

第 6 章

階層構造を持つ認識・理解機構による画像理解

6.1 概要

近年、映像情報の高度利用の要求はますます高まっており、こうした映像情報をより高度に概念化し、高度利用が可能なデータモデルを構築できる画像認識・理解システムへの期待が高まっている。2章ではこうした画像認識・理解技術の高度化についても考察を行ったが、中でもモデルベースの手法と、異なる抽象レベルの処理間での協調を利用した手法とが特に有望であると考えられる。モデルベースの手法を用いた画像理解システムとしては ACRONYM[4] などがあげられる。この手法では、モデルを階層的に与えるようにして、高い記述力をもってモデルを作成することができるようにしておくことが重要である。異なる抽象レベルの処理間での協調を利用したシステムとしては、IGS[66]、SIGMA[27-29]、MIRACLE-IV[25, 26, 43] などがあげられる。IGS は意味的なレベルからのフィードバックを用いてセグメンテーションをより高度に行うシステムである。より高次の認識・理解処理を行うシステムとしては後の2つがあげられ、SIGMA は航空画像の認識を行うシステムであり、MIRACLE-IV はある対象物のシルエット画像を与えられてその機能的特徴を学習を用いて抽出するシステムである。この両システムは、異なる抽象レベルで動作するシステムの間の比較的疎な協調処理により画像認識を実現しているが、より高度な認識プロセスの実現のためにはこれらの処理間でのより密な協調処理が望ましいと考えられる。

3章では、さまざまな対象に対応可能な画像理解システムのための機構として状態遷移モデルについて述べた。この状態遷移モデルに基づく画像理解システムは、性質の異なったさまざまな対象について、システムに与える認識ルールの再構成で対応することができる。これを実際に各種の図面や一般画像などに適用し、ある程度の実用性について確認した。しかしながらこれをより複雑な対象に適用し、より高度な処理結果を得ようとした場合、現状ではオブジェクトのグルーピングを扱う能力の不十分さや、階層的に表現されたモデルとのマッチング能力の不十分さなどから、満足の行く結果が得られないことがあった。

本章では、より高度な画像理解プロセスを実現するため、階層的なモデルとの照合が可能であり、異なる抽象レベルの処理間での協調が行えるような多階層認識理解モデルを考え、状態遷移モデルを拡張してこれを実現する多階層状態遷移モデルについて述べる [52]。

従来の状態遷移モデルでは、対象のイメージの構成要素のシンボル(トークン)を並列に評価することにより「部分間での協調」を用いて認識プロセスを実現していた。これにより、画像データにおける平面的な局所性を生かした認識を行うことができたが、トークンをグルーピングしたトークン群の間の関係のようなより大局的な状況が扱えなかった。そこで多階層状態遷移モデルでは、ある一定の規則に従ってトークンをグルーピングしたものをもとのトークンとは別の階層に属する新たなトークンとして扱い、これらの階層関係にあるトークン間でも相互作用させることにより「部分と全体との協調」を実現し、イメージの細かい状況を調べつつより大局的な関係も扱える認識機構を提供している。これらの階層は互いに抽象度の異なるトークンが存在していると考えられるが、この抽象度階層により階層的に表現されたモデルとの照合にもうまく対応できる。多階層状態遷移モデルでは、これらの抽象度の異なる協調動作を統一的に実現できる共通の枠組を提供している。

以下、多階層状態遷移モデルの概要について述べ、実験システムとして線画の解釈システムと一般画像からの人物の抽出システムを紹介し、その有効性と実用性を検証する。

6.2 多階層状態遷移モデル

6.2.1 多階層認識理解モデル

異なる抽象レベルの処理間での協調を実現するための多階層型の認識理解モデルについて述べる。その説明のため、まず画像理解プロセスにおいて有効である相互作用としての協調動作のあり方について考える。このような協調動作として、以下のような2種類のものが考えられる。

部分間での協調 画像データは一般に平面的な局所性が強い。このため、画像理解プロセスにおいて画像上のある部分の解釈はそこごく周囲の部分の状況を参照することにより決定できる場合が多い。この周辺どうしの部分間の相互作用を「部分間の協調」と呼ぶ。

部分と全体の協調 画像理解プロセスでは、相対的な部分と全体の関係において「部分が分からなければ全体が分からない、全体が分からなければ部分が分からない」となってしまふことが本質的な問題であると考えられる。これを解決するためには、部分の解釈と全体の解釈とを相互作用させつつ画像理解プロセスを進める必要がある。これを「部分と全体の協調」と呼ぶ。

従来の画像認識のアプローチでは、部分間の協調はかなり積極的に利用されてきたと考えられるが、部分と全体の協調の方はあまり利用されていないように思われる。部分間の協調は、画像理解プロセスにおいて、ある部分の解釈を行うのにその周辺部分の状況を参照している場合、まさにこれが行われていると考えられ、画像理解プロセスではごく普通に行われている操作である。この参照は相互に行われるので、画像データ中の各部分を少しずつ均等に処理する必要がある。状態遷移モデルにおけるボトムアッププロセスでは、画像データの平面的な局所性に着目し、この部分間の協調を並列に同時進行させることにより効率良く実現していた。

部分と全体の協調処理は、通常はボトムアップ的な処理により部分からこれにより構成される部分を作成したり、トップダウン的な処理により全体を構成するようなモデルを立ててこれを部分により構成するように照合を取ったりすることにより行う。この場合、いずれもその作用の及び方は一方向のみであり、「相互作用」とは言い難い。逆方向の作用は、例えば仮説検証のような形で行われる。すなわち、まずモデルを立ててこれを部分により構成されるように、各部分について仮説を立てる。これはいわば全体から部分への作用である。次いでこの各部分がモデルを構成し得るような条件を満たしているかどうか検証される。これは部分から全体(モデル)への作用といえる。このような、非常に制限された形での部分と全体との協調の実現が主であった。状態遷移型の認識理解システムでも、この部分と全体の協調はトップダウンプロセスにおけるバックトラックにより消極的に行うのみであり、より密な相互作用は実現されていなかった。

この部分と全体との関係は、部分により全体が構成されているという意味で通常全体の方が部分よりも抽象レベルが高いと考えられる。そこでこの部分と全体の協調を異なる抽

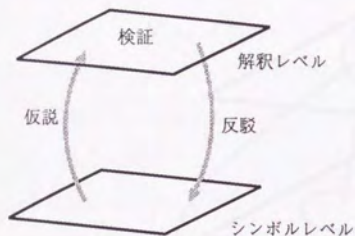
象レベルの処理間の協調として考える。この異なる抽象レベルの処理間の協調動作を模式的に表したものを図 6.1 に示す。図 6.1(a) は通常の仮説検証による協調を表しており、セグメント (シンボル) の抽象レベルから意味を持つ解釈の抽象レベルに仮説を立て、シンボルレベルで検証を行い、これが誤っていたら解釈レベルの仮説を反駁するようになっている。この場合、シンボルレベルと解釈レベルの間の相互作用が疎であることが分かる。

図 6.1(b) は IGS の場合を示している。画像データの抽象レベルでセグメントを生成すると、セグメントの抽象レベルで各セグメントについて簡単な解釈を行い、これに基づいてセグメント間での拘束条件のチェックを行って、この結果を画像レベルでのセグメンテーション処理に反映させるようになっている。異なる抽象レベル間での協調という視点に立つと、このシステムではかなり密な協調処理が実現できていると考えられる。

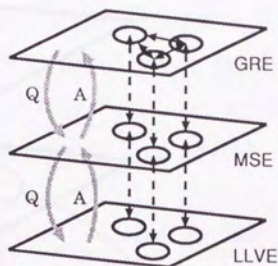
図 6.1(c) は SIGMA の場合を表している。SIGMA では GRE, MSE, LLVE という 3 つのシステムがそれぞれ別の抽象度の処理を受け持ち、このシステム間の協調処理により異なる抽象レベル間の協調を実現している。しかしながらこのシステムでは、上位のシステムと下位のシステムとの間の相互作用としては、上位から下位への問い合わせと下位から上位への答とによるもののみであり、あまり密な協調動作は行えない。

図 6.1(d) は MIRACLE-IV の場合を示している。このシステムは対象物の機能的側面を扱うモデル獲得部と画像データを扱う処理戦略部とからなっており、これらの間の協調処理により異なる抽象レベル間での協調を実現している。機能面を扱うレベルから画像レベルへの作用は、ある機能的特徴と対応した視覚的特徴の所定の場所での抽出要求としてなされる。画像レベルの処理を行う処理戦略部はこの視覚特徴の抽出過程を学習していくことができるので、やがてさまざまな視覚特徴を自分から抽出することができるようになる。画像レベルから機能面を扱うレベルへの作用は、こうして画像レベルから得られた視覚的特徴を機能面を扱うレベルへと伝えることにより実現されることになる。こうして、異なる 2 つのレベルの間で相互に作用を及ぼすことができる。

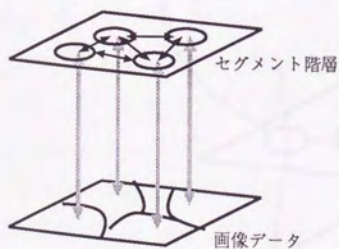
ここで、本章で述べる多階層認識理解モデルについて考える。この認識理解モデルでの異なる抽象レベル間での協調動作の様子を図 6.2 に示す。この図に示す通りより多くの抽象レベルを考え、それぞれが階層をなすと考えている。この層間での相互作用により異なる抽象



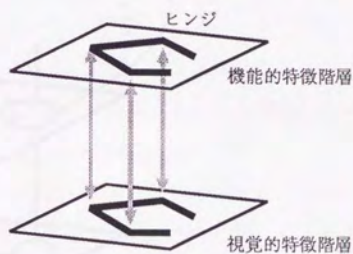
(a) 仮説検証型システム



(c) SIGMA



(b) IGS



(d) MIRACLE-IV

図 6.1: 異なる抽象レベル間の協調処理

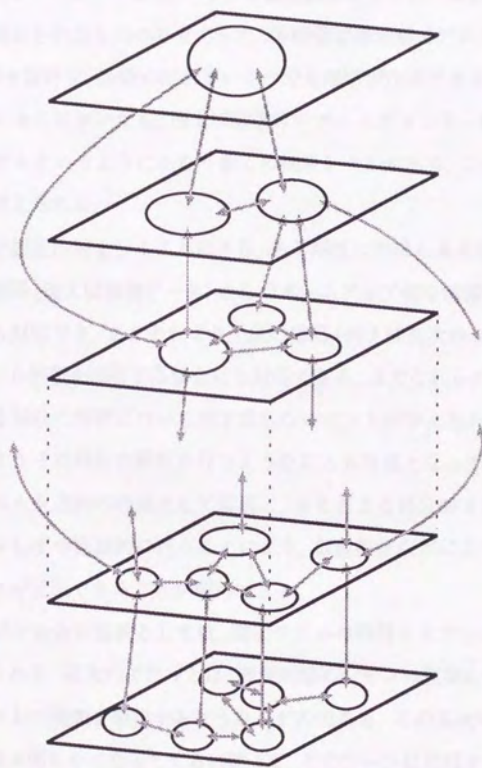


図 6.2: 多階層認識理解モデルにおける階層間の協調

レベル間の協調動作を実現する。また各層の内部でも部分間の協調を行えるような構成となっている。この両者を並列に同時進行させることにより密な協調処理を実現することができ、これによりさらに高度な認識プロセスを実現できると考えている。

先に述べたいくつかのシステムは、いずれも協調動作を行うレベルとしては2ないし3程度のあらかじめ固定されたもののみであった。多階層認識理解モデルでは、このレベル(階層)の数には制限を設けず、必要に応じていくつでも階層が作成できるようにしている。実際の画像認識プロセスにおいても、例えば画像データ—セグメント—幾何学的図形—構成部品—高次のモデルというようになり多くの抽象レベルがあり、これらの間の相互作用は有効であると考えられる。

多くの階層間で相互作用を行うことにより、ある階層におけるある部分の解釈について、それより下位の階層(例えば画像データ)からのボトムアップ的な情報からその部分の解釈ができる場合にも対応でき、またそれより上位の階層(例えば高次のモデル)からのトップダウン的な情報から解釈が進行する場合にも対応できる。またこれらの両者から、どちらも不完全ながらある部分の解釈について相矛盾しないヒントが与えられ、これらの情報を総合することによりその部分の解釈を行うようなことも可能となってくる。またこれらの協調関係をいずれも双方向の作用として実現し、さまざまな部分のさまざまな抽象レベルにおける解釈を少しずつ協調的に行うことにより、仮説検証などによるよりも効率の良い画像理解プロセスが実現できることが期待できる。

特に有効な部分と全体の協調としては、高次モデルの利用とサブシンボルの利用による画像理解が考えられる。高次のモデルとは、認識対象のシーンの種類によりそのシーンに出現するオブジェクトの種類を限定するようなモデルである。この高次モデルの利用により、シーンの種類が段々明らかになってくるに従って、そのシーンに出現すると考えられるオブジェクトのモデルの種類が減少してくることになり、モデル照合の効率が向上することになる。サブシンボルとは、通常の長方形や円などといったシンボルより画像レベルよりの情報であり、線分などのシンボルで表されているが、それからもとの画像データにかなり近い情報を取り出すことができるような記述である。図面データの場合の輪郭線や、一般画像の場合のゼロクロス情報などがこれにあたることになる。

従来の一般的な画像認識のアプローチでは、部分間の協調はかなり積極的に利用されてきたと考えられるが、部分と全体の協調の方はあまり利用されていない場合が多い。部分と全体の協調が行われているものもあるが、大抵は個別の対応により高次モデルの利用のみができる、あるいはサブシンボルの利用のみができるというような構成であった。多階層認識理解モデルでは、さまざまな抽象レベルにおける協調を可能にする枠組を与えることを目標としており、より高い抽象レベルでの協調である高次モデルの利用や、より低い抽象レベルでの協調であるサブシンボルの利用などを統一的に実現する。

6.2.2 多階層状態遷移モデルに基づく画像理解システムの構成

多階層認識理解モデルの実現として、状態遷移モデルの拡張による多階層状態遷移モデルについて述べる。従来の状態遷移モデルでは、実際の認識対象の画像と1対1に対応する形で平面上にトークンが存在し、それらの間の相互作用による状態遷移で画像理解プロセスを実現していた。多階層状態遷移モデルでは、複数の階層上にそれぞれ例えば部分と全体の関係をなすような形でトークンが存在し、それらが同じ階層上でも異なる階層間でも同様に相互作用することにより状態遷移を行い、総体として画像理解プロセスを実現するように動作する。このようにして多階層認識理解モデルの実現を行う。

この考え方に基づく、多階層状態遷移モデルに従った画像理解システムの構成を図6.3に示す。処理の流れは、大きく2つのフェーズに分けて考えられる。

- 大まかなグルーピング処理により抽象度階層(初期階層)を生成する。
- 各階層での状態遷移により認識プロセスを進める。

この方法では、グルーピングベースの抽象度階層を実際の認識プロセスの開始前に生成することになるが、これは大まかなグルーピングは各部分の認識を行わなくともできるという仮定に基づいている。一般には、例えば異なる σ のガウシアンフィルタで階層的にはかした画像の解析や、動画像における動きの解析などから大域的なグルーピングはある程度得られるので、この仮定は妥当であると考えられる。ただし、このグルーピングに誤りが含ま

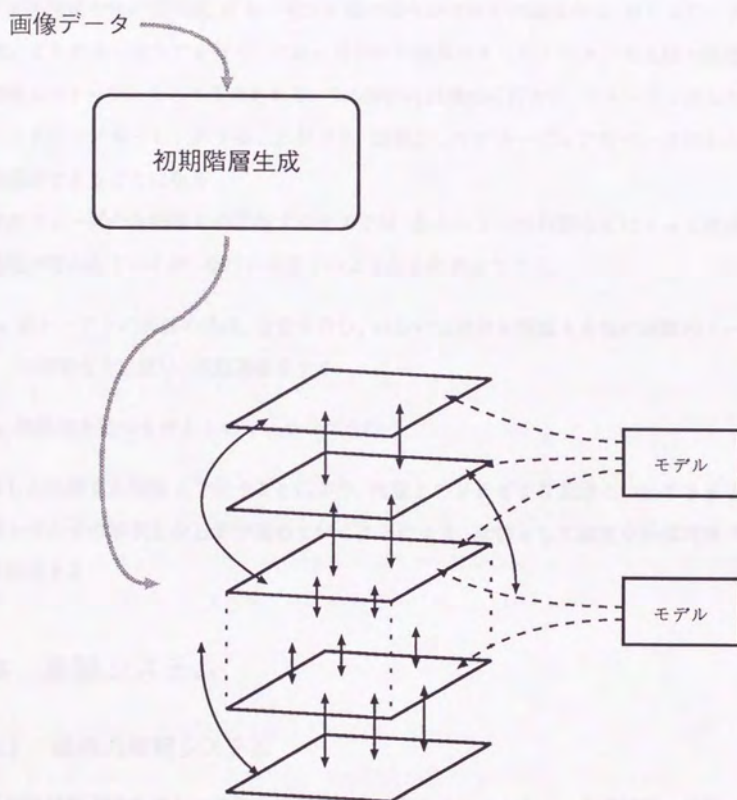


図 6.3: 多階層状態遷移モデルによる画像理解システムの構成

れることは十分考えられるので、後に認識が進んで行った段階でこれを修正するような機構が必要である。

このグルーピングの具体的な方法であるが、はじめのフェーズの抽象度階層の生成では、まず隣接関係や色の類似度、グルーピング後の滑らかさなどの観点から、ボトムアップ的な処理によるグルーピングを行う。グルーピングの結果はもとのトークンの上位の階層に別の階層上のトークンとして生成される。この操作は再帰的に行われ、グルーピングしたトークンもさらにグルーピングすることができ、結果としてグルーピングをベースにした抽象度階層ができることになる。

次のフェーズの各階層での認識プロセスでは、各トークンの状態などによって適宜適切な処理が行われていくが、それには以下のようなものがありうる。

- 各トークンの周囲の状況、自分を含む、あるいは自分を構成する他の階層のトークンの状態などに従い、状態遷移をする。
- 階層性を持つモデルとのマッチングを行う。

こうした処理を各階層上で行うことにより、画像上のさまざまな部分についてさまざまな抽象レベルでの解釈を少しずつ進めていくことにより、全体として高度な画像理解プロセスを実現する。

6.3 実験システム

6.3.1 線画の解釈システム

多階層状態遷移モデルの考え方に基いた実験システムとして、まず線画の解釈システムについて述べる。対象となる線画は図 6.4 のようなものである。システムは、この線画を構成する各オブジェクトを、モデルとの照合を用いて識別することができる。

システムへ入力される線画は、折れ線近似などの手法を用いて線分による記述に変換される。この記述を用いてまずグルーピングによる初期階層を生成し、次いでこの階層間での協

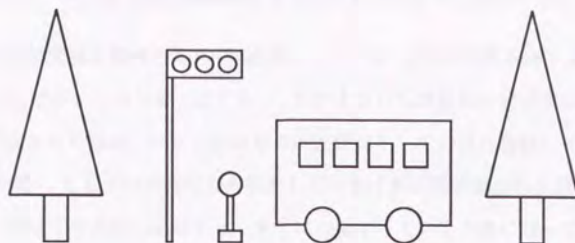


図 6.4: 線画によるシーンの例

調処理とモデルとの照合により認識処理を行う。モデルとの照合は、対象の線画を基本的なシンボルによる記述に変換したものをを用いて行うが、このシンボル化そのものが完全に行えるとは限らないため、この照合過程において、適宜サブシンボル情報を参照することによりシンボルそのものを修正することができる。また、認識できる対象の種類の制限をなるべくなくし、森羅万象を含むシーンの理解を指向するため、対象のシーンに存在するオブジェクトのモデルの種類を制限せず、高次モデルとしてシーンの種類を表す階層の利用により、シーンに存在し得るモデルの種類を徐々に絞り込んでいく過程を実現している。

システムのシーン理解の過程を明らかにするため、初期階層の生成、サブシンボルの利用を伴うモデルとの照合、高次のモデルの利用のそれぞれについて述べる。

初期階層の生成

認識対象の線図形は、折れ線近似などの手法を用いることにより、線分による記述としてシステムに与えられる。この記述を用いた初期階層作成の基本的な流れは、以下のように示される。

1. 線分の接続関係を調べる。
2. 求めた接続関係をベースにオブジェクトを生成する。
3. 隣接関係、包含関係などから各オブジェクトをグルーピングする。

4. 各オブジェクトと基本的な幾何学オブジェクトのモデルとの照合を行う。

まず線分の接続関係を調べるため、各線分トークンは、自分の両端点に「点」という幾何学特性を持った端点トークンを生成する。これがすなわち線分間の接続関係を表すトークンとなる。生成された端点トークンは自分の周囲状況として自分に接続している線分トークンの状況を調べ、もし1つの線分しか接続していなければ開放端点の状態に、2つの線分が接続していればその角度に応じてまっすぐにつながっているか角になっているかあるいは単に接続しているのかなどを判断し、適当な状態に遷移すると同時にその様子を接続している線分に知らせる。この結果を受けて線分トークンはその接続の様子を取り込んだ状態に遷移し、自分と他の線分との接続状況の分かっている線分トークンの状態となる。

このようにして生成した接続関係をもとにして、暫定的なオブジェクトトークンを生成する。同一の端点トークンを共有している線分同士は互いに接続関係にあると考える。接続関係にある線分が同一のオブジェクトを構成するように、この接続関係に基づくオブジェクトを生成する。処理としては、任意の線分トークンからその接続関係を順次たどっていくことによって求めるようになっていく。こうして求めた線分トークンの集合を内部状態に持つオブジェクトトークン(\$obj) が生成され、各線分トークンにはそのオブジェクトトークンの構成要素になったという情報がブロードキャストされる。これによりオブジェクトトークンとこれを構成する線分トークンは互いに部分と全体の関係にあるということが明示された状態として得られる。

求められたオブジェクトトークンに対し、それらの間の基本的な空間的な関係をもとにしたグルーピングを行う。この空間的な関係としては、認知心理学などの知見から得られているようなものが望ましいと考えられるが、ここでは簡単のため大きさのそろったオブジェクトの隣接関係と包含関係とからグルーピングを行うこととする。隣接関係の調査は、各オブジェクトが自分の比較的近くに存在する他のオブジェクトと相互に通信し合い、そのサイズがある一定基準より近いもの同士を隣接関係にあるとすることにより行う。そして互いに隣接関係にあるオブジェクト群をグルーピングし、隣接関係を表すトークンを生成する。包含関係はオブジェクトトークンおよび隣接関係でグルーピングしたトークンのうちのあ

る2つの間で定義される。この包含関係にあるトークンの対をグルーピングし、包含関係を表すトークンを生成する。

ついで各オブジェクトトークンについて、あらかじめ持っている基本的な幾何学的オブジェクトのモデルと照合し、これらがこうした幾何学図形であるかどうか調べられる。この基本的な幾何学図形としては、三角形、長方形、円の3種類が用意されている。ここでは実験システムのためこの3種類のモデルにとどめているが、これらはもちろん拡張可能である。こうしたモデルとの照合に成功すると、オブジェクトトークンはその幾何学的図形を表す状態に移移する。この照合過程は基本的にトークンの状態遷移により実現されるので、もし照合に失敗してもその途中結果として不完全に照合された状態で得られることになる。例えば長方形モデルとの照合に失敗しても、これを構成する4本の辺のうち3本までは正しく抽出できているような場合、これらの結果はそのまま保存され、オブジェクトトークンも長方形になり得る状態として得られる。

以上のような一連の処理で線図形の理解のための初期階層ができる。この処理により生成した初期階層の例を図6.5に示す。これは図6.4におけるバスの部分から生成したものであり、窓の部分は正しくそれぞれ一つのオブジェクトとして得られているが、車体の部分は車輪により一部分隠されてしまっているため、この車輪と融合した一つのオブジェクトとして得られてしまっている。また、各部分ともそれぞれのサブシンボルとして、対応する線分が参照可能な形で得られるようになっている。

モデルとの照合とサブシンボルの利用

認識対象のシーンを構成する各オブジェクトについて、適当なモデルが与えられると、システムは適宜サブシンボル情報を参照しながら、対象とこのモデルとの照合を行う。その機構について述べる。

基本的なモデル照合過程は、以下に示すような流れに従って行われる。

1. 対象とモデルとの構造に着目し、不完全さを許して照合を試みる。
2. 細部に至るまで完全に照合が行えたら、このモデルとの照合に成功したとして終了

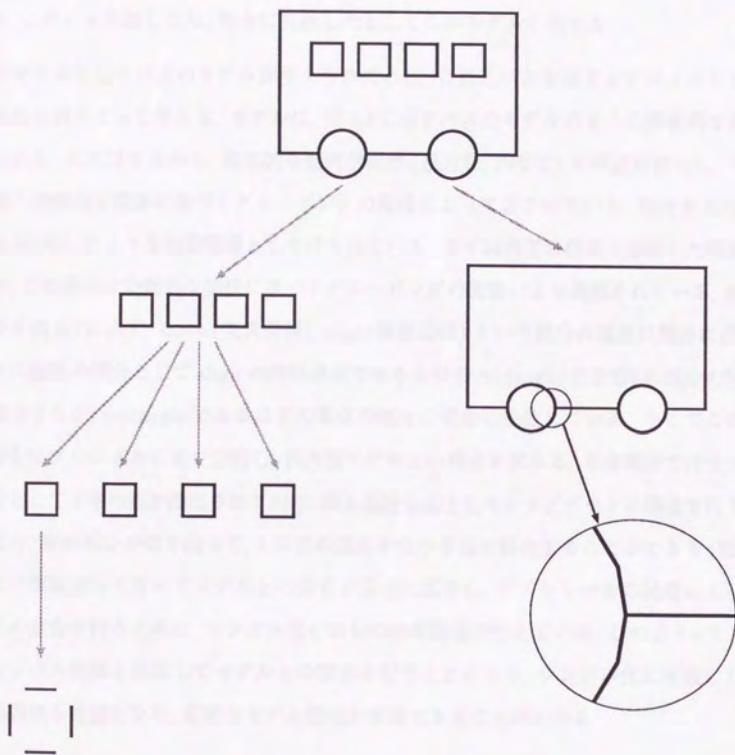


図 6.5: バスの線画による初期階層

する。

3. 大まかな構造の照合に成功したが一部分の照合に失敗した場合、その部分のサブシンボルを参照し、モデルの対応する部分(サブモデル)との照合を再帰的に行う。
4. これにも失敗したら、照合に失敗したとしてこのモデルを捨てる。

今モデルとしてバスのモデルが与えられたとし、これとバスを表すオブジェクトとの照合過程を例にとって考える。モデルは、図 6.6 に示すバスのモデルのように階層的な形で与えられる。これはすなわち、基本的な幾何学図形(長方形、円など)を構成要素とし、それらの間の空間的な関係に基づくグルーピングの階層によって表されている。照合する対象は、図 6.5 に示したような初期階層として得られている。まずは両者の構造に着目した照合を行うが、この構造は空間的な関係に基づくグルーピングの階層により表現されている。照合の様子を図 6.7 に示す。inside(包含関係)、align(隣接関係)という部分の構造に照合に成功し、さらに細部の照合として align の内部構造である 4 つの rectangle(長方形)の部分の照合にも成功するが、rectangle であるはずの車体の部分の照合に失敗している。そこでこの車体部分をサブシンボルにまで分解し、長方形モデルとの照合を試みる。車体部分ではサブシンボルとして 3 本の辺が抽出されており、残る部分も辺としてトップダウンに照合を行うことにより、車輪部分が切り出せて、4 本目の辺もかなり妥当に照合することができる(図 6.8)。

この車輪部分も含めてモデルとの照合が完全に成功し、サブシンボルの利用によりモデルとの照合を行うと共に、シンボル化そのものの再調整が行えている。このようにして、サブシンボル情報を利用してモデルとの照合を行うことにより、シンボル化に失敗した部分の再調整も可能となり、柔軟なモデル照合が実現できることがわかる。

高次モデルの利用

対象のモデルさえ与えられれば柔軟にその照合が実現できることを示したが、一般に森羅万象を含むシーンを理解しようとした場合、そのシーンに存在し得るモデルの種類は無数にあり得ることになる。そのシーンの理解において、シーンに存在するオブジェクトとこうしたモデルとを逐一照合していくわけにはいかない。そこでここでは、シーンの種類によ

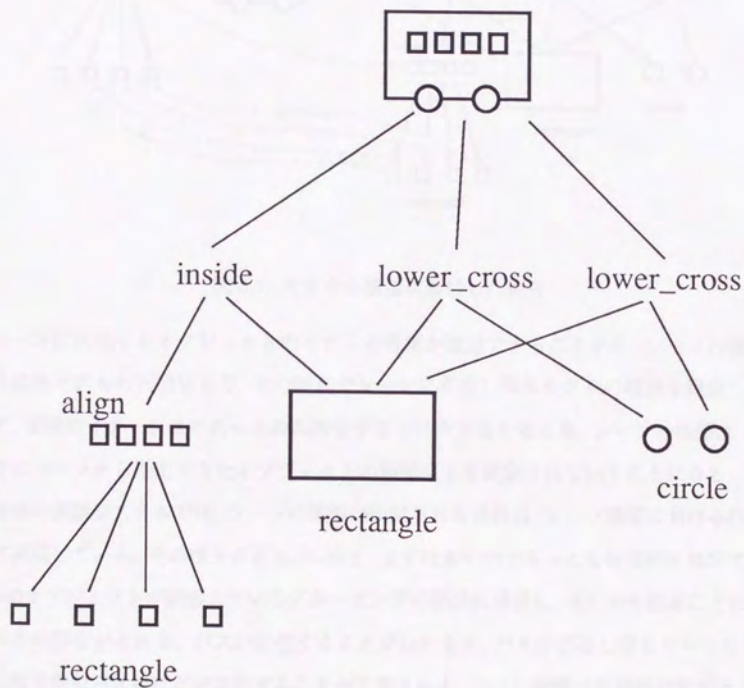


図 6.6: バスのモデル

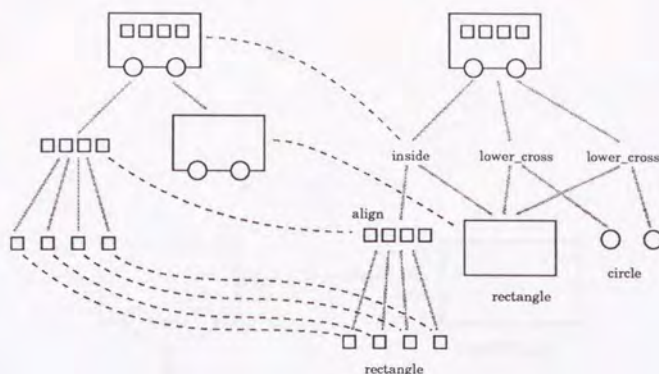


図 6.7: 大まかな構造に着目した照合

でシーンに出現するオブジェクトのモデルの種類が規定できることから、シーンの種類という高次モデルの利用により、その時点でシーンに存在し得るモデルの種類を限定しておいて、実際のイメージはそれらとのみ照合するという方法を考える。シーンの種類は、それまでにシーンから抽出できたオブジェクトの種類により限定されていくことになる。

今回の実験システムでは、シーンの種類が限定される過程は、シーン階層における状態遷移で実現している。その様子を図 6.9 に示す。まず対象の中でもっとも特徴的な部分である 4 つのオブジェクトが隣接しているグルーピングの部分に着目し、モデルを限ることによりバスとの照合がとれる。バスが存在することがわかると、バスが存在し得るシーンとして、他に信号機やバス停などが存在することが予想される。シーン階層での状態遷移では、このようにしてモデルの絞り込みを実現している。

こうして本システムは、図 6.4 の線図形に含まれるすべてのオブジェクトを、それぞれ対応するモデルとの照合をとることにより、正しく同定することができた。その際、サブシンボルの利用により、シンボル化に失敗した部分についても、モデルとの照合によりトップダウンに得られる情報からシンボル化そのものを再調整して正しいシンボルが得られている。

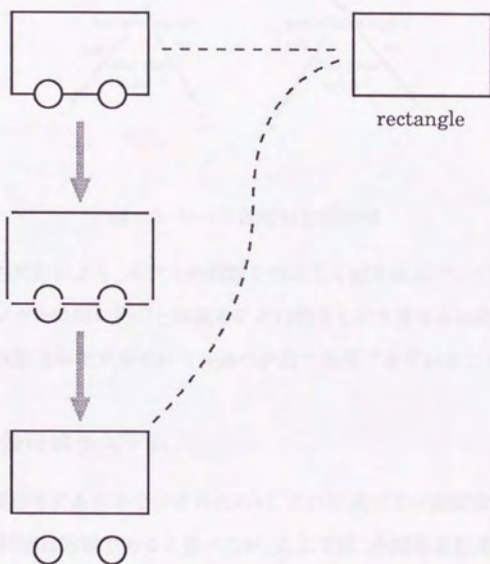


図 6.8: 照合におけるサブシンボルの利用

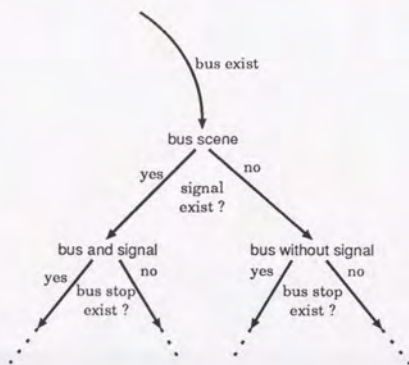


図 6.9: シーン階層の状態遷移

また、高次モデルの利用により、モデルの種類を効率良く絞り込んでいく過程が実現されている。このサブシンボル情報の利用と高次モデルの利用という異なる抽象度の情報を利用した処理が、多階層状態遷移モデルという共通の枠組で実現できていることが示されている。

6.3.2 一般画像理解システム

5 章では、状態遷移モデルの不十分さのために、これに基づく一般画像認識システムでは画像からの人物の抽出は困難であると述べたが、ここでは、多階層状態遷移モデルに基づく構成による一般画像からの人物の抽出を行う実験システムについて述べる。

図 6.10 に対象となる一般画像の例を示す。システムはほぼ先の線図形理解システムと同様の構成となっており、与えられた画像からまず初期階層を生成してから、モデルとの照合などを行って認識処理を行う。初期階層としては、一般の画像からシンボル記述を生成する必要があるため、ゼロクロス画像から生成した線分を利用している。また、ここでのシステムの目的が人物の抽出と定まっているため、モデルとしては人物のモデルのみとなり、システムはこの人物モデルと対象画像との照合を行うことになる。以下、その各ステップについ



図 6.10: 対象となる一般画像

て述べた上で、このシステムをスポーツシーンの解釈システムに拡張する場合の高次モデルの利用の可能性について検討する。

初期階層の生成

与えられた一般画像から初期階層を生成する手順は以下のように示される。

- Marr-Hildreth のオペレータ [24] の利用によりもとの画像からゼロクロスを得る。
- これらのうちからグラジエントの大きいもののみを選択し、折れ線近似により線分を生成する。
- これを異なる σ で複数作成する。

こうして何種類かのゼロクロスを表す線分を作成し、これらを線分トークンとしてそれぞれ別の階層に属するものとするにより基本的な初期階層とする。

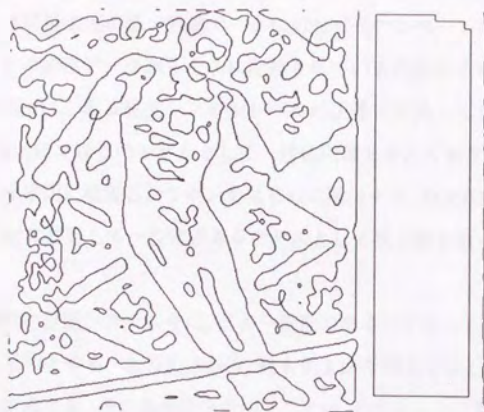
Marr-Hildreth のオペレータは $\nabla^2 G$ で表される。 ∇^2 はラプラス演算子、 G はガウシアンフィルタであり、それぞれ

$$\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \quad (6.1)$$

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (6.2)$$

と表すことができる。 σ というパラメータによってそのぼかしの度合いが調整できる。 σ を大きくすることによりぼかしを強くして対象の大まかな構造を表す記述が生成され、小さい値の σ によって対象のより細かい構造を表しうる記述が生成される。これらを協調的に用いることにより、対象の大まかな構造と細かい構造とを適宜表現しているような強力なモデルの獲得が可能となる。

$\sigma=5$ 画素とした時、図 6.10 から生成したゼロクロスと線分トークンの例を図 6.11 に示す。実際にはこの線分トークンを $\sigma=1, 3, 5$ 画素の 3 種類のオペレータで作成し、それぞれ異なる階層に属するトークンとすることにより、対象の大まかな構造と細かな構造の両方を扱えるように考慮している。



(a) ゼロクロス画像



(b) 線分トークン

図 6.11: $\sigma = 5$ 画素の Marr-Hildreth のオペレータにより生成したゼロクロスと線分トークン

ついで、得られた線分トークンから、もう少し抽象度の高いシンボルを抽出する処理を行う。一般画像のように抽象度の低い画像データに対してモデルベースなどの手法で認識処理を行う場合、もとの画像データからそこに記述されている内容によらず一般性のあるモデルをあらかじめ抽出し、ある程度シンボルレベルの記述に変換してしまうという必要がある[72]。ACRONYM[4]ではこのモデルとして一般化円錐を考慮しており、画像データからはこのモデルを2次元平面に投影したりボンをはじめに抽出する。SCERPO[22]では、もとの画像中の並行線分対に着目して一般性のあるモデルとして長方形を取り出すようになっている。

ここでの一般画像の認識システムでは、この一般性のあるモデルとしてSCERPOと同様に長方形を用いるものとする。この長方形は、最も大まかな構造を表し得る $\sigma=5$ 画素の時のゼロクロスから生成する。その具体的な方法としては、線分トークン同士の相互作用を用いて互いに並行で両者ともある程度の長さを持ち長方形を構成し得るものをふるい分けることにより行っている。こうして生成した長方形トークンの例を図6.12に示す。

こうして抽出された長方形シンボルは長方形トークンとして新たな階層に作成される。この長方形トークンの形状などは固定的なものではなく、後のより上位のモデル(すなわち人間モデル)との照合や下位の階層との相互作用などにより柔軟に変化し得るようになっている。

人間モデルとの照合とサブシンボルの利用

認識したい対象のモデルは、ここまでで得られているトークンにより構成することになる。従って、人間のモデルを長方形を使って作成する必要がある。モデルのより柔軟性の高い記述としては、ACRONYMのモデルのように対象を構成する一般的モデル(ACRONYMの場合は一般化円錐)と、それらの間の代数的な拘束条件とで記述するのが良い。今回のシステムでは、簡単のためこのような一般性の高い記述ではなく人間のモデルとしてかなり固定的に作成している。しかしながらこれを一般性の高いモデルの記述を扱えるように変更するのは可能であろう。

長方形による人間のモデルの例を図6.13に示す。人間の構成要素である腕や足などが長



図 6.12: 生成した長方形トークンの例

($\sigma = 5$ 画素の時の線分トークンと一緒に表示している)

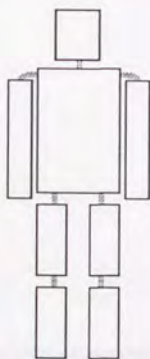


図 6.13: 長方形により構成された人間のモデル

方形で表現されている。先のバスの例と異なり、人間の場合は各部分が比較的自由に動けるようになっているため、モデルでもこの点に対処できる必要がある。より踏み込んだモデルとしては、人間の各関節の動き方までもモデル化した3次元のモデルを持っていくと人間のさまざまな動きにも対応できる強力なモデルが構成できるが、ここでは各構成要素が動き得るように接続しているかどうかだけをモデルとして持つようにした。

モデルとの照合では、線図形の理解システムの場合と同様、始めは対象とモデルの大まかな構造に着目して照合し、徐々により詳細な部分(サブシンボル)を参照してより細かい照合を行うようになっている。大まかな構造による照合が不完全であった部分には、トップダウンに長方形トークンを抽出したり長方形トークンの位置を変更したりという指示が与えられることになり、これを下位の階層のゼロクロスなどを参照することにより協調的に実現していくことになる。このような構成をとることにより、システムは人間を構成する一部分のヒントから人間モデルを立て、このモデルからのトップダウンの指示とゼロクロスからのボトムアップの情報の両面から人間の他の構成要素が段々と抽出されていくプロセスが実現できることになる。

図 6.14 に抽出された人物の例を示す。この例では、始めは頭の部分が抽出できなかった



図 6.14: 抽出された人物の例

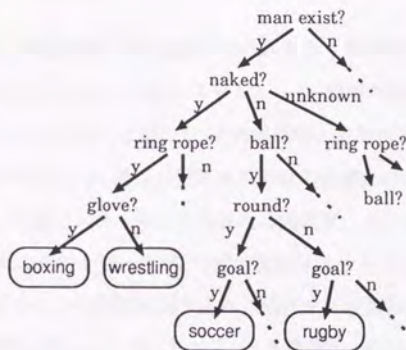


図 6.15: スポーツ画像を対象としたシーン階層の状態遷移

が、胴体や腕の部分の詳細な照合を行って首の位置を決め、頭の大体の場所を推定した上で抽出に成功している。結果としては、頭、体、腕、脚という基本的な部分の同定ができ、サブシンボルの利用によりそれらのより詳細な輪郭が抽出できた。

スポーツシーン理解システムへの拡張と高次モデルの利用

先に述べた線図形の理解の場合と同様、ここでもシーンの種類による高次モデルの利用が考えられる。ここでは対象をスポーツシーンと限り、スポーツの種類で最終的なシーンの種類を表すことを考える。

スポーツシーンには必ず人物が存在すると考えられるので、まず人物の抽出を行って人物が存在するシーンであるという状態に移る。ついでその人物の服装を見て、競技者が服を着ているかどうかでシーンの分類を行う。そしてさらに競技で用いられるはずの用具の抽出を行って、競技の種類の同定を行う。図 6.15 にシーン階層の状態遷移の様子を示す。一連の条件判断で、もし判断し切れない (unknown である) という結果が得られた場合、以降の条件判断をより詳細に行うことによりシーンの同定を行うことになることが示されている。

6.3.3 考察

異なる抽象度階層上で協調的に状態遷移プロセスを行う多階層状態遷移モデルについて述べた。ここではこのモデルに従う実験システムとして、線画の解釈システムと一般画像から人物を抽出するシステムについて紹介し、その実現性と有効性について考察した。

多階層状態遷移モデルにより、異なる抽象度における協調動作を統一的に実現できる枠組が提供されている。実験システムではこの異なる抽象度における協調としてサブシンボルの利用と高次モデルの利用を行ったが、今回の実験システムでの高次モデルの利用では、この高次モデルを表すシーンの種類の絞り込みの過程を状態遷移により決定木的に実現しているため、その拡張性に乏しいと考えられる。より網羅的・体系的にシーンの種類を表現するためには、ATMS[9-11]などの利用が有効であろうと考えられる。

第 7 章

画像データベース形成におけるコンピュータ学習の利用

7.1 概要

本論文では、実用的で高度な応用が可能な次世代画像データベースの実現のため、主としてデータモデル獲得の見地から行った研究について述べてきている。これまでに、より高度な画像データベースに必要なデータモデル獲得手法について検討し、いくつかの画像データモデル獲得システムを作成してきた。これはより高度な画像データモデル獲得を行うという視点に立ったアプローチであると考えられる。このような視点はもちろん重要であるが、この他に実用的な大規模画像データベースの構築のためには、より多様な画像データからのデータモデル獲得を可能とするために、データモデル獲得手段そのものを自動生成するという視点が重要となってくる。ここでいうデータモデル獲得手段とは、すなわちデータモデル獲得システムのことであり、これを自動生成するシステムではいわゆる「学習」が行われることになる。2章では、画像データベースシステムにおけるコンピュータ学習の利用に関するアプローチについて紹介したが、ここで言っている画像データモデル獲得手段の獲得の実現のためには、このコンピュータ学習手法の利用が必要不可欠であろう。本章では、これまでに述べた画像データモデル獲得システムに対し、コンピュータ学習の方法を用いてその作成を支援する手法について考察する。

7.2節では、4章で述べた状態遷移モデルに基づく図面認識システムと5章で述べた画像認識システムに対する、その状態遷移型認識機構におけるルールの作成支援システムについて述べる。状態遷移モデルに基づく図面／画像理解システムでは、さまざまな対象に対する認識システムの再構成はシステムに与えるルールの再構築のみで済み、そのルールの作成も比較的簡単に行えるようになっている。しかしながら、画像データベースで扱いたいような極めて膨大な種類の画像の認識システムをいちいち作成するのはやはり困難なことである。そこでこのシステムでは、マウスとグラフィックディスプレイを用いたユーザとの平易なセッションからこのルールをある程度自動生成することにより、ルールの作成作業を支援する。状態遷移モデルのルールの形式は定型的な扱い易いものであり、コンピュータ学習との親和性が高いので、システムはコンピュータ学習の手法を用いてユーザから得られた情報をもとにルールを自動生成することになる。

7.3 節では、上記の状態遷移型認識システムや、6 章で述べた多階層状態遷移モデルに基づく画像理解システムなどで利用する認識対象のモデルを、構造記述という形で自動獲得するシステムについて述べる。2 章ではモデルベースの認識手法の有効性について述べたが、同時にそのために必要なモデルの作成が簡単ではないことにも触れた。このシステムでは、獲得したいモデルの図形を表している例（正の例と呼ばれる）、あるいは表していない反例（負の例と呼ばれる）を図形的な表現で受け付け、これから推論してこのモデルの構造記述を抽出する。この図形的な表現による事例は、画像データの形で与えられることを想定しており、これを認識対象の具体的なモデルを持たない画像理解システムにより不完全ながらシンボル化したものが学習システムに与えられる。具体的には、多階層状態遷移モデルに基づく線図形理解システムにより不完全ながら構造化された記述を受けとり、これからモデルを抽出するように動作する。

これら 2 つのシステムを通じて、画像データモデル獲得におけるコンピュータ学習手法の重要性について考察する。なお、本論文の論点からはずれないようにするため、ここではこれらのシステムをあくまで画像データモデル獲得のための支援システムとして簡単に述べる。両システムの詳細については、それぞれ付録 B、付録 C に述べているので、そちらを参照されたい。

7.2 状態遷移型図面／画像理解システムのルール作成支援システム

7.2.1 システムの構成

図 7.1 に状態遷移型図面／画像理解システムのルール作成支援システムの構成を示す。システムは画像理解システムの部分と状態遷移ルール学習システムの部分により構成される。画像理解システムの部分は、これまでに述べてきた状態遷移型図面／画像理解システムである。学習システムは、グラフィックディスプレイとマウスを用い、ユーザとのセッションを繰り返すことにより、認識システムのための状態遷移ルールを生成する。

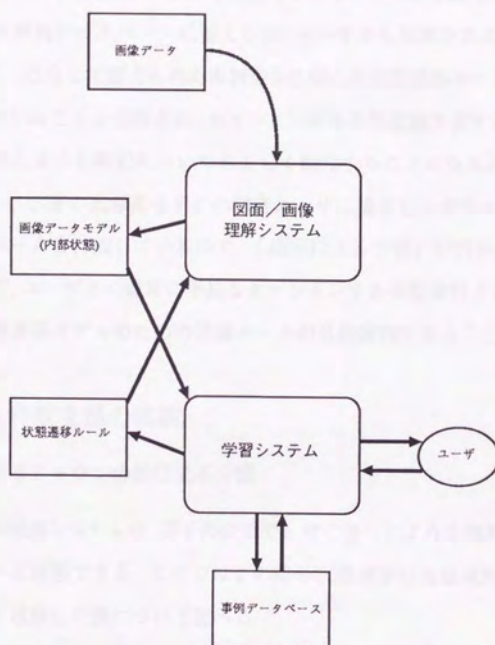


図 7.1: 状態遷移型図面／画像理解システムのルール作成支援システム

このセッションは、まず認識システムのその時点の認識結果をグラフィックディスプレイに表示し、その中の誤りについてユーザの指摘を促す。システムはユーザからの一連の認識誤りの指摘を受け、これから帰納学習することによりその指摘に矛盾しない結果を生成する状態遷移ルールを生成する。このルールで認識処理を行い、その認識結果をユーザに表示して認識誤りの指摘を受ける、というようにしてこのセッションを繰り返すことになる。一連のセッションは事例データベースに蓄えられ、ユーザから指摘された同じ誤りを繰り返すことを回避する。こうして蓄えられた事例から生成した状態遷移ルールは、ある程度一般的なものとなっていることが期待され、セッションをある程度繰り返すことにより、認識対象の他の場所の似たような局面についても正しく動作することになると考えられる。こゝで行われる学習は、学習の進み具合をその都度ユーザに提示しながらユーザからの適切な指示で段階的にルールを作成しているので、「指示による学習」が行われているといえる。

このようにして、ユーザとの非常に平易なセッションをある程度行うことにより、非常に少ない労力で状態遷移モデルのための認識ルールが自動獲得できることになる。

7.2.2 ルール作成支援の実験

カラー画像におけるトークンの色による分類

状態遷移型図面理解システムは、若干の変更で5章で述べたような簡単なカラーの一般画像認識システムへと拡張できる。ここではこの時の状態遷移型画像理解システムに本ルール作成支援手法を適用した例について述べる。

状態遷移モデルにおいて、ボトムアッププロセスの初期過程における状態遷移は、後々までかなりの影響を持つため、スケールや前後関係などの相対的な条件に左右されない情報を参照して状態遷移するのが望ましい。カラー画像の認識においては、スケールなどに左右されない情報としては色情報が考えられる。そこでここでは、状態遷移プロセスの初期過程において、色によってトークンを各色トークンに分類するルールの作成支援を考える。

各トークンには、色情報として色相(H)、明度(I)、彩度(S)が与えられているとする。この3つの数値によって色トークンの分類のための条件が生成できる。しかしながら、実際に

例えば緑色のトークンを取り出すためのルールを作成する場合、各数値がどのような範囲の時に人間は緑と感じるかは明白には分からないため、数値の範囲を少しずつ変えて状態遷移させてその結果を見ながらまた少しずつ範囲を変えてみるという、きわめて手間のかかる試行錯誤のフェーズが必要であった。ここでもしユーザがイメージを見ながら、「これは緑である」、「ここは緑ではない」という指示を与えて行くだけで自動的に緑を取り出すための条件が生成できれば、このルールの作成の時間は大きく減少するものと思われる。

実際にスポーツ画像にこの支援システムを適用し、緑色のトークン認識のためのルールを作成してみた。システムは3枚の画像から6つの正の例、8つの負の例の指示を受け、

$$\begin{aligned} & \$chain((R, G, B), H, I, S) \longrightarrow \\ & \quad -125 \leq H, H \leq -56, \\ & \quad I \leq 432, \\ & \quad 743 \leq S, \\ & \quad \$transform(\$green((R, G, B))). \end{aligned} \tag{7.1}$$

というルールを生成した。このルールは、色相が -125 度から -56 度までの範囲で、明度が 432 以下、彩度が 743 以上という条件が満たされれば緑であるということを示している(色相は -180 度から 180 度に、明度は $768(=256 \times 3)$ に、彩度は 10000 にそれぞれ正規化されている)。これは緑色としてかなり妥当な条件である。この条件により、ルール作成のもととなった3枚のイメージからはもちろんすべての緑を正しく抜き出し、その他のイメージにこれを適用しても妥当な結果が得られた。しかもユーザからのシステムへの指示は非常に簡便であった。

他の数種類の色のトークンに対しても実験を行い、いずれも妥当な結果が得られた。

地図図面における植生界を構成するトークンの抽出

地図図面の認識において、植生界認識のためのボトムアップルール作成を支援させることを考える。

まず、面積によって植生界を構成する小さい点を識別するためのルールはシステムによって簡単に生成することができた。このあと各点の近傍を調べて、ある距離以内に自分以外に2つだけ同じような点が存在するかどうかを見て、もしそうであれば点列を構成する点であるという状態に遷移させなければならない。ところがこの「ある距離」の設定が難しく、大き過ぎるとたとえ理想的な点列の状況でも点が密集している複雑な場面として認識されてしまう。また小さ過ぎるとすべての点がバラバラに見えてしまい、点列が全く抽出できなくなってしまう。ここでは後のトップダウンプロセスの効率化のため、なるべく多くの点列を正しく認識しておきたいので、この「ある距離」を適切に設定しておいてできるだけ多くの点列の局面を正しく認識するようにしたい。

今回のシステムではパラメータ計測の部分を作成する能力はなく、この「適切な距離」を自動的に生成することはできない(詳細は付録を参照されたい)。しかし、これを模擬的に行うことはできる。具体的に言うと、いくつかの距離で計測した結果をあらかじめある状態にパラメータとして持たせてしまい、後はこのルール作成支援システムによりそれらのうち適切なパラメータを選んだ適切なルールを自動生成させることにより、疑似的に適切な距離を選ばせるのである。

実際には、まず次のようなルールによって、複数の距離で近傍の状況を調べた結果をパラメータに持つ状態を作る。

\$dot \longrightarrow

\$get_near_loop(100, P₁),

\$select(P₁, \$dot, Q₁),

\$get_near_loop(105, P₂),

\$select(P₂, \$dot, Q₂),

\$get_near_loop(110, P₃),

\$select(P₃, \$dot, Q₃),

\$get_near_loop(115, P₄),

$$\begin{aligned}
& \$select(P_4, \$dot, Q_4), \\
& \$get_near_loop(120, P_5), \\
& \$select(P_5, \$dot, Q_5), \\
& \$transform(\$dot(Q_1, Q_2, Q_3, Q_4, Q_5)).
\end{aligned} \tag{7.2}$$

Q_1 から Q_5 まだが、それぞれ 100, 105, 110, 115, 120 という異なった距離をもって調べた近傍の状況であり、自分以外の $\$dot$ 状態のトークンをリスト形式に並べたものである。このルールを各トークンに適用した後、 $\$dot/5$ という状態のトークンについてそれぞれ点列を構成し得るかどうかを判断し、その結果を例としてシステムに与えればよい。実験の結果、正の例 1 つと負の例 4 つにより、もっとも望ましい結果を得ることができるルールとして次に示すようなルールが生成された。

$$\begin{aligned}
& \$dot(\neg, \neg, \neg[A, B], \neg) \longrightarrow \\
& \$transform(\$dot(A, B)).
\end{aligned} \tag{7.3}$$

すなわち、距離 115 で調べた近傍の状況について調査し、自分以外に 2 つの $\$dot$ 状態のトークンがあるというものが点列を構成し得るということを示している。よって、与えられた例からは調査すべき近傍のもっとも適切な距離は 115 であったということがわかる。

7.3 図形事例からのモデル獲得システム

7.3.1 システムの構成

図 7.2 に図形事例からのモデル獲得システムの構成を示す。入力された画像による事例は、多階層状態遷移モデルに基づく線図形理解システムにより構造化された記述として学習システムに与えられる。この記述は、対象の内部の隣接関係や包含関係などによる構造をある程度反映したものとなっており、学習システムは必要に応じてこの情報を参照してより効率良く構造化された構造記述を生成することができる。

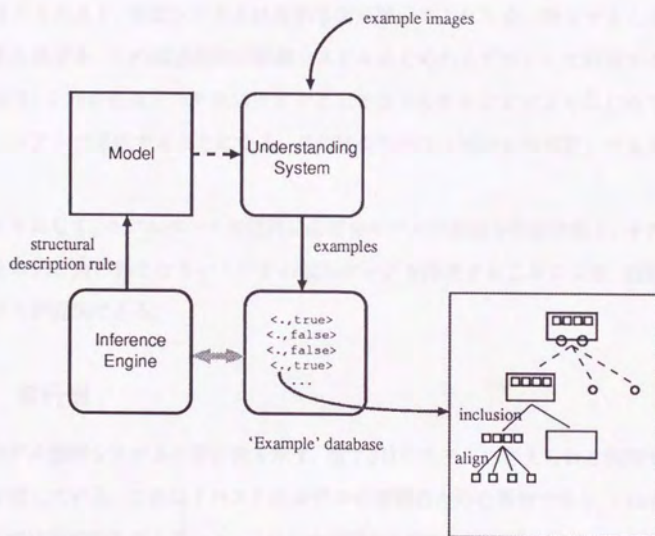


図 7.2: 図形事例からのモデル獲得システムと線図形認識システム

あるモデルを獲得したい場合、そのモデルに関する正の例と負の例を画像データの形の図形として用意し、これを線図形理解システムで処理することにより構造化された記述として学習システムに与える。この時の線図形理解システムには、基本的なプリミティブのモデルは与えられるが、入力される図形を表すようなモデルは与えられないまま、不完全に構造化された状態で学習システムにその図形の記述を渡される。この図形のモデルは学習システムにより生成されることになる。こうしてこのような不完全に構造化された事例がいくつか与えられると、学習システムは帰納学習を用いてこれらを一般化することにより構造記述を生成する。この構造記述は認識システムのためのモデルとして利用することを想定しており、これが認識システムにモデルとして与えられることによりはじめて完全な図形認識システムが完成することになる。この時の学習は「例からの学習」であると考えられる。

このようにして、モデルベースの認識に必要なモデルの複雑な作成作業を、そのモデルに関する正の例と負の例となるいくつかの図形データを用意することにより、自動的に行えるシステムが実現できる。

7.3.2 実行例

このモデル獲得システムの実行例を示す。図 7.3はシステムに与えられる図形で表現された事例を表している。これは「パス」のモデルの学習のための事例であり、1 は正の例で2 から5 までは負の例を表している。これらの事例から推論して、システムは以下に示すような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),
```

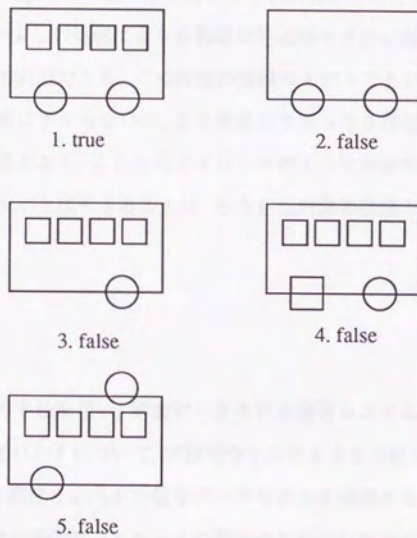


図 7.3: 学習システムに与えられる図形事例


```
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。ここで, B は車体部分, D, F は車輪部分, H は窓野部分にそれぞれ相当することになる。

この構造記述は, 今少しの処理により多階層状態遷移モデルに基づく線画解釈システムのためのモデルとして利用できる。この程度の規模のモデルであれば人間が手で作成したとしてもさほどの手間はかからないが, より大規模であったり複雑な構造を持つような対象のモデルについて考えると, ここでのアプローチのように図形的な事例を与えることによって自動的にモデルが生成できることは, そうした対象の認識システムの作成の大きな助けとなる。

7.4 考察

コンピュータ学習の手法を用い, 画像データモデル獲得システムそのものの構築を自動的に行おうとするアプローチについて, 実験的なシステムを2つ紹介した。画像データベース獲得について, より高度な応用が可能なデータモデルを獲得するという視点はもちろん重要であるが, 実用的な画像データベースの構築のためには極めて多様な画像データからのデータモデル獲得が必要であり, そのためには獲得システム自体の自動獲得という視点の重要性がかなり大きなものとなると考えられる。本章で紹介したシステムはいずれも必ずしも最終的な目的を満足していない不完全なものではあるが, 本章では, この実験的なシステムによりデータモデル獲得手法の獲得という視点の重要性を明らかにしている。今後はこれらのシステムを完成させ, 実用的な画像データベースの構築に貢献させることが課題となってこよう。

第 8 章

終章

本論文では、画像データを扱うデータベースにおいて、特にデータモデル形成の重要性に着目し、その完全自動化を実現するための諸問題について論じた。

2章では、画像データベースに関する特に重要と考えられる背景技術について述べ、実現が望まれている高度な画像データベースの実現のための指針について考察した。データベースシステムはこれで扱うデータモデルを中心としてデータモデル獲得、データモデル管理、データモデル利用という3つの機能部分からなっている。次世代画像データベースシステムとしては、データベース獲得では大規模なデータベースの構築のために完全に自動化されたものを指向し、データモデルの利用では今日の画像／マルチメディアデータに対する要求を満たす高度な応用が可能であるようなものが望まれているが、その実現のためには、これら3つの機能部分の適正な役割分担が必要である。そのための各機能部分に要求される要件について述べた。

3章では、この要件の多くを満足する新しい視点に立った画像データモデルとして状態遷移モデルに基づく画像データモデルについて述べた。状態遷移モデルに基づくデータモデルは、画像データモデル獲得過程をそのままデータモデルとしたような形となっており、データモデル獲得を特に重視したデータモデルである。これに基づく画像データモデルには以下のような特徴がある。

- 画像認識・理解機構を、内部状態を持つ計算モデルで実現している。
- 見通しの良い認識・理解機構が実現できる。
- 不完全な認識結果をデータモデルとして利用可能な形で生成することができる。
- より概念的なデータモデル(オブジェクト指向データモデル)との親和性が高い。

これらはいずれもここで考えている次世代画像データベースの実現のためには重要な性質である。

4章では、この状態遷移モデルに基づく図面理解システムについて述べた。状態遷移モデルの利用により、システムが作成した記述は認識結果としては不完全ながらデータモデルとして利用可能な形で生成できる。また、図面理解システムそのものの機構が見通しの良い

形で作成することができ、新たな処理対象にシステムを対応させるためには作成の容易な認識ルールの再構成により容易に対応することができる。

5章では、状態遷移モデルに基づく画像理解システムにより自動獲得された画像データベースを用い、これを高度に検索する画像検索システムについて述べた。状態遷移モデルにより生成された不完全に得られている画像データモデルを利用し、確からしさを落してある程度の検索結果が得られることを示した。これにより状態遷移モデルの基づくデータモデルでの不完全に認識されたデータモデルの利用の可能性について検証できた。

6章では、より高度な画像理解を目指した多階層認識理解モデルについて述べ、状態遷移モデルの拡張でこの機構を実現した多階層状態遷移モデルと、それに基づく線図形理解の実験システムと一般画像理解システムとについて述べた。多階層状態遷移モデルでは、従来の状態遷移モデルでは不十分であったグルーピングのより自然な実現と階層性を持つ高次のモデルとの照合が可能となっている。この認識理解モデルによりさらに強力な画像理解機構が実現できると考えられる。

7章では、画像データモデル獲得におけるコンピュータ学習の手法の利用方法について考察した。まず状態遷移モデルに基づく画像理解システムにおける認識ルールの作成支援システムについて述べた。グラフィックディスプレイとマウスを使った平易なセッションによりある程度実用的なルールが容易に生成できることが示された。次いでユーザから与えられた事例から画像理解のためのモデルを自動獲得するシステムについて述べた。システムには入力事例として図形が与えられ、その空間性を保持したまま大まかな構造が参照できる形で内部にとり込まれる。これを用いることによりシステムはより複雑な構造を持つような構造記述を柔軟に獲得することができる。

本論文では、高度な利用が可能な画像データベースの実現のため、データベース利用側に高い可用性を提供できる、完全に自動化されたデータモデル獲得の実現を目指し、そこに存在する問題点の解決を図った一連の研究について述べた。本論文における一連の研究は必ずしも十分な成果が得られているとは限らないが、そのアプローチは目標達成のためにはいずれも必要不可欠なものであり、本研究により一つの重要な方向性を示したものと信じる。

謝辞

本研究の遂行に当たり、問題点の所在や研究の方向性などさまざまな視点に立った御指導をいただき、暖かい御激励を下された坂内正夫教授に心から感謝致します。坂内先生には、研究上の御指導のみならず、私事にわたっても親身になって相談にのっていただき、小生の今後の人生のあり方を方向づける上で重要な指針を示していただきました。まさに汲めども尽きぬ感謝の念で一杯であります。

平成1年秋まで坂内研究室の助手を勤められた、現埼玉大助教授の大沢裕氏には、助手在職中はもとより埼玉大に移られてからも貴重な御助言を数多くいただきました。ここに感謝の意を表します。

山根淳氏には、本論文5章に関する研究に御協力をいただきました。ここに感謝致します。研究室での打ち合せなどの折りに触れ御討論、御批判をいただいた全炳東氏、龔怡虹氏をはじめとして、坂内研究室の諸氏に深謝致します。

最後に、小生の研究者としての生活に暖かい理解を示してくれた家族に感謝します。

- [1] Abiteboul, S. and Hull, R.: "Update Program in the IFO Database Model", in *Foundation of Data Organization*, pp. 319-331, 1985.
- [2] Banerjee, J., Chou, tai H., Garza, J. F., Kim, W., Woelk, D., Ballou, N. and Kim, joo H.: "Data Model Issues for Object-Oriented Applications", *ACM Transaction on Office Information Systems*, Vol. 5, No. 1, pp. 3-26, 1987.
- [3] Bentley, J. L.: "Multi Dimentional Binary Search Trees used for Associative Searching", *Commun. of the ACM*, Vol. 18, No. 9, 1975.
- [4] Brooks, R. A.: "Symbolic Reasoning Among 3-D Models and 2-D Images", *Artificial Intelligence*, Vol. 17, pp. 285-348, 1981.
- [5] Carbonell, J. G., Michalski, R. S. and Mitchell, T. M.: "An Overview of Machine Learning", in Michalski, R. S., Carbonell, J. G. and Mitchell, T. M., editors, *Machine Learning*, Springer-Verlag, 1983.
- [6] Conklin, J.: "Hypertext: An Introduction and Survey", *IEEE Computer*, Vol. 20, No. 9, 1987.
- [7] Connell, J. H. and Brady, M.: "Generating and Generalizing Models of Visual Objects", *Artificial Intelligence*, Vol. 31, pp. 159-183, 1987.
- [8] Copeland, G. and Maier, D.: "Making Smalltalk a Database System", in *proceedings of ACM SIGMOD conference*, pp. 316-325, 1984.
- [9] de Kleer, J.: "An Assumption-based TMS", *Artificial Intelligence*, Vol. 28, pp. 127-162, 1986.
- [10] de Kleer, J.: "Extending the ATMS", *Artificial Intelligence*, Vol. 28, pp. 163-196, 1986.

- [11] de Kleer, J.: "Problem Solving with the ATMS", *Artificial Intelligence*, Vol. 28, pp. 197-224, 1986.
- [12] Dietterich, T. G. and Michalski, R. S.: "A Comparative Review of Selected Methods for Learning From Examples", in Michalski, R. S., Carbonell, J. G. and Mitchell, T. M., editors, *Machine Learning*, Springer-Verlag, 1983.
- [13] Doyle, J.: "A Truth Maintenance System", *Artificial Intelligence*, Vol. 12, pp. 231-272, 1979.
- [14] 栄藤稔, 伴野明, 岸野文郎: "ステレオ輪郭像を入力とした物体の一般化円筒複合体による記述", 電子情報通信学会論文誌, Vol. J73-D-II, No. 9, 1990.
- [15] 栄藤稔, 岸野文郎: "仮説に基づくシーン記述", 電子情報通信学会技術研究報告, 1990, AI90-55, PRU9-49.
- [16] Guttman, A.: "R-TREES: A Dynamic Index Structure for Spatial Searching", in *Proc. of the 1984 ACM-SIGMOD int'l conf.*, pp. 44-57, 1984.
- [17] 長谷川純一, 久保田浩明, 鳥脇純一郎: "サンプル図形提示方式による画像処理エキスパートシステム IMPRESS", 電子情報通信学会論文誌, Vol. J70-D, No. 11, pp. 2147-2153, 1987.
- [18] 長谷川純一, 久保田浩明, 高須晶英, 鳥脇純一郎: "画像処理エキスパートシステム IMPRESS における画像処理手順集約化機能について", 情報処理学会論文誌, Vol. 29, No. 2, pp. 126-133, 1988.
- [19] Havens, W. and Mackworth, A.: "Representing knowledge of the visual world", *IEEE Computer*, Vol. 16, No. 10, pp. 90-96, 1983.
- [20] Herman, M. and Kanade, T.: "Incremental Reconstruction of 3D Scenes from Multiple, Complex Images", *Artificial Intelligence*, Vol. 30, pp. 289-341, 1986.

- [21] 饅怡虹, 島海有紀, 大沢裕, 坂内正夫: “重視領域の指定によりデザイナーの好みを反映しうる限定色表示手法”, テレビジョン学会誌, Vol. 45, No. 1, pp. 86-93, 1991.
- [22] Lowe, D. G.: “Visual Recognition from Spatial Correspondence and Perceptual Organization”, in *Proc. of 9th IJCAI*, pp. 953-959, 1985.
- [23] Lu, W., Ohsawa, Y. and Sakauchi, M.: “A database capture system for mechanical drawings using an efficient multi-dimensional graphical data structure”, in *Proc. of 9th ICPR*, pp. 266-269, 1988.
- [24] Marr, D.: *Vision*, W.H.Freeman and co., 1982, (邦訳) 乾, 安藤 訳: “ビジョン-視覚の計算理論と脳内表現”, 産業図書, 1987.
- [25] 松原仁, 山本和彦, 坂上勝彦, 山岸健太郎: “認識機構学習システム MIRACLE-IV における内部構造モデルの獲得”, 電子情報通信学会技術研究報告, 1988, PRU88-19.
- [26] 松原仁, 坂上勝彦, 山本和彦, 山岸健太郎: “画像学習システム MIRACLE-IV における機能的特徴と視覚的特徴の対応付け”, 情報処理学会論文誌, Vol. 31, No. 9, pp. 1302-1311, 1990.
- [27] 松山隆, 尾崎正治: “LLVE: トップダウン・セグメンテーションのための画像処理エキスパートシステム”, 情報処理学会論文誌, Vol. 27, No. 2, pp. 191-204, 1986.
- [28] 松山隆, 村山直樹, 伊藤貴康: “画像解析における戦略の表現”, 情報処理学会論文誌, Vol. 29, No. 2, pp. 169-177, 1988.
- [29] Matsuyama, T. and Hwang, V. S.: *SIGMA — A Knowledge-Based Aerial Image Understanding System*, Plenum publishing co., 1990.
- [30] McKeown, D. M.: “Rule-based interpretation of aerial imagery”, *IEEE Trans. on PAMI*, Vol. PAMI-7, No. 5, pp. 570-585, 1985.

- [31] Michalski, R. S.: "A Theory and Methodology of Inductive Learning", in Michalski, R. S., Carbonell, J. G. and Mitchell, T. M., editors, *Machine Learning*, Springer-Verlag, 1983.
- [32] Michalski, R. S. and Stepp, R. E.: "Learning from Observation: Conceptual Clustering", in Michalski, R. S., Carbonell, J. G. and Mitchell, T. M., editors, *Machine Learning*, Springer-Verlag, 1983.
- [33] Minsky, M.: "A framework for representing knowledge", in Haugeland, J., editor, *Mind Design*, pp. 95-128, The MIT Press, Cambridge, MA, 1981.
- [34] 沼尾雅之, 石塚満: "フレーム形探索システム (FBSS) による 3 次元画像の解釈", 電子通信学会技術研究報告, 1982, AL82-65.
- [35] 大沢裕, 坂内正夫: "良好な動特性を持つ多次元点データ管理構造の一提案", 電子情報通信学会論文誌, Vol. J66-D, No. 10, pp. 1193-1200, 1983.
- [36] 大沢裕, 坂内正夫: "空間的な位置関係に依存した検索に適した線情報の管理方式", 電子情報通信学会論文誌, Vol. J69-D, No. 5, pp. 724-732, 1986.
- [37] 大沢裕, 滝嶋康弘, 坂内正夫: "会話的な認識による信頼性の向上を図った地図自動入力システム", 電子情報通信学会論文誌, Vol. J72-D-II, No. 4, pp. 545-554, 1989.
- [38] 大沢裕, 坂内正夫: "2 種類の補助情報により検索と管理性能の向上を図った多次元データ構造の提案", 電子情報通信学会論文誌, Vol. J74-D-I, No. 8, pp. 467-475, 1991.
- [39] 坂内正夫, 大沢裕: 画像データベース, 昭晃堂, 1987.
- [40] 坂内正夫: "画像検索技術", 電子情報通信学会誌, Vol. 71, No. 9, pp. 911-914, 1988.
- [41] 坂内正夫, 佐藤真一: "画像データベースにおけるモデル形成", 電子情報通信学会論文誌, Vol. J74-D-I, No. 8, pp. 455-466, 1991.

- [42] Sakauchi, M.: "Two interfaces in image database systems", in *Proc. of MIV*, pp. 22-27, 1989.
- [43] 坂上勝彦, 山本和彦, 松原仁, 山岸健太郎: "認識機構学習システム MIRACLE-IV における画像処理戦略の学習", 電子情報通信学会技術研究報告, 1988, PRU88-20.
- [44] Sammut, C. and Banerji, R. B.: "Learning Concepts by Asking Questions", in Michalski, R. S., Carbonell, J. G. and Mitchell, T. M., editors, *Machine Learning volume II*, Morgan Kaufmann, 1986.
- [45] 佐藤真一: "画像データベース獲得の一般枠組に関する研究", 修士論文, 東京大学大学院工学系研究科 情報工学専攻, 1988.
- [46] 佐藤真一, 大沢裕, 坂内正夫: "状態遷移モデルによる地図の理解", 第39回情報処理全国大会, 1989, 1B-6.
- [47] 佐藤真一, 大沢裕, 坂内正夫: "図面理解システムにおける抽象化規則のモデル化", 電子情報通信学会春季全国大会, 1989, D-490.
- [48] 佐藤真一, 大沢裕, 坂内正夫: "状態遷移モデルに基づく図面理解システムに関する考察", 電子通信学会技術研究報告, 1990, AI90-52, PRU90-46.
- [49] 佐藤真一, 坂内正夫: "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", 情報処理学会研究報告, Vol. 91, No. 56, 1991, コンピュータビジョン研究会, 91-CV-73.
- [50] 佐藤真一, 坂内正夫: "状態遷移モデルの一般画像への適用の試み", 電子情報通信学会春季全国大会, 1991, D-576.
- [51] 佐藤真一, 坂内正夫: "図形間の空間的關係に基づく構造記述ルールの学習アルゴリズムに関する考察", 情報処理学会第43回全国大会, 1991, 7D-1.
- [52] 佐藤真一, 坂内正夫: "多階層状態遷移モデルによる画像理解手法の検討", 情報処理学会第43回全国大会, 1991, 1D-3.

- [53] 佐藤真一, 大沢裕, 坂内正夫: “対象の多様性に対応しうる図面理解システムの一提案”, 情報処理学会論文誌, 1991, 投稿中.
- [54] 佐藤真一, 坂内正夫: “与えられた図形事例からの構造記述ルールの学習”, 1991 年電子情報通信学会秋季大会, 1991, SD-9-1.
- [55] Satoh, S., Ohsawa, Y. and Sakauchi, M.: “Drawing image understanding framework using state transition models”, in *Proc. of 10th ICPR*, pp. 491-495, 1990.
- [56] Satoh, S. and Sakauchi, M.: “Descriptive ability of drawing image understanding framework using state transition models”, in *Proc. of MVA '90*, pp. 199-202, 1990.
- [57] Sellis, T., Roussopoulos, N. and Faloutsos, C.: “The R⁺ Tree: A Dynamic Index for Multidimensional Objects”, in *Proc. 13th int'l conf. of Very Large Databases*, pp. 507-518, 1987.
- [58] Shafer, G.: *A Mathematical Theory of Evidence*, Princeton University Press, Princeton, N. J., 1976.
- [59] Shapiro, E. Y.: “Inductive Inference of Theories From Facts”, Technical Report TR192, Yale University, Dept. of Computer Science, 1981, (邦訳: “知識の帰納的推論”. 共立出版, 1986).
- [60] Shapiro, E. Y.: *Algorithmic Program Debugging*, The MIT Press, Cambridge, Massachusetts, 1982.
- [61] Shipman, D. W.: “The Functional Data Model and the Data Language DAPLEX”, *ACM Transaction on Database Systems*, Vol. 6, No. 1, pp. 140-173, 1981.
- [62] Stefik, M. and Bobrow, D. G.: “Object-Oriented Programming: Themes and Variations”, *The AI Magazine*, pp. 40-62, 1986.

- [63] 高須晶英, 長谷川純一, 鳥脇純一郎: “サンプル図形提示方式による面図形抽出手順の自動構成法について”, 情報処理学会論文誌, Vol. 29, No. 2, pp. 134-141, 1988.
- [64] 滝嶋康弘: “ヒューマンフレンドリーな対話を特徴とする地図入力システム”, 修士論文, 東京大学大学院 工学系研究科 電子工学専攻, 1988.
- [65] 田中克巳: “ハイパーメディアとデータベース”, 1990 年テレビジョン学会年次大会, 1990, S2-5.
- [66] Tenenbaum, J. M. and Barrow, H. G.: “Experiments in Interpretation-Guided Segmentation”, *Artificial Intelligence*, Vol. 8, pp. 241-274, 1977.
- [67] 上野晴樹, 石塚満 (編): 知識の表現と利用, 知識工学講座 2, オーム社, 1987.
- [68] Winston, P. H.: “Learning Structural Descriptions from Examples”, in Winston, P. H., editor, *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).
- [69] Winston, P. H., Binford, T. O., Katz, B. and Lowry, M.: “Learning Physical Descriptions from Functional Definitions, Examples, and Precedents”, in *AAAI-83*, pp. 433-439, 1983.
- [70] Wirth, N.: *Algorithms+Data Structure = Programs*, Prentice-Hall, 1976, (邦訳: 片山卓也, “アルゴリズム+データ構造=プログラム”, 日本コンピュータ協会).
- [71] 山根淳, 佐藤真一, 坂内正夫: “キーワード自動獲得を考慮した画像データベースの実験による評価”, 電子情報通信学会秋季全国大会, 1991, SD-9-2.
- [72] Zucker, S. W., Rosenfeld, A. and Davis, L. S.: “General Purpose Models: Expectations about the Unexpected”, in *Proc. of 4th IJCAI*, pp. 716-721, 1975.
- [73] “特集:オブジェクト指向データベースシステム”, 情報処理学会誌, Vol. 32, No. 5, 1991.

付録 A

BD 木ライブラリを用いた空間 Prolog 処理系

A.1 概要

論理型言語 Prolog は、プロダクションシステムやエキスパートシステム、学習システムといった、演繹や帰納などの知的処理を行うシステムの記述に適した汎用プログラミング言語である。しかしながら、Prolog での assert や retract といった述語で扱える内部データベースは、基本的に線形な探索しかできないため、Prolog を画像理解などの空間性を持った対象を扱うようなアプリケーションの記述に利用しようとした場合、対象の空間性を保持した形でこれを行うのはきわめて困難である。

一方、一般の逐次処理マシンで多次元の空間性を持ったデータを効率良く扱うために、木構造を応用したデータ構造がいくつか考案されている [3, 16, 35, 36, 38, 57, 70]。これらはそれぞれ特徴があるが、基本的には n 個のオブジェクトに対する空間探索などの操作に対し、 $O(\log n)$ の計算量を保証するなど、これらを効率良く行えるようになっている。

そこで本章では、Prolog の処理系に多次元データ構造の一つである BD 木 [35, 36] の 2 次元版を扱える機能を追加し、さらにいくつかの基本的な幾何形状をもつデータを扱うような拡張を施した処理系を作成したので、それについて述べる。本論文中で述べた一連のシステムは高次の処理の部分でいずれも Prolog を利用しており、その扱う対象が 2 次元空間性を持つものであるため効率化のために本章で述べる拡張 Prolog 処理系を用いている。以下、まず処理系の特徴について述べ、その適用分野について述べる。ついで処理系の構成とその空間性を持ったデータの扱いについて述べ、新たに追加した述語をいくつか紹介する。

A.1.1 処理系の特徴

通常の Prolog システムで 2 次元の空間性を持ったデータを扱う場合、各データにその存在場所の座標値などを属性として持たせることになるが、それを空間的に検索することを考えると、結局各データに対して総当りのにその座標値を調査しなければならず、極めて効率が悪い。本処理系では、2 次元の空間的な属性を持ったデータの扱いを、多次元データ構造 BD 木の利用により効率良く、しかも Prolog の内部データベースへのアクセスと同様の操作で行えるようになっている。

今回は多次元データ構造として BD 木を用いているが、本処理系ではこれをあくまで空間的な探索のできるブラックボックスとして用いており、他のデータ構造でも全く同様に用いることができる。

ここで、本処理系の特徴についてまとめておく。

- (1) 処理系で扱う空間的なデータには、それが存在する 2 次元空間中の位置とその幾何形状という属性が付与される。
- (2) 各データに対し、ある範囲に存在するデータを列挙させる範囲検索、あるデータの近くに存在する近接データ検索などの空間検索を行う述語が用意されている。
- (3) 各データ間の幾何学的な関係を調査する述語がいくつか用意されている。
- (4) 空間検索は、BD 木の利用により効率良く行うことができる。

A.1.2 処理系の適用分野

本処理系の利用に適したアプリケーション分野について述べる。

本処理系で扱える空間的なデータは基本的な幾何形状を持った図形であるので、こうした基本的な図形の集合で表された対象を扱うのに適している。そして Prolog の強力な推論機構とユニフィケーションによるパターンマッチング能力により、それらを知的に扱うようなシステムの記述に適している。具体的には、与えられたモデルに従って演繹することによりモデルベースの認識を行うシステムや、基本的な図形で表された世界で問題解決をはかるシステム、あるいは与えられた図形データからあるパターンを検索するようなシステムなどがあげられる。

また、処理系で直接扱いうるデータは基本的な図形だけであるが、Prolog のデータ記述能力によりそれらを構造化して扱うことも可能であり、基本的な図形の組み合わせによる複雑な図形を扱うことも可能である。これにより、自然画像などのように複雑な対象を扱うことも考えられる。

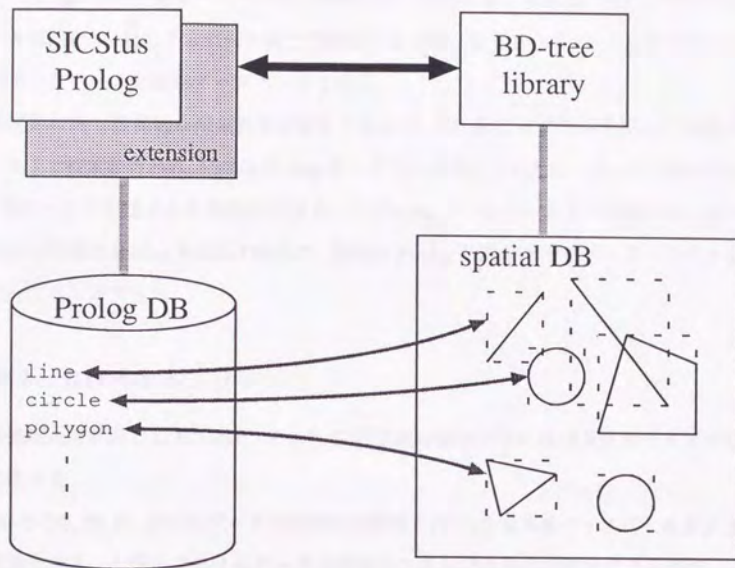


図 A.1: 本処理系の構成

A.2 処理系の構成

A.2.1 全体構成

図 A.1に本処理系の全体の構成を示す。

現在のところ、本処理系はSun 3とSparc Stationで稼働している。実装としては、Prolog処理系としてはSICStus Prologを用い、Prolog言語およびC言語で拡張機能のための部分を記述し、それにC言語で記述されたBD木ライブラリをリンクする形で実現している。Prolog処理系にC言語をリンクする部分には、SICStus Prologのforeign述語などを用いている。

本処理系で扱う空間性を持った図形を表すデータ(図形データ)は、Prologの内部データ

ベースと BD 木の空間データベースとの両方で二元的に蓄えられる。両データベースの各データは、互いに対応するデータ同士で参照し合う形になっている。こうしてできた 2 重構造のデータベースを図形データベースと呼ぶ。

処理系に対し空間的な探索の要求がなされると、BD 木ライブラリを用いて空間データベース上で探索が行われ、結果は Prolog データベースのデータとして得ることができる。また、各データに付加される属性などはすべて Prolog データベース上で管理され、各データの個別の処理は Prolog 処理系の内部で、通常の Prolog の内部データベースへのアクセスとして行うことができる。

A.2.2 BD 木ライブラリ

本処理系で利用した BD 木と、それを C 言語から利用するための BD 木ライブラリについて述べる。

BD 木 [35, 36] は、多次元データの空間的な管理を行う、2 分木をベースにした多次元データ構造である。木構造におけるデータの検索のコストはその木の深さによるので、一般にデータの個数 n に対して $O(\log n)$ のコストで可能である。また BD 木では、多次元データを単純に 2 分木構造に投入した場合に問題になる、空間的に偏りのあるデータによる「無効な分割」を抑えるために「領域式」を導入し、データの偏りによらず木の深さを小さく抑え、ノードの利用率を高く保つように考慮している。詳細については参考文献を参照されたい。図 A.2 に BD 木による 2 次元空間中の点データの表現を示す。多次元データ構造 BD 木の特徴を以下に挙げる。

- 「領域式」により、データの分布によらず木の深さは低く抑えられ、ノードの利用効率も高く保たれている。
- データの投入／削除によるデータ構造の再構築の必要がなく、動特性に優れている。

BD 木ライブラリは、この BD 木を C 言語から利用するためのライブラリである。このライブラリで実現している BD 木の仕様としては、2 次元の最高 22 段まで分割可能な非バ

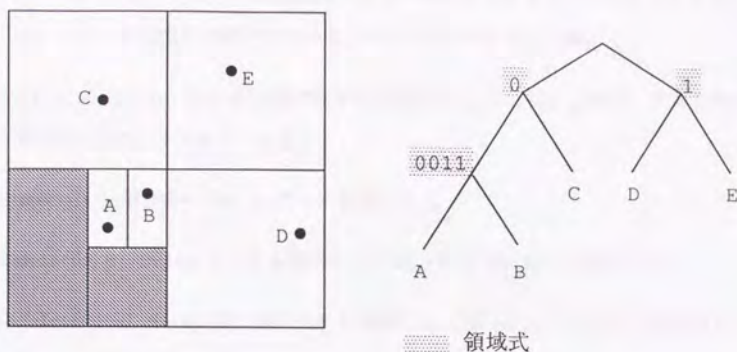


図 A.2: BD 木による 2 次元空間データの表現

ケット型 BD 木となる。データはその中心点の位置とデータの占める矩形領域により管理される。このライブラリの特徴としては、ライブラリの利用者に BD 木の内部構造を全く見せず、これを意識することなく単に空間的な探索のできるブラックボックスとして扱えるという点である。まず利用者はライブラリ関数により BD 木を生成し、これに空間的なデータをライブラリ関数を通して投入する。投入したデータは、ライブラリ関数で用意された範囲検索などの空間探索によってのみアクセスされる。このようにすべての操作がライブラリ関数を介して行われるため、ライブラリを利用する側はライブラリの内部構造や実装法に左右されることがない。従って例えばこの実装を、BD 木を多分木構造ベースに拡張して特性の向上を図った GBD 木 [38] などに変更することにより、ライブラリの利用者 (例えば本 Prolog 処理系) 側を変更することなく、より一層の検索特性の向上を図ることも可能である。

ライブラリ関数の仕様は以下のようにになっている。

BdNew() 新しい BD 木を作成し、その識別子を返す。

BdThrow(id, data) id という識別子で表される BD 木に data を投入する。data は Bd-

Head ストラクチャをヘッダ部に持つ実データストラクチャへのポインタであり、そのヘッダに空間情報が書かれている (サンプルプログラム参照)。

BdGet(id, xl, yl, xu, yu) 引き数で表される領域 $((xl,yl)-(xu,yu))$ のデータを **BdList**(付録参照) へのポインタとして返す。

BdUnget(p) 使用済みの **BdList** データを開放する。

BdDelete(id, p) **BdList** データを識別子 **id** で表される BD 木から削除する。

BdDeleteOne(id, data) 単一のデータを識別子 **id** で表される BD 木から削除する。

BdFree(id) 識別子 **id** で表される BD 木を開放する。

なお、図 A.3にこの BD 木ライブラリを用いたサンプルプログラムを示し、図 A.4に **BdList** の定義を含むヘッダーファイルの内容を示す。プログラムは、まず BD 木を生成し、これに乱数を用いて発生させた 100 個の矩形データを投入する。次いでそれらのうちからある領域内に含まれるデータを取り出し、これを削除するというものである。プログラム自体は全く意味をなさないものであるが、空間探索のための基本的な操作は含まれている。またこのプログラミングにおいて、BD 木の内部構造には全く関知する必要がないことが分かる。

A.2.3 図形データの内部構造

本処理系で扱う図形データの内部構造について述べる。ここでいう図形データとは、ある幾何形状を持ち、一定の 2 次元空間中のある場所に存在するようなデータをいう。本処理系で扱う図形データは、現在の実装では線分・円・多角形の 3 種類の基本図形のみである。しかし、これらの図形を組み合わせで構造化し、まとめて一つの図形として扱うことも可能であり、これによりかなり複雑な図形も処理することができる。もちろん、この 3 種類以外にも拡張は可能である。

図 A.5に図形データの Prolog データベース上での表現を示す。*Area* はデータの存在範囲を表し、データに外接して辺が X,Y 両軸に並行である長方形 (外接長方形と呼ぶ) で表され

```
#include "bd.h"

struct {
    BdHead head;
    int body;
} data[100];

main()
{
    void *id;
    int i;
    BdList *p;

    srand(time(0));
    id = BdNew();
    for (i = 0; i < 100; i++) {
        data[i].head.xu = (data[i].head.xl = (random() & 255)) +
            (random() & 63);
        data[i].head.yu = (data[i].head.yl = (random() & 255)) +
            (random() & 63);
        BdThrow(id, &data[i]);
    }
    p = BdGet(id, 0, 0, 128, 128);
    BdDelete(id, p);
    BdUnget(p);
    BdFree(id);
}
```

図 A.3: サンプルプログラム


```
/*
 *      BD tree manipulation library header file
 *      @(#)bd.h          1.1 (sato@sak.iis.u-tokyo.ac.jp) 8/18/89
 *
 *      id = (VOID *)   BdNew()
 *                      BdThrow(id, data)
 *                      VOID *data;
 *      p = (BdList *)  BdGet(id, xl, yl, xu, yu)
 *                      BdUnget(id, p)
 *                      BdDelete(id, p)
 *                      BdDeleteOne(id, data)
 *                      BdFree(id)
 *
 */

#include "h/void.h"

typedef struct BdHead_s {
    VOID *back;
    short xl, yl, xu, yu;
} BdHead;

typedef struct BdList_s {
    struct BdList_s *forw, *back;
    BdHead *data;
} BdList;

VOID    *BdNew();
VOID    BdThrow();
BdList  *BdGet();
VOID    BdUnget();
VOID    BdFree();
```

図 A.4: bd.h ヘッダーファイル

`$sp_data(Area , Data , Attrib) :- $data_id(DataId).`

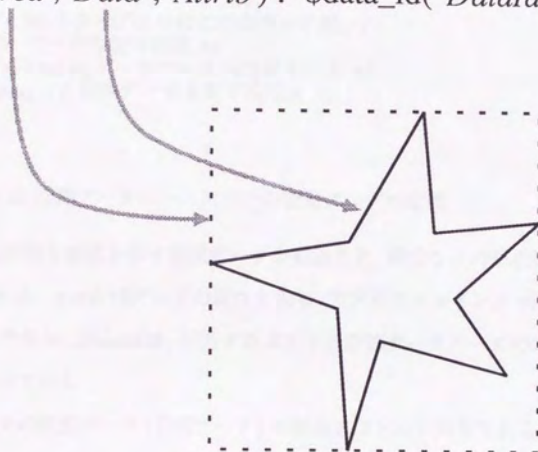


図 A.5: 図形データの Prolog データベース上での表現

表 A.1: Prolog データベース上での幾何形状の表現

システム上での表現	意味
<code>line((X₁,Y₁)-(X₂,Y₂))</code>	(X ₁ ,Y ₁) から (X ₂ ,Y ₂) を結ぶ線分.
<code>circle((X,Y),R)</code>	中心 (X,Y), 半径 R の円.
<code>polygon([(X₁,Y₁),(X₂,Y₂), ..., (X_n,Y_n)])</code>	(X ₁ ,Y ₁),(X ₂ ,Y ₂),..., (X _n ,Y _n) を 頂点とした多角形.

```

struct data {
    BdHead head;      /* BD 木ライブラリのためのヘッダ部 */
    char type;        /* データの幾何形状 */
    int ref1, ref2;    /* Prolog データベースへのポイント */
    union g_data g_data; /* 幾何データを表す共用体 */
};

```

図 A.6: 空間データベースにおける空間データの定義

る。Data はデータの幾何的な形状を示す形状データが格納され、線分などの形状はそれぞれ表 A.1 のように表される。Attrib はデータの属性を表し、アプリケーションプログラムで自由に使用することができる。DataId は、対応する BD 木の空間データベースのエントリへのポイントが格納されている。

空間データベース上での図形データ (空間データ) の構造もほとんど同様である。図 A.6 に C 言語の構造体による空間データの定義を示す。外接長方形によるデータの存在領域は head 部に記述されており、データの幾何形状は type と g_data に Prolog データベースと等価の定義がなされ、ref1 と ref2 に Prolog データベースのエントリへのポイントが格納される。ただし、Prolog データベースにあった属性を表す部分は存在せず、空間データベースではデータの空間性のみを扱うようになっている。

A.3 拡張した述語

2重構造を持つ図形データを通常の Prolog システムの内部データベースと同様の操作で扱えるようにするため、本処理系では新しい Prolog の述語を定義する形でそのインターフェイスを提供している。ここでは、こうして新たに拡張した述語のうちのいくつかを紹介する。

BD 木の生成・削除

`new_tree(DB)` 新しい図形データベースを作成し、それを *DB* とする。

`free_tree(DB)` *DB* で表される図形データベースを削除する。

データの投入・削除

`throw(DB, Data, ID)` *DB* で示される図形データベースに *Data* の形状データ (表 A.1 に従う) で表される図形データを投入し、結果の空間データベース上のデータのポイントを *ID* とする。

`delete.data(DB, ID)` 空間データベース上のデータのポイント *ID* で示されるデータを図形データベース *DB* から削除する。

空間検索

`scan_area(DB, Area, List)` 図形データベース *DB* から矩形領域 *Area* に含まれる図形データを検索し、結果を *List* とする。

`get_near_from_line(DB, Line, Dist, List)` 図形データベース *DB* において、線分 *Line* から距離 *Dist* 以内にある図形データを検索し、結果を *List* とする。

データの種別

`is_triangle(Obj)` 図形データ *Obj* は三角形である。

`is_circle(Obj)` 図形データ *Obj* は円である。

データの相対関係

$\text{on}(A, B)$ 図形データ A は B に接している.

$\text{right}(A, B)$ 図形データ A は B の右にある.

これらの拡張により, A.1.2で述べたようなアプリケーションに必要な基本的な操作は可能となっている.

A.4 考察

本章では, 多次元データ構造の利用により空間性を持ったデータを効率良く扱えるように拡張した Prolog 処理系について述べた. 実際のアプリケーションシステムに応用することにより, Prolog の強力な推論機構と非数値データ操作能力を利用し, しかも空間性を持ったデータを効率良く扱う高度な推論システムが実現可能である. また, Prolog 言語レベルでの拡張性にも優れている. 今後の拡張としては, システムが扱える基本的な図形をさらに拡充し, 図形間の空間的な関係としてもさらに高度なものを導入することや, 現在扱えるのは 2 次元のデータであるが, これを 3 次元に拡張して 3 次元空間での問題解決に利用できるようにすることなどが考えられる.

付録 B

状態遷移型図面理解システムにおける人間 機械協調によるルール作成支援システム

B.1 概要

図面などからのデータベース獲得をより高品質なものにするために、データベース獲得を行う画像理解システムを、ユーザに問い合わせを行うような人間機械協調型のものにするというアプローチがある。大沢らのシステム [37, 64] では、地図図面認識システムにおいてこのようなアプローチを取っている。このシステムでは、まずシステムが自動的に認識できる部分についてはすべて処理を行ってしまい、認識できなかった部分についてのみユーザに編集処理を求める。その際、システムが想定し得る認識結果の例を提案として提示することによりユーザの負担を軽減するようになっている。

こうしたシステムの場合、一度ユーザに問い合わせたことがらについては、再びユーザに問い合わせしてしまうことのないようにすることが望ましい。すなわち、ユーザからシステムの知らない局面についての情報が得られた場合、それ以後同じ局面に出会った時にユーザに再び問い合わせることなくその局面の解釈を行うように、その得られた情報に従って何らかの形でシステムの内部状態を変化させるような機構が必要である。こうしたことを行うためには、コンピュータ学習の手法を利用するのが適当であると思われる。

ここでは人間機械協調型認識システムの一つのあり方として、人間機械協調による認識システム作成支援について考える。4章で述べた状態遷移モデルに基づく認識システムでは、対象ごとの認識システム作成のためには認識知識を状態遷移ルールで与えれば良いため、コンピュータ学習システムとの親和性が良い。そこで、コンピュータ学習の一手法である帰納推論の手法を用い、状態遷移モデルにおけるボトムアッププロセスルールをユーザとの対話によって段階的に作成するシステムについて述べる。その上で、この手法を人間機械協調型の画像／図面認識システムに応用するための方策について検討する。

B.2 システムの構成

図 B.1に状態遷移ルール作成支援システムの構成を示す。システムは大きく分けて状態遷移型図面認識システムの部分と状態遷移ルール学習システムの部分からなる。図面認識

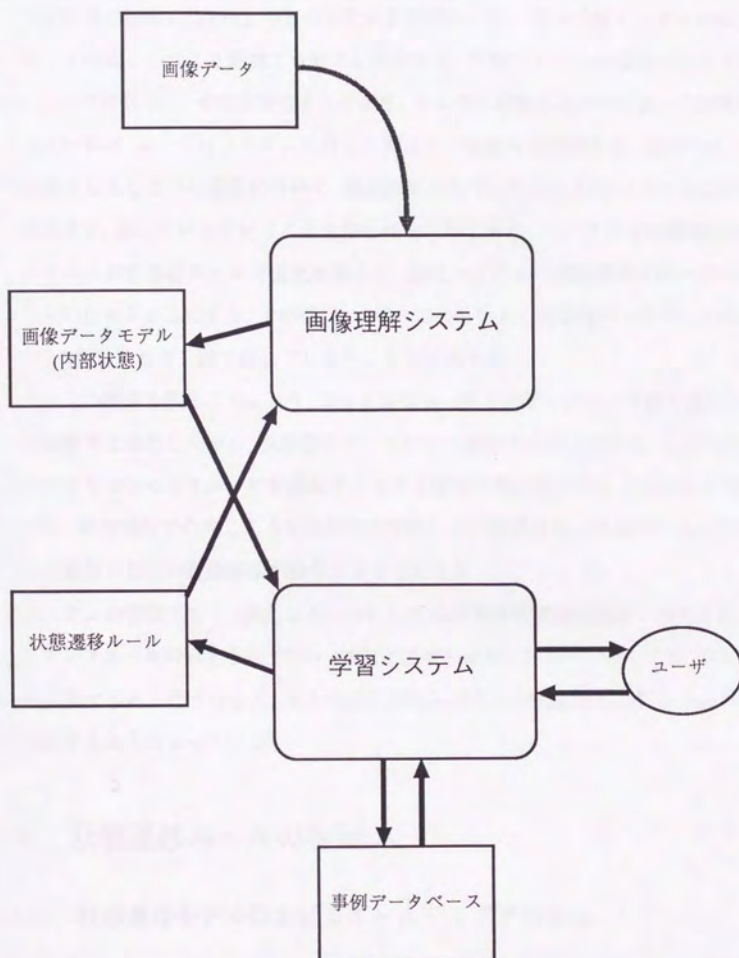


図 B.1: 状態遷移型図面理解システムのルール作成支援システム

システムの部分は状態遷移モデルに基づいた構成になっており、トークンの形式で記述された認識対象の図面について、与えられた状態遷移ルールに従って各トークンの状態を順次遷移していくことにより認識プロセスを実現する。学習システムは認識システムの認識結果をユーザに提示し、その妥当性をうかがう。もしその認識結果の中に誤った認識結果が含まれていれば、ユーザはシステムに対して誤っている箇所を指摘する。現状では、そうした誤認識をしてしまった箇所について、認識結果として正しいものをシステムに知らせることはできず、誤っているということを知らせるのみである。ユーザからの指摘に対し、学習システムは状態遷移ルールに変更を加えて、認識システムの認識結果がユーザの指摘に応えるものになるようにする。その際、ユーザとのセッションを事例データベースに蓄えておき、同じ問い合わせを繰り返してしまうことを回避する。

このような構成を取ることにより、システムはユーザとのセッションを繰り返し、妥当でない認識結果を修正しながら、状態遷移ルールの方も修正することになる。こうして、ユーザとのセッションによりユーザが満足するような認識結果が得られるようになった局面については、他の場所での同じような局面では学習した状態遷移ルールを用いることによりもはや自動的に正しい認識結果が得られるようになる。

本システムの制限として、扱えるルールとしては状態遷移型図面理解システムにおけるボトムアップルールのみとなっている。またこのボトムアップルールとしても、全く無の状態から生成できるわけではなく、ある程度の骨組みができた状態からこれを完成させるように動作するようになっている。

B.3 状態遷移ルールの学習

B.3.1 状態遷移モデルにおけるボトムアッププロセス

状態遷移モデルにおけるボトムアップルールを自動生成する学習システムを作成するにあたり、ボトムアッププロセスの動作の状況を形式的に扱うための導入を行う。

一般に状態遷移モデルにおけるボトムアッププロセスは、パラメータ計測→状況判断→

状態遷移→パラメータ計測→...という繰り返しによりとらえることができる。これを実現する状態遷移ルールは、次のように記述できる。

$$\begin{aligned}
 S_1(\pi_{in_1}, \dots, \pi_{in_n}) \longrightarrow \\
 \mathcal{P}(\pi_{in_1}, \dots, \pi_{in_n}, \pi_{out_1}, \dots, \pi_{out_m}), \\
 \text{\$transform}(S_1(\pi_{in_1}, \dots, \pi_{in_n}, \pi_{out_1}, \dots, \pi_{out_m})).
 \end{aligned} \tag{B.1}$$

$$\begin{aligned}
 S_{11}(\pi_1, \dots, \pi_{m+n}) \longrightarrow \\
 C_{111}(\pi_1, \dots, \pi_{m+n}), \\
 \text{\$transform}(S_{111}(\pi_1, \dots, \pi_{m+n})).
 \end{aligned} \tag{B.2}$$

$$\begin{aligned}
 S_{11}(\pi_1, \dots, \pi_{m+n}) \longrightarrow \\
 C_{112}(\pi_1, \dots, \pi_{m+n}), \\
 \text{\$transform}(S_{112}(\pi_1, \dots, \pi_{m+n})).
 \end{aligned} \tag{B.3}$$

⋮

このルールにより実現される状態遷移の様子を図 B.2 に示す。これらのルールのうち、(B.1) はパラメータ計測をするためのルールであり、(B.2) と (B.3) とは計測したパラメータに従って状況判断を行い、適当な状態に遷移するルールである。ルールの内部について見ると、 $\mathcal{P}(\pi_{in_1}, \dots, \pi_{in_n}, \pi_{out_1}, \dots, \pi_{out_m})$ は入力パラメータ π_{in_n} ($n = 1, \dots$) に従って出力パラメータ π_{out_m} ($m = 1, \dots$) を計算する部分である。ここで $\text{\$transform}(S)$ は状態 S に遷移することを示す。従って (B.1) は、状態 S_1 のトークンについて新しいパラメータの計測を行い、もとのパラメータと計測したパラメータの両方をパラメータに持つ次状態 S_{11} に遷移させるルールである。 $C(\pi_1, \dots, \pi_n)$ は、パラメータ π_1, \dots, π_n がある条件を満たしているかどうかをテストする部分である。従って (B.2) と (B.3) は、状態 S_{11} のトークンについて条件 C_{111} が満たされれば次状態 S_{111} に、条件 C_{112} が満たされれば次状態 S_{112} に、...とそれぞれ遷移させるルールである。

こうしたルールの集積で、先にあげたボトムアッププロセスの基本的な繰り返しが実現されている。これまでに作成したボトムアップルールも、上記のようなルール群によってで

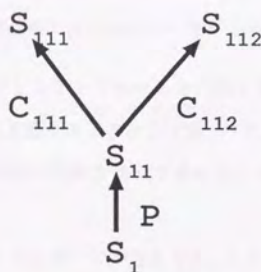


図 B.2: 一般的な状態遷移の1ステップ

きていると考えられる。

実際に認識システムのためのボトムアップルールを作成する場合、状態遷移の仕方の骨組みは簡単に決定できるが、あとの調整に手間取ることが多い。上記のようにボトムアップルールが構成されている場合、骨組みとはルール群から条件判断の述語 C_n を除いたもので表され、調整とは条件判断におけるパラメータのテストの仕方の調整となる。ルール作成支援としては、骨組みはユーザが与えてしまうのを許すとしても、条件判断の部分の調整を支援してくれるようなものが望ましい。

B.3.2 ボトムアップルール作成支援

先の考察をもとにして、ボトムアッププロセスの作成支援プロセスとして以下のようなものを考える。

- 1° あるトークンについて、それがどういう状態になるべきかの指示をユーザより得る。
- 2° 現在トークンが持つパラメータでこれまでの指示と矛盾なく今の指示を実現するルールが存在すれば、それを生成して仮定する。

3° ルールが生成できなければ、状況判断のためのパラメータが不足していると判断し、新たなパラメータ生成のためのルールをユーザに求める。

これを繰り返すことにより、ボトムアップルールを作成していくことができる。このように、ここで実現したルール作成支援システムとしては、パラメータ計測のための部分はユーザが入力するものとし、状況判断の部分をユーザから与えられる例からの学習によって獲得するようになっている。

今、トークンにより記述されたイメージが与えられ、これらのトークンのうち、ある状況にあるものを定められた状態に遷移させるルールを生成することを考える。その状況を認識するのに十分な幾何的・空間的パラメータはすでに計測されており、生成すべきルールでは新たにトークンの空間的な関係を参照する必要はなく、これらのパラメータの構造や数値等を参照するだけで認識したい状況の判断ができるようになっているとする。

ユーザからの例示とは、与えられたイメージの中で、定められた状態になるはずのトークンのうちのいくつかと、定められた状態にならないはずのトークンのうちいくつかを、それぞれ正の例および負の例としてシステムに与えることである。システムはこれらの例を一般化し、正の例すべてが成り立ち、負の例すべてが成り立たないような条件のうちの任意の一つを帰納推論の手法を用いて生成する。例えば、 $a(5)$, $a(10)$, $a(20)$ が正の例として与えられ、 $a(-3)$, $a(-100)$ が負の例として与えられた時、システムは条件として $a(X)$ に対して $X > 0$ などを生成することになる。

ある時点までに与えられた例によって条件が生成されると、システムはそれをイメージに対して適用し、現時点での条件が成り立つトークンと成り立たないトークンをそれぞれ識別できるようにユーザに表示する。ユーザはこれを見て、意図した結果が得られていればその条件を求めたい条件として得て終了する。意図せぬ結果であれば、不適切な判断をされている部分のうち任意のものについて正の例もしくは負の例としてシステムに与える。これを適当な条件が得られるまで続けることになる。

もし与えられた例に矛盾しない条件が見つからなかった場合、システムは与えられているパラメータでは状況判断ができなかったものとし、停止する。例えば $a(10)$ が正の例とし

でも負の例としても与えられてしまった場合、この例に矛盾しない条件は生成できないのでシステムは停止し、 X 以外の新たなパラメータの計測をユーザに求める。そしてこの新たなパラメータの導入により、例えば $a(X, Y)$ における Y というパラメータの導入により正の例として $a(10, 10)$ 、負の例として $a(10, -10)$ が与えられると、システムは正しく状況判断ができるようになる。

B.3.3 帰納推論の手法

ここで用いた帰納推論の手法は、Shapiro の Model Inference System[59, 60] を基にし、条件として数値の大小比較が生成できるようにすると同時にパラメータの型が扱えるように拡張し、ボトムアッププロセスルール向きに効率化を図ったものである。

基本的な動作は、例として与えられた状態のパラメータの型に従って生成可能な条件を次々に生成し、与えられた例をこの条件に適用することによって、すべての例に矛盾しない条件としてはじめに見つかったものを出力するというものである。条件の生成をより一般的なものから段々特殊化したものへという順序で行うことにより、与えられた例に矛盾しない条件のうちでもより一般性の高いものを得ることを期待している。また、処理の効率化のため、条件の生成の過程においてある種の枝刈りを行っている。

ここでは具体的な帰納推論のアルゴリズムについては述べないが、このアルゴリズムで扱い得るパラメータの型と、扱い得る型の数値データからの条件の生成について述べる。

パラメータの型

扱えるパラメータの型としては、リスト構造データと数値データがある。リスト構造については、リストを構成する各要素の内容は参照しないが、任意の構造のリストを識別するルールが生成可能である。数値については、数直線的に大小比較ができる一般の実数と、角度などのようにサイクリックになっているものとを扱うことができる。

こうした型はユーザにより与えられる。システムは必要となった時点である状態についての各パラメータの型を問い合わせる。それに対しユーザは、リスト構造、一般の実数、

サイクリックな数およびその値がとり得る上限と下限、条件生成に無関係などといった情報を与える。以降システムは同じ状態のパラメータの型はすべてそれと同一であるとして処理を行う。

帰納推論のアルゴリズムにおいて、生成した条件を順次特殊化していく部分でパラメータのための条件を一つずつ付け加えていくことになるが、この時リスト構造についての条件の特殊化は Model Inference System の方法と同様の手法を用いている。すなわち、リスト構造を調べる条件として調べるべき構造の複雑さが順次増していくように条件を特殊化していく。数値データの場合の条件の生成については次で述べる。

数値データからの条件の生成

まずサイクリックではない通常の実数の場合を考える。この場合、与えられた正の例に出てくる数値データの上限と下限を取り出し、それぞれを T_{max} と T_{min} とする。また T_{max} 以下にならない最小の負の例がもしあればこれを F_{max} とし、 T_{min} 以上にならない最大の負の例がもしあればこれを F_{min} とする。このとき実数 X に対する条件は

$$X_{max} = \begin{cases} \frac{T_{max} + F_{max}}{2} & \text{if } F_{max} \text{ exists} \\ T_{max} & \text{otherwise} \end{cases}$$

$$X_{min} = \begin{cases} \frac{T_{min} + F_{min}}{2} & \text{if } F_{min} \text{ exists} \\ T_{min} & \text{otherwise} \end{cases}$$

$$X \leq X_{max} \quad (\text{B.4})$$

$$X_{min} \leq X \quad (\text{B.5})$$

$$X_{min} \leq X \leq X_{max} \quad (\text{B.6})$$

のいずれかとなる。

サイクリックな数の場合はもう少し複雑になる。まず領域の上限と下限をそれぞれ R_{max} と R_{min} とする。また、正の例の数値データのうち互いに異なる任意の3つをとりだし、小さい順に X_1, X_2, X_3 とする。今、

$$D_{13} = X_3 - X_1 \quad (\text{B.7})$$

$$D_{21} = (X_1 - R_{\min}) + (R_{\max} - X_2) \quad (\text{B.8})$$

$$D_{32} = (X_2 - R_{\min}) + (R_{\max} - X_3) \quad (\text{B.9})$$

という3つの数値を考える。これらの3つの数値を比較して、 D_{13} が一番小さい時は範囲は $[X_1, X_3]$, D_{21} が一番小さい時は範囲は $[R_{\min}, X_1], [X_2, R_{\max}]$, D_{32} が一番小さい時は範囲は $[R_{\min}, X_2], [X_3, R_{\max}]$ であると考え。その後、他の正の例の数値データすべてによってこの範囲を拡大していき、最終的に与えられた正の例すべてが当てはまる値の範囲が求まる。この範囲の拡大は、拡大しなければならない部分がつねに最小になるように選ばれてなされていく。その結果、値の範囲が $[X_{\text{low}}, X_{\text{high}}]$ である時、実数 X に対する条件は $X_{\text{low}} \leq X \leq X_{\text{high}}$ である。また、値の範囲が $[R_{\min}, X_{\text{low}}], [X_{\text{high}}, R_{\max}]$ である時、実数 X に対する条件は $X \leq X_{\text{low}}, X_{\text{high}} \leq X$ となる。

ただし、実際には両条件とも先の通常の実数の場合と同様に負の例を用いた範囲の拡張を行っている。ここでは説明の複雑さを避けるためにその詳細は省略しているが、その実現方法は明らかであろう。この範囲の拡張により、現在得られている事例のみから生成できる条件よりも若干一般性の高い条件が生成されることになる。

考察

あるトークンの状態においてそのパラメータにいくつかの数値パラメータがある場合、上記の条件が互いに組み合わされて実際の条件が生成される。これはリスト型のパラメータでも同じで、その状態のすべてのパラメータに従って生成されるそれぞれの条件が互いに組み合わされ、それらの連言として実際の条件が生成される。そして、それらが順次すべての例に矛盾していないかチェックされて行くことになる。

一般には、ある状況を認識するための条件としてはこうしてできた条件をさらに組み合わせ、それらの選言として最終的なルールが得られる。しかし今回は、学習システムの簡素化のためと現実のそのようなルールの必要性について考慮し、この選言によるルールには対応しないこととした。すなわち、ある状態のトークンからある状況を認識するためにシステムが生成できるルールは、その状態の各パラメータから生成される条件の連言によって

表された単一のルールということになる。

数値データを扱う部分に関して言うと、複数の数値パラメータに関してそれらの条件の連言による単一のルールによる条件は、その複数の数値パラメータにより構成される超空間における各軸に並行した超直方体の形で表現できる。ここでの帰納推論のアルゴリズムではこのような単純な形状の条件しか生成できないことになる。より一般的には、取り出したい条件としてはこのような超空間ではかなり複雑な形状をなすことがあり得る。

このようなパラメータによる超空間中で、より複雑な条件を抽出するアルゴリズムとしては STAR アルゴリズム [31, 32] などがある。これは基本的には本手法で言うところの連言による条件を複数生成して、これらの選言でより複雑な条件が生成できるようになっているアルゴリズムである。この場合、よりアルゴリズムは複雑化し、かつ得られている事例に矛盾しない条件に収束するまでより長い時間がかかることになる。

ここでは、このような複雑な条件が必要になる局面はあまりないであろうと考え、また人間とのインタラクションを行うためにより収束の速い単純なアルゴリズムを用いている。

B.3.4 状態遷移ルール作成支援システム

先に述べたように、状態遷移ルール作成支援システムは状態遷移型図面認識システムと状態遷移ルール学習システムとから構成される。システムの実装としては、4 章で述べた状態遷移型図面理解システムに状態遷移ルール学習システムを付加する形で行った。図面理解システムの能力としては以前のものと全く同一であり、これに状態遷移ルール学習機能がついたような構成となっている。状態遷移ルール学習システムの部分はすべて Prolog により記述され、800 行程度の規模となっている。図面理解システムの部分には以前のものからほとんど変更はない。

状態遷移ルール作成支援システムの実際の動作の様子について述べる。従来の状態遷移型図面理解システムと同様、処理対象はまず初期トークンとして表現される。この対象を表現しているトークンから、ある一定の特徴を持つようなトークンを分類するルールを生成することを考える。

まずこのような分類が可能となるようなパラメータを計測するための状態遷移ルールをシステムに与え、これにより各トークンの状態を遷移させておく。これで各トークンの内部状態には、考えている分類を実現し得るようなパラメータが含まれることになる。

次いでシステムを学習モードにする。はじめは、すべてのトークンの内部状態は分類されたトークンではないことを表している。これはユーザにとっては不満足な認識結果であるといえる。そこではじめに想定したある一定の特徴を持っていると考えられる、分類されるべきトークンを初期トークンから任意の一つ選び、これは分類されるべきトークンであるということを表す正の例としてシステムに指示を行う。システムはこれを受けて、ユーザが想定している分類を実現するルールを先に述べた帰納推論のアルゴリズムを用いて推定し、生成したルールでトークンを処理してその結果をユーザに提示する。通常は一回めのセッションの後はすべてのトークンが分類すべきトークンであるという結果が得られる。そこで今度は分類されるべきトークンではないトークンを任意の一つ選び、これを負の例としてシステムに指示する。システムはこれらのユーザからの例を用い、これを満足するような分類ルールを推論し、これでトークンを処理した結果をまたユーザに提示してその指示を仰ぐ。

このような一連のセッションを通じ、システムにはユーザが想定している分類に関する正の例と負の例が蓄えられる。これらの例が入力されるごとにシステムはそれまでに入力されたすべての例に矛盾しないような分類のための状態遷移ルールを生成し、その都度このルールによる分類結果を提示してくる。この分類結果がユーザの想定していた分類結果と一致するまで繰り返すことにより、そのような分類を実現する実際の状態遷移ルールが生成できることになる。

もし学習モードでのトークンの分類結果がユーザの希望に沿うものとなったら、その時点でのルールが生成したい状態遷移ルールであるという指示を与え、これを実際の状態遷移ルールに付け加え、学習モードを終了する。後はこの状態遷移ルールで実際に認識プロセスを行うことができる。

その結果を新たな初期状態として次の分類を行うルールを生成するため、新たなパラメータ計測ルールをシステムに与えて再び学習モードを繰り返すことも可能である。このよう

にして、ルールの学習を繰り返すことにより大規模な状態遷移ルールを生成することも可能である(図 B.3).

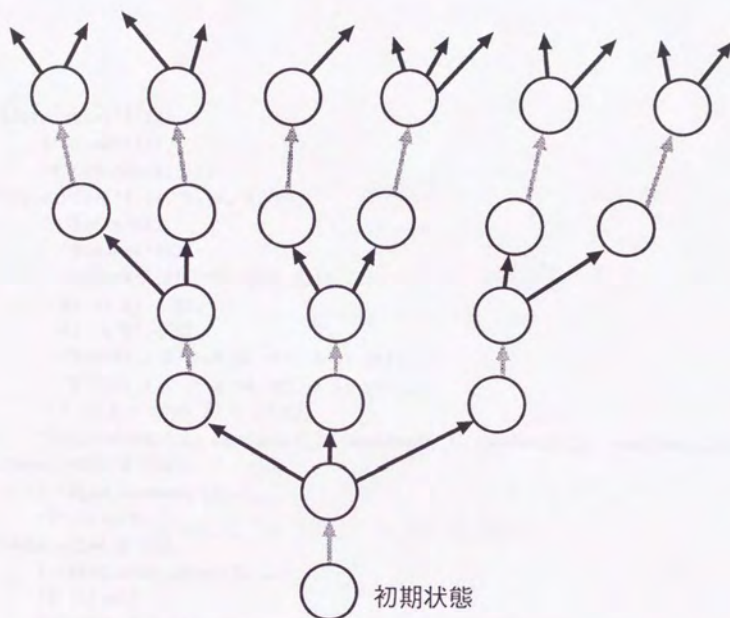
B.4 ルール作成支援システムの高度化の検討

B.4.1 パラメータ計測ルールの生成

ここまで述べてきたルール作成支援システムでは、トークンの状態に含まれるパラメータを用い、与えられた例に矛盾しないルールを生成するために条件判断のためのルールの解空間を探索するが、空間をすべて探索しつくしても例に矛盾しないルールが発見できなかった場合、システムは例に矛盾が存在することを検出する。ここでシステムはユーザに新たなパラメータを計測するためのルールの入力进行を要求することになっていたが、ここではシステムが自らパラメータ計測ルールを生成してくれるようなシステムの拡張を考える。

まずどの例(で示されたトークン)によって矛盾が発生したかを検出し、そのトークンに対して新たなパラメータ計測ルールを作成する。この計測ルールはあらかじめいくつかのパターンのものを与えておき、それらを順次試していくようになっている。図 B.4にパラメータ計測ルールのテンプレートを示す。

このテンプレートは、生成すべきパラメータ計測ルールとこれにより生成されるパラメータの型との組で構成されている。このテンプレートは上から順番に試行されていく。まずはトークンの面積を算出し、これをパラメータとするルールが生成される。ついでトークンの面積、周囲長、全幅、全高、そして面積を周囲長の2乗で割った形状パラメータという5つのパラメータを算出するルールが生成される。次は対象を簡単な凸包で見て、その頂点をリストとして求めるルールが生成される。最後はトークンの周囲状況としてその近傍に存在するトークンを抽出してこれをリストとして求めるルールを生成するようになっている。この際、近傍であることを表す距離をいくつかのものから選ぶようになっている。具体的には、まず150を用い、200まで5きざみで増加させる。次いで145からはじめ、100まで5きざみで減少させるというものである。



→ 分類ルール(システムにより生成可能)

→ パラメータ計測ルール(ユーザが作成)

図 B.3: 大規模な状態遷移ルールの作成支援

```

candidate_rule('$'(A),
    ('$area'(A)),
    '$'(cardinal(_))).
candidate_rule('$'(A, R, W, H, P),
    ( '$area'(A),
      '$around'(R),
      '$frame'((X1, Y1)-(X2, Y2)),
      W1 is X1 - X2,
      H1 is Y1 - Y2,
      '$'((W1 < 0 -> W is -W1; W is W1)),
      '$'((H1 < 0 -> H is -H1; H is H1)),
      P is A * 1000 // R // R),
    '$'(cardinal(_), cardinal(_), cardinal(_), cardinal(_), cardinal(_))).
candidate_rule('$'(Cs),
    ( '$get_corners'(Cs),
      '$'(list))).
candidate_rule('$'(L),
    ( '$get_near_loop'(D, L)),
    '$'(list)) :-
    D = 150; D = 155; D = 160; D = 165; D = 170; D = 175;
    D = 180; D = 185; D = 190; D = 195; D = 200;
    D = 145; D = 140; D = 135; D = 130; D = 125; D = 120;
    D = 115; D = 110; D = 105; D = 100.

```

図 B.4: パラメータ計測ルールのテンプレート

このテンプレートから、順番に新たなパラメータを計測するルールを選びだし、矛盾を生じた状態のパラメータ群に続いてこの新たなパラメータを追加するように生成するルールを追加する。こうしてパラメータ計測ルールを選んだ後、このルールに従って各トークンについて新たなパラメータを計測し、その結果について改めて例に矛盾しないルールの探索を行う。矛盾のないルールが存在すれば、その条件判断のルールとパラメータ計測ルールとがその時点における学習の結果となる。もし再び矛盾が生じれば、次のパラメータ計測ルールを選んで処理を繰り返すことになる。

この方法により、テンプレートに記述した範囲内ではパラメータ計測のルールが自動生成できるが、このテンプレートにパラメータ計測のすべての可能性を記述してしまわなければあらゆる場合に対応できたことにはならず、これは極めて非現実的である。また学習過程で矛盾が生じてパラメータ計測ルールを生成する場合、このテンプレートをはじめてから1次元的に探索しなければならず、この効率の悪さは否めない。実用的なパラメータ計測ルール生成としては、何らかの方法で矛盾の状況に応じて動的にパラメータ計測ルールを生成するようなものが必要である。

B.4.2 人間機械協調型図面理解システムへの適用の検討

本章で述べている状態遷移ルール作成支援システムは、ユーザとのインタラクションにより状態遷移ルールを少しずつ作成してくれるようなシステムである。これを人間機械協調型の図面理解システムに適用する方法について検討する。

先に述べた大沢らのシステムを例にとる。このシステムでは地図図面の認識において状況が複雑なため自動認識できなかった部分はそのまま残しておき、後の対話修正段階でこの部分を拡大してユーザに提示して修正を求める。その際、システムが想定し得るいくつかの解釈の可能性を提案として同時に提示してきて、ユーザはそれらのうちから適切なものを選ぶことにより修正作業に替えることも可能である。

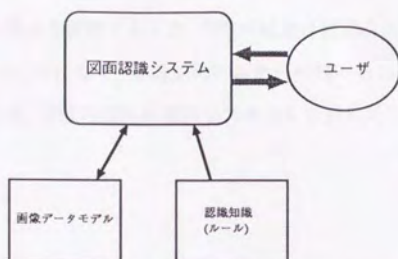
この対話修正段階であるが、処理対象の地図図面中에서도自動認識できなかった部分は比較的似たような局面のものが多く、システムから同ような部分が数多く提示され、そのた

びにユーザは同じような修正を繰り返したり同じような提案を選択したりという作業を行わなければならない。同じ局面については、一回(ないしは数回)ユーザの指示を受ければ後はシステムが自動的に判断して同じように処理してくれるような機構が望まれていた。本手法の利用により、この部分の改善が期待できる。

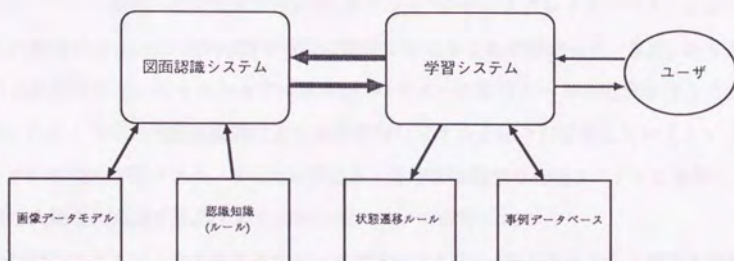
このような人間機械協調型の図面認識システムに対し、本手法を適用するための方策を表した模式図を図 B.5に示す。(a)は通常の間人機械協調型の図面認識システムである。この場合、同じような局面について毎回ユーザとのインタラクションを行わなければならない。ユーザとシステムとの情報のやりとりが極めて多いことを示している。

(b)はこれに学習システムを外付けすることにより人間機械協調における学習効果を期待する構成である。この構成は、認識システム側の変更が非常に少ないという利点を持つ。認識システムからは(a)の構成と同じように似たような局面について何回でもユーザとのインタラクションを求める。ところがこれに答えるのが学習システムとなるようになっていて、学習システムは必要に応じて実際にユーザに対して修正要求を行うが、このユーザとのセッションはすべて事例データベースに蓄えておき、また内部的に状況判断を行うための状態遷移ルールを持っており、データベースの事例からこの状態遷移ルールを学習する。図面認識システムから学習システムへの問い合わせは人間に対する場合のように認識できなかった箇所の図形を表示するのではなく、これをパラメータ化した内部状態の形で行う。学習システムではこの内部状態を分類する状態遷移ルールを学習していくことになる。従って学習システムは認識システムからなされる問い合わせのうちで「似たような局面」を区別できるようになってくるので、事例データベースからユーザが以前答えた内容を取り出すことによりユーザの代わりに認識システムの問い合わせに答えることができる。この構成により、認識システムによりなされる似たような局面に対する問い合わせは学習システムが代行することができるようになり、認識システムと学習システムの間のインタラクションは多いが、学習システムとユーザとの間のインタラクションが大きく削減できると考えられる。しかしながらこの構成では、学習の結果は認識システムの内部には変化を及ぼさないことになる。

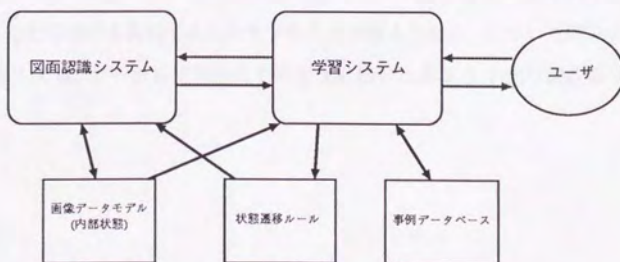
(c)は認識システムを状態遷移型の認識機構で実現することにより、本章で述べた状態遷



(a) 通常の人間機械協調型図面認識システム



(b) 通常の図面認識システムと学習システムの融合



(c) 状態遷移モデルを用いた図面認識システムと学習システムの融合

図 B.5: 状態遷移ルール学習システムを利用した人間機械協調型図面認識システムの構成

移ルール作成支援システムと同様の構成を取ったものである。認識システムからの問い合わせの流れは大体(b)の構成と同様であるが、学習の結果は認識システムが参照する状態遷移ルールにも反映されることになり、認識システムからの問い合わせ自体が減少することになる。この構成の場合は、学習の効果は認識システムにも表れることになる。

B.5 考察

状態遷移モデルに基づく図面理解システムに付加する形で動作する認識ルール作成支援システムについて述べた。状態遷移ルールの作成において、従来は試行錯誤により手間をかけて行っていた閾値の設定などの部分が、グラフィックディスプレイとマウスによるユーザからの指示によるいくつかの例で非常に簡便に行えることが分かった。また、あらかじめ決められたパターンのものであっても、パラメータ計測ルールの生成も行うことができた。このように人間機械協調により画像理解システムを徐々に構築していくというアプローチの有効性が示された。さらに本手法を人間機械協調型の認識システムに適用し、より高度な人間機械協調を実現するための方策について検討した。

今回のシステムにより生成できるルールで扱うのは、あくまでリスト構造か数値データかのパラメトリックなデータのみであった。しかしながら、画像や図形データという空間性を持ったデータの扱いにおいては、どうしてもパラメータで表し得ない部分が存在したり、あるいはそうした情報が本質的であったりすることが考えられる。こうしたデータを扱う学習システムとしては、データの空間性をそのままに扱うような能力が必要と考えられる。

付録 C

図形で与えられた事例からの構造記述の 学習

C.1 概要

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ど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[49, 51, 54]。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。今回のシステムの生成するモデルは、基本的には6章で述べた多階層状態遷移モデルに基づく線図形理解システムのためのモデルとして用いることを考えている。

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、意味ネットワークなどのような、図形を構成するオブジェクト間の関係が固定的に与えられている記述で与えられるのではなく、オブジェクトの空間性をそのまま保持し、かつその部分的な構造を反映した記述を用いている。

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。

以下本章では、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。

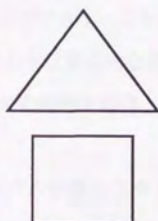
C.2 構造記述の学習

ここで考えている図形的に与えられた事例から構造記述を学習するシステムとは、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 C.1 で示されるような例を与えるとする。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。

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 ANALOGY システム [68] 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 [7, 12, 44, 68]。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おき、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。



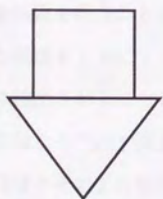
1. true



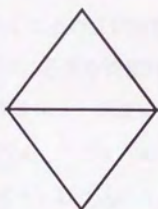
2. false



3. false



4. false



5. false

図 C.1: 「家」の学習のための「例」

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。このような問題点に対し、学習システムに意味ネットワークなどの抽象化された形で事例を入力するような構成では十分に対応することができないことになる。これに対処するためには、与えられた事例を意味ネットワークのような固定的な形態にしてしまうのではなく、ある程度もとの事例の図形ないし画像としての特徴を保存した形で学習システムに与える必要がある。

このようなアプローチをとったモデル学習システムとしては MIRACLE-IV システム [25, 26, 43] があげられる。このシステムは、シルエット画像の形で与えられた複数の事例から、その画像で表された対象物の内部構造を学習するシステムである。システムの構成としては、モデル獲得部と処理戦略部の2つの部分から構成される。与えられた事例としての画像はそのままシステムに入力され、モデル獲得部からの指示に従って処理戦略部が動的に画像処理を行うことにより、必要に応じてその視覚的特徴が抽出されることになる。このような構成をとることによりモデル獲得に必要な視覚的特徴が必要に応じて動的に、かつ柔軟に抽出されることになり、もとのシルエット画像に含まれるさまざまな視覚特徴を用いた柔軟なモデルが獲得できることになる。このように与えられた事例の図形／画像としての特徴をそのまま保持した形でシステムに取り込めるようにしておくというアプローチは、柔軟なモデル獲得が可能なモデル学習システムとしては重要であると考えられる。

この MIRACLE-IV システムの問題点をあげると、システムの生成できるモデルとしては獲得されるモデルの機能的な特徴と処理戦略部から得られる視覚的特徴とが単純に対応付けられたようなものに限られ、より複雑な構造を持つような対象には対応できないと考えられる。ある程度複雑な構造を持つようなモデルを獲得するには、与えられた事例から得られる視覚的特徴のみを扱うのではなく、ここから得られるその物理的構造もある程度扱える必要があると考えられる。

本章で述べる構造記述学習システムでは、これらの問題点にある程度応えるため、図形の

形で与えられた事例はその空間的な特徴がそのまま保存されたままの形で学習システムに与えられるようになっている。またその際、その構造がある程度参照できるように与えられた図形を部分的に構造化して扱うようになっている。次の節でこのシステムについて概説する。

C.3 構造記述学習システム

C.3.1 システムの構成

図 C.2に本章で述べる構造記述学習システムの構成を示す。ユーザより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ようになっている。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 [35] 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 C.3.2 節で説明する。

学習アルゴリズムは、Shapiro の Model Inference System [59, 60] をベースにし、構造記述規則の生成向きに作成した構造記述規則生成アルゴリズムを用いることにより実現している。この構造記述規則の生成は、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 C.3.3 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規則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る。このように与えられた例に矛盾しないような構造記述規則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るため、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

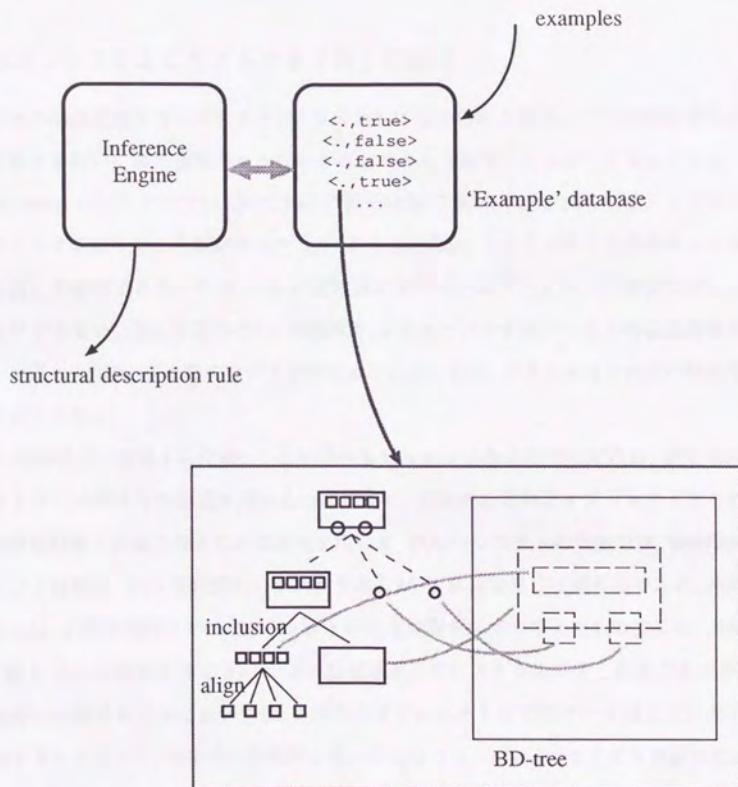


図 C.2: 構造記述学習システムの構成

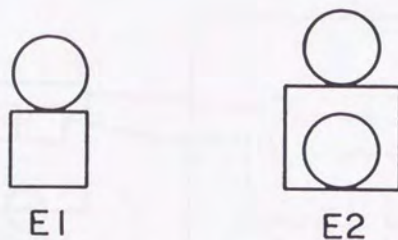
ル生成アルゴリズムでは、かなり特殊な構造記述ルールまで生成することができるため、より複雑な対象についてもこれを表す構造記述が生成できる。学習アルゴリズムについては C.3.4 節で述べる。

C.3.2 システムに与えられる「例」の記述

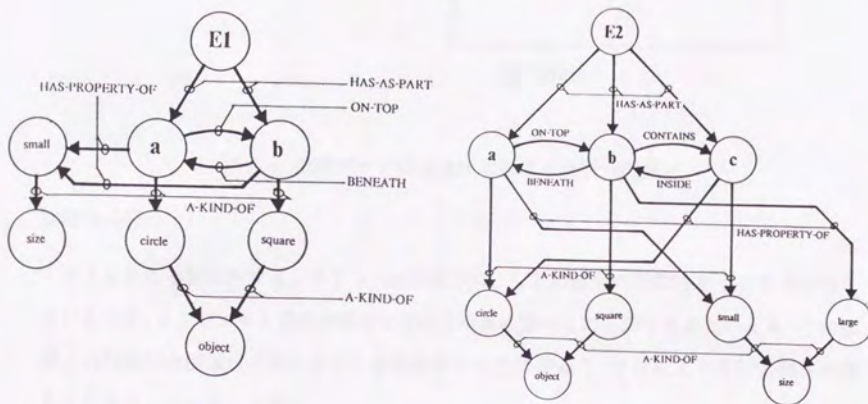
従来の構造記述学習システムでは、システムに与えられる例としては空間記述作成システムなどを用いて得た意味ネットワークなどのような形で与えるようになっていた。例えば Winston のシステムでは、図 C.3(a) の図形は図 C.3(b) で表される意味ネットワークの形でシステムに与える必要があった [12]。こうした場合、システムはこの意味ネットワークの記述しか参照できないため、これに記述されていないオブジェクトの関係は決して知ることができない。例えば図中の E2 の図形で、a と c というオブジェクトの位置関係を調べたいと思った場合、その間のリンクが張られていないため、システムはその間の関係を知ることができない。

この問題点に対処するため、ここで述べるシステムに与える例の記述は、固定的な意味ネットワークのような形式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中の存在位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が、これは容易に拡張可能である。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 C.3(a) の図形は図 C.3(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。

実際の学習プロセスにおいては、これらのオブジェクトを空間データ構造 BD 木 [35] で管理することにより、効率良い空間的な扱いが可能となっている。またより複雑な構造を持つ対象でも扱えるように、与えられた事例はある程度構造化した形によりその大まかな構造が扱えるように管理する。以下、これらの点について述べる。



(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 C.3: システムに与える「例」の記述

((a),(b) は文献 [12] より)

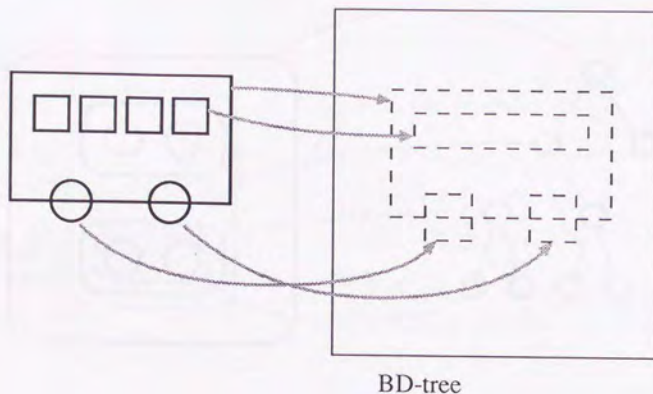


図 C.4: 空間データ構造 BD 木による空間性の扱い

空間性の扱い

与えられた事例に対するシステム内部の記述は、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関係を柔軟に調べることができるといえる。その反面この空間的な関係は必要に応じて直接調査する必要があり、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。

ここでは各オブジェクトを空間データ構造 BD 木 [35] で管理することにより、空間的な探索を高速に行えるようにしておいてこの効率の低下を最小限に抑えている。その簡単な概念図を図 C.4 に示す。BD 木の利用により、事例を構成するオブジェクトの個数 n に対し空間探索は $O(\log n)$ で実現可能である。

対象の大まかな構造の扱い

画像データは、その構造の複雑さから、ある対象の内部構造が知られていなかったり処理に無関係なため知る必要がなかったりする場合があります。このために対象をその大まかな構

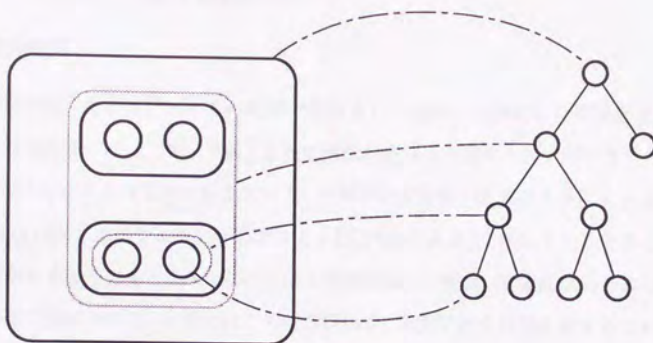


図 C.5: 部分的に構造化されたオブジェクトの表現

造で扱えるような機構が必要である。これを実現するための一手法として、与えられた画像データを幾何学的なオブジェクトでシンボル化する際に、これを部分的に構造化し、グルーピングされたオブジェクトとして扱う方法が考えられる。

本システムでは、このグルーピングを“align”と“inclusion”という関係に沿って実現している。こうして構成された構造を図 C.5に示す。図の左側がもとの図形を表し、右側がこれによる内部構造を表す。上方にあるノードが対象のより大まかな構造を表している。実際の学習プロセスでは、この構造を参照することにより必要以上に対象の細かい構造に捕らわれる必要がなくなる。

Winston のシステムなどにおける意味ネットワークによる記述では、学習プロセスはこの意味ネットワークに沿った関係しか参照できなかった。本システムでのこの部分的に構造化された記述は、あくまで学習プロセスの補助として参照されることを目的としており、大まかな構造を表すために部分的にグルーピングしているだけの記述である。実際のモデルの記述における図形間の関係は、その都度データ間の空間関係を調査することにより求められることになる。

C.3.3 システムで扱う構造記述

構造記述の分類

C.3.2 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 C.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って分類し、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。

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 $1 \sim n$ 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。

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。
- $1 \sim n$ 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。 $0 \sim n$ 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いずれ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「 $1 \sim n$ 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。

1 次空間関係の高度化

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。
- より複雑な構造を持った対象を、その大まかな構造で扱う。

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、
 - 4章で述べた状態遷移モデル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。

- 対象を大まかな構造で扱う — 対象を表す例の図形が複雑な構造を持つ図形で構成されているとき、学習プロセスでははじめはその大まかな構造に着目してモデルを生成しようとし、それに失敗したら図形のより細かい構造を見に行くようにするのが良い。本システムの例の記述では、この例を構成する幾何学オブジェクトのグルーピングによる部分的な構造化を行っているため、はじめはこの記述でより大まかな構造に着目し、認識プロセスが進むに従って少しずつ細かい構造を見るように制御することができる。

2 ~ n 次空間関係の高度化

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今後特に考察する必要がある。

C.3.4 学習アルゴリズム

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 [59].

- (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。

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つの例を満足し、すべての負の例を満足しない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る。具体的には、

- (1) もっとも一般的な true のみのルールを生成する。
- (2) 物体の存在を調べるルール (0 次空間関係) を付加する。
- (3) それぞれの物体の種類を調べるルール (1 次空間関係) を付加する。
- (4) 2つの物体間の位置関係を調べるルール (2 次空間関係) を付加する。
- (5) 物体を下位構造に分解し、2 からくりかえす。

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る。5 番目の手続により、始めは対象の大まかな構造に着目して構造記述を生成しようとし、これで矛盾のない記述が生成できなければ順次図形の構造の細かい部分まで見るようになる構造記述ルール生成が実現できる。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。

C.4 システムの実装と実行例

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。

システムに与えられた例の図形は、先に述べた部分的に構造化した記述に変換されて学習システムに与えられる。例えば、図 C.7 の 1 の図形による部分的に構造化された記述は図 C.6 のように表現できる。このような記述はやはり外部の一種の空間記述作成システムにより作成することになるが、ここでは 6 章で述べた多階層型の状態遷移モデルによる線図形認識システムにより作成した。認識システムには例の図形に記述されている対象のモデルは与えずに、これを構成する幾何学的なオブジェクトのモデルのみを用いて認識プロセスを行わせることにより、複数の階層にある程度の構造が得られた状態で不完全な認識結果が得られる。これをそのまま学習システムのための入力として用いた。

つぎに、システムの実行例を紹介する。まず図 C.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) :-
```

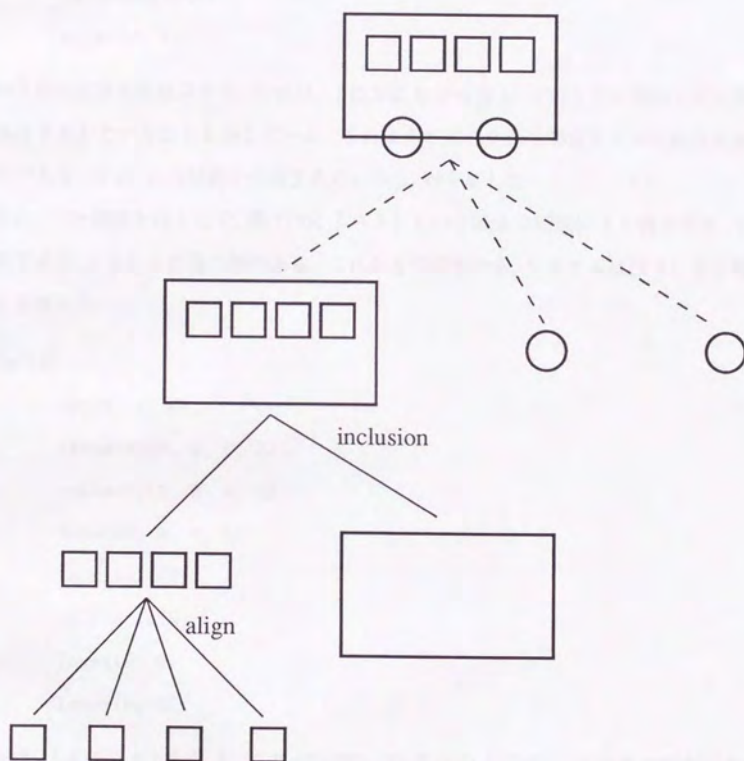



図 C.6: 実際の図形データから生成した部分的に構造化された記述

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

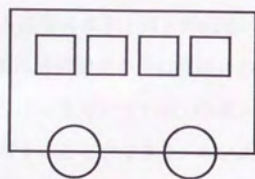
という構造記述が生成された。これは、「長方形 B が存在し、その上方に接触して三角形 D が存在する」ということを表している。これはまさに C.2 節で想定していた構造記述そのものである。なお、この記述を生成するのに 2cpu 秒を要した。

次に、やや複雑な例として、図 C.7 に「バス」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。

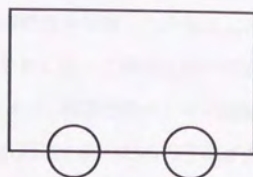
model(A) :-

```
obj(B, A, C),  
crossobj(D, B, C, E),  
crossobj(F, B, E, G),  
inobj(H, B, G, I),  
is_circle(F),  
is_circle(D),  
lower(F, B),  
lower(D, B).
```

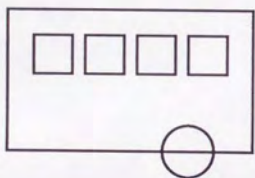
これは、「オブジェクト B, D, F, H が存在し、D, F は B と交差し、H は B の内部にあり、F と D は円であって、それぞれ B の下方に存在する」ということを表している。「バス」を構成する 2 つのタイヤ (F, D) の形状と位置については言及しているが、車体 (B) と窓 (H) の形状などには何も言及していないことが分かる。従って、この構造記述のままさまざまな形状の車体や窓を持ったものが「バス」として認識されることになる。なお、この記述を生成するのに 42.5cpu 秒を要した。



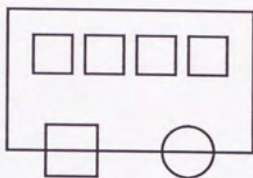
1. true



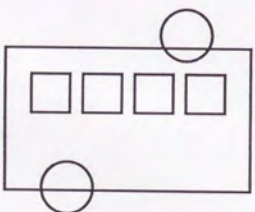
2. false



3. false



4. false



5. false

図 C.7: 「バス」の学習のための例

C.5 考察

図形の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、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この部分を拡充する必要がある。

索引

- ACRONYM, 17, 97, 119
- AI-Mudams, 68
- ANALOGY, 32, 183
- ATMS, 36
- BD 木, 68, 82, 150, 186, 190
—ライブラリ, 153
- GBD 木, 154
- GIS, 4
- IGS, 19, 97, 100
- IMPRESS, 18, 31
- Mapsee2, 16
- Marr-Hildreth のオペレータ, 117
- MARVIN, 32
- MIRACLE-IV, 18, 20, 33, 97, 100, 185
- Model Inference System, 169, 186
- Model-based Vision, 17
- model-based vision, 97
- SCERPO, 119
- SIGMA, 20, 97, 100
- STAR アルゴリズム, 172
- TMS, 36
- 一般化円錐, 17
- 一般画像, 86
—理解システム, 115
- 意味データモデル, 25
- 意味ネットワーク, 33, 183
- 演算, 41, 42, 88
- オブジェクト指向データモデル, 5, 24, 25, 56
- 型階層, 25, 27
- 型継承, 25, 27
- 関係データベース, 4, 24
- 関係データモデル, 24
- 関数型データモデル, 25
- 概念化, 10
- 学習アルゴリズム, 195
- 画像検索, 22, 86
- 画像データベース, 2, 11
- 画像認識・理解技術, 6, 10, 14
- 帰納学習, 31, 134

- 帰納推論, 163, 169, 182
協調, 97, 98
—処理, 15, 19
人間機械—, 31, 163
空間関係, 192
クラス, 27
グルーピング, 52, 104
高次モデル, 103, 111, 123
構造記述, 182, 192
構造記述ルール生成アルゴリズム, 196
コンピュータ学習, 30, 126, 163
—暗記による学習, 30
—観察および発見からの学習, 31
—指示による学習, 30, 129
—類推による学習, 30
—例からの学習, 31, 134
サブクラス, 27
サブシンボル, 103, 109, 119
初期階層, 104, 117
—の生成, 107
状態遷移型認識理解機構, 50
状態遷移図, 54
状態遷移モデル, 50, 51, 59, 62, 86, 97, 163
状態遷移ルール, 54
スーパークラス, 27
スキーマ, 16
線画, 106
—の解釈, 106, 136
線図形, 63
—の認識, 73
ゼロクロス, 117
対話
—ユーザとの—, 15, 17, 127
多階層状態遷移モデル, 97, 98, 104
多階層認識理解モデル, 97, 98, 100
知識表現, 15
地図図面, 63
—の認識, 77
—認識システム, 163
抽象データ型, 25, 29
抽象度階層, 104
抽象レベル, 15, 19, 97, 98
チョイスポイント, 71
データモデル, 2
—獲得, 2, 10, 11, 14, 50
—の要件, 33, 62
—管理, 2, 11, 24
—の要件, 45
—利用, 2, 11, 21

- の要件, 41
- トークン, 52, 63, 104
- トークンエクストラクタ, 66, 68
- トップダウン処理, 62
- トップダウンプロセス, 71
- 内部状態
 - を持つ計算モデル, 37, 50
 - トークンの—, 52
- ニアミス, 32
- 認識カーネル, 65, 66, 70
- 認識ルール, 65
- ハイパーテキスト, 21, 23
- ハイパーメディア, 4, 5, 21, 23, 42
- バックトラック, 71
- 不完全な認識結果, 41, 54, 62, 86
 - の利用, 35, 57
- 複合オブジェクト, 25
- フレーム, 15, 17, 66
- ブラウジング, 23
- プロダクションシステム, 15, 65
- ベクトル, 68
- ボトムアップ処理, 62
- ボトムアッププロセス, 70
- マルチメディアデータベース, 2
- メソッド, 29
- メッセージ, 29
- モデル, 182
 - 幾何的—, 15, 17
 - との照合, 97, 109, 119
- モデルベース, 182
- ルール作成支援, 163
- ルール作成支援システム, 126

