

画像理解のための認識ルール獲得支援システムの一提案

佐藤真一, 坂内正夫
東京大学生産技術研究所

画像理解システムの認識ルールとなることを想定した構造記述ルールを, ユーザーより与えられる図形の形で表現された「例」から, 帰納学習を用いて獲得するシステムについて述べる. 例の表現としては, 空間性を保持した柔軟なものを使い, より高度な構造記述の生成にも対応できるようになっている. また構造記述を, その複雑さという観点から分類し, その分類に従った構造記述の学習に適した学習アルゴリズムについて考察している. 今回作成したシステムは, 考察した構造記述のサブセットの比較的簡単なものを生成するシステムであるが, 生成しうる範囲であれば比較的複雑な構造記述でもうまく生成できることが確認された.

A Proposal of Recognition Rules Acquisition Supporting System Intended to be Used for Image Understanding

Shin'ichi SATOH, Masao SAKAUCHI
Industrial Institute of Science, University of Tokyo
7-22-1, Roppongi, Minato-ku, Tokyo 106, Japan

We propose a system which learns structural description rules from given "example" figures by user, supposed to be used as a recognition rules generation system for image understanding system. "Examples" hold spatial features to deal with more complex structural descriptions generation. And after introducing a new classification method of structural descriptions paying attention to its complexity, we propose a new learning algorithm appropriate to generating structural descriptions following that classification. We embody an experimental learning system that can generate subset of structural descriptions referred here, however, this system can generate relatively complex structural descriptions to its extent.

1 はじめに

画像データベースやマルチメディアデータベースなど、画像データを扱うデータベースを実用可能な水準で構築するには、内部に蓄える画像データを高度に抽象化してデータベースユーザーからの高度な要求に答えられるような形にしておく必要がある。この抽象化を実現するため、高度な認識プロセスを行うことのできる画像理解システムが望まれる。大規模な画像／マルチメディアデータベースの構築のためには、莫大な種類の画像データに対応してその種類ごとに画像理解システムが必要となるが、その構築は大変な作業であり、各種画像向きの画像理解システムをそのつと構築していたのでは、実用的なデータベースの構築は極めて困難である。もしこの画像理解システムがユーザーからの簡単な指示で自動的に構築できるとしたら、これは画像／マルチメディアデータベースの実現のための1つの大きな突破口となりうる。

一般に、複雑で多様な対象物を認識するような画像理解システムには、モデルベースで認識を行うものが多い。この時の「モデル」とは、それが表現する対象物の構成要素による構造記述であると考えることができる。そこでここでは、画像理解システムの自動構築の実現のための第一歩として、ユーザーより与えられたいくつかの図形で表現された事例から、これらを帰納学習により一般化してその構造記述を生成するシステムについて述べる。このシステムにより、図形で構成されたある対象物について、それに関する事例をシステムに与えることによって、対象物の構造記述という形でそのモデルを自動生成できることになる。

システムは、構造記述を得たいある対象について、その対象の範疇に含まれるいくつかの事例(正の例と呼ぶ)と、その範疇に含まれないようないくつかの事例(負の例と呼ぶ)とをユーザーからの入力とし、帰納推論を行うことにより対象の構造記述を表す構造記述ルールを生成するようになっている。この「例」は、特別な記述作成システムによって生成される意味ネットワークなどのような記述で与えられるのではなく、基本的なオブジェクトが2次元空間中に配置されている、いわばCADデータのような形で与えることができる。また、構造記述の構成方法について考察し、システムが行う帰納推論は、その考察を基に

して図形の関係についての推論に適したアルゴリズムを用いている。

以下本稿では、まずシステムの概要について述べる。ついでシステムの特徴であるユーザーからの「例」の記述について述べ、空間性／幾何学性を持っている画像データのような対象に対する構造記述、およびそのための帰納学習について考察し、これに対応した学習アルゴリズムについて述べる。そしてシステムの実装の概要について述べ、その実行例を紹介する。最後に本研究に関する考察を行う。

2 システムの概要

本システムは、図形的に表現しうるある対象について、ユーザーから与えられる図形的に表されたいくつかの例から帰納的に推論し、その対象を表す構造記述を学習するシステムである。この「例」には、その対象を表現している「正の例」とそれを表現していない「負の例」とがある。例えば「家」という対象の構造記述が得たいとし、図1で示されるような例を与えんとする。図中で true と書かれているものは正の例であり、false と書かれているのは負の例である。これらの例が入力された時、システムはそれから推論して、例えば「三角形と長方形の2つのオブジェクトが存在し、それらは互いに接していて、三角形が長方形の上方にある」というような構造記述を生成することになる。

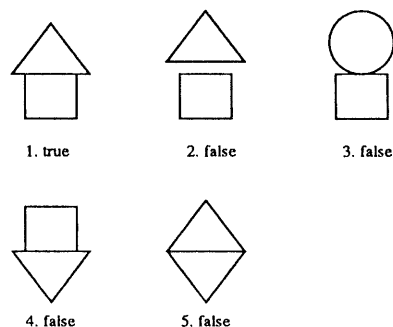


図1: 「家」の学習のための「例」

このように与えられた例からその構造記述を学習する研究は、Winstonのシステム[1]をはじめとしていくつかなされているが、それらはいずれもシステ

ムに与える例としては強力な空間記述作成システムなどを用いて得た意味ネットワークなどの形で与えるようになっていて、学習システムとしては、与えられた意味ネットワークを一般化することによってその概念を表すより一般的な構造記述を生成するようになっている [1, 2, 3]. この場合、当然得られる結果はその記述作成システムの生成する記述に強く依存することになるため、満足しうる構造記述モデルを得るためには記述作成システムを十分強力にしておく、それが生成する空間記述を十分豊かなものにする必要がある。

このようにして十分強力な記述作成システムによって作成した豊かな空間性を持った記述であるが、実際にそれから構造記述を学習する際、一般には作成した詳細な記述の内のごくわずかな部分しか参照しないことは十分に考えられる。また、空間記述を作成した際には予想されなかったような幾何関係を学習の際に参照したくなる場合もありうる。

そこでここでのアプローチは、与えられた例はそれを構成するいくつかのオブジェクトの種類とその形状、およびその存在場所などをそのままの形で記述しておき、学習する段階で必要に応じてそれらの空間的關係を作成するというものである。こうすることにより不必要な空間的關係を調べる必要はなくなり、かつ学習の際に生じた新たな空間的關係の調査への要求に対しても新たに「例」の記述を作り直す必要はなく、動的に対応することができる。必要に応じて調査することによる効率の低下に対しては、オブジェクトを空間データ構造 BD 木 [5] で管理することにより、この空間的な探索を高速に行うようにして対処している。この「例」の記述については 3 節で説明する。

学習アルゴリズムは、Shapiro のモデル推論システム [4] をベースにし、構造記述ルールの生成向きに作成した構造記述ルール生成アルゴリズムを用いることにより実現している。この構造記述ルールの生成は、新たに考察した構造記述の分類に基づいたアルゴリズムを用いている。構造記述については 4 節で述べる。学習アルゴリズムでは、構造記述を表す構造記述ルールとしてはじめはより一般的なルールを生成し、順次これを特殊化しつつ与えられた例との整合性を調べることにより、より一般的な空間的状况を表すルールを生成してくるようになっていく。このように与えられた例に矛盾しないような構造記述

ルールのうちより一般的なものを生成してくるようになっていくため、少ない例でかなり多くの状況を識別できるような一般的な構造記述を生成することができる。構造記述ルール生成アルゴリズムでは、かなり特殊な構造記述ルールまで生成することができるため、より複雑な対象についてもこれを表す構造記述が生成できる。学習アルゴリズムについては 5 節で述べる。

3 システムに与えられる「例」の記述

前節で述べた通り、従来の構造記述学習システムでは、システムに与えられる例としては空間記述作成システムなどを用いて得た意味ネットワークなどのような形で与えるようになっていた。例えば Winston のシステムでは、図 2(a) の図形は図 2(b) で表される意味ネットワークの形でシステムに与える必要があった [2]。こうした場合、システムはこの意味ネットワークの記述しか参照できないため、これに記述されていないオブジェクトの關係は決して知ることができない。例えば図中の E2 の図形で、a と c というオブジェクトの位置關係を調べたいと思った場合、その間のリンクが張られていないため、システムはその間の關係を知ることができない。

そこでここで述べるシステムに与える例の記述は、特殊な記述生成システムを用いるのではなく、基本的な幾何学オブジェクトとその空間的な配置との組で与えられる記述を用いる。現在のシステムの実装では、幾何学的オブジェクトは線分・円・多角形のいずれかである。空間的な配置とは、2 次元空間中でのオブジェクトの存在位置を座標で与えたものである。具体的には、図 2(a) の図形は図 2(c) のような記述としてシステムに与えられることになる。

この記述はもとの図形の空間性をそのまま保持しているため、オブジェクト間の空間的な關係を柔軟に調べることができる。その反面この空間的な關係は必要に応じて直接調査しなければならないことになり、それにより実行の効率が落ちてしまうことが考えられるが、ここでは各オブジェクトを空間データ構造 BD 木 [5] で管理して、空間的な探索を高速に行うことにより、この効率の低下を最小限に抑えている。

4 システムで扱う構造記述

4.1 構造記述の分類

3 節で述べたオブジェクトの配置で構成された図形を一般化した構造記述の表現は、例えば 2 節で述べたように、「家」を表す構造記述は「三角形と長方形の 2 つのオブジェクトが存在し、それらは互いに接していて、三角形が長方形の上方にある」というように表される。これは、「オブジェクトが存在する」「三角形である」「接している」などのいくつかの要素から成っている。これをここでは「空間関係」と呼ぶ。構造記述はこれらの空間関係の組合せで構成される。本節ではこの空間関係を、構造記述を自動生成するために必要な構造記述の複雑さという観点に立て分類し、それによって構造記述を分類することを考える。

まず空間関係の分類は以下のように行う。

0 次空間関係 — オブジェクトの存在を調べる。

1 次空間関係 — オブジェクトの種類を分類する。

2 次空間関係 — 2 つのオブジェクト間の関係を調査する。

...

n 次空間関係 — n 個のオブジェクト間の関係を調査する。

これに従うと、「オブジェクトが存在する」は 0 次空間関係、「三角形である」は 1 次空間関係、「接している」は 2 次空間関係ということになる。

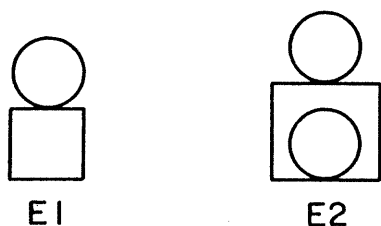
次に、これらを用いて構成する構造記述を以下のように分類する。

0 次構造記述 — 0 次空間関係のみを用い、対象を構成するオブジェクトの個数のみで表す。

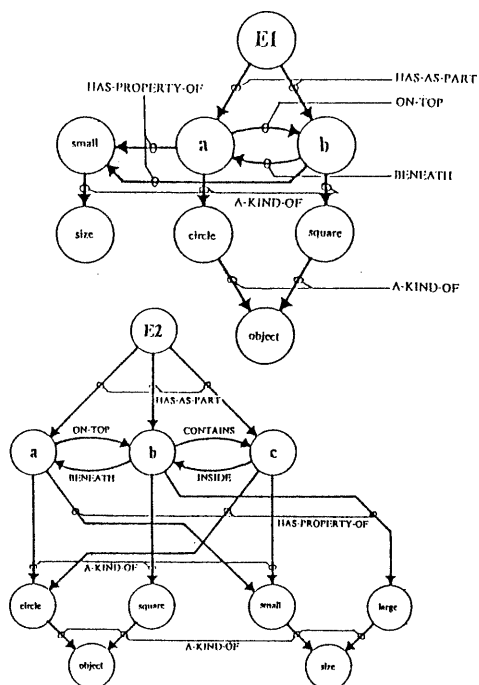
1 次構造記述 — 0 次構造記述に加え、各オブジェクトの種類 (1 次空間関係) によって表す。

...

n 次構造記述 — $n - 1$ 次構造記述に加え、対象を構成するオブジェクトのうちの n 個の間の n 次空間関係によって表す。



(a) 図形による表現



(b) 意味ネットワークによる表現

E1: [circle((15,25),5),
polygon([(10,10),(20,10),(20,20),
(10,20)])]

E2: [circle((15,29),5),
polygon([(8,10),(22,10),(22,24),
(8,24)]),
circle((15,15),5)]

(c) 本システムに与えられる表現

図 2: システムに与える「例」の記述 ((a),(b) は [2] より)

従って、上で述べた「家」を表す構造記述は2次構造記述であることになる。

一般には、ある1つの「構造記述ルール」は、対象を構成するオブジェクトの個数を規定する部分(0次構造記述)と各オブジェクトの間の1~ n 次空間関係の連言で表される。さらにある対象を表す構造記述としては、これらのルールの選言の形をとれるものとする。

4.2 構造記述の複雑さ

n 次構造記述の考え方は、一般的な構造記述として十分利用できると思われる。ただし、

- n が十分大きいこと。
- 1~ n 次空間関係がそれぞれ十分に強力なこと。

が必要である。

今、構造記述の複雑さを表す1つの指標として「構造記述空間 S 」というものを考えて、その空間 S の広がりを考える。0~ n 次空間関係のそれぞれの複雑さは互いに独立と考えられるので、これらは空間 S の次元を表すと考えられる。上で述べた条件のうち、「 n が十分大きいこと」は空間 S の次元数を対象の構造を表すのに十分なほど大きくとることに相当する。また、次元数を大きくするだけでは空間 S は広がらないので、各次元での範囲を大きくする必要がある。各次元ごとの広がりはそれぞれ対応する空間関係の複雑さにあたると考えると、さらに「1~ n 次空間関係がそれぞれ十分に強力」になると空間 S が対象の構造記述を表すのに十分な程度広くなることになる。

以下で、各空間関係をより高度なものにするための方策について考察する。

4.3 1 次空間関係

学習システムに与えられる図形の記述について考えると、現在のシステムでは、いくつかの基本的な幾何学オブジェクトとその空間的な配置との組で与えられるようになっている。1 次空間関係では、オブジェクトの種類を分類するようになっているので、今回の実装では線分・円・多角形を分類するルールが生成でき、多角形についてはさらに三角形・長方形を分類するルールが生成できるようになっている。今回はこの状態で実験を行ったが、それぞれのオブジェ

クトの種類は、学習プロセスに先だってあらかじめ認識されている必要があるため、認識されなかったオブジェクトに対応できない。また、各オブジェクトは基本的な幾何学オブジェクトのみであり、より複雑な構造を持った対象物をオブジェクトとして扱うことができない。今後より複雑な対象の構造記述を生成するようにシステムを拡張することを考えると、この1次空間関係をより高度化する必要が出てくる。

高度化の方針としては以下のようなものが考えられる。

- 形状などの情報が不完全でオブジェクトの種類が完全に認識されていないようなものを扱う。
- いくつかのオブジェクトをグルーピングして扱う。
- より複雑な構造を持った対象を単一のオブジェクトとして扱う。

これらを実現するためには、具体的には次のような方法が考えられる。

- 完全に認識されていないオブジェクトを使う — これに対しては、
 - 状態遷移モデル [6] で得られる記述のように、認識の中間状態を用いる。
 - オブジェクト指向データモデルで記述されたモデルのように、オブジェクトの内部構造にまで立ち入らなくても利用できる記述を用いる。

などの方法が考えられる。いずれの方法でも、オブジェクトが既知のある種類に分類されていない状態でも、それまでに分かっている画像的な特徴は利用可能であり、これを手がかりにそのオブジェクトを含む図形の構造記述ルールを生成するようになる。

- オブジェクトをグルーピングして扱う — オブジェクトの対象性や配置の規則性に着目して、例えばいくつかのオブジェクトが極めて規則正しく一列にならんでいるような場合、これを一つのオブジェクトであるかのように扱うことが考えられる。このように、規則性などのボトムアップ的に得られる情報からグルーピングできる場

合、これをまとめて一つのオブジェクトとして扱うのが効率が良い場合がある。

- より複雑な構造を持ったオブジェクト — ある構造記述にしたがって構成されている複雑な構造を持ったオブジェクト群を、一つのオブジェクトとして扱いたい場合がありうる。これを実現するには、別に求めた構造記述ルールで認識された図形を一つのオブジェクトであるとしておき、この記述を用いてさらに上位の階層の構造記述を求めるといった方法が考えられる。これによって構造記述の入れ子構造ができることになる。学習プロセスが進んで、さまざまな対象の構造記述ルールが蓄えられていけばいくほど、さらに複雑な構造記述が識別できるようになる。

4.4 2～ n 次空間関係

2 次空間関係としては、今回のシステムでは adjacent, on, cross, in, upper, lower, right, left の 8 つが生成できる。これらは固定的にシステムに搭載されているが、システムに与えられる図形は前述のように空間的に与えられるため、この 2 次空間関係を動的に生成することもできる。これを考慮して 2 次空間関係の一般形を考えると、以下のようになる。

- 両図形間の交差状況。
- 図形の主軸の長さと同図形間の距離の比。
- 主軸と両図形の重心を結ぶ線とのなす角度。

これらで一般の 2 次空間関係を表すことができると考えられる。

今回のシステムでは 3 次以上の n 次空間関係を生成する能力はないが、より複雑な対象物の構造記述を表すためには n 次空間関係も必要になることもありうる。この n 次空間関係の一般形については、現在考察中である。

5 学習アルゴリズム

5.1 学習アルゴリズムの基本的な流れ

いま、 $T_1, \dots, T_n$ をシステムに与えられた正の例、 $F_1, \dots, F_m$ を負の例とする。前述のように正の例と

は対象の構造記述が満たされるべき図形であり、負の例とは構造記述が満たされざるべき図形であるので、この時システムは、正の例はすべて満たし負の例はすべて満たさないような構造記述のうちで、もっとも一般的な構造記述を生成しようとする。より一般的な構造記述を生成することにより、例として与えられなかったような図形に対しても、構造記述の分類ができるようになることを意図している。

学習アルゴリズムの基本的な流れは以下のようになっている ([4])。

- (1) ルールの集合 R を空にする。
- (2) 任意の正の例 T_i を選び、それを着目している正の例 T とする。
- (3) T を満たし、すべての F_j を満たさないルールのうち、もっとも一般的なものを構造記述ルール作成アルゴリズムにより探索し、 R に追加する。
- (4) R ですべての T_i が満たされるかどうかを調べ、満たされないものがあればそれを T として (3) から繰り返す。
- (5) すべての T_i が満たされ、すべての F_j を満たさないルール集合 R が決定されれば、それを結果の構造記述として出力する。

このアルゴリズムにより、与えられたすべての正の例を満足し、すべての負の例を排除するような構造記述が生成できることになる。より一般的な構造記述を生成するための部分や、対象の空間性に依存した部分は、もっぱら構造記述ルール生成アルゴリズムに依存している。このアルゴリズムについては以下に述べる。

5.2 構造記述ルール生成アルゴリズム

構造記述ルール生成アルゴリズムの基本的な考え方は、「与えられた 1 つの例を満足するようなルールのうち、より一般的なルールから、それを少しずつ特殊化したものを順次生成して行く」というものである。このルールを段階的に特殊化していく過程は、前に導入した構造記述空間中を最も一般的な点から順次それを特殊化した点へと探索する過程となる。特殊化の方向は、構造記述空間をさまざまに広がっていくため、最も一般的なルールを根として探索の過

程は木状に広がって行き、各ノードがまたそれぞれさまざまな方向に特殊化されていくことになる。探索の途中で生成されたノードで、満たすべき与えられた例を満たさないものがあればそのノードは刈り取られ、以後それが特殊化されることはなくなる。こうして生成される一連のルールは、すべての負の例を満たさないかどうか調べられ、もしそうであれば結果として出力され、1つでも負の例を満たしてしまえばそのルールはさらに特殊化されることになる。

構造記述空間が多次元空間であることから分かるように、構造記述の複雑さは1次元的に並べて評価することができない。それでも探索もれがないように、少しずつ特殊化した構造記述を順次生成しなければならない。具体的には、構造記述の次数を上げればその複雑さは一般に増加するから、これを順次上げていくことにより生成される構造記述を少しずつ特殊化することができる。その上各次数の空間関係の複雑さもあるので、並行してこれらの空間関係も少しずつ複雑にしていくようなアルゴリズムを使う必要がある。

今回の実装では各空間関係はあらかじめ定まったものに限られており、その複雑さは固定であると考えることができるので、特に空間関係の次数を増やす方向のみで構造記述の特殊化を行うようになっていく。具体的には、

- (1) もっとも一般的なルールとしては、何のチェックもせずに true とするもの。
- (2) そのルールに 0 次空間関係 (オブジェクトの存在) を調べるルールを追加したもの。
- (3) これまで生成したそれぞれのルールに対し、1 次空間関係 (オブジェクトの種類) を調べるルールを追加したもの。
- (4) さらに 2 次空間関係 (任意の 2 つのオブジェクト間の関係) を調べるルールを追加したもの。

という順番に、構造記述向きに少しずつルールを特殊化するようになっていく。今回の実装では 2 次空間関係までを生成できるようになっているが、3 以上の空間関係の生成にこのアルゴリズムを用いる方法は自明であろう。

6 実行例

ここまで述べてきたアルゴリズムに従い、構造記述学習システムを作成した。システムは 15Mips の Sparc Station 1+ 上で動作し、SICStus Prolog で記述した本体部分と、それにリンクする形の C 言語で記述した BD 木ライブラリとからなる。

まず図 1 の図形をシステムに与えた場合の実行例を示す。1 から 4 までの例をシステムに提示すると、システムは次のような構造記述を出力してきた。

```
model(A) :-
    obj(B, A, C),
    onobj(D, B, C, E),
    is_triangle(D),
    upper(D, B).
```

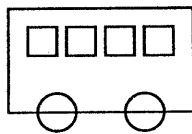
これを書き下すと、「あるオブジェクト B が存在し、その上方に接触して三角形 D が存在する」となる。システムができるだけ一般的な構造記述を生成しようとしているため、オブジェクト B の形状について言及していないことが分かる。1 から 5 までのすべての例をシステムに提示すると、

```
model(A) :-
    obj(B, A, C),
    onobj(D, B, C, E),
    is_rectangle(B),
    is_triangle(D),
    upper(D, B).
```

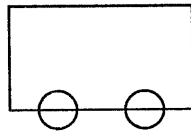
という構造記述が生成された。これは、「長方形 B が存在し、その上方に接触して三角形 D が存在する」ということを表している。これはまさに 2 節で想定していた構造記述そのものである。なお、この記述を生成するのに 2cpu 秒を要した。

次に、やや複雑な例として、図 3 に「パス」という概念の図形による例を示す。1 が正の例であり、2 から 5 が負の例である。これら 5 つの例から、システムは以下に示す構造記述を生成した。

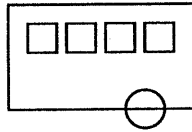
```
model(A) :-
    obj(B, A, C),
    crossobj(D, B, C, E),
    crossobj(F, B, E, G),
    inobj(H, B, G, I),
    is_circle(F),
```



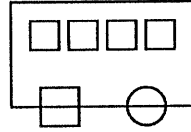
1. true



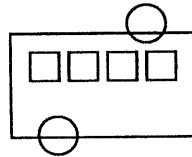
2. false



3. false



4. false



5. false

図 3: 「バス」の学習のための例

```
is_circle(D),
lower(F, B),
lower(D, B).
```

これは、「オブジェクト B, D, F, H が存在し, D, F は B と交差し, H は B の内部にあり, F と D は円であって, それぞれ B の下方に存在する」ということを表している。「バス」を構成する 2 つのタイヤ (F, D) の形状と位置については言及しているが, 車体 (B) と窓 (H) の形状などには何も言及していないことが分かる。従って, この構造記述のままでさまざまな形状の車体や窓を持ったものが「バス」として認識されることになる。なお, この記述を生成するのに 42.5cpu 秒を要した。

7 おわりに

図形形で与えられた例から構造記述を学習するシステムについて述べた。システムに提示する図形による「例」の記述は, オブジェクトの空間性を保持した柔軟なものになっている。また構造記述をその複雑さの観点から分類し, それに従った構造記述生

成向きの学習アルゴリズムを用いている。作成された構造記述ルールは, 画像理解のための認識ルールとして利用することができる。本システムによって, 構造記述生成のための手間が大きく削減されることが期待される。

しかしながら, 本文中でも述べたように, 今回のシステムで作成できる構造記述はその複雑さの点でまだ不十分なところがあり, より高度な対象の構造記述まで生成できるようにするには, まずこの部分を拡充する必要がある。

参考文献

- [1] P. H. Winston, "Learning Structural Descriptions from Examples". in P. H. Winston eds., "The Psychology of Computer Vision". McGraw-Hill, 1975. (邦訳: 白井良明, 杉原厚吉訳, "コンピュータビジョンの心理". 産業図書, 1979.)
- [2] T. G. Dietterich and R. S. Michalski, "A Comparative Review of Selected Methods for Learning From Examples". in R. S. Michalski, J. G. Carbonell and T. M. Mitchell eds., "Machine Learning". Springer-Verlag, 1983.
- [3] C. Sammut and R. B. Banerji, "Learning Concepts by Asking Questions". in R. S. Michalski, J. G. Carbonell and T. M. Mitchell eds., "Machine Learning volume II". Morgan Kaufmann, 1986.
- [4] E. Y. Shapiro, "Inductive Inference of Theories From Facts". TR192, Yale University, Dept. of Computer Science, 1981. (邦訳: "知識の帰納的推論". 共立出版, 1986.)
- [5] 大沢 裕, 坂内正夫, "画像データベース". 昭晃堂, 1987.
- [6] S. Satoh, Y. Ohsawa and M. Sakauchi, "Drawing image understanding framework using state transition models". Proc. of 10th ICPR, pp. 491-495, IEEE, 1990.