

論文 / 著書情報
Article / Book Information

題目(和文)	二次元図形の記述と認識に関する研究
Title(English)	
著者(和文)	麻田治男
Author(English)	
出典(和文)	学位:博士(工学), 学位授与機関:東京工業大学, 報告番号:乙第3278号, 授与年月日:1999年2月28日, 学位の種別:論文博士, 審査員:
Citation(English)	Degree:Doctor (Engineering), Conferring organization: Tokyo Institute of Technology, Report number:乙第3278号, Conferred date:1999/2/28, Degree Type:Thesis doctor, Examiner:
学位種別(和文)	博士論文
Type(English)	Doctoral Thesis

二次元図形の記述と認識に関する研究

麻 田 治 男

目次

第1章 序論	1
1.1 図形の形状記述の意義	1
1.2 形状表現の方法論	2
1.3 図形の形状記述に対する要請	2
1.4 本研究の目的	4
1.5 本論文の構成	5
第2章 Curvature Primal Sketch による図形輪郭の階層的記述	7
2.1 輪郭ベースの図形記述方式の従来研究	7
2.2 本研究における輪郭分割のアプローチ	7
2.3 図形輪郭のプリミティブの定義	8
2.3.1 Corner	10
2.3.2 Smooth Join	12
2.4 その他の複合型プリミティブ	15
2.4.1 端点 (End)	15
2.4.2 クランク (Crank)	16
2.4.3 バンプ (Bump)	17
2.5 Scale Space 内でのプリミティブの探索	17
2.6 処理例	22
2.7 3次元曲面の解釈への応用	30
2.8 まとめ	31
第3章 Smoothed Local Symmetries による図形のネットワーク記述	33
3.1 領域ベースの図形記述方式の従来研究	33
3.2 Local Symmetry (局所対称) の定義	34
3.3 パラメータ化と記号化	35
3.4 S L S 計算アルゴリズム	35
3.4.1 輪郭セグメントの円弧・直線近似	36
3.4.2 2つの輪郭セグメントから生成される S L S の解析的計算方法	37
3.5 S L S の計算例	47
3.6 負の S L S	48
3.7 図形のネットワーク表現	50
3.8 部分図形の検出	52

3.8.1	図形の「Side」と「End」	52
3.8.2	部分図形接合の検出	53
3.9	S L S の再計算	55
3.10	図形の階層的ネットワーク表現	56
3.11	まとめ	56
第4章	輪郭記述に基づく手書き文字認識	59
4.1	文字認識の従来研究	59
4.2	認識方式の概要	60
4.3	方向コード列の抽出	60
4.4	等価方向コード変換	62
4.5	文字図形の輪郭セグメント特徴系列の作成	63
4.6	認識辞書の構成とマッチング方式	66
4.7	認識実験	68
4.8	まとめ	70
第5章	高速図形処理アーキテクチャ	71
5.1	デジタル画像処理の問題点	71
5.2	デジタル画像処理高速化の流れ	73
5.3	画像処理ハードウェアの設計問題	75
5.3.1	画像処理機能の選択	75
5.3.2	高速化の諸方策	77
5.4	高速図形処理ハードウェア	78
5.4.1	空間フィルタ演算部 (Spatial Filter)	78
5.4.2	論理フィルタ演算部 (Logical Filter)	81
5.4.3	領域番号付け演算部 (Catalog Mapper)	81
5.4.4	輪郭追跡回路	85
5.5	高速図形処理装置の処理速度	85
5.6	まとめ	88
第6章	結論	89

第1章 序論

1.1 図形の形状記述の意義

図形の形状記述 (Shape Description) はパターン認識やコンピュータビジョンの中で基本的な問題である。パターン認識、特に文字・図形認識は、図形パターンから抽出された形状特徴を用いて精度良く分類することを目的としている。分類能力の高い識別方式の開発も重要であるが、特徴抽出の過程において、形状特徴を正確かつ効率的に表現できる記述方式が望まれており、そのための研究がなされてきた。一方、コンピュータビジョンにおいては、エッジ抽出や、直線抽出、距離測定、オプティカルフローなどのローレベルビジョンと、モデルとのマッチングなどのハイレベル処理とを結びつける部分、すなわち、三次元世界の理解に結びつけることのできる形状記述を得るための二次元画像の解析方法が重要である。これは、[Marr 76] が提唱した画像理解の枠組みの中で指摘されていることである。しかしながら、この分野の研究は必ずしも活発であったとはいえない。これまでのところ、実用に供することのできるロバストなビジョンシステムを構築する上での最大の障害が、ローレベル処理の性能（処理速度、精度、ロバスト性）であり、これらの克服に力が注がれてきたからである。現在では、計算機およびその周辺の発達によって処理能力が向上したこととあいまって、実時間で動作できる物体追跡や画像監視システムが作られるようになってきている。これらはローレベル処理を駆使したものであるが、今後、より複雑な要求に対処できるビジョンシステムへのニーズが高まることが予想され、より高度な認識処理を支えることのできる、形状記述の方式の重要性が今後ますます増加すると考えられる。

本研究では、文字や図形の認識、および、ロボット視覚における物体の認識を目的として、そのために有用な形状記述のあり方について考察し、新しい方法を提案する。さらに、三次元曲面の記述への拡張、および、文字認識に応用した結果を述べるとともに、実際の画像データに対して高速に処理するための高速図形処理アーキテクチャを提案する。

1.2 形状表現の方法論

これまでのビジョンシステムの多くで用いられている図形表現の方式は、図形の中心、領域や穴の数、主軸のアスペクト比、周囲長の二乗と面積の比、などの広域的な特徴が主体であった。これらの特徴は簡単にかつ高速に計算することができ、ノイズや量子化に強く、低解像度の二値画像に対しても計算可能であり、これらの計測量を用いて対象物の認識を行うことができる。しかしながら、このような方法を適用するには対象物が分離されている必要があり、利用範囲が限定される。特に、対象物が重なりあっているような場合にはこれらの特徴は全く無力である。図形全体に対して計算した値と、部分に対する値との間には関係は全くないからである。

局所的な特徴を表現することのできる、さらに詳細な図形記述の方法論は以下のように3種類に分類できる。図形はそれが空間に占める領域と背景、およびそれらの境界（＝輪郭）から成り立っている。従って、どの要素に着目するかによって3種類のアプローチが考えられる。

まず、図形の輪郭に着目したアプローチでは、輪郭の形状をコード化する。例えば、フーリエ級数展開、チェインコード表現、区分的円弧・直線近似[上坂 84]やスプライン近似[Pavlidis 74]がある。一方、領域に着目するアプローチは図形が占有している2次元空間の形状を表現するものである。この方法の例としては、Quad-trees、Symmetric axis transform(SAT)[Blum 78]、2次元 Generalized Cylinder [Brooks 81]が挙げられる。また、背景に着目した方法は主として文字認識の分野で考えられており、Glucksmanの方法、場の効果法などがある。

いずれのアプローチにも長所と短所がある。つまり、輪郭ベースの方法では、局所的な特徴は細かく記述できるが、それらの二次元的な位置関係を簡潔に表現することは困難である。そのため、図形の大局的な形状、つまり、対称であるとか、長細いとか、曲がっているという特徴を Explicit に表現できない。一方、図形の領域や背景の解析法では単一図形の2次元的な広がりはいまよく記述できるが、微細構造を表現するにはコストがかかる。

1.3 図形の形状記述に対する要請

過去の研究を基に、文字・図形認識や物体認識における形状記述に対する要請をまとめると、次の3つになる。

- **部分図形の階層表現：**

認識とは、図形や物体のモデルのデータベースが与えられたとき、観測によって得られた対象の表現をモデルと対応づけすることである。多くの場合、背景や他の図形から分離されているものを認識対象としている。しかし、対象物を背景から分離するためには対象物を認識できていなければならない、そのためには対象物を分離しなければならないという自己矛盾に陥る。そのため、地図のように線図形と文字やシンボルが混在するような状況では、局所領域の特徴量を用いるなど別の観点から文字やシンボルを分離しているが、これは精度低下を招く。本来は文字やシンボルの分離と認識とを同時に行うことが望ましい。

また、物体認識の場合には物体間のオクルージョンに対応する必要がある。ロボット視覚において、部品の山の中の個々の部品を認識する場合にこのような状況は頻繁に発生する。重なり合っていない部分の形状記述から、元の図形を認識できなければならない。重なり合わない部分がどこなのかは予め決められない。各々の部分の形状が的確に表現されている必要があり、それをモデルとの部分的な対応づけに使える必要がある。過去の例では、Bolles & Cain [1982] は、"local features" の概念を導入して、限られた範囲ではあるが、重なりのあるケースを扱った。彼らの認識の枠組みでは、対象物体のモデルはグラフで与えられ、グラフのノードにはコーナや穴、という特徴を表現し、2つのノードを結ぶエッジには両ノードの特徴部品の距離と相対方向を記述している。認識は、最大 Cliques をモデルであるグラフの中に見つけるという処理で行う。Bolles & Cain の表現方式は広域特徴の表現と local support のある表現との妥協案であるが、計量的には精密であり、簡単な特徴のみを用いているという利点がある。一方、直線で構成された輪郭をもつ形状のみを扱えるという制約がある。

- **ローカルサポート：**

また、認識とは似て非なる処理として、検査（インスペクション）がある。画像処理の産業応用としての部品の検査は、損傷があるかどうかを調べることであるが、そこでは、部品のモデルと、観測した形状との間で厳密なマッチングが得られることは期待できない。損傷のある観測形状がどのモデルであるかを認識して、その損傷個所を決定する必要がある。損傷とは通常、局所の形状変化であるので、図形表現の方式としては局所の変形があっても、全体の表現が大きく変わることなく、しかも、局所の変形そのものにはセンシティブであること

が要求される。このような性質をここでは、「ローカルサポートがある」ということにする。

- **クラスをわたる階層表現：**

モデルのクラスには階層があり、それを表現することができる必要がある。活字文字「2」にはゴシック、明朝など多くのフォントがあるが、人間はそれらの形状の違いは知りつつ、「2」というカテゴリであることは認識できる。また、タックハンマーはウォーリントンハンマーやボールピーンハンマーと比べると明らかに異なる形状をしているが、これらは全てハンマーであることに違いはない。セマンティックネットワークを使えば、これらの物体のクラスやサブクラスを系統立てて表現することができるが、そのような概念構造に対応できる、実際の形状記述の体系が必要である。

1.4 本研究の目的

本研究で提案する方法は、輪郭ベースと領域ベースの各々の特長を生かして、これらをネットワークの形で表現してローカルサポートの性質を付け加え、さらに、階層性をもたせることで、状況に応じて必要な情報を取り出して、多様な認識ができるだけの豊富な情報量をもつ、「リッチな表現」を実現することを目指している。

従来から Quad-tree にも見られるように画像を階層的に表現しようという試みはあったが、単に機械的に階層化するのではなく、図形の部分パターンがもつ物理的な意味によって、階層化を行うべきである。物の形はそれを見る位置を変えると変化するように、物を見る解像度（スケール）を変えることにより、観測される図形の特徴も変化する。本研究では、特徴的な事象（信号）をモデル化し、その事象の存在を Scale Space [Witkin 1983] の中で解釈し、その結果から階層的な記述を得るという新しいアプローチの表現形式を提案する。

一般的にいて、ノイズに埋もれた信号の同定と位置決めという問題は、粗い平滑化は信号の発見に強いが位置精度は悪く、逆に、細かい平滑化を使うと位置精度は良いが、信号の発見は不安定になるという不確定性原理に支配されている。Scale Space はこの点を考慮したものであり、Optimal Edge Finder[Canny 1983] や Marr-Poggio ステレオの根底となっている考え方である。このような考えに基づいて、大きなスケールで検出される事象（信号）は細かい位置の変動には影響されないような認識処理やクラスの同定に用い、小さなスケールで検出される事象（信号）は、サブクラス間の識別や、検査の段階での損傷の発見などに用いるという考えは妥当である。本研

究ではこのような考えに基づいた階層記述を与えることが目的である。

1.5 本論文の構成

第2章では、図形輪郭における曲率の主要な変化を Scale Space の中で解析することによって得られる図形輪郭の記述の方式を提案する。この記述方式を Curvature Primal Sketch と呼ぶ。曲率の変化の形態（つまり不連続性）を分類し、それを輪郭のプリミティブとして定義し、これらのプリミティブを Scale Space の中で検出するという手法を用いており、輪郭ベースの階層的記述が得られる。さらにこの方法を三次元曲面の記述に応用して、曲面上の曲率の不連続（線）の検出が可能であることを示す。第3章では、領域ベースの図形表現として、Smoothed Local Symmetries について述べる。特に、Curvature Primal Sketch によって得られた輪郭表現を用いて輪郭を直線と円弧で近似し、その結果から効率的に計算するアルゴリズムを提案する。さらに、この領域ベースの形状記述と輪郭ベースのそれとを結合することにより図形の階層的ネットワーク記述方式を構築する。これは本研究が目的とする表現形式である。第4章では、輪郭記述の応用例として、手書き文字認識システムについて述べる。図形輪郭を主要な曲率変化に基づいてセグメント化し、この系列を特徴として自由手書き数字を認識する。実用的な認識精度が得られることをシミュレーションによって示す。第5章は第2～4章で述べた図形記述方式と文字認識方式を実際の画像データに対して実行するために必要な高速図形処理のアーキテクチャについて述べる。前処理として必要な二値化処理、エッジ検出、平滑化、ラベリング処理を高速に行うことのできる局所並列型の図形処理のアーキテクチャと、輪郭追跡のためのハードウェアを提案する。

第2章 Curvature Primal Sketchによる図形輪郭の階層的記述

2.1 輪郭ベースの図形記述方式の従来研究

図形輪郭の解析を認識のための前処理として捕らえる立場では輪郭をいくつかのセグメントに分割し、各セグメントを直線あるいは円弧で近似する方法が取られることが多い。また、曲線のデータ圧縮を目的とする立場もある [名倉 1979]。これらの場合、どのようにセグメントをパラメータ化するかは要求される近似の精度や経済性で決まり、直線、円弧、Cornel Spiral、などのうちから適当なものを選べばよい。しかし、最も重要なことはセグメントへの分割点をどのように定めるかということである。[Pavlidis 74] は split & merge の手法により区分的な直線近似を求めている。また、[上坂 84] は全曲率関数の折れ線近似の区分点をダイナミックプログラミングもしくは逐次近似法で求めている。その他、[Perkins 78]、[Tsai 80a] では曲率の極大点を適当に閾値処理して区分点としている。これらの方法ではわずかな変形に対して安定な分割が得られるという保証はなく、閾値は対象に応じて適応的に決めなければならない。これは分割点の選択に際して、物理的な意味が考慮されていないからと考えられる。ここではまず分割点は物理的に見て、どのような点であるべきかということから考えてみる。

2.2 本研究における輪郭分割のアプローチ

曲線を分割するのに最も適した点は「角（かど）」である。角は曲率が極大となる点といわれているが [Rosenfeld 76]、これは曲率の値がある程度輪郭に沿って平滑化されている場合のことで、本来的には曲率の値は無限大である。むしろ、角では接線の方が不連続であるといった方が正確である。次に重要な点に変曲点であり、この点を境にして曲率の正負が変わっている。変曲点を一般化した Smooth Join では接線の方が連続だが、曲率が不連続であると定義できる。これらの不連続点は、我々人間が図形を見たときにまず知覚される特徴であるという理由で、図形の形

状を特徴づける上で重要な意味を持ち、輪郭を分割する点としてふさわしいものであるといえる。

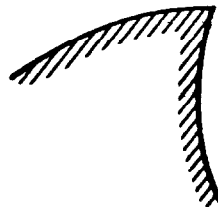
このように角や変曲点ではある種の不連続性があるが、実はこれらを見つけるという問題は、画像の中で明るさのエッジを求めることと同じ問題である。[Marr 74]、[Marr 76] は画像理解の計算機プログラムは画像そのものではなく画像の記号的解釈すなわち Primal Sketch を見るべきだと主張し、数種類の明るさの変化のモデルを立てて記号的解釈を試みた。エッジ型とバー型のマスクと画像とのコンボルーションの中でモデルに適合した出力を探し、記号表現を得る方法である。また、ノイズに対するロバスト性を向上させるため、2種類の大きさのマスクを用い、2つの出力間でマッチングのとれるもののみを記号化している。これはスケールスペース的考え方の基礎となるものである。本論文では Marr の方法論に従い、図形輪郭のプリミティブを定め、それらに対応した応答を数種類の大きさのマスクの出力の中で検出し、輪郭の記号表現を得るというアプローチを取る。

さて、信号の次元にマスクの大きさ（スケール）の次元を加えた表現形式は Scale Space[Witkin 83] と呼ばれている。Witkin は零交差の Tree 表現を考えたが、このような考えは Marr の考えに由来するものであることは明らかである。また、Witkin の場合には、ある事象の解釈を行うところまではいっていないが、ここでは主要な曲率変化を具体的なモデルとして、Scale Space をその解釈を行う場としてとらえる点が異なる。また、Yuille&Poggio [1983a, b] は2次微分の零交差系列には元の信号を再現するのに十分な情報が保存されていることや、零交差が大きな Scale で突然発生することがないという性質を保証するマスクは Gaussian だけであるなどの ScaleSpace に関する重要な定理を証明している。

2.3 図形輪郭のプリミティブの定義

前節で述べたよう、輪郭における曲率の主要な変化は「角 (Corner)」と「Smooth Join」である。Corner では接線方向に不連続性があり、Smooth Join では接線方向は連続であるが曲率に不連続性がある。これらを基本として図形輪郭のプリミティブを選ぶと、図 2.1 に示すものが妥当である。図 2.1 には Primal Sketch における明るさの変化のプリミティブとの対応付けも示されている。ここで、端点、クランク、バンプは複数の Corner が近接してできた複合体である。以下において、これらのプリミティブのモデル化を行って、どのような特性を持つのかを調べる。

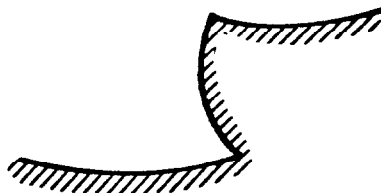
Corner



End



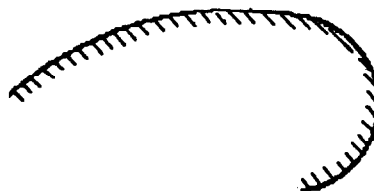
Crank



Bump (or Dent)



Smooth Join



Inflection

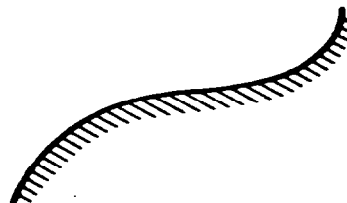


図 2.1: 図形輪郭のプリミティブ。Marr が定義した明るさ変化との対比は、ステップエッジ (Corner)、ラインエンディング (End)、細いバー (Crank)、グラディエントエッジ (Smooth Join) である。

2.3.1 Corner

図 2.2 (a) はふたつの円弧からできた角 (Corner) であり、各々の円弧の曲率を k_1 、 k_2 、角度変化を ϕ とする。図 2.2 (b) は周長 s と接線方向との関係を示すグラフである。

この関数 f_{corner} は、

$$f_{corner}(s) = \begin{cases} k_1 s + c & \text{if } s < 0 \\ k_2 s + c + \phi & \text{if } s > 0 \end{cases}$$

と書ける。これを標準偏差 σ のガウシアン $G_{\sigma(t)}$ とのコンボリューションの 1 次微分および 2 次微分、つまり、ガウシアン の 1 次微分フィルタの出力と、2 次微分フィルタ出力を計算すると次のようになる。

$$\begin{aligned} \sqrt{2\pi}\sigma(G_{\sigma} * f_{corner})'(s) &= \phi \exp\left(-\frac{s^2}{2\sigma^2}\right) + k_2 \int_{-\infty}^s \exp\left(-\frac{t^2}{2\sigma^2}\right) dt \\ &\quad + k_1 \int_s^{\infty} \exp\left(-\frac{t^2}{2\sigma^2}\right) dt \\ \sqrt{2\pi}\sigma(G_{\sigma} * f_{corner})''(s) &= \frac{-\phi}{\sigma^2} s \exp\left(-\frac{s^2}{2\sigma^2}\right) + (k_2 - k_1) \exp\left(-\frac{s^2}{2\sigma^2}\right) \end{aligned}$$

図 2.2 (c) と (d) はある σ の値についてこれらを図示したものであり、Corner に対するガウシアン の 1 次微分フィルタの出力は単一のガウス型ピークであり、ガウシアン の 2 次微分フィルタの出力は、異なる符号を持つ一対のピークであることがわかる。二つのピークの間距離は簡単に計算できて、

$$d_{corner} = \frac{\sigma^2}{\phi} \sqrt{(k_1 - k_2)^2 + \frac{4\phi^2}{\sigma^2}}$$

となる。特に、 $k_1 = k_2$ (これを pure corner と呼ぶ) の場合には、

$$d_{pc} = 2\sigma$$

である。つまり、ピーク間の距離は ϕ にはよらず、 σ の 2 倍となる。この性質は重要であり、 σ を変化させるに従って収束するようなピーク対を探すことにより、近接した Corner を分離して、独立した Corner を見つけることができる。また、ピークの高さは $|\phi|$ に比例し、次の式で与えられる。

$$h_{pc} = \frac{1}{\sqrt{2e\pi}} \frac{|\phi|}{\sigma^2}$$

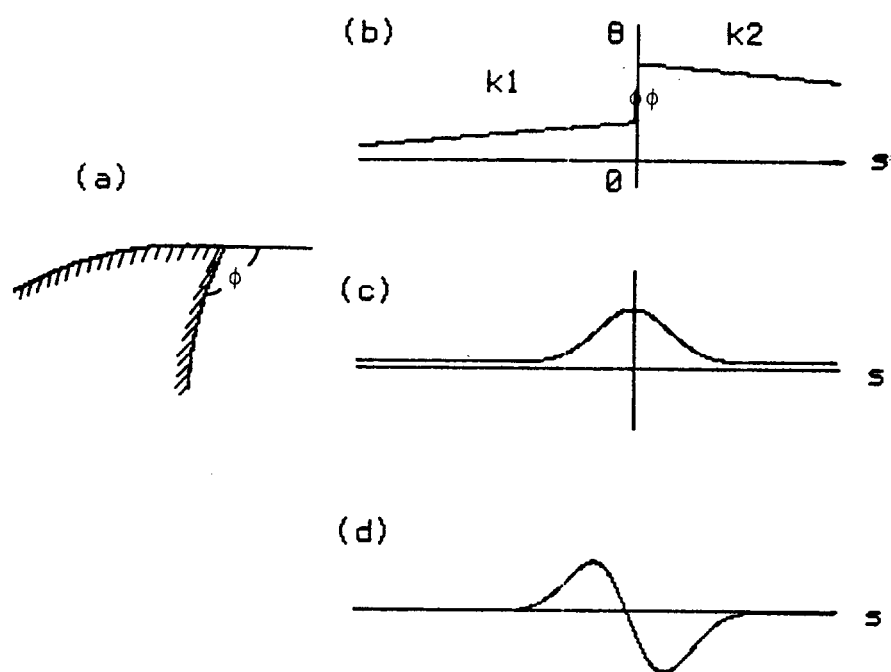


図 2.2: Corner のモデル:(a)Corner の形状、(b) 接線方向関数、(c) ガウシアン1次微分フィルタの出力、(d) ガウシアン2次微分フィルタの出力

2.3.2 Smooth Join

Smooth Join とは接線方向が連続で、曲率が不連続な点であり、図 2.3(a) にその形状を示す。変曲点 (Inflection) は曲率の符号がその点で変化するもので、Smooth Join の特殊な形である。その接線方向関数 f_{sj} は、

$$f_{sj}(s) = \begin{cases} k_1 s + c & \text{if } s < 0 \\ k_2 s + c & \text{if } s > 0 \end{cases}$$

のように表現できる。ガウシアン の 1 次微分フィルタ出力と 2 次微分フィルタ出力は Corner の場合の出力に $\phi = 0$ を代入したものとなるので、

$$\begin{aligned} \sqrt{2\pi}\sigma(G_\sigma * f_{sj})'(s) &= k_2 \int_{-\infty}^s \exp\left(-\frac{t^2}{2\sigma^2}\right) dt \\ &\quad + k_1 \int_s^{\infty} \exp\left(-\frac{t^2}{2\sigma^2}\right) dt \\ \sqrt{2\pi}\sigma(G_\sigma * f_{sj})''(s) &= (k_2 - k_1) \exp\left(-\frac{s^2}{2\sigma^2}\right) \end{aligned}$$

となる。図 2.3 にこれらを図示するが、2 次微分フィルターの出力は単独のピークを持つ。ピークの高さは次の式で与えられる。

$$h_{sj} = (k_2 - k_1) / \sqrt{2\pi}\sigma$$

Corner の場合と比べるために等価角度変化 $\phi_{sj} = (k_2 - k_1)\sigma\sqrt{e}$ を導入すると、

$$h_{sj} = \frac{1}{\sqrt{2e\pi}} \frac{|\phi_{sj}|}{\sigma^2}$$

となり、Corner と同じ形式が得られる。これからわかることは、 σ と曲率変化 $(k_2 - k_1)$ との積が十分に大きくなければ Smooth Join を安定に見つけることは困難だということである。

また、ふたつの Smooth Join が近接しているような場合、たとえば図 2.4(a) に示す、丸みのある角は大きな σ では図 2.4 (b) と区別がつかない。しかし、 σ を小さくすれば、前に述べた Corner のピーク対の収束性からこれらを区別することができ、このことは人間の感覚ともよく合っているとと言える。

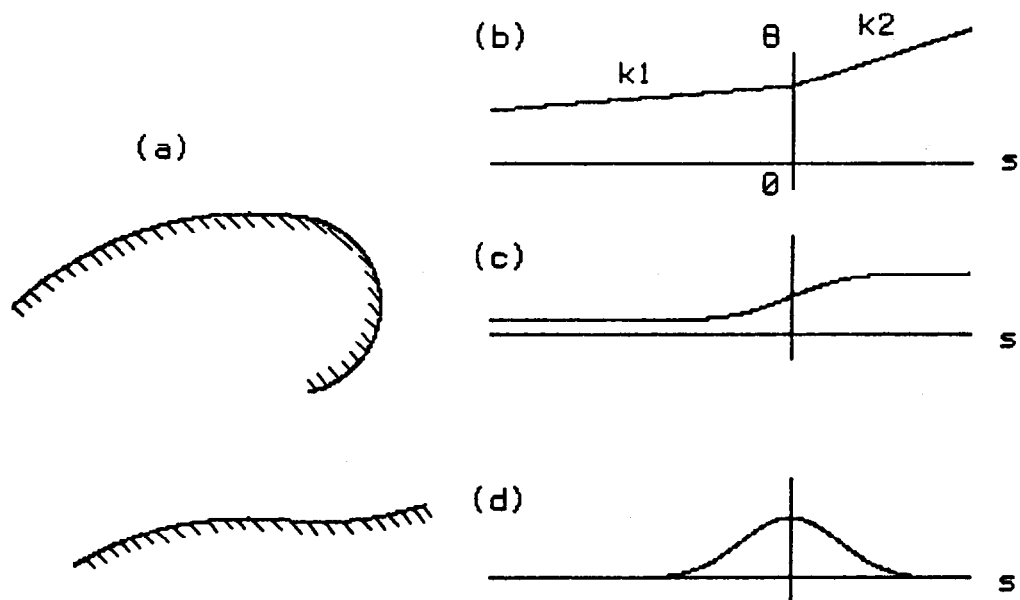


図 2.3: Smooth Join のモデル: (a) Smooth Join の形状 (2 種類)、(b) 接線方向関数、(c) ガウシアン 1 次微分フィルタの出力、(d) ガウシアン 2 次微分フィルタの出力

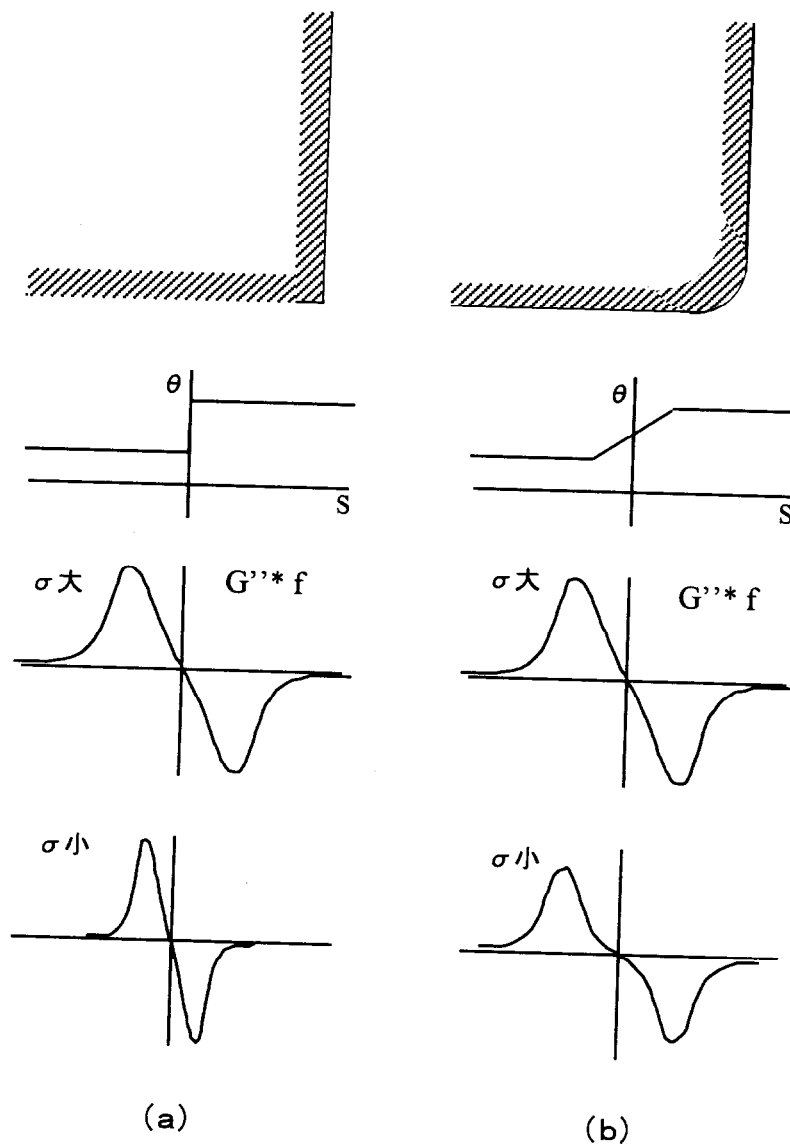


図 2.4: 丸みのある Corner : (a) ふたつの近接した Smooth Join で作られた丸みのある Corner ではスケールが小さくなってもピークの位置は移動しない (b) 角のある Corner ではピークの位置が原点に収束する

2.4 その他の複合型プリミティブ

2.4.1 端点 (End)

端点は同符号の角度変化の Corner がふたつ近接しているもので、図 2.5(a),(b) にそのモデルを示す。図 2.5(c) は異なる σ でのガウシアン 2 次微分フィルタの出力を示したものである。端点のエッジの長さを a とすると、 σ/a が 0.5 よりも大きいときは一對のピークとなり、これは Corner 出力と区別できない。しかし、 σ/a が 0.5 よりも小さくなると、ふたつの独立した Corner の出力が得られる。

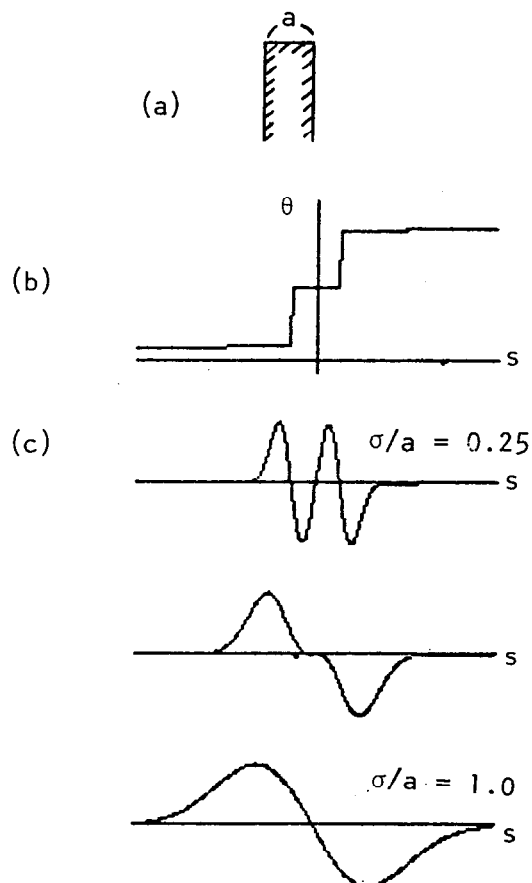


図 2.5: 端点 (End) のモデル : (a) 端点の形状、(b) 接線方向関数、(c) ガウシアン 2 次微分フィルタの出力

2.4.2 クランク (Crank)

クランクは符号が異なるふたつの Corner で構成されたもので、図 2.6(a),(b) にそのモデルを示す。図 2.6(c) は異なる σ でのガウシアン 2 次微分フィルタの出力を示したものである。端点の場合と同じようにエッジの長さを a とすると、 σ/a が 0.5 よりも大きいときはソンプレロ型であるが、 σ/a が 0.5 よりも小さくなると、ふたつの逆向きの Corner の出力が得られる。

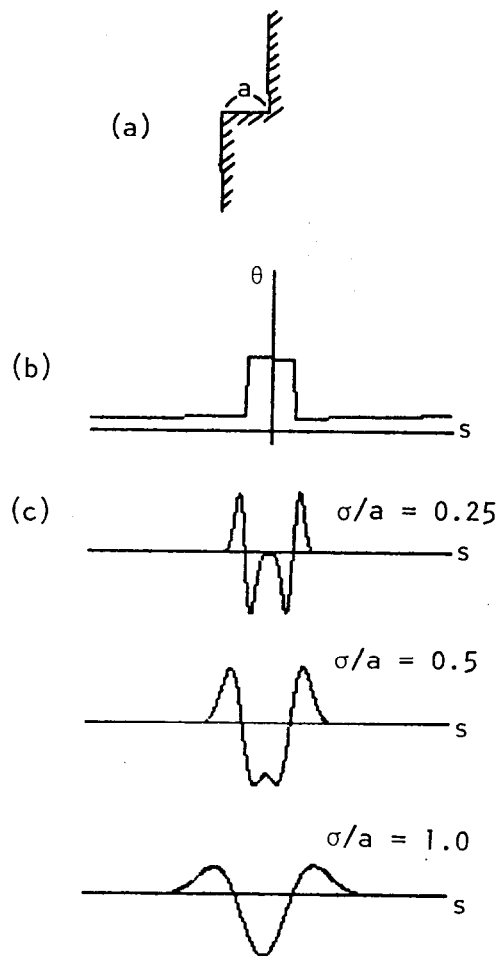


図 2.6: クランク (Crank) のモデル: (a) クランクの形状、(b) 接線方向関数、(c) ガウシアン 2 次微分フィルタの出力

2.4.3 バンプ (Bump)

バンプ (出っ張り)、あるいはデント (へこみ) は3つの Corner で構成されており、図 2.7 に示すように4つ組のピークを出力とする。

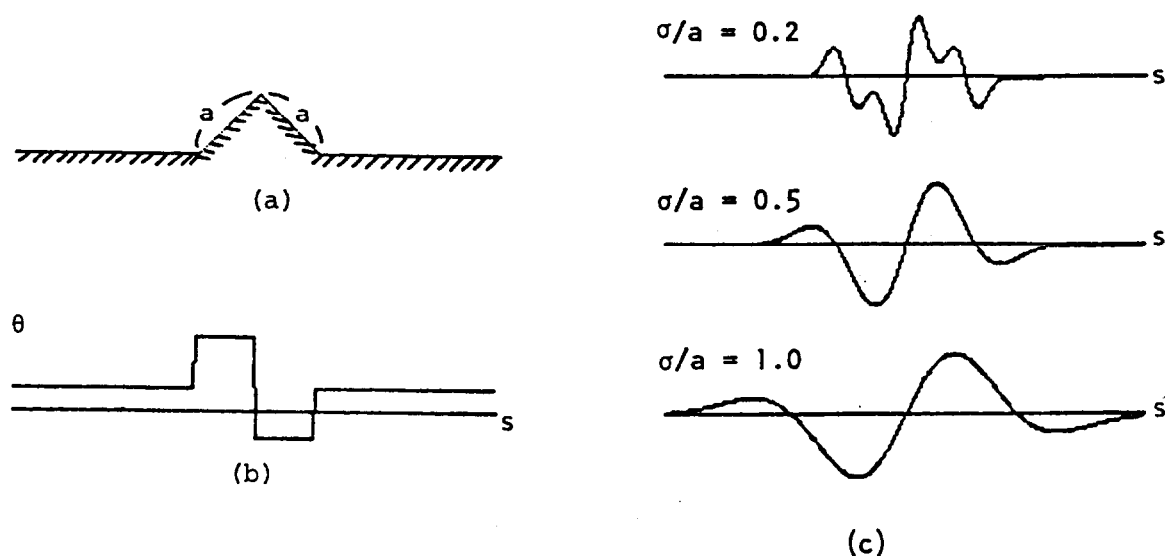


図 2.7: バンプ (Bump) のモデル : (a) バンプの形状、(b) 接線方向関数、(c) ガウシアン2次微分フィルタの出力

2.5 Scale Space 内でのプリミティブの探索

ここでは実際の図形を例にあげながら、図形輪郭のプリミティブを Scale Space の中で探索するアルゴリズムを説明する。プリミティブ探索の基本となるのは輪郭の各点における接線方向と周長との関係を示す接線方向関数である。2次元画像として図形が与えられたとき、その輪郭を求める方法には種々の方法があるが、ここで採用した方法については第4章で述べる。図 2.8(a) は工具の一種であるパンチをTVカメラで撮像し、エッジをつないでその輪郭を求めたものである。図 2.8(b) は点 S を始点として右回りに輪郭を追跡して得られた Chain Code 列 (8方向) をもとに計算した接線方向関数である。これは平滑化は全くなされておらず、45度の量子化誤差をそ

のまま残している。この関数を用いて輪郭プリミティブの探索を行う。

ステップ1：ガウシアン、1、2次微分フィルタとのコンボルーションを計算する

接線方向関数と $G'_\sigma(t)$ および $G''_\sigma(t)$ とのコンボルーションをいくつかの σ について計算する。スケール σ の値と範囲は図形の解像度や解釈したい図形の形状の大きさに依存し、本来的には自動的に決定すべきものであるが、ここでは 4, 5, 7, 11, 15, 22 の 6 種類を与えている。これは 4 を最小値として、順次 $\sqrt{2}$ 倍したものを整数化したものである。コンボルーションの結果 (フィルタの出力) を図 2.8(c) および (d) に示す。

ステップ2：スケール間でピークのマッチングをとる

フィルタ出力の正の値を持つ極大点と、負の値を持つ極小点を抽出し、隣り合うスケール間でマッチングを取り、暫定的な Tree を作成する。ここでのマッチングとは、スケール変化によってフィルタ出力のピークが移動するが、それがプリミティブのモデルに基づいて移動可能な範囲内にあるかどうかの判定を行うことをいう。隣り合うスケール間で、この条件を満足するピークのペアを結び付けることで Tree が形成される。この Tree は [Yuille & Poggio 1983] のいう “fingerprint” と同等のものである。しかし、ピークの解釈がなされるまでは、正しいマッチングは得られないので、この段階での Tree は不完全なままで残さざるを得ない。なぜなら、正しいマッチングを得るためにはプリミティブの種類が分かっており、そのモデルに基づいてピーク間の距離が範囲内にあるかどうかの判定をする必要があるが、その種類がまだ決定されていないためである。そのため、ステップ3以降の処理ではこの不完全性を考慮しておく必要がある。図 2.9 では正のピークを + で、負のピークを x で図示している。暫定的な Tree は図示されていない。

ステップ3：暫定 Tree のパーシング

次には暫定的な Tree の中で輪郭プリミティブに対応した応答を探す。ここでとった方法は簡単である。まず最大のスケールのピークの中で、複合的なプリミティブ (端点、クランク、バンプ) の応答を捜す。つまり、ピーク対、ソングレロ、4つ組の応答である。ある応答が見つかったら、Tree をスケールの小さな方向へ追跡して、ピークがプリミティブのモデルで解析されたような所定の動きをしているかをチェックする。正しい場合にはそのラベルをつけて Tree を更新する。これを用意されたスケールが終わるまで繰り返す。複合プリミティブの探索が終了した後に Corner の探

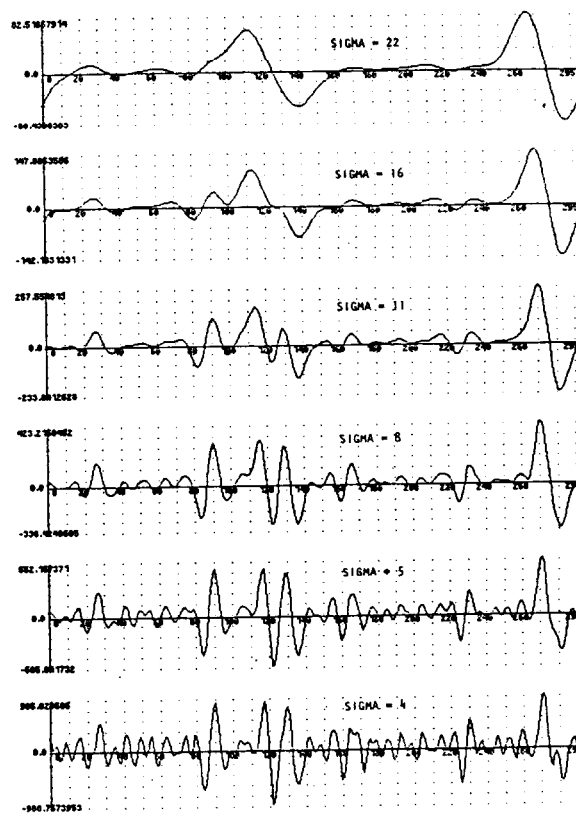
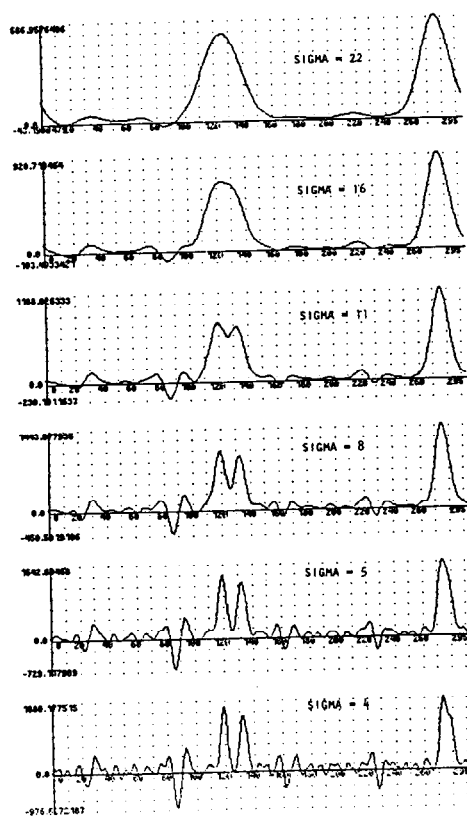
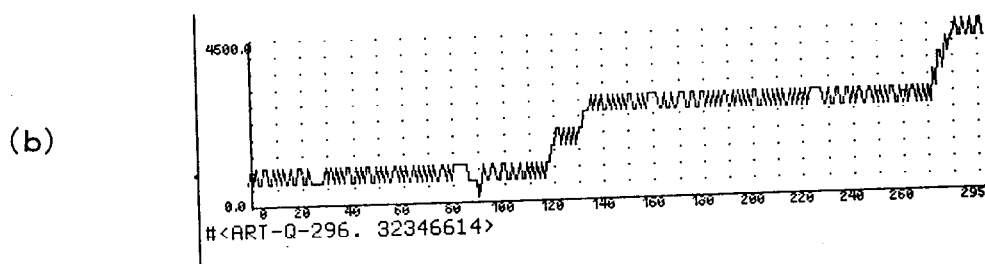
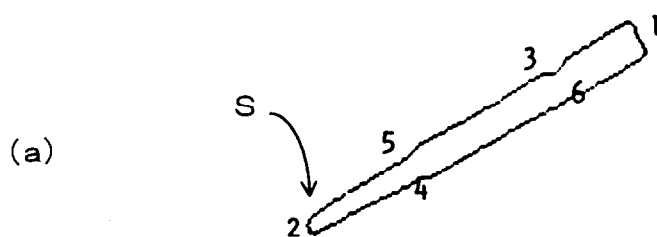


図 2.8: 図形パンチの (a) 輪郭、(b) 接線方向関数、(c) ガウシアン 1 次微分フィルタの出力、(d) ガウシアン 2 次微分フィルタの出力

索を同じ操作で行い、その後 Smooth Join を探索する。図 2.9 にパーズングの結果を示す。まず 2 つのデントとクランクが小さなスケールで検出され、そのあとでふたつの端点が $\sigma = 22$ で検出されている。

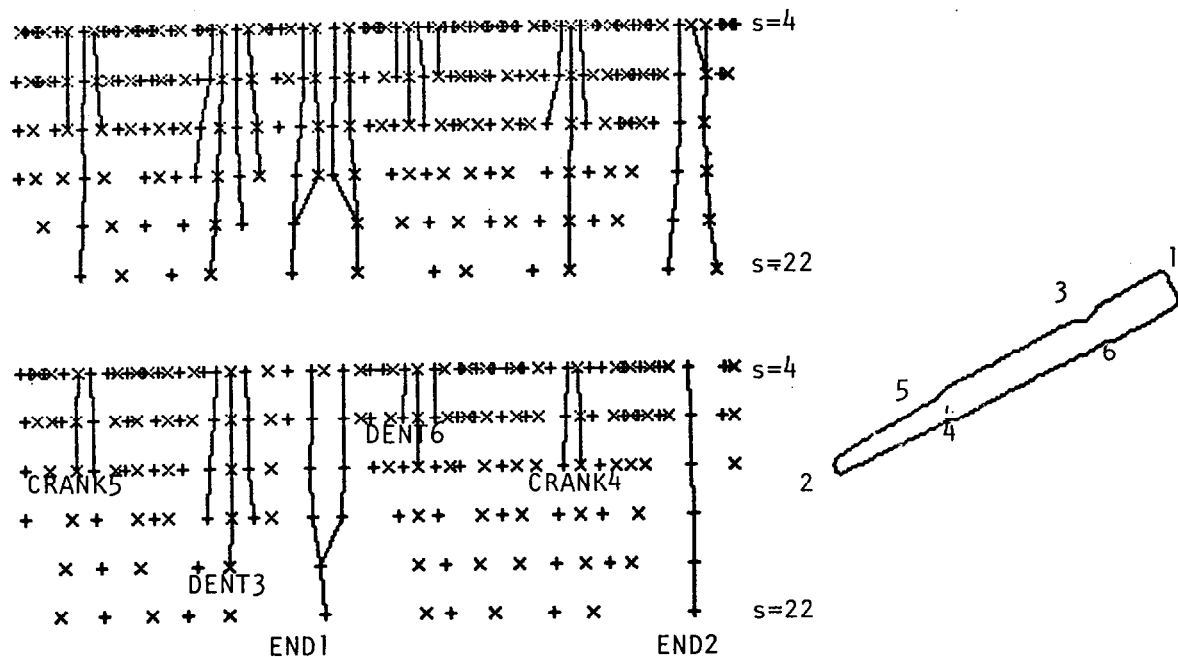


図 2.9: プリミティブのパーズングにより作成された Tree (パンチの例) (a) 2 次微分フィルタ出力、(b) 1 次微分フィルタ出力

研究の初期の段階では、暫定 Tree の不完全性をカバーするために Relaxation やダイナミックプログラミングなどの手法を用いることを考慮したが、実際にはそれらは必要でないことが分かった。なぜなら、大きなスケールで一意に定まらない場合でも、Corner の出力の収束性を考慮すると、ピークの動きをいくつかのスケールにわたって追いかけるだけで容易に分解することができるからである。

ステップ 4 : 分割点を求める

Corner の位置はピーク対の間の零交差点で与えられる。しかも、より小さいスケールの方が、より正しい位置を与えることができる。プリミティブの安定な検出と、その位置の測定には不確定性原理があり、位置の測定にはより小さいスケールが有用である。従って、検出された輪郭プ

リミティブの Tree のうちで、最も小さいスケールでのピークから輪郭の分割点を計算する。

一方、Smooth Join の位置は 2 次微分フィルタのピークで与えられ、より大きなスケールの方が安定である。

図 2.10 の左の列はこの様にして求めた分割点をそれが初めて見つけれられたスケール毎に表示したものである。同図の右の列はそれらの分割点間の輪郭を次章で述べる直線・円弧近似して得られた近似図形であり、ここでは、目につきやすい図形の特徴は大きなスケールで検出されており、逆に小さなスケールで見つけれられたプリミティブは細部構造を表現しているといえる。

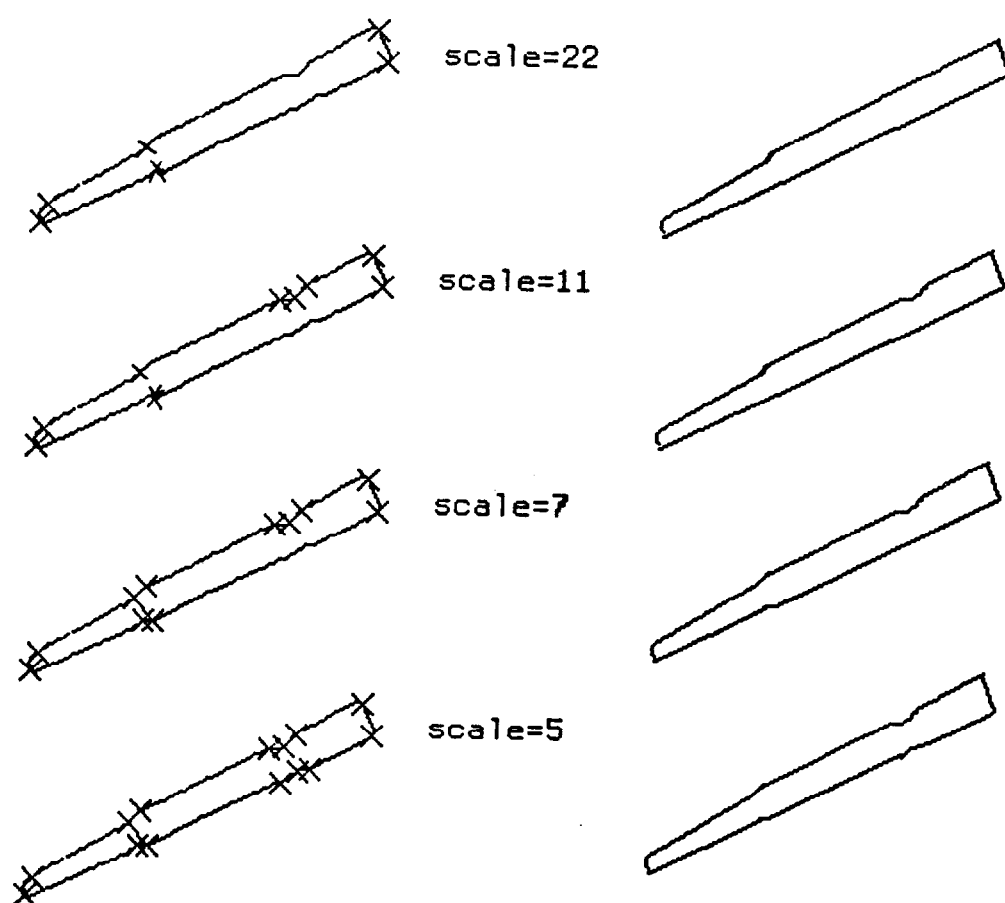


図 2.10: スケール毎に示した分割点 (左列) と対応する近似図形 (右列)

2.6 処理例

本節ではいくつかの工具を対象とした処理結果を示す。図 2.11 では大きなスケールでナイフの刃の先端の Corner と柄の端点、および、柄と刃との接合部を示すクランクが検出されており、小さいスケールで刃先の微妙な曲線の境界を示す Smooth Join が検出されている。図 2.12 でも同様に大きなスケールでドライバの大まかな形状を示す分割点が検出されており、スケールが小さくなるにつれて微細構造が検出されている。図 2.13 はハンマーについての結果を示したものである。

また、図 2.14～図 2.16 は 3 種類のハンマーについてスケール毎に輪郭の分割点と近似図形を示したものである。ハンマーは基本的には「柄」と「頭」からなり、その構造を示す分割点が同じスケールで検出されており、さらに個々の特有の形状を細かく規定する分割点が、より小さなスケールで検出されているのが分かる。このように、Scale Space の中で検出されたモデル応答をスケールでソートし、特徴の重要さとスケールとを対応づけして、階層的な輪郭の記述を実現することができる。

図 2.17 は同じ物体（図 2.12 のドライバ）を 3 種類の異なる方向に置いて観測して得られる 3 つの図形に対してアルゴリズムを適用した結果を示している。微小な違いはあるが、ふたつの端点（end）は安定に検出されており、また、ドライバの柄とシャフトの接合部であるふたつのクランクは同じスケールで検出されているなど、その輪郭プリミティブの解釈は安定であり、このアルゴリズムのロバスト性を示している。

最小のスケールまでに検出された輪郭プリミティブによって、輪郭をセグメント化し、その形状を円弧あるいは直線を当てはめて得られる図形は元の図形をよく近似している。この様子を図 2.18 に示す。

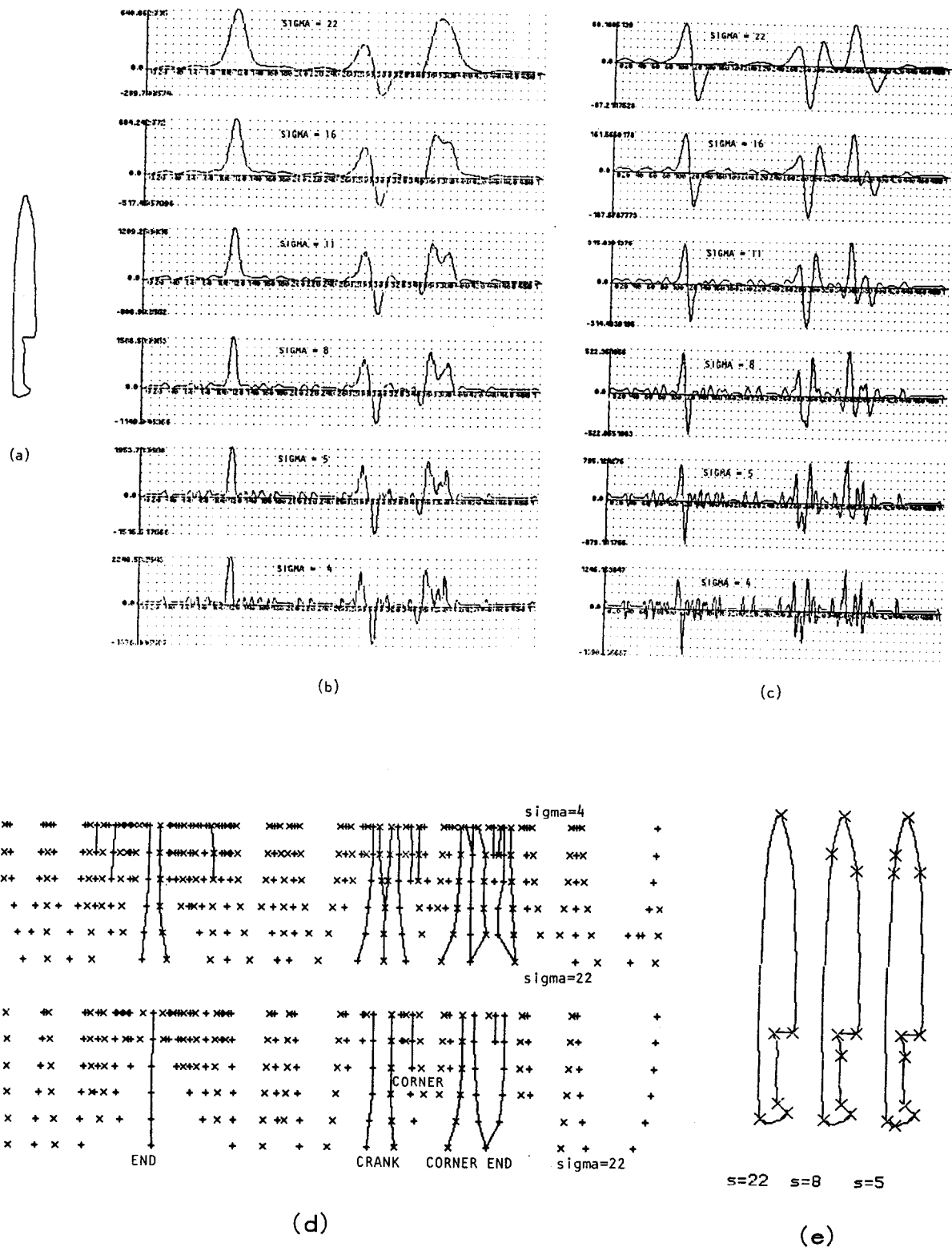


図 2.11: 図形「ナイフ」に対する処理結果: (a) 輪郭形状、(b) ガウシアン1次微分フィルタ出力、(c) ガウシアン2次微分フィルタ出力、(d) 正の極大点 (+) と負の極小点 (x) の上に作られた輪郭プリミティブの検出結果を Tree で示したもの (下は1次微分フィルタ出力、上は2次微分フィルタ出力に対する)、(e) スケール毎に示した輪郭のプリミティブ

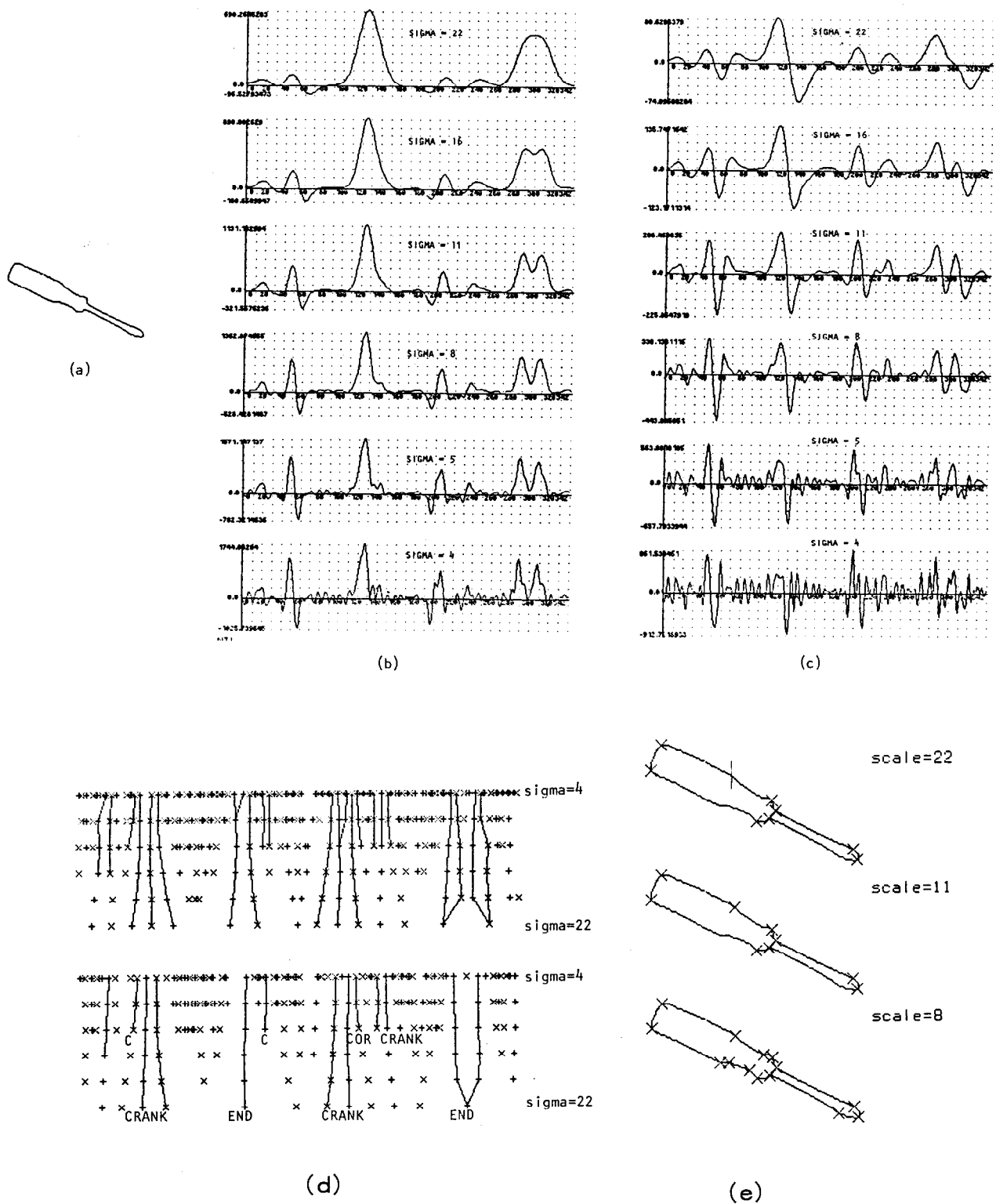


図 2.12: 図形「ドライバ」に対する処理結果: (a) 輪郭形状、(b) ガウシアン1次微分フィルタ出力、(c) ガウシアン2次微分フィルタ出力、(d) 正の極大点(+)と負の極小点(x)の上に作られた輪郭プリミティブの検出結果をTreeで示したもの(下は1次微分フィルタ出力に対する)、(e) スケール毎に示した輪郭のプリミティブ

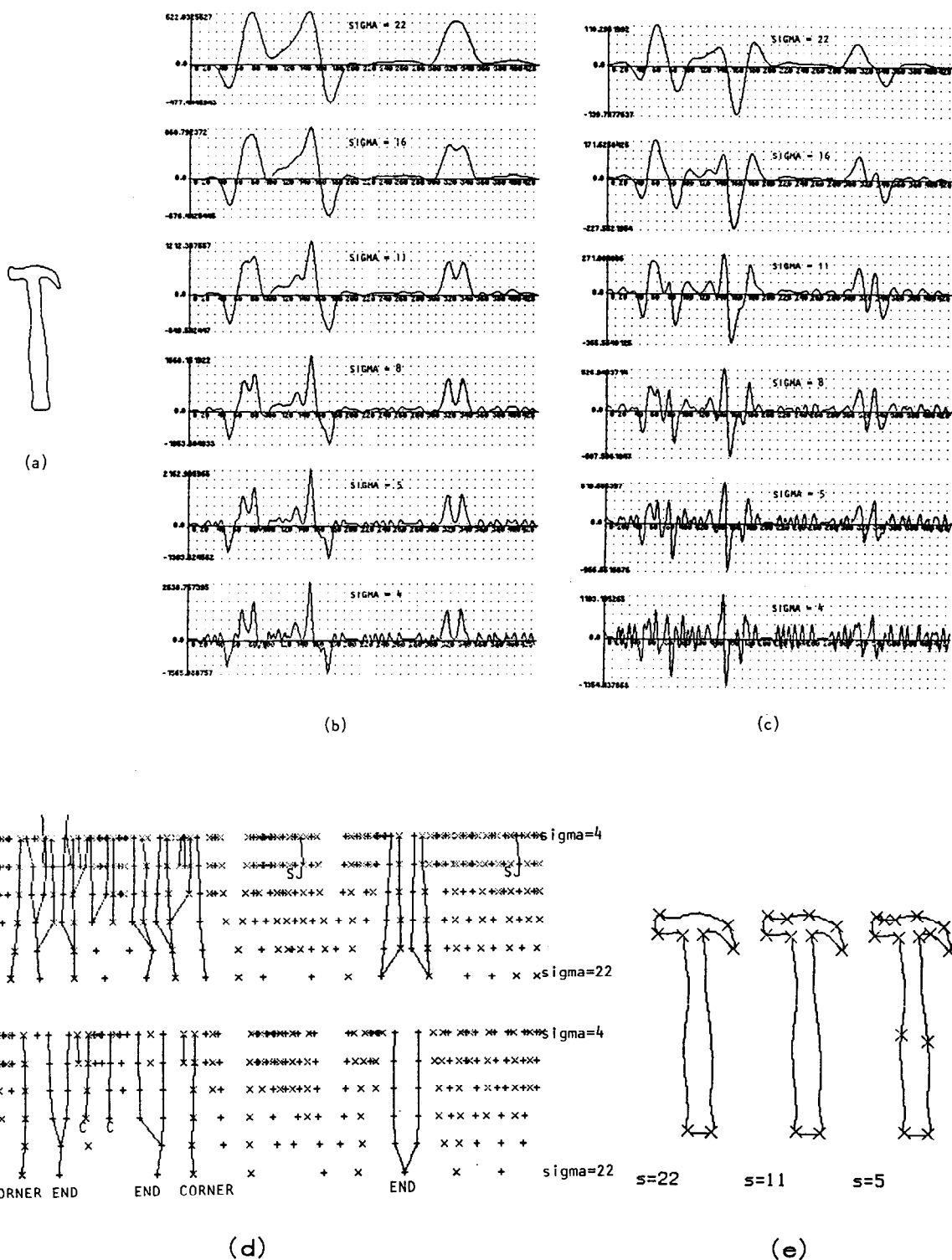


図 2.13: 図形「ハンマー」に対する処理結果: (a) 輪郭形状、(b) ガウシアン1次微分フィルタ出力、(c) ガウシアン2次微分フィルタ出力、(d) 正の極大点(+)と負の極小点(x)の上に作られた輪郭プリミティブの検出結果を Tree で示したもの(下は1次微分フィルタ出力、上は2次微分フィルタ出力に対する)、(e) スケール毎に示した輪郭のプリミティブ

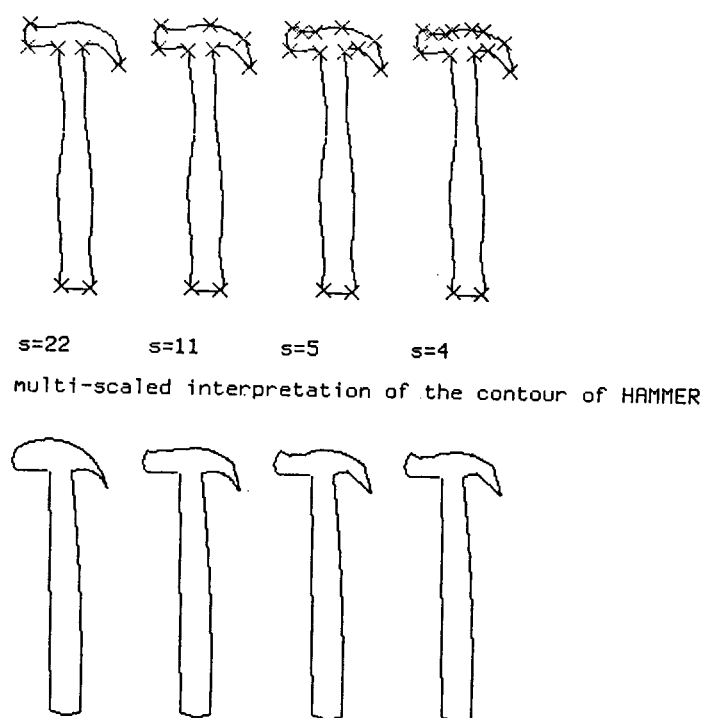


図 2.14: 図形「ハンマー」のスケール毎に示した輪郭の分割点（上）と、近似図形（下）

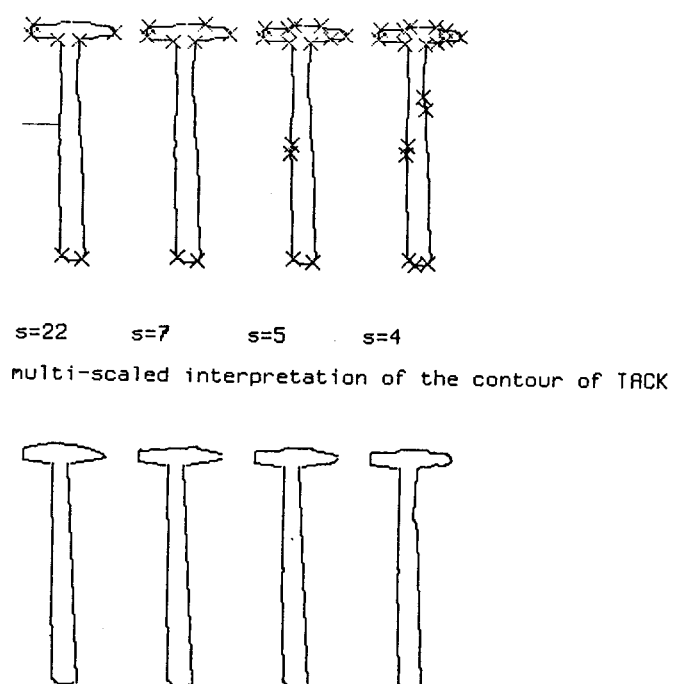
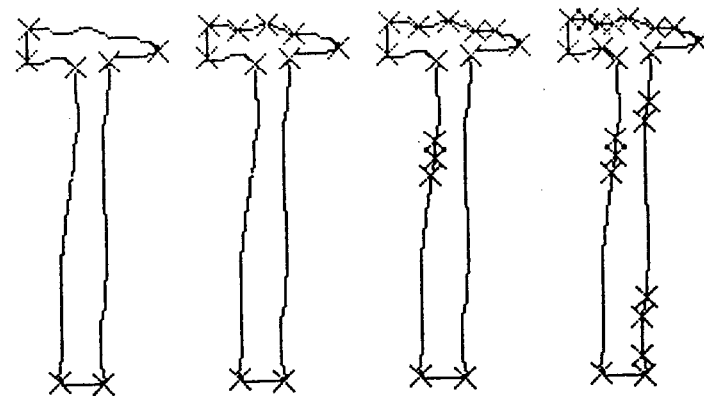


図 2.15: 図形「タックハンマー」のスケール毎に示した輪郭の分割点（上）と、近似図形（下）



$s=22$

$s=11$

$s=5$

$s=4$

multi-scaled interpretation of the contour of WARRINGTON

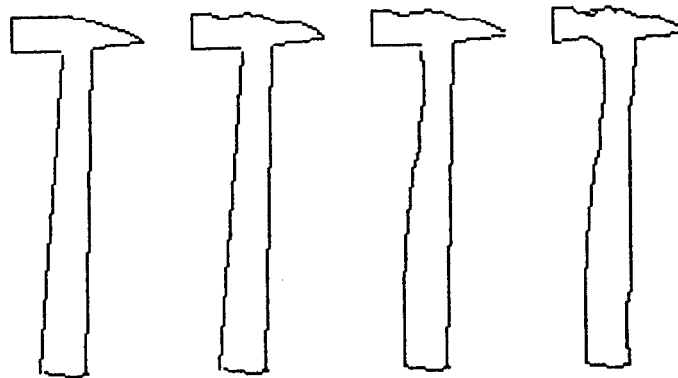


図 2.16: 図形「ワーリントンハンマー」のスケール毎に示した輪郭の分割点（上）と、近似図形（下）

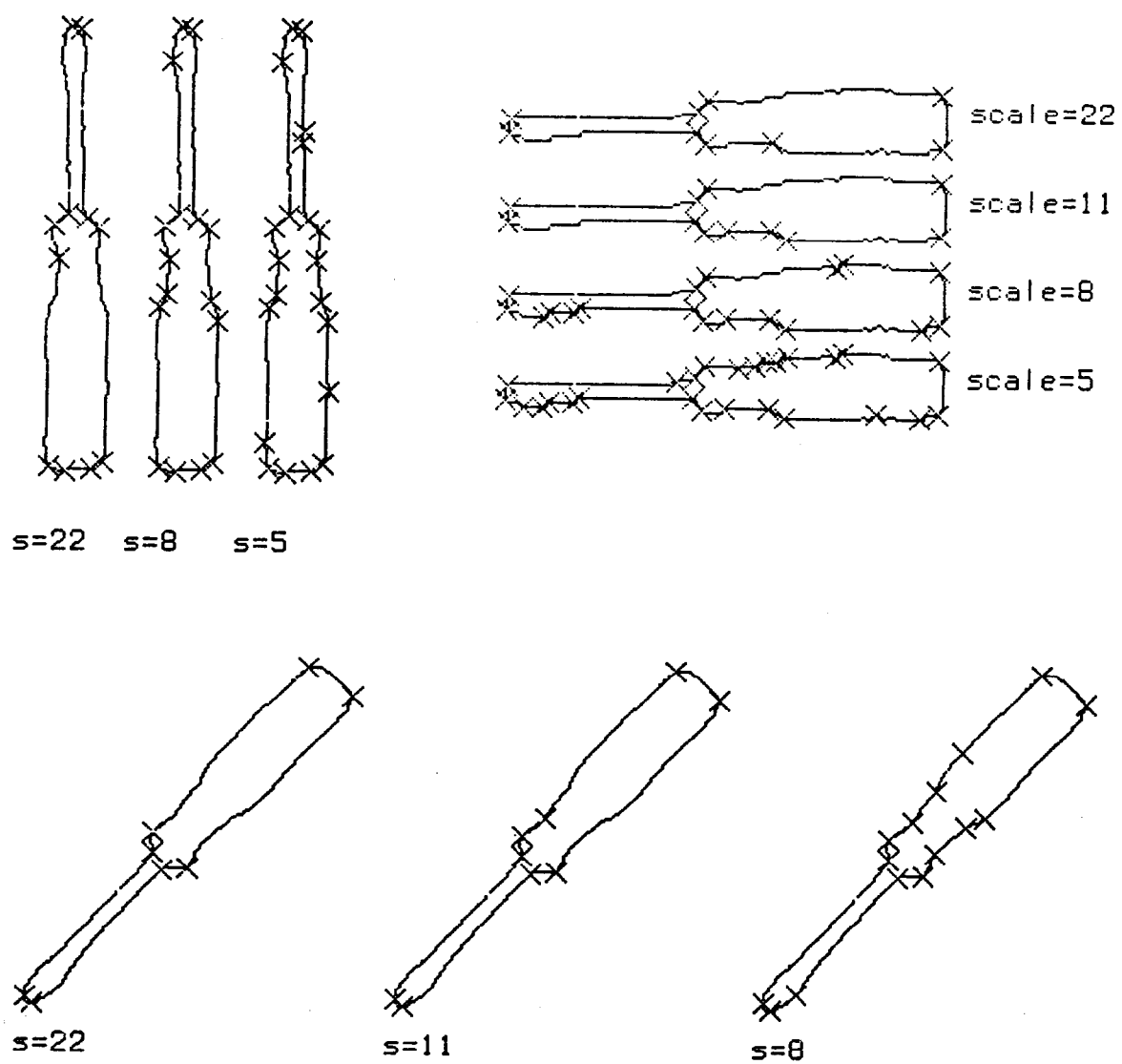


図 2.17: 異なる向きに置いたドライバ (図 2.12(a)) に対する分割点

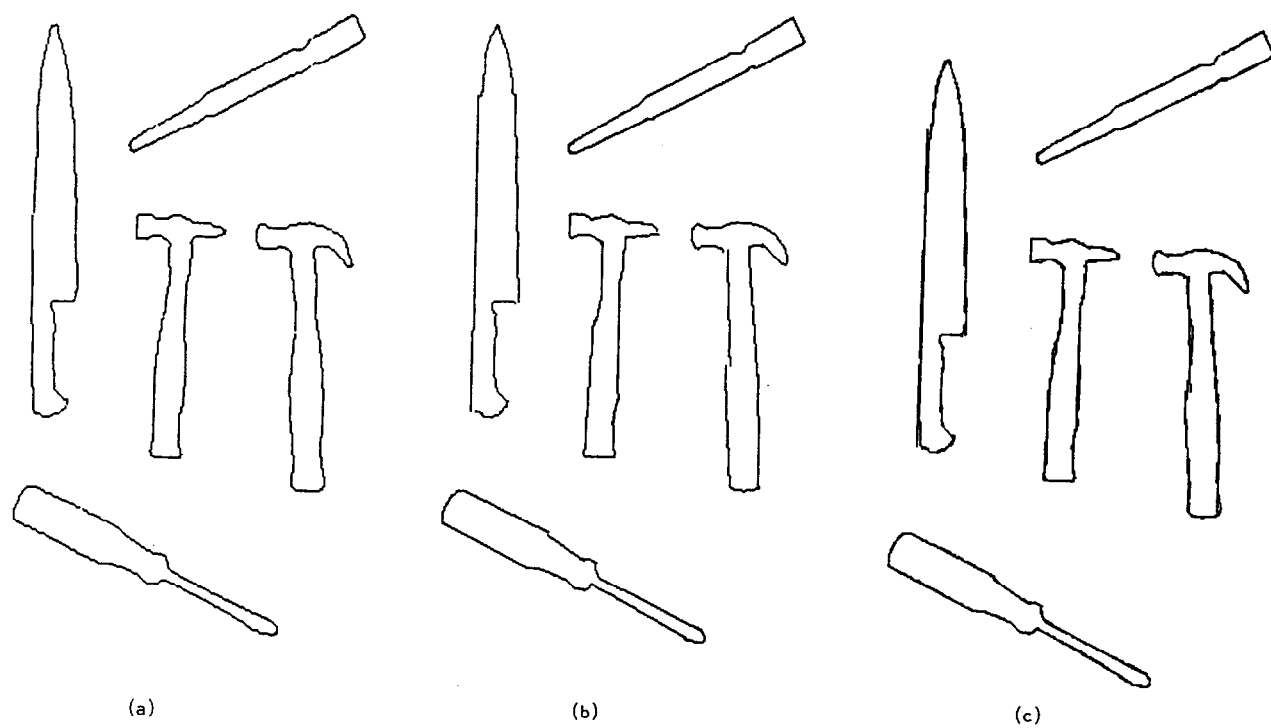


図 2.18: 原図形 (a) と近似図形 (b)。 (c) はそれらを重ねて表示したもの

2.7 3次元曲面の解釈への応用

前節までに述べた、二次元曲線の記号的解釈の考え方を発展させ、球面や円筒面、円錐面などの基本的な曲面を区分的に繋ぎ合わせて3次元曲面を解釈するというアプローチは自然である。これはSurface Primal Sketchと名づけることができる。Curvature Primal Sketchは平面曲線を対象としたものであるが、3次元曲面の主曲率線の集合の各々に対して、Curvature Primal Sketchを適用して、主要な曲率の変化点を求め、それらの点を主曲率線をわたって繋ぎ合わせることにより、主要な曲面の不連続性（線）を抽出することができる。図2.19に電話の受話器の曲面に適用した結果を示す。濃度が濃くなっている部分が検出された不連続線を示している。 σ の値を20から80まで変えた4通りの結果を示しており、 σ の大きな値では物体の大まかの構造が、小さい値ではノイズも含まれているが、細部構造が示されている。また、別の物体として、マグカップとボトルの例を図2.20に示す。マグカップでは取っ手の接合部分が抽出されており、ボトルでは底から上に上がっていくにつれて、円筒面、円錐面などが繋ぎ合わされているのを見ることができる。

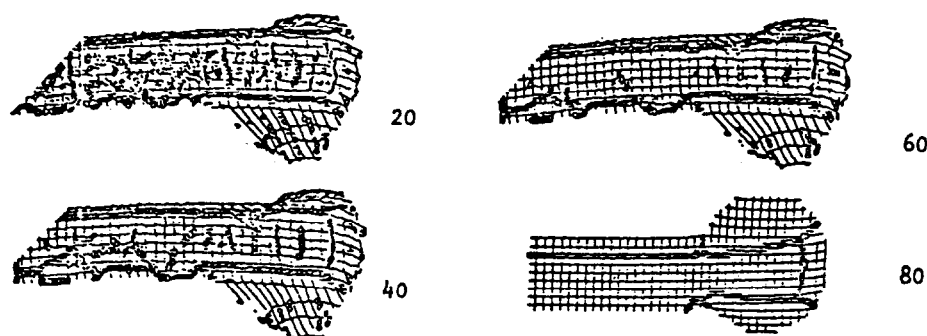


図 2.19: 3次元曲面の主要な不連続線の検出（電話の受話器）。濃度の濃い部分が不連続線を示している。数字は σ の値。

2.8 まとめ

2次元図形の輪郭ベースの階層的記述方式を提案した。曲率の変化の形態（つまり不連続性）を分類し、それらを輪郭のプリミティブとして定義し、これらのプリミティブを Scale Space の中で探索するという手法であり、画像理解における明るさの変化の記号的解釈を意味する Primal Sketch との類似性から、提案する方式を Curvature Primal Sketch と名づけた。輪郭の接線方向関数に対するガウシアン の 1、2 次微分フィルタの出力の挙動を各プリミティブのモデルについて明らかにし、Scale Space 内での探索アルゴリズムを構築した。この方式をいくつかの平面図形に適応して、人間の感覚にマッチした階層的な記述が得られることを示した。また、このアルゴリズムを 3 次元曲面の主曲率曲線に適用して、その上での曲率の主要な変化点を検出し、それらを繋ぎ合わせることで、曲面上の曲率の不連続線の検出が可能であることを実験的に示した。

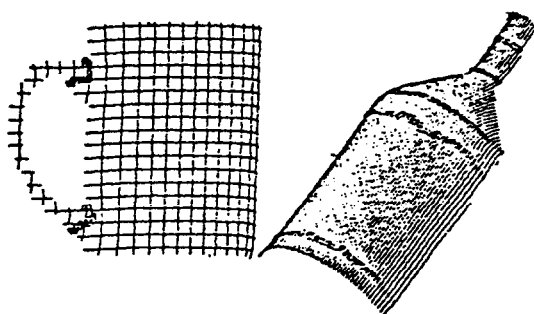


図 2.20: 曲面の接合部の抽出。左はマグカップの例であり取っ手の部分が抽出されている。右はボトルであり円形の接合部分が示されている。

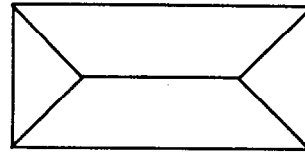
第3章 Smoothed Local Symmetriesによる図形のネットワーク記述

本章では図形の領域に基づいた記述方法を提案する。この方法はSmoothed Local Symmetries(SLS)と呼ばれ、局所的に対称な関係にある二つの輪郭点の中点を滑らかに結んだ曲線 (SLS) を軸として、この軸に沿って図形の領域を記述しようとするものである。これは元々、[Brady 82] で提案されたものであるが、本論文では第2章の輪郭解析の結果を元に効率的に計算するアルゴリズムを示し、さらに、この領域ベース方法と第2章の輪郭ベースの方法とを融合させた、図形のネットワーク記述の方法を提案する。輪郭情報と領域情報とは相補的な関係にあるため、認識過程の要請に応じて、必要な情報を参照して認識処理を進めることができるようになる。また、SLSの計算の元となる輪郭ベースの記述は階層的であるため、このネットワーク記述も階層的なものを得ることができる。

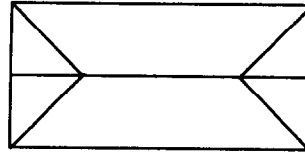
3.1 領域ベースの図形記述方式の従来研究

図形の領域を直接的に記述しようという試みの代表的なものに、Symmetric Axis Transform (SAT) [Blum 78] と Ribbon と呼ばれる2次元 Generalized Cylinder[Brooks 79] がある。これらはいずれも、ある軸に沿って、定まった規則でディスクあるいは線分をスイープすると元の図形が再現できるという記述方法である。SATは距離変換画像の尾根と同等であり、スイープされたディスクの和集合が元の図形となる。しかし、輪郭点とディスクとの1対1対応を無理につけようとするため、直感的に不自然な表現になっている。特に長方形のような簡単な図形でも5個の部分図形に分割されたり、輪郭にへこみをつけるなどの少しの変形で表現が大きく変わってしまうという欠点がある(図3.1)。一方、ribbonは本質的にほとんどすべての図形を簡潔に表現できる能力があるが、ACRONYM [Brooks 1981] では直線軸しか許されていず、自由な曲線軸やスイープの規則を実現することは困難である。

Bradyは2次元 Generalized Cylinder のアイデアと SAT の方法論とをミックスして、Smoothed



SAT



SLS

図 3.1: 長方形の SAT(Symmetric Axis Transform) と SLS

Local Symmetries (S L S) と呼ばれる記述方式を提案した [Brady 82]。これは、局所的に対称な関係にある二つの輪郭点の中点を滑らかに結んだもの (SLS) を軸として、この軸に沿って図形の領域を記述しようとするものである。図 3.2 は目標となる理想的な軸を示したものであり、この軸は人間の知覚にマッチした、図形を簡潔に表現できる座標系であるといえる。

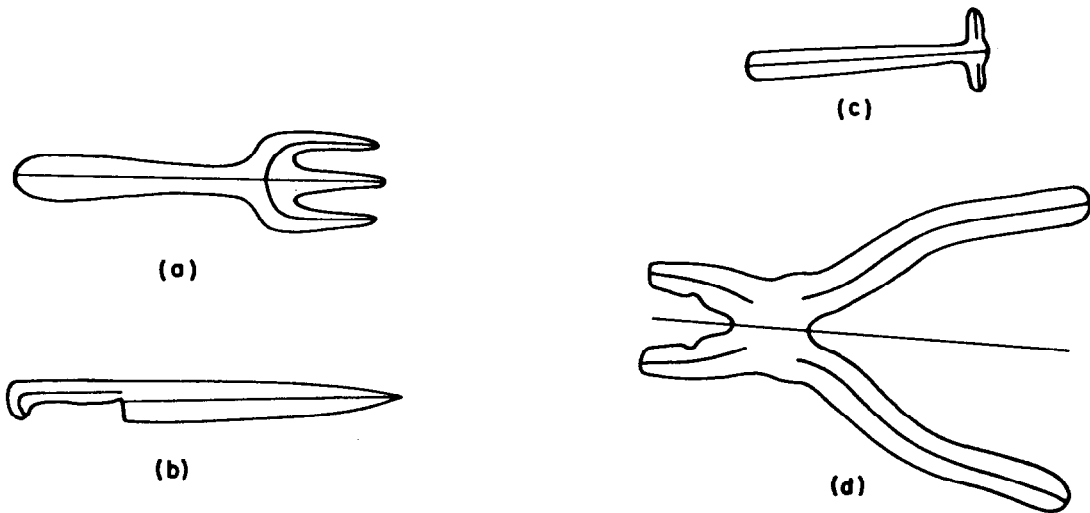


図 3.2: 目的とする S L S の形状をいくつかの理想的な図形で示したもの

3.2 Local Symmetry (局所対称) の定義

図 3.3(a) において、輪郭上の点 A が点 B と局所的に対称であるための条件は、A における法線 N_A と線分 $\overline{BA}$ のなす角が、B における法線 N_B と線分 $\overline{BA}$ のなす角に等しいことである。P を A, B の中点とすると、A を動かしたときの P の軌跡を Smoothed Local Symmetries と呼ぶ。図 3.3

(b) に示すようにひとつの点Aに対して、ふたつの点 B_1 、 B_2 が局所対称となる場合もあるが、S A Tのようにこれを妨げることはしない。

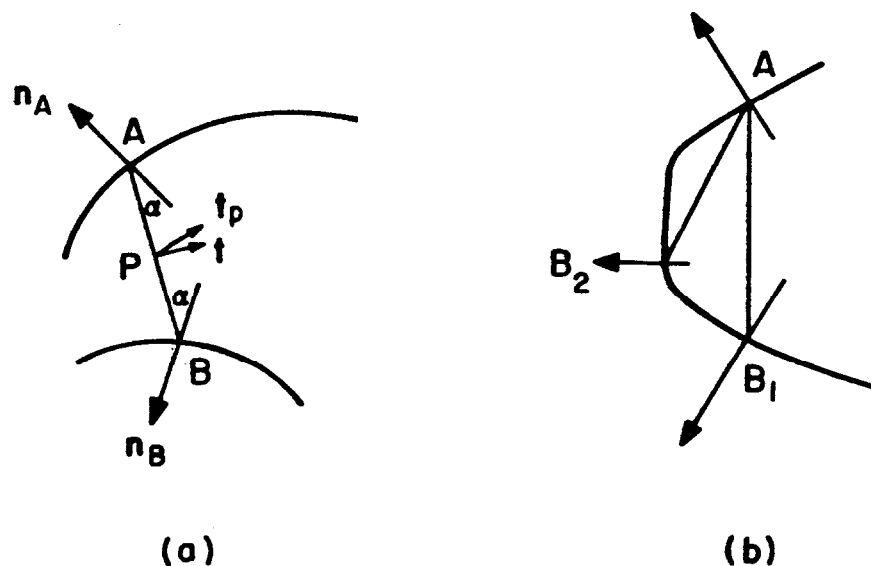


図 3.3: 局所対称の定義

3.3 パラメータ化と記号化

図形の領域は S L S に沿った各種の量、たとえば、 $h = \overline{PA}$ の長さ、 α 、A および B での曲率 (K_a 、 K_b) などをパラメータとして記述することができる。図 3.4 は K_a と K_b の符号だけで定義できる基本図形を示している。また、測定されたパラメータの値を量子化することにより、領域の属性を記号的に表現することもできる。

3.4 S L S 計算アルゴリズム

図 3.3 に示す S L S の定義に従って、輪郭上の点 A の局所対称点を全輪郭の中で求め、この操作をすべての輪郭点について行うという $O(n^2)$ のアルゴリズムは容易にプログラム化できる。しかし、点 P の平滑化軌跡を求めることは比較的困難であるし、輪郭のわずかな変形の影響も受けやすい。したがって、ここでは前章で述べた Curvature Primal Sketch によって輪郭をセグメント化し、各セグメントの円弧・直線近似を行った後、各セグメント間で S L S を解析的に計算するという S L S 計算アルゴリズムを提案する。セグメント化により大幅にデータが圧縮されるため、

計算量が減るだけでなく、平滑化があらかじめなされているので、ノイズの影響を軽減することができる。

		K_A			
		$h' > 0$	> 0	$= 0$	< 0
K_B	> 0	 cup	 sector	 beak	
	$= 0$	 sector	 wedge	 plinth	
	< 0	 beak	 plinth	 flare	

図 3.4: 曲率の符号だけで決まる基本図形

3.4.1 輪郭セグメントの円弧・直線近似

輪郭の分割点が与えられたとき、ふたつの分割点間の輪郭（周長＝ n ）を直線または円弧で近似するアルゴリズムを説明する。ひとつの部分輪郭を輪郭点の集合 $\{(x_i, y_i), 1 \leq i \leq n\}$ とし、まずはこの部分輪郭を円弧近似できるかどうかを試してみて、その解に安定性がない場合には直線近似するという方法を取る。

$$(x - x_0)^2 + (y - y_0)^2 - r^2 = 0$$

の円を仮定し、二乗誤差、

$$J(x_0, y_0, r) = \sum_{i=1}^n \left\{ (x_i - x_0)^2 + (y_i - y_0)^2 - r^2 \right\}^2$$

を最小にする (x_0, y_0) は連立一次方程式、

$$x_0 \sum (x_i - \bar{x})^2 + y_0 \sum (x_i - \bar{x})(y_i - \bar{y}) = \frac{1}{2} \sum (x_i^2 + y_i^2)(x_i - \bar{x})$$

$$x_0 \sum (x_i - \bar{x})(y_i - \bar{y}) + y_0 \sum (y_i - \bar{y})^2 = \frac{1}{2} \sum (x_i^2 + y_i^2)(y_i - \bar{y})$$

の解で与えられ、 r は

$$r^2 = \frac{1}{n} \left\{ \sum (x_i - \bar{x})^2 + \sum (y_i - \bar{y})^2 \right\}$$

で与えられる。ここに $\bar{x}$ 、 $\bar{y}$ は各々 x_i 、 y_i の平均である。

この解は r が大きくなると poorly conditioned となるが、この条件は係数行列

$$A = (a_{ij}) = \begin{pmatrix} \sum (x_i - \bar{x})^2 & \sum (x_i - \bar{x})(y_i - \bar{y}) \\ \sum (x_i - \bar{x})(y_i - \bar{y}) & \sum (y_i - \bar{y})^2 \end{pmatrix}$$

の固有値 λ_1 、 λ_2 で定義される指標

$$L = \frac{|\lambda_1 - \lambda_2|}{|\lambda_1 + \lambda_2|} = \frac{\sqrt{(a_{11} - a_{22})^2 + 4a_{12}^2}}{Tr(A)}$$

を用いて判定する。もし L がある閾値より小さい場合には円弧近似を適用し、そうでない場合は直線近似とする。現状では閾値は 0.96 としているが、これは、最大固有値と最小固有値との比 ($= \max(\lambda_1, \lambda_2) / \min(\lambda_1, \lambda_2)$) を指標とした場合の 49 に対応している。直線近似は輪郭の各点からの垂線の長さの二乗和を最小にする直線を求めることで実行できる。

3.4.2 2つの輪郭セグメントから生成される S L S の解析的計算方法

輪郭セグメントは直線または円弧で近似されているので、二つのセグメントの組み合わせには 3通りあるが、図 3.3 の N_A と N_B の交点 Q (図示されていない) の軌跡は、組み合わせの種類によって、以下のように直線または二次曲線となる。

直線セグメント 対 直線セグメント : 直線になる
 直線セグメント 対 円弧セグメント : 放物線となる
 円弧セグメント 対 円弧セグメント : 楕円または双曲線となる

各々の輪郭セグメントの存在範囲が限られているため、それらの輪郭セグメントから作られる交点Qの軌跡が実際に存在するかどうかは分からない。その判定も必要である。以下において、各々の場合の軌跡と存在範囲を解析的に求める。

直線セグメント 対 直線セグメント

直線セグメントの法線方向（x軸となす角度）を θ とすると、直線セグメントは、

$$x \cos \theta + y \sin \theta + c = 0$$

と書ける。ここで、 θ に $\theta + \pi$ を代入し、 c に $-c$ を代入すると、上の式は同一の直線を表すが、図形の存在領域が逆側になっていることに注意する必要がある。図 3.5 はその様子を示したものである。

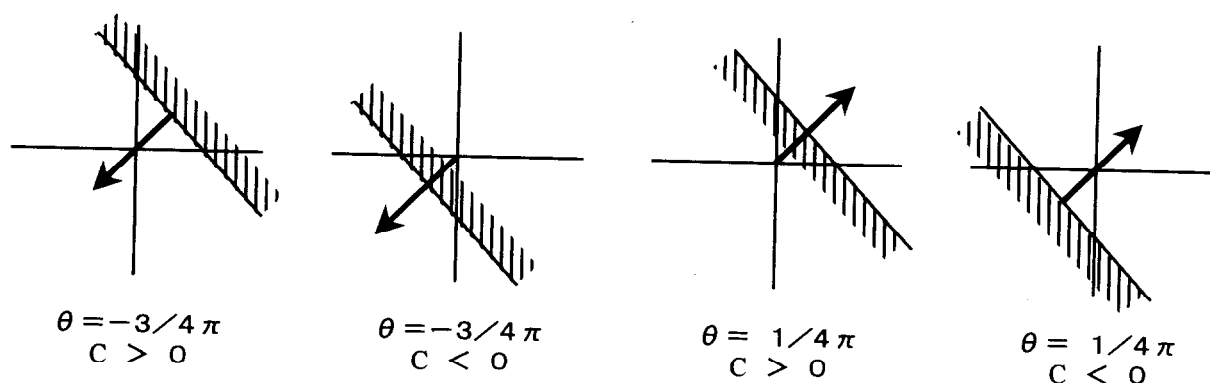


図 3.5: 直線セグメントの定式化

一方の直線セグメントを、

$$x \cos \theta_1 + y \sin \theta_1 + c_1 = 0$$

$$x \in [x_{s1}, x_{e1}], y \in [y_{s1}, y_{e1}]$$

とし、他方を

$$x \cos \theta_2 + y \sin \theta_2 + c_2 = 0$$

$$x \in [x_{s2}, x_{e2}], y \in [y_{s2}, y_{e2}]$$

とすると、二つの直線セグメントの法線の交点は、もし存在するとすると、次の直線上に存在する。

$$x(\cos \theta_1 - \cos \theta_2) + y(\sin \theta_1 - \sin \theta_2) + c_1 - c_2 = 0$$

これは、

$$x \cos \alpha + y \sin \alpha + c' = 0$$

$$\text{where } \alpha = \frac{\theta_1 + \theta_2}{2} + \frac{\pi}{2}, c' = \frac{c_1 - c_2}{2 \sin \frac{\theta_1 - \theta_2}{2}}$$

と書ける。

S L Sが存在するかどうかは、第1のセグメントの存在区間 $[x_{s1}, x_{e1}]$ と第2のセグメントの存在区間 $[x_{s2}, x_{e2}]$ を上記直線上に投影して、共通部分を持つかどうかで判定する。投影の変換式は、

$$\xi = x - (x \cos \alpha + y \sin \alpha + c') \cos \alpha$$

で与えられ、区間 $[\xi_{s1}, \xi_{e1}]$ と区間 $[\xi_{s2}, \xi_{e2}]$ とが共通部分を持てば、その共通部分がS L Sの存在領域となる。なお、蛇足であるが y に関する投影式は、

$$\eta = y - (x \cos \alpha + y \sin \alpha + c') \sin \alpha$$

となる。

直線セグメント 対 円弧セグメント

円弧セグメントは適当に座標変換して、原点を中心とする半径 r の円で表現する。この場合も直線セグメントの場合と同様に図形の存在範囲に対する注意が必要であり、図形が中心側に存在

するとき、つまり、凸の場合には $r > 0$ とし、図形が中心とは反対側に存在するとき、つまり、凹の場合には $r < 0$ で表現する。また、直線セグメントは座標軸を原点の回りに適当に回転して y 軸に平行な直線であって、図形領域が x 軸の負の方向に存在するように表現する。このような座標系を考えても一般性を失わない。この座標系における円弧セグメントは、

$$\begin{aligned}x &= r \cos \theta \\y &= r \sin \theta \\&\text{where } \theta \in [\theta_s, \theta_e]\end{aligned}$$

と書ける。一方、直線セグメントは、

$$\begin{aligned}x + c &= 0 \\&\text{where } y \in [y_s, y_e]\end{aligned}$$

となる。 r の符号、および、円と直線の位置関係によって場合分けする

1. $r > 0$ の場合

図 3.6 に示すような 3 種類の位置関係が有りうるが、

(a) $-c > r$ の場合

軌跡は、

$$y^2 = 2(r+c)x + (r+c)^2$$

の放物線となり、存在するかどうかは、

$$\begin{aligned}y_{s1} &= \frac{\sin \theta_s}{1 - \cos \theta_s} (c - r) \\y_{e1} &= \frac{\sin \theta_e}{1 - \cos \theta_e} (c - r)\end{aligned}$$

としたとき、区間 $[y_s, y_e]$ と区間 $[y_{s1}, y_{e1}]$ とが共通部分を持つかどうかで判定する。共通部分あれば、それが S L S の存在範囲である。

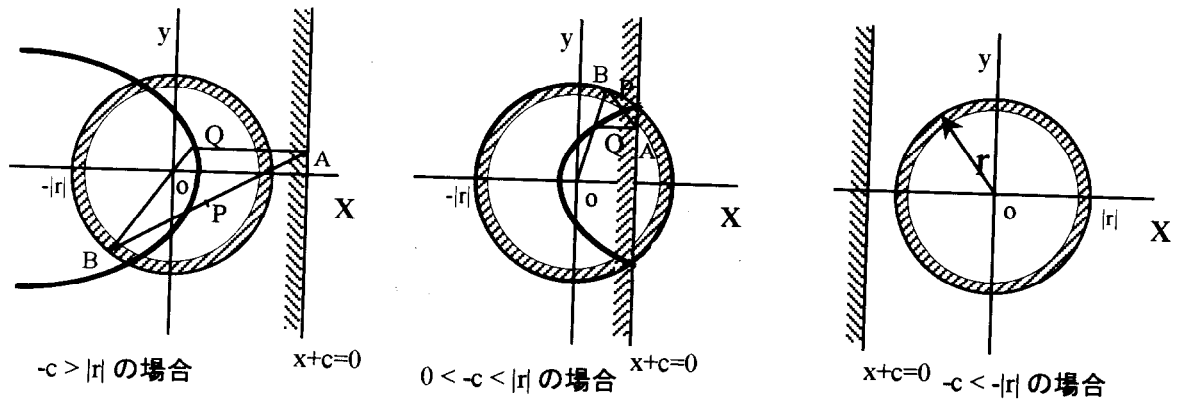


図 3.6: 直線セグメント 対 円弧セグメントの位置関係 ($r > 0$)

(b) $-r < -c < r$ の場合

軌跡は、

$$\begin{aligned}
 x &= (r+c) \frac{\cos \theta}{1 - \cos \theta} \\
 y &= (r+c) \frac{\sin \theta}{1 - \cos \theta} \\
 \text{where } \arctan \frac{\sqrt{r^2 - c^2}}{-c} &< |\theta| < \pi
 \end{aligned}$$

の放物線となり、存在するかどうかの判定は、

$$\begin{aligned}
 y_{s1} &= \frac{\sin \theta_s}{1 - \cos \theta_s} (c+r) \\
 y_{e1} &= \frac{\sin \theta_e}{1 - \cos \theta_e} (c+r)
 \end{aligned}$$

としたとき、区間 $[y_s, y_e]$ と区間 $[y_{s1}, y_{e1}]$ とが共通部分を持つかどうかで行う。共通部分が SLS の存在範囲である。

(c) $-c < -r$ の場合

図 3.6 を見てわかるように、SLS は存在し得ない。

2. $r < 0$ の場合

図 3.7 に示すような 3 種類の位置関係が有りうるが、

(a) $-c > |r|$ の場合

軌跡は、

$$\begin{aligned} x &= (|r| - c) \frac{\cos \theta}{1 + \cos \theta} \\ y &= (|r| - c) \frac{\sin \theta}{1 + \cos \theta} \\ \text{where } 0 \leq |\theta| < \pi \end{aligned}$$

の放物線となり、存在するかどうかの判定は、

$$\begin{aligned} y_{s1} &= \frac{\sin \theta_s}{1 + \cos \theta_s} (|r| - c) \\ y_{e1} &= \frac{\sin \theta_e}{1 + \cos \theta_e} (|r| - c) \end{aligned}$$

としたとき、区間 $[y_s, y_e]$ と区間 $[y_{s1}, y_{e1}]$ とが共通部分を持つかどうかで行う。共通部分が S L S の存在範囲である。

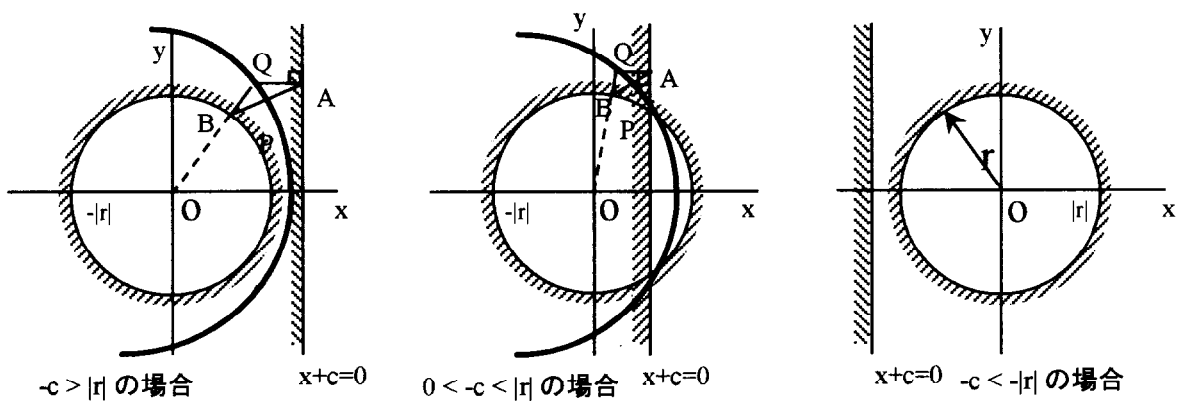


図 3.7: 直線セグメント 対 円弧セグメントの位置関係 ($r < 0$)

(b) $-|r| < -c < |r|$ の場合

軌跡は、

$$\begin{aligned}
x &= (|r| - c) \frac{\cos \theta}{1 + \cos \theta} \\
y &= (|r| - c) \frac{\sin \theta}{1 + \cos \theta} \\
\arctan \frac{\sqrt{r^2 - c^2}}{-c} &< |\theta| < \pi
\end{aligned}$$

の放物線の一部となり、存在するかどうかの判定は、上の式での θ の範囲内に $[\theta_s, \theta_e]$ を制限した上で、

$$\begin{aligned}
y_{s1} &= \frac{\sin \theta_s}{1 + \cos \theta_s} (|r| - c) \\
y_{e1} &= \frac{\sin \theta_e}{1 + \cos \theta_e} (|r| - c)
\end{aligned}$$

を計算し、区間 $[y_s, y_e]$ と区間 $[y_{s1}, y_{e1}]$ とが共通部分を持つかどうかで行う。共通部分がS L Sの存在範囲である。

(c) $-c < -|r|$ の場合

図 3.7 を見てわかるように、S L Sは存在し得ない。

円弧セグメント 対 円弧セグメント

ふたつの円弧セグメントの半径を各々、 r_1 、 r_2 とし、中心を $(-x_0, 0)$ 、 $(+x_0, 0)$ となるように座標系を定める。二つの円の位置関係で3つの場合に分けて考える。

1. 二つの円に共通部分がない場合 ($|r_1| + |r_2| \leq 2x_0$)

軌跡は以下のような双曲線となる。

$$\begin{aligned}
x &= d \sec \theta \\
y &= \sqrt{x_0^2 - d^2} \tan \theta \\
\text{where } 2d &= r_2 - r_1, |\theta| < \pi/2
\end{aligned}$$

2. 一方の円が他方の円を含んでいる場合 ($|r_1| + |r_2| > 2x_0 \wedge ||r_1| - |r_2|| \geq 2x_0$)

(a) $r_1 + r_2 \leq 0$ の場合には S L S は存在しない。

(b) $r_1 + r_2 > 0$ であって、 $r_1 \cdot r_2 < 0$ の場合には以下の楕円になる。

$$\begin{aligned} x &= d \cos \theta \\ y &= \sqrt{d^2 - x_0^2} \sin \theta \\ \text{where } 2d &= r_1 + r_2, |\theta| \leq \pi \end{aligned}$$

(c) $r_1 + r_2 > 0$ であって、 $r_1 > 0 \wedge r_2 > 0$ の場合には以下の楕円になる。

$$\begin{aligned} x &= d \cos \theta \\ y &= \sqrt{d^2 - x_0^2} \sin \theta \\ \text{where } 2d &= |r_1 - r_2|, |\theta| \leq \pi \end{aligned}$$

3. 共通部分はあるが包含関係にはない場合 ($|r_1| + |r_2| > 2x_0 \wedge ||r_1| - |r_2|| < 2x_0$)

(a) $r_1 \cdot r_2 < 0$ の場合には以下の楕円となる

$$\begin{aligned} x &= d \cos \theta \\ y &= \sqrt{d^2 - x_0^2} \sin \theta \\ \text{where } 2d &= |r_1 + r_2| \\ |\theta| &< \arccos \frac{|r_1| - |r_2|}{2x_0} \quad (r_1 < 0 \wedge r_2 > 0) \\ \arccos \frac{|r_1| - |r_2|}{2x_0} &< |\theta| < \pi \quad (r_1 < 0 \wedge r_2 > 0) \end{aligned}$$

(b) $r_1 > 0 \wedge r_2 > 0$ の場合には以下の双曲線の一部になる。

$$x = d \sec \theta$$

$$y = \sqrt{x_0^2 - d^2} \tan \theta$$

$$\text{where } 2d = r_1 - r_2, |\theta| < \arccos \frac{2x_0}{r_1 + r_2} \text{ or } \pi/2 < |\theta| < \pi$$

(c) $r_1 < 0 \wedge r_2 < 0$ の場合には以下の双曲線の一部になる。

$$x = d \sec \theta$$

$$y = \sqrt{x_0^2 - d^2} \tan \theta$$

$$\text{where } 2d = r_2 - r_1, \arccos \frac{2x_0}{r_1 + r_2} < |\theta| < \pi/2$$

図 3.8 はふたつの円弧が交わっている場合の状況を示しているが、A と B の法線の交点の軌跡は Region I では楕円になり、Region II では双曲線となる。

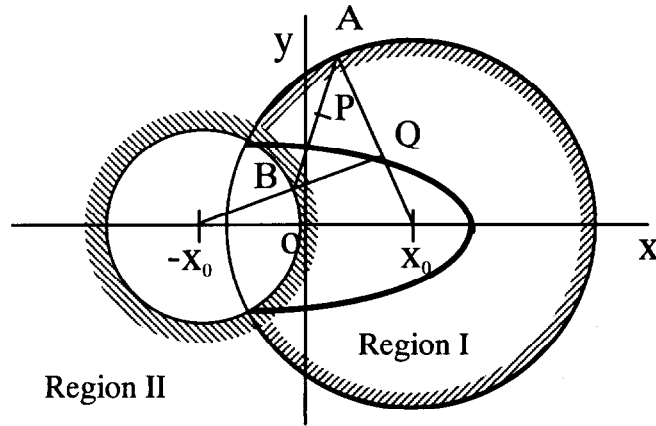


図 3.8: 二つの円弧セグメントが交わっている場合の P の軌跡。Region I では楕円になり、Region II では双曲線となる。

S L S の存在は θ の空間で共通区間があるかどうかによって判定する。具体的には、各々の円弧セグメントの存在区間を、 $[\theta_{i1}, \theta_{i2}]$, $i = 1, 2$ としたときに、

1. Q の軌跡が楕円の場合には、

$$\begin{aligned}\alpha_{1j} &= \cos^{-1} \left(\frac{d \cos \theta_{1j} - x_0}{d - x_0 \cos \theta_{1j}} \right) \\ \alpha_{2j} &= \cos^{-1} \left(\frac{d \cos \theta_{2j} + x_0}{d + x_0 \cos \theta_{2j}} \right)\end{aligned}$$

2. Q の軌跡が双曲線の場合、

$$\begin{aligned}\alpha_{1j} &= \cos^{-1} \left(\frac{d - x_0 \cos \theta_{1j}}{d \cos \theta_{1j} - x_0} \right) \\ \alpha_{2j} &= \cos^{-1} \left(\frac{x_0 \cos \theta_{2j} - d}{d \cos \theta_{2j} - x_0} \right)\end{aligned}$$

によって α_{ij} に変換し、 $[\alpha_{11}, \alpha_{12}]$ と $[\alpha_{21}, \alpha_{22}]$ に共通区間が θ の定義範囲に存在するかどうかを判定する。共通区間が存在したとき、それを $[\theta_s, \theta_e]$ とすると、S L S の解析式は、

1. Q の軌跡が楕円の場合には、 $\theta \in [\theta_s, \theta_e]$ の範囲で、

$$\begin{aligned}x &= \frac{1}{2} \frac{d}{d^2 - x_0^2 \cos^2 \theta} (2(d^2 - x_0^2) \cos \theta + (|r_1| - |r_2|) x_0 \sin^2 \theta) \\ y &= \frac{1}{2} \frac{\sqrt{d^2 - x_0^2} \sin \theta}{d^2 - x_0^2 \cos^2 \theta} (2d^2 - x_0(|r_1| - |r_2|) \cos \theta)\end{aligned}$$

2. Q の軌跡が双曲線の場合には、 $\theta \in [\theta_s, \theta_e]$ の範囲で、

$$\begin{aligned}x &= \frac{1}{2} \frac{d}{x_0^2 \sec^2 \theta - d^2} (2(x_0^2 - d^2) \sec \theta + (|r_1| + |r_2|) x_0 \tan^2 \theta) \\ y &= \frac{1}{2} \frac{\sqrt{x_0^2 - d^2} \tan \theta}{x_0^2 \sec^2 \theta - d^2} (2d^2 + x_0(|r_1| + |r_2|) \sec \theta)\end{aligned}$$

で与えられる。

3.5 S L S の計算例

いくつかの図形に対して計算した S L S の例を示す。図 3.9 は工具プライヤのシルエット画像に対して計算した S L S であるが、全体の形の対称性を示す軸が検出されているとともに、部分ごとの中心軸が求まっている。予想されない軸も存在しているが、これらは部分図形をまたがって検出された本来必要でないものであるが、これらを抑制する方法については後述する。図 3.10 はいくつかの別の工具に対して得られた S L S を示したものである。

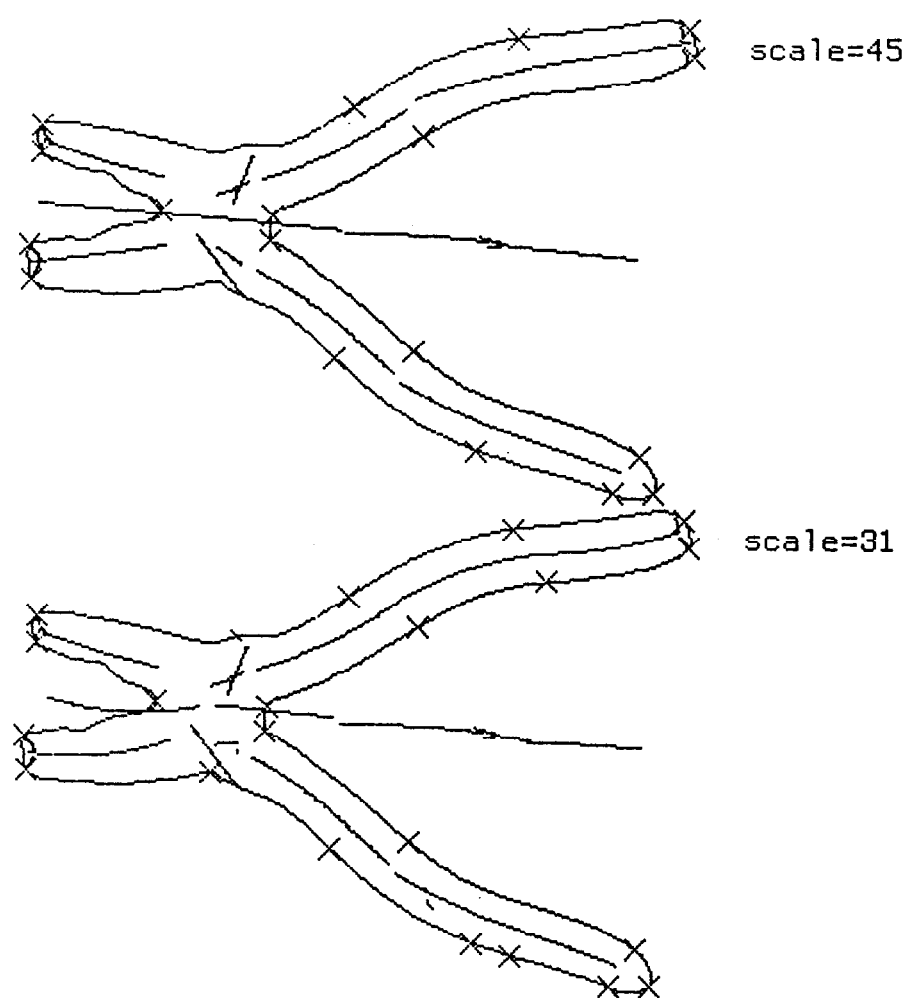


図 3.9: 工具プライヤのシルエット画像に対して計算された S L S。各々、記載された σ の値で検出した輪郭分割セグメントから計算したもの

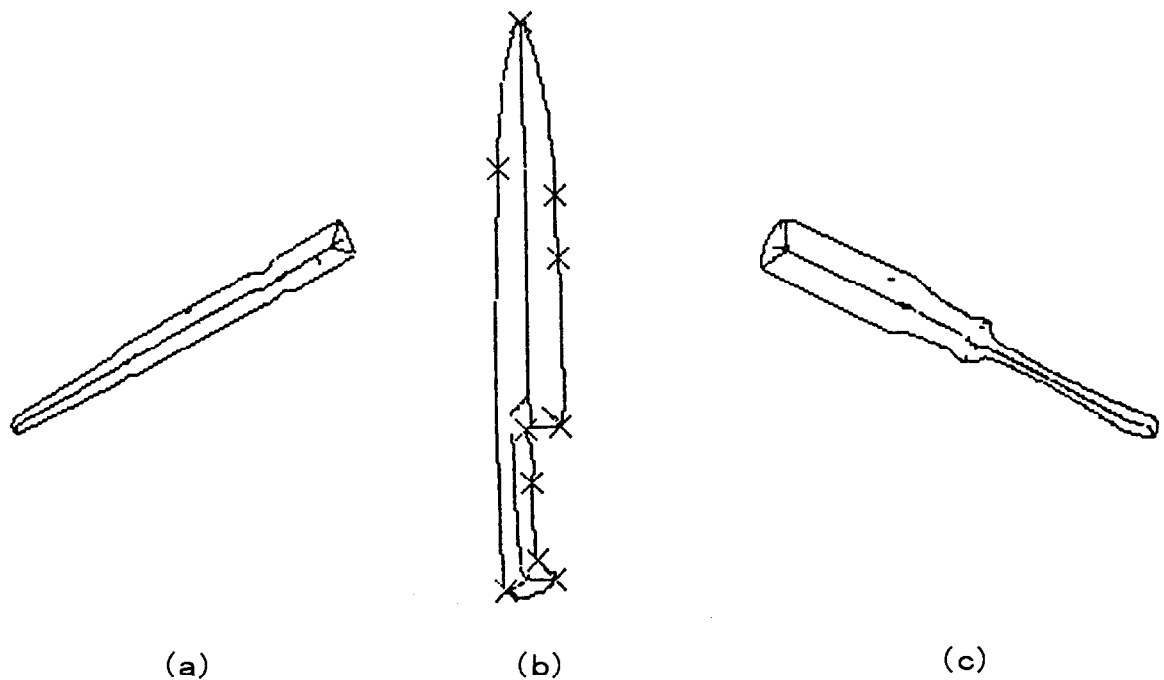


図 3.10: いくつかの工具のシルエット画像に対して計算されたSLS。(a) パンチ、(b) ナイフ、(c) ドライバ

3.6 負のSLS

図3.11は通常のSLSとともに「負のSLS」を並べて表示したものである。負のSLSは背景部の局所対称によって作られたもので、局所対称の定義において法線ベクトルを逆向きにすることで正のSLSと同様に計算することができる。図3.11におけるハンマーの図形は一般的に柄の部分図形と頭の部分図形とから構成されているが、部分図形の結合部に対応したものが検出されているのが分かる。

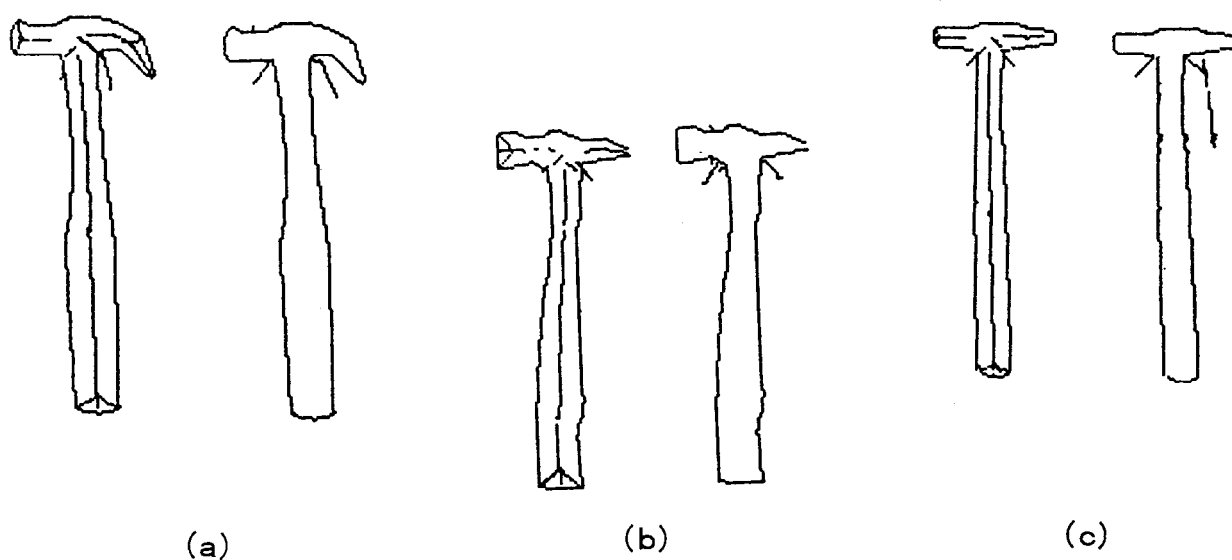


図 3.11: いくつかのハンマーのシルエット画像に対して計算されたSLS。左に正のSLS、右に負のSLSを並べて表示している

3.7 図形のネットワーク表現

第2章では二次元図形を輪郭プリミティブを基本として記述した。本章では前節までにSLSによる領域記述法を示した。輪郭のセグメントによってSLSが規定され、逆にSLSによって図形領域が規定されると同時に対応する輪郭のセグメントが定まる。このように輪郭と領域とはお互いに補完するものであるため、これらを統合した図形記述が考えられる。図3.12にその概念図を示す。SLSと輪郭セグメントは各々、関連付けられているのでこのような表現方式は容易に求めることができる。このような表現を行う利点は、例えばCorner点Bが他の物体に隠されたような場合でも、SLSで規定される形状によりこの図形を同定することが可能となる。これは冗長な表現形式であるためであり、序論で述べた図形記述への要請の第三番目に答えたものである。図3.13はこのような表現のもととなるハンマー図形に対して計算したSLSによる領域記述の結果である。

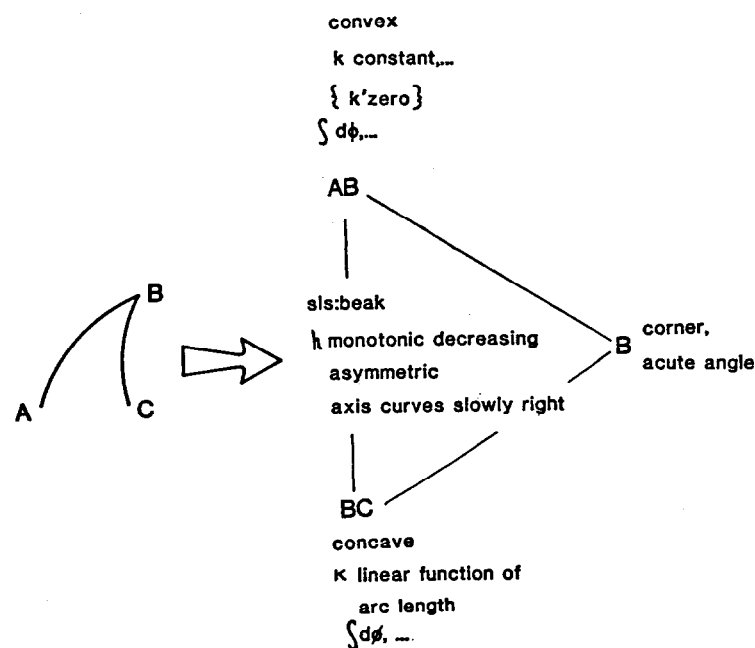


図 3.12: 刃先のような図形のネットワーク表現：輪郭情報と領域情報の統合

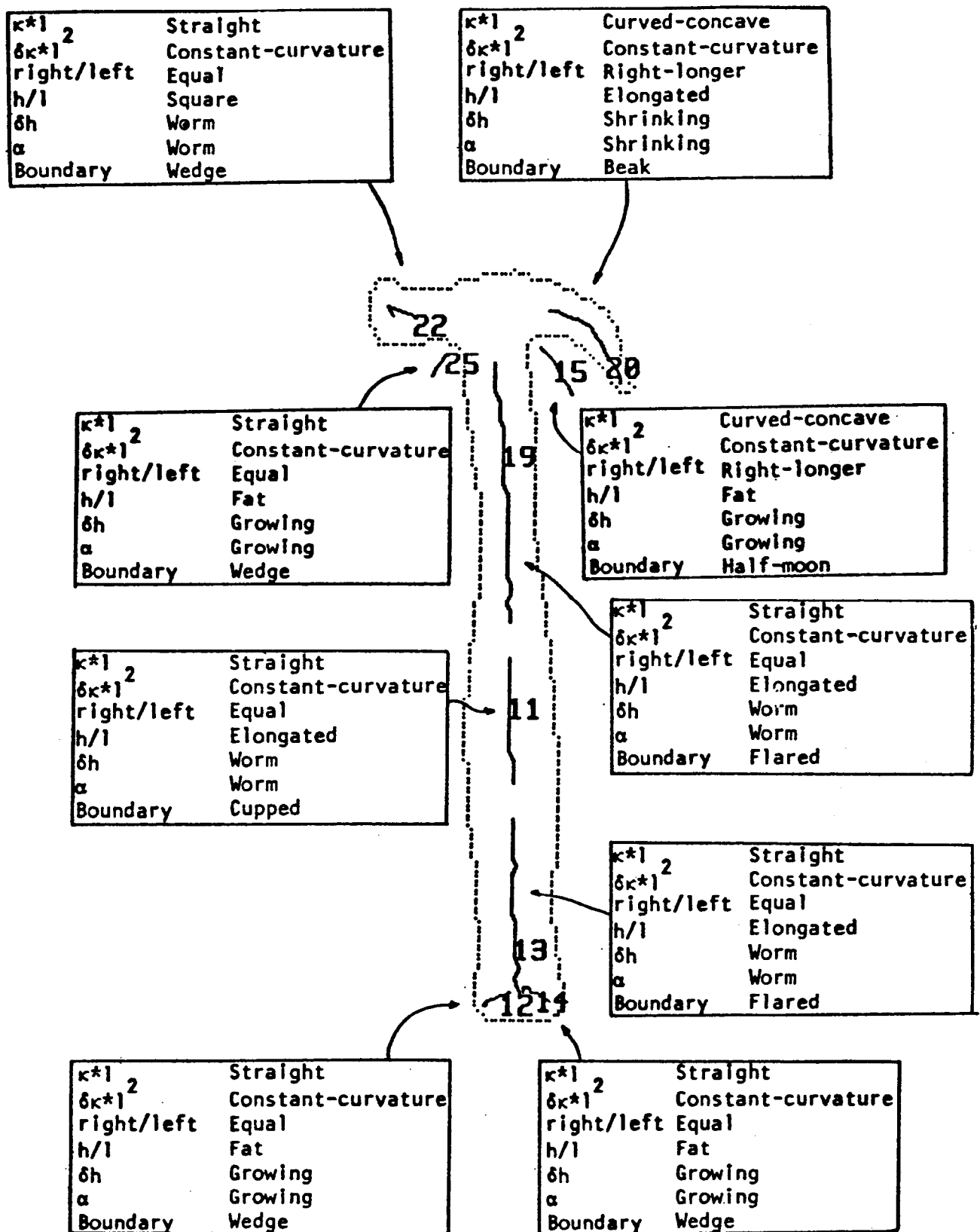


図 3.13: 「ハンマー」図形に対して計算したSL Sの記述

3.8 部分図形の検出

これまでに例示した図形からも明らかなように、物体の形状は複数の部分図形に分解できるのが通常である。例えば、ハンマーは頭と柄に分解できる。本節では、与えられた図形から部分図形を切り出す方法について考察する。

これまでの過去の例では、部分図形の存在を示唆する信号をどこに求めるかが追求されてきた。[Blum 78]のSATは元々、部分図形への分割を目的に提案されたものであるが、長方形のような図形でも軸が5本検出されるなど、実際にはうまく動作しない。[Nevatia&Binford 77]はGeneralized Cylinderの軸の方向が大きく変化したとき部分図形の存在を示唆することを主張しているが、ハンマーの頭と柄との結合のような簡単なものでも検出できない。

[Marr 77]は部分図形の接合は「side-end」か「end-end」の2種類に分類できると主張した。「side」と「end」の概念は曖昧ではあるが、図形の「側面」に他の図形の端が接合している「side-end」の幾何学的条件を解析し、「side-end」接合には対応する凹部が存在し、理想的にはその接線方向変化の和が π になるというヒューリスティックな法則を提案した。この考えは実際にプログラム化されてはおらず、現実の図形に対して有効に働くものではないが、後世に多くの影響を残している。本論文においても、この考えに基づいて複数の部分図形が接合する状況について考察する。

3.8.1 図形の「Side」と「End」

部分図形の接合を議論するためには、図形のSide（側面）とEnd（端）を区別する必要がある。SLSの軸は、その定義により、図形を二つの側面に分けている。ただし、軸から側面の輪郭までの距離との対比で、軸の長さがあまりにも短い場合は側面とは言えない。したがって、厳密には規定できないが、軸にはある長さが要求される。こういう図形を「elongated」と呼ぶことにする。そういう条件が成り立てば、SLSの軸を生成している輪郭成分は「Side」と定義することができる。さらに、SLSの軸が終了する部分に存在する輪郭成分は「End」とあるといえる。部分図形の接合は複雑であり、全ての状況を理論化することは困難であるため、ここでは「elongated」な図形（これも曖昧な概念ではあるが）を対象としている。文字図形は長いストロークで構成されており、この範疇に含まれる。

3.8.2 部分図形接合の検出

ここでは「elongated」な部分図形の接合の存在を発見するためのふたつのルールを提案する。ルール1は前に述べた、ふたつの対応する凹部の存在が部分図形の接合を示唆するという Marr の法則を一般化したものである。

ルール1： 対応する二つの凹部が存在するとき、その凹部を代表する負の S L S を作り出している輪郭成分（4つある）のうちの2つの輪郭成分が正の S L S を作り出しているとする、その正の S L S で定義される図形領域は部分図形である。

図 3.14 がその状況を示している。輪郭成分 a と b とから負の S L S (λ) が作られており、輪郭成分 c と d とから負の S L S (ν) が作られている。さらに、輪郭成分 b と c とから正の S L S (μ) が作られているすると、S L S (μ) は a や d の輪郭成分で構成されている本体に接合している部分図形を示すということを主張するものである。S L S の軸と輪郭成分との関係において、図 3.14 の右側に示すネットワークがあれば部分図形の存在を示唆しているというのがルール1である。図 3.14 の下側は図形と背景を入れ替えて、物体の側面に穴を開けた図形を示している。このような穴は負の部分図形として取り扱うことができるため、S L S の符号を反対にすることで、同様のルールが適応できる。図 3.15 にルール1で検出された部分図形接合部を示す。スケールの違いで検出結果が異なる。

ルール1は部分図形の接合の存在を発見するための十分条件ではあるが必要条件ではない。二つの凹部の存在を要求しており、これは厳しい条件である。そのため、一つの凹部しか存在しない場合でも部分図形の接合を示唆するための条件をルール2とする。

ルール2： 一つの輪郭成分が正の S L S と負の S L S の共通の輪郭成分となっている場合、その正の S L S が作り出す図形領域が「elongated」であるか、終端している場合には、その図形領域は部分図形である。

ルール2で検出される部分図形の接合の例を 3.16 に示す。一方の輪郭が滑らかに接合された「end-end」接合や、Corner がはっきりしない「side-end」接合の場合の検出ができる。ルール2は凹部がひとつしか存在しない場合でもそれに関連する図形部分が長細い場合には部分図形である可能性があるという考えに基づいている。

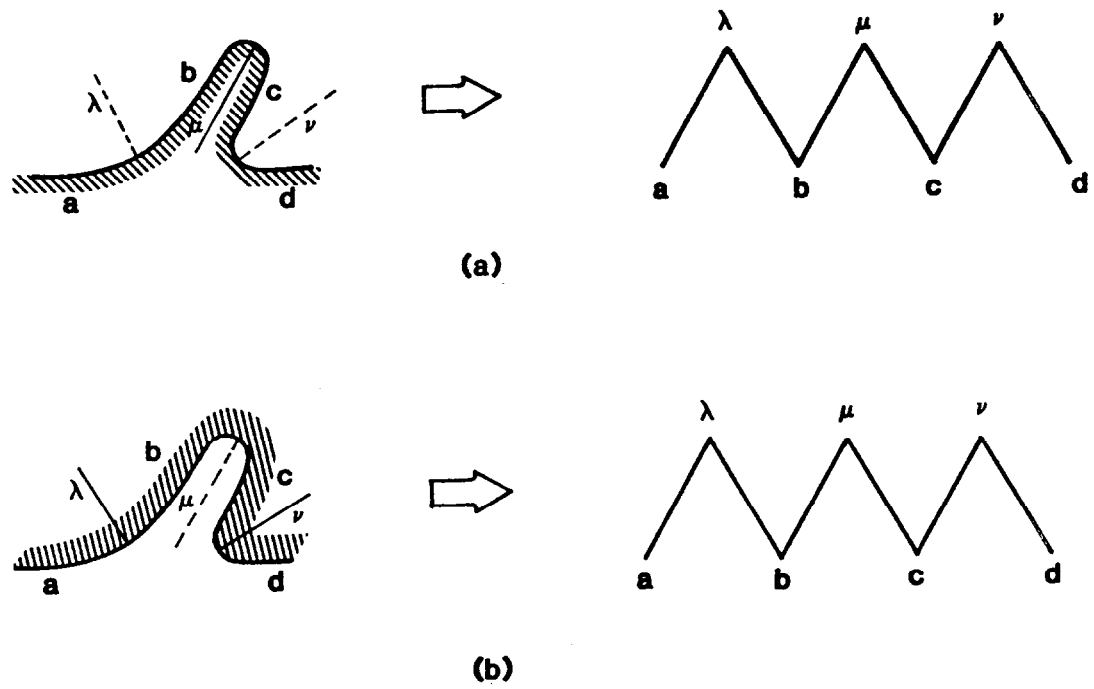


図 3.14: ルール1に基づいて検出される部分図形接合のパターン

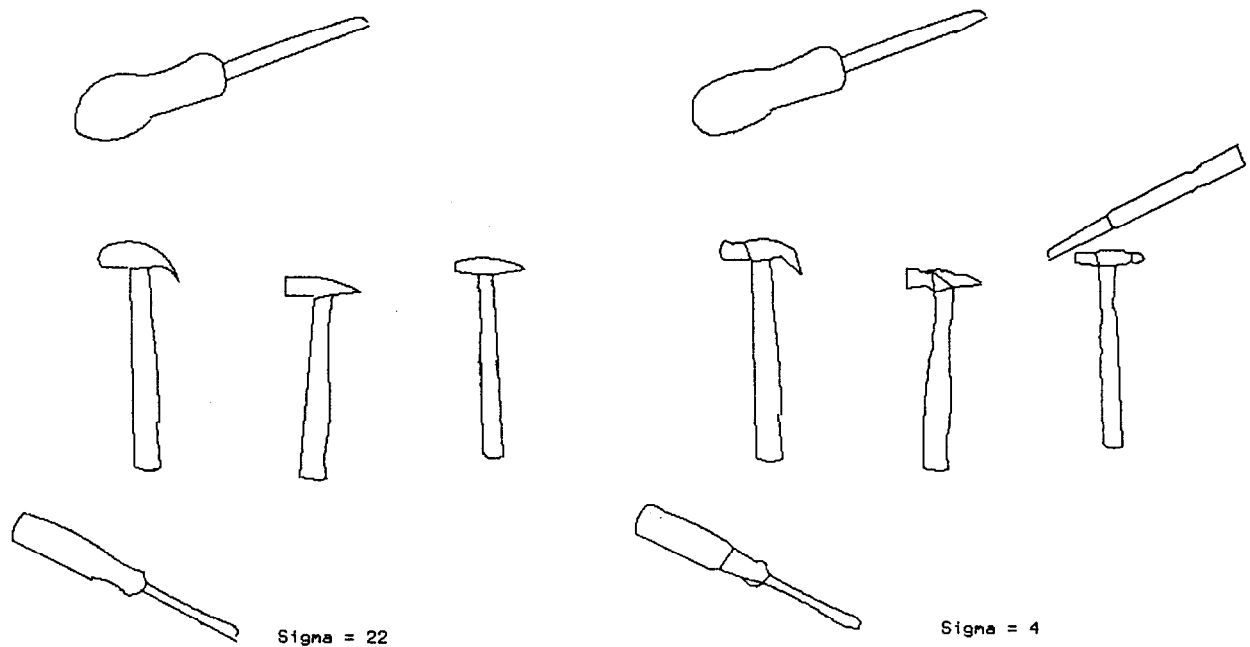


図 3.15: ルール1によって検出された部分図形接合部（凹のコーナを結んだ実線）。 $\sigma = 22$ （左）と $\sigma = 4$ （右）

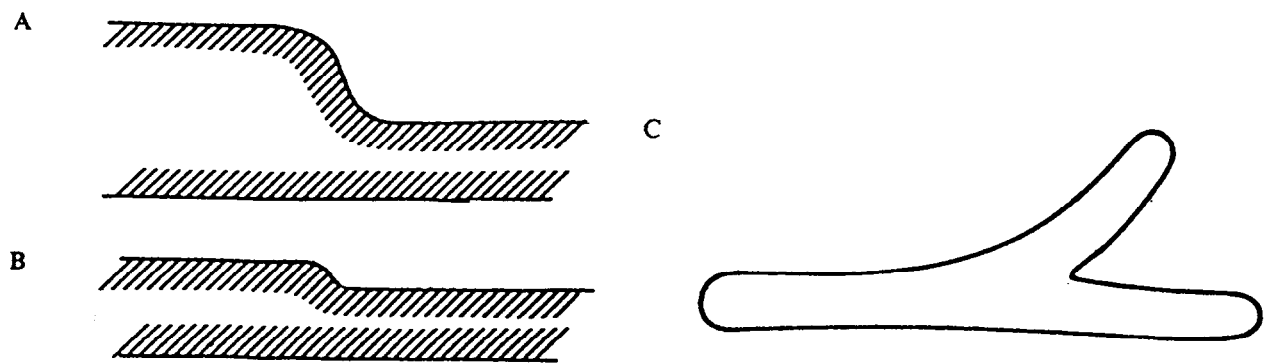


図 3.16: ルール2で発見できる部分図形の接合の例

3.9 S L Sの再計算

3.5章で示したS L Sの計算例では、思いもよらないS L Sの軸が存在していた。これらは異なる部分図形の輪郭成分から生成された軸であり、本来あってはならないものである。部分図形の接合が認識されてはじめて、このような軸の除去が可能となる。部分図形をまたがって計算される偽のS L Sの除去の処理の例を欠けのある長方形に適用した結果を図3.17に示す。

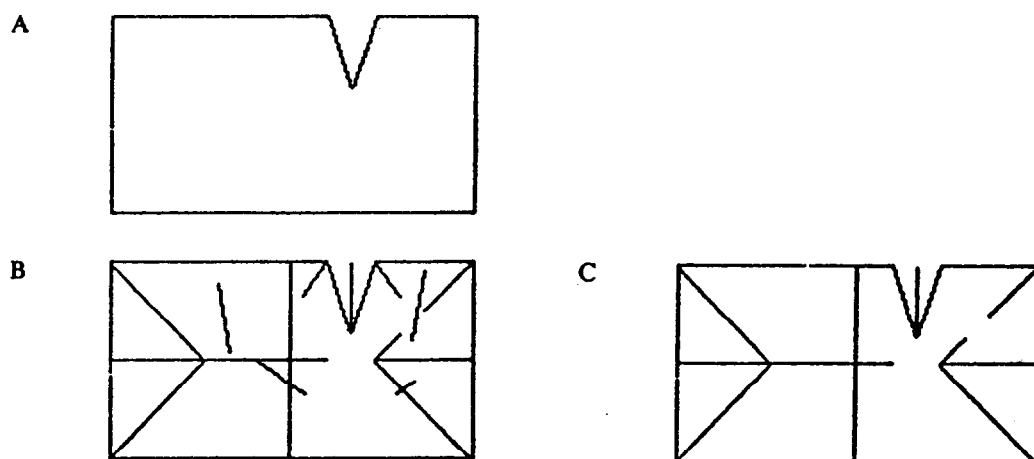


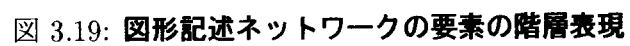
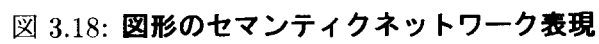
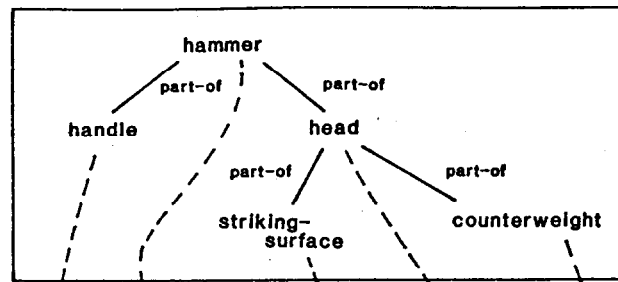
図 3.17: 部分図形をまたがって計算されたS L Sの除去。(A) 欠けのある長方形（入力図形）、(B) 入力図形に対して計算されたS L Sの軸。多くの偽のS L Sが含まれている、(C) 偽のS L Sを除去した結果

3.10 図形の階層的ネットワーク表現

図形の輪郭ベース記述と領域ベース記述の統合については前に述べた。実はこれらの個々の記述はスケールに応じた階層記述である。さらに部分図形の抽出についても述べた。これらをベースとして、「もの」の概念をセマンティックネットワークで表現する場合、形状に即した階層的ネットワーク表現が可能となる。図 3.18 は「ハンマー」という「もの」の表現を示したものである。上の矩形内には「ハンマー」のジェネリックな表現が示されている。「ハンマー」は「頭」と「柄」からなり、頭は「打面」と「カウナタウエイト」からなっている。それらの関係が「part-of」のリンクで示されている。下の矩形内にはこれらの構成要素の具体的な形状が記述されている。「ハンマー」とは長細い（Worm 型）の「柄」と直線の S L S を持つ矩形の「頭」とが「Side-End」接合したものとなっている。さらに、個々の構成要素はスケールに応じて複数の記述を持つことができる。図 3.19 にその例を示す。「ハンマー」を他の物体（例えばナイフ）と区別して識別する場合には大きなスケール（Scale2）を用いればよいが、「ハンマー」のなかの種類を同定する場合には小さいスケール（Scale1）を用いる必要がある。図 3.20 に「a-kind-of」リンクで表現される「一般的なハンマー」のセマンティックネットワークを示す。この図の中には図 3.18 や図 3.20 の表現が暗に含まれており、これは形状に立脚した物体の階層的記述といえる。[Connell 85] はこのような階層的記述によって飛行機の形状を認識する学習プログラムを開発している。

3.11 まとめ

局所的に対称な関係にある二つの輪郭点の中点を滑らかに結んだ曲線（SLS）は図形の領域を記述する方法であり、これを効率的に計算するアルゴリズムを示した。この方法は Curvature Primal Sketch の結果を用いて輪郭のセグメント化を行い、これを直線・円弧近似することで解析的に求めるものである。さらに、領域記述と輪郭記述と統合したネットワーク表現を提案し、この中である形の部分ネットワークを探索することで部分図形を検出するルールを提案した。さらに、これらの記述はスケール解析に立脚しているため、階層表現とすることもでき、物体の階層的セマンティックネットワーク表現に発展できることを示した。



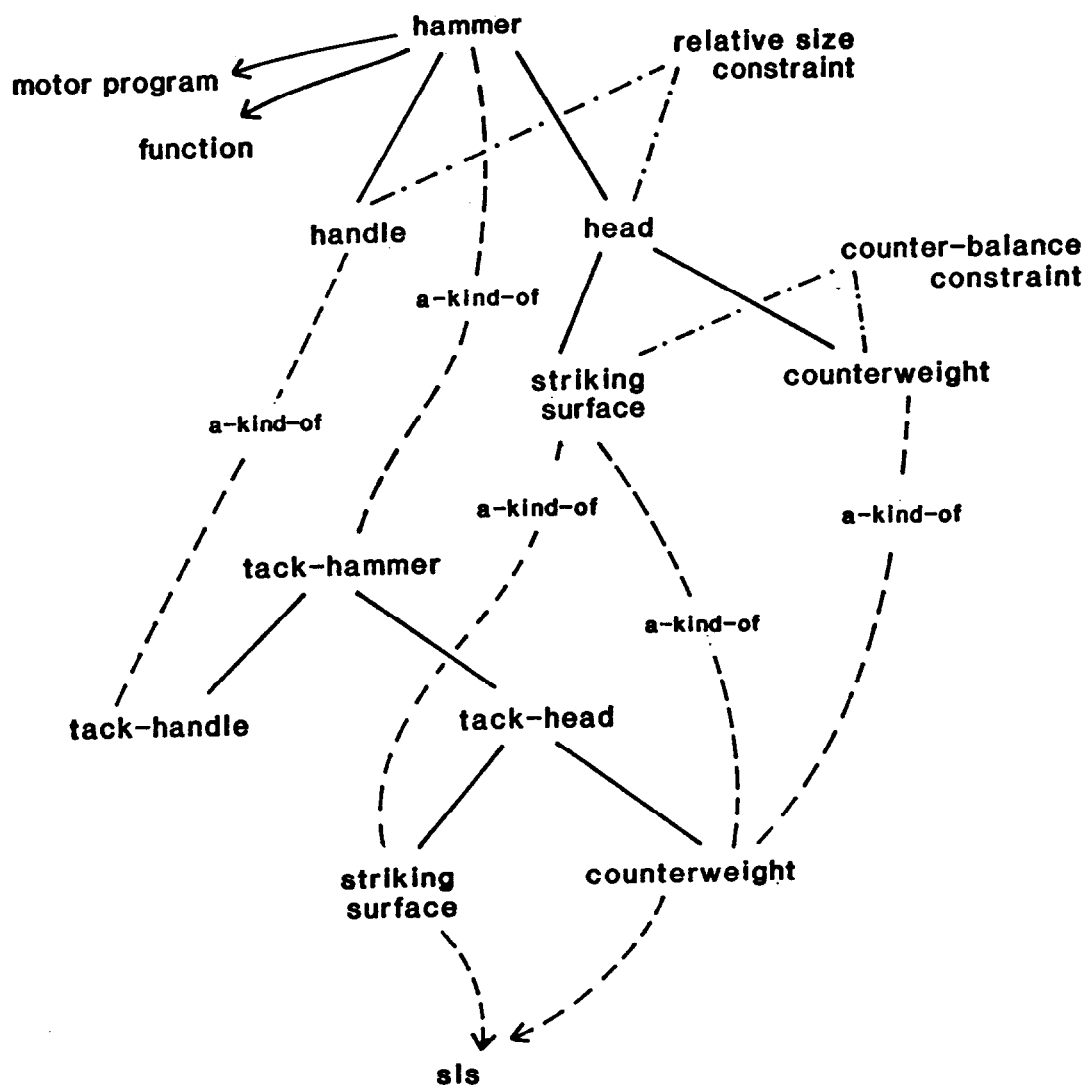


図 3.20: 物体の形状に立脚した階層的記述

第4章 輪郭記述に基づく手書き文字認識

本章では提案する輪郭記述方式を手書き文字の認識に適用し、変形が甚だしい文字に対しても高精度な認識が可能であることを述べる。

4.1 文字認識の従来研究

文字認識の手法は大きく分けて2つの方法、つまり、「重ね合わせ法」と「構造解析法」に分類することができる。重ね合わせ法は文字画像をレファレンス画像に重ね合わせて、相関の度合いを計算することにより、文字の認識を行う。2値画像どうしを単純に重ね合わせるテンプレートマッチング法が最も単純な方法であるが、印刷文字程度の変形しか許容できない。複合類似度法 [Iijima 73] は、ひとつのカテゴリに含まれる多数の文字パターンをK-L展開し、主要な固有ベクトルで張られる部分空間でそのカテゴリを表現する。入力パターンと部分空間との相関度を定義して、この値をもとに認識する。これは線形近似の範囲内であるが、漢字を含む手書き文字の認識においても実用的な性能を示している。近年では顔画像の認識にも同様の方法が用いられている。

一方、構造解析法 [Fu 74] は文字図形の形状特徴を抽出し、これに基づいて認識を行うというものである。この範疇に含まれる方式は多数あるが、もっとも単純なものは、重心位置、モーメントなど文字図形の広域的特徴を何種類か計測し、多数の文字パターンに関する統計的性質に基づいて認識を行っている。しかし、この程度の処理では大分類的な認識精度しか得られない。図形特徴の種類を複雑化していくと、認識精度を向上させることができるが限界はある。初期の郵便番号読み取り装置にはこのような方式が用いられていた。構造解析法の別の方法としては、図形の領域、輪郭、背景のどれかひとつに着目して、文字パターンの形状を記述し標準パターンと入力パターンとの間で特徴系列または特徴群のマッチングの度合いを計算し、認識するというものがある [You 79] [Tsai 80b] [Tsai 85]。このような方法において、文字パターンの大幅な変形に対処して精度よく認識するためには、ノイズに影響されないロバスト性と、微小な形状の特徴をも検出できるローカルサポートをもつ形状記述方式が必要である。本章で述べる文字認識の手法は、

大きな変形を持つ自由手書き文字パターンを高精度に認識することを目的としたものである。

4.2 認識方式の概要

本方式においては、各文字パターンの輪郭を曲率の主要な変化点を基準として分割し、セグメント化する。輪郭セグメントは、結果的に、端点、凸部、凹部、直線部に分類される。さらに、各セグメントには曲率の大きさ、始点の方向、終点の方向、長さなどの計測値を特徴量として付与し、ひとつの文字パターンは特徴量を付与された輪郭セグメント系列で表現されることになる。ひとつのカテゴリを代表する標準パターンは、上と同様に、輪郭セグメント系列で与えられるが、入力パターンのセグメント列との対応づけを depth-first の探索で行い、対応づけが可能なカテゴリをその入力パターンが属するカテゴリであると認識する。

手書きの数字に対して認識実験を行った結果、標準パターン学習用については98.9%（誤読0.2%）、テストパターンについては98.1%（誤読0.2%）であった。また、手書きアルファベットとカタカナについては、各々、95.5%（誤読0.2%）、96.8%（同0.6%）であった。更に学習パターンを増加させて性能向上したものが、商用OCRに搭載されている。

以下において、入力された文字パターンから輪郭の方向コード列を抽出するアルゴリズム、輪郭のセグメント化後の特徴系列の抽出と辞書とのマッチングの具体的な方法、および、認識実験結果について述べる。

4.3 方向コード列の抽出

図形の輪郭情報は、図形の輪郭を追跡して各輪郭点での輪郭の接線方向を4方向または8方向などに量子化した方向コード列として与えられる。この手法はいわゆるフリーマンチェーンコードと称されるもので、図4.1に示すような方向コードを用いることができる。例として図4.2に示したものは、文字図形を輪郭点Sを始点として、右回りに追跡して得られる方向コードを図形の外部に表記したものである。このような方向コード列は元の図形の形状情報を保存しており、従って、この方向コード列から元の図形を完全に再現できるという性質を有している。

このような方向コード列は45°で量子化されているため、コード列から輪郭の接線方向関数を計算する場合には平滑化などの処理が必要である。しかし、方向コード列の中には、方向コード

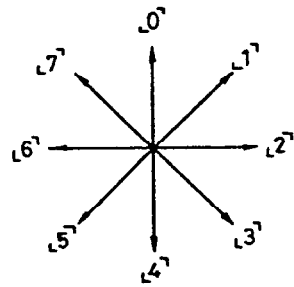


図 4.1: 方向コード (8方向) の例

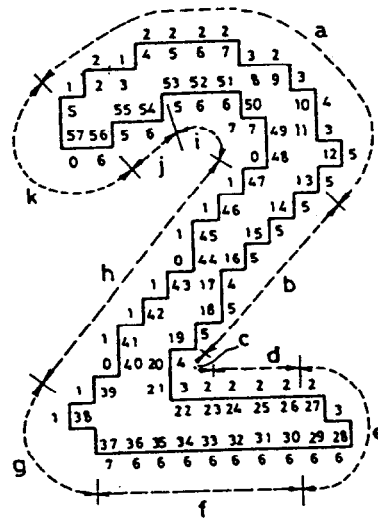


図 4.2: 文字図形に対する方向コード列の例

が「0」から「7」へ、あるいは、逆に、「7」から「1」に変化するような不連続点が存在し、そのままでは平滑化や微分などのコンボリューション演算ができないという問題がある。図 4.3(a) は図 4.2 の方向コード列を周長を横軸としてプロットしたものであり、丁度 g に対応した部分に不当な不連続が観測できる。本来的には図 4.3(b) のようなグラフになるべきである。そのため、ここでは、等価方向コード変換を導入して、連続性が保持できるようにする。

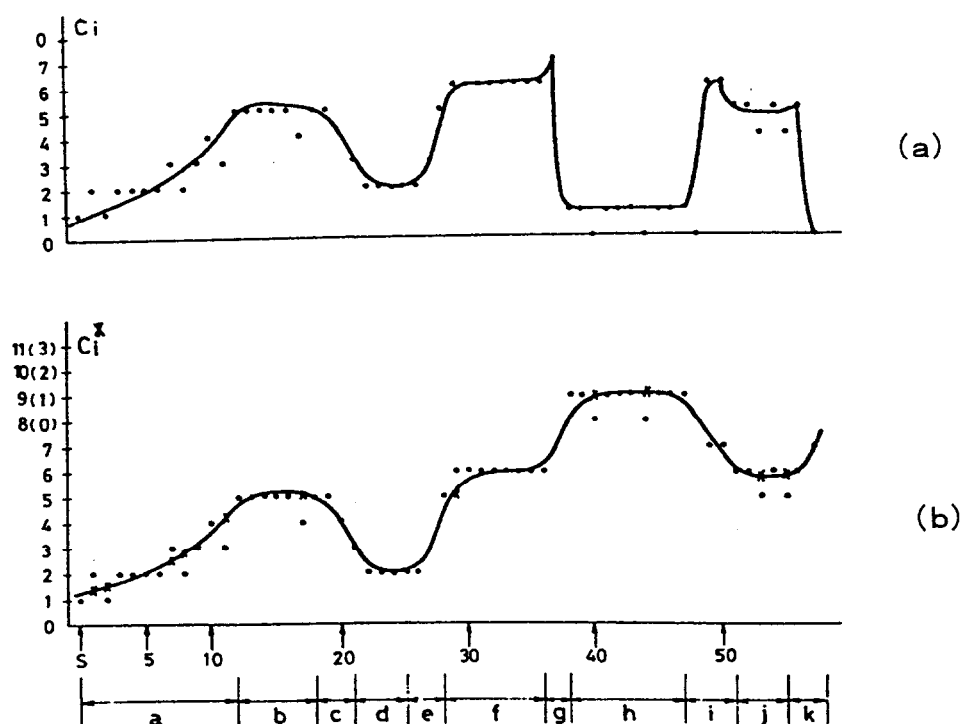


図 4.3: 方向コードを周長を横軸としてプロットしたもの。(a) 方向コードを 0～7 の範囲に限定した場合 (b) 等価方向コード変換した場合

4.4 等価方向コード変換

等価方向コード変換は Mod8 で得られている方向コードに $8 \cdot n$ の値を加算あるいは減算して、連続したコード列に変換する操作である。図 4.4 は方向コード列 $C_1 \cdots C_{i-1} C_i \cdots C_n$ の C_{i-1} と C_i との組み合わせで決まる図形の局所形状の全てを表示したものである。この中で、例えば、 $C_{i-1} = 5$ の

行を見ると、6種類の形状しかなく、 $C_i = 1$ と $C_i = 2$ の形状は存在しない。さらに、 $C_i = 5$ の場合は $C_i - C_{i-1} = 0$ で接線方向の変化がゼロであることを、 $C_i = 3$ の場合は $C_i - C_{i-1} = -2$ で接線方向の変化が -90° の凹部であることを示しているのに対し、 $C_i = 0$ の場合は $C_i - C_{i-1} = -5$ となり、凸である形状での接線方向の変化が負の値となってしまう。これは、 $C_i = 0$ ではなく $C_i = 0 + 8$ であるべきであることを示している。また、 $C_{i-1} = 0$ で $C_i = 6$ の場合には C_i から8を減算する必要のあることがわかる。つまり、 3×3 の近傍のパターンから方向コード（0～7）を求めたときに、ひとつ前の輪郭点の方向コードの値によって、8を加減算する必要のある場合があることがわかる。図4.5は8を加減算する必要のある 3×3 近傍パターン全てを示したものである。

以上を考慮して、等価方向コード変換の定式化を行うと以下ようになる。輪郭画素 i について 3×3 の8近傍パターンから求められた方向コード（0～7の値をもつ）を C_i^* 、ひとつ前の隣接する輪郭画素（ $i-1$ ）について既に決定された輪郭方向コードを C_{i-1} とすると、連続性を保存して求められる輪郭画素 i の輪郭方向コード C_i は次の式で決定される。

$$C_i = C_i^* + [C_{i-1}/8] * 8 + f(\text{mod}(C_{i-1}, 8), C_i^*)$$

ただし、 $\text{mod}(a, b)$ は a を b で割った余り、 $[a]$ は a の小数点以下を切り捨てた整数部を与える演算であり、 $f(a, b)$ は、

$$f(a, b) = \begin{cases} 8 & \text{if } b = 0, \quad 4 \leq a \leq 7 \\ 8 & \text{if } 1 \leq b \leq 2, \quad 6 \leq a \leq 7 \\ -8 & \text{if } b = 7, \quad 0 \leq a \leq 1 \\ 0 & \text{otherwise} \end{cases}$$

で定義される。第2章における接線方向関数はこのようにして求められたものである。

4.5 文字図形の輪郭セグメント特徴系列の作成

本節では、各文字パターンの輪郭を曲率の主要な変化点を基準として分割し、セグメント化し、各セグメント毎に認識に必要な特徴量を付与する方法について述べる。前節までに述べた方法で入力パターンの輪郭を方向コード列で表現し、さらに、第2章で述べた Curvature Primal Sketch により主要な曲率の変化点を抽出し、セグメント化する。第2章では一般的な図形の形状を対象としていたのに対し、文字図形は本質的に線図形であるので、クランクなどの複合プリミティブ

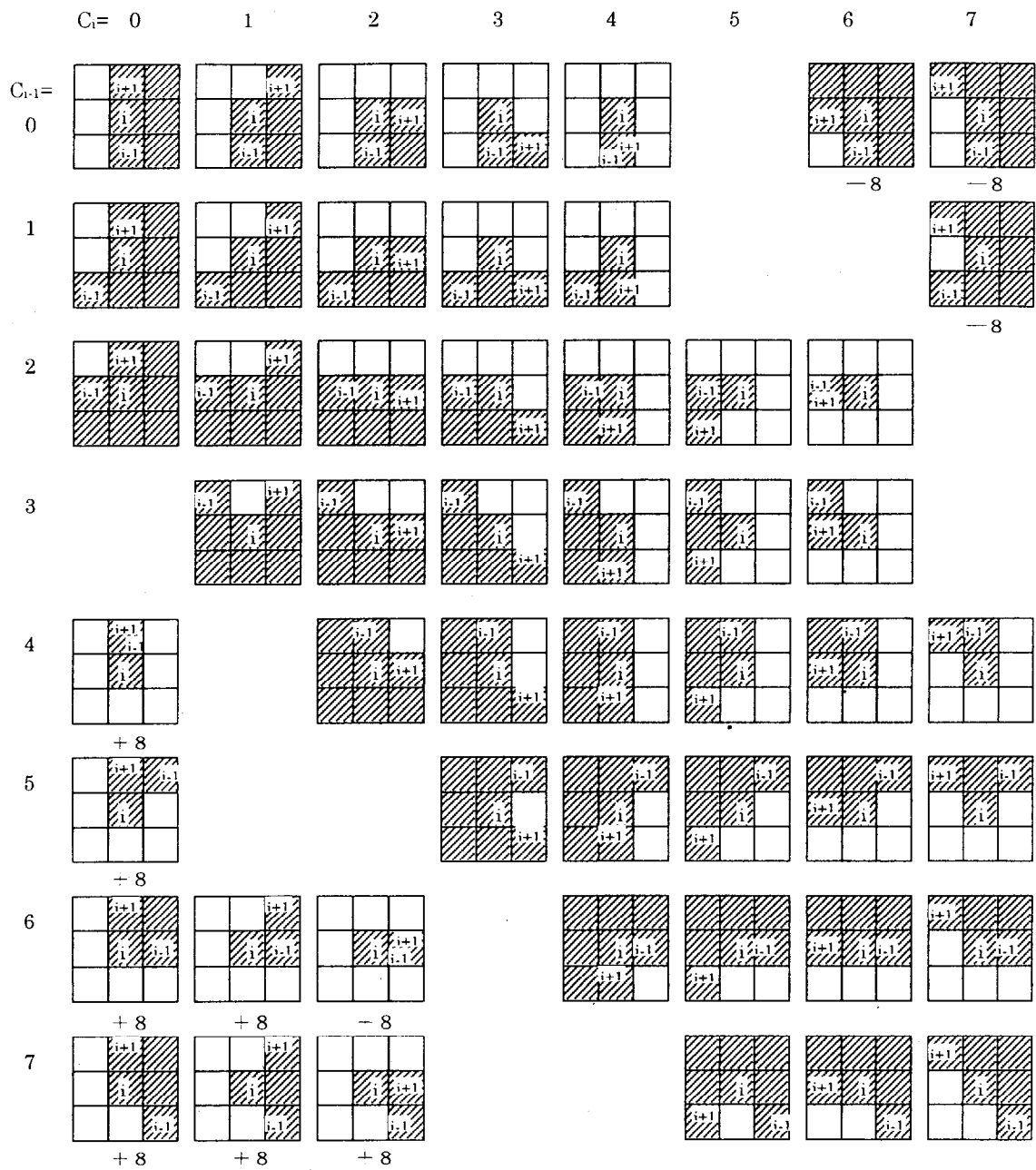


図 4.4: 方向コードの等価変換

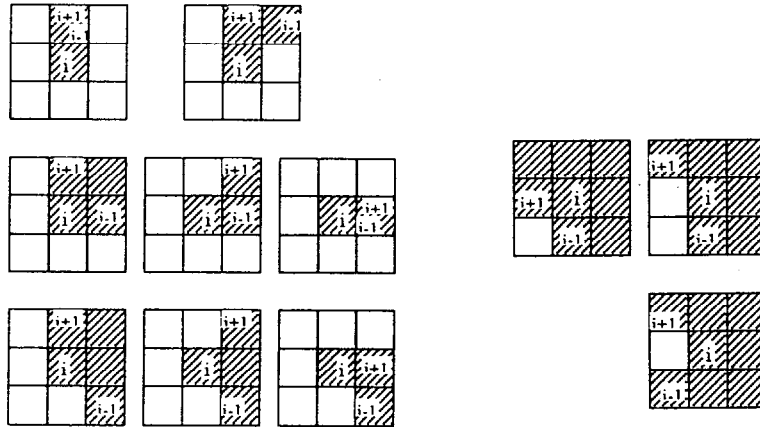


図 4.5: 方向コードに8を加減算する必要がある3 x 3近傍パターン

は不要である。そのため輪郭セグメントは Corner、Smooth Join で区分されるのみである。このようにして作られた各輪郭セグメントは、その平均曲率の符号により凸部、凹部、直線部に分類される。さらに、凸部は曲率の大きいものを端点としてもよい。前節の図 4.2 の例では、セグメント *a*、*e*、*g*、*k* が凸部を示し、セグメント *c*、*i* が凹部、セグメント *b*、*d*、*f*、*h* が直線部を示している。図 4.6 はいくつかの手書き文字についてのセグメント化の結果を示したものである。この図においては、凸部を「#」で、凹部を「o」で、長い直線部を「*」、短い直線部を「x」、非常に短い直線部を「N」で示している。

次の段階では、認識に必要な特徴量を各セグメントに付与する。特徴として採用したものは始・終点の位置、終点の輪郭接線方向、セグメント長の3種類である。始・終点の位置は文字パターンの外接長方形の左下端を原点として、その大きさを正規化した値で表現する。この特徴量はセグメントが文字パターンのどの部分に存在するかの判断のために必要である。始・終点の方向はセグメントがどの方向に向いているかを知るためのもので、例えば、手書き数字「1」の縦のストロークの右側の輪郭セグメントは下方を向いており（注：ここでは輪郭を右回りに追跡している）、その終点の方向は -90° を含むある範囲内になければならないなどの制限を加えることができる。これらの特徴量は後述するマッチングの段階でも使用されるが、さらに類似カテゴリ間の詳細識別において特に有用である。

4.6 認識辞書の構成とマッチング方式

標準パターン（認識辞書）は、入力パターンと同様に、輪郭セグメント系列で与える。ただし、標準パターンの各セグメントには以下のようにして4種類のタイプ分けをしておく。

- TYPE1(S) : 単独の入力セグメントに対応づけられるもので、この対応は必須であるもの
- TYPE2(U) : 複数の入力セグメントを統合して対応づけてもよいもので、この対応は必須であるもの
- TYPE3(I) : 単独の入力セグメントに対応づけられるものであるが、対応がなくてもよいもの
- TYPE4(W) : 複数の入力セグメントを統合して対応づけてもよいもので、かつ、対応がなくてもよいもの

TYPE1はその文字種（カテゴリ）を特徴付けるうえで無くてはならない輪郭セグメントを示している。これは主要な端点、あるいは、角度変化の大きい凸部や凹部に付けられる。TYPE2は長い曲線部のように、少しの輪郭線の変形でセグメント結果が変動してしまうような部分に対応している。TYPE3、および、TYPE4は、多数の文字パターンにわたって存在が不安定であるような輪郭成分に対応している。標準パターンの輪郭セグメントをタイプ分けしておくことによって、無くてはならない特徴と変動していてもよい特徴とを分離することができ、フレキシブルなマッチングを実現することができる。

マッチング処理では、入力パターンのセグメント列と標準パターンのセグメント列との対応づけを depth-first の探索で行う。この具体的な方法はここでは述べないが、[Kurosawa 86] に詳述されている。セグメント列のレジストレーションや探索の際のバックトラッキングの方法が重要である。図 4.7 に数字「3」におけるマッチング結果の典型的な例を示す。標準パターンのセグメント列を英小文字（a～l）で、入力パターンのセグメント列を数字（1～14）で示している。この例では a を開始セグメントとしてマッチングが探索されている。標準パターンセグメント a、f は端点であり、これは必ず存在しないとけないものである。また、d と i はおのおの凹部と凸部であるがこれは「3」という文字パターンを特徴付けるうえでなくてはならないとの考えに基づいている。

複数のカテゴリに対してマッチングが成功した場合には「コンフリクト」となり、さらに詳細な識別処理が必要とである。図 4.8 はこのような例を示している。図 4.8(a) は文字「1」であるが、

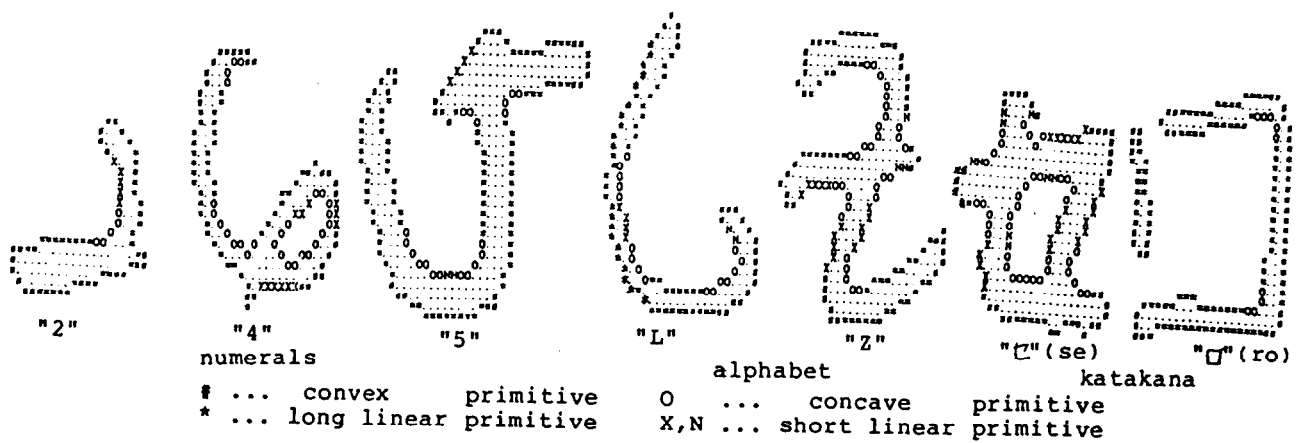
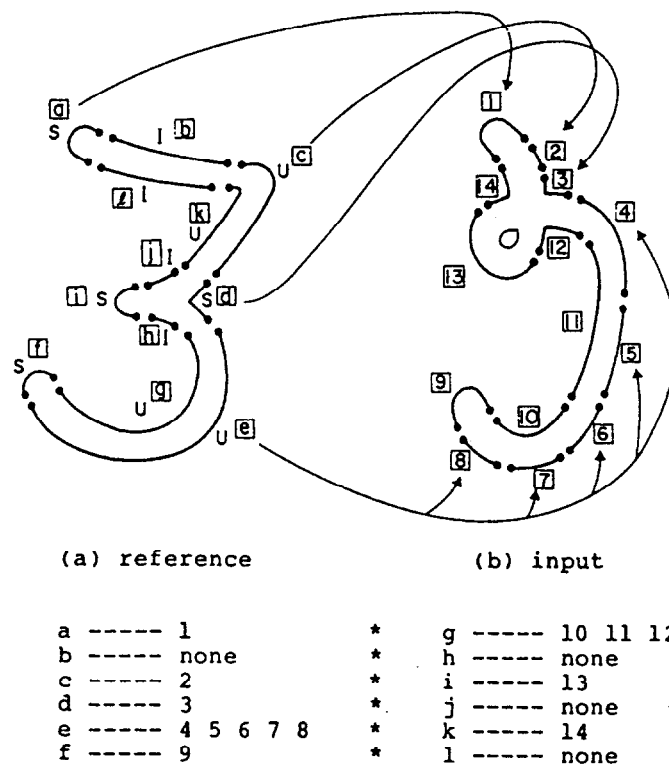


図 4.6: 文字パターン輪郭のセグメント化



(c) Correspondence between reference and input.

図 4.7: マッチング結果例、(a) 標準パターン、(b) 入力パターン、(c) 輪郭セグメント間の対応付け

これは標準パターン「1」だけでなく標準パターン「7」とのマッチングにも成功することは容易に推察できる。これらの区別をするものはカギの部分の長さと接線方向である。マッチング処理が終了しているので、図 4.8 におけるセグメント 1 を同定することは可能であり、セグメント 1 に付与された特徴量を閾値処理することで詳細識別ができる。この例の場合には、

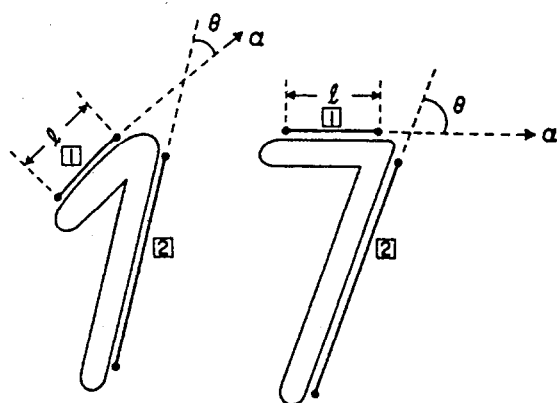


図 4.8: 類似文字の詳細識別、(a) 数字パターン「1」、(b) 数字パターン「7」

```

If  $\theta < T1$ , then 入力はカテゴリ「1」
else if  $l < T2$ , then 入力はカテゴリ「1」
else if  $\alpha < T3$ , then 入力はカテゴリ「7」
else
    リジェクト

```

のような閾値処理が考えられる。このような詳細識別を可能としているのは形状記述にローカルサポートがあるためである。

4.7 認識実験

自由手書き文字を対象として認識実験を行った。「自由手書き」とは、筆記者に字形の制限を課さないで自由に筆記させた字形の種類をいう。文字の種類は数字 (10 カテゴリ)、英字 (26 カテゴリ)、カタカナ (46 カテゴリ) の 3 種類である。文字パターンは同一のカテゴリに含まれる場合でも、連結成分の数や穴の数が異なっていたり、字形が甚だしく異なる場合 (例えば、カギつきの「1」と一本線の「|」) には、標準パターンを別々に用意した方が効率的である。表 4.1 に実験で用いた各文字種ごとのカテゴリ数と標準パターン数を示す。

数字を対象とした認識実験の結果を表 4.2 に示す。標準パターン作成は人間の視察によって行ったが、その時の検証に用いたデータセット (学習) 12500 パターンでの認識率は 98.9%、誤読率は

表 4.1: 認識辞書の構成

文字種	カテゴリ数	標準パターン数
数字	10	38
英字	26	38
カタカナ	46	106

0.2%であった。一方、未知のデータセット（テスト）9800 パターンに対しては認識率は98.1%、誤読率は0.2%であった。一般に、学習パターンに対する認識精度はテストパターンに対する認識精度よりも良いが、その差は、それぞれのパターンの数を増やせば増やすほど小さくなることが知られている。この実験の例ではその差が十分に小さく、実験に用いた未知のデータ以外の未知パターンに対しても同程度の精度が得られることが主張できる。また、誤読率はいずれの場合にも0.2%と非常に低い値が得られている。

表 4.2: 数字データに対する認識結果

データ	認識率	コンフリクト率	リジェクト率	誤読率	入力パターン数
学習セット	99.8%	0%	0.9%	0.2%	12,500
テストセット	98.1%	0.1%	1.6%	0.2%	9,800

英字とカタカナについての認識実験の結果を表 4.3 に示す。この場合の認識率はそれほど高くなく、実用に供するのは困難であるが、誤読率が低いため、他の認識方式と併用することにより、誤読の少ない認識システムを構築できることを示している。商用の文字入力システム（OCR）に対しては誤読が少ないことが強い要請となっている。

表 4.3: 英字とカタカナに対する認識結果

データ	認識率	コンフリクト率	リジェクト率	誤読率	入力パターン数
英字	95.5%	0%	3.9%	0.2%	2,600
カタカナ	96.8%	0.2%	2.4%	0.6%	4,600

4.8 まとめ

輪郭セグメント化に基づいた手書き文字認識方式を用いて認識実験を行った結果、

1. 数字、英字、カタカナについて高い認識精度が得られた。このことから、輪郭セグメント化による文字パターンの記述方式が自由手書き文字認識において有用であることが分かった。
2. 自由手書き文字認識においては、文字の大きな変形を許容できるフレキシブルな対応づけのための核となる、端点や凸部、凹部などの特徴部分の安定な検出が必要であるが、文字パターンのような2次元図形に対しても、輪郭の主要な曲率の変化によってそれらを安定に検出することができることが分かった。
3. マッチング後の詳細識別を可能とするローカルサポートのある形状特徴抽出方式を提供できることが分かった。

第5章 高速図形処理アーキテクチャ

本章では図形処理の基本演算を高速に実行するためのハードウェア・アーキテクチャについて述べる。ここには前処理としての濃淡画像処理と、二値図形に対する二値画像処理を含む。画像処理ハードの研究の過程で明らかとなった画像処理アーキテクチャの問題をまとめた上で、二次元図形処理に適した局所並列型の画像処理アーキテクチャと、輪郭追跡のためのハードウェアを提案する。

5.1 デジタル画像処理の問題点

画像処理に初めてコンピュータが利用されたのは文字認識のシミュレーションであった。この時にもフォン・ノイマン型と呼ばれる逐次型コンピュータの二次元データ処理の不得意さが指摘されてはいたが、実験用であったので、逐次型コンピュータの処理速度に対する問題は深く追求されなかった。その後、画像処理研究が進むにつれ、処理速度が問題とされるようになり、その費用効果比を向上させる努力がなされるようになった。本節では、デジタル画像処理を逐次型計算機で行う場合の問題点をいくつか挙げ、画像処理ハードウェアの設計に際して、これらの問題点をどのように解決すればよいかを述べる。

1. データの膨大さ

画像は二次元配列されたデータであり、画素の数は解像度の二乗のオーダーで増加するため、一般に大容量となってしまう。処理を逐次的に行うと処理時間は画素数に比例する。従って、簡単な処理・演算でも画像全体に行ったり、大量の画像に対して行う場合には非常に多くの時間を必要とする。このことが画像処理アルゴリズムの開発や、実用機器に応用する場合の最大のネックとなっている。

2. 無駄なデータ転送

画像はデータ量が多いため、それを転送するだけでも多くの時間を必要とする。実際のところ

ろ、画像処理は入力画像の1画素当たりの処理の量は多くない。ひとつの画像演算を行って、その結果に別の演算を加えるというのが基本であるが、近傍処理が入ると、画像全体にひとつの処理を施し、その結果を、一旦、記憶して次の処理の入力とする必要があり、1画素に対して継続して処理を加えるということはまれである。従って、キャッシングのような高速化手段は有効でなくなり、画像メモリと演算器の間を大量の画像データが行き来するという、無駄なデータ転送が存在する。

3. 処理の逐次性

画像処理の過程を分解すると、

- (a) 処理すべき画素の画像内のアドレスを計算する
- (b) そのアドレスが予め定められた処理ウインドウ内に存在するかどうかをチェックする
- (c) ウインドウ内にあれば、その画素データを転送する
- (d) 転送された画像データに演算処理を施す
- (e) 出力アドレスを計算する
- (f) そのアドレスが処理ウインドウ内に存在するかどうかをチェックする
- (g) 演算結果を転送する

のようになる。逐次型計算機では、これらの機能を文字通り逐次的にしか実行できないため、その分だけ時間がかかる。ハードウェアの設計に際しては、これらの機能を並列化すべきである。

4. 近傍処理の問題

画像処理では近傍処理、すなわち、ある画素を中心とした、局所領域内の複数の画素を使用する演算が多く用いられる。近傍演算の例としては、一次微分（x微分、y微分）、二次微分（ラプラシアン）、相関演算などの線形演算の他に、近傍を考慮した二値化方式がある。また、細線化、雑音除去などの論理演算もある。前者はTVカメラやスキャナで撮像して選られた濃淡画像の中からエッジを抽出して物体の形状を取り出す場合などに用いられる。後者は二値図形に対する前処理である。近傍の大きさは 3×3 、 5×5 、 7×7 などが多く用いられているが、大きいものではDOG（Difference of Gaussian）フィルタのように 32×32 程度まである。逐次的な計算では、これらの近傍のアクセスは1近傍点ごとに逐次的に行われ、アドレス計算やデータ転送はその都度実行されることになる。

5.2 デジタル画像処理高速化の流れ

前節で述べた画像処理の問題を解決すべく、多くの試みがなされたが、そのアーキテクチャを分類すると以下のようになる。

1. アレイプロセッサの利用

アレイプロセッサはベクトルや行列の演算を高速に実行するためのアーキテクチャであり、画像処理の特殊な分野で利用されている。本質的には1次元信号の高速処理が特長であるため、CTスキャナにおけるRadon変換や、フーリエ変換などのSeparableな処理に向いている。

2. 完全並列型画像処理プロセッサ

完全並列型画像処理プロセッサは、画像の大きさに相当するだけの数の単位プロセッサ（P E : Processor Element）を持ち、各々のP Eが画素毎に配置されているSIMDマシンである。P Eは画像データを記憶するためのレジスタやメモリ、演算器などを保有しており、近傍の他のP Eと互いに接続されている。実装上の観点から見れば、画素の数だけの単位プロセッサが必要なため画像の大きさに制限のあること、P E間の接続はせいぜい8近傍に限られること、多くの演算機能を実現するためにはP Eの構成が複雑になり、ひとつのLSIとして実現しようとするとは現在のLSI技術でも困難である、などの問題がある。しかし、処理は完全に並列的に行われるので速度は非常に早い。このようなアーキテクチャはUnger[Unger 59]によって初めて提案され、以降、CLIPシリーズ[Duff 76]など、いくつかの装置が開発されている。

3. 局所並列型画像処理プロセッサ

局所並列型プロセッサは、ある適当な大きさの局所領域内の画像データに対して並列アクセスし、そのデータを並列処理する専用回路を有し、その局所処理単位を走査して画像全体の処理を行うという機構を持っている。その概念図を図5.1に示す。また、近傍領域を並列的にアクセスする標準的な回路を図5.2に示す。

局所並列型プロセッサでは回路の規模と処理速度とのトレードオフを考慮して、ある程度の回路量によって大きな画像でも比較的高速に処理ができる。また、画像のバッファメモリにも低速のものが使用できるため、費用効果比が高く、現在の画像専用プロセッサの主流となっている。

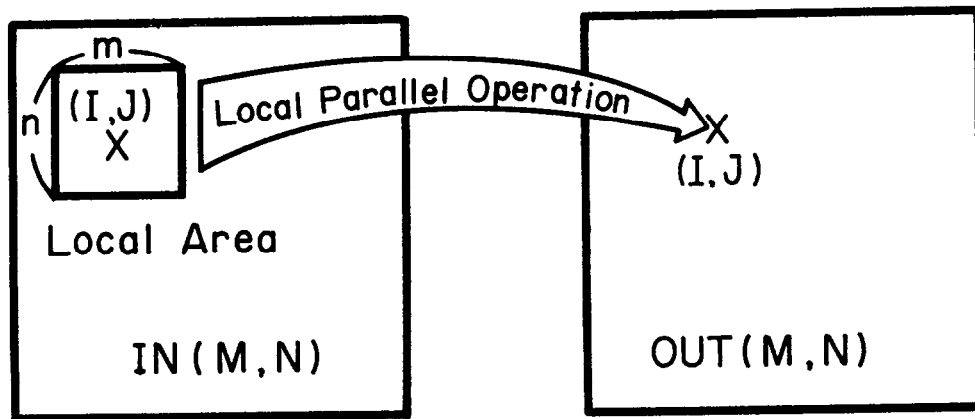


図 5.1: 局所並列処理の概念図

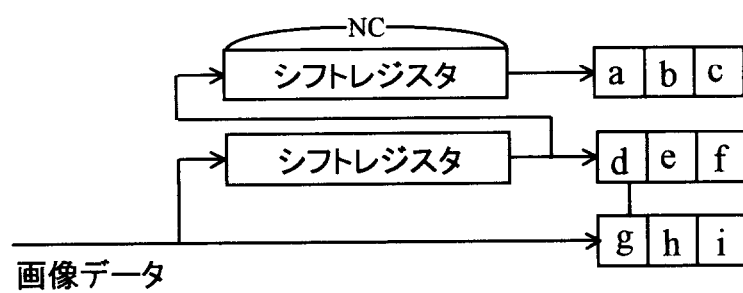


図 5.2: 近傍領域の並列アクセス回路の典型例

5.3 画像処理ハードウェアの設計問題

画像処理ハードウェアの設計に際して考慮すべき点はいくつかある。その中で重要な項目について述べる。

5.3.1 画像処理機能の選択

画像処理ハードウェアの持つべき機能について述べる。専用装置の場合はその目的がはっきりしているため、ソフトウェアとハードウェアとの作業分担を決めるだけでよいが、若干の汎用性を持たせる場合には機能を選択する必要がある。その場合、種々の画像処理アルゴリズムでよく使用され、しかも時間がかかる処理を回路化して、高速化を図ることが得策である。以下において、多く使用されている画像処理の機能と、そのハードウェア化を議論する。以降、 F は入力画像、 G は出力画像を表わす。

積和演算

$$G(x,y) = \sum_{(i,j) \in \Omega(x,y)} w_{ij} F(x+i, y+j)$$

$\Omega(x,y)$ は (x,y) の近傍

積和演算は画像 $F(x,y)$ の近傍に対してある決められた荷重係数 w_{ij} との内積を計算し、それを画像全体について行う線形演算である。荷重係数の値により、平均化、微分、強調などの空間フィルタリングや、標準パターンとのマッチング、相関演算などに広く用いられる。近傍の大きさとしては 3×3 から 32×32 程度までが考えられるが、画像データのアクセスや積和計算の量はその大きさに比例し、非常に時間のかかる演算であり、高速化が最も望まれているもののひとつである。ハードウェアとしては、並列アクセス回路の後で、近傍画素だけの乗算器を用意し、乗算結果を加算器のカスケードで加算すれば局所的には完全並列的に動作し、非常に高速になるが、近傍の大きさが大きくなれば乗算器の個数が増え、コスト的に問題がある。このトレードオフを考えた、時分割制御方式とパイプライン制御方式を組み合わせたアーキテクチャを後で提案する。また、積和演算自体はアレイプロセッサの得意とする演算であるが、2次元にした場合、データの並べ替えが必要であったり、同じデータを何度も重複して転送する必要があるなどの理由から、

アレイプロセッサを使用することは得策ではない。

論理フィルタ

$$G(x, y) = \text{Logic}(\{F_{ij} \mid (i, j) \in \Omega(x, y)\})$$

二値画像に対する処理で、近傍内の画素間の論理演算を行う。細線化、シュリンキング、雑音除去、特徴点抽出、境界線抽出、線素方向など図形認識の前処理として頻繁に用いられる。上式で *Logic* は近傍の画素数だけの引数を持つ論理演算である。ハードウェアは、近傍の大きさが 3 x 3 程度であれば、近傍パターンをアドレスとしたテーブル参照方式で任意の論理フィルタが容易に実現できるが、それより大きくなると近傍の並列アクセス回路の後に論理演算回路を固定的に作る必要がある。

領域番号付け

これは二値画像処理のひとつで、連結した領域に番号を付加し、各領域を分離する演算である。文字図形の各ストロークを分離するなどの基本演算である。連結性には左右上下のみが連結しているとみなす 4 連結と、斜め方向にも連結しているとする 8 連結とがある。この演算のアルゴリズムは何種類も考えられているが、ソフトウェアで実行するものが殆どである。ハードウェア化に対しては 1 画素を 1 度だけアクセスするだけ、つまり、画像を 1 回だけスキャンするだけで演算できることが望ましい。

座標変換

$$G(x_2, y_2) = F(x_1, y_1)$$

$$\text{where } x_2 = f_x(x_1, y_1), y_2 = f_y(x_1, y_1)$$

画像の拡大、縮小、回転などの幾何学的変換を行う演算である。通常、出力画像のアドレス (x_2, y_2) を独立変数として、座標変換関数の逆関数 f_x^{-1} 、 f_y^{-1} によって入力画像のアドレスを計算する。

f が線形であればアフィン変換と呼ぶ。ハードウェア構成は座標変換関数の形が決まれば、それに適した回路を作れるが、線形の場合には任意のアフィン変換が行える回路構成が望ましい。また、これをデータ転送、プログラム制御などと並列的に動作できるようにすることが高速化に役立つ。

線素追跡

これまでに述べた演算は局所領域内の処理を画像全体にわたって均一に施すタイプのもので、「局所並列」演算と呼ばれている。一方、ここで述べる線素追跡は図形の輪郭部分を追跡したり、二値線画を追跡するような演算であり、局所領域内の処理が終了して初めて次の局所領域が定まるというタイプである。したがって、ラスタ操作のときのように、演算処理とアドレス計算を並列化して高速化を図ることはできない。画像の大きさに比例した処理時間ではなく、追跡した画素の数に比例した処理時間となる。

以上述べた機能は画像処理の基本演算とでもいうべきものであり、これらを複数個、組み合わせることにより多くの画像処理機能を実現することができる。

5.3.2 高速化の諸方策

画像処理プロセッサの高速化に対する方式をまとめる。

1. 画像データ転送の無駄を省くシステム構成

画像メモリを中心に置いた構成が望ましい

2. 画像処理機能の並列演算化

画像データのアクセスのためのアドレス計算と、画像データの転送、画像データに対する演算処理を並列的に行うことで高速化できる。

3. 画像の並列アクセス

画像は2次元配列であり、2次元性を保存したまま、近傍のアクセスを行う必要があるが、それらのデータを逐次的にアクセスすることは低速化を招く。ハードウェア的に並列アクセスできることが望ましい。

4. パイプライン制御、時分割制御の採用

パイプライン制御は単位時間内に単一の演算を行う演算器を直列に並べたもので、パイプラインの段数に比例した時間遅れだけで出力が得られる。データ量が多いほど効率が良いので

画像処理には向いている。また、時分割制御では、データの発生速度（入力装置の速度、メモリの読み出し速度）と演算速度との差をうまく使えば、効率のよい処理ができる。

5. LSI化

データ転送のバンド幅を考慮して、機能単位にモジュール化することにより、LSI化することはコストの点から見ても有利である。

5.4 高速図形処理ハードウェア

前節に述べた図形処理高速化の方策に基づいて、並列画像演算装置（PPP：Parallel Pattern Processor）を開発した。図 5.3 にその構成図を示す。PPPはホストコンピュータの補助装置であり、その主な構成要素は、プログラム制御部、アドレス制御部、8ビット画像データバス、テーブルメモリ（512バイト）、および、複数の並列演算回路である。また、画像データは外部の画像メモリより供給され、画像メモリとのインタフェースも含まれている。マイクロプログラム制御部は、ホストコンピュータと画像メモリとの通信、データ転送、データバスの制御、各演算部のモード選択など、PPP全体の制御を行うほかに、ハードウェア化されていない画像処理演算をプログラムの的に実行する。アドレス制御部では2次元の座標計算が高速に行われる。テーブルメモリは各種の情報格納のためのものである。各演算部はパイプライン処理や並列計算により高速化されているが、PPP全体としても入力画像データのアドレス計算、入力画像データの転送、各種演算処理、出力画像データのアドレス計算、および出力画像データの転送が並列に実行できるようになっており、高速化を図っている。以下において、各演算部について説明する。

5.4.1 空間フィルタ演算部（Spatial Filter）

空間フィルタの処理内容は二次元コンボリューションであり、これは典型的な近傍演算である。乗算、加算、データアクセスの回数が多く、処理時間が非常にかかる演算である。PPPでは図 5.4 に示す回路で高速化と回路の簡単化を図っている。 $n \times n$ のコンボリューションでは、ひとつの演算を得るのに n^2 回の乗算と加算が必要であり、この積和演算を全て並列に行うには n^2 個の乗算器と $n^2 - 1$ 個の加算器が必要となる。 $n = 3$ 程度であれば実現可能な範囲であるが、 $n = 8$ ともなると、これは膨大な回路量となる。PPPでは 8×8 のコンボリューションが実行できるようになっている。

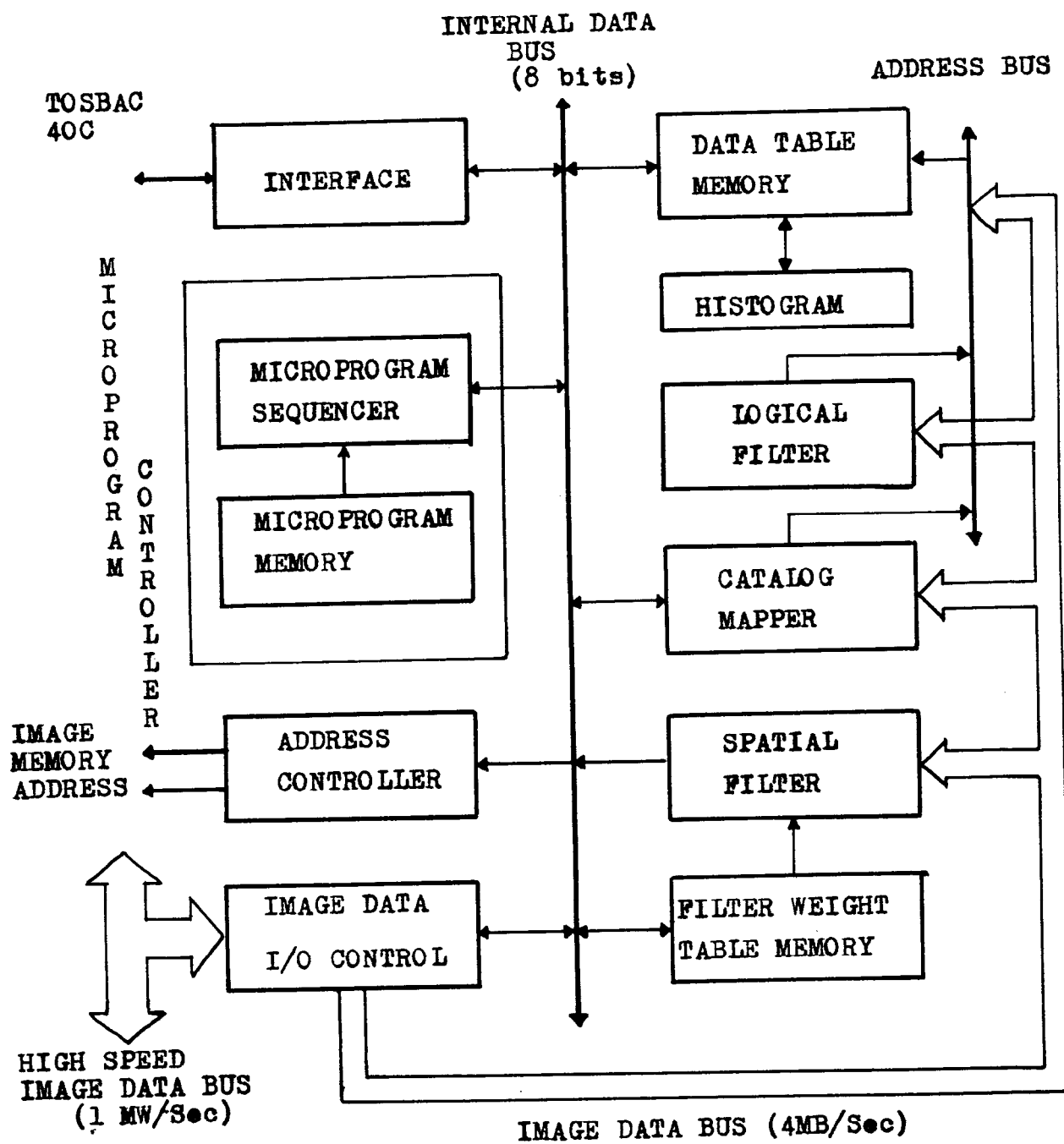


図 5.3: PPPの構成図

ここで考案した効率的な演算回路を説明する。1 x 8 の局所領域をラインバッファにより並列アクセスし、入力画像データ 8 画素分、 $F_{ij}|j = 1, 8$ 、を得る。8 x 8 の荷重テーブルからは $W_{kj}|j = 1, 8$ の 8 個の重み係数を読み出し、8 個の乗算器と 7 個の加算器によって並列的に $\sum_{j=1}^8 W_{kj} F_{ij}$ を計算する。荷重テーブルを画像のラインバッファの 8 倍の速度でシフトすると、順次 $\sum_{j=1}^8 W_{k,j} F_{ij}, k = i, i+7$ が計算され、これは $k-1$ 画素左の $F_{i-k+1,j}|j = 1, 8$ が必要とする部分積和となっている。これらを図 5.5 に示すようにシフトレジスタを荷重テーブルと同じ速度でシフトしながら、そこにアキュムレートする。シフトレジスタが一巡したときの出力が、最終結果、 $\sum_{k=1}^8 \sum_{j=1}^8 W_{i-k+1,j} F_{ij}$ を k 画素遅れで得られる。この計算方式は、画像データの入力速度が遅いことを利用して、その 8 倍で並列回路を動かすことで回路の簡単化を図ったものである。

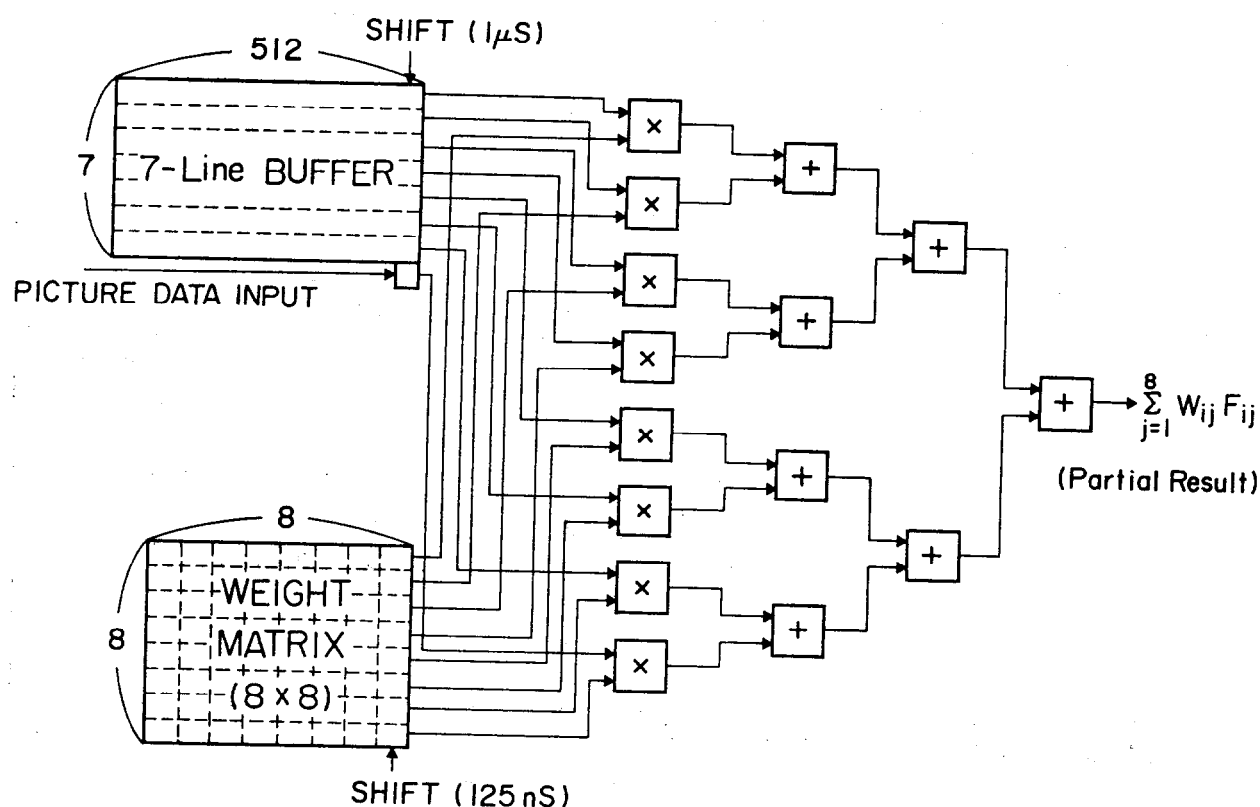


図 5.4: 空間フィルタ演算回路の構成図

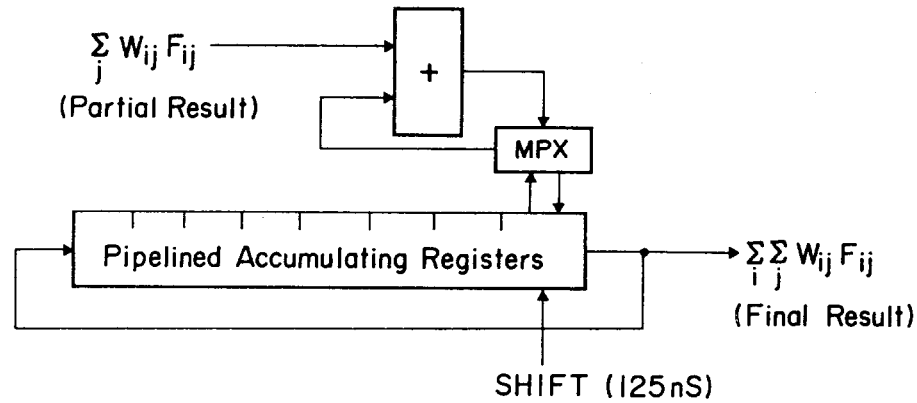


図 5.5: 空間フィルタ演算回路の最終段

5.4.2 論理フィルタ演算部 (Logical Filter)

3 x 3の近傍マスクを9ビットのアドレスとしてScratch Pad Memoryの内容を出力する。図5.6に構成図を示す。3 x 3の近傍マスクは2行のラインバッファと3行3段のシフトレジスタで構成し、1から9までの番号のついた各ビットがScratch Pad Memoryのアドレスとして供給される。Scratch Pad Memoryの内容を入れ替えるだけで任意の論理フィルタ演算を実行することができる。

5.4.3 領域番号付け演算部 (Catalog Mapper)

図5.7に構成図を示す。入力画像をラスタ走査する中で、A～EおよびSで作られる2 x 3のマスクで連結性の判定が行われる。入力画像の1画素はレジスタEに入力され、次の画素を読み込むタイミングでレジスタSにシフトされる。レジスタAおよびB、Cは一行前の画像に対して付与された領域番号が格納されている。これらのレジスタもシフトレジスタになっており、画素が読み込まれるタイミングで左にシフトされる。レジスタDは直前の画素に付られた領域番号が格納されており、この出力は画像の横幅に対応した長さのシフトレジスタを通じてレジスタCに結合されている。

画素Sの連結性はA～Eの5つのレジスタの内容で判定される。各レジスタの値が0でない場合には「1」とし、0の場合は「0」として5ビットのアドレスに変換し、Label Selectionメモリに供給される。Label Selectionメモリの内容は図5.8に示した通りであるが、このテーブルに

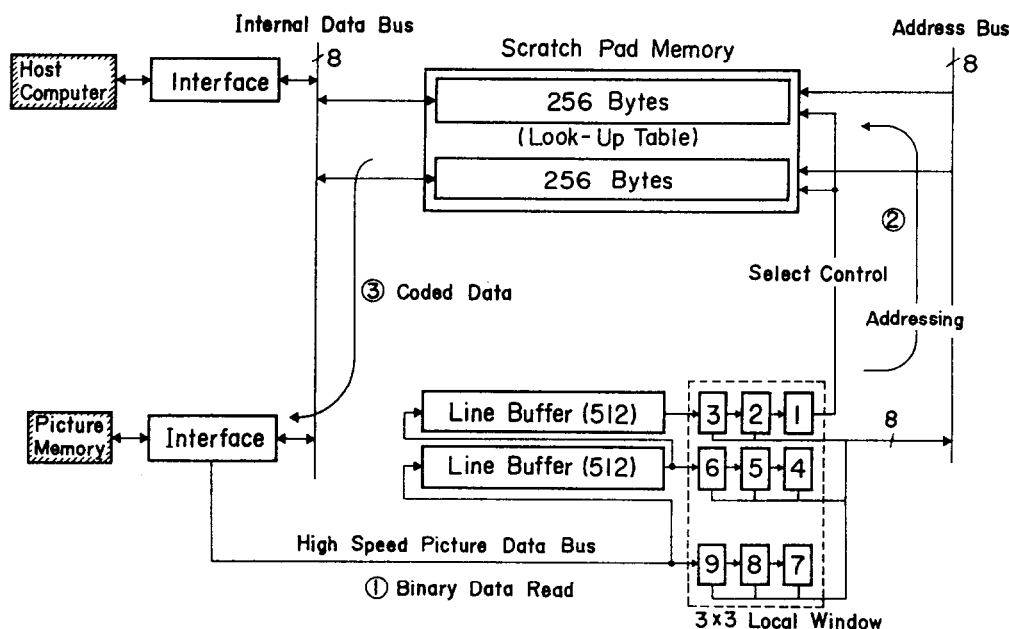


図 5.6: 論理フィルタ演算回路の構成図

は $S = 1$ のとき、 $A \sim E$ までの 2×3 マスクの形状により、 S に対して付与すべき領域番号を決定するとともに、既に付与されている領域番号を書き換える必要があるかどうかを決定する。付与する領域番号を与えるレジスタの記号が Y の欄に書かれてあり、 X の欄には書き換えが必要な場合に対象となるレジスタ記号が書かれている。ここで N は New No. Counter を示す。

4 連結の場合を例に取ってこれらの意味を説明する。 $A = B = C = 0$ の場合、 $D = 1$ であれば S には D と同じ領域番号をふるのが適当であるので、アドレス = 0001^* ($*$ は don't care を示す) の Y の欄には D と書かれている。また、 $D = 0$ の場合は S に新しい領域番号を付与するのが適当であるので、アドレス = 0000^* の Y の欄には N が書かれている。

既に付与された領域番号を変更する必要があることは、例えば、アドレス = 00111 に示されている。

つぎに、 $A = B = 0$ 、 $C = D = E = 1$ の場合を考える。このとき、 S および D 、 E 、 C は連結しているので、 S には D もしくは C の領域番号を付けるのが適当である。しかし、 D には C とは別の領域番号がつけられているはずである。何故なら、1 画素前の処理の段階では $B = 0$ であるため、 D と C とが連結しているという情報はなかったからである。この段階になって初めて D と

Cとの連結が発見されたのである。したがって、既に付けられたDの領域番号をCの領域番号に書き換える必要がある。しかし、Dの1画素前が“= 1”のときはDと同じ領域番号が付けられているはずで、それを書き換えることはできない。そこで、Dの領域番号がCの領域番号と同一の連結成分を表していることを記憶することで対処する。この情報はXをアドレス、Yを内容として Scratch Pad Memory に記憶される。この書き込み制御を行っているのが Table Control 部である。

8連結の場合は領域番号の付け方と書き換えが必要なパターンが4連結の場合とは異なるのでメモリの内容は4連結の場合とは別に持つ必要がある。この選択はプログラムの制御する。また、NNの欄はNew No. Counterのカウンタアップを指示する信号となる。画像全体の走査が終了した段階で Scratch Pad Memory を編集して真の領域番号に変換する。この演算回路は1度のラスタ走査で領域の連結性を調べた上で番号を付けることのできる構成になっているのが特徴である。

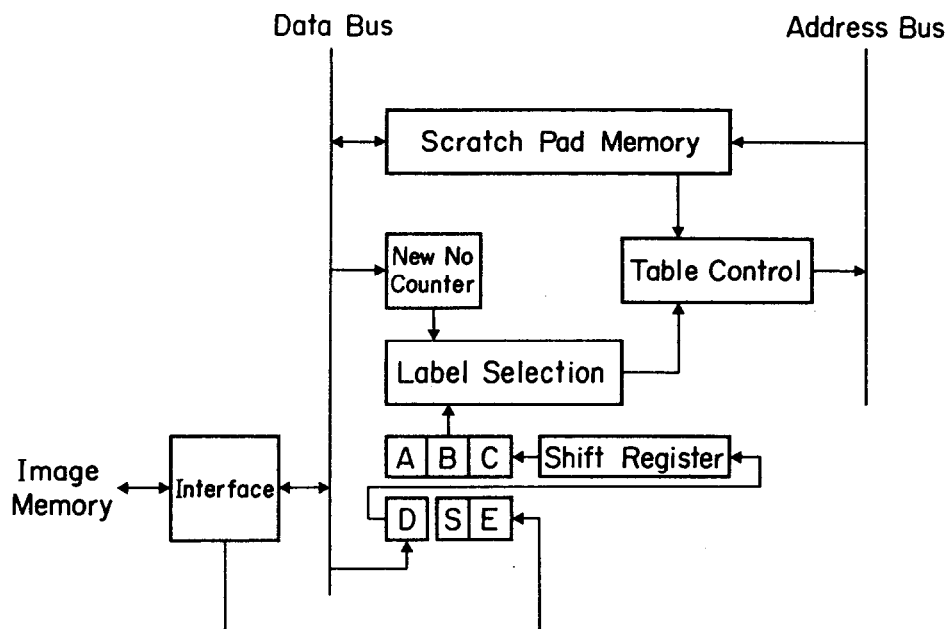


図 5.7: 領域番号付け演算回路の構成図

(S=1)					4-Connectivity			8-Connectivity		
A	B	C	D	E	X	Y	N.N.	X	Y	N.N.
0	0	0	0	0	Z	N	U	Z	N	U
0	0	0	0	1	Z	N	U	Z	N	U
0	0	0	1	0	Z	D	*	Z	D	*
0	0	0	1	1	Z	D	*	Z	D	*
0	0	1	0	0	Z	N	U	Z	C	*
0	0	1	0	1	Z	C	*	Z	C	*
0	0	1	1	0	Z	D	*	D	C	*
0	0	1	1	1	D	C	*	D	C	*
0	1	0	0	0	Z	B	*	Z	B	*
0	1	0	0	1	Z	B	*	Z	B	*
0	1	0	1	0	Z	B	*	Z	B	*
0	1	0	1	1	Z	B	*	Z	B	*
0	1	1	0	0	Z	C	*	Z	C	*
0	1	1	0	1	Z	C	*	Z	C	*
0	1	1	1	0	Z	C	*	Z	C	*
0	1	1	1	1	Z	C	*	Z	C	*
1	0	0	0	0	Z	N	U	Z	A	*
1	0	0	0	1	Z	N	U	Z	A	*
1	0	0	1	0	Z	D	*	Z	D	*
1	0	0	1	1	Z	D	*	Z	D	*
1	0	1	0	0	Z	N	U	A	C	*
1	0	1	0	1	Z	C	*	A	C	*
1	0	1	1	0	Z	D	*	D	C	*
1	0	1	1	1	D	C	*	D	C	*
1	1	0	0	0	Z	B	*	Z	B	*
1	1	0	0	1	Z	B	*	Z	B	*
1	1	0	1	0	Z	B	*	Z	B	*
1	1	0	1	1	Z	B	*	Z	B	*
1	1	1	0	0	Z	C	*	Z	C	*
1	1	1	0	1	Z	C	*	Z	C	*
1	1	1	1	0	Z	C	*	Z	C	*
1	1	1	1	1	Z	C	*	Z	C	*

("0" means all 0 in 8 bits)
 ("1" means non-zero)

Z: Zero
 U: Count up New No. Counter
 *: None

図 5.8: 領域番号付け回路の制御メモリの内容

5.4.4 輪郭追跡回路

二値図形の輪郭を右回りに追跡する回路であるが、これは局所並列処理ではないのでPPPには含まれていない。画像の 2×2 の近傍パターンを4ビットで表し、これを1アドレスでアクセスできる近傍アクセスメモリを用いる。近傍アクセスメモリへの書き込みの回路を5.9に示す。輪郭追跡の始点としてX、Yアドレスを指定して近傍アドレスメモリを読み出し、その出力によって両アドレスレジスタを制御し輪郭を追跡するとともに、8方向コードを出力する回路を5.10に示す。図中の制御テーブルは近傍アクセスメモリの出力に前回出力された方向コードのMSB (P)を加えた5ビットでアドレッシングされるが、その内容を表5.1に示す。表において、Xの欄はXアドレスをインクリメント(+1)、デクリメント(-1)あるいは変化させない(0)の制御を示しており、Yの欄も同様である。Zの欄は方向コードが有効かどうかを示している。 2×2 の近傍パターンを見るだけで追跡できる点が特徴である。

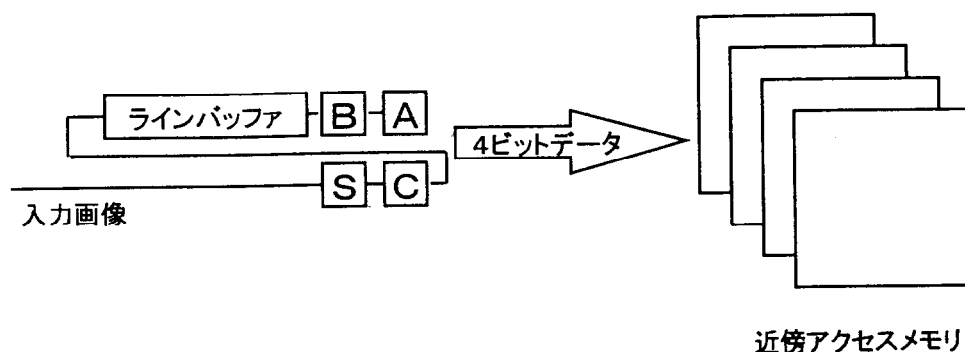


図 5.9: 近傍アクセスメモリの構成と書き込み回路

5.5 高速図形処理装置の処理速度

図5.11に示すように画像メモリを中心としたシステムにおける並列画像処理演算 (PPP) の速度を表5.2に示す。画像データの読み出し、1画素当たりの画像処理演算、出力画像の書き込みが同期して行われるので、画像データの読み出し時間以内に処理が完了するが、この値を示している。実際に製作した装置では、画像データの読み出し時間が画素当たり $1 \mu \text{ sec}$ であった。

また、輪郭追跡の速度は、方向コードを出力しない 2×2 マスクのパターンが約半数存在するので、輪郭の長さの2倍にほぼ比例する。

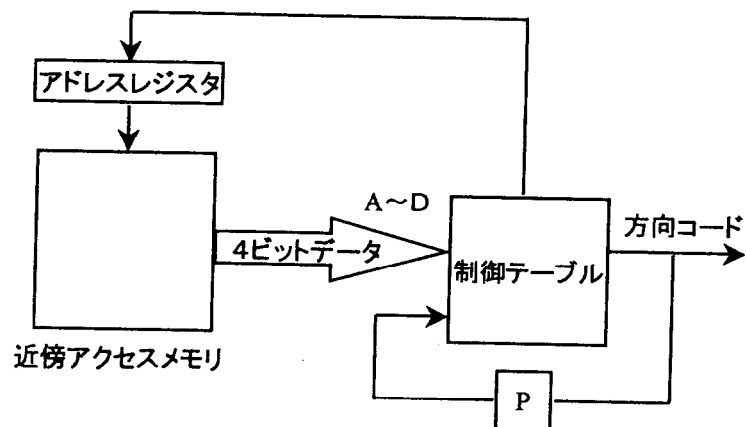


図 5.10: 近傍アクセスメモリを用いた輪郭追跡回路

表 5.1: 輪郭追跡制御テーブルの内容 (*は0または1)

A	B	C	D	P	X	Y	Z	方向コード
0	0	0	0	*	0	0	0	-
0	0	0	1	*	-1	0	0	-
0	0	1	0	*	0	-1	0	-
0	0	1	1	*	-1	0	1	6
0	1	0	0	*	0	1	0	-
0	1	0	1	*	0	1	1	0
0	1	1	0	0	0	1	0	-
0	1	1	0	1	0	-1	0	-
0	1	1	1	*	0	1	1	7
1	0	0	0	*	1	0	0	-
1	0	0	1	0	1	0	0	-
1	0	0	1	1	-1	0	0	-
1	0	1	0	*	0	-1	1	4
1	0	1	1	*	-1	-1	1	5
1	1	0	0	*	1	0	1	2
1	1	0	1	*	1	1	1	1
1	1	1	0	*	1	1	1	3
1	1	1	1	*	0	0	0	-

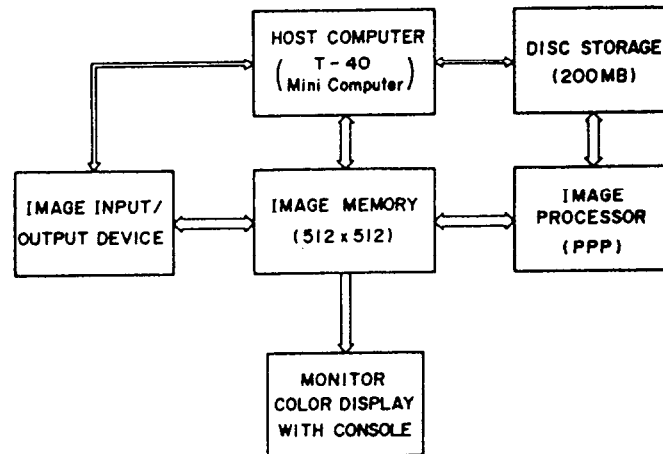


図 5.11: 画像処理システム

表 5.2: P P P の処理速度 (画像の大きさは 512x512、画像データの読み出し時間が $1\mu s$ のとき)

基本演算	画像処理	処理時間 (ms)
空間フィルタ	1 次微分	262
	2 次微分	262
	平滑化	262
論理フィルタ	細線化	262x 線の最大径
	輪郭の検出	262
領域番号付け	図形分離	262
	粒度分布計測	262
座標変換	移動・回転・拡大	262
	アフィン変換	262

5.6 まとめ

二次元画像の高速処理アーキテクチャについて述べた。画像処理は近傍演算が基本であり、データ量も多いため、近傍演算を専用回路化し、さらに、データ入出力との並列処理、パイプライン処理などの高速化手法を適用することが有効である。二次元図形の記述と認識のための前処理には局所並列性を持つものが多く、空間フィルタ、論理フィルタ、領域番号付けについて専用回路を新たに考案し、ハードウェアを製作した。これらは画像データの入力に同期して、その時間内に処理を終えることができる。また、輪郭追跡を効率的に実行する専用回路を示した。これは 2×2 の近傍パターンのみを観測して、輪郭を追跡することができ、輪郭長にほぼ比例した時間のみで処理が終了する。

第6章 結論

本研究では、図形や物体の認識を目的として、そのために有用な形状記述のあり方について考察し、新しい方法を提案した。この方法は、図形記述に対して要請されている、ひとつの図形内での部分図形の階層表現、クラスを超えた階層表現、および、局部の変形があっても、全体の表現が大きく変わることなく、しかも、局部の変形そのものにはセンシティブであるというローカルサポートの性質を持ったものである。

輪郭ベースと領域ベースの記述方式の各々について新しい方式を提案し、また、各々の特長を生かして、それぞれの記述をネットワークの形で表現してローカルサポートの性質を付け加え、さらに、状況に応じて必要な情報を取り出して、図形を認識できるだけの豊富な情報量をもつ、リッチな表現を実現した。

輪郭ベースの階層的記述方式は、曲率の変化の形態（つまり不連続性）を分類し、それらを輪郭のプリミティブとして定義し、これらのプリミティブを Scale Space の中で探索するという手法であり、画像理解における明るさの変化の記号的解釈を意味する Primal Sketch との類似性から、提案する方式を Curvature Primal Sketch と名づけた。輪郭の接線方向関数に対するガウシアン¹、2次微分フィルタの出力の挙動を各プリミティブのモデルについて明らかにし、Scale Space 内での探索アルゴリズムを構築した。この方式をいくつかの工具の図形に適用して、人間の感覚にマッチした階層的な記述が得られることを示した。また、このアルゴリズムを3次元曲面の主曲率曲線に適用して、その上での曲率の主要な変化点を検出し、それらを繋ぎ合わせることで、曲面上の曲率の不連続線の検出が可能であることを実験的に示した。

また、輪郭ベースの図形記述方式として、局所的に対称な関係にある二つの輪郭点の中点を滑らかに結んだ曲線（Smoothed Local Symmetries）によって領域記述を行う方法を示し、これを効率的に計算するアルゴリズムを提案した。この方法は Curvature Primal Sketch の結果を用いて輪郭のセグメント化を行い、これを直線・円弧近似することで解析的に求めるものである。いくつかの工具の図形に対してこのアルゴリズムを適用し、全体的な対称軸や部分的な対称軸が検出

されることを実験的に示した。

さらに、部分図形の検出に対して、これらの領域記述と輪郭記述と統合したネットワーク表現が有効であることを示した。ある形の部分ネットワークを探索することで部分図形を検出するための条件を二つのルールとして定式化した。さらに、これらの記述は Scale Space 解析に立脚しているため、階層表現とすることもでき、物体のクラスを超えた階層的セマンティックネットワーク表現に発展できることを示した。

このような記述方式を、数字、英字、カタカナの自由手書き文字の認識に応用し、高い認識精度が得られることをシミュレーションで示した。この結果、輪郭セグメント化による文字パターンの記述方式が自由手書き文字認識において実用的な有用性があることが分かった。特に、自由手書き文字認識においては、文字の大きな変形を許容できるフレキシブルな対応づけのための核となる、端点や凸部、凹部などの特徴部分の安定な検出が必要であるが、文字パターンのような二次元図形に対しても、輪郭の主要な曲率の変化によってそれらを安定に検出することができた。また、マッチング後の詳細識別を可能とするローカルサポートのある形状特徴抽出方式を提供することが分かった。

最後に、本論文で提案した図形記述方式を現実の画像データに適応し、実用的な処理速度を提供できる二次元画像の高速処理アーキテクチャについて述べた。画像処理は近傍演算が基本であり、データ量も多いため、近傍演算を専用回路化し、さらに、データ入出力との並列処理、パイプライン処理などの高速化手法を適用することが有効である。二次元図形の記述と認識のための前処理には局所並列性を持つものが多く、空間フィルタ、論理フィルタ、領域番号付けについて専用回路を新たに考案し、ハードウェアを製作した。これらは画像データの入力に同期して、その時間内に処理を終えることができる。また、輪郭追跡を効率的に実行する専用回路を示した。これは 2×2 の近傍パターンのみを観測して、輪郭を追跡することができ、輪郭長にほぼ比例した時間内で処理が終了するという高速性を持つ。これらのアーキテクチャはソフトウェア処理においても効率的な処理アーキテクチャとなる。

謝辞

本研究ならびに論文執筆にあたり、ご指導、ご鞭撻を賜りました、東京工業大学精密工学研究所の佐藤誠教授に心から感謝いたします。

本論文の内容のうち、第2章と第3章に関する部分は、筆者が米国マサチューセッツ工科大学人工知能研究所に客員研究員として滞在中に行った研究成果に基づいております。多くのアイデアは指導いただいた Dr. Mike Brady の Computer Vision への広い知識と深い洞察に負うところが多く、このような研究成果を得ることができたことに深く感謝いたします。同時に、MIT 人工知能研究所において指導・議論いただいた、Prof. Tomaso Poggio、Dr. Alan Yuille、Dr. Jean Ponce、Mr. Scot Heide、Mr. Mike Genert に感謝します。また、MIT への留学の機会を授けていただいた東芝総合研究所の永井淳 元所長、情報システム研究所の筑後道夫 元所長、森健一 元所長、渡辺貞一博士に深く感謝いたします。

第4章に述べた文字認識の研究は東芝総合研究所において実施したものであります。共同研究者の黒沢由明氏と指導者の坂井邦夫博士に深く感謝いたします。また、本研究のベースとなった種々の文字認識研究でご指導いただいた東京工業大学の飯島泰蔵名誉教授と小川英光教授に感謝申し上げます。前田賢一氏、中村好勝氏、須田正人氏には同僚として種々の議論をいただき、感謝いたします。東芝青梅工場の長谷川健治氏、西村一夫氏、田辺吉久氏、三宮昭夫氏にはこの技術の製品化にご尽力いただきました。厚くお礼申し上げます。

第5章に述べた図形処理アーキテクチャは通産省大型プロジェクト「パターン情報処理」の一環として、東芝総合研究所の森健一博士の卓越した将来ビジョンの下に開発した装置を基本としたものであります。このような研究の機会を与えていただき、ご指導いただいた森健一博士に深く感謝いたします。また、共同研究者である木戸出正継博士、篠田英範氏、田端光男氏に感謝するとともに、ハードウェアの開発に携わっていただいたアジアエレクトロニクス株式会社の横田氏に感謝いたします。

最後に、長年の間、企業での研究生活を支えてくれた妻の恵美子に最大限の感謝を捧げます。

付録：発表論文リスト

第2章に関連する論文

Asada, Haruo and Mike Brady, "The Curvature Primal Sketch," Proceedings of the IEEE Workshop on Computer Vision: Representation and Control, Annapolis(1984)

Asada, Haruo and Mike Brady, "The Curvature Primal Sketch," IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol.PAMI-8, No. 1, pp.2-14(1986)

Brady, Mike, Jean Ponce, Alan Yuille, and Haruo Asada, "Describing surfaces," Computer Vision, Graphics, and Image Processing, Vol.32, pp.1-28(1985)

第3章に関連する論文

Brady, Mike and Haruo Asada, "Smoothed Local Symmetries and their implementation," International Journal of Robotics Research, Vol.3, No.3, pp.36-61(1984)

第4章に関連する論文

Kurosawa, Yoshiaki and Haruo Asada, "Attributed String Matching with Statistical Constraints for Character Recognition," Proceedings of the 8th International Conference on Pattern Recognition. pp.1063-1067 (1986)

第5章に関連する論文

Mori, Ken-ichi, Masatsugu Kidode, Hidenori Shinoda, and Haruo Asada, "Design of local parallel pattern processor for image processing," AFIP Conference Proceedings Vol.47, pp.1025-1031(1978)

Asada, Haruo, Mitsuo Tabata, Masatsugu Kidode and Sadakazu Watanabe, "New Image Processing Hardware and its applications to industrial automation," SPIE vol.182, Imaging Application for Automated Industrial Inspection & Assembly, pp.14-21, 1979

麻田治男、篠田英範、田端光男、木戸出正継、渡辺貞一：会話型画像処理用高速ハードウェアの開発と評価、電子通信学会論文誌 '85/4 Vol.J68-D、No.4、pp861-868 (1985)

その他の発表論文

土井康弘、小松忠紀、麻田治男，“ホログラフィの光弾性への応用”，精密機械，37 巻 8 号，pp.600-606 (1971)

Mori, Ken-ichi, Masatsugu Kidode, Haruo Asada, "An Iterative Prediction and Correction Method for Automatic Stereo Comparison," Computer Graphics and Image Processing, vol.2, pp.393-401 (1973)

Takebayashi, Yoichi, Haruo Asada, Hidenori Shinoda, Tsuneo Nitta, Shoichi Hirai, and Sadakazu Watanabe, "Telephone Speech Recognition Using a Hybrid Method," Proceedings of the 7th International Joint Conference on Pattern Recognition, pp.1232-1235 (1984)

渡邊睦、小野口一則、久野義徳、麻田治男，“視差予測ステレオ法を用いた障害物検出”，電子情報通信学会論文誌 D-II，Vol.J73-D-II，No.6，pp.862-870 (1990)

Tsujimoto, Shuichi and Haruo Asada, "Understanding Multi-Articled Documents," Proceedings of the 10th International Conference on Pattern Recognition, pp. 551-556 (1990)

Tsujimoto, Shuichi and Haruo Asada, "Document Image Processing for Accurate and High Speed Text Reading," Proceedings of Syntactic & Structural Pattern Recognition, pp.501 (1990)

Tsujimoto, Shuichi and Haruo Asada, "Resolving Ambiguity in Segmenting Touching Characters," Proceedings of the 1st International Conference on Document Analysis and Recognition, pp.701-709 (1991)

Tsujimoto, Shuichi and Haruo Asada, "Major Components of a Complete Text Reading System," Proceedings of the IEEE, Vol.80, No.7, pp.1133-1149 (1992)

参考文献

- [Blum 78] Blum, H and Nagel, R. N., " Shape description using wighted symmetric axis features," *Pattern Recognition*, vol. 10, pp. 167-180, 1978.
- [Boissonat 82] Boissonat, J. D., " Stable matching between a hand structure and an object silhouette", *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. PAMI-4, pp. 603-611, 1982.
- [Bolles 82] Bolles, R. C., and R. A. Cain, " Recognizing and locating partially visible objects: The local-feature-focus method", *International Journal of Robotics Research*, vol. 1, No.3, pp. 57-82, 1982.
- [Brady 82] Brady, Michael, " Smoothed local symmetries and local frame propagation", *Proc. of Pattern Recognition and Image Processing* , pp. 629-633, Las Vegas, June, 1982.
- [Brooks 79] Brooks, R. A., " Goal-directed edge linking and ribbon finding", *Proc. of IU Workshop*, pp. 72-78, 1979.
- [Brooks 81] Brooks, R. A., " Symbolic Reasonig Among 3-D Models and 2-D Images", *Artificial Intelligence* , vol. 17,pp. 285-348, 1981.
- [Canny 83] Canny, J. F., " Finding lines and edges in images", *MIT AI-IR-720*, 1983.
- [Connell 85] Connell, J. H. and Brady, M., " Learning shape descriptions", *Proc. of 8th IJCAI* , Los Angels, 1985.
- [Duff 76] Duff, M. J. B., " CLIP4 : A large scale integrated circuit array parallel processor", *Proc. of 3 rd. IJ CPR*, pp. 728-733, November,1976.
- [Fu 74] Fu, K. S., " Syntactic Methods in Pattern Recognition", *Academic Press*, New York, 1974.
- [Iijima 73] Iijima, T., Genchi, H. and Mori, K., " A theory of character recognition by pattern matching method", *Proc. of 1st Intl. Joint Conference on Pattern Recognition*, November, 1973.

- [Kurosawa 86] Kurosawa, Y. and Asada, H., "Attributed string matching with statistical constraints", *Proc. of 8th ICPR*, pp. 1063-1067, Paris, 1986.
- [Marr 74] Marr, D., "The low-level symbolic representation of intensity changes in an image", *MIT AIM-325*, 1981.
- [Marr 76] Marr, D., "Early processing of visual information", *Philosophical Trans. Royal Society of London*, B, vol. 275, pp.483-524, 1976.
- [Marr 77] Marr, D., "Analysis of occluding contour", *Proceedings of Royal Society of London*, B, vol. 207, pp. 441-475, 1977.
- [名倉 76] 名倉, "手書き線図形を対象とした直線と円弧による近似法", 信学技報, IE79-72, pp. 37-44, 1976.
- [Nevatia&Binford 77] Nevatia, R., and Binford, T. O., "Description and recognition of curved objects", *Artificial Intelligence*, vol. 8, pp. 77-98, 1976.
- [Pavlidis 74] Pavlidis, T. and Horowitz, S. L., "Segmentation of plane curves", *IEEE Trans. on Computer*, vol. C-23, No. 8, pp. 860-870, 1974.
- [Perkins 78] Perkins, W. A., "A model-based vision system for industrial parts", *IEEE Trans. on Computer*, vol. C-27, No. 2, pp. 126-143, 1978.
- [Rosenfeld 76] Rosenfeld, A. and A. C. Kak, "A model-based vision system for industrial parts", *Digital Picture Processing*, Academic Press, 1976.
- [上坂 84] 上坂他, "平面曲線の線分と円弧による近似", 信学技報, PRL84-3, pp. 19-28, 1984.
- [Unger 59] Unger, S. H., "Pattern Detection and Recognition", *Proc. of IRE*, vol. 47, pp. 1737-1752, 1959.
- [Tsai 80a] Tsai, W. H. and Fu, K. S., "A syntactic-statistical approach to recognition of industrial objects", *Proc. of 6th ICPR*, pp. 251-259, Munchen, 1980.
- [Tsai 80b] Tsai, W. H. and Fu, K. S., "Attributed grammar - a tool for combining syntactic and statistical pattern recognition", *IEEE Trans. on SMC*, vol.SMC-10, No.12, 1980.
- [Tsai 85] Tsai, W. H. and Yu, S. S., "Attributed string matching with merging for shape recognition", *IEEE Trans. on PAMI*, vol.PAMI-7, No.4, 1985.
- [Winston 82] Winston, P. H. et al., "Learning physical descriptions from functional descriptions, example and precedents", *MIT AIM-679*, 1982.

- [Witkin 83] Witkin, A., “ Scale-space filtering”, *Proc. of 7th IJCAI* , pp. 1019-1021, Karlsruhe, 1983.
- [You 79] You, K. C. and Fu, K. S., “A syntactic approach to shape recognition using attributed grammars”, *IEEE Trans. on SMC*, vol.SMC-9, 1979.
- [Yuille 83a] Yuille, A. L. and Poggio, T., “ Fingerprints Theorems for zero-crossings”, *MIT AIM-730*, 1983.
- [Yuille 83b] Yuille, A. L. and Poggio, T., “ Scaling theorems for zero-crossings”, *MIT AIM-722*, 1983.