成は後向き予測によって行われるため，まず遷移後の状態集合である spout が選択され，次に mouth が選択されるという過程を繰り返す．

1. spout の選択

行動モデルの spout は，ロボットによって観測される状態空間の部分集合で表される．実環境における状態空間は広大かつ連続であるため，行動による観測値から考えられるすべての状態集合を spout とすることは困難である．よって実環境で行われる行動獲得において有用な行動モデルを生成するためには，有用な spout のみを限定して選択することが必要となる．

また，行動モデルを拡張するために用いられる強化学習において，ロボットアームの先端がゴール状態へ到達して与えられる報酬は，ゴール状態から状態遷移の履歴を遡って伝播される．このため目標状態と関係のない spout を選択することは，行動モデルのルール列のなかにタスクの達成によって得られる報酬が伝播しないルールを生成してしまう可能性を生む．

これら二つの問題を解決する spout の選択方法として，本手法では，最初の spout にタスクの目標状態を選択し，帰納推論により獲得された mouth 領域のみを次の spout 候補とするという過程を繰り返す方法を採用する．この方法により，無数に存在する spout の候補の中から，強化学習で報酬の伝播が可能であるもののみを選択することが可能である．

2. mouth の算出

spout S が選択されると，mouth M の算出は C4.5 [16] を用いた概念学習問題となる．観測された状態遷移の集合を T と表すと，spout S に対して，観測値と行動の組 $[x, a]$ は以下のように定義される正例 $pluses(S)$ と負例 $minuses(S)$ にラベル付けされる．

$$\begin{aligned} pluses(S) &= \{[x,a] \mid [x,a,x'] \in T \wedge x' \in S\} \\ minuses(S) &= \{[x,a] \mid [x,a,x'] \in T \wedge x' \notin S\} \end{aligned} \tag{5.6}$$

式（5.6）によって正例または負例にラベル付けされた x を，訓練事例として C4.5 に与える．C4.5 は，与えられた訓練事例を用いて決定木を構築し，正例および負例の一般的な概念を導出する．構築された決定木において，正例にラベル付けされた葉に分類される観測値の一般的な概念を獲得することは，spout S に遷移可能な状態空間の部分集合 mouth M の論理規則を獲得することにあたる．また，獲得された概念は次のステップにおける spout の候補となる．

行動モデルの生成は，新しい spout の候補が獲得されなくなるまで続けられるべきであるが，実環境において状態を表す観測値は無数に存在し，かつ不完全，不確実であることから spout の探索が収束することは困難である．また，問題の設定者が行動モデルの持つルールの数を設定するなどの方法によって探索を一意的に終了させてしまうと，生成される行動モデルに偏りが生じたり，ルール数が冗長でありすぎたりする可能性が生じる．本手法では，パラメータ generation を用いてこの問題に対応した．generation にしきい値を設けることにより，行動モデルの生成手法に適応した意味ある spout の探索終了を行うことができる．

5.5.3 行動モデルの拡張

行動モデルの問題点

実環境においては，状態の観測値に不完全性や不確実性が存在し，行動モデルを生成する際に，新しい generation のルールの探索が既存のルールの状態空間を考慮しないため，行動モデルにおける異なるルールが同じ状態空間を共有してしまうという問題が生じる．図 5.2 は，異なるルール Rule1 と Rule2 の mouth が状態空間の同じ領域を共有している例である．

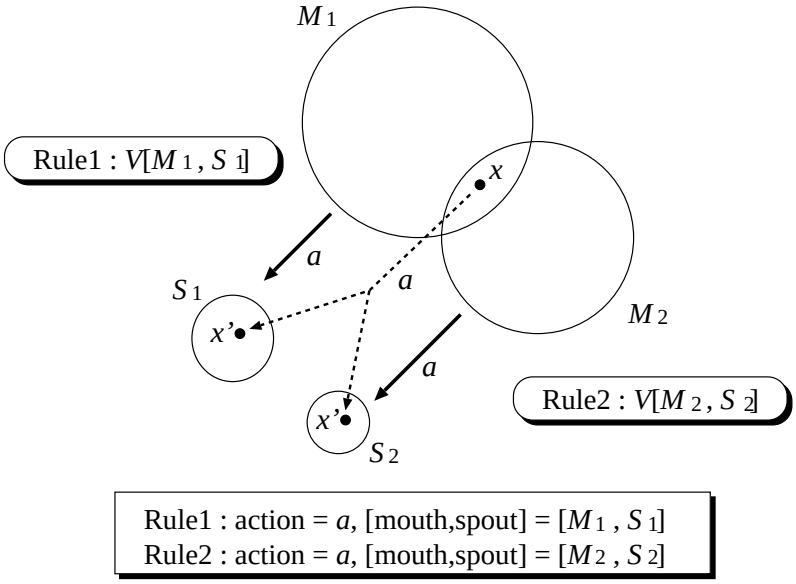


図 5.2: 観測値 x と行動 a に対する Q 値

図 5.2 において，x，x' は遷移前，遷移後の観測値を示す．2 つのルールの generation が等しい場合，行動主体はどちらのルールに従って行動を決定すればよいか判断できない．仮に，いずれか一方のルールを選択したとしても，選択されたルールの遷移確率が 100% でなければ，行動は不正確となる．

本章では，行動モデルの各ルールに，新たなパラメータとして信念状態空間におけるルールの優先度を表す V 値を導入する．信念を用いた強化学習法で強化した V 値を，状態と行動に対する期待報酬の見積りである Q 値に反映する手法を用いてこれらの問題を克服し，より有効な行動を選択できることを目指す．

Q 値の算出

現在の観測値とある行動の組 $[x, a]$ を mouth M_i $(i = 1, 2, \ldots, n)$ に含む行動モデルの各ルールを $[M_1, S_1], [M_2, S_2], \ldots, [M_n, S_n]$ と表し，遷移確率パラメータをそれぞれ $P_1, P_2, \ldots, P_n$ と表すと，遷移後に観測される観測値が各 spout S_i に含まれる割合 ρ_i は以下の式で予測することができる．

$$\rho_i = \frac{P_i}{\sum_{j=1}^{n} P_j} \tag{5.7}$$

遷移後の観測値の各 spout S_i に対する確率分布は，ρ_i を各要素とするベクトルとして $\vec{\rho} = \langle \rho_1, \rho_2, \ldots, \rho_n \rangle$ と表される．ここで，ルールの V 値を $V[M_i, S_i]$，状態と行動の組 $[x, a]$ の Q 値を $Q(x, a)$ と記述すると，$Q(x, a)$ はベクトル $\vec{\rho}$ および $V[M_i, S_i]$ により，以下のように近似される．

$$Q(x, a) \approx \sum_{i=1}^{n} \rho_i \cdot V[M_i, S_i] \tag{5.8}$$

現在の観測値に対する各ルールの V 値を見積って Q 値を算出すれば，行動主体は最大の Q 値を持つ行動を選択することが可能となる．

Q 値の学習

本手法における Q 値は，行動モデルの各ルールが保持するパラメータの値を修正して学習される．以下では，状態遷移 $[x, a, x']$ が行われ，報酬 r を得た場合を考える．

遷移後の観測値 x' を spout S_i $(i = 1, 2, \ldots, n)$ に含み，かつ遷移前の観測値と行動の組 $[x, a]$ を mouth M_i に含む行動モデルの各ルール $[M_i, S_i]$ に対して，実際に $[x, a]$ が mouth M_i に含まれる確率 ρ_i' は，それぞれの遷移確率パラメータ P_i を用いて，以下のように求めることができる．

$$\rho_i' = \frac{P_i}{\sum_{j=1}^{n} P_j} \tag{5.9}$$

式（5.9）により，$[x, a]$ の各 mouth M_i に対する確率分布 $\vec{\rho_i'}$ が求まれば，行動モデルの各ルールの $V[M_i, S_i]$ は，以下の規則によって更新される．

$$V[M_i, S_i] = V[M_i, S_i] + \alpha \rho_i' (r + \gamma \max_b Q(x', b) - V[M_i, S_i]) \tag{5.10}$$

観測値 x' はspout S_i に含まれる値であるから，$V[M_i, S_i]$ は spout S_i における期待報酬を一段階割り引いた値となる．また，各ルールが保持する遷移確率パラメータは，過去に経験した遷移事例 $[x, a, x']$ の集合より，ルールの mouth に $[x, a]$ が含まれる訓練事例数に対

する，ルールの mouth に $[x, a]$ が含まれ，かつルールの spout に x' が含まれる訓練事例数の割合として計算できる．

Q 値の学習は，状態が遷移するたびに更新規則を適用してなされるが，本手法では reverse trajectory 法 [50] [56] に従って，式（5.10）を適用して Q 値が更新される．reverse trajectory 法では，問題を解いている間は Q 値の更新を行わず，代わりに観測された状態遷移の履歴を記憶しておく．目標を達成すると，最後に観測された遷移から履歴を後向きにたどりながら更新規則を適用していく．

5.5.4 モデル信頼度を用いた変動する環境への対応

モデル信頼度の意義

行動モデルは，環境が静的であるという仮定に基づいて，実環境における広大で連続な状態空間と環境入力の不確実性，不完全性を克服することを目的としている．しかしながら，行動モデルの生成・拡張は静的な環境で行われるため，ロボットの行動とは独立して変動する環境へ対応することができない．また，行動モデルの生成や拡張には時間を要するため，環境が変化する度に新しいモデルを作り直すことは，ロボットの行動の即応性を保持するために好ましくない．

本章では，行動モデルが変動しうる各環境それぞれの状態空間のモデルであることに注目し，ある遷移行動における状態と行動の組 $[x, a]$ と，その行動の結果として得られる状態 x' を既存のモデルと参照して，現在の環境において選択されるべき行動を決定する手法を提案する．ここで，あるモデルをどの程度信頼して行動決定に反映するかを表すパラメータを**モデル信頼度**と呼ぶ．モデル信頼度を更新すれば，環境の変動に順応した期待報酬を獲得できると考えられる．

モデル信頼度に基づく行動の決定

本手法の枠組みを 図 5.3 に示す．図 5.3 において，3 つの Action Model は，それぞれ第 5.6 節における 3 つの光源位置 A，B，C に対する行動モデルであり，それぞれ M_A，M_B，M_C と表す．

ステップ n において状態の観測値 x_n が与えられたとき，M_A，M_B，M_C は，各自が生成された環境と同じ位置に光源があるものとして，次の遷移で実行されるべき有効な行動を選択するために，状態と行動の組 $[x_n, a]$ の期待報酬を見積る．それぞれのモデルが見積る期待報酬を $r_A(n)$，$r_B(n)$，$r_C(n)$ と表すと，各行動モデルは式（5.11）のような関数の形で表現できる．

$$\begin{aligned} r_A(n) &= f_A(x_n) \\ r_B(n) &= f_B(x_n) \\ r_C(n) &= f_C(x_n) \end{aligned} \tag{5.11}$$

ここで，$r_A(n)$，$r_B(n)$，$r_C(n)$ を入力とし，現在の環境における期待報酬を見積る関数を

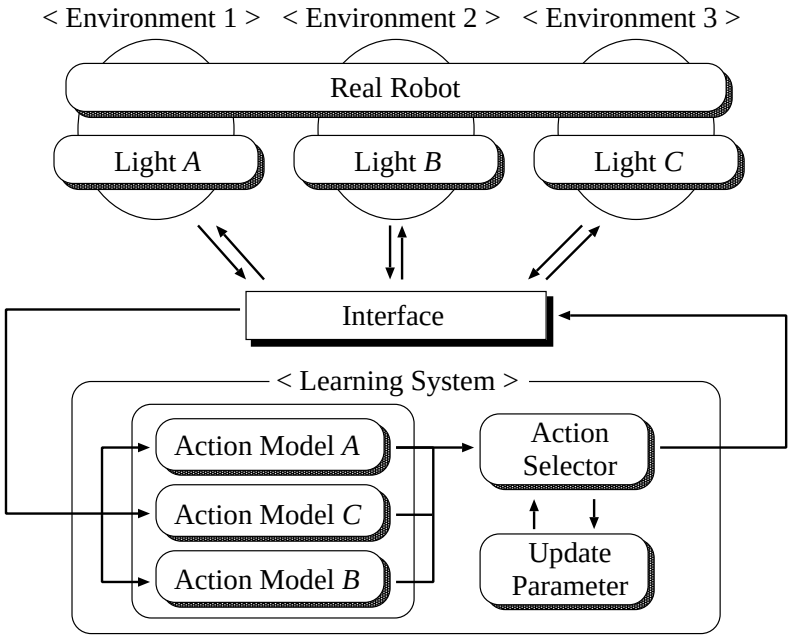


図 5.3: システム概要

以下のように設定する．

$$R_n \leftarrow F_n(r_A(n), r_B(n), r_C(n)) \tag{5.12}$$

つまり，式（5.12）は静的環境で生成された各モデル M_A，M_B，M_C に対し，どのモデルの期待報酬が現在の環境において信頼し得るかを表す関数である．本章では，次のような線形関数を用いる．

$$F_n(r_A(n), r_B(n), r_C(n)) = w_A r_A(n) + w_A r_B(n) + w_C r_C(n) \tag{5.13}$$

ここで w_A，w_B，w_C は，それぞれ M_A，M_B，M_C に対するモデル信頼度である．

変動する環境下における経験によってモデル信頼度を常に更新し続けることにより，システムから環境に対する情報を明示的に与えられることなく，現在の環境に順応した行動を獲得することができる．

モデル信頼度の強化

モデル信頼度の更新規則について説明する．モデル信頼度を更新するための有力な方法として，ある遷移における遷移前の状態，行動，遷移後の状態の組 $[x, a, x']$ を各モデルと参照する方法が考えられる．

ある試行のステップ数が N 回で，そのうち各モデル M_A，M_B，M_C の保持するルールに包括される遷移行動がそれぞれ k_A，k_B，k_C 回あったとする．これらのパラメータは，

現在の環境で各モデルがどの程度参照されたかを示す指標となる．本手法では，モデル M のモデル信頼度 w_M を次のように更新する．

$$w_M \leftarrow w_M(1 + \frac{1}{N}\frac{k_M}{k_A + k_B + k_C}) \tag{5.14}$$

また，モデル M に対して k_M が 0 であった場合，その試行における環境でモデル M は全く参照されなかったことになるので，モデル M に対応するモデル信頼度 w_M には式（5.15）によって罰を与える．

$$w_M \leftarrow \lambda w_M \tag{5.15}$$

ここで，λ は 1 未満の正定数である．

1 回の試行終了ごとに各モデル信頼度を更新し，現在の環境に対応した行動を獲得できる．また，特定の環境に対してモデル信頼度の学習がある程度進んだ後に，ある試行を境に環境が変化した場合でも，新しい環境に順応したモデル信頼度の学習が期待される．

5.6 実験

5.6.1 実験環境

行動モデルを生成・拡張する過程において，環境は静的であると仮定する．つまり，ロボットの環境における状態は，自身の行動の影響によってのみ変化する．また，競合する他のロボットなどは存在せず，目標は固定されているものとする．

図 5.4 に本実験で使用したロボットアームの概観を示す．ロボットアームとして三菱電機社製の MOVEMASTER EX [57] を使用している．アームの先端には 3 方向に向けて光センサが取り付けてあり，感知した光の強さによりそれぞれ 0 ～ 255（光が強いほど大きい値をとる）の整数値を出力する．

ロボットが環境から得られる情報は，光センサの出力値のみである．光センサの出力値はノイズを含むため，観測には不確実性が生じる．また，ロボットには関節の位置や関節角の情報を与えていない．よって，ロボット自身はアームの先端の正確な位置は認識できず，観測には不完全性が残る．図 5.5 に実験環境を示す．実験環境には光源 A，B，C が存在し，それぞれがロボットアームに対するゴール領域を持っている．実験には，5 種類の関節のうちショルダ，エルボ，リストの 3 種類の関節のみを使用するため，環境は 2 次元的となる．それぞれの関節が回転する角度の範囲は，ショルダが水平線に対して $0^\circ \sim 90^\circ$，エルボがショルダに対して右回りに $5^\circ \sim 105^\circ$，リストがエルボに対して右回りに $-30^\circ \sim 80^\circ$ である．ロボットアームの動作範囲は，各関節の回転角度の範囲内で地面に衝突しない領域である．各関節は 10° を最小単位として回転し，1 種類の関節を右回りまたは左回りに 10° 回転させる 6 種類の動作が存在する．また，アームの動作範囲を越える行動は選択しない．

本章では，ロボットの 6 種類の最小動作を**行動**，ロボットが自身のセンサを用いて状態を観測し，観測値に対して行動を選択し実行する遷移過程を**ステップ**，ステップを繰り返して初期位置からタスクを達成するまでの一連の動作を**試行**と呼ぶ．

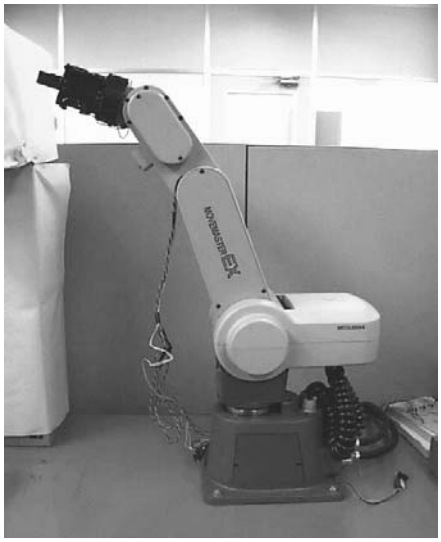


図 5.4: ロボットの概観

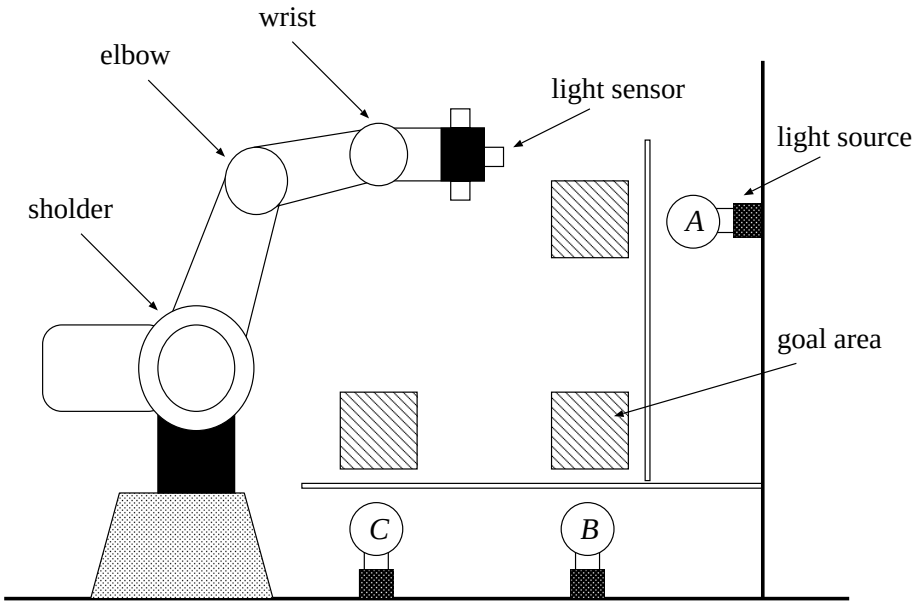


図 5.5: 実験環境での状態空間

5.6.2 行動モデルの生成と拡張：光源到達タスク

光源到達タスクの設定

まず，行動モデル生成のための訓練データ収集を目的として，ロボットに各光源ごとに約 6000 回の行動を実行させた．この際，ロボットの 6 種類の行動はランダムに選択され，行動選択の指標は一切与えられないものとする．得られた訓練データを入力とし，C4.5 を用いて行動モデルを生成した．ただし，mouth の候補は generation 10 までを限定して探索する．

次に，行動モデル拡張の学習問題として，図 5.5 に示す環境において，各光源ごとにより少ないステップ数でアーム先端を光源付近のゴール領域に到達させるタスクを設定し，これを**光源到達タスク**と呼ぶ．ここで，アーム先端が光源に到達したかどうかは，アーム先端の座標が各光源ごとに定められたゴール領域内に入れば判断できるものとする．ただし，アーム先端の座標の情報は，光源への到達を外部から判断するためのみに用いられ，学習のための入力としては用いられない．

ロボットは定められた 5 つの初期位置から環境の観測と行動を繰り返し，ゴール領域を目指して状態空間を探索する試行を行う．各ステップにおける行動の選択は，行動の組 $[x, a]$ に対する Q 値を見積ることによって行われる．ただし，新規の遷移列を探索するために，一定の確率でランダムな行動が選択される．光源に到達するかステップ数が 50 を越えるとその試行を終了し，次の初期位置に配置される．

光源に到達した試行には，報酬として基本の報酬 150 からその回の試行のステップ数を引いた値を与え，報酬の伝播を行う．50 ステップ以内に到達しない試行では，報酬の伝播は行われず，その試行で実行された各遷移に対応するルールに罰を与える．失敗試行はステップ数を 100 としてカウントする．また 5 つの初期位置が一巡する 5 試行ごとに，全く強化されなかったルールに対して罰（税金 [58]）を与える．行動モデルが状態遷移を含むルールを持たない場合は新たなルールを生成するが，観測値が連続的であるため，ある程度離散値化した観測値を用いて生成している．また，学習率を $\alpha = 0.6$，割引率を $\gamma = 0.9$，ランダムな行動を選択する確率を 20% に設定している．

本実験における行動モデル拡張のための強化学習手法をまとめると，以下のようになる．

1. 行動モデルの各ルールの V 値を，ルールの generation に対応した値で初期化し，ロボットを初期状態にセットする．

2. 現在の観測値 x に対するそれぞれの行動の Q 値を式（5.8）によって計算し，一番高い Q 値を持つ行動を選択し実行する．ただし，計算した Q 値の結果が 0 の場合，または一定の確率でランダムに行動を選択する．

3. 観測値 x，選択された行動 a，行動後の観測値 x' および行動後に得た報酬 r の組 $[x, a, x', r]$ を LIFO 列に記憶する．

4. 遷移確率パラメータを更新する．

5. 目標を達成するまで 2 〜 4 を繰り返す．

6. 目標の達成後，LIFO 列から $[x, a, x', r]$ を一つ取り出し，x' を spout に含み，かつ $[x, a]$ を mouth に含むルールの V 値を式（5.10）によって更新する．状態遷移を含むルールがなければ，新たにルールを生成する．

7. LIFO 列が空になるまで（6）を繰り返す．

8. ロボットを初期状態にセットして（2）に戻る．

結果と考察

実験は，150 回の試行を 1 回の実験とし，各光源位置ごとに 2 回ずつの実験を行った．以後のデータはすべて 2 回の実験の結果の平均を示している．

図 5.6 に各光源位置ごとの学習の進行度を示す．実線は強化学習で拡張した行動モデルを，破線は拡張を行わない行動モデルのステップ数の推移を表している．図 5.6 において，縦軸は 1 試行に要したステップ数，横軸は試行回数を表す．3 つの光源それぞれに対して，試行が進むにしたがってモデルが強化され，試行回数が 100 回を越えたあたりで 1 試行あたりにかかるステップ数が 15 回前後に収束していることがわかる．

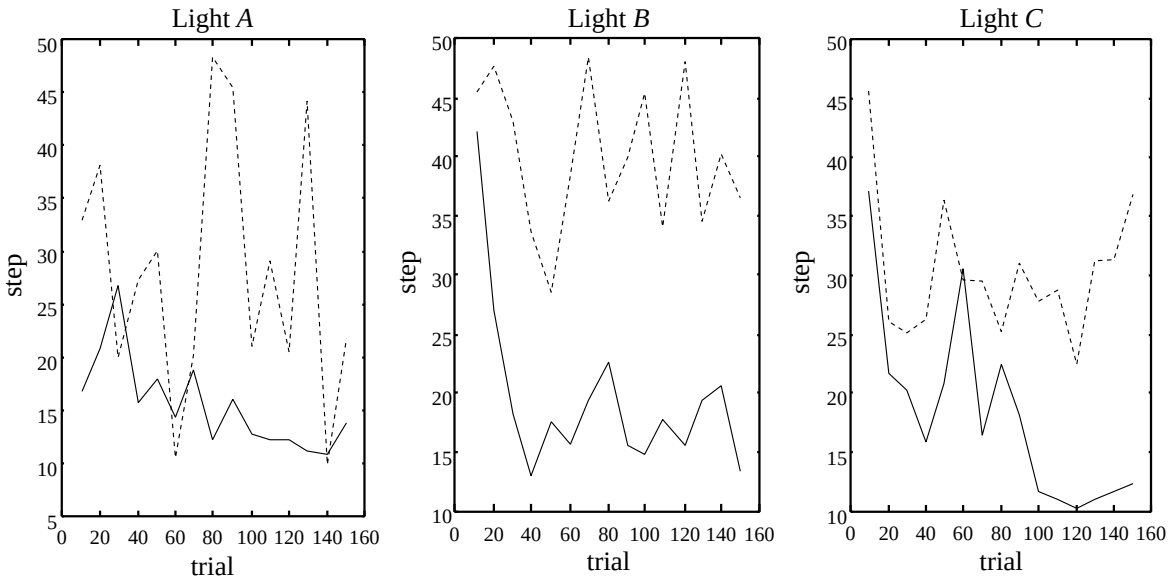


図 5.6: ステップ数の遷移

また，図 5.6 おいて，拡張を行わない行動モデルは，V 値の初期値として与えられた generation の指標のみによって行動を選択するため，局所解に陥ることが多く，効率のよい政策が得られていない．この結果から，強化学習で行動モデルを拡張すれば，試行回数の増加につれ適切な行動を選択できることがわかる．

5.6.3 モデル信頼度を用いた実験：光源変更タスク

光源変更タスクの設定

本実験において，ロボットは実験開始時に定められた初期位置に配置され，その後設計者によって変更される光源に対してタスクを切り替えるために，各モデル信頼度の学習を行う．これを**光源変更タスク**と呼ぶ．ロボットの設定や各光源，ゴール領域の位置は光源到達タスクと同じである．実験環境を図5.7に示す．

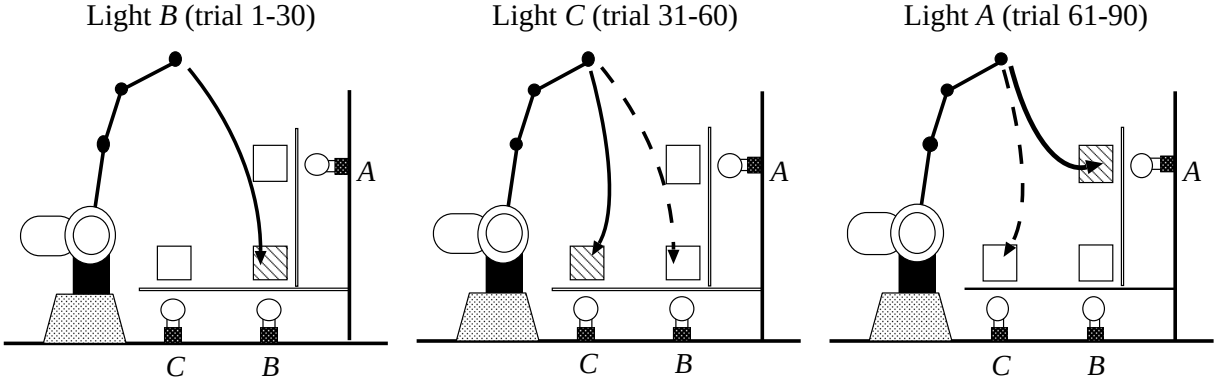


図 5.7: 実験環境

本実験では，モデル信頼度の強化による独立した結果を示すために，各試行においては，第 5.6.2 項で行われた行動モデルのルールに対する強化は行わない．光源変更タスクの実験の流れを以下に示す．

1. 光源 B を点灯し，初期位置から光源 B に対応するゴール領域までの行動に対して，モデル信頼度の学習を行う．ここでの試行回数は 30 回とする．

2. 光源 B を消灯し，継続して光源 C を点灯させ，30 回の試行でモデル信頼度の学習を行う．

3. 光源 C を消灯し，継続して光源 A を点灯させ，30 回の試行でモデル信頼度の学習を行う．

各ステップにおいて，ロボットの行動は式（5.13）に示した期待報酬によって決定される．モデル信頼度が現在の環境に合わせて強化され，図 5.7 の実線で表したように，現在の光源位置を直接目指す行動が選択できることを学習の目的とする．各試行は，現在のゴール領域にロボットアームの先端が到達するか，試行のステップ数が 50 回 を越えた時点で終了し，初期位置に配置される．

実験結果

実験における学習の進度状況を 図 5.8 に，モデル信頼度学習の進度状況を 図 5.9 に示す．

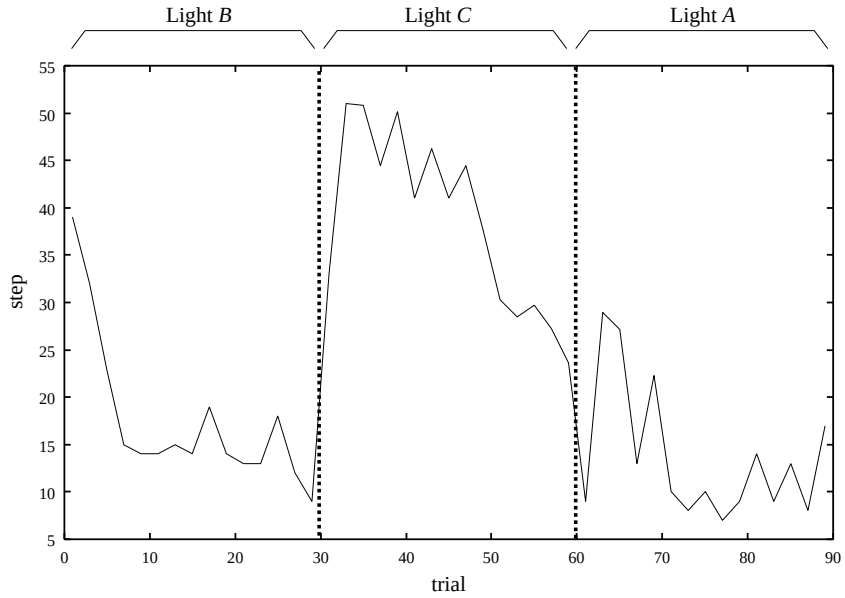


図 5.8: 光源変更タスクにおけるステップ数の遷移

図 5.8 において，横軸は試行回数，縦軸はその試行においてゴール到達までに要したステップ数である．グラフ中に点線で示す 30 回目と 60 回目の試行の後で，光源位置が切り替わっている．各光源位置ごとの試行において，初めは各モデルの期待報酬が競合するために正しい期待報酬を見積ることができず，ゴール領域に到達するまでに多くのステップ数を必要とするが，モデル信頼度の学習が進めば，現在の環境に対応したモデルを重視して行動を選択できることがわかる．

図 5.9 において，横軸は試行回数，縦軸はその試行における各モデル信頼度の値である．現在の光源位置に対応するモデルのモデル信頼度が強化され，それ以外のモデル信頼度はほぼ減少していることがわかる．この結果より，モデル信頼度は，ある光源に順応して強化された後で環境に変化があった場合においても，新しい環境に順応できることが示された．

5.6.4 モデル信頼度を用いた実験：光源追従タスク

光源追従タスクの設定

光源変更タスクでは，いくつかの試行後に光源位置が変更された場合に，モデルに対する重みを変更してタスクを達成していた．この場合，充分な試行の後に重みの変更を行うため，切り替わりに対しての反応に充分な試行を要した．重みの変更幅の小さい場合，す

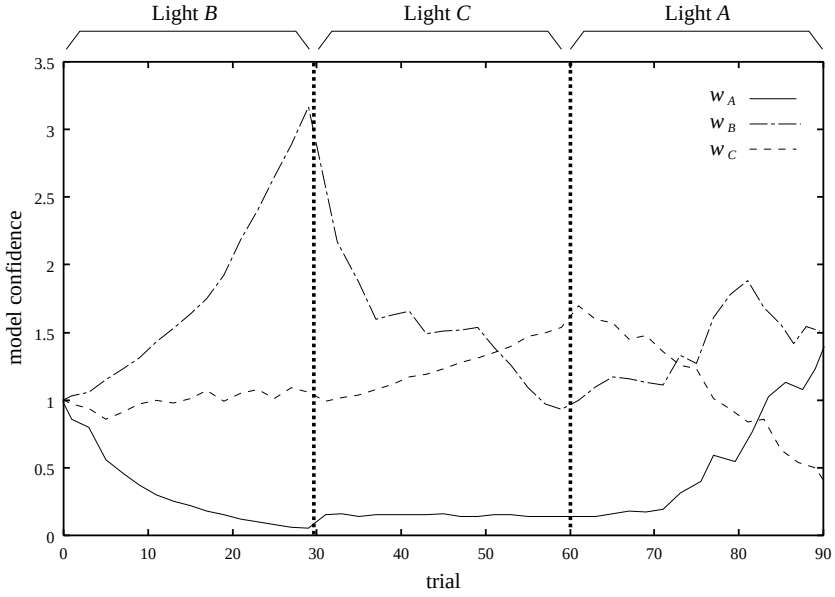


図 5.9: 光源変更タスクにおけるモデル信頼度の遷移

なわち試行途中で光源の位置が切り替わった場合には，光源変更タスクよりも迅速な順応が期待できる．これを**光源追従タスク**と呼ぶ．ロボットの設定や各光源，ゴール領域の位置は光源到達タスクと同じである．実験環境は図 5.7に示す環境と同じで，光源の切り替えのタイミングを変更している．

光源追従タスクでは，以下の流れにしたがって光源を変更する．

1. 光源 B を点灯し，初期位置から光源 B に対応するゴール領域までの行動に対して，モデル信頼度の学習を行う．

2. 10 ステップ経過後，光源 B を消灯し，継続して光源 C を点灯させ，モデル信頼度の学習を行う．

3. 10 ステップ経過後，光源 C を消灯し，継続して光源 A を点灯させ，モデル信頼度の学習を行う．

各ステップにおいて，ロボットの行動は式（5.13）に示した期待報酬によって決定される．モデル信頼度が現在の環境に合わせて強化され，図 5.7 の実線で表したように，現在の光源位置を直接目指す行動が選択できることを学習の目的とする．各試行は，現在のゴール領域にロボットアームの先端が到達するか，試行のステップ数が 50 回 を越えた時点で終了し，初期位置に配置される．

実験結果

実験における学習の進度状況を図 5.10 に，モデル信頼度学習の進度状況を図 5.11 に示す.

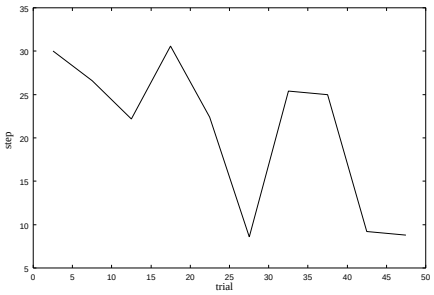


図 5.10: 光源追従タスクにおけるステップ数の遷移

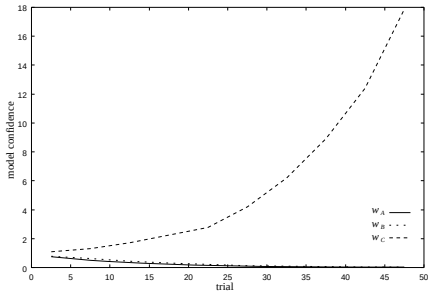


図 5.11: 光源追従タスクにおけるモデル信頼度の遷移

図 5.10 において，横軸は試行回数，縦軸はその試行においてゴール到達までに要したステップ数である．全体として，なだらかな収束傾向にあり，ゴールまでにかかったステップ数が減少している．学習初期には到達に失敗した試行も多く，平均すると上限の 50 ステップに近い値をとっている．学習中期より，到達時のステップ数が 10 ステップ以下となる場合も多く見られ，光源 B が点灯している間に到達が可能となっていることが分かる．光源変更タスクの場合と同様，初めは各モデルの期待報酬が競合するために正しい期待報酬を見積ることができず，ゴール領域に到達するまでに多くのステップ数を必要とするが，学習が進めば光源が切り替わる前にタスクを達成することができる．

図 5.11 において，横軸は試行回数，縦軸はその試行における各モデル信頼度の値である．学習初期より，光源 C に対するモデル信頼度が増加し，他の二者は減少傾向にある．実際の到達目標は，学習後期にはほとんどが光源 B または光源 C であり，光源 A に到達することは稀であった．このことより，光源 A と光源 B および C は，その到達経路が異

なり，且つ最初に到達した光源 C に対するモデルが強化されたことで，光源 B に対する経路も同時に強化されたことになる．

考察

図 5.8 において，光源位置によってステップ数の減少の傾向が異なるのは，各モデルの強化状況や，各ゴール領域間の軌道による学習の難易度の違いが原因であると考えられる．特に光源 C が点灯している区間では，他の 2 区間と比べてグラフの波形が大きく異なっている．これは，図5.12 の case 1 で示すように，光源 C に対する行動モデルの強化において，報酬の伝播によって重点的に強化されたロボットの遷移軌道は，初期位置から光源 C を直接目指すもの（破線）であり，光源 B のゴール領域付近からの遷移軌道（実線）が十分に学習されておらず，モデル学習の進行が不完全であったためであると考えられる．しかしながら図5.8 において，試行回数が 50 回を過ぎたあたりから徐々にモデル信頼度に対する学習の効果が現れはじめ，試行回数が 60 回 に近づくにつれてステップ数が減少していることが確認できる．この現象の裏付けとして，図 5.9 において，試行回数が 50 回を過ぎたあたりで，w_C が w_B を逆転していることがあげられる．

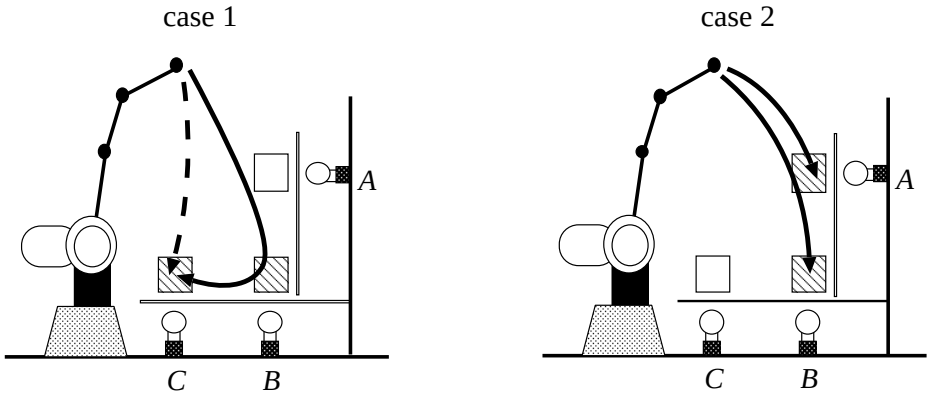


図 5.12: 光源変更タスクにおける問題

モデル信頼度は，試行において実行された遷移が，各モデルの保持する遷移ルールに包括される場合に強化されるため，類似した遷移が起こりうるモデル間では，モデル信頼度の学習が正しく行われない可能性がある．このことは，図5.9 における光源 A の点灯区間において， w_A 以外に w_B も増加していることが顕著な例として現れている．実際に，図 5.12 の case 2 で示すように，初期位置から光源 A および B に遷移した場合，同じ行動を選択したときの観測値の変化が類似することが起こりやすい．ただし，この区間において，図 5.8 のステップ数が試行を重ねるにつれて減少していることから， w_B は光源 A が点灯している環境におけるロボットの行動獲得に対して障害にはならず，モデル間の協調がなされているといえる．

5.7 結言

本章では，変動する実環境において，実時間性を保持しながらロボットが自律的に行動を獲得することを目標とし，概念学習と強化学習を用いて生成・拡張された行動モデルと，各行動モデルのモデル信頼度を利用する手法を用いて，その実現を試みた．

実環境における実験では，ロボットアームが移動する光源を追従する問題を想定し，静的な環境で生成された行動モデルに対応するモデル信頼度を強化して，変動する環境に順応した行動選択が可能であることを示した．実験により得られた結果は以下の 2 点である．

- 未知の実環境において，ロボットがタスクを遂行するための行動モデルを概念学習を用いて自律的に獲得できることを示した．また，強化学習を用いた拡張により，行動モデルが効率化された．

- モデル信頼度を導入し，変動する環境において光源を追従することができた．また，ステップ数の推移とモデル信頼度の推移から，本手法によってモデル信頼度が環境に順応していることを示した．

本手法においては，モデルを構築する概念学習アルゴリズムは静的な環境を対象としており，ゴールの数と同じ数のモデルを設定している．今後の課題としては，生成されたモデルの利用性を高めるために，タスクの細分化やサブゴールの設定などがあげられる．

第 6 章

結論

本論文では，複雑化する情報化社会において氾濫する情報の中から，機械学習を用いて適切な特徴を抽出し，固有の特徴空間を構成して，認識や分類を行うための方法論を提案した．本研究では，対象とする実験領域として，人工的な離散空間，周波数帯域ごとの画像空間，および連続的な実空間を採用した．いずれの対象領域においても，適切な特徴抽出手法が存在し，抽出された特徴を用いた学習が可能であることを示した．

第 2 章では，本研究に関連する機械学習の主要な学習手法と，それら学習手法の概要について説明した．また，機械学習における特徴の重要性を指摘し，その利用法について述べた．

第 3 章では，離散環境における特徴の抽出手法と，それらの特徴を強化学習に導入するための方法論について述べた．本章では，FCQL を人工的な迷路問題に適用し，報酬の推移についてシミュレーションを行った．環境の例として取り上げた迷路問題は，ロボット学習の範疇で用いられる代表的な問題の一つである．しかしながら，実際のロボットを実験に用いる場合には，センサからの情報も連続値であり，FCQL の枠組に組み込むには適切な方法で離散化する必要がある．本章での迷路問題はすでに離散化された状態にあり，一般的な実数値の連続空間を対象とする問題に対しては，適切な離散化を行う操作子の適用によって対応が可能であると考えられる．また FCQL では，特徴を構成する手法として論理演算を採用したが，他にも特徴を構成する手法に発見的な手法の採用が考えられ，この点についても検討の余地がある．以上，第 3 章の結果より，離散環境において特徴構成法を用いて適切な特徴を抽出し，それらの特徴が学習の効率化に貢献していることを示した．

第 4 章では，現実的な事例として静止画像を扱い，それらの画像から特徴を抽出して，適切な分類が行える学習手法について述べた．本章では，テクスチャ分類において特にその構造に着目し，構造の分布を表わす新たな指標としてテクスチャエントロピーを提案した．また，各テクスチャごとにテンプレートを作成する手法を提案し，様々な条件と構造化手法においてテンプレートを用いた分類精度の比較を行った．実験の結果，構造化されたテクスチャの分布が構造化アルゴリズムに強く依存していることを示し，定量的な評価を行った．分類においては，$\mathrm{TSWT}_{\mathrm{TD}}$ および PSWT に対する $\mathrm{TSWT}_{\mathrm{BU}}$ の優位性を示した．さらに，モーメントの導入による分類精度の向上と，変化に対するパフォーマンスの評価を行った．以上，第 4 章の結果より，画像の特徴の利用法を提案し，それらの特徴を

利用した適切な学習手法による分類が可能であることを示した．

第 5 章では，実環境における多関節型ロボットを対象とした，ロボットの作業のための空間構成法と構成された空間を用いた学習の方法論について述べた．変動する実環境を対象とし，実時間性を保持しながらロボットが自律的に行動を獲得することを目標とし，概念学習と強化学習を用いて生成・拡張された行動モデルと，各行動モデルのモデル信頼度を利用する手法を用いて，その実現を試みた．実環境における実験では，ロボットアームが移動する光源を追従する問題を想定し，静的な環境で生成された行動モデルに対応するモデル信頼度を強化して，変動する環境に順応した行動選択が可能であることを示した．実験により得られた結果は以下の 2 点である．まず第一に，未知の実環境において，ロボットがタスクを遂行するための行動モデルを概念学習を用いて自律的に獲得できることを示した．また，強化学習を用いた拡張により，行動モデルが効率化された．第二に，モデル信頼度を導入し，変動する環境において光源を追従することができた．また，ステップ数の推移とモデル信頼度の推移から，本手法によってモデル信頼度が環境に順応していることを示した．以上，第 5 章の結果より，実環境の一つであるロボットの作業空間において，特徴空間の構成手法を提案し，これらの特徴が学習に有効であることを示した．

以上，本論文では機械学習における特徴の重要性と，それらの特徴の抽出手法を提案した．また，抽出された特徴を利用して，適切な学習や分類が可能であることを示した．これらのことから，近年の複雑化する情報を機械学習を用いて適切に認識・分類することが可能であると言える．

謝辞

本研究を遂行するにあたり，その機会を与えてくださいました，神戸大学自然科学研究科　金田 悠紀夫教授に慎んで感謝の意を表すとともに，厚く御礼申し上げます．本研究の遂行ならびに論文の作成にあたり，終始懇切丁寧な御指導および御鞭撻を賜りました神戸大学都市安全研究センター　上原 邦昭教授に慎んで感謝を申し上げます．本論文の作成にあたり，有益な御助言と御教示を賜りました神戸大学工学部　田中 克己教授に感謝の意を表します．研究途上において常日頃懇切丁寧な御指導および御教示を賜りました神戸大学都市安全研究センター　角谷 和俊助教授に深く感謝致します．

本研究の遂行にあたり懇切丁寧な御指導および御助言を頂き，共同研究において熱意のこもった御討論を頂きました通信総合研究所関西先端研究センター　マハダド ヌリ シラジ博士には最大限の感謝の意を表します．本研究を進めていく上で常日頃より的確な御指導および御助言を頂き，研究生活において公私共に支えて頂きました摂南大学工学部　諏訪 晴彦講師には慎んで感謝の意を表します．

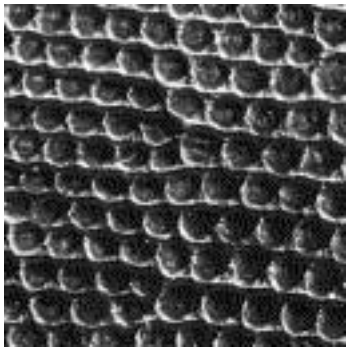
研究生活を進めていく上で常日頃より苦楽を共にした，神戸大学工学部　槙本 希美助手には改めて深く感謝致します．最後の一年間，研究生活においてよき時間を共に過ごした神戸大学経営学研究科　平松 治彦助手に感謝の意を表します．また，日頃より研究生活を共にした神戸大学工学部情報知能工学科上原研究室の皆様に感謝致します．

最後に，日頃より研究生活を支えてくれた家族，両親ならびに妻に感謝致します．

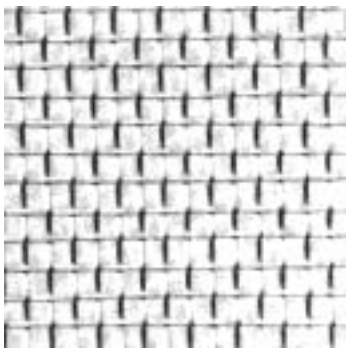
付録 A

画像データ

- D3: Reptile skin

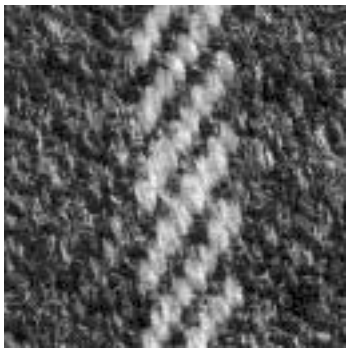


- D6: Woven aluminum wire

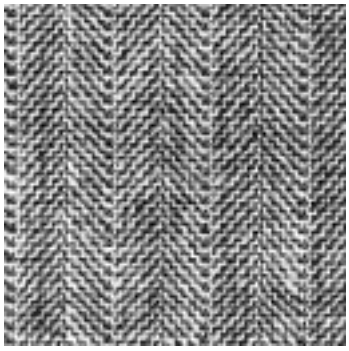


- D9: Glass lawn

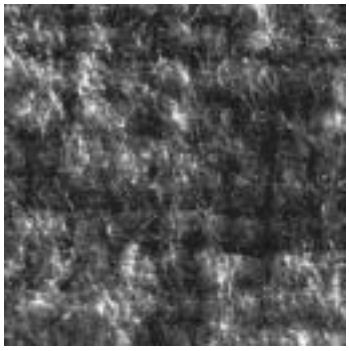
- D11: Homespun woolen cloth



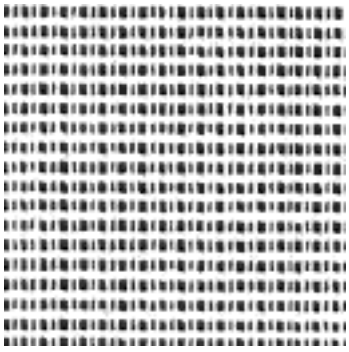
- D16: Herringbone weave



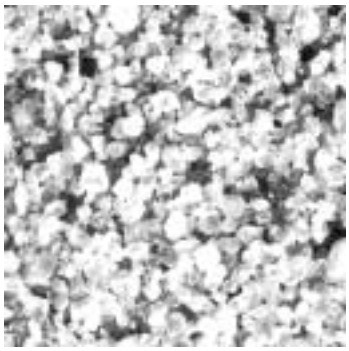
- D19: Woolen cloth



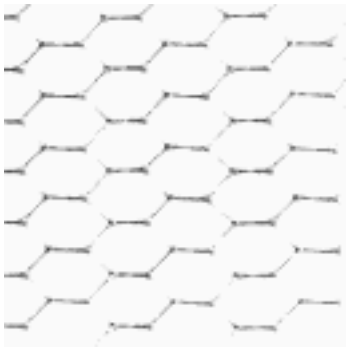
- D21: French canvas



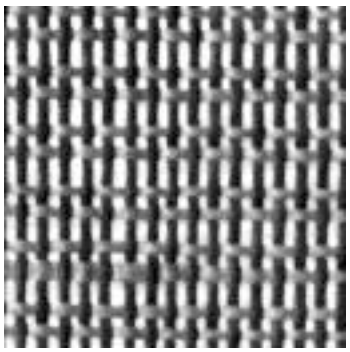
- D29: Beach sand



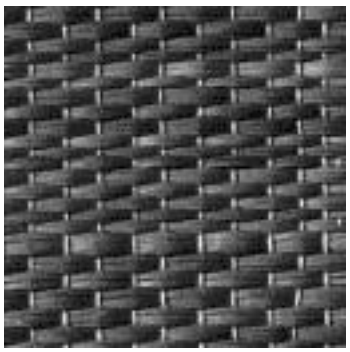
- D34: Netting



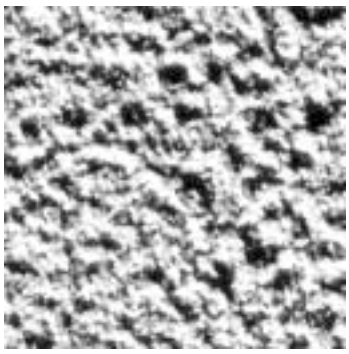
- D53: Oriental straw cloth



- D55: Straw matting

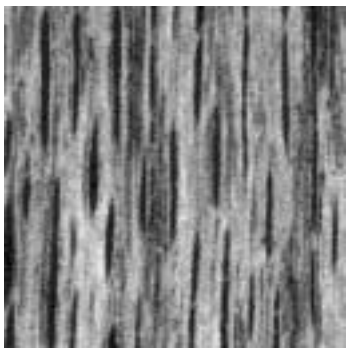


- D57: Handmade paper canvas



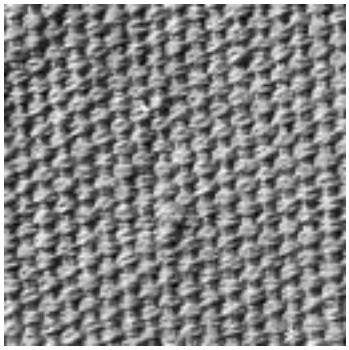
- D65: Handwoven Oriental rattan

- D68: Wood grain cloth

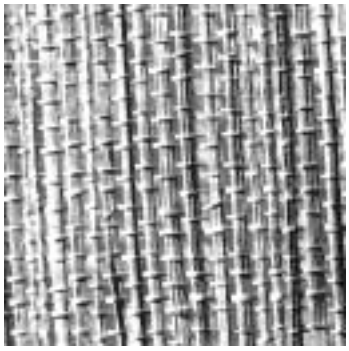


- D74: Coffee beans

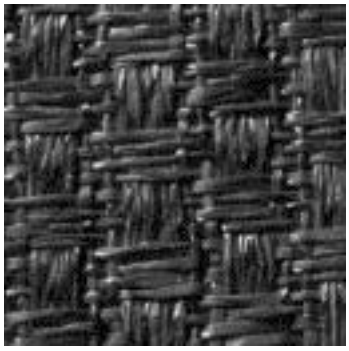
- D77: Cotton



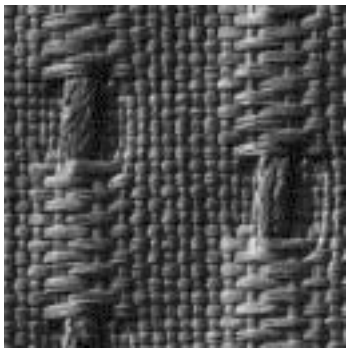
- D79: Oriental grass fiber



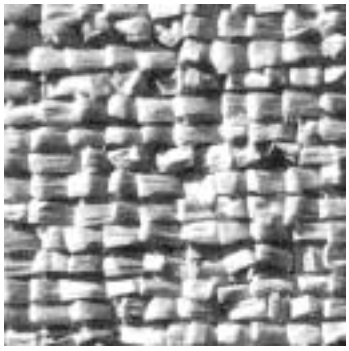
- D82: Oriental straw cloth



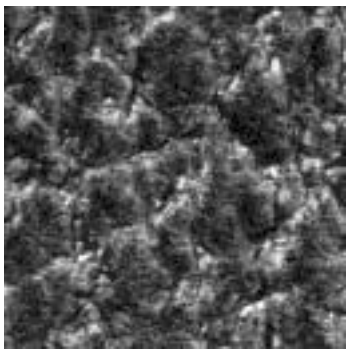
- D83: Woven matting



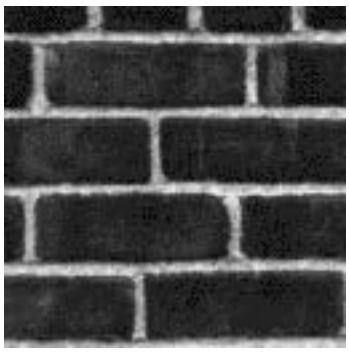
- D84: Raffia looped to a high pile



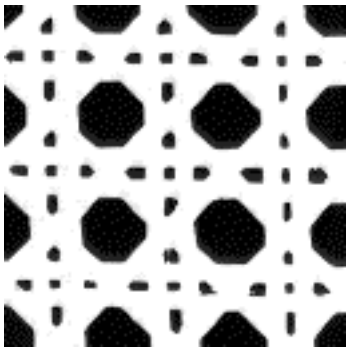
- D92: Pigskin



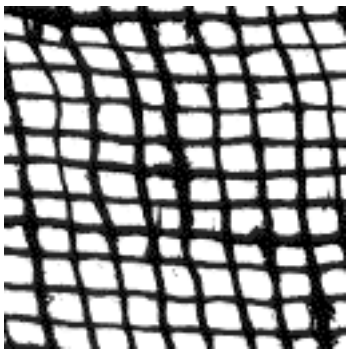
- D95: Brick wall



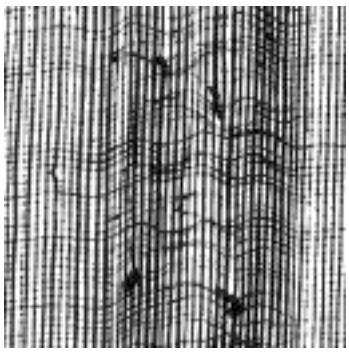
- D102: Cane



- D103: Loose burlap



- D105: Cheesecloth



付録 B

ロボットアームの仕様

項　目		仕　様	備考
構造		５自由度 垂直多関節形	
動作範囲	ウエイスト回転	300°　（MAX 120°/sec）	J1 軸
	ショルダ回転	130°　（MAX 72°/sec）	J2 軸
	エルボ回転	110°　（MAX 109°/sec）	J3 軸
	リストピッチ	± 90°（MAX 100°/sec）	J4 軸
	リストロール	±180°（MAX 163°/sec）	J5 軸
アーム長　さ	アッパーアーム	250mm	
	フォアアーム	160mm	
可搬重量		max. 1.2kgf（ハンド重量を含む）	重心
最大合成速度		1000mm/sec （リストツール面）	
位置繰り返し制度		±0.3mm （リストツール面のロール中心）	
駆動方式		DC サーボモータによる電気サーボ駆動	
本体重量		約 19kgf	
モータ容量		J1〜J3 軸 30W, J4,J5 軸 11W	

参 考 文 献

[1] 畝見達夫. 強化学習. 人工知能学会誌, Vol. 9, No. 4, pp. 830–836, 1994.

[2] C. J. C. H. Watkins. *Learning from Delayed Rewards*. PhD thesis, Cambridge University Psychology Department, 1989.

[3] C. J. C. H Watkins and P. Dayan. Technical note on Q-learning. *Machine Learning*, Vol. 8, pp. 279–292, 1992.

[4] 滝寛和. 構成的帰納学習とバイアス. 人工知能学会誌, Vol. 9, No. 6, pp. 818–822, 1994.

[5] D. W. Aha. Incremental constructive induction: An instance-based approach. In *Proc. of the Eighth International Workshop on Machine Learning*, pp. 117–121, 1991.

[6] D. W. Aha. Case-based learning algorithms. *Case-Based Reasoning Workshop*, pp. 147–158, 1991.

[7] D. Chapman and L. P. Kaelbling. Input generalization in delayed reinforcement learning: An algorithm and performance comparisons. In *Proc. of the Twelfth International Joint Conference on Artificial Intelligence*, pp. 726–731, 1991.

[8] M. Asada, S. Noda, and K. Hosoda. Action-based sensor space categorization for robot learning. In *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems 1996*, pp. 1502–1509, 1996.

[9] H. Ishiguro, R. Sato, and T. Ishida. Robot oriented state space construction. In *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems 1996*, pp. 1496–1501, 1996.

[10] G. Drastal, R. Meunier, and S. Raatz. Error correction in constructive induction. In *Proc. of the Sixth International Workshop on Machine Learning*, pp. 81–83, 1989.

[11] 宮本行庸, 上原邦昭. 特徴構成法を用いたQ学習の効率改善. 情報処理学会論文誌：数理モデル化と応用, Vol. 40, No. SIG9 (TOM2), pp. 62–71, 1999.

[12] 宮本行庸, 上原邦昭. 特徴構成法を用いたQ学習の効率改善. 情報処理学会研究報告, Vol. 98, No. 105, pp. 57–62, 1998.

[13] 宮本行庸, 上原邦昭. 構成的帰納学習と強化学習の統合による知的エージェントの学習効率の向上. 情報処理学会第53回全国大会講演論文集, 分冊(2), pp. 183–184, 1996.

[14] L. P. Kaelbling. Learning functions in k-dnf from reinforcement. In *Proc. of the Seventh International Conference on Machine Learning*, pp. 162–169, 1990.

[15] Y. J. Hu and D. Kibler. Generation of attribute for learning algorithms. In *Proc. of AAAI-1996*, pp. 806–811, 1996.

[16] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.

[17] R. S. Sutton. Integrated architectures for learning, planning, and reacting based on approximating dynamic programming. In *Proc. of the Seventh International Conference on Machine Learning*, pp. 216–224, 1990.

[18] R. Maclin and J. W. Shavlik. Incorporating advice into agents that learn from reinforcements. In *Proc. of AAAI-1994*, pp. 694–699, 1994.

[19] M. Tan. Multi-agent reinforcement learning: Independent vs. cooperative agents. In *Proc. of the Tenth International Conference on Machine Learning*, pp. 330–337, 1993.

[20] T. R. Reed and J. M. H. du Buf. A review of recent texture segmentation and feature extraction techniques. *CVGIM: Image Understanding*, Vol. 57, No. 3, pp. 359–372, 1993.

[21] T. Randen and J. H. Husøy. Filtering for texture classification: A comparative study. *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 21, No. 4, pp. 291–310, 1999.

[22] R. W. Conners and C. A. Harlow. A theorical comparison of texture algorithms. *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 2, No. 3, pp. 204–221, 1980.

[23] T. Ojara, M. Pietikäinen, and D. Harwood. A comparative study of texture measures with classification based on feature distributions. *Pattern Recognition*, Vol. 29, No. 1, pp. 51–59, 1996.

[24] A. Teuner, Pichler O., and B. J. Hosticka. Performance evaluation for four classes of textual features. *IEEE Trans. Image Processing*, Vol. 4, No. 6, pp. 863–870, 1995.

[25] R. M. Haralick, K. Shanmugam, and I. Dinstein. Textual features for image classification. *IEEE Trans. Systems, Man, Cybernetics*, Vol. 3, No. 6, pp. 610–621, 1973.

[26] J. Strand and T. Taxt. Local frequency features for texture classification. *Pattern Recognition*, Vol. 27, No. 10, pp. 1397–1406, 1994.

[27] M. Vetterli and J. Kovacevic. *Wavelets and Subband Coding.* Prentice-Hall, New Jersey, 1995.

[28] S. G. Mallat. A theory for multiresolution signal decomposition. *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 11, No. 7, pp. 674–693, 1989.

[29] R. R. Coifman and M. V. Wickerhauser. Entropy-based algorithms for best basis selection. *IEEE Trans. Information Theory*, Vol. 38, No. 2, pp. 713–718, 1992.

[30] R. R. Coifman and M. V. Wickerhauser. Best-adapted wavelet packet bases. preprint, Yale Univ., Feb. 1990.

[31] T. Chang and C.-C. J. Kuo. Texture analysis and classification with tree-structured wavelet transform. *IEEE Trans. Image Processing*, Vol. 2, No. 40, pp. 429–441, 1993.

[32] 宮本行庸, マハダド ヌリシラジ, 野田秀樹, 上原邦昭. 木構造ウェーブレット変換の構造を用いたテクスチャ解析および分類. 電子情報通信学会技術研究報告, Vol. 99, No. 575, pp. 97–104, 2000.

[33] Y. Miyamoto, M. N. Shirazi, and K. Uehara. Texture analysis and classification using bottom-up tree-structured wavelet transform. In *Proceedings of the Sixth Pacific Rim International Conference on Artificial Intelligence*, p. 802, 2000.

[34] M. N. Shirazi, Y. Miyamoto, H. Noda, and K. Uehara. Structural stability and discrimination capability of tree-structured wavelet transform. *International Journal of Pattern Recognition and Artificial Intelligence* (submitted).

[35] Y. Miyamoto, M. N. Shirazi, and K. Uehara. Structural stability analysis for texture recognition. In *Proceedings of IAPR Workshop on Machine Vision Applications*, pp. 275–278, (2000).

[36] 宮本行庸, マハダド ヌリシラジ, 上原邦昭. ウェーブレット特徴量を用いたテクスチャのボトムアップ的構造化と分類. 情報処理学会論文誌 (投稿中).

[37] P. Brodatz. *Textures: A Photographic Album for Artists & Designers.* Dover, New York, 1966.

[38] M. Unser. Texture classification and segmentation using wavelet frames. *IEEE Trans. Image Processing*, Vol. 4, No. 11, pp. 1549–1560, 1995.

[39] R. Milanese and M. Cherbuliez. A rotation, translation, and scale-invariant approach to content-based image retrieval. *Journal of Visual Communication and Image Representation*, Vol. 10, pp. 186–196, 1999.

[40] J. C. Terrillon, M. David, and S. Akamatsu. シーン画像中の人物顔パターンの自動検出. 第3回画像センシングシンポジウム講演論文集, pp. 11–16, 1999.

[41] Y. Li. Reforming the theory of invariant moments for pattern recognition. *Pattern Recognition*, Vol. 25, No. 7, pp. 723–730, 1992.

[42] M. K. Mandal, T. Aboulnasr, and S. Panchanathan. Image indexing using moments and wavelets. *IEEE Trans. Consumer Electronics*, Vol. 42, No. 3, pp. 557–565, 1996.

[43] 浅田稔, 野田彰一, 細田耕. ロボットの行動獲得のための状態空間の自律的構成. 日本ロボット学会誌, Vol. 15, No. 6, pp. 886–892, 1997.

[44] 高橋泰岳, 浅田稔. 実ロボットによる行動獲得のための状態空間の漸次的構成. 日本ロボット学会誌, Vol. 17, No. 1, pp. 118–124, 1999.

[45] 宮本行庸, 河合克己, 草間利晃, 角谷和俊, 上原邦昭. 概念学習を用いた環境の分類に基づく行動モデルの強化学習による拡張. 情報処理学会論文誌：数理モデル化と応用 (投稿中).

[46] 宮本行庸, 上原邦昭, 前川禎男. 強化学習を用いたロボットアームの障害物回避問題の学習. 平成6年電気関係学会関西支部連合大会講演論文集, p. G305, 1994.

[47] N. J. Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, 1991.

[48] 宮本弘之, 川人光男. 作業レベルのロボット学習のための見まねによる教示. 電子情報通信学会論文誌, Vol. J81-D-II, No. 10, pp. 2401–2410, 1998.

[49] 宮本弘之, 森本淳, 銅谷賢治, 川人光男. 経由点を用いた強化学習. 電子情報通信学会論文誌, Vol. J82-D-II, No. 11, pp. 2111–2117, 1999.

[50] T. G. Dietterich and N. S. Flann. Explanation-based learning and reinforcement learning: A unified view. In *Proc. of the Twelfth International Conference on Machine Learning*, pp. 92–101, 1992.

[51] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*, pp. 629–632. Prentice Hall, 1995.

[52] 木村元, L. P. Kaelbling. 部分観測マルコフ決定過程下での強化学習. 人工知能学会誌, Vol. 12, No. 6, pp. 822–829, 1997.

[53] L. Chrisman. Reinforcement learning with perceptual aliasing: The perceptual distinctions approach. In *Proc. of Tenth National Conference on Artificial Intelligence*, pp. 183–188, 1992.

[54] R. A. McCallum. Overcoming incomplete perception with utile distinction memory. In *Proc. of Tenth International Conference on Machine Learning*, pp. 190–196, 1993.

[55] A. D. Christiansen. Learning to predict in uncertain continuous tasks. In *Proc. of Ninth International Conference on Machine Learning*, pp. 72–81, 1992.

[56] L. J. Lin. Scaling up reinforcement learning for robot control. In *Proc. of Tenth International Conference on Machine Learning*, pp. 182–189, 1993.

[57] 三菱電機株式会社. 三菱電機マイクロロボット RV-M1 MOVEMASTER EX 取扱説明書, 1991.

[58] 山口智浩, 増渕元臣, 藤原一継, 谷内田正彦. 抽象化副目標の自動生成による実ロボット強化学習の高速化. 人工知能学会誌, Vol. 12, No. 5, pp. 712–723, 1997.

研究業績

学術論文

1. 宮本行庸, 上原邦昭. 特徴構成法を用いた Q 学習の効率改善. 情報処理学会論文誌：数理モデル化と応用, Vol. 40, No. SIG9 (TOM2), pp. 62–71, 1999.

2. Y. Miyamoto, M. N. Shirazi, and K. Uehara. Texture analysis and classification using bottom-up tree-structured wavelet transform. In *Proceedings of the Sixth Pacific Rim International Conference on Artificial Intelligence*, p. 802, 2000.

3. Y. Miyamoto, M. N. Shirazi, and K. Uehara. Structural stability analysis for texture recognition. In *Proceedings of IAPR Workshop on Machine Vision Applications*, pp. 275–278, 2000.

4. 宮本行庸, マハダド ヌリ シラジ, 上原邦昭. ウェーブレット特徴量を用いたテクスチャのボトムアップ的構造化と分類. 情報処理学会論文誌 (投稿中).

5. 宮本行庸, 河合克己, 草間利晃, 角谷和俊, 上原邦昭. 概念学習を用いた環境の分類に基づく行動モデルの強化学習による拡張. 情報処理学会論文誌：数理モデル化と応用 (投稿中).

6. M. N. Shirazi, Y. Miyamoto, H. Noda, and K. Uehara. Structural stability and discrimination capability of tree-structured wavelet transform. *International Journal of Pattern Recognition and Artificial Intelligence* (submitted).

学術報告

1. 宮本行庸, 上原邦昭. 特徴構成法を用いた Q 学習の効率改善. 情報処理学会研究報告, Vol. 98, No. 105, pp. 57–62, 1998.

2. 宮本行庸, マハダド ヌリ シラジ, 野田秀樹, 上原邦昭. 木構造ウェーブレット変換の構造を用いたテクスチャ解析および分類. 電子情報通信学会技術研究報告, Vol. 99, No. 575, pp. 97–104, 2000.

学術講演

1. 宮本行庸, 上原邦昭, 前川禎男. 強化学習を用いたロボットアームの障害物回避問題の学習. 平成 6 年電気関係学会関西支部連合大会講演論文集, p. G305, 1994.

2. 宮本行庸, 上原邦昭. 構成的帰納学習と強化学習の統合による知的エージェントの学習効率の向上. 情報処理学会第 53 回全国大会講演論文集, 分冊 (2), pp. 183–184, 1996.