

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の開発
An Expert System for Top-Down
Image Segmentation

松山隆司 尾崎正治
Takashi MATSUYAMA Masaharu OZAKI
(京都大学 工学部)
Department of Electrical Engineering, Kyoto University

An expert system for top-down image segmentation is presented. A goal specification for the system includes an image feature(e.g. rectangle, line) to be detected from the image and a set of constraints on its properties(e.g. area, length). Given a goal, the system first reasons about the best process sequence to detect the specified image feature by combining primitive image processing operators. Then it executes image processing automatically according to the planned sequence. The reasoning and execution processes are controlled by several types of production rules to realize flexible trial-and-error image analysis. The knowledge organization and reasoning process of the system are described with some illustrative experiments.

1 はじめに

最近のデジタル画像処理技術の進展・普及には目を見はるものがある。その理由としては、画像処理用LSI等のハードウェアの進歩とともに画像処理ソフトウェアの充実が挙げられる。もちろん、現在の画像処理技術では、人間の視覚能力に匹敵するようなものを実現することは困難であるが、対象や問題をうまく限定すれば充分有効であることは、多くの実用システムが存在することから明らかである。

しかし、与えられた問題を解決するためにどのような画像処理手法を選択し、組合せればよいのかといった画像処理技術に関する知識は、画像処理の専門家のノウハウとして経験的に知られているにすぎない。このため、画像処理を1つの要素技術として利用しようとする応用分野の技術者は、そうした画像処理自身に関する知識を身につけない限り画像処理技術を使いこなせないのが現状である。この原因としては、画像から得られる種々の特徴を表現するために多様なデータ構造が必要であること、同一の処理を実現するためのアルゴリズムが多数存在し、問題に応じて有効なアルゴリズムが異なること、処理に必要なパラメータが多く、アルゴリズムを十分に理解していないと適切な値を指定できないこと、などが挙げられる。

こうした問題を解決するために、画像処理に関する知識を計算機上に明示的な形で表現し、それを利用したエキスパートシステムを開発しようとする動きが盛んになりつつある[1-3]。[1]では、画像処理用サブルーチン・パッケージSPIDER

[4]に含まれるサブルーチン間の引数の整合性を調べ、複数のサブルーチンを組合せた実行モジュールを作成する。[2]では、領域の分割・併合を行うための各種のヒューリスティックをプロダクション・ルールによって表現し、柔軟なセグメンテーションを実現している。また[3]では、初心者向の画像処理コンサルテーション・システムが提案されている。

本報告では、[5]で述べた画像理解システムSIGMAにおける1つのモジュールであるLLVE(Low Level Vision Expert)について述べる。図1にSIGMAの構成図を示す。GREは、認識対象のモデルとそれらの間の幾何学的関係に基づいた推論を行う。推論の途中で未発見の対象物があることが分ると、GREはMSE, LLVEにその検出を要求する。これがSIGMAにおけるトップダウン解析である。MSEはGREから要求された対象物の画像上での見え方を推論し、LLVEにその見え方に対応する画像特徴の検出を要求する(図2)。

このように本報告で述べる画像処理エキスパートシステム(LLVE, 以下単にシステムと書く)の目的は、外部(MSE)から指定された条件を満たす画像特徴(長方形、領域、線など)を画像から検出することにある。図2(b)の例では、対角線が(24, 30), (120, 230)で指定される座標軸に平行な矩形領域の中で面積が125から230の間であるような長方形を検出せよ、というのがシステムに与えられたゴール(要求)である。

一般に画像特徴「長方形」を画像から検出するには様々な方法が存在する。システムは、内部に蓄えられている画像処理に関する種々の知識に基づいて推論を行い、有効な基本画像処理モジュールを選択し、それらを組合せた処理方針を決

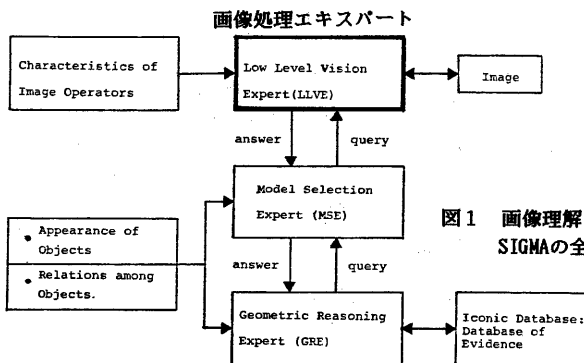


図1 画像理解システム SIGMAの全体的構成

```
((GOAL (AND (EQUAL OBJECT-TYPE HOUSE)
              (AND (LESSP AREA 475)
                    (GREATERP AREA 250))))
 (LOCATION (AND (LESSP X 1000)
                (GREATERP X 100)
                (LESSP Y 2000)
                (GREATERP Y 300))))
(CONTEXT (HOUSE-GROUP002, HOUSE-GROUP006, ROAD005)))

(a) GRE からMSEへの要求
((GOAL (EQUAL IMAGE-FEATURE-TYPE RECTANGLE)
 (AND (LESSP AREA 230)
       (GREATERP AREA 125)))
 (LOCATION (AND (EQUAL START-I 24)
                (EQUAL START-J 30)
                (EQUAL END-I 120)
                (EQUAL END-J 230))))

(b) MSEからLLVEへ要求
```

図2 トップダウン解析における要求

定する。次に決定された処理方針に従い画像処理モジュールを起動し、実際に画像を処理する。当然のことながら、実行の途中で特徴の抽出に失敗する場合がある。このときシステムは、失敗した状況に応じてアルゴリズムやパラメータ、あるいは処理モジュールの組合せを変更し、再度実行を試みる。こうしてゴールで指定された画像特徴が発見できるか、あるいは新たに実行すべき処理がなくなるとその結果を上位のシステムに返す(後者の場合は『失敗』を返す)。

本システムは、LISPマシンSymbolics 3600上で実現されており、現在のところ上位のシステム(GRE,MSE)とは結合されておらず、処理対象画像の指定およびゴールの設定は人間が会話的に行っている。しかし、これら以外の部分(推論および画像処理の実行等)は全てシステムが自動的に行う。

2 画像特徴・処理プロセスによる知識のネットワーク表現

画像から線、領域、角などの画像特徴を抽出するとき、一般的にどのような処理の過程を経るかを考える。たとえば濃淡画像から連続した線要素を抽出する場合、濃淡画像→(エッジ抽出)→エッジ画像→(しきい値処理)→エッジ点→(連結操作)→線要素といった一連の処理が考えられる。この例のように画像処理における処理過程の多くは、ある画像特徴に対して何らかの処理を施し別の画像特徴を求め、得られた画像特徴に次の処理を施すという一連の処理の並びであるということが出来る。この例では、濃淡画像、エッジ画像、エッジ点、線要素といった『画像特徴』が、エッジ抽出、しきい値処理、連結操作といった『処理プロセス』によって次々と処理・変換される。

以上の考察から、本システムでは、画像処理に関する知識

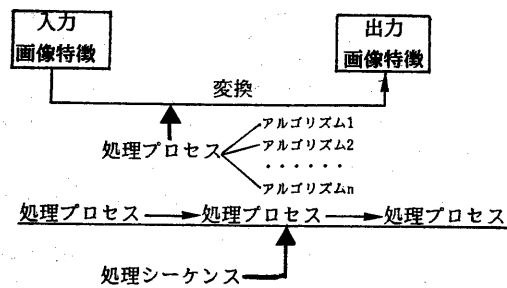


図3 画像特徴、処理プロセス、処理シーケンス

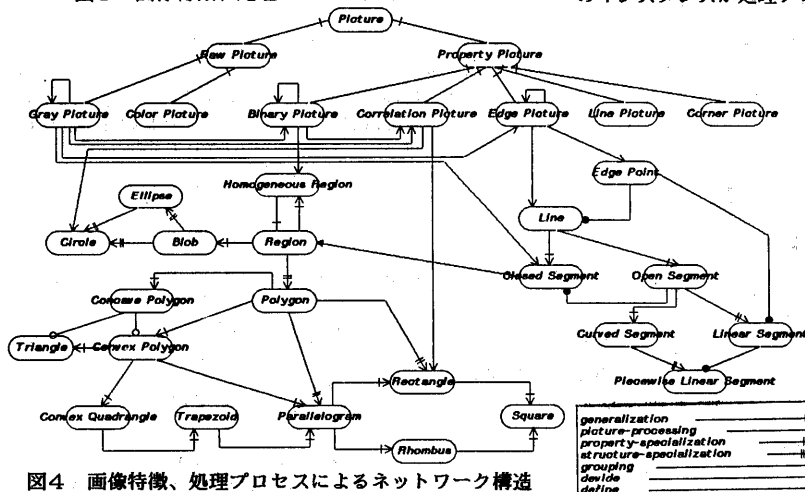


図4 画像特徴、処理プロセスによるネットワーク構造

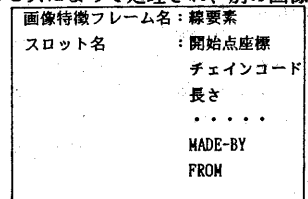
表現のための基本要素として、『画像特徴』および『処理プロセス』を採用した(図3)。個々の画像特徴は、画像の持つ情報を構造的に表現するための基本的な語彙を表すとともに、処理の対象となるシステム内の基本的なデータ構造を規定する。また、処理プロセスはある画像特徴を別の画像特徴に変換する操作として定義される。さらに、複数の処理プロセスの順序付けられた並びを『処理シーケンス』と呼ぶ(図3)。

画像から抽出できる画像特徴には多種多様なものがあり、種々の異なった処理プロセス、シーケンスが考えられる。したがって、各画像特徴を1つの節点として表し、各処理プロセスを節点を結ぶ弧と考えると1つのネットワークが形成される。図4は現在のシステムに蓄えられている画像特徴、処理プロセスによって構成されるネットワークを表す。ここで各槽円は1つの画像特徴、各矢印は1つの処理プロセスを表す。・や-の記号は矢印の先(処理の方向)を示し、後述するように処理プロセスのタイプを表す。一般にある画像特徴を求めたい場合、様々な処理シーケンスが考えられる。図4のネットワークでは、こうした同じ目的のための異なった処理シーケンスは、2つの離れた節点を結ぶ複数のパスとして表現されることになる。たとえば、濃淡画像(Gray Picture)から領域(Region)を求めるためには、大きく分けると均一領域(Homogeneous Region)を経るパスと線要素(Line)-閉曲線(Closed Segment)を経るパスとがある。

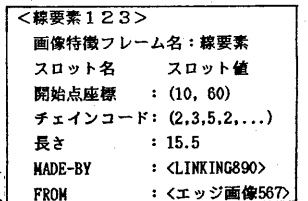
2.1 画像特徴の表現

本システムでは各画像特徴を1つのフレームによって表現し、画像特徴のフレーム表現を画像特徴フレームと呼ぶ。たとえば、線要素を表す画像特徴フレームは、図5(a)のように開始点座標、チェーン・コード、長さ等をスロットとして持つ。こうした画像特徴の属性を表すスロットのほか、全ての画像特徴フレームにはMADE-BY, FROMという2つのスロットが用意されている。MADE-BYのスロットにはその画像特徴を生成した処理プロセスが入れられ、FROMにはその処理プロセスの入力となった元の画像特徴が入られる。これらのスロットの値を参照することにより、ある画像特徴を求めるのに実行された処理の履歴を調べることができる(2.2参照)。

各画像特徴フレームによって定義されるのは画像特徴の『クラス』であり、実際に画像から抽出された画像特徴はそのクラスの『インスタンス』として表現される。図5(b)は線要素のインスタンスの例で、各スロットに具体的な値が入られている。実際の処理では、こうした画像特徴フレームのインスタンスが処理プロセスによって処理され、別の画像



(a) 画像特徴フレーム



(b) 画像特徴フレームのインスタンス

図5 画像特徴フレーム

特徴フレームのインスタンスに交換される。

複雑な画像を解析する場合、対象物認識のためには幾度となくトップダウン解析を行うことが必要となる。本システムでは、処理の途中あるいは過去においてなされた解析において得られた全ての画像特徴フレームのインスタンスを記憶している。こうした画像特徴フレームのインスタンスの管理を行うのが画像特徴フレーム総括管理モジュールである(図7参照)。このモジュールの機能としては、

- (1) 画像特徴フレームの新たなインスタンスを作る。
- (2) 現在管理している画像特徴フレームのインスタンスの中から、ある条件を満たすインスタンスを検索する。
- (3) ネットワーク構造(図4)における各画像特徴フレームについての接続関係を記憶し、要求があればその情報を返す。がある。

(2)の検索要求に付けられる条件は、“長さが10から20の線要素”などといった画像特徴の属性に関するものと、“ある特定の処理プロセスによって作られた線要素”などといった処理の履歴に関するものがある。また、外部から新たな要求が与えられた場合、システムはまずこの管理モジュールに対して、ゴールで指定された条件を満たす画像特徴のインスタンスがすでに検出されているかどうかを問い合わせ、もしそうであるならば直ちにそのインスタンスを結果として返す。もちろんこの場合は実際の画像処理は行われない。

2.2 処理プロセスの表現

処理プロセスとは、入力となる画像特徴から出力となる画像特徴を生成する操作であり、実際の画像処理プログラムがその中に書かれる。一方、1つの処理プロセスであるエッジ抽出を考えてみると、その操作を実現するためのアルゴリズムとしては非常に多くのものがある。すなわち、本システムでいう処理プロセスとは、画像処理における基本操作を機能レベルでとらえたもので、一般に1つの処理プロセスを実現するための方法としては多くの具体的なアルゴリズムが存在する(図3)。

また知識表現の観点からいうと、処理プロセスには様々なタイプが存在し、それらを明確に区別する必要がある。表1に処理プロセスのタイプ分けとその意味を示す。現在のシステムではdefined-by以外の意味付けは積極的に利用していない。図4では、defined-byのタイプの処理プロセスとしてはclosed segmentからregionへの処理プロセスがある。これは、closed segmentのインスタンスが作られると同時に、それによって定義されるregionのインスタンスが生成されなければならないことを意味している。

さらに処理過程の履歴の表現という観点から考えれば、実際に処理を行った後、その処理プロセスがどのようなアルゴリズムによって実行されたか、またそのアルゴリズムに対するパラメータはどのような値であったかを記録・参照できるようにしておかなければならない。

以上の考察から本システムでは処理プロセスの表現法としてもフレームを用いることになった。画像特徴フレームの場合とは異なり、全ての処理プロセスのフレーム表現は図6のような構造をしている。先の画像特徴フレームの場合と同様、図6は処理プロセスのクラスを表しており、実際に実行された処理プロセスはそのクラスのインスタンスとして表現される。さらに全ての処理プロセスのインスタンスは処理プロセス総括管理モジュール(図7参照)によって管理されており、必要に応じてその情報を調べることができる。

処理プロセスの実行(すなわち処理プロセスのインスタンスの生成)は次のようにして行われる。各処理プロセスには“activate”という手続きが用意されており、これが実際の画

像処理を起動する。手続きactivateを実行すると以下のような手順で処理が行われる。(activateを実行するのはシステム自身であり、システムの利用者がこの手続きを呼ぶわけではない。)

- (1) 今実行しようとしているクラスの処理プロセスの新たなインスタンスを生成する(各スロットの値は空)。
 - (2) 処理対象(入力)として与えられた画像特徴フレームのインスタンス(複数でもよい)を新たな処理プロセスのインスタンスの所定のスロットに格納する(図6)。
 - (3) 後に述べるルールによって適切なアルゴリズムとパラメータを選択し、それらを所定のスロットに格納する。
 - (4) 選ばれたアルゴリズムに対応するプログラムを実行する。
 - (5) 出力として得られた画像特徴(複数でもよい)を表すための画像特徴フレームのインスタンスの生成を画像特徴フレーム総括管理モジュールに要求する。このとき、新たな画像特徴フレームのインスタンスのMADE-BY, FROMのスロットに値を格納する(図5)。
 - (6) 処理に要した時間(秒数)を所定のスロットに格納する。
- 以上の過程によって、実行された処理プロセスに関する情報が処理プロセスのインスタンスのスロット値として表され、以降処理プロセス総括管理モジュールによって管理される。

3 システムの推論・処理方式

一般に、効果的な画像処理を実現するには、画像処理手法に関する様々なノウハウが必要となる。たとえば、画像の雑音レベルが高ければ平滑化を行うとか、エッジ画像上で線の追跡を行うにはまず非極大点の抑制を行ったほうがよい、などといったものである。さらに、画像処理には試行錯誤的な要素が多く含まれる[6]。つまり画像処理では、実際に処理

処理プロセス名: -----	
スロット名	: 処理プロセスのタイプ アルゴリズム パラメータ 入力画像特徴フレームインスタンス 出力画像特徴フレームインスタンス 処理時間

図6 処理プロセスのフレーム表現

タイプ名	機能
generalization ←———+———	ある画像特徴を一般化した画像特徴に変換するもの。 例: 一様領域から領域への処理プロセス
picture-processing ←———→———	単にある画像特徴を画像処理によって別の画像特徴に変換するもの。概念レベルの推移はない。 例: エッジ抽出
property-specialization ←———→———	ある画像特徴の一つのプロパティが何らかの条件を満たす場合、特殊化された画像特徴として作り出すもの。また、画像特徴のデータ構造はほとんど変化しない。 例: 線要素から曲線要素への処理プロセス
structure-specialization ←———→———	上記とほぼ同じものであるが、画像特徴のデータ構造が大きく異なるもの。 例: 線要素から直線要素への処理プロセス
grouping ———●———	複数の画像特徴を一つにまとめることで新しい画像特徴が作られるような処理プロセス。 例: エッジ点から線要素への処理プロセス
divide ———○———	一つの画像特徴から複数の画像特徴を分割して求めるもの。 例: 凹多角形から凸多角形への処理プロセス
defined-by ———→———	ある画像特徴が別の画像特徴が発見されたことによって定義されるような、二つの画像特徴への異なった視点をつなぐようなもの。 例: 閉曲線(線として)から領域(面として)への処理プロセス

表1 処理プロセスのタイプ

を行って、その結果に応じてパラメータやアルゴリズムを変更しなければならない場合が多くある。

一方、知識表現の観点からは、2で述べたネットワーク構造は、画像処理という対象世界の持つ構造を静的に表現したものであり、そうした静的な知識構造をどのように利用するかといったことに関する知識も必要である。ここでは、様々な画像処理に関するノウハウをネットワーク構造の利用法に関する知識として表現する方法および、そうした知識を利用したシステムの推論・処理方式を述べる。

3.1 システムの構成

本システムの構成を図7に示す。システム管理モジュールは、処理対象画像、ゴールの指定という外部とのインターフェイスであるとともに、下位のモジュールを総合的に管理・

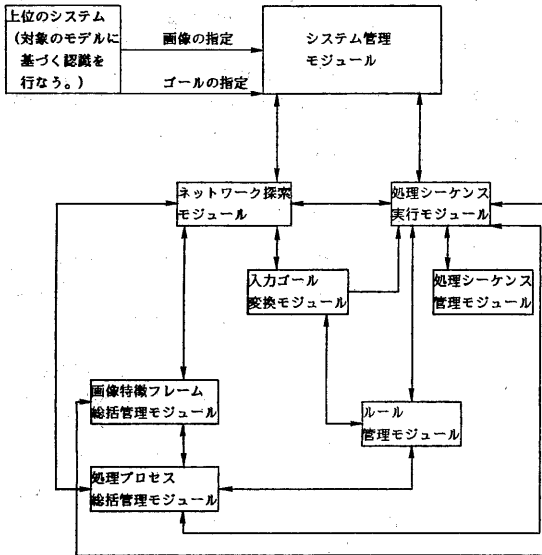


図7 エキスパートシステムの構成

タイプ	機能	例
処理プロセス限定のためのルール	ネットワーク探索時において処理シーケンス決定のために用いられる。	AT Region IF isolation is no, THEN select closed-segment-to-region
処理プロセス間の依存関係のルール	ネットワーク探索時において処理シーケンス決定のために用いられる。	AT edge-point-to-line IF noise-level is not low THEN select edge-picture-to-edge-picture
コスト計算のためのルール	ネットワーク探索において処理プロセスのコストを計算する。	AT edge-point-to-line IF algorithm equal linking complexity is high, THEN コスト計算のための計算式
ゴールの制約条件変換のためのルール	処理プロセスと逆方向に制約条件の変換を行なう。	AT region-to-polygon FOR area IF T THEN set polygon.area to region.area
アルゴリズム選択のためのルール	処理プロセス実行時にアルゴリズムを選択する。	AT edge-point-to-linear-segment IF accuracy is high, THEN select fine-hough-transform
パラメータ選択のためのルール	処理プロセス実行時にパラメータを選択する。	FOR thresholding IF failed before, THEN set threshold to former-threshold * 1.2
処理プロセス失敗時のルール	処理プロセスが失敗した時に次に行なう処理を指定する。	AT polygon-to-rectangle IF T THEN find-another-path from polygon

表2 ルールのタイプ

制御する。ネットワーク探索モジュールは、図4のネットワークを探索し、目的とする画像特徴を処理対象画像から抽出するための処理シーケンスを求める。処理シーケンス実行モジュールは、求められた処理シーケンスを実行し、もしそれが失敗したならば別の処理シーケンスを選び再実行する。実際の全ての画像処理は、このモジュールの管理の下で行われる。画像特徴、処理プロセス、処理シーケンス(総括)管理モジュールは、それぞれ画像特徴フレーム、処理プロセスのインスタンスおよび実行された処理シーケンス(処理プロセスのインスタンスの並び)を管理する。入力ゴール変換、ルール管理モジュールの機能は後に述べる。

本システムでは、上で述べたような各種の画像処理手法に関するノウハウをプロダクション・ルールによって表現している。表2に示すように、ルールには様々なタイプのもがあり、それぞれのタイプのルールが表す知識、適用される状況が異なる。(各ルールの意味、役割は以下適宜説明する。)図7のルール管理モジュールはこれら全てのルールを管理し、要求に応じてルールを選択・適用する。

3.2 処理対象画像、ゴールの指定

起動された直後、システム内には画像特徴フレームのインスタンスは1つも存在しない。そこでまず処理対象となる画像を(計算機のファイルに蓄えられているものの中から)指定する。指定された(原)画像は、そのタイプに応じてGray PictureあるいはBinary Pictureの画像特徴フレームのインスタンスとしてシステムに登録される。この際、その画像の持つ特徴を同時に与える。図8は図11(a)の画像に対応するGray Pictureのインスタンスの記述で、画像データが格納されている配列名のほかに雑音レベル、コントラストなどの特徴記述が与えられる。以後新たな処理対象画像を指定しない限り、全ての処理はこの画像を対象として行われる。

つぎにゴールの指定であるが、ゴールは大きく分けると次の5つの部分から構成される。(図9)

- (1) 対象画像から抽出したい画像特徴フレーム名
- (2) 処理対象範囲(位置的制約条件)
- (3) 目的とする画像特徴の属性に対する制約条件

```

#GRAY-PICTURE 26340462
MADE-BY          : ORIGINAL
FROM            : NIL
GRAY-LEVEL-ARRAY-ID : #ART-00-020-160 26210550
GRAY-LEVEL-ARRAY-TYPE : ART-00
OFFSET          : (0 0)
NOISE-LEVEL    : LOW
TEXTURE        : PARTIALLY
SCENE-TYPE     : NATURAL
CONTRAST       : HIGH
    
```

図8 処理対象画像を表す画像特徴フレームGray Pictureのインスタンス

```

Goal Frame : RECTANGLE
Goal Properties :
AREA-SIZE (INTEGER) :
(200 400)
ISOLATION (SYMBOLIC) :
YES
TEXTURE (SYMBOLIC) :
LOW
NOISE-LEVEL (SYMBOLIC) :
INTERMEDIATE
CONTRAST (SYMBOLIC) :
HIGH
COMPLEXITY (SYMBOLIC) :
LOW
ACCURACY (SYMBOLIC) :
HIGH
ALLOWABLE-COST (SYMBOLIC) :
HIGH
WINDOW (COORDINATE) :
((227 40) (293 72))
    
```

図9 ゴールの記述

(4)目的とする画像特徴の周囲の環境

(5)ゴール自身の特徴(制約条件)

(1)としては、図4に示した画像特徴フレームの1つを指定する。(2)としては目的とする画像特徴が存在すると考えられる座標軸に平行な矩形領域を指定する。ゴールの指定が終わると、システムは直ちに処理対象画像からこの矩形領域を切り出し、以後はこの部分画像に対して実際の処理が施される。この切り出し操作は、Gray pictureからGray Pictureへの(自己ループ)処理プロセスとして表されており、切り出された部分画像はGray Pictureの1つのインスタンスとなる。

(3)では、(1)で指定した画像特徴フレームの持つ属性に対する制約条件が指定される。どのような属性があるかは目的とする画像特徴の種類に依存する。たとえば、長方形に対しては面積、線に対しては長さといった制約が付けられる。こうした属性に対する制約条件はその下限と上限の値を指定することによって表現する。

(4)としてはISOLATION, TEXTURE, NOISE-LEVEL, CONTRAST, COMPLEXITYといった目的とする画像特徴の周囲の画質や他の画像特徴との関係を指定する。たとえば、ISOLATIONはYES, NOで指定し、目的とする画像特徴が孤立しているかどうかを表す。その他のものに対しては、HIGH, INTERMEDIATE, LOWのいずれかを指定する。

(5)は、ゴールとして表された要求がどのようなものであるかをACCURACY, ALLOWABLE-COST, TEMPLATEの項目によって表現する。ACCURACYはどの程度の精度で画像特徴を抽出したいか、ALLOWABLE-COSTはどの程度の処理コストまで許すか、それぞれHIGH, INTERMEDIATE, LOWで指定する。たとえば、目的とする画像特徴がほぼ確実に存在することが分かっている場合には、コストがいくらかかってもそれを抽出せよと要求することが考えられる。TEMPLATEは、テンプレート・マッチングによって画像特徴を抽出せよという要求を表し、マッチングに用いるテンプレート画像を指定する。(1), (2)以外の制約条件は必要がなければ指定しなくてもよい。

3. 3 ネットワークの探索による処理方針の決定

システム管理モジュールは、ゴールが与えられると、ネットワーク探索モジュールに対して、ゴールで指定された画像特徴を抽出するための処理方針(シーケンス)の決定を要求する。ネットワーク探索モジュールは、図4のネットワークから、処理コスト、処理の有効性の観点から最適と思われる処理シーケンスを選び、システム管理モジュールに返す。

3. 3. 1 最小コスト探索

処理方針を表す処理シーケンスは、トップダウン的にネットワークを探索することにより決定される。すなわち、ゴールで指定された画像特徴フレームを出発点として、処理プロセスを実際の処理の方向とは逆の方向に探索してゆき、最終的に処理対象画像を表す画像特徴フレーム(一般にGray Picture)に至るコスト最小の処理シーケンスを求める。(注:後に述べるように、探索がGray Pictureに到達するまでに停止される場合もある。また、ここでいう処理シーケンスは、2で述べた本来の処理シーケンスとは逆向きに処理プロセスを並べたものを表す。)

最小コスト探索アルゴリズムを以下に示す。

(1)ゴールで指定された画像特徴フレームを注目する画像特徴フレーム、注目する処理シーケンスを空とする。一般に、現在注目している画像特徴フレーム、処理シーケンスをそれぞれF、PS: (P1, P2, P3) (P1-P3は処理プロセス)とすると、PSは、ゴールで指定された画像特徴フレームを出発点とし、今注目している画像特徴フレームを終点とする(処理プロセスを逆方向に辿った)ネットワーク上のパスを表す。すなわ

ち、処理プロセスP1の出力がゴールとなる画像特徴フレーム、P3の入力が今注目している画像特徴フレームFとなる。

(2)注目する画像特徴フレームFを出力とする処理プロセス(一般に複数)を求め。

(3)得られた各処理プロセスに対して、予想される処理コストを計算する。このとき、表2に示したコスト計算のためのルールが起動され、それぞれの処理プロセスを実行するのに必要な計算量が推定される。一般に、推定される計算量は、処理対象画像の特徴、ゴールの制約条件に応じて変化する。

(4)現在注目している処理シーケンスをPS: (P1, P2, P3)、(2)で得られた処理プロセスの集合を{S1, ..., Sn}とする。PSにS1, ..., Snをそれぞれ付け加えた新たな処理シーケンスPS1: (P1, P2, P3, S1), ..., PSn: (P1, P2, P3, Sn)を作り、記録する。(元の処理シーケンスPSは記録から除去される。)また、(3)で求めたS1, ..., Snの計算コストをPSの計算コストに加えたものをPS1, ..., PSnの計算コストとする。

(5)現在記録されている全ての処理シーケンスの中からコストが最小のものに注目する。今NPS: (NP1, ..., NPn)を新たに注目する処理シーケンスとする。このとき処理プロセスNPnの入力となる画像特徴フレームを新たに注目する画像特徴フレームとする。

(6)新たに注目した画像特徴フレームが処理対象画像を表すものであるか、あるいはその画像特徴フレームのインスタンスがすでに存在し(以前の処理で抽出されたもの)、それがゴールで指定された制約条件を満たすならば、現在注目している処理シーケンスを結果として返す。そうでなければ(2)に戻る。

上の(6)において、ネットワーク探索中に以前の処理において抽出された画像特徴フレームのインスタンスを利用することにより、同じ処理を何回も繰返す必要がなくなり、処理の無駄をなくせる。また、ゴールの制約条件は、目的とする画像特徴フレームの属性に対するものであり、ネットワーク探索の途中で発見された画像特徴フレームのインスタンスに対するものではない。このため、ネットワーク探索モジュールは、上の(5)において新たに注目する画像特徴フレームを求めた場合、表2のゴールの制約条件変換のためのルールを起動し、当初ゴールの画像特徴フレームに対して与えられた制約条件を新たに注目する画像特徴フレームに対するものに交換する(実際の変換は入力ゴール変換モジュールが行う)。この際変換されるのは、3. 2のゴール記述の(3)の項目のみで他のものはそのまま新たな画像特徴フレームに対する制約条件となる。

3. 3. 2 ルールによる処理プロセスの選択

前節で述べた最小コスト探索のみでは、処理シーケンスの選択基準として充分であるとはいえない。本システムでは、処理コストに加えて、与えられた状況における処理プロセスの有効性をもう1つの選択基準としている。具体的には、表2に示した(a)処理プロセス限定のためのルール、(b)処理プロセス間の依存関係のルールによって、上の最小コスト探索の過程を制御している。

まず(a)のルールは、上のアルゴリズムの(2)において適用され、注目する画像特徴フレームFを出力とする処理プロセス群のうち、現在の状況から考え有効と思われるもののみを選び、他のものは以降の探索から除外する(探索ルートの枝刈り)。ルールの条件部(現在の状況記述)は、注目する画像特徴フレームに対する(ルールによって変換された)制約条件、処理対象画像の特徴によって記述される。

(b)のルールは、個々の処理プロセスに対して記述されており、上のアルゴリズムの(2)において新たな処理プロセス

が得られた場合に適用される。このタイプのルールは、今新たに得られた処理プロセスを実行する前に別の処理プロセスが実行されていなければならないことを表す。たとえば、図10(a)に示すように、新たにPS1という処理プロセスが(2)で得られ、それに関するルールが処理プロセスPS2を要求しているとする。このルールが適用されると、図10(b)のようにPS1の入力画像特徴フレームとPS2の出力画像特徴フレームを結ぶ処理シーケンスがネットワーク探索モジュールを再帰的に起動することによって求められる。この結果、上のアルゴリズムの(4)において新たに求められる処理シーケンスとしては、今まで注目していた処理シーケンスにPS1,PS3,PS4,PS2(図10(b))が同時に付け加えられたものとなる。もちろん、ルールが存在しない場合やルールの条件が満たされない場合にはPS1のみが付け加えられる。またコストとしては、新たに付け加えられた全ての処理プロセスの計算コストの合計が今までのコストにたされる。

以上述べた方法によって、ゴールで指定された画像特徴フレームから実際の処理対象データとなるインスタンスを持つ画像特徴フレーム(ほとんどの場合Gray Picture)までの"逆向き"の処理シーケンスが求まる。これが第一処理方針となる。(後で述べるように、この処理シーケンスが失敗した場合、システムは、ネットワーク探索を再起動し、第二の処理方針—2番目にコストの小さい処理シーケンス—を求める。)

3.4 処理シーケンスの実行

ネットワーク探索モジュールからシステム管理モジュールに返される処理シーケンスを(P1,P2,...,Pn)とする。このとき、ゴールで指定された画像特徴フレームが処理プロセスP1の出力画像特徴フレーム、Pnの入力画像特徴フレームには処理対象となるインスタンスが存在する。またこのシーケンスの途中に現れる全ての画像特徴フレームには、ゴールの制約条件からルールによって変換された制約条件が付けられて

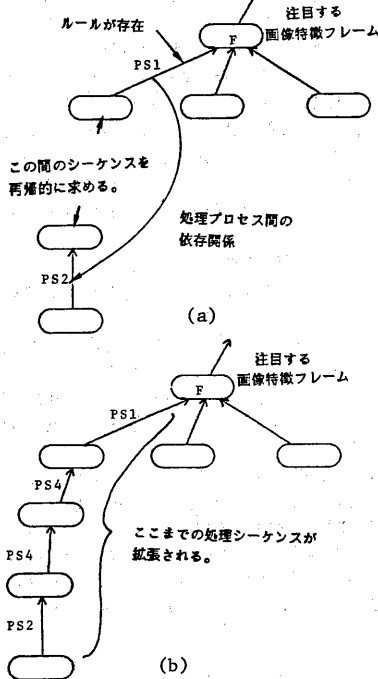


図10 処理プロセスの依存関係

いる。システム管理モジュールは、このシーケンスを処理シーケンス実行モジュールに渡し、実行を要求する。

処理シーケンス実行モジュールはPnから逆に1つずつ処理プロセスを"activate"し、実際の画像処理を行う。2. 2で述べたように、ある処理プロセスにおける手続きactivateが起動されると、まずそのプロセスに対するアルゴリズム選択のためのルールが適用され、最適なアルゴリズムが求められる。アルゴリズムが決まると、パラメータ選択のためのルールが適用され、処理の実行に必要なパラメータの値が決定される。アルゴリズム・パラメータ選択に利用できる情報としては、その処理プロセスの出力画像特徴フレームに対する(ゴールから変換された)制約条件、処理対象となる入力画像特徴フレームのインスタンスの属性、処理対象画像の特徴がある。さらに、今与えられたゴールを求めるためにそれまでに実行された処理プロセスの履歴(処理プロセスのインスタンスの属性値)も利用できる。この情報は、後に述べるように、あるプロセスが失敗した場合にアルゴリズムやパラメータの値を変えて再度実行する際に利用される。

処理シーケンス実行モジュールは、処理プロセスの実行によって制約条件を満たす出力画像特徴フレームのインスタンスが生成されると、そのプロセスは成功したとして、新たに生成されたインスタンスを入力データとして次の処理プロセスをactivateする。一方、そのようなインスタンスが生成されない場合は、次節で述べる処理プロセス失敗時のルールを適用し、それによって選ばれたアルゴリズム、パラメータ、あるいは処理シーケンスを変更して処理を再開する。

3.5 特徴抽出に失敗したときの動作

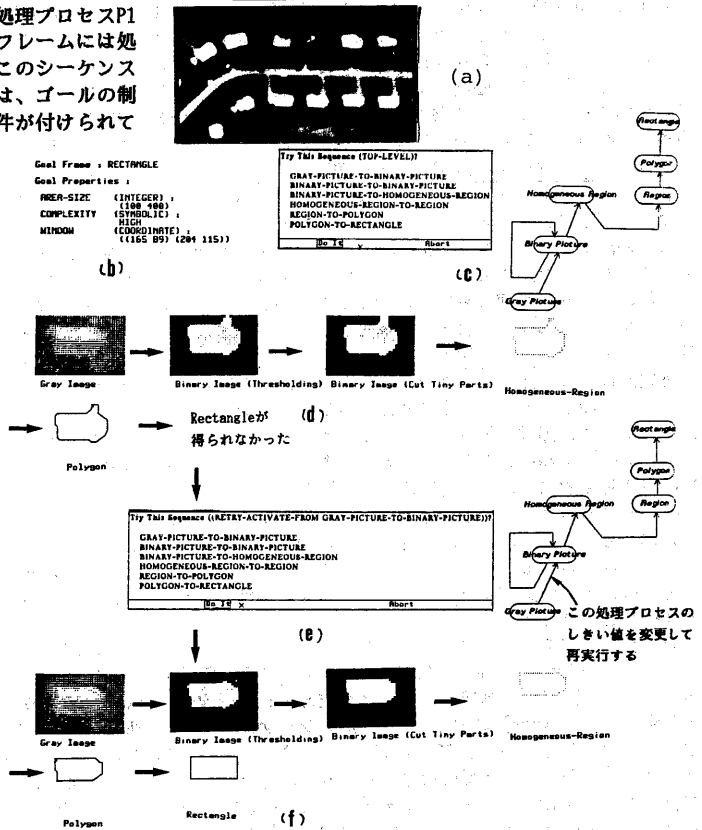


図11 実行例1

以前にも述べたように、画像処理には試行錯誤的な要素が多く含まれる。本システムでは、処理プロセスの実行によって制約条件を満たす出力画像特徴フレームのインスタンスが生成されないこと(失敗)を1つの情報としてとらえ、より良い処理方式を見出すために積極的に利用している。

ある処理プロセスが失敗すると、処理プロセス実行モジュールはその処理プロセスに対応する失敗時のルールを適用する。このルールの条件部で利用できる情報には、(a)失敗した処理プロセスのインスタンスの属性(アルゴリズム、パラメータ、入力画像特徴フレームのインスタンス)、(b)失敗するまでに実行された処理プロセスのインスタンスの並び、がある。ルールの実行部では次の3種類のものが行える。

(a)RETRY:これは、今失敗した処理プロセス自身をアルゴリズム、パラメータを変更して再度実行することを意味する。この場合、新たに実行するアルゴリズム、パラメータの値はこのルール中で直接指定する。

(b)RETRY-ACTIVATE-FROM:これは、今までに実行された部分的な処理シーケンスを調べ、その中で"弱い"と思われる処理プロセスを選び、そのアルゴリズム、パラメータの変更を要求し、その処理プロセスに戻って再度実行することを意味する。この場合、アルゴリズム、パラメータの変更は、選ばれた処理プロセスのアルゴリズム・パラメータ選択のルールに従って行われる。

(c)FIND-ANOTHER-PATH:これは、今までに実行された部分的な処理シーケンスによって生成されたインスタンスを持つ画像特徴フレームから、ゴールである画像特徴フレームまでの別の処理シーケンスの発見をネットワーク探索モジュールに要求し、それを実行することを意味する。

一般に、ある処理プロセスが失敗したときに適用できるルールは複数ある。このルールの選択は、ルール管理モジュールが行い、最初に適用できるルールを1つ処理プロセス実行モジュールに返す。このルールによって変更された処理シーケンスがまた失敗する場合も考えられる。その場合は、新たに失敗した処理プロセスに関する失敗ルールが適用される。また、変更された処理シーケンスが完全に失敗した(目的とする画像特徴が抽出できず、適用するルールもない)場合は、初めに処理プロセスが失敗した状態に戻り、そこで適用できる第二のルールがあればそれを適用する。適用できるルールが全くない場合は、ネットワーク探索モジュールに対し、処理方針としての処理シーケンスの次の候補の探索を要求し、得られたシーケンスを新たに実行する。

また次のような場合には、ゴールを満たす画像特徴が存在しないとしてシステム全体の処理が終了する。

(a)全ての可能な処理シーケンスが失敗し、これ以上実行できる処理シーケンスが存在しない。

(b)今までの処理シーケンスの実行に要した処理時間の合計が、ゴール自身の特徴であるALLOWABLE-COSTで指定された値を越えた。(現在この値は、HIGH:300, INTERMEDIATE:200, LOW:100(秒)としている。)

処理プロセスの失敗により処理シーケンスが変更され、さらに変更後の処理シーケンス中の処理プロセスの失敗により再び処理シーケンスが変更されるといった場合、多くの処理シーケンスが再帰的に実行される。処理シーケンス管理モジュール(図7)は、これまでに実行された処理シーケンスを全て記録・管理しており、同じ処理シーケンスが再び実行されるのを防いでいる。

4 実行例

図11(a)は白黒の航空写真で、[5]において用いたものと同じ画像である。図11(b)は、家の画像上での見え方を表す長方形の検出を要求するゴールを表す。図11(c)はネットワーク探索モジュールによって求められた第一の処理シーケンスを表す。ここでBINARY-PICTURE-TO-BINARY-PICTUREという2値画像の雑音除去の処理プロセスが含まれているのは、BINARY-PICTURE-TO-HOMOGENEOUS-REGIONの処理プロセスに書かれている処理プロセス間の依存関係のルールが起動されたためである。その意味は、2値画像から均一領域を抽出する前には、2値画像中の細かな雑音を除去しておく必要があるということである。また、HOMOGENEOUS-REGION-TO-REGIONは表1のGENERALIZATIONのタイプの処理プロセスで、この2つの画像特徴の差はほとんどない。さらに、POLYGONから(PARALLELOGRAMを経ずに)直接RECTANGLEに至る処理プロセスが選ばれているのは、ゴールのCOMPLEXITYをHIGHとしたため、ネットワーク探索において処理プロセス限定のためのルールが起動されたためである。

図11(d)は、処理シーケンスの実行によって得られた中間的な画像特徴フレームのインスタンスを表す。2値化、2値画像の雑音除去、均一領域の抽出、多角形近似、が次々と実行され図に示す多角形が得られる。しかし、この多角形がPOLYGON-TO-RECTANGLEの処理プロセスで処理された結果、長方形が検出できず(この多角形を長方形と近似するには誤差が大きすぎた)、処理プロセスが失敗する。そこでこの処理プロセスに対する失敗ルールが起動される(図11(e))。このルールはRETRY-ACTIVATE-FROMのタイプのもので、これまでに実行された処理シーケンスの中にしきい値処理(GRAY-PICTURE-TO-BINARY-PICTURE)というパラメータの値に敏感な処理プロセスが含まれているので、このパラメータを変更してしきい値処理から処理をしないことを指示する。

しきい値処理のアルゴリズムに対応するパラメータ選択ルールでは、以前用いたパラメータの値をその処理プロセスのインスタンスから求め、その値を変更して(現在のルールでは単純に1.2倍する)、再度処理を実行する。図11(f)が再実行された処理シーケンスによって抽出された画像特徴フレームのインスタンスである。この場合は、うまく多角形を長方形と近似することができ、得られた長方形がゴールの制約条件(面積が100と400の間)を満たすことから、それが答えとして返される。

図12は同様な長方形の抽出例であるが、ゴールとしてISOLATIONがNOと指定した(COMPLEXITYはLOW)。このため、探索の結果得られた処理シーケンス(図12(b))は、図11とは異なりエッジ抽出、線追跡によって領域の検出を行うものとなっている。また、COMPLEXITYがLOWであるため、POLYGONからPARALLELOGRAMを経てRECTANGLEに至るシーケンスとなっている。これらは、ネットワーク探索中に処理プロセス限定のためのルールが適用されたためである。さらにEDGE-PICTURE-TO-EDGE-PICTUREの処理プロセスは、エッジ画像における非極大点の抑制を表し、EDGE-PICTURE-TO-LINEの処理プロセスに書かれている処理プロセス間の依存関係のルールによって選ばれた。

得られた処理シーケンスを実行した結果、線の中から閉曲線であるものがみつからずLINE-TO-CLOSED-SEGMENTの処理プロセスが失敗する(図12(b))。このため失敗時のルールが起動される。このルールはFIND-ANOTHER-PATHのタイプで、LINEからゴールであるRECTANGLEまでの新たな処理シーケンスの発見を要求する。この結果、LINEからOPEN-SEGMENTを経てCLOSED-SEGMENTを求め(線分の連結を行う)RECTANGLEに至るシーケンスが得られ、その実行によって閉曲線、ゴールで

ある長方形がうまく抽出できた(図12(c))。

5 おわりに

本稿では、画像処理に関する知識を利用して、要求された画像特徴を自動的に抽出するエキスパート・システムを提案し、そのための知識表現、推論機構について述べた。本システムでは、画像のセグメンテーションにおいて必要な基本的な概念を画像特徴、処理プロセスによって表し、単なるコンサルテーションを行うのではなく、実際の処理を起動しその結果を評価することによって、新たな推論を行う。こうした試行錯誤的な推論過程は画像処理にとって特に重要であると言える。

現在稼働中のシステムの開発にあたっては、システム全体の基本機能の充実を第一に考えたため、システム内の処理プログラム、抽出できる画像特徴、各種のルールはごく基本的なものに限られている。ルールの数は400ほどあるが、そのほとんどは、単にネットワーク探索を実行するために必要なもので、真に意味のある知識を表すルールは数十程度である。また、ここでは述べなかったが、画像特徴フレーム、処理プロセスの追加、更新のための会話型エディタが開発されており、知識投入の簡易化を図っている。

今後検討の必要な点としては次のようなものがある。

- (1) 2. 2で述べたように、処理プロセスには現在7つのタイプがあるが、DEFINED-BY以外のタイプは利用していない。少なくとも意味の明確なGENERALIZATION, SPECIALIZATIONのタイプは積極的に利用すべきである。
- (2) 単純な画像特徴を組合せた複合図形を認識したい場合は、このシステムの上位にそれらのモデルに従って単純な画像特徴の抽出を制御するシステムを作る必要がある。
- (3) 現在のネットワーク探索は、ゴールで指定された画像特徴フレームからの完全なトップダウン探索によって行っているが、処理対象画像の特徴を積極的に用いたボトムアップ的な探索も組み込む必要がある。すなわち、画像の性質によって効果的な処理シーケンスやアルゴリズムが異なるため、処

理対象画像の特徴を処理シーケンスの決定にもっと積極的に利用しようというものである。このためには、画像データの持つ特徴を詳細に記述するための十分な語彙を定める必要がある。

(4) 画像処理のノウハウとして、幾つかの異なった処理による複数の処理結果を組合せることにより、処理効果、精度の向上を図るという考え方があ。たとえば、解像度の異なる画像集合の利用、エッジ抽出と領域分割の組合せによる領域の検出などである。このような処理戦略に関する知識の表現、利用法は、今後の大きな課題である。

(5) 以前有効に働いた処理シーケンスを記録しておき、新たなゴールが与えられた場合、そうした過去の経験を利用することにより、効果的な処理方針の決定を行う。こうした学習の問題は、システムの推論や処理の効率を向上させるうえで重要な問題となる。

最後に、本研究を行うのにあたり、終始適切な助言と建設的な議論をしていただいた京都大学長尾真教授をはじめ長尾研究室の方々に感謝致します。

【参考文献】

- [1] 坂上, 田村: DIA-Expertシステム構造処理部の開発, 情報処理学会研究会資料, CV28-4, 1984.
- [2] Nadif, A.M. and Levine, M.D.: Low Level Image Segmentation: An Expert System, IEEE Trans., Vol. PAMI-6, No. 5, pp. 555-577, 1984.
- [3] 星, 末田, 三亀: 画像処理エキスパートシステム, 情報処理学会第30回全国大会, 6L-6, pp. 1491-1492, 1985.
- [4] 田村他: ボータブル画像処理ソフトウェアパッケージSPIDERの開発, 情報処理学会論文誌, Vol. 23, No. 3, pp. 321-328, 1982.
- [5] 松山, Hwang, V.: 画像理解におけるトップダウン, ボトムアップ解析の統合, 電子通信学会研究会資料, AL84-47, 1984.
- [6] 長尾, 試行錯誤的の形状認識方式とそのハードウェア化, 電子通信学会論文誌D, 1985.

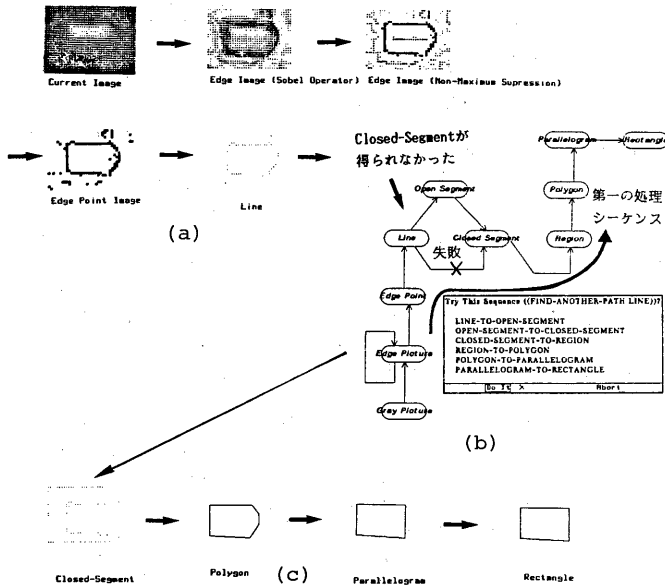


図12 実行例2