The University of Osaka
Institutional Knowledge Archive

Title	画像理解におけるモデル獲得に関する研究
Author(s)	上田, 修功
Citation	大阪大学, 1992, 博士論文
Version Type	VoR
URL	https://doi.org/10.11501/3087994
rights	
Note	

The University of Osaka Institutional Knowledge Archive : OUKA

https://ir.library.osaka-u.ac.jp/

The University of Osaka

画像理解におけるモデル獲得に関する研究

1991年

上田修功

①

画像理解におけるモデル獲得に関する研究

上田修功

序文

本論文は、筆者が行なってきた、画像理解におけるモデル獲得に関する研究をまとめたもので、全文は次の7章から構成されている。

第1章は序論であり、本研究の歴史的背景および目的について概説するとともに従来の諸研究との関連について述べる。

第2章および第3章では画像から考察対象物体の輪郭形状を抽出するための前処理系について述べる。次いで、第4章から第6章において、本研究の中核をなす、同一クラスに属す複数の輪郭形状サンプルから、クラスの形状モデルを自動獲得するための要素技術、およびその応用を述べる。

第2章では、図面等の線要素で構成される線画を計算機に入力する際、潰れとかすれのない線画を得るための手法について述べ、その効果を実験的に明らかにする。

第3章では、エネルギー最小化アプローチに基づく、濃淡画像中からの対象物の輪郭形状の抽出・追跡手法、さらに、濃淡動画像系列からの輪郭追跡手法について述べる。また、最適解を効率良く得るためのエネルギー最小化解法を示し、一般の正則化理論の統一的解法への拡張性についても考察する。

第4章では、本モデル獲得手法の核となる2次元輪郭形状の多重スケール凹凸構造マッチング手法について述べる。本手法により各部分輪郭毎に最適スケールの凹凸セグメント対応を得ることができ、従来困難であった変形の大きな図形間のマッチングにも適用可能であることを明らかにする。

第5章では、複数の輪郭形状サンプルからそれらに共通する、より一般的な構造記述を得るための一般化手法について述べる。従来手法を考察し、形状の一般化においては、記号や数値ではないパターン自身の一般化が重要であることを示し、そのための具体的手法を明らかにする。本一般化手法の効果を判別分析の基準であるクラス分離度による定量的評価についても言及する。

第6章では、同一クラスに属す複数の形状サンプルから一つの形状モデルを生成する手法について述べる。本手法が、少数サンプルから学習が可能で、しかも学習と認識を統一的アルゴリズムで扱え、さらに対象に関する事前知識を用いずに広範な形状認識が実現可能であることを示す。

第7章は結論であり、本研究で得られた成果のまとめを行ない、今後の課題について述べる。

目次

1 序論 1

1.1 研究の背景および目的 1

1.2 論文の構成 3

2 線画の画質改善 5

2.1 序言 5

2.2 潰れとかすれの要因 5

2.3 濃度度補正法 7

2.3.1 準備 7

2.3.2 基本原理 7

2.3.3 濃度補正式 8

2.4 実験結果と考察 9

2.4.1 手書き図面への適用 11

2.4.2 2値化閾値変動に対する安定性 11

2.4.3 主観評価実験 18

2.5 結言 19

3 エネルギー最小化原理に基づく濃淡画像からの輪郭抽出および動輪郭追跡 20

3.1 序言 20

3.2 Active contour モデル (Snakes) 21

3.2.1 Snakes の特長 21

3.2.2 エネルギー汎関数 21

3.3 エネルギー最小化の従来解法とその問題点 23

3.3.1 変分法による解法 23

3.3.2 動的計画法による解法 27

3.4 動的計画法に基づくエネルギー最小化解法の再定式化: 新解法 31

3.4.1 DP 方程式の導出 31

3.4.2 比較実験 38

3.4.3 正則化問題の統一的解法への一般化 40

3.5 動輪郭追跡における Snakes の問題点とその解決法 42

3.5.1 追跡処理への適用とその問題点 42

3.5.2 弾性輪郭モデルの導入 45

3.5.3 距離変換に基づくポテンシャル場の生成 46

3.5.4 追跡実験 49

3.6 結言 53

4 多重スケール凹凸構造マッチング 54

4.1 序言 54

4.2 曲率スケールスペース表現 55

4.2.1 基本思想 55

4.2.2 算法 56

4.2.3 階層的記述 59

4.2.4 凹凸構造の階層化アルゴリズム 63

4.3 Coarse-to-fine 戦略による多重スケールマッチング 69

4.3.1 木構造マッチング 69

4.3.2 区間分割マッチング 69

4.4 最適スケール凹凸構造マッチング 73

4.4.1 準備 73

4.4.2 マッチング戦略 73

4.4.3 多重スケールセグメント相違度 76

4.4.4 1対1のセグメント相違度 78

4.4.5 置換コスト 79

4.4.6 最適化問題としての定式化 79

4.5 動的計画法による解法 80

4.6 実験 83

4.7 結言 86

5 輪郭形状からの凹凸構造の一般化 82

5.1 序言 82

5.2 一般化処理の概要 83

5.2.1 問題の所存 83

5.2.2 一般化の処理過程 90

5.3 一般化のためのセグメント対応結果の統合 92

5.3.1 分割セグメント集合族 92

5.3.2 極大集合族 93

5.3.3 無矛盾性の検証 97

5.4 一般化図形の作成 99

5.4.1 線形補間 99

5.4.2 一般化図形作成例 102

5.5 評価 106

5.5.1 クラス分離度 106

5.5.2 評価結果 109

5.6 結言 109

6 Visual モデルの生成とその汎用形状認識系への応用 107

6.1 序言 107

6.2 Visual モデルの生成 108

6.2.1 基本的な考え方 108

6.2.2 Visual モデル生成アルゴリズム 110

6.3 重要度 113

6.3.1 重要度の意味 113

6.3.2 定義式 114

6.4 汎用形状認識への応用 115

6.5 認識実験とその評価 116

6.5.1 未知サンプルの認識 116

6.5.2 重要度の評価 119

6.5.3 車種認識への適用 120

6.6 結言 123

7 結論 120

謝辞 123

参考文献 124

本論文に関する原著論文 130

第 1 章

序論

1.1 研究の背景および目的

外界から様々な知識を獲得、記憶し、それを巧みに利用する能力は、人間が最も得意とし、かつ現在の計算機が最も不得意とするものの一つである。こうした人間の知的能力、すなわち高レベルの情報処理機構を計算機上で実現することは、高度な自動システムを構築する上で極めて重要である。人工知能研究における知識工学は、"知識"を体系的に取り扱う学問領域であり、ここ十数年の間、計算機処理能力の急速な向上と相まって著しい発展を遂げ、他の人工知能分野に多大の影響を与えている。同時に、知識工学で得られた思想・成果は他の多くの専門領域にわたって応用されており、研究面のみならず、実用面での有効性も評価されている。

近年、画像認識の分野においても、知識ベースに基づく画像認識・理解システムが開発され、これまでにない柔軟で高度な認識が実現されている。こうした成果は、"知識"を処理手続きの中に埋め込んだ従来の認識アルゴリズムの枠組では実現困難で、知識が画像処理手続きを制御し、さらに得られた情報から知識に基づく高度な推論を行なういわゆる知識工学的アプローチにより始めて実現可能になったと言えよう。代表的研究例として、航空写真を対象にした画像理解システム[1][2]、屋外シーンの画像理解システム [3]、論理回路図のような図面を対象にした図面理解システム [4] 等が挙げられる。また、視覚情報からの物体形状認識においても、対象の構造に関する知識、すなわち、対象モデルを計算機内部に蓄え、それらと入力データとの照合により物体を認識する手法が主流を占めつつある。これらは、一般にモデルベースの画像理解システムと呼ばれ、従来のボトムアップ的な形状解析のみによる認識とは本質的にアプローチを異にしている。Brooks らの ACRONYM[5][6] はその代表的システムである。

これらの画像理解システムでは、これまで、設計者、すなわち人間が試行錯誤的に対象に関する知識 (モデル) を与えてきた。すなわち、これまでの画像理解においては、知識の利用に研究の関心が

集中し、そうした知識を如何に獲得するかについての議論は殆んどなされていなかった。しかしながら、より汎用性のある画像理解システムを構築するためには、知識の利用のみならず、計算機にそうした知識を自動獲得させることが必要不可欠となる。

一方、計算機による知識獲得は、これまでの精力的な研究の結果、知識工学の中心的課題の一角を形成するに到っている。この分野の先駆的研究が、Winston の積木の学習 [7] であり、これを契機に概念学習の研究が盛んになったと言える。Winston の手法は事例からの構造的概念を学習するものである。ここでいう概念学習とは、帰納的学習とも呼ばれ、ある概念カテゴリーに属すいくつかの具体例から、それらに共通する構造記述を見い出しその概念を規定することを意味する。Winston の研究は、それ以前に研究されてきた神経回路網モデルによるパラメータ学習 [8] とは本質的に異なったアプローチをとるものであり、今日の帰納的学習のまさにルーツ的存在である。その後、帰納的学習は Mitchell[10]、Michalski[9] らによってさらなる発展がもたされた。

しかしながら、これら人工知能における学習の研究は、すべて、予め定義された高次の記号概念をベースにしており、実画像としてのパターン形状を直接取り扱っていない。それゆえ、既存の学習手法を画像データのようなパターン情報にそのまま適用することは困難である。既存の学習アルゴリズムを利用可能にするためには、画像と記号とを何らかの形で"接続"する必要があった。

こうした背景から、実画像から記号概念による対象物体の構造記述が試みられた。Brady & Asada[11] は"smoothed local symmetry" と呼ばれる局所対称性の概念を導入して、対象物体 (2 次元形状) を部分構造に分解し、さらに、意味ネットワーク (グラフ表現) を用いて部分構造相互間の関係記述を得ている。そして、Connell & Brady[12][13] は、この概念記述を用いて、Winston の学習システム ANALOGY[14] に基づく実画像からの概念学習システムを構築した。ANALOGY による一般化は、"2 つのインスタンスが同一の概念に属すとき、両者の間で異なる属性はその概念に無関係である" といった一般化規則 (ablation) に基づいている。

Connell らの研究は、それまで独立に研究されていた画像解析と概念学習とを始めて統合させたという点で意義深いが、このアプローチは次のような問題点を抱えている。

(1) 形状を部分分解して記述するため、一意に部分分解可能な対象にしか適用できない。

(2) 概念記述を行なう際、対象の構造に関する事前知識を必要とする。

(3) 一般化を行なう際、2 つの意味ネットワークの照合 (グラフマッチング) が必須となり、それゆえ、学習・認識が非常に困難となる。(一般にグラフマッチング (グラフの同型問題) は NP 完全の典型的例題である。)

(4) 得られたモデルを可視化することができない。

(4) の可視化は得られたモデルの妥当性の評価を容易にするのに必要な機能である。最近、Segen[15] は上記 (2) の問題を解決する 2 次元形状モデル学習のアルゴリズムを提案しているが、そこでも、多重レベルグラフといったグラフモデルが用いられており、上述した他の問題点を解決するに至っていない。すなわち、上述の問題点は、パターン形状を部分分割し、グラフモデルで表現する従来のアプローチの延長では本質的に解決が困難であると言える。換言すれば、パターン形状からのモデル獲得においては、従来の人工知能研究における学習アルゴリズムの転用ではない、パターン情報との親和性の高い独自なモデル表現とそれに基づく一般化アルゴリズムの研究が渇望されている。

本研究では、これらの状況を踏まえ、パターン形状のモデル獲得に関する従来手法の諸問題を解決することを目的とする。すなわち、本論文は、2 次元形状 (2 次元輪郭図形) を対象として、前述した従来手法の問題点を克服すべく、その形状モデルの自動獲得法に関する著者の研究成果をまとめたものである。本論文では、まず前処理として、線要素で構成される線画から潰れとかすれのない線画を得るための画質改善手法、および濃淡画像中からの考察対象物体の輪郭抽出手法、さらに、濃淡の動画像系列からの対象物体の輪郭追跡手法を提案する。あるパターン形状のモデルを画像から学習するためには、学習に先立って、入力画像から考察対象物体を抽出しておく必要があり、上記、画質改善手法、輪郭抽出手法、および輪郭追跡手法はそのための要素技術である。次いで、同一概念 (同一クラス) に属す複数の輪郭図形サンプルからパターン形状を直接一般化する新たな学習方法を提案する。提案する学習方法は、輪郭の凹凸構造に関する多重スケール記述とそれに基づく凹凸構造の一般化、および visual モデルの生成の各要素技術から成る。本手法は、記号概念をベースに一般化を行なうのではなく、パターン情報との親和性を考慮したパターンベースの一般化を行なうものであり、その意味で従来の学習手法とは根本的にアプローチを異にしている。

1.2 論文の構成

本論文は、序論と結論を含め全体を 7 章で構成している。第 1 章の序論に続いて、まず第 2 章および第 3 章では画像から考察対象物体の輪郭形状を抽出するための前処理系について述べる。次いで、第 4 章から第 6 章において、本研究の中核をなす、同一クラスに属す複数の輪郭形状サンプルから、クラスの形状モデルを自動獲得するための要素技術、およびその応用を述べる。

第 2 章では、図面等の線要素で構成される線画を計算機に入力する際、潰れとかすれのない線画を得るための手法について述べる。すなわち、多値画像として入力した線画の局所的な濃度形状に着目した 2 値化の前処理のための画質改善手法について述べる。

第 3 章では、濃淡画像 (静止画・動画) 中から着目している対象物の輪郭形状を抽出・追跡するた

めの手法、さらに、濃淡動画像系列における輪郭追跡手法への拡張について述べる。ここでは、有力な輪郭抽出手法として最近提案されたエネルギー最小化アプローチ (active contour モデル) を取り上げ、その概要を述べるとともに、エネルギー最小化の解法、および輪郭追跡への適用における問題点を明らかにする。次いで、それらの問題点を解決するための手法について述べる。また、エネルギー最小化解法では、本章で提案する解法の、一般の正則化理論の統一的解法への拡張性について考察する。実験により、提案手法の有効性を示す。

第 4 章では、本モデル獲得手法の核となる 2 次元輪郭形状の多重スケール凹凸構造マッチング手法について述べる。まず、輪郭の凹凸構造を多重スケールで階層的に記述することの重要性を論じ、その有力手法として知られている曲率スケールスペース表現法の概要およびその問題点について述べる。さらに、曲率スケールスペースの枠組でその問題を解決するための手法について述べる。次いで、多重スケール表現に基づく 2 図形の凹凸構造マッチング手法の基本的考え方および具体的アルゴリズムについて述べる。本手法により各部分輪郭毎に最適スケールの凹凸セグメント対応を得ることができ、従来困難であったより変形の大きな図形間のマッチングにも適用可能であることを実際の輪郭形状への適用例により示す。

第 5 章では、複数の輪郭形状サンプルからそれらに共通する、より一般的な構造記述 (一般化図形と呼ぶ) を得るための手法 (一般化手法) について述べる。ここでは、まず従来の一般化手法を考察し、形状の一般化においては、記号や数値ではないパターンとしての一般化図形の重要性を示す。次いで、一般化図形を得るための具体的手法について述べる。本手法は、第 4 章で述べたマッチング手法を同一クラスに属す複数サンプル間に適用した結果に対する統合化手法である。これにより、同一クラスに属するサンプル形状ごとに、クラスに共通する凹凸構造を有し、かつ原形状を保存した一般化図形が得られる。また、本一般化手法を人為的に作成した輪郭図形に適用した例を示し、同時に判別分析の基準であるクラス分離度による定量的評価についても言及する。

第 6 章では、同一クラスに属す複数の形状サンプルから一つの形状モデル (Visual モデルとよぶ) を生成する手法について述べる。すなわち、第 4 章および第 5 章で論じた手法を適用して得た、複数の一般化図形からクラスのプロトタイプモデルである visual モデルを生成する手法について述べる。さらに、これまで述べた本モデル獲得手法の汎用形状認識系への応用について検討する。人工図形および車のプロフィールに対するモデル獲得結果と、未知サンプル認識実験結果を通して、本提案手法が、少数サンプルから学習が可能で、しかも学習と認識を統一的アルゴリズムで扱え、さらに対象に関する事前知識を用いずに広範な形状認識が実現可能であることを明らかにする。

第 7 章では、結論として以上についての成果のまとめを行なう。

第 2 章

線画の画質改善

2.1 序言

計算機への入力を想定しないで作成された既存図面等の線画では、筆記具や手書きによる筆圧の違い、用紙の汚れ等により下地の濃度と線画の濃度コントラストが小さい傾向にある。こうした低品質な線画を計算機へ入力しようとする場合、認識の前処理として 2 値化する際、従来より提案されてきた濃度ヒストグラムから最適 2 値化閾値を求める手法 [16][17][18] では線分の潰れとかすれを同時に除去することが困難であった。

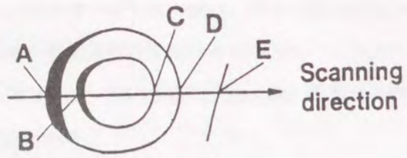
一方、局所的な濃度形状から尾根点、谷点を判定し、適応的に閾値を求め、その閾値より濃度値の低い尾根点と濃度値が高い谷点を各々 1(黒),0(白) とするといった濃度形状に着目した 2 値化手法も提案されている [19]。しかしながらこの手法は濃度値の大小関係のみから尾根・谷を判定するため局所的な濃度変化に敏感となり、また膨大な処理時間がかかるという問題を残していた。

本章では、このような下地と線画との濃度のコントラストが小さい低品質な線画を対象に、原画をそのまま 2 値化するのではなく、有意な山の濃度値を有意な谷の濃度値より大きくする濃度補正処理を施すことによって後段の 2 値化の際の潰れとかすれを抑制する画質改善手法 [20][21] について述べる。

本章で提案する画質改善手法を適用しておくことにより、上述した低品質な線画でも従来の 2 値化手法を用いて潰れとかすれをかなり抑制した 2 値画像を得ることができる。

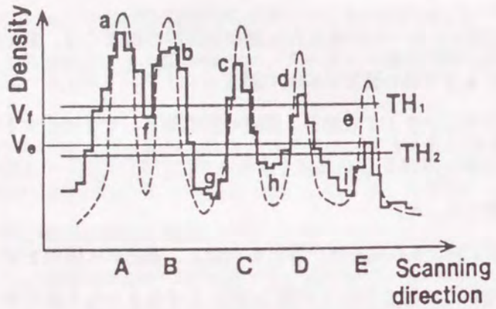
2.2 潰れとかすれの要因

図 2.1 に濃度のコントラストの小さな線画 (多階調) の濃度変化の一例 (説明の簡単化のために 1 次元とする) を示す。今、図 2.1(a) の線画を走査して、同図 (b) のような濃度分布が得られたとす



(a) 原画像

(a)Original image



(b) 濃度形状

図 2.1: 低品質線画の濃度形状の例

る。図 2.1 中で a,b,c,d,e を濃度変化の山、f,g,h,i を濃度変化の谷と呼び、山が有意な線分を形成し、谷が有意な下地 (線分間の空領域) を形成するものとする。また、いま谷 f および山 e の濃度値 V_f, V_e の関係が $V_f > V_e$ であるとする。この時、2 値化の閾値をたとえば図 2.1 の TH_1 に設定すると、山 e を形成する線分が消滅し (かすれ)、また TH_2 に設定すると山 a,b を形成する線分間が潰れる。図 2.1 の場合、2 値化閾値をどのように設定しようと $V_f > V_e$ であるかぎり潰れとかすれを同時に除去することはできない。すなわち $V_f > V_e$ がこの図における潰れとかすれの発生要因であり、これを抑止するには多値の段階で図 1 の破線で示すような濃度変化になるようにする必要がある。

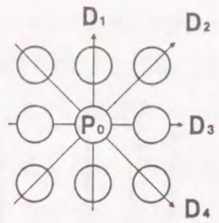


図 2.2: 補正時の画素の組合せ

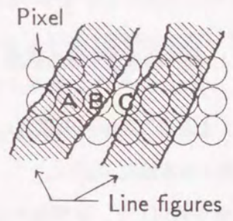


図 2.3: 線画と画素の関係

2.3 濃度度補正法

2.3.1 準備

以下のように記号を定義する。

[隣接画素] 着目画素およびその両隣の 2 画素 (着目画素に関する点対称の 2 画素) をそれぞれ P_0, P_1, P_2 で表わす。従って、(P_0, P_1, P_2) の組は P_0 を共有して 4 組存在することになる。

[画素値] 画素 P_0, P_1, P_2 の濃度をそれぞれ V_0, V_1, V_2 で表わす。ただし, $V_1 \leq V_2$ とする。また、濃度値を正規化し、$0.0 \leq V_0, V_1, V_2 \leq 1.0$, $0.0 =$ 白,$1.0 =$ 黒とする。

[4 方向] P_0 を中心とする両隣 2 画素 P_1, P_2 の方向を、図 2.2 に示す D_1, D_2, D_3, D_4 の 4 方向とする。

2.3.2 基本原理

線画を 8 ～ 16 本／mm、十数階調程度で読み取った場合の線画と画素の関係の一例を図 2.3 に示す。以下、図 2.3 を用いて本濃度補正処理の基本原理を述べる。隣接する画素との濃度値の関係を見ると図 2.3 中の画素 A のような線の中心付近の画素では D_4, D_1, D_3 の方向で図 2.4(a) に示す関係となり、図 2.3 中の画素 B のような近接する 2 本の線に挟まれた画素では D_4, D_1, D_3 の方向で

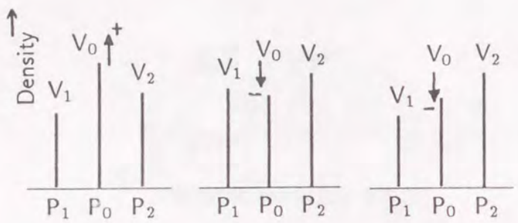


図 2.4: 局所的濃度形状の強調

図 2.4(b) に示す関係となる傾向が強い。また図 2.3 中の画素 C のような線のすそ付近の画素では D_4, D_1, D_3 の方向で図 2.4(c) に示す関係となる傾向が強い。

こうした線図形の持つ濃度形状の性質を利用して、本手法では、ある画素 P_0 の濃度補正値を求めるため、隣接 2 画素 P_1, P_2 との濃度値の関係を調べその関係が図 2.4(a) の時は V_0 をより大きくして山を強調し、図 2.4(b), (c) の時は V_0 をより小さくして谷を強調することにより各方向における濃度補正を行う。そして各方向で得た 4 つの値の相加平均をとり、その値を着目画素の濃度補正値とする。

しかしながら、図 2.4(a) における V_0 が下地と同程度の濃度値である場合、また図 2.4(b) における V_0 が有意な線分と同程度の濃度値である場合は、各々山・谷として強調する必要がない。そこで本手法では、濃度ヒストグラムから、前者については下限の閾値 (以下 TH_l とする)、後者については上限の閾値 (以下 TH_u とする) を設定し、$V_0 < TH_l, V_0 > TH_u$ に対しては濃度補正処理を行なわないこととした。これにより濃度補正対象画素が大幅に減少し、補正処理に要する時間の大幅短縮が可能となる。

2.3.3 濃度補正式

濃度補正式を導入するにあたり、P_0 の濃度補正量を V_0 と $V_2(\geq V_1)$ との差の関数により用めることとした。すなわち、図 2.4(a) の場合、V_2 に対してその濃度差をより強調するように V_0 を大きくすることとし、図 2.4(b),(c) の場合、V_2 より低い V_0 に対してその濃度差をより強調するように V_0 を小さくした。なお P_0 の補正値を V_1, V_2 との関係から求めるのがより厳密であるが、ここでは、V_2 のみでも方向依存性は 4 方向の相加平均により緩和され、山・谷の強調に決定的な影響がな

いと考えられること、V_2 のみを用いることにより、処理アルゴリズムの徹底した簡略化が図れることから $V_2(\geq V_1)$ との濃度差のみに着目することとした。1.3.1 で設定した方向 $D_i(i = 1,2,3,4)$ における着目画素 P_0 の補正値を R_i とすると、上記から R_i は次のようになる。

$$R_i = \begin{cases} V_2 - f(x) & V_0 \leq V_2 \text{のとき} \\ V_2 + f(x) & V_0 > V_2 \text{のとき} \end{cases} \tag{2.1}$$

ここで、$x = |V_2 - V_0|$ とする。また、式 (2.1) における V_2 は全て補正前の値とする。

式 (2.1) の $f(x)$ は補正後の P_0 と P_2 の濃度差を表している。本文の濃度補正は、V_0 と V_2 との差をより強調するものであり、両者の差が小さい程その差を強調しなければならない。しかし、両者の差が小さい時に極端な濃度補正をするのは原濃度を著しく変えてしまうので好ましくない。つまり補正前の濃度差 x に対する補正後の濃度差 $f(x)$ の比 $f(x)/x$ が x が小さい時程大きくなるようにすればよい。このような条件を満たす関数として次式を用いた。$f(x)$ を用いた R_i の例を図 2.5 に示す。

$$f(x) = \sqrt[n]{c^n - (c-x)^n} \tag{2.2}$$

ただし、

$$c = \begin{cases} V_2 & V_0 \leq V_2 \text{のとき} \\ 1.0 - V_2 & V_0 > V_2 \text{のとき} \end{cases} \tag{2.3}$$

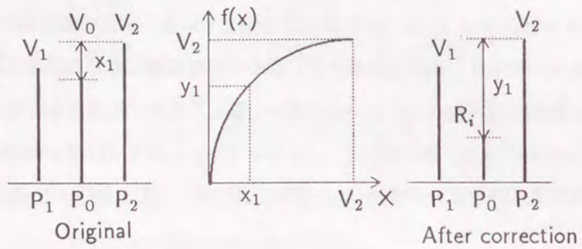
上式で得た $R_i(i = 1,2,3,4)$ に対し、R_i の相加平均をとることにより 4 方向を考慮した補正に拡張する。すなわち着目画素 P_0 の最終濃度補正値 R を次式により求める。

$$R = (R_1 + R_2 + R_3 + R_4)/4 \tag{2.4}$$

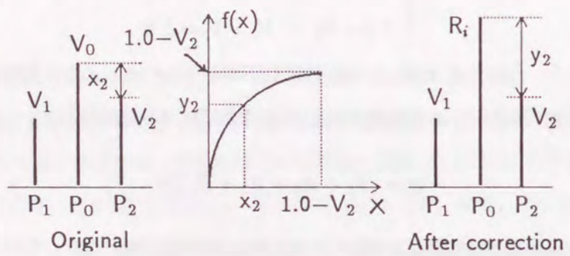
以上により、線画本来の濃度形状を強調した濃度補正が実現可能となる。また式 (2.2) の n の値を変えることにより山・谷の強調の度合を変えることができ、次節で考察するように適切な n の値を用いれば文献 [19] に見られるような、線中 (下地中) の谷 (山) を強制的に 0(1) にすることにより生じる線のかすれ (潰れ) を回避することが可能となる。

2.4 実験結果と考察

本文の画質改善処理法を実際の手書き図面に適用した例を示し，その結果に基づいて考察を加える。対象となる線画は多値読取りが可能な A1 判図面入力装置から解像度 8 本／mm × 8 本／mm, 濃度レベル 16 階調／画素 (4bits/dot) で計算機に入力した。



(a)$V_0 \leq V_2$ のとき



(b)$V_0 > V_2$ のとき

図 2.5: 濃度補正演算

2.4.1 手書き図面への適用

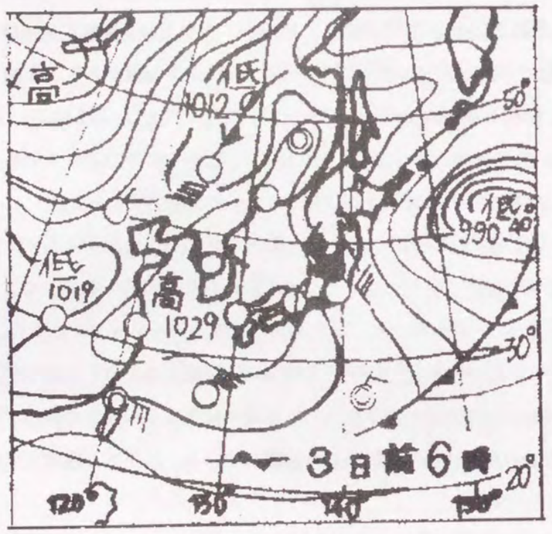
図 2.6(a),(b) に計算機に入力した多値図面を濃度補正処理を適用せずに単純に閾値 ($TH_b = 6, 7$) で 2 値化した (以下単純 2 値化と呼ぶ) 画像と，同図 (c) に本文の濃度補正処理後に最適閾値 ($TH_b = 6$) で 2 値化した画像の一例を示す。画像のサイズは 512×512 画素である。単純 2 値化の場合は、かすれをなくすべく 2 値化の閾値を上げる ($TH_b = 7$) と潰れが生じ (図 2.6(a))、潰れをなくすべく閾値を下げる ($TH_b = 6$) とかすれが生じる (図 2.6(b))。しかも両者の差は 1 であり、すなわち原図面は、単純 2 値化では潰れとかすれを同時に除去できないような品質である。

これに対して、濃度補正処理後に 2 値化した場合，図 2.6(c) に示すように，潰れとかすれがかなり除去され，その意味で単純 2 値化に比べて 2 値画像の品質が大幅に向上している。濃度補正処理では濃度補正式における n を 2 に設定し，また閾値は濃度レベル 0 ~ 15 のうち、$TH_u = 9, TH_l = 5$ に設定した。図 2.7 に他の線画 (設備図面) を用いた実験結果を示す。単純 2 値化では線や文字に潰れとかすれが目立つが、本画質改善後 2 値化した結果ではそれらがかなり抑制されていることがわかる。こうした結果は他の線画の適用実験からも確認している。以上から提案した画質改善処理の効果は非常に大きいことが分かる。

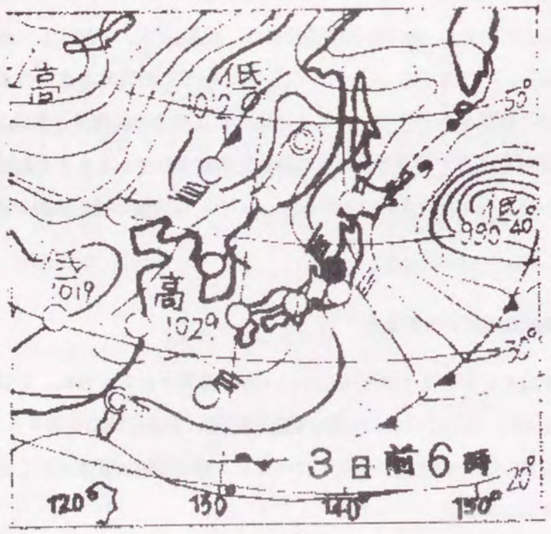
2.3 節で述べた濃度補正式において，補正の強弱をきめる n の値の補正効果を、$n = 1.5, 2, 3, 4$ の 4 段階について実際の処理例から検討する。図 2.8 に各 n に対する $f(x)$ の概形を示し、図 2.9 に各 n に対する処理結果を示す。図 2.9 の処理結果は、TH_l, TH_u を設定して濃度補正処理した後に 2 値化したものである。同図を見ると，$n = 2, 3$ については画質にあまり差がないことがわかる。$n = 1.5$ の場合は多少補正が不十分で，$n = 4$ の場合には線分の線幅が必要以上に細くなりすぎている。これは濃度補正処理により濃度形状の尾根線が先鋭化されていることに起因する。他の線画についても同様な傾向が見られた。以上のことから，$n = 2 \sim 3$ 程度の補正曲線で補正するのが適当だと言える。

2.4.2 2 値化閾値変動に対する安定性

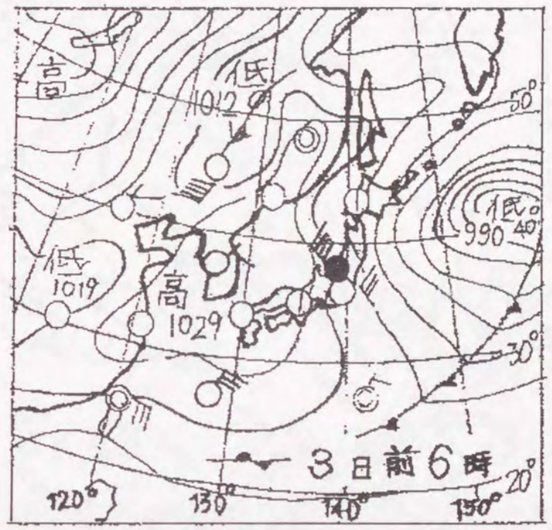
濃度補正処理を施すことにより濃度変化の山・谷が強調されるために、2 値化の閾値の設定範囲を広く設定できる。図 2.10(a),(b),(c) は濃度補正処理後、2 値化閾値を各々 6,7,8 に設定して得た 2 値画像であり、図 2.6(a),(b) の単純 2 値化に比べて、2 値化閾値の変動に対しかなり安定した画像に変換されていることがわかる。



(a) 単純 2 値化による結果 ($TH_b = 6$)

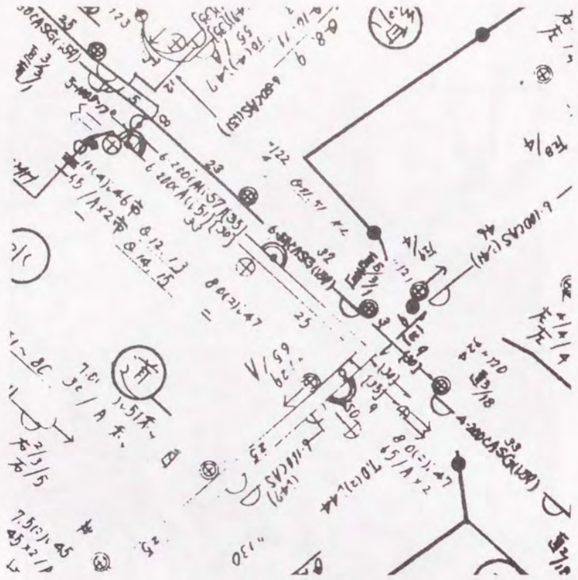


(b) 単純 2 値化による結果 ($TH_b = 7$)

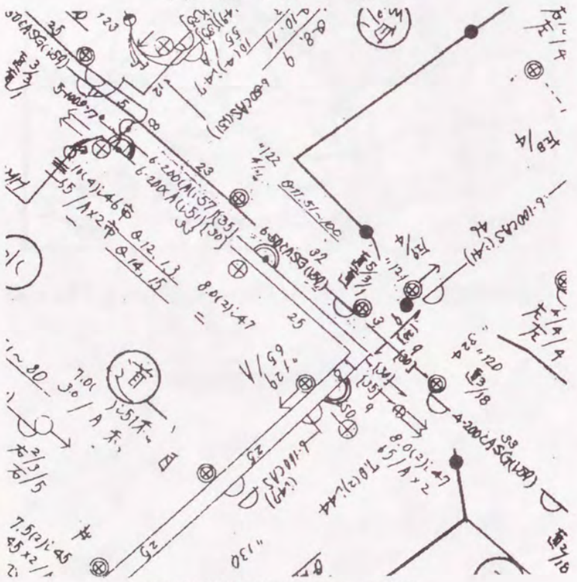


(c) 濃度補正後 2 値化した結果 ($TH_l = 5, TH_u = 9, TH_b = 6$)

図 2.6: 画質改善効果の例



(a) 単純２値化による結果



(b) 濃度補正後２値化した結果

図 2.7: 画質改善効果の例

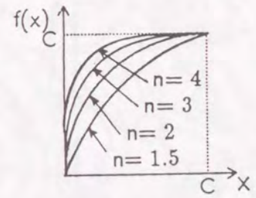
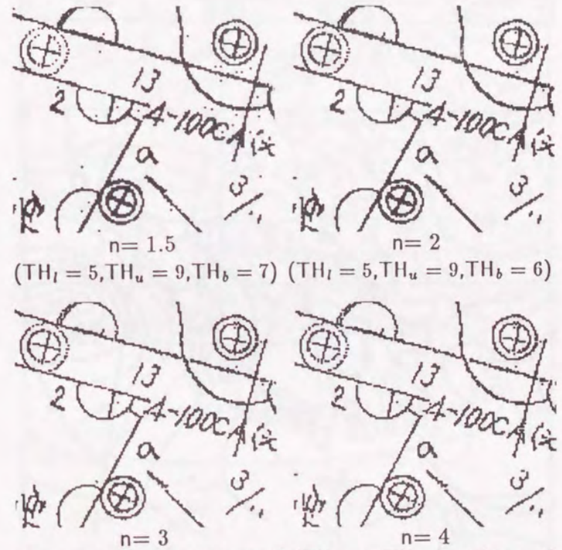
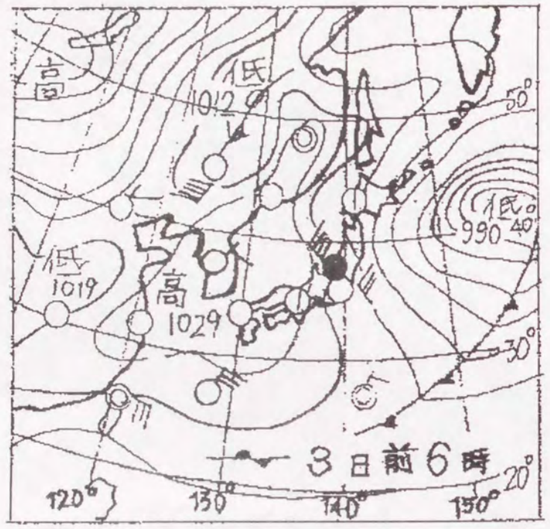


図 2.8: 補正曲線

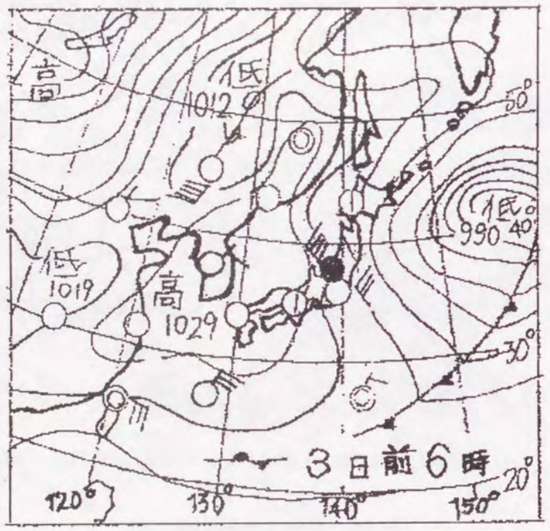


n= 1.5 ($TH_l = 5, TH_u = 9, TH_b = 7$)
n= 2 ($TH_l = 5, TH_u = 9, TH_b = 6$)
n= 3 ($TH_l = 5, TH_u = 9, TH_b = 6$)
n= 4 ($TH_l = 5, TH_u = 9, TH_b = 6$)

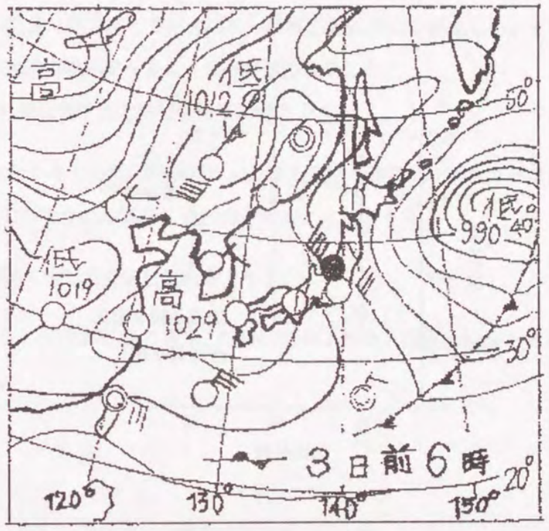
図 2.9: 補正量の評価



(a)$TH_b = 6(TH_l = 5, TH_u = 9)$



(b)$TH_b = 7(TH_l = 5, TH_u = 9)$



(c)$TH_b = 8(TH_l = 5, TH_u = 9)$

図 2.10: 2 値化閾値の変動に対する安定性

表 2.1: 評価カテゴリー

MOS 値	
5	非常に読み易い
4	読み易い
3	普通
2	読み難い
1	非常に読み難い

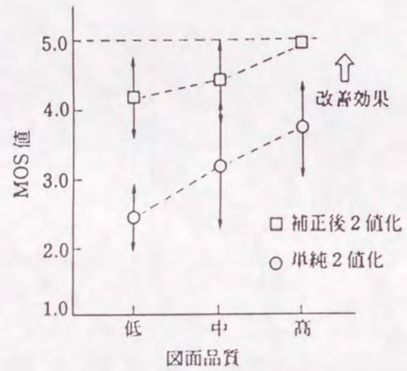


図 2.11: 主観評価結果

2.4.3 主観評価実験

ディスプレイ上に表示された図面中の文字・記号の読み易さが画質改善処理の有無によりどの程度変わるかについての定量的評価を得るため 5 段階評定尺度法による主観評価実験を行なった。評価結果を図 2.11 に示す。被験者は 10 名で、データは低、中、高品質の図面を対象とし、単純 2 値化と濃度補正後 2 値化したものを合計 64 サンプル提示した。評価カテゴリは表 1 に示す 5 段階とした。図 2.11 から画質改善により低品質図面に対して約 1.7 の MOS 値 (Mean Of Score: 評定値の平均値) の向上が見られた。

また、図面の図形認識システム [22] の前処理として本章で提案した画質改善手法を用いた結果、認識率で約 10% の向上が見られ、図形認識の前処理としての有効性を確認している [23]。

本文で提案した濃度補正処理は，2.3 節に示した補正式により，濃度値 V_0, V_2 のみから，濃度補正値 R_i を求めるものである。したがって，濃度補正処理のルックアップテーブル化 (LUT 化) が可能となり、2.3 で述べた閾値 TH_l, TH_u の導入に加えて、補正処理の計算時間のさらなる短縮を図ることができる。実際、本章で提案した濃度補正演算を、図面認識装置 [24] 上に実装し、A1 判図面 (6800×4800 画素) を処理した結果、わずか 30 秒程度であった。

2.5 結言

本章では、2 値化の前処理としての画質改善の必要性について論じ，その一手法として画素近傍の濃度変化に基づく簡易で高速度な実用性の高い画質改善処理法を提案した。また，実際の低品質線画に対する本手法の適用実験結果を基に，その有効性を示した。

本章で提案した画質改善手法の特長を以下に整理する。

(1) 従来困難であったかなりの低品質線画に対しても良好な濃度補正がなされるため、補正後、従来の 2 値化手法でかなりの品質の 2 値画像を得ることができる。

(2) 2 値化の閾値変動に対して安定した画像となる。

(3) LUT 化およびハード化が容易であり、設備図面のような大規模な線画を実用時間内で処理することができる。

第 3 章

エネルギー最小化原理に基づく濃淡画像からの輪郭抽出および動輪郭追跡

3.1 序言

画像中から着目対象物体 (パターン) を抽出することは、パターン認識の前処理として避けられない重要課題である。とりわけ、本論文の研究テーマである、画像からの対象物体形状のモデル獲得においては、それに先立って考察対象物体形状を抽出しておくことが必須となる。物体の知覚にはその境界線で十分であるという多くの心理学的根拠に基づき、パターンの抽出ではその境界線を検出することが通常行なわれている。実際、画像処理の分野で対象パターンの境界線を抽出すべく数多くのエッジ抽出手法が提案されている。古典的な手法に、微分オペレータによる方法 [25][26][27][28]、エッジモデルパラメータ推定による方法 [29] 等がある。また、近年では、逆問題 (inverse problem) としてエッジ抽出問題を定式化し最適フィルタを導く数理的手法 [30][31] も提案されている。ところが、これらの手法はあくまで局所的な処理のみによるため、本来接続されるべき輪郭線が不連続となることがあり、何らかの接続処理 [32] を施さねばならないという問題が残されていた。

一方、Kass ら [33] は、上記問題を解決するための、エネルギー最小化アプローチによるノンパラメトリックな輪郭抽出モデル (Active contour model: Snakes) を提案し、形状抽出のみならず、主観的輪郭の抽出さらには動輪郭追跡への適用も試みている。しかしながら、Snakes におけるエネルギー最小化は、従来、変分法により解かれていたため解の収束性・安定性に問題を残していた。さらに、Snakes による輪郭追跡処理では、追跡対象物体が隣接フレーム間でごくわずかしか移動、変形しないような対象に限定され、一般の追跡問題への適用が困難であった。

そこで、本章では Snakes の能動的なモデルの利点を生かすべく、上記問題点を解決するためのエネルギー最小化解法 [34] および動輪郭追跡手法 [35][36] を提案する。すなわち、前者については動的計画法 (DP:dynamic programming) に基づく定式化について述べ、後者については、Snakes に弾性的制約を付与した弾性輪郭モデルによる輪郭追跡手法について述べる。

3.2 Active contour モデル (Snakes)

3.2.1 Snakes の特長

Snakes は基本的には、変形可能なスプライン曲線で、その形状は、線やエッジのような画像特徴からのイメージ力により制御される。すなわち、図 3.1 に示すように、形状モデルとしての snake を画像中の抽出対象輪郭の近傍に置き (通常これは会話的に実行される)、これを初期値として、3.2.2 で述べるエネルギーの極小値を見い出すことにより対象物の輪郭を抽出する。従って、既存の輪郭抽出手法がボトムアップアプローチであるのに対して、Snakes はモデルベースのトップダウンアプローチと言える。Snakes は、従来手法にはない次のような特長を有する。

(1) **閉輪郭性**: 輪郭が (ノイズや遮蔽輪郭により) 欠損していても閉輪郭として抽出することができる。従って、主観的輪郭の抽出も可能である。

(2) **外力の陽な記述**: 種々の外力の導入による拘束の修正が容易にできる。すなわち、高レベルの知識のフィードバックが可能である。

(3) **高次元への拡張性**: Snakes は本来 1 次元であるが、これを 2 次元 (ネットモデル [37]) 等の高次元へ拡張可能である。

3.2.2 エネルギー汎関数

snake 上の点を $\boldsymbol{v}(s) = (x(s), y(s)),\ s \in [0,1]$ とパラメータ表示すると、snake による輪郭抽出では、まず抽出対象輪郭の近傍に snake を置き、これを初期値とし、snake 自身の内部スプラインエネルギー (E_{int}) とイメージ力から導かれる外部エネルギー (E_{ext}) との和:

$$E_{snakes}(\boldsymbol{v}) = \int_0^1 \Big(E_{int}(\boldsymbol{v}(s)) + E_{ext}(\boldsymbol{v}(s))\Big) ds \tag{3.1}$$

を最小化するように snake $(\boldsymbol{v}(s), s \in [0,1])$ を位置決めする。すなわち、数学的には $\boldsymbol{v}(s)$ を変関数とする汎関数の極値問題として定式化される。そして、snake の動的な振舞いはエネルギー最小化の反復演算の各近似解に対応している。

snake の内部スプラインエネルギーは次式で定義される。

$$E_{int}(\boldsymbol{v}(s)) = \frac{1}{2}\left(\alpha(s)\left|\frac{d\boldsymbol{v}(s)}{ds}\right|^2 + \beta(s)\left|\frac{d^2\boldsymbol{v}(s)}{ds^2}\right|^2\right) \tag{3.2}$$

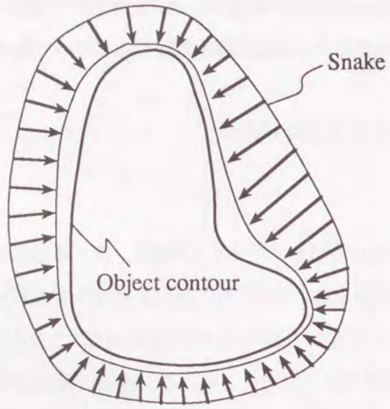


図 3.1: 輪郭抽出における snake の振舞い

$\alpha(s), \beta(s)$ は重み係数で、通常、正定数として扱われる。上式から容易に分かるように、内部スプラインエネルギーを減少させるということは、snake を収縮させ、かつ滑らかな曲線となるように変形させることと等価である。

一方、外部エネルギーは線分、エッジ等の画像特徴に snake を引き付けるためのものであり、換言すれば、外部エネルギーは snake の収縮と変形を抑える働きをする。外部エネルギーはイメージ力 (E_{image}) と、画像自身とは無関係にユーザが会話的に与える外部制約力 (E_{con}) とから成る。

E_{image} はさらに 3 種の汎関数の線形結合として次式のように表される。

$$E_{image} = w_{line}E_{line} + w_{edge}E_{edge} + w_{term}E_{term} \tag{3.3}$$

E_{line}, E_{edge} および E_{term} は各々、線分、エッジ、線分の終端もしくは角に対応し、重み $w_{line}, w_{edge}, w_{term}$ を適当に調整することにより種々の画像に適用させることができる。通常の濃淡画像では、$w_{edge} = 1.0, w_{line} = w_{term} = 0.0$ として十分である。これらの汎関数の具体例として、

$$E_{line} = I(x,y) \tag{3.4}$$

$$E_{edge} = -|\nabla I(x,y)| \tag{3.5}$$

$$E_{term} = \frac{\partial^2 C(x,y)/\partial \mathbf{m}^2}{\partial C(x,y)/\partial \mathbf{n}} \tag{3.6}$$

が用いられる。ここで, $I(x,y), C(x,y)$ は画像中の点 (x,y) での濃度値、および濃度勾配の曲率の値を表す。また、$\mathbf{n}$ は点 (x,y) での濃度勾配角を表す単位ベクトル $\mathbf{n} = (\cos\theta, \sin\theta)$ で、$\mathbf{m}$ はそれに垂直な単位ベクトル $\mathbf{m} = (-\sin\theta, \cos\theta)$ である。

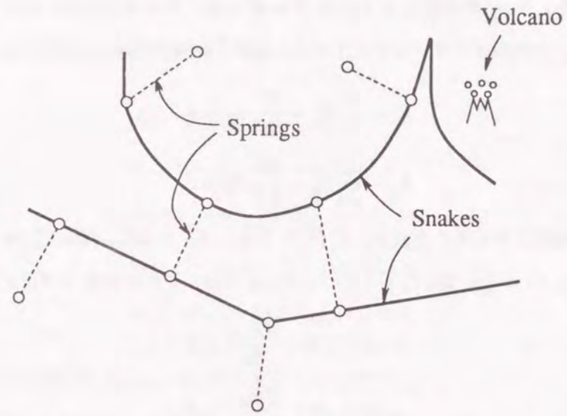


図 3.2: Snakes モデルにおける外部拘束の例

E_{con} では図 3.2 に示す様な"バネ (spring)" と"火山 (volcano)" がユーザインタフェースとして用意されており、外部拘束力に用いられる。図 3.2 では 2 つの snake(太線) に対して、それらを結ぶバネと、一方を snake に他方をスクリーン上 (画像上のある点) に固定したバネ、そして一つの"火山" が snake の近傍に置かれている。こうしたユーザインタフェースは、誤った輪郭に snake が収縮したような場合における強制的な修正手段として利用される。

最近では、これらのオリジナルなエネルギー関数に加え、圧力エネルギー [38] 等が提案され改良が加えられている。この圧力エネルギーは snake を凹の部分輪郭にフィットさせるのに有効である。

3.3 エネルギー最小化の従来解法とその問題点

3.3.1 変分法による解法

前節で説明したエネルギー関数を再整理すると、Snakes におけるエネルギー積分は、

$$E_{snakes}(\boldsymbol{v}) = \int_0^1 \left(E_{ext}(\boldsymbol{v}(s)) + \frac{1}{2}\left(\alpha(s) \left| \frac{d\boldsymbol{v}(s)}{ds} \right|^2 + \beta(s) \left| \frac{d^2\boldsymbol{v}(s)}{ds^2} \right|^2 \right) \right) ds \tag{3.7}$$

となる。故に上式を最小ならしめる関数 $\boldsymbol{v}(s)$ を求めれば良いことになる。こうした汎関数の極値問題は、一般に変分法を用いて解くことができる [39]。

(1) 解法

$\boldsymbol{v}(s) = (x(s), y(s))$ であるから式 (3.7) の汎関数は、

$$F\Big(x(s), y(s), \dot{x}(s), \dot{y}(s), \ddot{x}(s), \ddot{y}(s)\Big) \tag{3.8}$$

の形をしている。但し、$\dot{x} = dx/ds, \dot{y} = dy/ds, \ddot{x} = d^2x/ds^2, \ddot{y} = d^2y/ds^2$。従って、式(3.8)の極値を与える $x(s), y(s)$ は次のオイラー・ラグランジュ方程式を満たさなければならない[†]。

$$F_x - \frac{\partial}{\partial s}F_{\dot{x}} + \frac{\partial^2}{\partial s^2}F_{\ddot{x}} = 0 \tag{3.9}$$

$$F_y - \frac{\partial}{\partial s}F_{\dot{y}} + \frac{\partial^2}{\partial s^2}F_{\ddot{y}} = 0 \tag{3.10}$$

F_x 等は F を x で偏微分することを表す。式(3.7)より、$F_x = \partial E_{ext}/\partial x, F_y = \partial E_{ext}/\partial y, F_{\dot{x}} = \frac{1}{2}\alpha \cdot 2\dot{x}, F_{\dot{y}} = \frac{1}{2}\alpha \cdot 2\dot{y}, F_{\ddot{x}} = \frac{1}{2}\beta \cdot 2\ddot{x}, F_{\ddot{y}} = \frac{1}{2}\beta \cdot 2\ddot{y}$ であるから、これらを式(3.9),(3.10)に代入して、

$$-\alpha\ddot{x} + \beta\ddot{x} + \frac{\partial E_{ext}}{\partial x} = 0 \tag{3.11}$$

$$-\alpha\ddot{y} + \beta\ddot{y} + \frac{\partial E_{ext}}{\partial y} = 0 \tag{3.12}$$

を得る。

以上は、連続系での議論であるが、ディジタル画像上で実現するためのには離散近似した以下に述べる方法により解くことになる。すなわち、snake上の離散化点列を $\boldsymbol{v}_i = (x_i, y_i) = (x(ih), y(ih)), i = 1, 2, \ldots, n$ (h はステップサイズ) とおくと式(3.7)は、

$$E_{snake} = \sum_{i=1}^{n}\Big(E_{ext}(i) + E_{int}(i)\Big) \tag{3.13}$$

と書きかえられる。そして、$d\boldsymbol{v}(s)/ds \simeq (\boldsymbol{v}_i - \boldsymbol{v}_{i-1})/h, \quad d^2\boldsymbol{v}(s)/ds^2 \simeq (\boldsymbol{v}_{i+1} - 2\boldsymbol{v}_i + \boldsymbol{v}_{i-1})/h^2$ と差分法により近似することにより、

$$E_{int}(i) = \frac{1}{2}\Big\{\alpha_i \left| \frac{\boldsymbol{v}_i - \boldsymbol{v}_{i-1}}{h} \right|^2 + \beta_i \left| \frac{\boldsymbol{v}_{i+1} - 2\boldsymbol{v}_i + \boldsymbol{v}_{i-1}}{h^2} \right|^2\Big\} \tag{3.14}$$

となる。従って、式(3.11),(3.12)に対応する離散系でのオイラー方程式は、

$$\begin{aligned}
&\frac{1}{h^2}\Big\{\alpha_i(\boldsymbol{v}_i - \boldsymbol{v}_{i-1}) - \alpha_{i+1}(\boldsymbol{v}_{i+1} - \boldsymbol{v}_i)\Big\} \\
&+\frac{1}{h^4}\Big\{\beta_{i+1}(\boldsymbol{v}_{i+2} - 2\boldsymbol{v}_{i+1} + \boldsymbol{v}_i) - 2\beta_i(\boldsymbol{v}_{i+1} - 2\boldsymbol{v}_i + \boldsymbol{v}_{i-1}) + \beta_{i-1}(\boldsymbol{v}_i - 2\boldsymbol{v}_{i-1} + \boldsymbol{v}_{i-2})\Big\} \\
&+(f_x(i), f_y(i)) = 0
\end{aligned} \tag{3.15}$$

となる。但し、

$$f_x(i) = \frac{\partial E_{ext}(i)}{\partial x_i},$$
$$f_y(i) = \frac{\partial E_{ext}(i)}{\partial y_i}$$

[†]オイラー・ラグランジュ方程式の導出についてはエルスゴルツの著書[39]が詳しい。

とする。

式(3.15)を $h = 1$ として整理すると、

$$\begin{aligned}
&\boldsymbol{v}_{i-2} \cdot \beta_{i-1} + \\
&\boldsymbol{v}_{i-1} \cdot (-\alpha_i - 2\beta_i - 2\beta_{i-1}) + \\
&\boldsymbol{v}_i \cdot (\alpha_i + \alpha_{i+1} + \beta_{i+1} + 4\beta_i + \beta_{i-1}) + \\
&\boldsymbol{v}_{i+1} \cdot (-\alpha_{i+1} - 2\beta_{i+1} - 2\beta_i) + \\
&\boldsymbol{v}_{i+2} \cdot \beta_{i+1} + (f_x(i), f_y(i)) = 0
\end{aligned} \tag{3.16}$$

となる。但し、$\boldsymbol{v}_i$ は閉輪郭 ($\boldsymbol{v}_{n+k} = \boldsymbol{v}_n (k < n)$) としておく[†]。

さらに、式(3.16)において、$a_k = \beta_{k+1}, b_k = -\alpha_{k+1} - 2\beta_{k+1} - 2\beta_k, c_k = \alpha_k + \alpha_{k+1} + \beta_{k+1} + 4\beta_k + \beta_{k-1}$ とおき、$i = 1, 2, \ldots, n$ を逐次代入することにより、次式に示すような連立方程式を得る。

$$A\boldsymbol{x} + \boldsymbol{f}_x(\boldsymbol{x}, \boldsymbol{y}) = 0 \tag{3.17}$$

$$A\boldsymbol{y} + \boldsymbol{f}_y(\boldsymbol{x}, \boldsymbol{y}) = 0 \tag{3.18}$$

但し、行列 A は $n \times n$ の実対象行列かつ5重帯行列である:

$$\begin{pmatrix}
c_1 & b_1 & a_1 & & & & & & a_{n-1} & b_n \\
b_1 & c_2 & b_2 & a_2 & & & & & & a_n \\
a_1 & b_2 & c_3 & b_3 & a_3 & & & & & \\
 & a_2 & b_3 & c_4 & b_4 & a_4 & & & & \\
 & & & \cdots & \cdots & \cdots & & & & \\
 & & & & \cdots & \cdots & \cdots & & & \\
 & & & & & \cdots & \cdots & \cdots & & \\
 & & & & & & \cdots & \cdots & \cdots & \\
 & & & & a_{n-3} & b_{n-2} & c_{n-1} & b_{n-3} & a_{n-3} & \\
 & & & & & a_{n-4} & b_{n-3} & c_{n-2} & b_{n-2} & a_{n-2} \\
a_{n-1} & & & & & & a_{n-3} & b_{n-2} & c_{n-1} & b_{n-1} \\
b_n & a_n & & & & & & a_{n-2} & b_{n-1} & c_n
\end{pmatrix}$$

[†]Snakes は本来、開輪郭でも、閉輪郭でも定義可能であるが、対象物の輪郭抽出という観点で閉輪郭として定式化しておく。

また、

$$
\begin{aligned}
\boldsymbol{x} &= (x_1, x_2, \ldots, x_n)^t \\
\boldsymbol{y} &= (y_1, y_2, \ldots, y_n)^t
\end{aligned} \tag{3.19}
$$

$$
\begin{aligned}
\boldsymbol{f}_x(\boldsymbol{x}, \boldsymbol{y}) &= \Big(\frac{\partial E_{ext}(1)}{\partial x_1}, \ldots, \frac{\partial E_{ext}(n)}{\partial x_n}\Big)^t \\
\boldsymbol{f}_y(\boldsymbol{x}, \boldsymbol{y}) &= \Big(\frac{\partial E_{ext}(1)}{\partial y_1}, \ldots, \frac{\partial E_{ext}(n)}{\partial y_n}\Big)^t
\end{aligned} \tag{3.20}
$$

である。t は転置を表す。式 (3.20),(3.21) から分かるように、$\boldsymbol{f}_x(\boldsymbol{x}, \boldsymbol{y})$, $\boldsymbol{f}_y(\boldsymbol{x}, \boldsymbol{y})$ は共に $\boldsymbol{x}, \boldsymbol{y}$ の関数である。従って、式 (3.17),(3.18) は非線形連立方程式となり、その解ベクトル $\boldsymbol{x}, \boldsymbol{y}$ は次式に示す反復法により解くことができる:

$$
\begin{aligned}
\boldsymbol{x}_t &= (A + \gamma I)^{-1}(\gamma \boldsymbol{x}_{t-1} - \boldsymbol{f}_x(\boldsymbol{x}_{t-1}, \boldsymbol{y}_{t-1})) \\
\boldsymbol{y}_t &= (A + \gamma I)^{-1}(\gamma \boldsymbol{y}_{t-1} - \boldsymbol{f}_y(\boldsymbol{x}_{t-1}, \boldsymbol{y}_{t-1}))
\end{aligned} \tag{3.21}
$$

ここに、I は単位行列であり、γ は収束の速度を制御するためのパラメータである。

(2) 問題点

(1) で述べた変分法には次のような問題点がある。

(i) 十分性: オイラー・ラグランジュ方程式は極値であるための" 必要条件" で、十分条件ではない。従って、極小値を求める問題に対して、極大値を求めかねない。

(ii) 誤差: ディジタル画像で Snakes を実現する際、離散データの高階微分を必要とし数値誤差が大きくなる。

(iii) 安定性: 微分法は変数が連続的に変動する性質にその基礎をおいているため、それらから得られる結果は部分的な変動や小さな誤差にも大きく影響される。

(i) に関してはより高次の必要条件 (ヤコビ、ワイヤストラス)[39] を用いることも可能であるが、これらの条件はさらに高階の微分を含むため適用が困難である。(iii) の安定性に関する議論は、式 (3.20),(3.21) を基に解析的な考察が可能である [40]。

すなわち、式 (3.21) を次のように書き直す。

$$
\boldsymbol{x}_{t+1} = \gamma B^{-1} \boldsymbol{x}_t + B^{-1} C_t \tag{3.22}
$$

ここで、$B = A + \gamma I$, $C_t = -\boldsymbol{f}_x(\boldsymbol{x}_t, \boldsymbol{y}_t)$ である。式 (3.22) を逐次代入することにより次式が得られる。

$$
\boldsymbol{x}^{t+1} = (\gamma B)_{t+1} \boldsymbol{x}_0 + \sum_{i=0}^{t} \gamma^i B^{i+1} C_{t-i} \tag{3.23}
$$

今、最終解を $\boldsymbol{x}_* = \gamma B \boldsymbol{x}_* + B C_\infty$ とおくと、反復で集積される誤差は次式のようになる。

$$
\begin{aligned}
\boldsymbol{x}_{t+1} - \boldsymbol{x}_* &= \gamma B(\boldsymbol{x}_t - \boldsymbol{x}_*) + B(C_t - C_\infty) \\
&= (\gamma B)^{t+1}(\boldsymbol{x}_0 - \boldsymbol{x}_*) + \sum_{i=0}^{t} \gamma^i B^{i+1}(C_{t-i} - C_\infty)
\end{aligned} \tag{3.24}
$$

ここで、すべての i, t に対して、$C_{t-i} = C_\infty$ なる場合、すなわち各反復に対して $C_t = -\boldsymbol{f}_x(\boldsymbol{x}_t, \boldsymbol{y}_t)$ が変化しないとすると、式 (3.24) はより簡単に、

$$
\boldsymbol{x}_{t+1} - \boldsymbol{x}_* = (\gamma B)^{t+1}(\boldsymbol{x}_0 - \boldsymbol{x}_*) \tag{3.25}
$$

と書ける。さらに、行列 γB の固有値を $\lambda_1, \lambda_2, \ldots, \lambda_n$ とし、各固有値に対する固有ベクトルを $\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_n$ と書くことにすると、任意のベクトルはこれらの 1 次結合で表されることから、

$$
\boldsymbol{x}_0 - \boldsymbol{u}_* = a_1 \boldsymbol{u}_1 + a_2 \boldsymbol{u}_2 \cdots + a_n \boldsymbol{u}_n \tag{3.26}
$$

と書ける。但し、$a_i (i = 1, 2, \ldots, n)$ は定数である。従って、

$$
a_1 \lambda_1^t \boldsymbol{u}_1 + a_2 \lambda_2^t \boldsymbol{u}_2 + \cdots + a_n \lambda_n^t \boldsymbol{u}_n = 0 \tag{3.27}
$$

を得る。故に、式 (3.27) において、ある i に対して $\lambda_i > 1$ とすると、$t \to \infty$ のとき、$(\gamma B)^t$ の極限は存在しないので、上記過程は収束しないことになる。しかも、通常は各反復に対して C_t が変化するため $C_{t-i} \neq C_\infty$ となり一般に収束に関する予測が困難となる。

3.3.2 動的計画法による解法

Snakes モデルの別の解法に動的計画法 (DP:Dynamic Programming) を用いたものがある。これは、Amini[41] らにより提案された解法で、離散画像に直接適用でき、かつ汎関数が直接最適化 (最小化) される、という点で前節で述べた変分法より明らかに優れた解法である。

(1) 解法

再び、離散化したエネルギー関数、

$$
E_{snake}(i) = \sum_{i=1}^{n} \Big(E_{ext}(i) + E_{int}(i) \Big) \tag{3.28}
$$

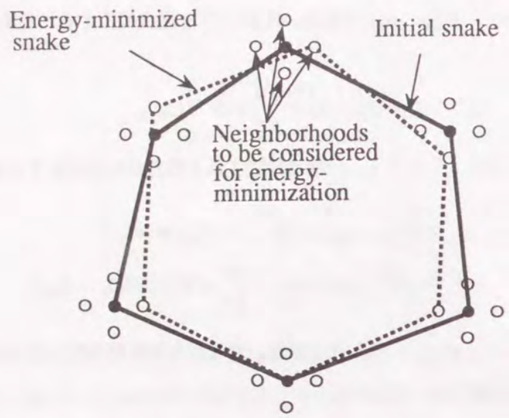


図 3.3: DP によるエネルギー最小化解法

を考える。上式を、初期輪郭を構成する制御点列 $\{v_i{}^0\}_{i=1}^n$ を初期値として、その周辺の局所最小値を求めることを考える。図 3.3 に 6 点からなる単純な snake を示す。今、各点の可動範囲が図 3.3 中の ∘ で示す 4 近傍であるとする。このとき、局所最小値を求めるということは、各点をその 4 近傍で動かし、その全ての組合せについてその都度エネルギー計算を行ない、その値が最小となる点の組を探索することに等しい。例えば、図 3.3 では点線で示す輪郭がそれに相当する。この処理をエネルギーの減少がなくなるまで繰り返すことにより最終的にエネルギーの最小値を得ることができる。

ところが、各繰り返し処理において、v_i のとり得る状態数の全ての組合せをしらみつぶし法で計算すると、その場合の数は m^n (図 3.3 では 5^6) となり、実際の snake のような n が数十以上場合ではもはや現実的な時間で解くことが不可能となる。従って、最適解を効率良く求める解法が必須となる。

動的計画法はまさにこの最適化問題を多項式時間で解くためのアルゴリズムを与えるものである。すなわち、上記最適化問題を、$v_1, v_2, \ldots, v_n$ を逐次決定する多段決定過程として定式化する。

式 (3.13),(3.14) より、式 (3.29) のエネルギー汎関数は、v_i 自身、および、(v_i, v_{i+1}) の 2 項間、および (v_i, v_{i+1}, v_{i+2}) の 3 項間の関数で決定され、形式的に次式のように表すことができる。

$$E_{total} = \sum_{i=1}^{n} \Big\{ f_i(v_i) + g_i(v_i, v_{i+1}) + h_i(v_i, v_{i+1}, v_{i+2}) \Big\} \tag{3.29}$$

但し、閉輪郭の周期性より、$v_{n+k} = v_k (k < n)$ とする。式 (3.29) において、f_i は外部エネルギーに対応し、g_i, h_i は内部エネルギーに対応する。

今、目的関数が次式のような形をしているとする。

$$E(v_1, v_2, \ldots, v_n) = e_1(v_1, v_2) + e_2(v_2, v_3) + \cdots + e_{n-1}(v_{n-1}, v_n) \tag{3.30}$$

すると、v_i は、i 番目の決定段階での状態変数に相当する。このとき、部分和 $e_1(v_1, v_2) + \cdots + e_{i-1}(v_{i-1}, v_i)$ の最小値 (最適値) が v_i の関数として得られたとし、それを $S_{i-1}(v_i)$ で表すと、最適性の原理 [43] から $S_i(v_{i+1})$ は $S_{i-1}(v_i)$ と $e_i(v_i, v_{i+1})$ のみにより決定される。すなわち、次の DP 方程式を得る。

$$S_i(v_{i+1}) = \min_{v_i} \Big\{ S_{i-1}(v_i) + e_i(v_i, v_{i+1}) \Big\}, \quad \text{for} \quad i \geq 1, \quad S_0 = 0 \tag{3.31}$$

次に、目的関数が次式のような形の場合を考える。

$$E(v_1, v_2, \ldots, v_n) = e_1(v_1, v_2, v_3) + e_2(v_2, v_3, v_4) + \cdots + e_{n-2}(v_{n-2}, v_{n-1}, v_n) \tag{3.32}$$

e_{i-2} と e_{i-1} に着目すると両者は 2 変数 v_i, v_{i-1} を共有している。そこで、(v_i, v_{i-1}) を一つの状態変数ベクトル組として固定して考え、部分和 $e_1(v_1, v_2, v_3) + \cdots + e_{i-2}(v_{i-2}, v_{i-1}, v_i)$ の最適値を $S_{i-1}(v_i, v_{i-1})$ とする。そうすると、$S_i(v_{i+1}, v_i)$ は式 (3.31) の導出と同様な考察から、

$$\begin{aligned} S_i(v_{i+1}, v_i) &= \min_{v_{i-1}} \{ S_{i-1}(v_i, v_{i-1}) + e_{i-1}(v_{i-1}, v_i, v_{i+1}) \} \\ &\text{for} \quad i \geq 2, \quad S_1 = 0 \end{aligned} \tag{3.33}$$

となる†。

従って、目的関数が

$$E_{open} = \sum_{i=1}^{n} f_i(v_i) + \sum_{i=1}^{n-1} g_i(v_i, v_{i+1}) + \sum_{i=1}^{n-2} h_i(v_i, v_{i+1}, v_{i+2}) \tag{3.34}$$

の場合††、式 (3.32) で $e_i(v_i, v_{i+1}, v_{i+2}) = f_i(v_i) + g_i(v_i, v_{i+1}) + h_i(v_i, v_{i+1}, v_{i+2})$ とおくことにより、式 (3.33) から次式の DP 方程式を導くことができる [40]。

$$\begin{aligned} S_i(v_{i+1}, v_i) &= \min_{v_{i-1}} \Big\{ S_{i-1}(v_i, v_{i-1}) + f_{i-1}(v_{i-1}) + g_{i-1}(v_{i-1}, v_i) + h_{i-1}(v_{i-1}, v_i, v_{i+1}) \Big\} \\ &\text{for} \quad i \geq 2, \quad S_1 = 0 \end{aligned} \tag{3.35}$$

但し、各 v_{i-1} の最適選択は、v_{i-1} の近傍で実行される。この近傍の数 (状態数) を一様に m とすると、式 (3.35) における v_{i-1} の最適選択は状態変数組 (v_{i+1}, v_i) の全ての組合せ (m^2) について各々

† Amini らはこれを"time-delayed dynamic programming" と呼んでいる。

†† この式は開輪郭の snake を表していることに注意。すなわち、Amini らは DP を適用するために、変数の順序関係を保持した開輪郭についてのみ定式化している。

実行され、記憶される。従って、アルゴリズムの計算の複雑さは、$m^2 \times m \times n$、すなわち、$\mathcal{O}(nm^3)$ となる。

(2) 問題点

以上述べた解法は開輪郭の場合であり、閉輪郭の場合、式 (3.29) に示したように、式 (3.34) を次式のように書きなおさなければならない。

$$E_{closed}(= E_{total}) = E_{open} + h_{n-1}(\boldsymbol{v}_{n-1}, \boldsymbol{v}_n, \boldsymbol{v}_1) + g_n(\boldsymbol{v}_n, \boldsymbol{v}_1) + h_n(\boldsymbol{v}_n, \boldsymbol{v}_1, \boldsymbol{v}_2) \tag{3.36}$$

ところが、式 (3.34) を式 (3.36) で置き換えると、$\boldsymbol{v}_i$ が巡回し、DP で最適解が求まるための必要条件であるマルコフ性が崩れてしまう。これを説明するために、次の 4 点から成る snake ($\{\boldsymbol{v}_i\}_{i=1}^4$) を考える。式 (3.29) に対応するエネルギー汎関数は次のようになる。

$$\begin{aligned} E(\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3, \boldsymbol{v}_4) = \Big(& (f_1(\boldsymbol{v}_1) + f_2(\boldsymbol{v}_2) + f_3(\boldsymbol{v}_3) + f_4(\boldsymbol{v}_4)) \\ & + (g_1(\boldsymbol{v}_1, \boldsymbol{v}_2) + g_2(\boldsymbol{v}_2, \boldsymbol{v}_3) + g_3(\boldsymbol{v}_3, \boldsymbol{v}_4) + g_4(\boldsymbol{v}_4, \boldsymbol{v}_1)) \\ & + (h_1(\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3) + h_2(\boldsymbol{v}_2, \boldsymbol{v}_3, \boldsymbol{v}_4) + h_3(\boldsymbol{v}_3, \boldsymbol{v}_4, \boldsymbol{v}_1) + h_4(\boldsymbol{v}_4, \boldsymbol{v}_1, \boldsymbol{v}_2)) \Big) \end{aligned} \tag{3.37}$$

式 (3.35) の漸化式に従って計算すると、まず、$S_1 = 0$ として、

$$S_2(\boldsymbol{v}_3, \boldsymbol{v}_2) = \min_{\boldsymbol{v}_1} \Big\{ f_1(\boldsymbol{v}_1) + g_1(\boldsymbol{v}_1, \boldsymbol{v}_2) + h_1(\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3) \Big\} \tag{3.38}$$

$$S_3(\boldsymbol{v}_4, \boldsymbol{v}_3) = \min_{\boldsymbol{v}_2} \Big\{ S_2(\boldsymbol{v}_3, \boldsymbol{v}_2) + f_2(\boldsymbol{v}_2) + g_2(\boldsymbol{v}_2, \boldsymbol{v}_3) + h_2(\boldsymbol{v}_2, \boldsymbol{v}_3, \boldsymbol{v}_4) \Big\} \tag{3.39}$$

と、ここまでは問題なく計算できる。すなわち、$\boldsymbol{v}_1$ の最適選択は $(\boldsymbol{v}_3, \boldsymbol{v}_2)$ の全ての組合せに対して求められ、記憶される。換言すれば、$\boldsymbol{v}_1$ は $\boldsymbol{v}_3, \boldsymbol{v}_2$ の関数とみなすことができる。同様に、$\boldsymbol{v}_2$ は $\boldsymbol{v}_4, \boldsymbol{v}_3$ の関数となる。

次に、$\boldsymbol{v}_3$ の最適選択は、式 (3.35) の漸化式に従えば、

$$S_4(\boldsymbol{v}_1, \boldsymbol{v}_4) = \min_{\boldsymbol{v}_3} \Big\{ S_3(\boldsymbol{v}_4, \boldsymbol{v}_3) + f_3(\boldsymbol{v}_3) + g_3(\boldsymbol{v}_3, \boldsymbol{v}_4) + h_3(\boldsymbol{v}_3, \boldsymbol{v}_4, \boldsymbol{v}_1) \Big\} \tag{3.40}$$

となる。ところが、式 (3.40) を見ると、$\boldsymbol{v}_3$ の決定は $\boldsymbol{v}_1, \boldsymbol{v}_4$ の関数となっている。すなわち、$\boldsymbol{v}_3$ の決定は、すでに決定された $\boldsymbol{v}_1$ にも依存することになり、DP が適用できるための必要条件であるマルコフ性が崩れる。言い替えれば、DP の基本原理である最適性の原理[†]に反するため、たとえ解が得られたとしてもその解の最適性は保証されない。

以上のことから、Amin の解法も開輪郭に対してのみ有効で閉輪郭には適用できないことが分かる。この原因は、閉輪郭にすることによる変数の巡回性 (順序性の崩壊) によるものである。但し、この問題の帰結として、DP がエネルギー最小化解法に適用不可能であるとするのは誤りであり、次節で述べる定式化によれば閉輪郭でも DP の枠組で最適解を得ることができる。

[†]最適政策は、"最初の状態と決定が何であっても、残りの決定は最初の決定から生じた状態に関して最適政策を構成しなければならない" という性質をもつ。

3.4 動的計画法に基づくエネルギー最小化解法の再定式化: 新解法

3.4.1 DP 方程式の導出

本節では、閉輪郭に対する最適解を与えるための解法について述べる。基本的には、3.3.2 で述べた動的計画法 (DP) に基づいているが、閉輪郭、すなわち変数が巡回しているような場合でも解の最適性を保証する DP 方程式 (漸化式) を導出するのが本節の目的である。解法の理解を容易にするために、3.3.2 と同様な論法を用いて漸化式 (DP 方程式) を導出する。

閉輪郭上の点集合を、$V = \{\boldsymbol{v}_i | i = 1, 2, \ldots, n\}$ とする。但し、閉輪郭故、$\boldsymbol{v}_{n+k} = \boldsymbol{v}_k$ $(k < n)$ が成り立つことに注意する。

(1) 目的関数が隣接 2 変数の関数の和で表される場合

目的関数が

$$\begin{aligned} E_c(\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n) &= \sum_{i=1}^{n} e_i(\boldsymbol{v}_i, \boldsymbol{v}_{i+1}) \\ &= e_1(\boldsymbol{v}_1, \boldsymbol{v}_2) + \cdots + e_{n-1}(\boldsymbol{v}_{n-1}, \boldsymbol{v}_n) + e_n(\boldsymbol{v}_n, \boldsymbol{v}_1) \end{aligned} \tag{3.41}$$

の形をしているものとする。このとき、$\boldsymbol{v}_1$ に着目すると、$\boldsymbol{v}_1$ は $e_1(\boldsymbol{v}_1, \boldsymbol{v}_2), e_n(\boldsymbol{v}_n, \boldsymbol{v}_1)$ にのみ依存することから、E_c の最小化は次のように書ける。

$$\begin{aligned} \min_V E_c &= \min_V \sum_i e_i(\boldsymbol{v}_i, \boldsymbol{v}_{i+1}) \\ &= \min_{V-\{\boldsymbol{v}_1\}} \min_{\boldsymbol{v}_1} \sum_i e_i(\boldsymbol{v}_i, \boldsymbol{v}_{i+1}) \\ &= \min_{V-\{\boldsymbol{v}_1\}} \Big\{ \sum_{i \neq 1,n} e_i(\boldsymbol{v}_i, \boldsymbol{v}_{i+1}) + \min_{\boldsymbol{v}_1} \{ e_1(\boldsymbol{v}_1, \boldsymbol{v}_2) + e_n(\boldsymbol{v}_n, \boldsymbol{v}_1) \} \Big\} \end{aligned} \tag{3.42}$$

ここで、まず右辺第二項の $\boldsymbol{v}_1$ に関する最小化を計算する。明らかに、この最小化は $\boldsymbol{v}_2$ と $\boldsymbol{v}_n$ の関数であるから、

$$\psi_1(\boldsymbol{v}_2, \boldsymbol{v}_n) = \min_{\boldsymbol{v}_1} \Big\{ e_1(\boldsymbol{v}_1, \boldsymbol{v}_2) + e_n(\boldsymbol{v}_n, \boldsymbol{v}_1) \Big\} \tag{3.43}$$

と書ける。すなわち、$\boldsymbol{v}_2$ と $\boldsymbol{v}_n$ のとり得る値の全ての組合せの各々に対して $\boldsymbol{v}_1$ の最適値 ($\boldsymbol{v}_1^*$ とする) を記憶しておく。この $\psi_1(\boldsymbol{v}_2, \boldsymbol{v}_n)$ を用いて原問題を書きかえると

$$\min_V E_c = \min_{V-\{\boldsymbol{v}_1\}} \Big\{ \sum_{i \neq 1,n} e_i(\boldsymbol{v}_i, \boldsymbol{v}_{i+1}) + \psi_1(\boldsymbol{v}_2, \boldsymbol{v}_n) \Big\} \tag{3.44}$$

となる。式 (3.44) の最小化問題は元の問題と同じ形をしており、関数 ψ_1 が新たな目的関数の要素とみなされる。

次に、$\boldsymbol{v}_2$ に着目すると、$\boldsymbol{v}_2$ は式 (3.44) において、$e_2(\boldsymbol{v}_2, \boldsymbol{v}_3)$ および、先に計算した $\psi_1(\boldsymbol{v}_2, \boldsymbol{v}_n)$ にのみ依存する。故に、

$$\begin{aligned}
\min_V E_c &= \min_{V-\{\boldsymbol{v}_1\}} \Big\{ \sum_{i \neq 1,n} e_i(\boldsymbol{v}_i, \boldsymbol{v}_{i+1}) + \psi_1(\boldsymbol{v}_2, \boldsymbol{v}_n) \Big\} \\
&= \min_{V-\{\boldsymbol{v}_1\}-\{\boldsymbol{v}_2\}} \min_{\boldsymbol{v}_2} \Big\{ \sum_{i \neq 1,n} e_i(\boldsymbol{v}_i, \boldsymbol{v}_{i+1}) + \psi_1(\boldsymbol{v}_2, \boldsymbol{v}_n) \Big\} \\
&= \min_{V-\{\boldsymbol{v}_1\}-\{\boldsymbol{v}_2\}} \Big\{ \sum_{i \neq 1,2,n} e_i(\boldsymbol{v}_i, \boldsymbol{v}_{i+1}) + \min_{\boldsymbol{v}_2} \{ e_2(\boldsymbol{v}_2, \boldsymbol{v}_3) + \psi_1(\boldsymbol{v}_2, \boldsymbol{v}_n) \} \Big\} \qquad (3.45)
\end{aligned}$$

となる。式 (3.45) の右辺第二項の $\boldsymbol{v}_2$ に関する最小化は $\boldsymbol{v}_3, \boldsymbol{v}_n$ の関数であるから、式 (3.44) と同様に次式のように書ける。

$$\psi_2(\boldsymbol{v}_3, \boldsymbol{v}_n) = \min_{\boldsymbol{v}_2} \Big\{ e_2(\boldsymbol{v}_2, \boldsymbol{v}_3) + \psi_1(\boldsymbol{v}_2, \boldsymbol{v}_n) \Big\} \qquad (3.46)$$

$\boldsymbol{v}_2$ の最適値 $\boldsymbol{v}_2^*$ も $\boldsymbol{v}_1^*$ と同様に、$\boldsymbol{v}_3, \boldsymbol{v}_n$ の全ての組合せの各々の組について計算され、記憶される。このとき、原問題は

$$\min_V E_c = \min_{V-\{\boldsymbol{v}_1\}-\{\boldsymbol{v}_2\}} \Big\{ \sum_{i \neq 1,2,n} e_i(\boldsymbol{v}_i, \boldsymbol{v}_{i+1}) + \psi_2(\boldsymbol{v}_3, \boldsymbol{v}_n) \Big\} \qquad (3.47)$$

となる。以下同様に $\boldsymbol{v}_3, \boldsymbol{v}_4, \ldots$ の順に最小化を実行していくと、$\boldsymbol{v}_i$ $(i \leq n-2)$ の最小化では次のようになる。

$$\begin{aligned}
\min_V E_c &= \min_{V-\{\boldsymbol{v}_1\}-\{\boldsymbol{v}_2\}-\cdots-\{\boldsymbol{v}_i\}} \min_{\boldsymbol{v}_i} \Big\{ \sum_{k \neq 1,2,\ldots,i,n} e_k(\boldsymbol{v}_k, \boldsymbol{v}_{k+1}) + \psi_{i-1}(\boldsymbol{v}_i, \boldsymbol{v}_n) \Big\} \\
&= \min_{V-\{\boldsymbol{v}_1\}-\cdots-\{\boldsymbol{v}_i\}} \Big\{ \sum_{k \neq 1,\ldots,i,n} e_k(\boldsymbol{v}_k, \boldsymbol{v}_{k+1}) + \min_{\boldsymbol{v}_i} \{ e_i(\boldsymbol{v}_i, \boldsymbol{v}_{i+1}) + \psi_{i-1}(\boldsymbol{v}_i, \boldsymbol{v}_n) \} \Big\} \qquad (3.48)
\end{aligned}$$

従って、ψ_i と ψ_{i-1} $(i \leq n-2)$ との間には次のような関係が成り立つことが分かる。

$$\psi_i(\boldsymbol{v}_{i+1}, \boldsymbol{v}_n) = \min_{\boldsymbol{v}_i} \Big\{ e_i(\boldsymbol{v}_i, \boldsymbol{v}_{i+1}) + \psi_{i-1}(\boldsymbol{v}_i, \boldsymbol{v}_n) \Big\} \quad \text{for} \quad 2 \leq i \leq n-2 \qquad (3.49)$$

また、$\boldsymbol{v}_{n-1}$ に関する最小化は次のようになる。

$$\begin{aligned}
\min_V E_c &= \min_{V-\{\boldsymbol{v}_1\}-\{\boldsymbol{v}_2\}-\cdots-\{\boldsymbol{v}_{n-1}\}} \min_{\boldsymbol{v}_{n-1}} \Big\{ e_{n-1}(\boldsymbol{v}_{n-1}, \boldsymbol{v}_n) + \psi_{n-2}(\boldsymbol{v}_{n-1}, \boldsymbol{v}_n) \Big\} \\
&= \min_{\boldsymbol{v}_n} \min_{\boldsymbol{v}_{n-1}} \Big\{ e_{n-1}(\boldsymbol{v}_{n-1}, \boldsymbol{v}_n) + \psi_{n-2}(\boldsymbol{v}_{n-1}, \boldsymbol{v}_n) \Big\} \\
&= \min_{\boldsymbol{v}_n} \Big\{ \psi_{n-1}(\boldsymbol{v}_n) \Big\} \qquad (3.50)
\end{aligned}$$

故に、$i = n-1$ については次の関係が成り立つ。

$$\psi_{n-1}(\boldsymbol{v}_n) = \min_{\boldsymbol{v}_{n-1}} \Big\{ e_{n-1}(\boldsymbol{v}_{n-1}, \boldsymbol{v}_n) + \psi_{n-2}(\boldsymbol{v}_{n-1}, \boldsymbol{v}_n) \Big\} \qquad (3.51)$$

式 (3.49),(3.51) により所望の DP 方程式が導出できたわけである。従って、E_c の最小値は、上記漸化式を逐次計算し、最終的に

$$\min_{\boldsymbol{v}_n} \Big\{ \psi_{n-1}(\boldsymbol{v}_n) \Big\} \qquad (3.52)$$

を求めることにより得られる。さらに、この最小値を与える最適解 $\{\boldsymbol{v}_i^*\}_{i=1}^n$ は次のような手順により容易に得られる。

式 (3.52) において ψ_{n-1} を最小化する $\boldsymbol{v}_n^*$ が一意に定まる。一方、$\boldsymbol{v}_{n-1}$ の最適値は式 (3.51) の計算の際に $\boldsymbol{v}_n$ の関数として記憶されているので、$\boldsymbol{v}_n^*$ が一意に決定されれば、$\boldsymbol{v}_{n-1}^*$ も一意に決定されることになる。同様に、$\boldsymbol{v}_{n-1}^*, \boldsymbol{v}_n^*$ が決定すれば、式 (3.49) の計算の際に、すでに求めておいた $\psi_{n-2}(\boldsymbol{v}_{n-1}, \boldsymbol{v}_n)$ から $\boldsymbol{v}_{n-2}^*$ が決定される。以下同様なトレースバックを実行することにより、最適解 $\{\boldsymbol{v}_i^*\}_{i=1}^n$ が得られることになる。

(2) 目的関数が隣接 3 変数の関数の和で表される場合

次に、目的関数が隣接 3 変数の和、すなわち、

$$\begin{aligned}
E_c(\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n) &= \sum_{i=1}^{n} e_i(\boldsymbol{v}_i, \boldsymbol{v}_{i+1}, \boldsymbol{v}_{i+2}) \\
&= e_1(\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3) + e_2(\boldsymbol{v}_2, \boldsymbol{v}_3, \boldsymbol{v}_4) + \cdots \\
&\quad \cdots + e_{n-1}(\boldsymbol{v}_{n-1}, \boldsymbol{v}_n, \boldsymbol{v}_1) + e_n(\boldsymbol{v}_n, \boldsymbol{v}_1, \boldsymbol{v}_2) \qquad (3.53)
\end{aligned}$$

で表される場合について考えよう。基本的には、(1) と同様な導出手順となる。

すなわち、まず $\boldsymbol{v}_1$ に着目する。$\boldsymbol{v}_1$ に依存する関数群は、$e_1(\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3), e_{n-1}(\boldsymbol{v}_{n-1}, \boldsymbol{v}_n, \boldsymbol{v}_1)$ および、$e_n(\boldsymbol{v}_n, \boldsymbol{v}_1, \boldsymbol{v}_2)$ であるから、E_c の最小化は次のように書ける。

$$\begin{aligned}
\min_V E_c &= \min_{V-\{\boldsymbol{v}_1\}} \min_{\boldsymbol{v}_1} \sum_i e_i(\boldsymbol{v}_i, \boldsymbol{v}_{i+1}, \boldsymbol{v}_{i+2}) \\
&= \min_{V-\{\boldsymbol{v}_1\}} \Big\{ \sum_{i \neq 1,n-1,n} e_i(\boldsymbol{v}_i, \boldsymbol{v}_{i+1}, \boldsymbol{v}_{i+2}) \\
&\qquad + \min_{\boldsymbol{v}_1} \{ e_1(\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3) + e_{n-1}(\boldsymbol{v}_{n-1}, \boldsymbol{v}_n, \boldsymbol{v}_1) + e_n(\boldsymbol{v}_n, \boldsymbol{v}_1, \boldsymbol{v}_2) \} \Big\} \qquad (3.54)
\end{aligned}$$

式 (3.54) の右辺第二項の $\boldsymbol{v}_1$ に関する最小化は $\boldsymbol{v}_2, \boldsymbol{v}_3, \boldsymbol{v}_{n-1}, \boldsymbol{v}_n$ の関数となるから、

$$\psi_1(\boldsymbol{v}_2, \boldsymbol{v}_3, \boldsymbol{v}_{n-1}, \boldsymbol{v}_n) = \min_{\boldsymbol{v}_1} \Big\{ e_1(\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3) + e_{n-1}(\boldsymbol{v}_{n-1}, \boldsymbol{v}_n, \boldsymbol{v}_1) + e_n(\boldsymbol{v}_n, \boldsymbol{v}_1, \boldsymbol{v}_2) \Big\} \qquad (3.55)$$

と書ける。従って、原問題を次式のように書きかえることができる。

$$\min_V E_c = \min_{V-\{\boldsymbol{v}_1\}} \Big\{ \sum_{i \neq 1,n-1,n} e_i(\boldsymbol{v}_i, \boldsymbol{v}_{i+1}, \boldsymbol{v}_{i+2}) + \psi_1(\boldsymbol{v}_2, \boldsymbol{v}_3, \boldsymbol{v}_{n-1}, \boldsymbol{v}_n) \Big\} \qquad (3.56)$$

以下同様な手順により、v_i $(i \le n-4)$ に関する最小化では、

$$\begin{aligned}\min_V E_c &= \min_{V-\{v_1\}-\cdots-\{v_i\}} \min_{v_i}\Big\{\sum_{k\neq 1,2,\ldots,i,n-1,n} e_k(v_k,v_{k+1},v_{k+2}) + \psi_{i-1}(v_i,v_{i+1},v_{n-1},v_n)\Big\}\\ &= \min_{V-\{v_1\}-\cdots-\{v_i\}}\Big\{\sum_{k\neq 1,\ldots,i,n-1,n} e_k(v_k,v_{k+1},v_{k+2})\\ &\quad + \min_{v_i}\{e_i(v_i,v_{i+1},v_{i+2}) + \psi_{i-1}(v_i,v_{i+1},v_{n-1},v_n)\}\Big\} \qquad (3.57)\end{aligned}$$

となることが容易に確かめられる。式 (3.57) の右辺の内側の最小化計算 (v_i に関する最小化) は、$v_{i+1}, v_{i+2}, v_{n-1}, v_n$ の関数となることから、次の所望の DP 方程式が得られる。

$$\begin{aligned}\psi_i(v_{i+1},v_{i+2},v_{n-1},v_n) &= \min_{v_i}\Big\{e_i(v_i,v_{i+1},v_{i+2}) + \psi_{i-1}(v_i,v_{i+1},v_{n-1},v_n)\Big\}\\ &\qquad \text{for} \quad 2 \le i \le n-4 \qquad (3.58)\end{aligned}$$

また、$v_{n-3}, v_{n-2}, v_{n-1}$ に関する最小化では、各々、

$$\begin{aligned}\min_V E_c &= \min_{v_{n-2},v_{n-1},v_n}\Big\{e_{n-2}(v_{n-2},v_{n-1},v_n) + \min_{v_{n-3}}\{e_{n-3}(v_{n-3},v_{n-2},v_{n-1})\\ &\quad + \psi_{n-4}(v_{n-3},v_{n-2},v_{n-1},v_n)\}\Big\} \qquad (3.59)\end{aligned}$$

$$\min_V E_c = \min_{v_{n-1},v_n}\Big\{\min_{v_{n-2}}\{e_{n-2}(v_{n-2},v_{n-1},v_n) + \psi_{n-3}(v_{n-2},v_{n-1},v_n)\}\Big\} \qquad (3.60)$$

$$\min_V E_c = \min_{v_n}\Big\{\min_{v_{n-1}}\{\psi_{n-2}(v_{n-1},v_n)\}\Big\} \qquad (3.61)$$

となるので、同様な考察により、$i = n-3, n-2, n-1$ に対する DP 方程式は各々以下のようになる。

$$\begin{cases}\psi_{n-3}(v_{n-2},v_{n-1},v_n) = \min_{v_{n-3}}\Big\{e_{n-3}(v_{n-3},v_{n-2},v_{n-1}) + \psi_{n-4}(v_{n-3},v_{n-2},v_{n-1},v_n)\Big\}\\ \psi_{n-2}(v_{n-1},v_n) = \min_{v_{n-2}}\Big\{e_{n-2}(v_{n-2},v_{n-1},v_n) + \psi_{n-3}(v_{n-2},v_{n-1},v_n)\Big\}\\ \psi_{n-1}(v_n) = \min_{v_{n-1}}\Big\{\psi_{n-2}(v_{n-1},v_n)\Big\}\end{cases} \qquad (3.62)$$

故に、E_c の最小値は、上記 DP 方程式を用いて、$\psi_2, \psi_3, \ldots, \psi_{n-1}$ を逐次計算すると、最終的には、

$$\min_V E_c = \min_{v_n}\Big\{\psi_{n-1}(v_n)\Big\} \qquad (3.63)$$

となるので、結局、$\psi_{n-1}(v_n)$ の最小値を求めれば良い。そして、その最小値を与える v_n^* の値を基に、(1) で述べたようなトレースバックを実行することにより残りの最適解 $v_{n-1}, v_{n-2}, \ldots, v_1$ を得ることができる。

以上、(1),(2) における DP 方程式の導出過程から容易に推察できるように、上記解法は目的関数が (1),(2) の場合に限定されるわけではなく、任意の変数の組合わせ (重複も含む) から成る関数群で

構成されるような一般の目的関数にも拡張可能である。より一般化した議論は後述する (3.4.2) として、まずその具体例として式 (3.29) に対する DP 方程式の導出を試みる。

(3) 式 (3.29) の目的関数に対する DP 方程式の導出

まず、式 (3.29) の目的関数を再掲する。

$$E_{total} = \sum_{i=1}^{n}\Big\{f_i(v_i) + g_i(v_i,v_{i+1}) + h_i(v_i,v_{i+1},v_{i+2})\Big\} \qquad (3.64)$$

v_1 に着目すると、v_1 は $f_1(v_1), g_1(v_1,v_2), h_1(v_1,v_2,v_3), h_{n-1}(v_{n-1},v_n,v_1), g_n(v_n,v_1)$, $h_n(v_n,v_1,v_2)$ にのみ依存する。そこで、これらの関数群の和を便宜上、

$$\begin{aligned}S &= f_1(v_1) + g_1(v_1,v_2) + h_1(v_1,v_2,v_3) + h_{n-1}(v_{n-1},v_n,v_1)\\ &\quad + g_n(v_n,v_1) + h_n(v_n,v_1,v_2) \qquad (3.65)\end{aligned}$$

とおくと、E_{total} の最小化は次のように書ける。

$$\begin{aligned}\min_V E_{total} &= \min_{V-\{v_1\}} \min_{v_1} \sum_i E_{total}\\ &= \min_{V-\{v_1\}}\Big\{(E_{total} - S) + \min_{v_1}\{S\}\Big\} \qquad (3.66)\end{aligned}$$

式 (3.66) の v_1 に関する最小化は式 (3.65) から v_2, v_3, v_{n-1}, v_n の関数となることがわかる。故に、

$$\psi_1(v_2,v_3,v_{n-1},v_n) = \min_{v_1}\{S\} \qquad (3.67)$$

とおける。すると、原問題は、

$$\min_V E_{total} = \min_{V-\{v_1\}}\Big\{(E_{total} - S) + \psi_1(v_2,v_3,v_{n-1},v_n)\Big\} \qquad (3.68)$$

と書きかえられる。明らかに、上式は元の問題と同じ形をしており、関数 ψ_1 が新たな目的関数の要素と見なされる。

次に、v_2 についての最小化を考える。式 (3.68) で v_2 に依存する関数は, $f_2(v_2), g_2(v_2,v_3)$, $h_2(v_2,v_3,v_4)$、および $\psi_1(v_2,v_3,v_{n-1},v_n)$ であるから、式 (3.68) は次式のように変形できる。

$$\begin{aligned}\min_V E_{total} &= \min_{V-\{v_1\}-\{v_2\}} \min_{v_2}\Big\{(E_{total} - S) + \psi_1(v_2,v_3,v_{n-1},v_n)\Big\}\\ &= \min_{V-\{v_1\}-\{v_2\}}\Big\{\Big(E_{total} - S - f_2(v_2) - g_2(v_2,v_3) - h_2(v_2,v_3,v_4)\Big)\\ &\quad + \min_{v_2}\{f_2(v_2) + g_2(v_2,v_3) + h_2(v_2,v_3,v_4) + \psi_1(v_2,v_3,v_{n-1},v_n)\}\Big\} \qquad (3.69)\end{aligned}$$

式 (3.69) の v_2 に関する最小化は、v_3, v_4, v_{n-1}, v_n の関数であるから、直ちに次式が得られる。

$$\psi_2(v_3,v_4,v_{n-1},v_n) = \min_{v_2}\Big\{\psi_1(v_2,v_3,v_{n-1},v_n) + f_2(v_2) + g_2(v_2,v_3) + h_2(v_2,v_3,v_4)\Big\} \qquad (3.70)$$

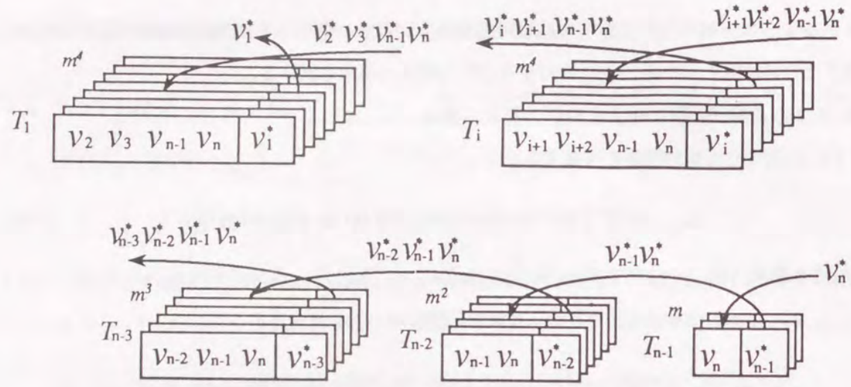


図 3.4: 最適選択値を記憶しておくための DP テーブル

以下同様な最小化を実行することにより、一般に次式の DP 方程式を導くことができる。

$$\begin{aligned}\psi_i(\boldsymbol{v}_{i+1},\boldsymbol{v}_{i+2},\boldsymbol{v}_{n-1},\boldsymbol{v}_n) &= \min_{\boldsymbol{v}_i}\Big\{\psi_{i-1}(\boldsymbol{v}_i,\boldsymbol{v}_{i+1},\boldsymbol{v}_{n-1},\boldsymbol{v}_n)\\ &\qquad +f_i(\boldsymbol{v}_i)+g_i(\boldsymbol{v}_i,\boldsymbol{v}_{i+1})+h_i(\boldsymbol{v}_i,\boldsymbol{v}_{i+1},\boldsymbol{v}_{i+2})\Big\}\\ &\qquad \text{for}\quad 2\le i\le n-4 \end{aligned} \tag{3.71}$$

但し、$i=n-3,n-2,n-1$ のときは以下の式に従う。

$$\begin{cases}\psi_{n-3}(\boldsymbol{v}_{n-2},\boldsymbol{v}_{n-1},\boldsymbol{v}_n)=\min_{\boldsymbol{v}_{n-3}}\Big\{\psi_{n-4}(\boldsymbol{v}_{n-3},\boldsymbol{v}_{n-2},\boldsymbol{v}_{n-1},\boldsymbol{v}_n)\\ \qquad +f_{n-3}(\boldsymbol{v}_{n-3})+g_{n-3}(\boldsymbol{v}_{n-3},\boldsymbol{v}_{n-2})+h_{n-3}(\boldsymbol{v}_{n-3},\boldsymbol{v}_{n-2},\boldsymbol{v}_{n-1})\Big\}\\ \psi_{n-2}(\boldsymbol{v}_{n-1},\boldsymbol{v}_n)=\min_{\boldsymbol{v}_{n-2}}\Big\{\psi_{n-3}(\boldsymbol{v}_{n-2},\boldsymbol{v}_{n-1},\boldsymbol{v}_n)\\ \qquad +f_{n-2}(\boldsymbol{v}_{n-2})+g_{n-2}(\boldsymbol{v}_{n-2},\boldsymbol{v}_{n-1})+h_{n-2}(\boldsymbol{v}_{n-2},\boldsymbol{v}_{n-1},\boldsymbol{v}_n)\Big\}\\ \psi_{n-1}(\boldsymbol{v}_n)=\min_{\boldsymbol{v}_{n-1}}\Big\{\psi_{n-2}(\boldsymbol{v}_{n-1},\boldsymbol{v}_n)+f_{n-1}(\boldsymbol{v}_{n-1})+g_{n-1}(\boldsymbol{v}_{n-1},\boldsymbol{v}_n)\Big\}\end{cases} \tag{3.72}$$

ここに、$\psi_0=0$ としておく。

式 (3.71),(3.72) の実行時に、$\boldsymbol{v}_i, i=1,2,\ldots,n-1$ の最適選択を各々、図 3.4 に示すテーブル $T_1,T_2,\ldots,T_{n-1}$ に記憶しておく。すなわち、$\boldsymbol{v}_i$ の最適選択は $\boldsymbol{v}_{i+1},\boldsymbol{v}_{i+2},\boldsymbol{v}_{n-1},\boldsymbol{v}_n$ の関数としてテーブル T_i に記憶される。上記 DP 方程式 (式 (3.71),(3.72)) から、$\boldsymbol{v}_i$ の最適決定が $\boldsymbol{v}_{i'}(i'<i)$ の最適決定に依存していないことが確認できる。換言すれば、$\boldsymbol{v}_i$ の最適選択は $\boldsymbol{v}_{i'}(i'<i)$ の最適選択に独立に実行でき、上記 DP 方程式が前述したマルコフ性 (DP が適用できるための必要条件) を満たしていること意味する。従って、解の最適性が保証される。

さて、$f_n(\boldsymbol{v}_n)$ のみが上記 DP 方程式で評価されずに残されており、かつ f_n は $\boldsymbol{v}_n$ のみの関数であるから、結局、残された最後の変数 $\boldsymbol{v}_n$ に関する最小化が E_{total} の最小化となる。すなわち、

$$\min_{V} E_{total}=\min_{\{\boldsymbol{v}_n\}}\Big\{\psi_{n-1}(\boldsymbol{v}_n)+f_n(\boldsymbol{v}_n)\Big\} \tag{3.73}$$

式 (3.73) の最小化は、式 (3.71),(3.72) と異なり、$\boldsymbol{v}_n$ のみにより実行できる。すなわち、その最小値を与える $\boldsymbol{v}_n{}^*$ が一意に決定される。$\boldsymbol{v}_n{}^*$ が一意に決定されれば、図 3.4 の矢印に示すように、その $\boldsymbol{v}_n{}^*$ の値でテーブル $T_{n-1}(\boldsymbol{v}_n \mid \boldsymbol{v}_{n-1}{}^*)$ を参照することにより、$\boldsymbol{v}_{n-1}{}^*$ を一意に決定できる。さらに、決定された $\boldsymbol{v}_{n-1}{}^*$、$\boldsymbol{v}_n{}^*$ の値でテーブル $T_{n-2}(\boldsymbol{v}_{n-1},\boldsymbol{v}_n \mid \boldsymbol{v}_{n-2}{}^*)$ を参照することにより、$\boldsymbol{v}_{n-2}{}^*$ を得る。以下、同様なトレースバックを実行することにより、残りの最適値 $\boldsymbol{v}_{n-3}{}^*,\boldsymbol{v}_{n-4}{}^*,\ldots,\boldsymbol{v}_1{}^*$ が得られる。

上記最小化手続きは、Amini のそれと同様に、全エネルギー E_{total} の減少がなくなるまで繰り返し実行される。すなわち、この繰り返し過程は以下のように整理される。

(1) 全エネルギー E_{total} を式 (3.71),(3.72),(3.73) により計算する。

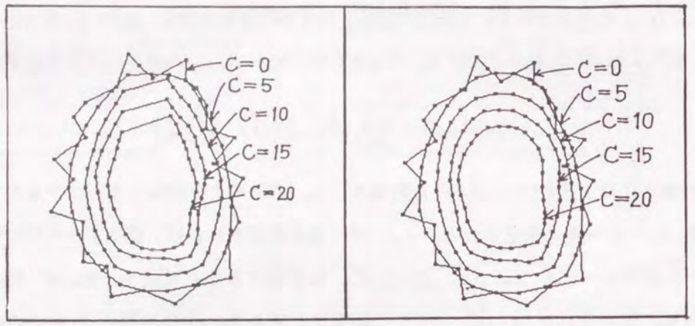
(2) もし、そのエネルギー値が前の繰り返しでの値より大きければ終了、さもなくば (3) へ。

(3) (1) で得られた最適値 $\boldsymbol{v}_i{}^*, i=1,2,\ldots,n$ に従って輪郭を移動させ、(1) へ。

上記繰り返し過程が輪郭モデルの動的な振舞いに相当する。

次に、得られた DP 方程式の計算量について考察する。式 (3.71) では、m^4 の状態変数の組合せに対して、それぞれ最適な v_i^* を算出するわけであるから、計算量は $m^4\times m=m^5$ となる。また、式 (3.72) の各式では、同様な考察により各々 m^4,m^3,m^2 となる。従って、総計算量はこれらの総和であるから $\mathcal{O}(nm^5)$ となる。3.3.2 で述べた Amini の解法に比べ、本解法は m^2 倍の計算量を要するが、DP では各ステップでの最適選択の計算を独立して実行できるため、並列処理により大幅な計算量の削減が可能である。実際、本解法において m^4 個のプロセッサを用い、$\psi_i(\boldsymbol{v}_i,\boldsymbol{v}_{i+1},\boldsymbol{v}_{n-1},\boldsymbol{v}_n)$ の計算を並列実行することにより、総計算量を $\mathcal{O}(mn)$ にまで削減できる。

以上により、DP の枠組で閉輪郭に対しても最適解を得ることができることを理論的に示した。



(a)Amini の解法　　(b) 提案解法

図 3.5: 内部エネルギーのみによる比較実験 (C は繰り返し数)

3.4.2 比較実験

ここでは、実験により提案解法と Amini の解法を比較する[†]。まず、厳密な比較を行なうために、Snakes の内部エネルギーのみに関するエネルギー最小化の比較実験を行なった。すなわち、適当な初期値 (初期輪郭) から内部エネルギーの最小化を実行することにより、輪郭がどのように変化するかを比較した。画像は存在しないので、外部エネルギーは一切無関係である。従って、3.2 で述べたように、閉輪郭の場合、snake は収縮しながら円形状となっていくはずである。ところが、図 3.5(a) に示す Amini の解法では、閉輪郭として定式化されていないため、始点と終点の不連続性が次第に強調されるという不都合が生じている。一方、図 3.5(b) の提案解法では snake の本来の振舞いが確認できる。

次に、外部エネルギーも考慮した比較実験を行なった。図 3.6 にその実験結果を示す。同図はコンピュータグラフィックスにより生成した球 (CG 球) に対して Amini の解法 (図 3.6(a)) と、提案解法 (図 3.6(b)) とによる輪郭抽出結果である。図で I は繰り返し回数を表す。画像のサイズは 256 × 256 画素、256 階調画素である。但し、同図では紙面の都合上画像の周辺部をカットしている。

Amini の解法では、開輪郭として定式化されているため、snake は対象輪郭にフィットしても、収束せずにさらに輪郭上で収縮してしまう。それ故、始点と終点が対象輪郭上で次第に離れていく。Amini らは、この問題に対して、すべての隣接 2 点間の距離が必ずある値より大きくなるようにし、Snakes の過剰な収縮を抑制している [40]。一方、本解法ではそうしたヒューリスティックを用いることなく良好な結果が得られており、提案解法の有効性を実験的にも確認することができた。

[†]変分法については、すでに Amini の論文 [40] で実験比較されているので省略する。

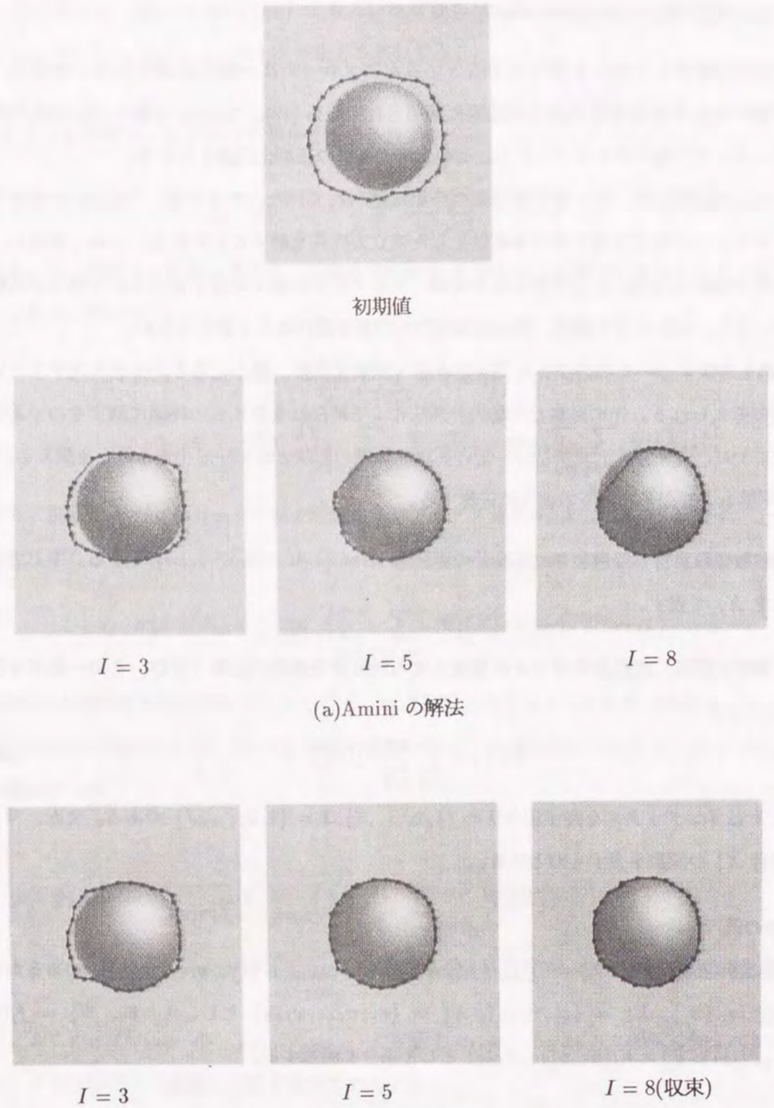


図 3.6: 輪郭抽出比較実験

3.4.3 正則化問題の統一的解法への一般化

様々な情報を統合しそれらを最小化するといったアプローチは一般に正則化技法と呼ばれ、近年、画像理解における初期視覚の多くの問題に適用されている [42]。Snakes も種々の汎関数の和で表現されたエネルギー最小化アプローチで、その意味で一種の正則化問題といえる。

こうした正則化問題は、これまで変分法の枠組で解かれていた。すなわち、Snakes の解法のときと同様、オイラー方程式を導きそれを離散化した連立方程式を解くことになる。しかしながら、オイラー方程式は偏微分方程式として得られるため、ディジタル画像で実現するためには何らかの離散近似が必要となり、それに伴う誤差、解の収束性等の問題を避けることができない。

この問題を解決する一つの方法としてエネルギー関数を直接、最小化あるいは最大化するという解法の枠組が考えられる。先に提案した動的計画法による解法はまさにこの枠組に属すものである。そこで、ここでは、先に提案した解法を一般の正則化問題の解法として一般化することを試みる。

まず、用語および記号を以下のように定義する。

定義 3.1 (離散変数集合) 離散化された変数の集合を $X = \{x_i | i = 1, 2, \ldots, n\}$ とする。そして、その部分集合を X_j で表す。

定義 3.2 (離散汎関数) 離散変数集合を変換数とする汎関数を離散汎関数と呼び、その一般形を次式で与える。

$$\Psi(X) = \sum_{i \in \mathbf{I}} \sum_{j \in \mathbf{J}} \Psi_j^i(X_j^i) \tag{3.74}$$

ここに、$\mathbf{I}, \mathbf{J}$ はインデックスを表す集合:$\mathbf{I} = \{1, 2, \ldots, I\}, \mathbf{J} = \{1, 2, \ldots, J\}$ である。また、Ψ_j^i はある変数集合 X_j^i の関数を表すものとする。

[離散汎関数の例] Snakes:

Snakes の離散汎関数は、$E(V) = \sum_{i=1}^{n} \{f_i(\boldsymbol{v}_i) + g_i(\boldsymbol{v}_i, \boldsymbol{v}_{i+1}) + h(\boldsymbol{v}_i, \boldsymbol{v}_{i+1}, \boldsymbol{v}_{i+2})\}$ であるから、これは、$X_j^1 = \{\boldsymbol{v}_j\}$, $X_j^2 = \{\boldsymbol{v}_j, \boldsymbol{v}_{j+1}\}$, $X_j^3 = \{\boldsymbol{v}_j, \boldsymbol{v}_{j+1}, \boldsymbol{v}_{j+2}\}$ とし、さらに、$\Psi_j^1 = f_j(\boldsymbol{v}_j)$, $\Psi_j^2 = g_j(\boldsymbol{v}_j, \boldsymbol{v}_{j+1})$, $\Psi_j^3 = h_j(\boldsymbol{v}_j, \boldsymbol{v}_{j+1}, \boldsymbol{v}_{j+2})$ とした場合に相当する。

定義 3.3 (相互作用) $\boldsymbol{x}_s \in X, \boldsymbol{x}_t \in X$ がともに X_j^i に属すような関数 $\Psi_j^i(X_j^i)$ が存在するとき、$\boldsymbol{x}_s \in X$ は $\boldsymbol{x}_t \in X$ と (あるいはその逆も含めて)"相互作用する"と言い、$Int(\boldsymbol{x}_i)$ で表す。

[相互作用の例] Snakes:

再び、Snakes を例にとると、例えば、$\boldsymbol{v}_1$ は $\boldsymbol{v}_2, \boldsymbol{v}_3, \boldsymbol{v}_{n-1}$ および $\boldsymbol{v}_n$ と相互作用している。すなわち、$Int(\boldsymbol{v}_1) = \{\boldsymbol{v}_2, \boldsymbol{v}_3, \boldsymbol{v}_{n-1}, \boldsymbol{v}_n\}$ となる。なぜなら、$g_1(\boldsymbol{v}_1, \boldsymbol{v}_2)$, $g_n(\boldsymbol{v}_n, \boldsymbol{v}_1)$, $h_1(\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3)$, $h_{n-1}(\boldsymbol{v}_{n-1}, \boldsymbol{v}_n, \boldsymbol{v}_1)$, $h_n(\boldsymbol{v}_n, \boldsymbol{v}_1, \boldsymbol{v}_2)$ が存在するからである。

求めるべき問題は、式 (3.74) の最小化であるから、

$$\min_X \Psi(X) = \min_X \Big\{ \sum_{i \in \mathbf{I}} \sum_{j \in \mathbf{J}} \Psi_j^i(X_j^i) \Big\} \tag{3.75}$$

と書ける。今、変数 $\boldsymbol{x}_1$ に着目すると、関数群は変数 $\boldsymbol{x} \in Int(\boldsymbol{x}_1)$ に依存するものとそうでないものに二分される。従って、

$$\begin{aligned} \min_X \Psi(X) &= \min_{X - \{\boldsymbol{x}_1\}} \Big\{ \min_{\boldsymbol{x}_1} \{ \sum_{i \in \mathbf{I}} \sum_{j \in \mathbf{J}} \Psi_j^i(X_j^i) \} \Big\} \\ &= \min_{X - \{\boldsymbol{x}_1\}} \Big\{ \sum_{i \in \mathbf{I}} \sum_{j \in \mathbf{J} - \mathbf{J}_1} \Psi_j^i(X_j^i) + \min_{\boldsymbol{x}_1} \{ \sum_{i \in \mathbf{I}} \sum_{j \in \mathbf{J}_1} \Psi_j^i(X_j^i) \} \Big\} \end{aligned} \tag{3.76}$$

と書ける。但し、$\mathbf{J}_1$ は $Int(\boldsymbol{x}_1)$ に属す変数のインデックス集合:$\{j | X_j^i \cap \{\boldsymbol{x}_1\}\}$ を表す。上式の内側の最小化は明らかに $Int(\boldsymbol{x}_1)$ の関数であるから、原問題は次のように書きなおすことができる。

$$\min_X \Psi(X) = \min_{X - \{\boldsymbol{x}_1\}} \Big\{ \sum_{i \in \mathbf{I}} \sum_{j \in \mathbf{J} - \mathbf{J}_1} \Psi_j^i(X_j^i) + \psi_1\Big(Int(\boldsymbol{x}_1)\Big) \Big\} \tag{3.77}$$

$\boldsymbol{x}_1$ を削除した後の残された問題 (式 (3.77)) は、元の問題と同じ形をしており、関数 $\psi_1\Big(Int(\boldsymbol{x}_1)\Big)$ が新たな目的関数の要素となる。従って、同様に変数 $\boldsymbol{x}_2, \boldsymbol{x}_3, \ldots$ を削除していくと、第 k ステップでの変数の削除後では、

$$\begin{aligned} \min_X \Psi(X) = \min_{X - \{\boldsymbol{x}_1\} - \cdots - \{\boldsymbol{x}_k\}} \Big\{ & \sum_{i \in \mathbf{I}} \sum_{j \in \mathbf{J} - \mathbf{J}_1 - \cdots - \mathbf{J}_k} \Psi_j^i(X_j^i) \\ & + \min_{\boldsymbol{x}_k} \{ \sum_{i \in \mathbf{I}} \sum_{j \in \mathbf{J}_k} \Psi_j^i(X_j^i) + \psi_{k-1}(Int(\boldsymbol{x}_{k-1} | \boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{k-2})) \} \Big\} \end{aligned} \tag{3.78}$$

となる。ここに、$Int(\boldsymbol{x}_{k-1} | \boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{k-2})$ は変数を $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{k-2}$ の順に削除していった後での $\boldsymbol{x}_{k-1}$ と相互作用する変数の集合を表すものとする。

従って、次の DP 方程式が得られることになる。

$$\psi_i\Big(Int(\boldsymbol{x}_k | \boldsymbol{x}_1, \ldots, \boldsymbol{x}_{k-1})\Big) = \min_{\boldsymbol{x}_k} \Big\{ \sum_{i \in \mathbf{I}} \sum_{j \in \mathbf{J}_k} \Psi_j^i(X_j^i) + \psi_{k-1}\Big(Int(\boldsymbol{x}_{k-1} | \boldsymbol{x}_1, \ldots, \boldsymbol{x}_{k-2})\Big) \Big\} \tag{3.79}$$

上記 DP 方程式を逐次計算する過程で、最適選択 $\boldsymbol{x}_k{}^*$ が $Int(\boldsymbol{x}_k | \boldsymbol{x}_1, \ldots, \boldsymbol{x}_{k-1})$ の関数としてテーブル $T_k\Big(Int(\boldsymbol{x}_k | \boldsymbol{x}_1, \ldots, \boldsymbol{x}_{k-1}) | \boldsymbol{x}_k{}^*\Big)$ の形で記憶される。第 n ステップでは、$Int(\boldsymbol{x}_n | \boldsymbol{x}_1, \ldots, \boldsymbol{x}_{n-1}) =$

ϕ となるから、x_n の削除により最小値 (最適値) が得られることになる。そして、その最適値を与える最適解 $\{x_i^*\}_{i=1}^n$ は先に記憶しておいたテーブルをトレースバックすることにより容易に得られる。

以上により、式 (3.75) の最小化問題は DP の枠組で理論的に解けたことになる。但し、相互作用変数の数が大きい問題に上記解法を適用する場合、計算量の問題が生じる。従って、そした問題では何らかの近似による準最適化が余儀なくされる。

3.5 動輪郭追跡における Snakes の問題点とその解決法

3.5.1 追跡処理への適用とその問題点

動物体追跡処理は動画像解析の最も基本的な処理であり、これまで多くの研究がなされている。そこで、まず、Snakes による追跡手法の問題点を述べる前に、従来手法を整理した上で、Snakes による追跡方式の長所およびその適用限界を述べる。

動画像解析における物体の運動・構造の復元問題ではそれに先だって画素・特徴点・線分等の対応づけ (追跡) が完了していることを前提としており、その意味で動物体の追跡は非常に重要な問題である。この対応づけの問題は一般に Correspondence Problem(CP) と呼ばれ、これまで多くの研究がなされており、古くは Ullman の極小写像理論 [44] にまで遡る。CP は画素の輝度の差分、相関に基づく方法と特徴点・線分等のトークンのマッチングに基づく方法に大別できる。前者の代表例として、時空間微分法によるオプティカルフローの算出 [45] がある。前者に比べ後者の方法は、一般に照明条件の変動に強く、また数画素以上の変移量を有するロングレンジの動画にも適用できるという点で実用的な手法といえる。

トークンの代表的なものとして、特徴点、線分、角等が挙げられる。Barnard ら [46] は弛緩法を用いた特徴点対応づけ手法を提案している。また、最近、Sethi ら [47] は複数フレームにわたる特徴点軌跡の滑らかさ (motion coherence) を対応の評価基準とする対応づけ手法を提案している。そしてこの手法は Salari ら [48]、さらに Cheng ら [49] によって改良が加えられている。

こうした特徴点の追跡は運動の定量的解析には有用であるが、有意な対象物体そのものを追跡するような応用においてはより高レベルのトークンの対応づけが期待される。さらに、“レベル”の高いトークンを用いればそれだけ対応の曖昧性を減少させることができ、対応づけ問題が容易になると考えられる。ところが、一般にトークンのレベルが上がる程、画像中からそうしたトークンを抽出することが難しくなる。これはトレードオフの問題であり、抽出、追跡といった 2 段階の処理構成をとる限り本質的に解決が困難といえよう。それゆえ、従来、輪郭等の高レベルのトークンを対応づけるためには、対象を限定し対象の形状に関する事前知識の利用を余儀なくされていた[50][51][52]。

これらの問題点を解決する一手法として、Kass ら [33] は、エネルギー最小化アプローチによるノンパラメトリックな輪郭抽出モデル (Snakes) を提案し、輪郭追跡への適用も試みている。Snakes による輪郭追跡は次のようにして実行される。まず、初期フレームにおいて追跡対象物の近傍に snake を置き (初期化し)、輪郭抽出を行なう。一旦、追跡対象輪郭が抽出されたなら、それを初期値として次フレームでの輪郭抽出を行なうことができる。この処理を逐次全フレームに渡って実行することにより snake を用いた輪郭追跡が実現できる。Snakes による輪郭追跡では、輪郭抽出と追跡の問題が同時に取り扱われているので、前述したトレードオフの問題が解消されるという利点がある。

以上述べたように、輪郭といったかなり高レベルのトークンを追跡する方法としては Snakes は非常に有力なものであることが分かる。ところが、Snakes を用いた輪郭追跡手法は、基本的には初期値として与えられた輪郭 (snake) が収縮しながら、抽出対象輪郭に張りついていくように振舞うため、追跡対象輪郭が隣接フレーム間で大きく平行移動するような場合 (図 3.7) には適用困難である。

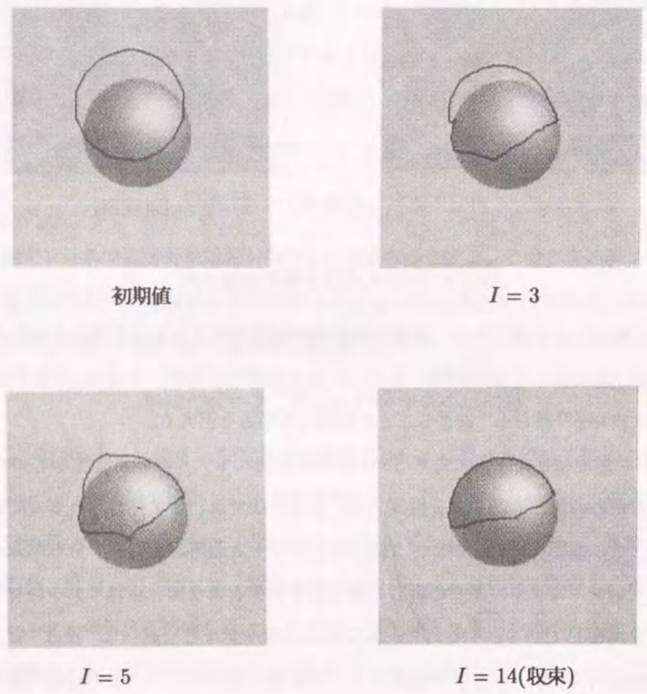


図 3.7: Snakes による輪郭追跡の例

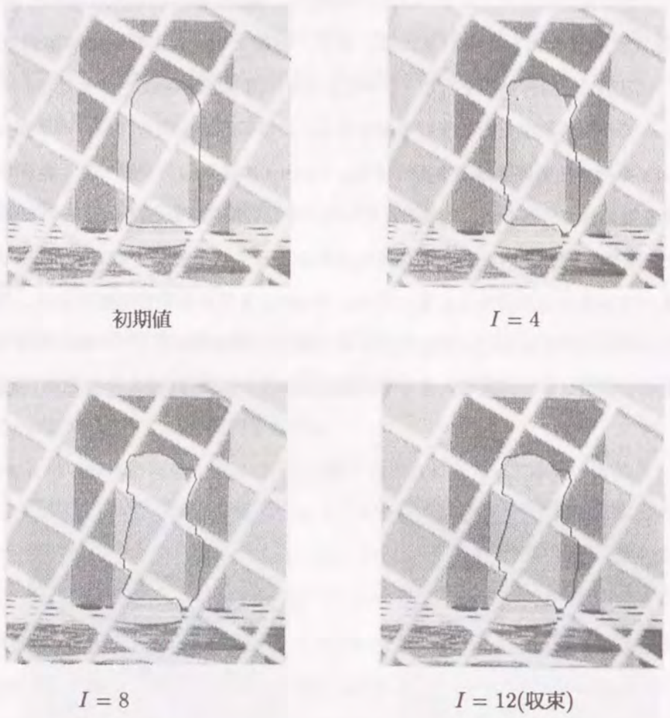


図 3.8: Snakes による輪郭追跡の例

また、移動量が比較的小さい場合でも、追跡対象物体が複雑なテクスチャを有していたり、遮蔽輪郭があるような場合 (図 3.8) には適用困難となる。これらの適用限界は、Snakes が初期値に大きく依存し、かつ snake 自身が"軟らか"過ぎることに起因していると言える。

そこで、以下の節では従来手法にない Snakes の能動的なモデルの利点を生かしつつ、かつ上記問題点を解決するための輪郭追跡手法を提案する。提案手法では、まず弾性体として記述された輪郭モデルを定義し、この輪郭自身の変形に伴う弾性エネルギーと追跡対象輪郭を含む画像の濃度勾配から生成されるポテンシャルエネルギーの両者の和を最小にするように、輪郭モデル自身を移動させることにより追跡を実現している。すなわち、輪郭を弾性体モデルとし適当な"硬さ"をもたせているため、追跡対象物体内部のテクスチャや遮蔽エッジの影響を受けにくく一般の追跡問題に適用可能な頑健な手法となっている。

尚、弾性マッチングという観点では、従来、幾つかの手法が提案されている。特に、山田ら

[53][54] は輪郭モデルを直接、画像中の対象物体にあてはめる手法を提案し、医用画像における臓器の抽出等への有効性を示している。ところがこれらの手法では、輪郭モデルは、ある許容範囲を自由に動くことができるだけで、モデル自身の弾性的制約が用いられていない。従って、前述した Snakes の問題点を解消するには至らず、ロングレンジの動物体の追跡には適用困難であると言える。

3.5.2 弾性輪郭モデルの導入

本追跡手法は、基本的には Snakes による輪郭追跡と同様に、前フレームでの輪郭追跡が完了していることを前提として現フレームの輪郭追跡を実行するものである。すなわち、前フレームで追跡された輪郭を現フレームでの追跡のための輪郭モデルとして利用する。

対象が剛体、非剛体に関わらず、動画像系列の隣接 2 フレーム間では、その移動量が大きい場合でも輪郭形状自身は極端に変形しないと考えられる。そこで本追跡手法では、輪郭モデルを Snakes で用いられているような柔軟なスプライン曲線ではなく、前フレームでの形状をできるだけ保存するような適当な"硬さ"を持った弾性モデルとすることで Snakes の前述した問題点を解決する。

輪郭モデルは単純曲線の連続モデルとして定義することも可能であるが、ここでは、ディジタル画像への適用を考慮して、輪郭モデルを n 個の頂点から成る多角形モデルで近似する。すなわち、輪郭モデルを、順序づけられた頂点の集合として、

$$\mathbf{C} = \{\boldsymbol{v}_i = (x_i, y_i)\}, \quad 1 \le i \le n \tag{3.80}$$

で表す。今、前フレームで追跡対象輪郭が得られているものとする。このとき、現フレームでの輪郭追跡問題を、前フレームでの追跡結果を初期値 $\{\boldsymbol{v}_i^0\}_{i=1}^n$ とし、式 (3.81) に示すエネルギー汎関数を最小とする変関数 $\{\boldsymbol{v}_i^*\}_{i=1}^n$ を求める問題として定式化する。

$$E_{total}(\boldsymbol{v}) = \sum_{i=1}^{n}\Big(E_{elastic}(\boldsymbol{v}_i) + E_{field}(\boldsymbol{v}_i)\Big) \tag{3.81}$$

ここで、$E_{elastic}$ は Snakes の前述した問題点を解決するために設けた弾性的制約から導かれる弾性エネルギーで、E_{field} は輪郭が位置している画像中のエッジポテンシャル場から導かれるポテンシャルエネルギーである。これらは、各々、Snakes の内部スプラインエネルギー、外部エネルギーに対応している。式 (3.81) は、直感的には、輪郭モデルが前フレームの形状を保存し、かつその輪郭モデルが、現フレームでその形状に類似したエッジ (輪郭) に配置されるとき最小となると解釈でき、所望の輪郭追跡が実現できる。以下では両者のエネルギーについて順に詳述する。

輪郭モデルは、前述したように n 多角形で表されているが、図 3.9 に示すように、各辺はその収縮に比例する復元力をもつバネで構成され、さらに隣接 2 辺間には、その挟角の角度変化に比例する

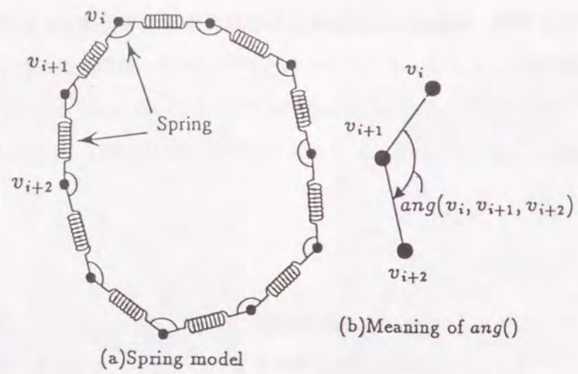


図 3.9: 弾性輪郭モデル

復元力をもつバネが付加されてある。このバネモデルは文献 [55] に用いられているものと同種のものである。全てのバネが自然長のときが、現フレームでの初期輪郭 $\{v_i^0\}_{i=1}^n$ に相当する。従って、輪郭モデルが変形を受ければ、上記 2 種類のバネから弾性エネルギーが生じる。すなわち、輪郭モデルの弾性エネルギーは次式で表される。

$$\begin{aligned} E_{elastic}(v_i) &= \frac{1}{2}\Big(\mu_1(|v_{i+1}-v_i| - |v_{i+1}^0 - v_i^0|)^2 \\ &+ \mu_2(ang(v_i, v_{i+1}, v_{i+2}) - ang(v_i^0, v_{i+1}^0, v_{i+2}^0))^2\Big) \end{aligned} \tag{3.82}$$

ここで、$ang(v_i^0, v_{i+1}^0, v_{i+2}^0)$ は図 3.9 に示すように辺 v_iv_{i+1} と辺 $v_{i+1}v_{i+2}$ とのなす角である。μ_1, μ_2 は正定数である。

3.5.3 距離変換に基づくポテンシャル場の生成

Snakes では画像中のエッジからのポテンシャル場として、画像の濃度勾配

$$z = -|\nabla I(x,y)|^2 \tag{3.83}$$

が用いられている。ところが濃度勾配によるポテンシャル場では、場がエッジの近傍にしか生成されない。それ故、追跡対象が隣接フレーム間で大きく平行移動しているような場合、場の影響が十分に及んでいないため、輪郭モデルが追跡対象輪郭エッジに引き寄せられず、追跡が困難となる。この解決策として、スケール (標準偏差)σ のガウス関数 G_σ による畳み込み演算 ($\otimes$) を用いて、エッジをぼかしておく方法が考えられる。すなわち、ポテンシャル場として

$$z = -|\nabla(G_\sigma \otimes I(x,y))|^2 = -(G_\sigma \otimes |\nabla I(x,y)|)^2 \tag{3.84}$$

を用いれば良い。しかしながら、この方法ではスケールの最適選択が新たな問題となる。スケールが小さすぎると場の影響が遠くに及ばず、逆に大きすぎると近傍のエッジ同士がマージされてしまう。すなわちこの方法では、画像中のエッジの空間的配置に応じた局所的なスケールの最適選択が必要となる。これは一般に容易なことではない。

そこで、この問題を解決するポテンシャル場の生成法として、距離変換 [56] に基づく方法を提案する。まず、微分演算子を用いて画像の濃度勾配を計算し、ある閾値で 2 値化する。これにより、濃度勾配の大きな領域 (エッジ領域に相当) が"1" となりそれ以外の領域 (平坦領域に相当) が"0" となる。得られた 2 値画像の 0/1 を反転し、その反転 2 値画像に対して距離変換を施す。すなわち、これにより、平坦領域の全画素に対し、エッジ領域からの離れ具合 (距離) が計算されることになる。明らかに、エッジ領域での距離は 0 である。

距離として、疑似ユークリッド距離 (水平・垂直方向の距離の増分 1、対角方向の距離の増分 $\sqrt{2}$ を各々 2、3 で近似) を用いた。距離変換は、"1" 画素 (平坦領域) には無限大の値を、"0" 画素 (エッジ領域) には値 0 を初期値として、平坦領域画素に対してのみ図 3.10 に示す計算を実行することにより得られる。そして、最終的に、画素 $I(x,y)$ のポテンシャルの値はその画素の距離値 $d(x,y)$ を用いて次式で定義する。

$$z = d(x,y) \tag{3.85}$$

図 3.11(a),(b) に各々エッジ画像とそのポテンシャル画像の例を示す。図 3.11(b) では輝度が大きい (明るい) 程ポテンシャル値が小さくなっている。同図から分かるように距離変換によるポテンシャル場では、エッジからの距離に比例したポテンシャルが生成され、接近したエッジは言うまでもなく、離れたエッジ間でも望ましいポテンシャル場が得られる。

ポテンシャル場 z が生成されると、モデル輪郭の各制御点を、質量 m をもつ質点とみなすことにより、各点のポテンシャル場のエネルギーを $m\mathcal{G}z$ で表すことができる。ここで、$\mathcal{G}$ は重力加速度である。従って、モデル輪郭のポテンシャルエネルギーは、式 (3.85) から、

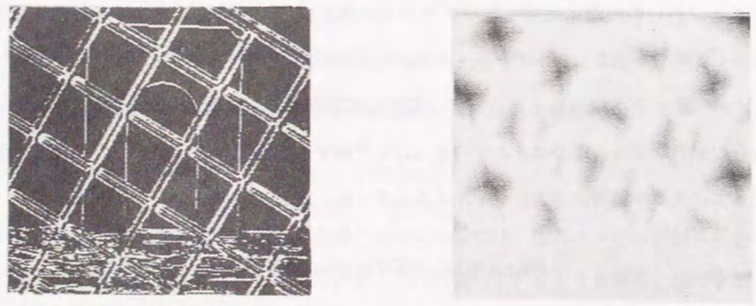
$$E_{feild}(v_i) = m\mathcal{G}z(v_i) = \mu_3 d(v_i) \tag{3.86}$$

となる。ここで、μ_3 は正定数である。

結局、輪郭追跡処理は式 (3.82),(3.86) を式 (3.81) に代入して得られる最小化問題を解くことに帰着される。そして得られた最小化問題は式 (3.29) と全く同じ関数形をしていることが分かる。それ故、この問題は 3.4 節で述べた解法をそのまま適用することにより容易に解くことができる。

```
begin
  for i := 2 to image_size do
    for j := 2 to image_size − 1 do
      I(i,j) := min{I(i,j),
                I(i−1,j)+2, I(i−1,j−1)+3,
                I(i,j−1)+2, I(i−1,j+1)+3}
  for i := image_size − 1 to 1 do
    for j := image_size − 1 to 2 do
      I(i,j) := min{I(i,j),
                I(i+1,j)+2, I(i+1,j+1)+3,
                I(i,j+1)+2, I(i+1,j−1)+3}
end
```

図 3.10: 距離変換アルゴリズム



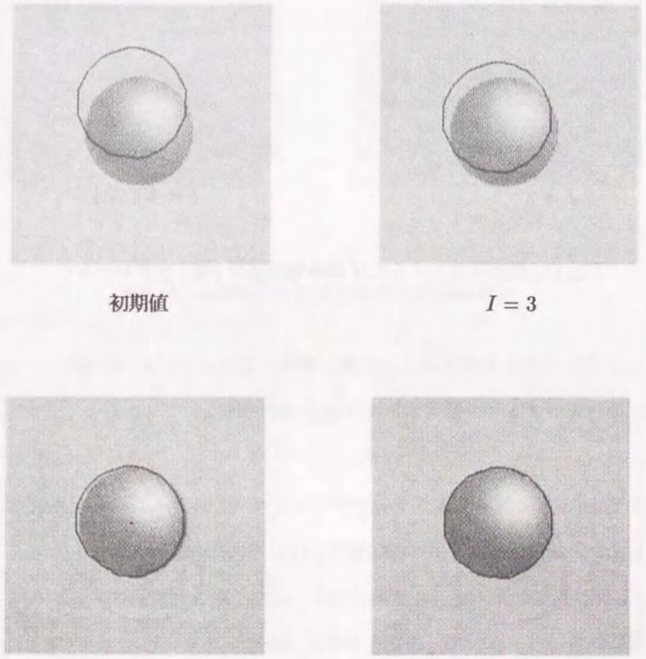
(a) エッジ画像　　(b) ポテンシャル画像

図 3.11: エッジ画像とそのポテンシャル画像の例

3.5.4 追跡実験

提案した輪郭追跡手法の有効性を検証すべく、CG 画像、および実画像に適用した結果を図 3.12 ～ 図 3.16 に示す。これらはいずれも 256×256,1 画素 256 階調の濃淡画像であるが、誌面の都合上、画像の一部を切り出して掲載している。図の説明における I は繰り返し数を表す。

輪郭の点列数は 40 から 80 程度である。また、初期輪郭 (初期値) は、前述したように、追跡対象輪郭のごく近傍に Snakes をおいて、輪郭抽出することにより得ている。図 3.12, 図 3.13 は弾性球が床でバウンドする動画 (15 フレーム) を CG でシュミレートしたものである。図 3.12 は図 3.7 と同じ実験画像である。図 3.12 では、20 画素以上の移動量があるにも関わらず、本手法では所望の追跡が実現できている。また、輪郭モデルは弾性体であるので、図 3.13 のような変形球の追跡にも適用できる。図 3.14 は 15 フレーム分の輪郭追跡結果を重畳表示したものである。



初期値　　$I = 3$

$I = 7$　　$I = 11$(収束)

Fig.3.12: 提案手法による CG 球の追跡結果 (第 1,2 フレーム)

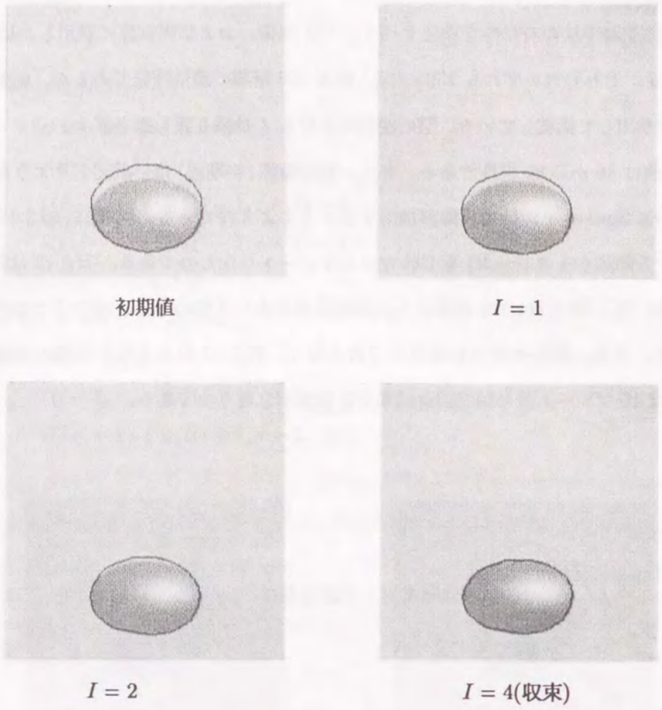


Fig.13: 提案手法による CG 球の追跡結果 (第 7,8 フレーム)

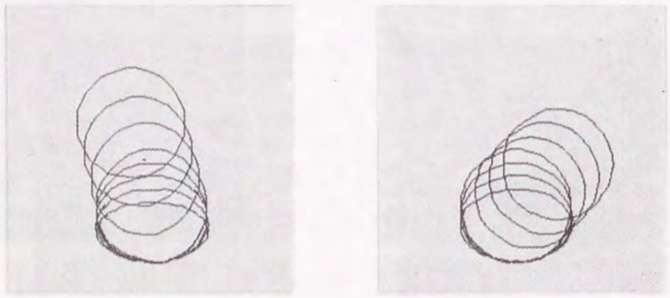


Fig.14: 提案手法による CG 球の追跡結果 (15 フレーム分)

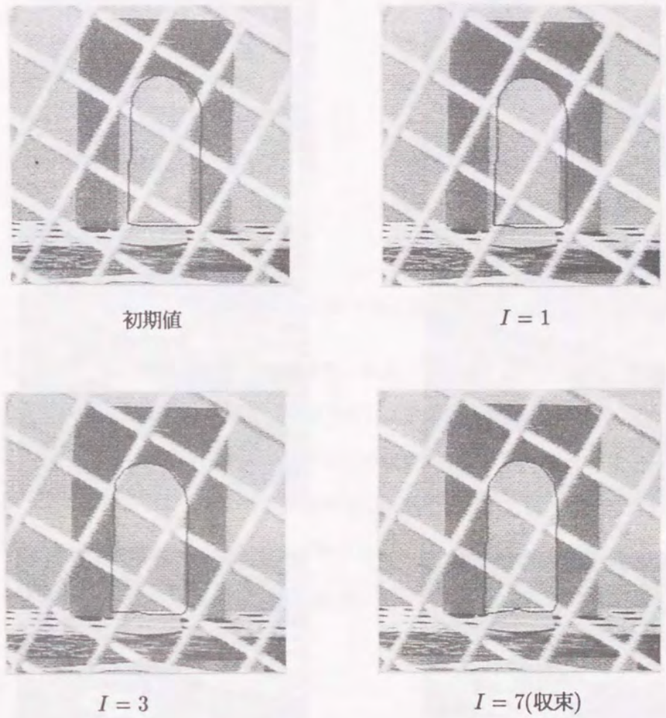


Fig.15: 遮蔽輪郭を有する場合での追跡結果

図 3.15 は回転テーブルの上にブックエンドを置き、格子ごしに回転させたものをカメラ入力したものである。図 3.8 で示したように、Snakes では遮蔽エッジの影響をまともに受けるが、本手法では良好な追跡結果が得られている。

図 3.16 は駐車場のシーンで、毎秒 30 フレームでビデオ入力した動画像を 10 フレームおきにサンプリングした 6 フレームから成る動画系列に対して本追跡手法を適用した結果である。また図 3.16 の最下段左図は第 6 フレームの画像に第 1 フレームから第 6 フレームまでの追跡結果を重畳したものである。図 3.16 の最下段右図のエッジ画像から分かるように、追跡対象物体には複雑なエッジがあり、しかも隣接フレームでの移動量も 10 画素以上あるにも関わらず、良好な追跡結果が得られている。尚、以上の実験では、処理パラメータを一律 $\mu_1 = 1, \mu_2 = 1, \mu_3 = 5$ とした。隣接フレームでの追跡が終了する (収束する) のにおよそ数回から十数回程度の繰り返しを要した。処理時間は 1 回の

図 3.16: 駐車場における車両の追跡結果

図 3.17: 点対応ベースのオプティカルフローの抽出結果

繰り返しに約数十秒 (Sun4/470) であった。また、処理パラメータ変動に対する実験も行なったが、極端に変動させない限り大差はなかった。一般的には、追跡対象輪郭形状が殆んど変化しないような対象では μ_3 に対して μ_1, μ_2 の値を大きくする方が収束が早くなる傾向にある。さらに、本追跡手法では、輪郭モデル自身が移動するので、隣接フレーム間での点対応が得られることになる。すなわち、図 3.17 に示すように点対応ベースのオプティカルフローが求まり、複数フレームにわたる特徴点軌跡も容易に得ることができるという長所がある。

3.6 結言

本章では、画像中から着目対象物の輪郭を抽出する手法、および動画像系列から着目対象輪郭を追跡する手法について述べた。これらの手法はいずれも Snakes の能動的モデルに基づき、その利点を生かし、かつ Snakes の欠点を補ったものである。すなわち、本章では、まず、Snakes モデルにおけるエネルギー最小化問題に対して、安定でかつ最適解を保証する新解法を提案し、その有効性を示した。そして、この解法が一般の離散正則化問題の統一的解法に一般化できることについて言及した。さらに、Snakes に弾性的制約を付与した弾性輪郭モデルによる動輪郭追跡手法を提案し、Snakes に対する優位性を示した。本追跡手法の特長を以下に整理する。

(1) ロングレンジの動画像に適用できる。

(2) 照明条件等の環境変動や、追跡対象物体内部のテクスチャ、遮蔽エッジの影響を受けにくい。

(3) 背景が動いていても、また追跡対象物体 (の一部) が静止していても追跡が可能である。

以上から、本章で提案した輪郭抽出・追跡手法は、画像理解の前処理として有効な手法と言える。

第 4 章

多重スケール凹凸構造マッチング

4.1 序言

同一クラスに属す輪郭図形サンプル群が与えられたとき、それらに共通する形状特徴を抽出するためには、それに先だって、まず各輪郭図形それ自身を記述する手段、すなわち形状表現法が重要課題となる。第 3 章の冒頭で述べたように形状の知覚にはその輪郭形状、すなわち輪郭の凹凸構造が極めて重要な役割を果たすことが心理学の知見から裏付けされている。また、凹凸構造特徴は、非常に一般性のある特徴であるため、任意の 2 次元輪郭図形の記述に適用可能である。しかしながら、単一の視野スケール (解像度) で図形を観測するのでは図形を正確に記述することはできず、十分な表現手段とは言い難い。換言すれば、正確な凹凸構造記述を実現するためには、多重スケールの視野尺度で図形を観測・記述可能な、より柔軟な凹凸構造表現手段が必要となる。

最近、1 次元波形や 2 次元図形の凹凸構造に関する多重スケール表現、そしてそれに基づくマッチング手法が幾つか提案されている [57][58][59][60][61] が、多重スケール表現における計算量・単調性の問題、そして多重スケール表現におけるマッチング戦略の問題等、重要課題が未解決のままである。

そこで、本章ではこれらの課題を解決すべく、多重スケール表現法およびそれを用いた多重スケールマッチング手法を提案する [62][63][64]。提案する多重スケール表現は基本的にはスケールスペース [57] の概念を用いているが、提案手法では、多重スケール表現の効率的な実現が可能で、しかも多重スケール表現における凹凸構造の一貫性を保証するといった従来手法にはない優れた特長をもっている。また、それに基づくマッチング手法においても、従来手法では困難であった変形の大きな図形間にも適用可能となっている。

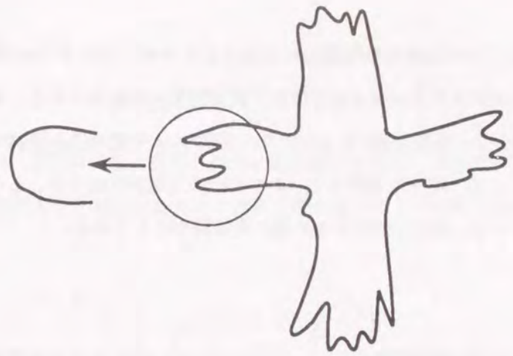


図 4.1: 輪郭図形の凹凸構造の観測

4.2 曲率スケールスペース表現

4.2.1 基本思想

今、図 4.1 に示す 2 次元図形に着目しよう。この図形は、大局的に見ると十字形のように見えるが、より詳細に観察すると、十字を構成する凸形状がさらに複数の凹凸構造を有していることが分かる。このように、凹凸構造というのは一見単純な特徴ではあるが、視野スケールを様々に変化させることにより、図形の複雑な構造をかなり正確に記述することができる。言い替えれば、変形の大きな図形間でも、単一スケールでは困難であった凹凸構造のマッチングが、部分輪郭毎に適切な視野スケールを選ぶことにより、それが可能となる。こうしたマッチングは我々人間が普段何気なく行なっているものである。従って、人間の直感にあった対応づけを実現するためにも、そのベースとなる多重スケール表現は極めて重要なものである。

図形輪郭の多重スケール表現を得る代表的な手法としてフーリエ記述子 [65] がある。これは、形状の輪郭を関数表現し、その関数をフーリエ級数展開して得られるフーリエ係数を形状表現に用いるものである。そして、フーリエ級数展開の第 N 項までの部分和で再構成されるパターン形状から、原形状を多重スケールで表現できる。ところが、この手法ではスケール間での形状の凹凸構造の変化、すなわち変曲点が未知であるため、各スケールでの近似パターンに対して何らかの手段により変曲点を求めなければならない。図形輪郭に対して、その輪郭が解析的な連続曲線として定義されていれば、変曲点を求めることは容易であるが、ディジタル画像上で与えられたディジタル曲線ではそう簡単な問題ではない [66][67]。これは、ディジタル曲線上で曲率を計算することが一般に困難であるからである。

近年、図形輪郭の凹凸構造に関する新たな多重スケール表現として曲率スケールスペース表現が

提案されている [59]。これは、1 次元波形を階層的に表現するために、先に Witkin[57] および趙ら [68] によって独立に提案されたスケールスペース表現の 2 次元図形への拡張である。曲率スケールスペース表現では、異なるスケール (標準偏差) をもつガウス関数[†]による畳み込み演算といった単純な線形演算を図形輪郭に施すことで、様々な視野スケールでの凹凸形状が生成でき、しかも同時に多重スケールの曲率が得られるという、従来手法にはない優れた特長を有している。

4.2.2 算法

輪郭 C 上の全ての点は 2 つの周期関数により、$C(t) = (x(t), y(t))$ のように表現できる。ここで、t はある開始点からの経路パラメータであり、通常 $t \in [0,1]$ で正規化される。このとき、$C(t)$ での曲率は次式で表される。

$$\kappa(t) = \frac{x'(t)y(t) - y'(t)x(t)}{(x'(t)^2 + y'(t)^2)^{3/2}} \tag{4.1}$$

$X(t,\sigma), Y(t,\sigma)$ を $x(t), y(t)$ とスケールファクタ σ をもつガウス関数 $g(t,\sigma)$:

$$g(t,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-t^2/2\sigma^2} \tag{4.2}$$

との各々の畳み込み演算 ($\otimes$) として次のように定義する。

$$\begin{aligned} X(t,\sigma) &= x(t) \otimes g(t,\sigma) \\ &= \int_{-\infty}^{+\infty} x(u) g(t-u,\sigma) du \\ &= \int_{-\infty}^{+\infty} x(u) \frac{1}{\sigma\sqrt{2\pi}} e^{-(t-u)^2/2\sigma^2} du \end{aligned} \tag{4.3}$$

$Y(t,\sigma)$ も同様である。明らかに, σ が大きい程、輪郭がより平滑化されることになる。この平滑化演算は、曲率スケールスペースフィルタリングと呼ばれる。以下ではこの平滑化された輪郭を平滑化曲線と呼ぶことする。

平滑化曲線上の曲率は、スケール σ を用いて次式で表わされる。

$$K(t,\sigma) = \frac{X'(t,\sigma)Y''(t,\sigma) - X''(t,\sigma)Y'(t,\sigma)}{(X'(t,\sigma)^2 + Y'(t,\sigma)^2)^{3/2}} \tag{4.4}$$

ここで、$X'(t,\sigma)$ は、$\partial X(t,\sigma)/\partial t$ を表わし、

$$\begin{aligned} X'(t,\sigma) &= \frac{\partial X(t,\sigma)}{\partial t} = \frac{\partial [x(t) \otimes g(t,\sigma)]}{\partial t} \\ &= x(t) \otimes \left(\frac{\partial g(t,\sigma)}{\partial t}\right) \end{aligned} \tag{4.5}$$

[†]趙ら [68] はグリーン関数による畳み込みを行なっている。

で計算され、また、$X''(t,\sigma)$ は、

$$X''(t,\sigma) = \frac{\partial^2 X(t,\sigma)}{\partial t^2} = x(t) \otimes \left(\frac{\partial^2 g(t,\sigma)}{\partial t^2}\right) \tag{4.6}$$

で計算される。Y', Y'' も同様に計算される。式 (4.5),(4.6) は、関数をガウス関数で畳み込んだ後に微分するのは、関数を予め微分したガウス関数で畳み込むことと同値であることを示している。

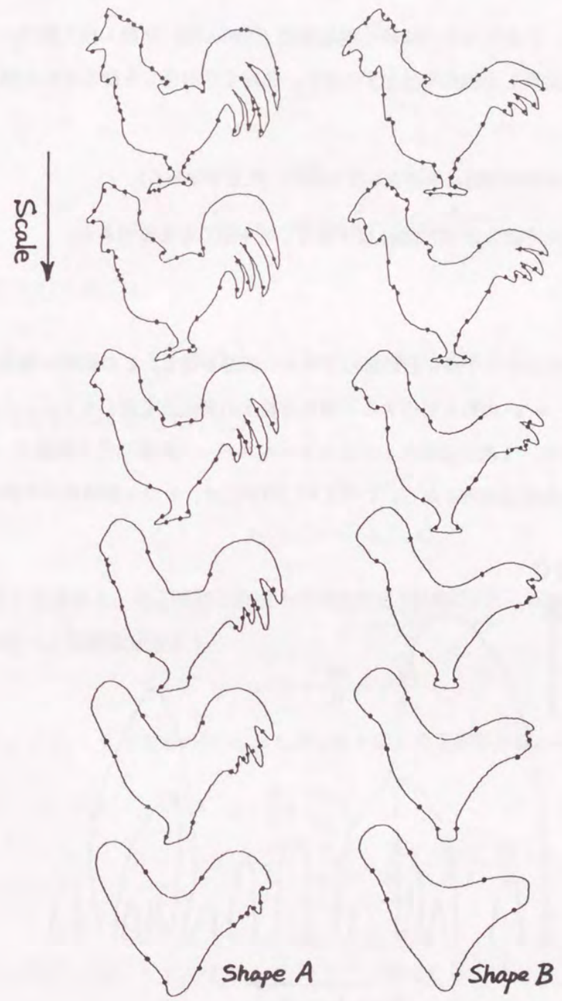


図 4.2: スケール変化に伴う曲線の平滑化

式 (4.4) で異なるスケールでの零交差点 (変曲点) を見い出すことにより、平滑化曲線の異なるスケールでの凹凸構造を得ることができる。図 4.2 に σ を離散的に変化させたときの平滑化曲線、および変曲点を示す。同図からわかるように、スケール (σ) を大きくしていくにつれて変曲点が減少し、曲線がより平滑化されていく様子がうかがえる。

一般に閉曲線が曲率スケールスペースフィルタリングにより平滑化されるとき、その平滑化過程において以下の性質が成り立つことが理論的に証明されている [70]。

性質 4.1 (相似変換に不変) 平滑化はその曲線の相似変換 (回転・平行移動・拡大縮小) に不変である。すなわち、曲線を相似変換した後に平滑化するのは、曲線を平滑化した後に相似変換するのと同値である。

性質 4.2 (連結性) 連結した平面曲線は平滑化しても連結したままである。

性質 4.3 (閉曲線性) 閉じた平面曲線 (閉曲線) は平滑化しても閉じたままである。

4.2.3 階層的記述

スケールを変化させて得られる平滑化曲線群の変曲点の軌跡を描く。この軌跡は輪郭上のある開始点からの経路 t とスケール σ の組に対応する平滑化曲線上の変曲点を表している。これは曲率スケールスペース画像と呼ばれ、1 次元波形におけるスケールスペース画像 [57] と同様に、図 4.3 に示すような上に凸で下に開いた形状境界となる。この上に凸の形状は、2 つの変曲点が平滑化とともに

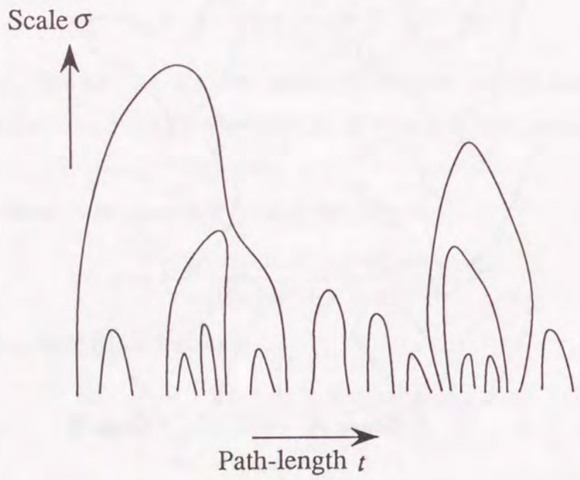


図 4.3: 曲率スケールスペース画像の例

輪郭上で次第に近づき最後には消滅していく様子を図示したものと解釈できる。より、厳密には次の定理の形で表現できる [70]。

定理 4.1 Γ を C_1 級の平面曲線とし、その平滑化曲線を Γ_σ で表すとき、全ての平滑化曲線 Γ_σ が C_1 級であるなら、Γ の曲率スケールスペース画像上での全ての極値は極大値である。

[証明] 便宜上、$s = \frac{1}{2}\sigma^2$ とし、

$$\begin{aligned} x(t,s) &= X(t,\sigma) \\ &= x(t) \otimes \left(\frac{1}{\sqrt{4\pi s}} e^{-t^2/4s}\right) \end{aligned} \tag{4.7}$$

$$\begin{aligned} y(t,s) &= Y(t,\sigma) \\ &= y(t) \otimes \left(\frac{1}{\sqrt{4\pi s}} e^{-t^2/4s}\right) \end{aligned} \tag{4.8}$$

とおくと、式 (4.4) の分子は、

$$\alpha(t,s) = x_t(t,s)y_{tt}(t,s) - x_{tt}(t,s)y_t(t,s) \tag{4.9}$$

とかける。このとき、式 (4.8),(4.9) より、

$$x_{tt}(t,s) = x_s(t,s) \tag{4.10}$$

$$y_{tt}(t,s) = y_s(t,s) \tag{4.11}$$

が成り立つことが分かる。今、平滑化曲線上の零交差点 (変曲点) は、$\alpha(t,s) = 0$ を満たすから、$s = s(t)$ とおけて、陰関数定理より

$$s'(t) = \frac{ds}{dt} = -\frac{\alpha_t}{\alpha_s} \tag{4.12}$$

となる。故に、$s'(t) = 0$ となるのは $\alpha_t = 0$ のときに限ることが分かる。一方、

$$\begin{aligned} s''(t) &= \frac{d}{dt}\left(-\frac{\alpha_t}{\alpha_s}\right) \\ &= \frac{\partial}{\partial t}\left(-\frac{\alpha_t}{\alpha_s}\right) + \frac{\partial}{\partial s}\left(-\frac{\alpha_t}{\alpha_s}\right) \cdot \frac{ds}{dt} \end{aligned} \tag{4.13}$$

となる。

証明したいのは、$S'(t) = 0$ のとき、$s''(t) < 0$ であるから、$s'(t) = 0$ を式 (4.13) に代入すると、

$$s''(t) = -\frac{\alpha_{tt}}{\alpha_s} \tag{4.14}$$

となるから、結局、$\alpha(t,s) = \alpha_t(t,s) = 0$ のとき、$\alpha_{tt}/\alpha_s > 0$ となることを示せば良い。式 (4.9),(4.10),(4.11) より、

$$\alpha = x_t y_s - x_s y_t \tag{4.15}$$

$$\alpha_t = x_t y_{st} - x_{st} y_t \tag{4.16}$$

となり、同様に、

$$\alpha_{tt} = (x_t y_{ss} - x_{ss} y_t) + (x_s y_{ts} - x_{ts} y_s) \tag{4.17}$$

となる。従って、$\alpha = \alpha_t = 0$ ならば、

$$\begin{aligned} x_s y_{ts} - x_{ts} y_s &= x_s(y_{ts} - x_{ts} \cdot \frac{y_s}{x_s}) \\ &= x_s(y_{ts} - x_{ts} \cdot \frac{y_t}{x_t}) \qquad (\leftarrow \alpha = 0) \\ &= \frac{x_s}{x_t}(x_t y_{ts} - x_{ts} y_t) \\ &= 0 \qquad (\leftarrow \alpha_t = 0) \end{aligned} \tag{4.18}$$

式 (4.18) を式 (4.17) に代入すると、

$$\alpha_{tt} = x_t y_{ss} - x_{ss} y_t \tag{4.19}$$

となる。一方、

$$\begin{aligned} \alpha_s &= x_{ts} y_s + x_t y_{ss} - x_{ss} y_t - x_s y_{ts} \\ &= (x_t y_{ss} - x_{ss} y_t) + (x_{ts} y_s - x_s y_{ts}) \\ &= x_t y_{ss} - x_{ss} y_t \qquad (\leftarrow Eq.(4.18)) \\ &= \alpha_{tt} \qquad (\leftarrow Eq.(4.19)) \end{aligned} \tag{4.20}$$

であるから、

$$s''(t) = -\frac{\alpha_{tt}}{\alpha_s} = -1 < 0 \tag{4.21}$$

となる。従って、$s(t)$ 平面 (スケールスペース画像平面) で $s'(t) = 0$ のときは必ず $s''(t) < 0$ となるから $s(t)$ の極値は極大値となる。[証明終り]

定理 4.1 は、曲率スケールスペースにおける零交差の単調性と呼ばれ、1 次元波形におけるスケールスペースの単調性 [71][72] に対応する極めて重要な定理である。

Witkin はスケールスペース画像を上位スケールからトレースしていき、変曲点軌跡の頂点の配置を区間木 (interval tree) と呼ぶ木構造で階層表現した。2 次元輪郭図形の場合も曲率スケールスペース画像に対して同様な木構造を得ることができる。定理 4.1 と関連して次の定理が成り立つ。

定理 4.2 曲率スケールスペース画像から得られる区間木は *3* 分木を構成する。

[証明] 定理 4.1 より明らか。[証明終り]

同様な木構造として、曲率原始スケッチ [73]、あるいは、安定視点木と呼ばれる図形の階層表現が、趙ら [69] により提案されている。また佐藤ら [60] は変曲点軌跡から幾何学的構造を備えた木構造を提案している。

しかしながら、こうしたスケールスペースに基づく多重スケール表現は、スケールを連続的に変化させるとそれに伴って計算量・データ量が膨大となり実用上問題がある。この問題を避けるため、実際には適当な離散スケールを使用することになる。ところが、離散的なスケールを用いると、スケール変化に伴う変曲点の位置変動のため、軌跡の輪郭が不連続になり上述したトレースが困難となる。従って、スケール変化に伴う凹凸の変化の一貫性が保証されなくなる。凹凸の変化の一貫性とは、図 4.4 に示すように、奇数個の連続する凹凸セグメントがスケールの増加に伴って必ず一つの凹または凸セグメントに置換される規則をいう。これは凹凸の階層構造を捉える上で重要な規則である。

そこで、次節では、上記の問題点を解決するために、変曲点軌跡を用いずに、図 4.4 に示すような凹凸の階層構造を得るためのアルゴリズムについて詳述する。得られる多重スケール表現は曲率スケールスペース表現と類似しているが、隣接スケール間で凹凸の変化の一貫性を満たし、かつ変曲点位置変動の総和が最小になるように直接、変曲点が対応づけられている点が異なる。故に、変曲点軌跡の生成 (曲率スケールスペース画像の生成) とそのトレースが不要となり、かつ離散スケールの使用が可能となる。それゆえ、スケールスペース表現における計算量・データ量を大幅に削減することができる。

4.2.4 凹凸構造の階層化アルゴリズム

いま、$\mathbf{P}^{(k+1)}$、$\mathbf{P}^{(k)}$ を隣接する離散的スケール $\sigma^{(k+1)}$, $\sigma^{(k)}(\sigma^{(k+1)} > \sigma^{(k)})$ に各々対応する 2 組の変曲点集合とする。すなわち、

$$\begin{aligned} \mathbf{P}^{(k+1)} &= \{p_i^{(k+1)} \mid i = 1, 2, \ldots, N^{(k+1)}\} \\ \mathbf{P}^{(k)} &= \{p_j^{(k)} \mid j = 1, 2, \ldots, N^{(k)}\} \end{aligned} \tag{4.22}$$

これらの各々は、ある開始点から平滑化輪郭に沿って見いだされる変曲点点列から成る。閉輪郭であることから次の性質が成り立つ。

性質 4.4 任意の自然数 k について、$N^{(k)}$ は偶数。

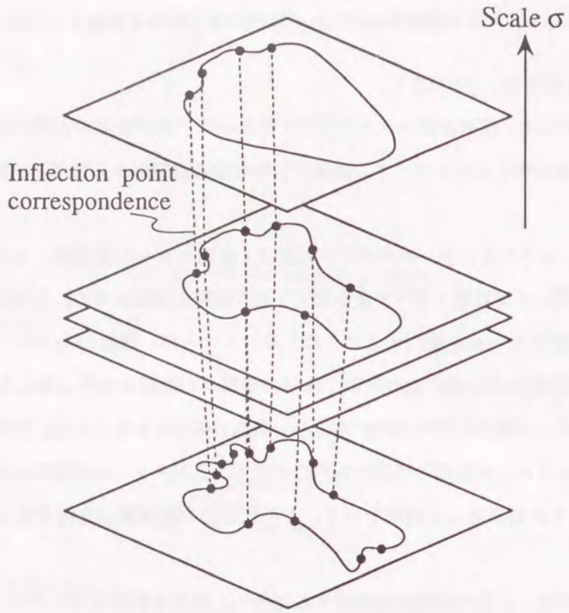


図 4.4: 凹凸の変化の一貫性を満たす多重スケール表現の例

性質 4.5 任意の自然数 k について、$\mathbf{P}^{(k)}$ は巡回集合である。

性質 4.4 より、任意の自然数 k について、$p^{(k)}_{N^{(k)}+i} = p^{(k)}_i$ とする。また、曲率スケールスペースにおいて、次の変曲点の単調性が成り立つ。

性質 4.6 すべての自然数 k について、$N^{(k+1)} \leq N^{(k)}$。

いま J を集合 $\mathbf{P}^{(k+1)}$ から集合 $\mathbf{P}^{(k)}$ への写像とすると、J は先に述べた凹凸の変化の一貫性を満たすために次の条件を満たさなければならない。

条件 4.1 J は単射である。但し、必ずしも全射ではない。

条件 4.1 は隣接スケールの 2 つの変曲点集合において上位スケールの変曲点に対しては必ず下位スケールの変曲点が対応しなければならないこと、およびその逆は必ずしも成り立つ必要がないことを意味している (図 4.5)。

条件 4.2 $p^{(k+1)}_i, p^{(k+1)}_{i+1}, p^{(k+1)}_{i+2}$ を、反時計回りに輪郭をトレースしたときの連続する *3* つの変曲点とし、$p^{(k)}_{J(i)}, p^{(k)}_{J(i+1)}, p^{(k)}_{J(i+2)}$ を各々対応する変曲点とすると、$p^{(k)}_{J(i+1)}$ は $p^{(k)}_{J(i)}$ と $p^{(k)}_{J(i+2)}$ の間の部分輪郭上になければならない *(図 4.6)*。

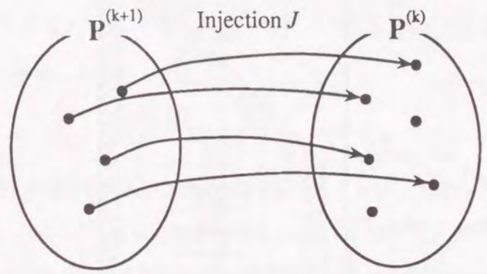


図 4.5: $\mathbf{P}^{(k+1)}$ から $\mathbf{P}^{(k)}$ への写像 J

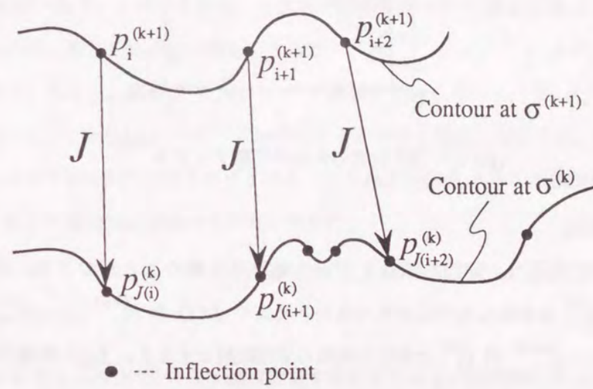


図 4.6: 条件 4.2 の説明

条件 4.3 $p^{(k+1)}_i$ と $p^{(k)}_{J(i)}$ の極性は等しい。ここで、極性とは、変曲点の前後での曲率の変化のタイプ *(正から負、またはその逆)* を表わす。

さて $d(i,j)$ を $p^{(k+1)}_i$ と $p^{(k)}_j$ の間のユークリッド距離を表すものとすると、解くべき問題は上記条件のもとで式 (4.23) を最小にする写像 J を見つけることに帰着される。

$$D(\mathbf{P}^{(k+1)}, \mathbf{P}^{(k)}) = \min_{\{J(i)\}} \sum_{i=1}^{N^{(k+1)}} d(i, J(i)) \tag{4.23}$$

この定式化により、得られた変曲点対応が凹凸の変化の一貫性を満たし、かつ対応する変曲点間の位置ずれの総和が最小になることが保証される。

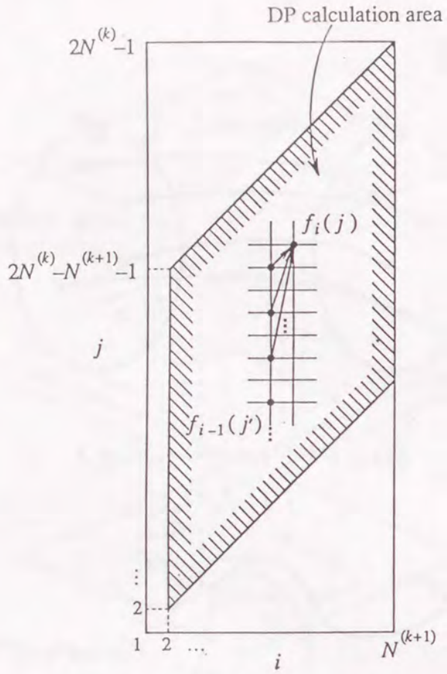


図 4.7: DP 計算のための距離テーブル

[動的計画法による解法]

式 (4.23) の最小化問題は、動的計画法を用いて効率良く解くことができる。条件 4.1 より各 $p_i^{(k+1)}$ は必ずある $p_j^{(k)}$ と対応しなければならない。そこで $f_i(j)$ を $p_1^{(k+1)},\ldots,p_{i-1}^{(k+1)}$ が $\mathbf{P}^{(k)}$ の各要素と対応し、かつ $p_i^{(k+1)}$ が $p_j^{(k)}$ と対応した時の累積距離とすると、$f_i(j)$ は式 (4.24) のように書ける。

$$f_i(j) = \min_{J(1),J(2),\ldots,J(i)} \sum_{l=1}^{i} d(l, J(l)) \tag{4.24}$$

従って、最適性の原理より、$f_i(j)$ は次の漸化式で書き表せる。

$$f_i(j) = \min_{j' \in \mathbf{R}} \{f_{i-1}(j') + d(i,j)\}, \qquad \text{for} \quad i = 2,3,\ldots,N^{(k+1)}. \tag{4.25}$$

ただし、$f_1(j) = d(1,j)$ とする。

$f_i(j)$ の計算を行なうのに図 4.7 に示す $N^{(k+1)} \times (2N^{(k)}-1)$ のテーブルを用いる。このテーブルの第 (i,j) 要素には $f_i(j)$ の値が格納される。そして、第 j 行と第 $N^{(k)}+j$ 行の両者はともに $p_j^{(k)}$ に対応し、これにより、すべての開始点対応を調べることが可能になる。これは、変曲点点列の巡回性 (性質 4.2) により $p_1^{(k+1)}$ が必ずしも $p_1^{(k)}$ に対応するとは限らないため必要な処理である。式 (4.26)

における $\mathbf{R}$ は、上記の 3 つの条件を満たすための j' に対する制約条件である。

$$\mathbf{R} = \{j' \mid i-1 \le j' \le j-1 \quad \text{かつ} \quad j-j' = 2n+1 \text{かつ} \quad 0 \le n \le T_{i-1}(j')/2\} \tag{4.26}$$

ここで、T は式 (4.27) を満たすものとする。

$$T_i(j) = T_{i-1}(j'^*) - j + j'^* + 1 \tag{4.27}$$

ただし、j'^* は式 (4.25) を最小にする j' の最適値である。さらに、上記の条件 1 から 3 を考慮すると、$f_i(j)$ は $i = 2,3,\ldots,N^{(k+1)}$, $j = i+c, i+2+c, \ldots, i+(2N^{(k)}-N^{(k+1)}-2)+c$ についてのみ計算すれば良いことが分かる。ここで、c は定数であり、異なる極性をもつ変曲点同士の対応を防止するためのもので、$p_1^{(k+1)}$ と $p_1^{(k)}$ の極性が同じなら 0、さもなくば 1 とする。

$\mathbf{R}$ の意味するところは次のとうりである。第 $i-1$ 段から第 i 段の遷移のみに着目すると、j' のとり得る値の範囲は、条件 4.1,4.2 から式 (4.26) の最初の不等式に限定される。さらに、条件 4.3 より、$j-j'$ は奇数となる。しかしながら、これらの制約条件は第 1 段から第 $i-1$ 段までの対応を考慮していないので、重複対応が生じ得る。すなわち、$p_i^{(k+1)}$ と $p_{i'}^{(k+1)}$ $(i \ne i')$ が共に $p_j^{(k)}$ に対応する可能性がある。そこで、関数 T を用いて j' の範囲を制限している。$T_{i-1}(j')$ は図 4.7 に示したテーブル上で $(i-1,j')$ から (i,j) へのパスが存在するか否かを動的に判定する。すなわち、$T_{i-1}(j')$ により、第 $i-1$ 段までに対応づけされずにスキップされた変曲点 $p_j^{(k)}$ の累積総数が $N^{(k)}-N^{(k+1)}$ を超えないようにして重複対応を防いでいる。明かに、

$$T_1(j) = N^{(k)} - N^{(k+1)} \tag{4.28}$$

である。

$f_i(j)$ を $i=0$ から $i = N^{(k+1)}$ まで逐次計算することにより、最終的に式 (4.29) に示す最小距離が求まる。

$$D(\mathbf{P}^{(k+1)}, \mathbf{P}^{(k)}) = \min_{\{j \mid T_{N^{(k+1)}}(j)=0\}} \{f_{N^{(k+1)}}(j)\} \tag{4.29}$$

ここで、式 (4.29) における $T_{N^{(k+1)}}(j) = 0$ は $j = N^{(k+1)}+c, N^{(k+1)}+2+c, \ldots, (2N^{(k)}-2)+c$ なる j についてのみ調べれば良い。式 (4.29) の最小値は式 (4.23) の最小値と完全に一致している。

上記の繰り返し計算の途中で最適パラメータ j' を $\mathbf{J}_i(j)$ テーブルに蓄えておく。いま、$\hat{J}(i)$ を隣接スケール間での変曲点の最適対応 $(i, J(i))$ を表わすこととすると、$\hat{J}(i)$ は $\mathbf{J}_i(j)$ テーブル参照により次のようにして逐次得られる。

$$\begin{aligned} &\hat{J}(N^{(k+1)}) = \arg\min_j \{f_N^{(k+1)}(j)\}, \\ &\hat{J}(i) = \mathbf{J}_{i+1}(\hat{J}(i+1)), \\ &\quad \text{for} \quad i = N^{(k+1)}-1, N^{(k+1)}-2, \ldots, 1. \end{aligned} \tag{4.30}$$

ここで、“$\arg\min_j$” は最適パラメータ j の値を表わす。

以上の処理により、先に述べた図形輪郭の凹凸の階層構造を得ることができる。

4.3 Coarse-to-fine 戦略による多重スケールマッチング

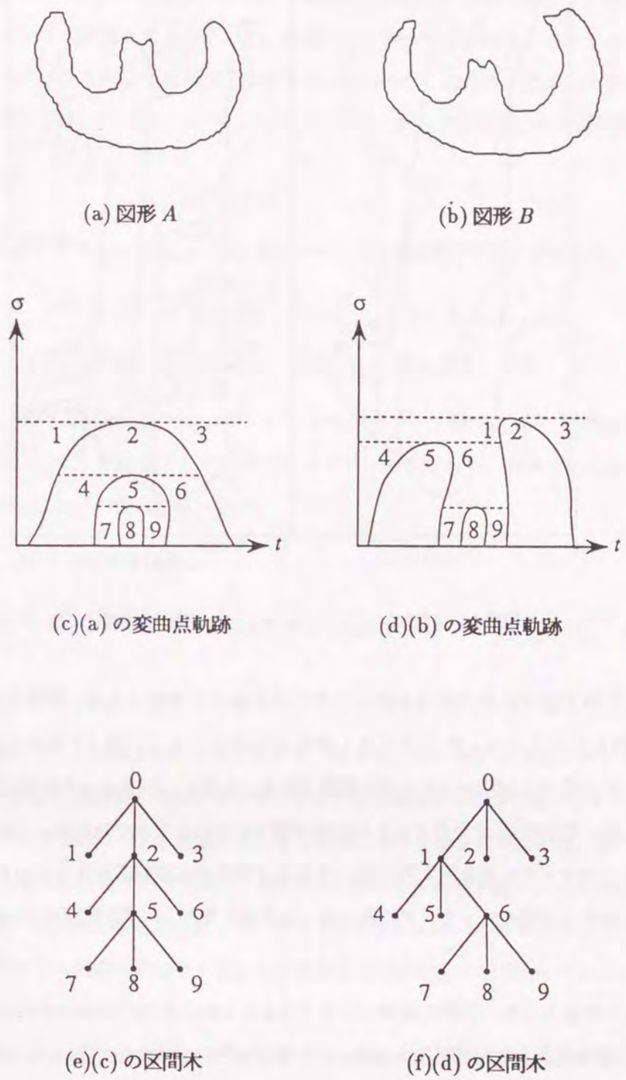
前節では、曲率スケールスペースに基づく多重スケール表現法について述べたが、本節ではその多重スケール表現を用いたマッチングアルゴリズムについて述べる。スケールスペース表現を用いた既存のマッチング手法は、全て Coarse-to-fine 戦略、すなわち、大局から詳細へとマッチングを実行する方式であり、その代表的手法に、Mokhtarian らの方法 [59] と、酒匂らの方法 [61] がある。前者の方法は、多重スケールの凹凸を木構造で記述し、2 つの木のマッチングを実行するものである。後者の方法は、マッチングの区間を上位スケールから順にしぼっていく方法である。以下では、この両者の手法の概略および問題点を述べ、coarse-to-fine 戦略によるマッチング手法の適用限界を明らかにする。

4.3.1 木構造マッチング

Mokhtarian らは、Witkin の提案した区間木を用いたマッチング法を提案している。図 4.8 にその例を示す。まず、図 4.8(a),(b) に示す様な 2 図形が与えられたとすると、その曲率スケールスペース画像は、各々図 4.8(c),(d) のようになる。従って、それらから図 4.8(e),(f) に示す区間木が得られる。2 図形のマッチングの問題は、木のマッチングの問題に帰着される。しかしながら、この方法では、比較的小さな変形に対しても変曲点の位相構造が大きく変化するため、それに伴い木構造が急激に変化する。従って、木構造のマッチングが困難となる。すなわち、この方法は軌跡の包含関係が保存されるようなかなり変形の小さな図形にしか適用できない。

4.3.2 区間分割マッチング

酒匂らの方法は、2 つの図形のスケール σ_a, σ_b を最上位スケールから互いに減少させ、マッチング誤差が最小となるスケール σ_a^*, σ_b^* で初期マッチングを行なう。そして対応づいた変曲点で部分輪郭に分割し、各々のスケールを σ_a^*, σ_b^* より小さくしていった時に現れる変曲点群のマッチングを最下位のスケールまで再帰的に行なうもので、変曲点の変化をかなり吸収でき、4.3.1 の区間木によるマッチングに比べより柔軟なマッチングが期待できる。この方法は、Baker & Binford[74] の階層的ステレオマッチング手法に類似している。



(a) 図形 A　(b) 図形 B

(c)(a) の変曲点軌跡　(d)(b) の変曲点軌跡

(e)(c) の区間木　(f)(d) の区間木

図 4.8: 区間木によるマッチングの例

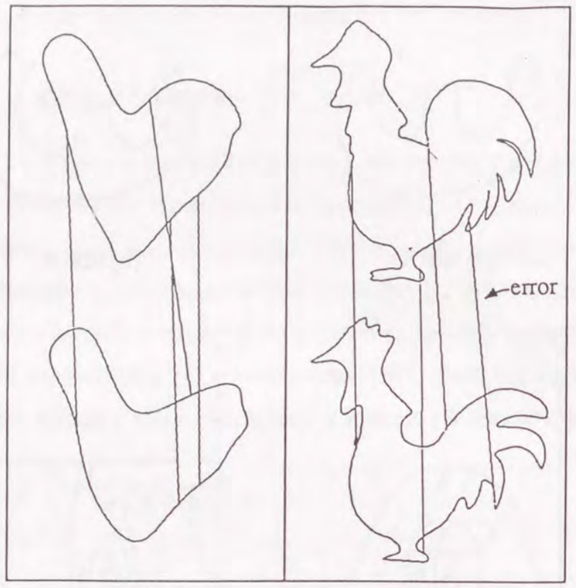


図 4.9: 区間分割法によるマッチング誤りの例

ところが、この方法では大局的な初期マッチングが極めて重要となる。すなわち、この段階でマッチングを誤ると以下のマッチングは全く意味を持たなくなる。図 4.9 にその一例を示す。図 4.9(a) は酒匂らの方法よる初期マッチングの結果である。ただし、パラメータはすべて文献 [61] と同じものを使用した。図 4.9(a) を見ると正しい結果が得られているようであるが、この結果に対応する最下位スケールでのマッチングを示す図 4.9(b) を見ると明らかに誤対応を含んでいる。従って、誤対応の部分輪郭における以降のマッチング処理は全く無意味となる。同様な問題は Baker らの手法でも生じる。

この例からわかるように、図形の変形が大きくなるにつれ、大局でのマッチングの信頼性が低くなるため、より変形の大きな図形間のマッチングにはこの方法では十分でない。これは、上記方法が大局的な初期マッチングの段階で輪郭全体を同一スケールで捉えていることに起因している。すなわち、より正確なマッチングを行なうには、最初から部分的にスケールを変化させてマッチングを行なう必要がある。

そこで、4.4 節では、こうした問題点を解決するための新たなマッチング手法について述べる。

4.4 最適スケール凹凸構造マッチング

本節で提案するマッチング手法は、変曲点の位置のみに着目するのではなく、変曲点間の凹凸セグメントをマッチングの単位とする。そして、多重スケール上で全てのセグメントの組合せの中から最適スケールのセグメント対応を効率良く決定する。それゆえ、部分的に変形の大きな図形間にも適用可能となる。以下では、この最適スケールセグメントマッチング手法について詳述する。

4.4.1 準備

$\mathbf{A}^{(h)}$,$\mathbf{B}^{(k)}$ を各々スケール $\sigma^{(h)}$, $\sigma^{(k)}$ での凹凸セグメント系列とする。すなわち、

$$\begin{aligned}\mathbf{A}^{(h)} &= a_1^{(h)}, a_2^{(h)}, \ldots, a_i^{(h)}, \ldots, a_{N^{(h)}}^{(h)}, \quad h = 0, 1, \ldots, H, \\ \mathbf{B}^{(h)} &= b_1^{(k)}, b_2^{(k)}, \ldots, b_j^{(h)}, \ldots, b_{M^{(k)}}^{(k)}, \quad k = 1, 2, \ldots, K.\end{aligned} \tag{4.31}$$

ここで、$N^{(h)}$、$M^{(k)}$ は各々スケール $\sigma^{(h)}, \sigma^{(k)}$ でのセグメント数で、$a_i^{(h)}$ は同一輪郭上の隣接する 2 つの変曲点 $p_i^{(h)}, p_{i+1}^{(h)}$ を両端点とする凹 (凸) セグメントを表わす。このとき任意のスケールのセグメント集合に対して以下の性質が成り立つ。

性質 4.7 $N^{(h)}$、$M^{(k)}$ は偶数である。

性質 4.8 $\mathbf{A}^{(h)}, \mathbf{B}^{(k)}$ は巡回系列となる。すなわち、$a_{N^{(h)}+i}^{(h)} = a_i^{(h)}$, $b_{M^{(k)}+j}^{(k)} = b_j^{(k)}$ となる。

4.4.2 マッチング戦略

提案する多重スケールセグメントマッチングは、最下位スケールから上位スケールにわたって、最下位スケールの奇数個の連続する凹凸セグメンをある上位スケールにおける一つの凹 (凸) セグメントで置換することにより対応ペアを探索する。すなわち、本マッチング手法では、最下位スケールのセグメント集合を基準に行なうのではなく、図 4.10 に示すように最下位スケールの奇数個のセグメントが 4.2 節で述べた凹凸の階層構造により、ある上位スケールの一つのセグメントに置換され得るとき、そのセグメントも対応の候補とする。この置換は 4.2 節で述べた凹凸セグメントの一貫した階層構造に基づいて実行される。

この階層表現における凹凸の変化の一貫性は、その階層性に基づいて置換され得る連続する凹凸セグメントのみがマッチングに用いられるという点で、マッチングにおける一つの拘束条件とみなすことができる。最下位スケールにおけるある連続する凹凸セグメントが上位スケールで置換され得ないということは、それらのセグメントが大局的に一つの凹 (凸) セグメントとして観測することができないことを意味しているからである。換言すれば、もし任意の連続する凹凸セグメントをマッチング

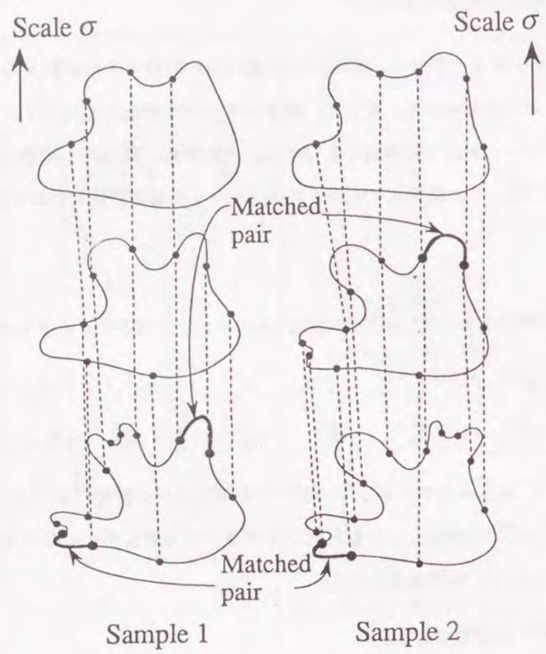


図 4.10: マッチング戦略

の候補として許せば、誤ったあるいは直観に合わないマッチングが生じる可能性がある。こうした任意のマッチングを禁止することは、余分なマッチングを抑制するのみならず、2 つの多重スケール図形間での妥当な対応を導くのに重要な役割を果たす。同様な制約は、ステレオビジョン [75] においても用いられている。文献 [75] では、2 つの画像 (ステレオ画像) の一方においてオクリュージョンによる奥行きの不連続なところでの誤対応を避けるために、"対応禁止領域 (forbidden zone)" なる対応の拘束条件を導入している。

さて、セグメントマッチングアルゴリズムの目標は、図形 A における多重スケールセグメントと図形 B のそれとの最良の対応ペアを見つけることである。最良の対応ペアは多重スケールセグメント相違度 (後述する) の総和を最小化 (最適化) するセグメント対応を指す。従って、本アルゴリズムにより、2 つの図形間で対応する部分輪郭毎に最適スケールのセグメント対応が得られる。それ故、2 つの図形間で大きく変形していても妥当な対応を得ることが可能となる。さらに、従来の coarse-to-fine マッチング戦略と違って、最適対応を同時に求めることができる。一方、4.3 で述べたように、coarse-to-fine 戦略に基づく従来の多重スケールマッチングアルゴリズムでは、大局的なマッチング

が局所的 (より詳細) なマッチングを考慮せずに独立に実行されるため、こうしたアルゴリズムでは大局的に最適な対応を得ることができない。故に、4.3 で述べたような限界が生じる。

4.4.3 多重スケールセグメント相違度

ここで、2 図形 A, B 間での多重スケールセグメント相違度の定義で用いる記法を簡単化するために、$\mathbf{A}^{(0)}, \mathbf{B}^{(0)}$ のかわりに、各々 $\mathbf{A} = a_1, \ldots, a_i, \ldots, a_N, \mathbf{B} = b_1, \ldots, b_j, \ldots, b_M$ と表すことにする。さらに、$\mathbf{A}$ の $2n+1$ 個の連続するセグメント、$a_{i-2n}, a_{i-2n+1}, \ldots, a_i$ を $\mathbf{a}(i-2n|i)$ で表し、同様に、$\mathbf{B}$ の $2m+1$ 個の連続するセグメント $b_{j-2m}, b_{j-2m+1}, \ldots, b_j$ を $\mathbf{b}(j-2m|j)$ で表すこととする。

$\mathbf{a}(i-2n|i)$ と $\mathbf{b}(j-2m|j)$ がスケール $\sigma^{(h)}, \sigma^{(k)}$ で各々 $a_{i'}^{(h)}, b_{j'}^{(k)}$ に置換され得るとき、$\mathbf{a}(i-2n|i)$ と $\mathbf{b}(j-2m|j)$ の間の多重スケールセグメント相違度を $\psi\Big(\mathbf{a}(i-2n|i), \mathbf{b}(j-2m|j)\Big)$ で表し、次式で定義する。

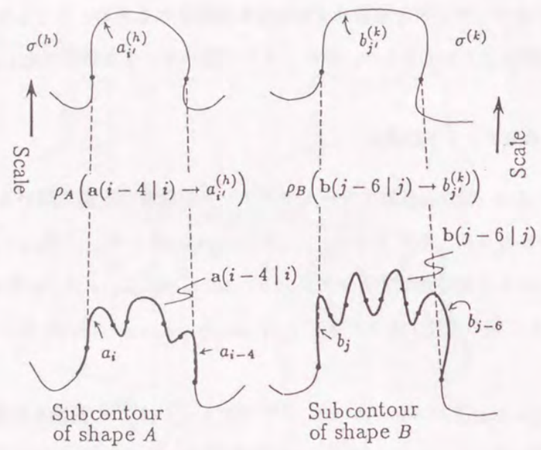
$$\psi\Big(\mathbf{a}(i-2n|i), \mathbf{b}(j-2m|j)\Big) = \delta(a_{i'}^{(h)}, b_{j'}^{(k)}) + \rho_A\Big(\mathbf{a}(i-2n|i) \to a_{i'}^{(h)}\Big) + \rho_B\Big(\mathbf{b}(j-2m|j) \to b_{j'}^{(k)}\Big). \tag{4.32}$$

ここで、$\delta(a_{i'}^{(h)}, b_{j'}^{(k)})$ は $a_{i'}^{(h)}$ と $b_{j'}^{(k)}$ の間の 1 対 1 のセグメント相違度である。$\rho_A\Big(\mathbf{a}(i-2n|i) \to a_{i'}^{(h)}\Big)$ および $\rho_B\Big(\mathbf{b}(j-2m|j) \to b_{j'}^{(k)}\Big)$ は $\mathbf{a}(i-2n|i)$ を $a_{i'}^{(h)}$ で $\mathbf{b}(j-2m|j)$ を $b_{j'}^{(k)}$ で各々置換するコストを表す。ここに、記号 $\to$ はその置換を意味する。図 4.11 に、4 つの連続する凹凸セグメント $\mathbf{a}(i-4|i)$ と 6 つの連続する凹凸セグメント $\mathbf{b}(j-6|j)$ との間の多重スケールセグメント相違度の例を示す。

式 (4.32) の意味は次のとうりである。もし、$\mathbf{a}(i-2n|i)$ と $\mathbf{b}(j-2m|j)$ が、上位スケールセグメント $a_{i'}^{(h)}, b_{j'}^{(k)}$ で各々置換されるならば、その 2 つのセグメントの対応付けが試行される。形式的には、2 組の連続セグメント $\mathbf{a}(i-2n|i), \mathbf{b}(j-2m|j)$ がマッチされると言うことにする。但し、この時、置換コスト ρ_A, ρ_B が 1 対 1 のセグメント相違度 δ に付加される。これらのコストは過剰な置換を抑制する働きをする。

式 (4.32) において、$n=0$ のときは、$\mathbf{a}(i-2n|i)$ は単に a_i に等しく、これは置換が実行されないことを意味し、それ故、$\rho_A = 0$ である。同様に、$m=0$ のときは、$\rho_B = 0$ である。さらに、$n=m=0$ のときは、上記セグメント相違度は、単に 1 対 1 のセグメント相違度に縮退する。すなわち、

$$\psi(a_i, b_j) = \delta(a_i, b_j) \tag{4.33}$$



$$\psi\Big(\mathrm{a}(i-4\mid i),\mathrm{b}(j-6\mid j)\Big) = \delta(a_{i'}^{(h)},b_{j'}^{(k)}) + \rho_A\Big(\mathrm{a}(i-4\mid i)\to a_{i'}^{(h)}\Big)+\rho_B\Big(\mathrm{b}(j-6\mid j)\to b_{j'}^{(k)}\Big)$$

図 4.11: 多重スケールセグメント相違度の例

換言すれば、式 (4.32) は、$n=0, m=0$ の特殊なケースとして 1 対 1 のセグメント相違度を含むセグメント相違度の一般的記述になっている。

上記置換について次の性質が成り立つ。

性質 4.9 $a_{i'}^{(u)}$ は、$a_i^{(0)}$ が凸 (凹) のときに限り、凸 (凹) となる。$b_{j'}^{(v)}$ についても同様である。

式 (4.32) において、多重スケールセグメント相違度を定義したが、1 対 1 のセグメント相違度 $\delta()$ 、置換コスト $\rho_A(),\rho_B()$ については正確な定義を与えていない。そこで、以下の 4.4.4, 4.4.5 でこれらについて詳述する。

4.4.4 1 対 1 のセグメント相違度

セグメント $a_i^{(h)}$ と $b_j^{(k)}$ が同じ極性のとき、2 つのセグメントの相違度を式 (4.34) で定義する。

$$\delta(a_i^{(h)},b_j^{(k)}) = \frac{|\theta_i^{(h)}-\theta_j^{(k)}|}{\theta_i^{(h)}+\theta_j^{(k)}}\left|\frac{l_i^{(h)}}{L_I^{(h)}}-\frac{l_j^{(k)}}{L_J^{(k)}}\right| \tag{4.34}$$

セグメントの極性が同じであるとは、2 つのセグメントが共に、凹あるいは凸であるということである。一つの凸 (凹) セグメントが一つの凸 (凹) セグメントとマッチするのは、直観的に妥当でないの

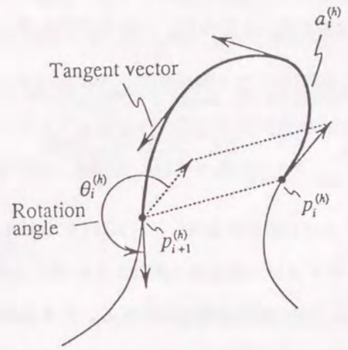


図 4.12: 回転角 $\theta_i^{(h)}$

で、極性が異なるときは、$\delta(a_i^{(h)},b_j^{(k)})=\infty$ とする。

式 (4.34) における $\theta_i^{(k)}$ は、図 4.12 に示すように $a_i^{(h)}$ に沿った接ベクトルの総回転角である。これは、セグメントの曲がり具合いを表す。$\theta_j^{(k)}$ も同様である。この回転角は正の値をとり、2π より大きくなり得ることに注意する。また 2 つの回転角の差を両者の和で割ることにより正規化している。$l_i^{(h)}$ は $a_i^{(h)}$ のセグメント長であり、$L_I^{(h)}$ はスケール $\sigma^{(h)}$ での平滑化曲線の全周囲長を表す。したがって、比 $l_i^{(h)}/L_I^{(h)}$ はそのセグメントの図形輪郭に対する寄与度を表わす。$l_j^{(k)}, L_J^{(k)}, l_j^{(k)}/L_J^{(k)}$ も同様である。

明らかに、上記相違度は 0 から 1 の値をとり、かつ相似変換 (回転、平行移動、拡大・縮小) に不変な尺度になっている。図形の構造は、本来、上記変換に不変であるため、この不変性は図形のマッチングの尺度として要請される必要条件と言える。

4.4.5 置換コスト

置換コストの妥当な定義を与えるために要請される条件として、次の 3 つが考えられる。

- 置換される凹 / 凸セグメントの数が大きい程、置換コストは大きくなるべきである。
- 置換されるセグメント内で、その隣接セグメントが異なっている程、置換コストは大きくなるべきである。
- 尺度の一貫性を満たすべく、コストは 1 対 1 のセグメント相違度と同じ単位 (次元) で定義されることが望ましい。

上記3つの条件を満たすべく、置換コストを置換される最下位スケールセグメント内の隣接する2つのセグメント相違度の総和として定義する。すなわち、$n \geq 1$ に対して、

$$\begin{aligned}\rho_A\Big(\mathrm{a}(i-2n|i) \to a_{i'}^{(h)}\Big) &= \lambda\Big(\delta(a_{i-2n}, a_{i-2n+1}) + \delta(a_{i-2n+1}, a_{i-2n+2}) + \cdots + \delta(a_{i-1}, a_i)\Big) \\ &= \lambda \sum_{s=i-(2n-1)}^{i} \frac{|\theta_s - \theta_{s-1}|}{\theta_s + \theta_{s-1}} \frac{|l_s - l_{s-1}|}{L_I} \qquad (4.35)\end{aligned}$$

とする。式(4.34)の1対1のセグメント相違度から、式(4.35)における $\frac{|\theta_s-\theta_{s-1}|}{\theta_s+\theta_{s-1}}\frac{|l_s-l_{s-1}|}{L_I}$ は隣接する2つのセグメント a_{s-1}, a_s 間のセグメント相違度に相当していることが分かる。但し、この場合、セグメントの極性は無視されている。ρ_B も同様に定義される。すなわち、$m \geq 1$ に対して、

$$\rho_B\Big(\mathrm{b}(j-2m|j) \to b_{j'}^{(k)}\Big) = \lambda \sum_{s=j-(2m-1)}^{j} \frac{|\theta_s - \theta_{s-1}|}{\theta_s + \theta_{s-1}} \frac{|l_s - l_{s-1}|}{L_J} \qquad (4.36)$$

とする。式(4.35),(4.36)における λ は重みパラメータで、通常は 1.0 とする。上で定義した置換コストが上記3つの条件を満たすことは容易に確かめられる。

4.4.6 最適化問題としての定式化

さて、問題は式(4.37)に示すように、上記セグメント相違度の総和が最小となる (i_w, j_w), $w = 1, 2, \ldots, W$ を見いだすことである。

$$\Psi(A, B) = \min_{\{(i_w, j_w)|w=1,\ldots,W\}} \sum_{w=1}^{W} \psi\Big(\mathrm{a}(i_{w-1}+1|i_w), \mathrm{b}(j_{w-1}+1|j_w)\Big) \qquad (4.37)$$

ここで、(i_w, j_w) は結果として得られた対応ペアのインデックスを表す。また、セグメントの巡回性より $i_0(j_0)$ は $i_W(j_W)$ に等しい。W は対応ペアの数で、明らかに $W \leq \min\{N, M\}$ となる。

例えば、$\mathbf{A} = a_1, a_2, \ldots, a_8$ と $\mathbf{B} = b_1, b_2, \ldots, b_{10}$ の間で、次の4つの対応が得られたとする: $(a_1, \mathrm{b}(10|2))$, $(\mathrm{a}(2|6), \mathrm{b}(3|5))$, (a_7, b_6), $(a_8, \mathrm{b}(7|9))$。このとき、$(i_1, j_1) = (1, 2)$, $(i_2, j_2) = (6, 5)$, $(i_3, j_3) = (7, 6)$, $(i_4, j_4) = (8, 9) = (i_0, j_0)$ となる。

一般に、セグメント置換における一貫性によれば、一つあるいは奇数個のセグメントだけが対応ペアとなり得るので、$i_w(j_w)$ と $j_{w-1}(j_{w-1})$ の差は必ず奇数になる。形式的には、$i_w(j_w)$ と $i_{w-1}(j_{w-1})$ には次の拘束関係が成り立たなければならない。

$$\begin{aligned} i_w - i_{w-1} &\equiv 2n+1 \pmod N \quad \text{かつ} \quad 0 \leq n \leq [(N-1)/2], \\ j_w - j_{w-1} &\equiv 2m+1 \pmod M \quad \text{かつ} \quad 0 \leq m \leq [(M-1)/2] \qquad (4.38)\end{aligned}$$

ここで、[.] はガウス記号を表わす。

上記最小化問題(式(4.37))の解法として、可能な多重スケールの全てのセグメントの組合せについて計算すること(しらみつぶし法)が考えられる。しかしながら、その組合せ数は、N, M の増加とともに指数関数的に増加するため現実的な時間で解くことが困難となる。ところが、式(4.37)の最小化問題は、全相違度が対応するセグメントの局所的な相違度の和として定義されているため、次節で述べるように、動的計画法を用いて効率良く解くことができる。

4.5 動的計画法による解法

$g(i_w, j_w)$ を a_{i_t} と b_{j_t}, $t = 1, 2, \ldots, w$ の最適対応が決定した時の累積された部分相違度とする。開始セグメントは未知であるので、4.2.4の変曲点対応と同様に、$g(i_w, j_w)$ を $1 \leq i_w \leq N$, $1 \leq j_w \leq 2M$ なる範囲で計算する。しかしながら、これでは3つのセグメント a_N, a_1, a_2 が b_j に対応するような a_i に関する回り込みを考慮していない。そこでこの回り込みを取り扱えるように、図4.13に示す $2N \times 2M$ のテーブルを用意する。このテーブルの第 (i_w, j_w) 要素に $g(i_w, j_w)$ の値を格納する。そして、テーブル上の (i_{w-1}, j_{w-1}) から (i_w, j_w) へのパスはセグメント $\mathrm{a}(i_{w-1}+1|i_w)$ と $\mathrm{b}(j_{w-1}+1|j_w)$ との対応を意味する。したがって、$1 \leq i_{w-1} \leq N < i_w \leq 2N$ かつ $1 \leq j_{w-1} < j_w \leq 2M$ を満たす (i_{w-1}, j_{w-1}) から (i_w, j_w) へのパスを設けることにより上記の回り込みを取り扱うことが可能となる。

今、初期値を式(4.39)で与える。

$$g(i_w, j_w) = 0, \qquad i_w = 1, 2, \ldots, N^{(0)}, j_w = 1, 2, \ldots, 2M^{(0)} - 1. \qquad (4.39)$$

そして、$i_w = N+1, N+2, \ldots, 2N$ かつ $j_w = 1, 2, \ldots, 2M$ なる i_w, j_w に対して、$g(i_w, j_w)$ を次式により計算する。

$$g(i_w, j_w) = \min_{i_{w-1}, j_{w-1}} \Big\{ g(i_{w-1}, j_{w-1}) + \psi\Big(\mathrm{a}(i_{w-1}+1|i_w), \mathrm{b}(j_{w-1}+1|j_w)\Big)\Big\} \qquad (4.40)$$

式(4.40)は、$g(i_{w-1}, j_{w-1})$ がいかなる経路を経て決定されても、$g(i_w, j_w)$ は $g(i_{w-1}, j_{w-1})$ とセグメントの局所相違度 $\psi\Big(\mathrm{a}(i_{w-1}+1|i_w), \mathrm{b}(j_{w-1}+1|j_w)\Big)$ の和で表せる。という動的計画法における最適性の原理に基づいている。

ところで、先に述べたスケール変化に伴う凹凸の変化の一貫性より、非負の整数 n_w, m_w を用いて、$i_{w-1} = i_w - (2n_w+1)$, $j_{w-1} = j_w - (2m_w+1)$ と書ける。それゆえ、式(4.40)は式(4.41)のように書き換え得る。

$$\begin{aligned} g(i_w, j_w) &= \min_{n_w, m_w} \Big\{ g(i_w - (2n_w+1), j_w - (2m_w+1)) \\ &\qquad + \psi\Big(\mathrm{a}(i_w - 2n_w|i_w), \mathrm{b}(j_w - 2m_w|j_w)\Big)\Big\} \qquad (4.41)\end{aligned}$$

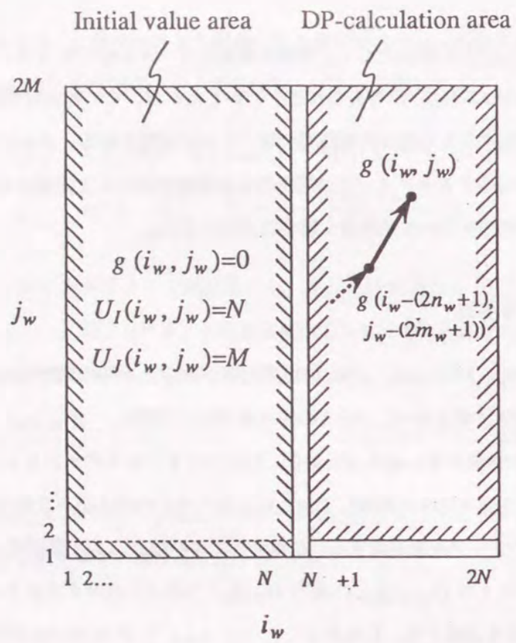


図 4.13: 最適スケールセグメントマッチングのための DP テーブル

ここで、n_w, m_w は式 (4.38) を満たすべく式 (4.42) により制約する。

$$\begin{cases} 0 \le n_w \le [(i_w - 1)/2] \\ 0 \le m_w \le [(j_w - 1)/2] \\ t_1 = t_2 = 0 \quad \text{または} \quad t_1, t_2 > 0 \end{cases} \tag{4.42}$$

ここで、[.] は、ガウス記号を表す。t_1, t_2 はそれぞれの輪郭図形において対応づけられずに残っているセグメントの総数を表し、式 (4.43),(4.44) で各々定義される。

$$t_1 = U_I\Big(i_w - (2n_w + 1), j_w - (2m_w + 1)\Big) - (2n_w + 1) \tag{4.43}$$

$$t_2 = U_J\Big(i_w - (2n_w + 1), j_w - (2m_w + 1)\Big) - (2m_w + 1). \tag{4.44}$$

そして、式 (4.41) を最小にする最適パラメータ n_w^*, m_w^* を用いて、$U_I(i_w, j_w)$,$U_J(i_w, j_w)$ を、式 (4.45),(4.46) により計算しておく。

$$U_I(i_w, j_w) = U_I\Big(i_w - (2n_w^* + 1), j_w - (2m_w^* + 1)\Big) - (2n_w^* + 1) \tag{4.45}$$

$$U_J(i_w, j_w) = U_J\Big(i_w - (2n_w^* + 1), j_w - (2m_w^* + 1)\Big) - (2m_w^* + 1) \tag{4.46}$$

なお、$1 \le i_w \le N$ なる i_w に対しては、$U_I(i_w, j_w) = N$, $U_J(i_w, j_w) = M$ とする。U_I,U_J は各々 a_i,b_j についての重複対応を防止するために用いている。すなわち、$U_I\Big(i_w - (2n_w + 1), j_w - (2m_w + 1)\Big)$ は図 4.13 に示したテーブル上で $\Big(i_w - (2n_w + 1), j_w - (2m_w + 1)\Big)$ から (i_w, j_w) へのパスが存在可能か否かを動的に判定する。このとき、t_1 が 0 より大きいかまたは等しくなっている場合、パスが存在可能となる。U_J についても同様である。

性質 4.9 を考慮すると、式 (4.41) の $g(i_w, j_w)$ は $i_w = N + 1, \ldots, 2N$ に対して式 (4.47) で定義される j_w についてのみ計算すれば良いことが分かる。

$$j_w = \begin{cases} 3 + c, 5 + c, \ldots, 2M - 1 + c, & (i_w \text{ が奇数のとき}) \\ 2 - c, 4 - c, \ldots, 2M - c, & (i_w \text{ が偶数のとき}) \end{cases} \tag{4.47}$$

ここで、c は定数で、a_1 と b_1 の極性が等しいときは 0 をとり、それ以外は 1 をとる。

a_{i_w} と $b_{j_w}, w = 1, 2, \ldots, W$ との重複のない完全なセグメント対応は、t_1 と t_2 の両者が同時に 0 になった時に得られる。ゆえに、$g(i_w, j_w)$ の計算の途中で、最適パラメータ n_w^*, m_w^* を各々テーブル $\mathbf{G_n}(i_w, j_w)$, $\mathbf{G_m}(i_w, j_w)$ に記憶し、また、t_1 および t_2 が 0 になったときの i_w, j_w および $g(i_w, j_w)$ の値も記憶しておく。

全ての $g(i_w, j_w)$ の計算終了後、最適なセグメント対応 $\Big((a_{\hat{i}_w} \sim a_{\hat{i}_{w-1}+1}), (b_{\hat{j}_w} \sim b_{\hat{j}_{w-1}+1})\Big)$, $w = 1, 2, \ldots, W$ を次のようにして求める。すなわち、t_1 および t_2 が 0 になったときの $g(i_w, j_w)$ の最小値とそのときの $(\hat{i}_W, \hat{j}_W)$ から、残りの $(\hat{i}_w, \hat{j}_w)$ の値は次式に示す $\mathbf{G_n}$ および $\mathbf{G_m}$ のテーブル参照により計算する。

$$\begin{cases} \hat{i}_w = \hat{i}_{w+1} - (2\mathbf{G_n}(\hat{i}_{w+1}, \hat{j}_{w+1}) + 1) \\ \hat{j}_w = \hat{j}_{w+1} - (2\mathbf{G_m}(\hat{i}_{w+1}, \hat{j}_{w+1}) + 1) \end{cases} \tag{4.48}$$

$$\text{for} \quad w = W - 1, W - 2, \ldots, 1.$$

以上の計算により、各部分輪郭毎に最適スケールのセグメントがマッチングされることになる。

動的計画法 (DP) は非線形の最適化問題、特に組合せ最適化問題に対する有力な解法として知られており、パターン認識、コンピュータビジョンにおける各種マッチング手法に数多く適用されている。その代表的な研究例として、音声認識 [76]、ステレオビジョン [74][77]、画像の位置合わせ [78]、図形の変形解析 [79]、等がある。(最近では、第 3 章で述べたように、エネルギー最小化問題に対する DP による解法が Amini[40] および筆者ら [36] によって試みられている。)

ところが、文献 [74] を除く、これら従来の DP マッチングは、多重スケールマッチングの問題を取り扱っていない。Baker[74] らは、多重スケールマッチングを取り扱ってはいるものの、4.3.2 で述べたように、coarse-to-fine マッチングの各レベルにおいて独立に DP マッチングを実行してお

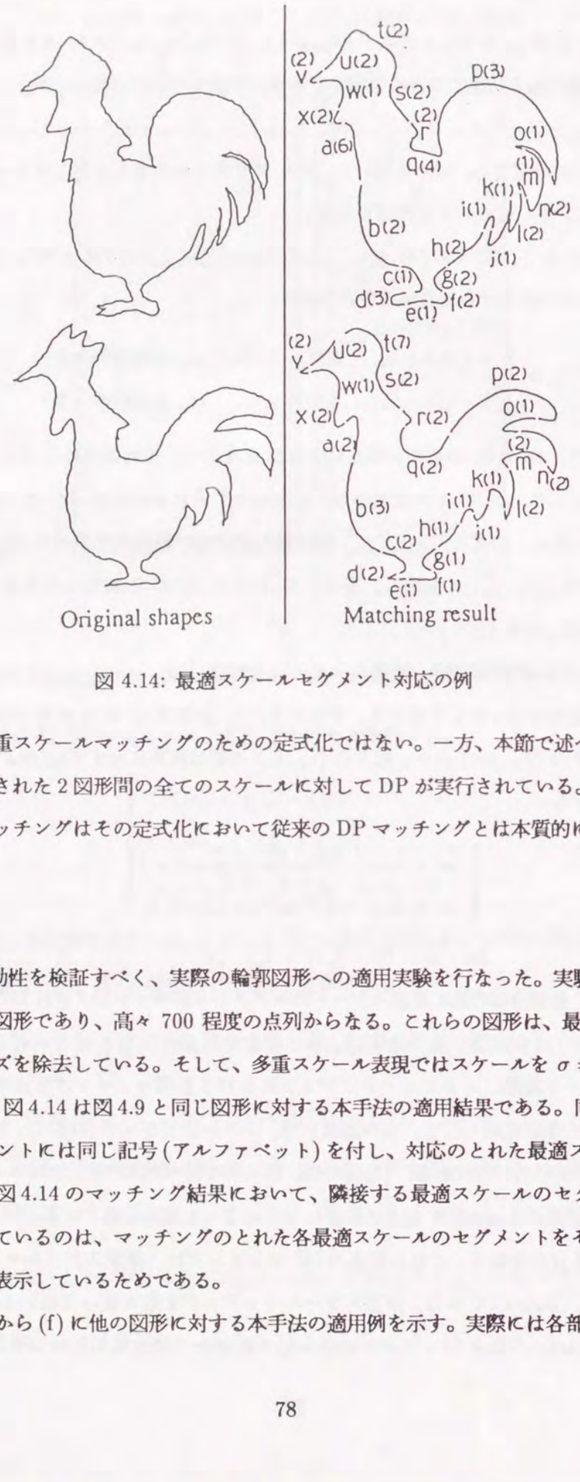


図 4.14: 最適スケールセグメント対応の例

り、その意味で多重スケールマッチングのための定式化ではない。一方、本節で述べた定式化では、多重スケール表現された 2 図形間の全てのスケールに対して DP が実行されている。すなわち、本節で提案した DP マッチングはその定式化において従来の DP マッチングとは本質的に異なっている。

4.6 実験

提案手法の有効性を検証すべく、実際の輪郭図形への適用実験を行なった。実験で用いた図形は手書きによる輪郭図形であり、高々 700 程度の点列からなる。これらの図形は、最初 $\sigma = 1$ で平滑化し、手書きノイズを除去している。そして、多重スケール表現ではスケールを $\sigma = 1$ から 20 までの整数値とした。 図 4.14 は図 4.9 と同じ図形に対する本手法の適用結果である。同図において対応する各凹凸セグメントには同じ記号 (アルファベット) を付し、対応のとれた最適スケールを () 内に記してある。尚、図 4.14 のマッチング結果において、隣接する最適スケールのセグメント同士に切れや重なりが生じているのは、マッチングのとれた各最適スケールのセグメントをそれらの $x-y$ 座標に基づいて重畳表示しているためである。

図 4.15 の (a) から (f) に他の図形に対する本手法の適用例を示す。実際には各部分輪郭毎に最適

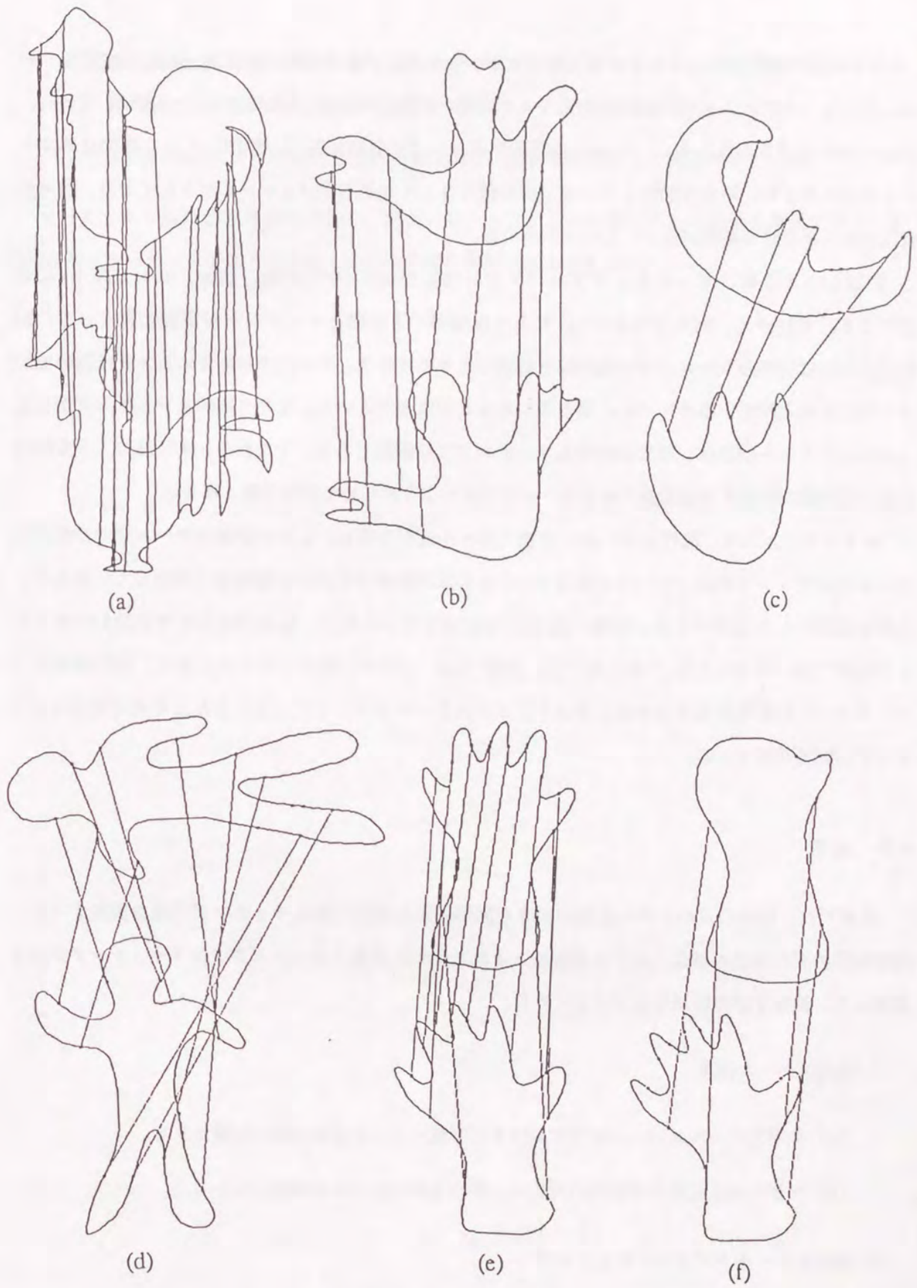


図 4.15: マッチング結果

スケールの輪郭同志がマッチングしているが、マッチング結果は最下位スケール上で表示している。そして、セグメント(部分輪郭)対応をその両端の変曲点同志を直線で結ぶことで表示している。同図の例から明らかなように、2つの輪郭図形において部分的に凹凸の類似した部分輪郭は細かくマッチングがとれ、あまり類似していない部分輪郭はより大局的なマッチングがとれており、我々の直観に合った良好な結果が得られているといえる。

前述したように、本マッチングアルゴリズムは相似変換(平行移動、回転、拡大／縮小)に不変である。それゆえ、例えば図4.15(d)にような図形でも正確なマッチングが実現できている。図4.15(e),(f)は、手のシルエットの輪郭線を意図したものであり、手が広げられているときは指単位でマッチングがとれているのに対し、握り拳になると手首と手という、より大局的なマッチングがとれている。こうした結果は、単に非線形なマッチングでは困難である。すなわち、本手法のような凹凸の変化を階層的に捉えた多重表現によるマッチングによりはじめて実現可能となる。

本アルゴリズムは、離散スケールの多重スケール表現を用い、しかも多重スケール間での最適スケールのセグメント対応をすべての組合せの中から動的計画法を用いて効率良く実行しているため、高速な処理として実現できた。実際、図4.15の実験で用いた例で、従来手法[59]では数分のマッチング時間であったのに対し、本手法では、数秒(Sun 4/260C)程度であった。また、動的計画法はハードウェア化が容易であるため、本アルゴリズムをハードウェア化することにより実時間でのマッチングが実現可能となる。

4.7 結言

本章では、効率の良いしかも変形の大きな図形間にも適用可能なマッチング手法を実現すべく、図形輪郭の凹凸構造の多重スケール表現法とそれを用いた多重スケールセグメントマッチング手法を提案した。提案手法の特長を以下にまとめる。

1. 多重スケール表現

   (a) 曲率スケールスペース表現における計算量・データ量を大幅に削減できる。

   (b) スケール変化に伴う凹凸の変化の一貫性を満たすことを保証している。

2. 多重スケールセグメントマッチング

   (a) 変曲点位置のみに着目するのではなく、変曲点間の凹凸セグメントをマッチングの単位としており、部分輪郭の類似性を反映したマッチングを実現している。

   (b) 部分輪郭毎に最適なスケールでのマッチングを実現しているため、従来手法に比べより変形の大きな図形間にも適用できる。

   (c) 動的計画法により最適解を求めているので、高速な処理として実現できている。

さらに、本手法は閉図形だけでなく、始点と終点が固定された開図形への拡張も容易であり、汎用的な形状マッチング手法として多くの応用分野に利用可能と考えられる。

第 5 章

輪郭形状からの凹凸構造の一般化

5.1 序言

モデルベースの画像理解においては、モデルの利用法もさることながら、そのモデルを如何に得るかということも無視できない重要課題である。とりわけ、高度な認識・理解システムにおいては動的にその性能を向上させる機能が必要となる。これはすなわち、学習によるモデル獲得、およびモデル更新を意味する。こうした学習機能は、我々人間が普段当たり前のように行なっているが、計算機上で実現するのは極めて難しい問題である。一方、人工知能研究において、これまで学習に関する研究が盛んに行なわれているが、これらはすべて記号概念をベースにした概念学習を取り扱っている。それ故、これらの成果を直接パターン形状のモデル獲得に適用することはできない。

最近、2 次元図形を対象としたモデル獲得の研究が若干報告されている [12][13][15]。しかしながら、任意形状に適用できない、対象の構造に関する事前知識を必要とする、あるいは得られたモデルを可視化することができない等の問題点を抱えており、有効な成果を出すには至っていない。こうした問題は、主にこれら従来手法がパターンを記号概念、あるいは数値で記述し、記号もしくは数値上で一般化を行なっていることに起因している。

そこで、本章および次章では 2 次元図形を対象として、従来手法とは全く異なるアプローチによるパターン形状の一般化 [80][81]、およびそれを用いた形状モデル自動獲得法 [82][83][84][85] を提案する。まず、本章では、2 次元図形を記号、数値ではないパターン形状として直接一般化することの重要性を論じ、次いでそのための具体的方策を与える。この一般化においては、前章で提案した最適スケールの凹凸構造マッチングを用いている。提案した一般化手法を評価すべく、実際の輪郭図形への適用例を示すと同時に、判別分析の基準であるクラス分離度による定量的な解析を行なう。

Gray-scale Image

⇓

EDGE DETECTION

⇓

CONTOUR FOLLOWING

⇓

SLS EXTRACTION & SEGMENTATION

⇓

Semantic Network

図 5.1: 意味ネットワークによる概念記述過程

5.2 一般化処理の概要

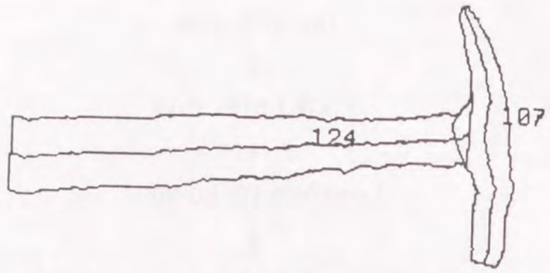
5.2.1 問題の所存

本章の冒頭で述べたように、パターン形状を取り扱った一般化の研究はその重要性にも関わらず、若干の報告がなされているに過ぎない。そして、それらは Connell & Brady の研究 [12][13] および Segen の研究 [15] に代表される。そこで、まずこの両者のアプローチを概説するとともにアプローチの問題点を明らかにする。

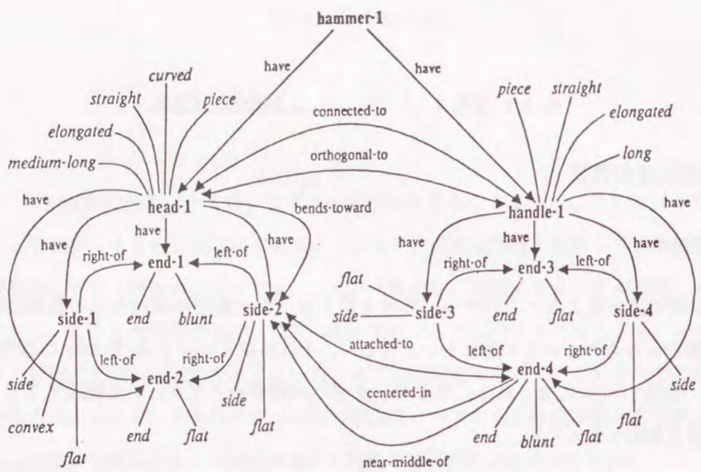
(1) 意味ネットワーク上での一般化

Connell らは 2 次元形状を対象に同一クラスに属す複数の形状サンプルからそれらに共通する概念構造記述を得ることを試みた。但し、一般の画像から概念記述を直接得るのは困難なので、与えられた濃淡画像から着目対象物体がすでに抽出されていることを前提とする。Connell らの方法では、まず抽出された物体形状から意味ネットワークによる概念記述を行なう。概念記述過程では、濃淡画像からエッジ抽出、輪郭線追跡を行なった後、Smoothed Local Symmetry(SLS)[11] を用いて 2 次元形状の部分構造への分割を行ない、次いで、そうして分割された各パーツの相互関係を意味ネットワーク上に記述する (図 5.1)。

SLS は Brady らによって先に提案された手法で、物体形状の局所対称軸を求め、それらを対称軸の滑らかさの尺度で統合し、より大局的な対称軸を得る。尚、Brady らが用いた局所対称軸の定義は、中心軸変換 (Symmetric Axis Transform: SAT)[86][87] として知られる形状の芯線の定義とは異なっており、形状の対称軸を過不足なく抽出する手段となり得る。



(a)SLS の抽出結果



(b) 意味ネットワークによる概念記述結果

図 5.2: 意味ネットワークによる概念記述 (文献 [12] による)

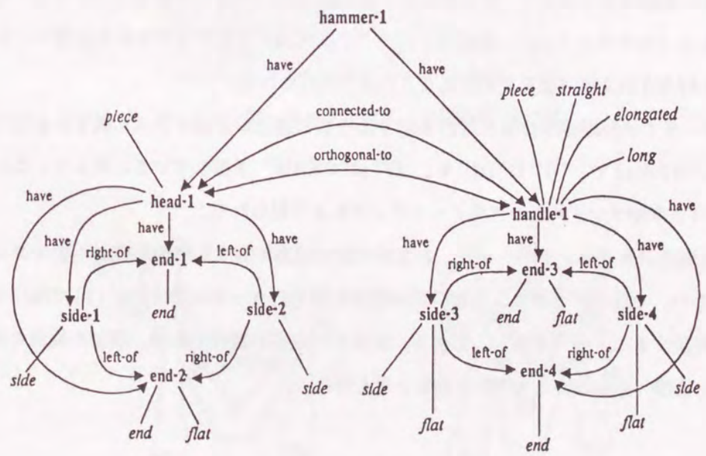
SLS によって部分分割された各パーツは意味ネットワーク上にそれらの相互関係が記述される。但し、この際、対象の構造に関する知識が必要となる。図 5.2(a) に tack-hammer の SLS 抽出結果を、図 5.2(b) にその概念記述の例を示す。図 5.2(b) では、"tack-hammer は head-1 と handle-1 から成り、さらに head-1 は end-1 と end-2 を有し、end-1 はその先が blunt であって、end-2 では flat である..." といった記述を意味する。

同一クラスに属す複数のサンプルが与えられたとき、各サンプルに対し各々の概念記述が生成される。そして、得られたネットワーク群が逐次照合され、より簡約化されたネットワークが生成させる。すなわち、Connell らのアプローチでは、一般化は意味ネットワークの簡約化と言い替えること

ができる。尚、この簡約化においては、ネットワーク上の各述語は一旦数値情報に変換され[†]、その上で一般化が行なわれる。図 5.3(b) は図 5.3(a) の 3 つのハンマーの例から一般化された概念記述の例である。

図 5.2(b) と比べるとネットワークがより単純になっていることがわかる。このアプローチは従来独立に行なわれてきた画像処理と概念学習とを結合した最初の研究として非常に有意義であるが、次のような問題を抱えている。

(a) 学習サンプル



(b) 意味ネットワーク上での一般化結果

図 5.3: Connell らのアプローチ (文献 [12] による)

[†]実際にはグレイコードに類似した数値化がなされる。

(i) 形状を部分分割して記述するため、対象が一意に部分分割可能なものに限定される。

(ii) 対象の構造に関する事前知識を必要とする。例えば、図 5.2 の例では"ハンマーはヘッドとハンドルから成る"といった構造に関する知識は予め分かっているものとしており、任意のサンプル形状が与えられるわけではない。

(iii) 一般化では意味ネットワークの照合、すなわち、グラフの (準) 同型問題に帰着されるため複雑な対象には適用困難[†]となる [88]。

(iv) 意味ネットワークによる概念記述では、一般化された結果を可視化することができない。この可視化は得られたモデル (一般化記述) の妥当性を容易にするのに必要な機能である。

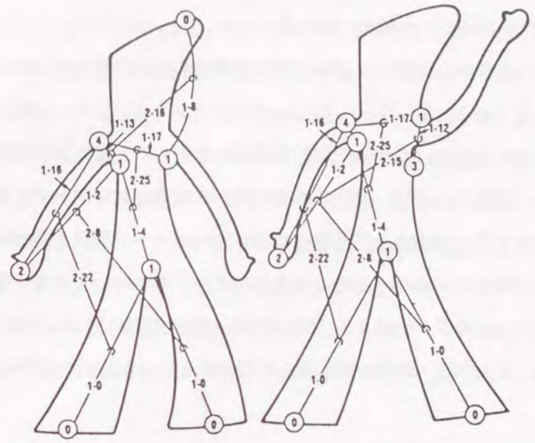
(2) 多重レベルグラフ上での一般化

Segen は、同じく 2 次元輪郭形状を対象に、その構造記述のための多重レベルグラフという独自のネットワークを提案している。まず図 5.4(a) のような輪郭形状 (左側はクラス"DRAW" を、右側はクラス"HI" を各々表している) に対して図 5.4(a) 中の円で示された特徴点を抽出し、それらの曲率等の属性および特徴点相互間の位置関係等の関係記述を行なう。この関係記述は、メタレベルの関係も含む。例えば、図 5.4(a) の"1-" でラベルづけされた直線は第一レベルの関係を示し、"2-" のそれは第二レベルの関係を表している。そして、これらは図 5.4(b) のような多重レベルグラフにより表現される。多重レベルグラフ上での一般化は、同一クラスに属す複数サンプルから多重レベルグラフの点および枝に付与された確率値を更新することにより実行される。

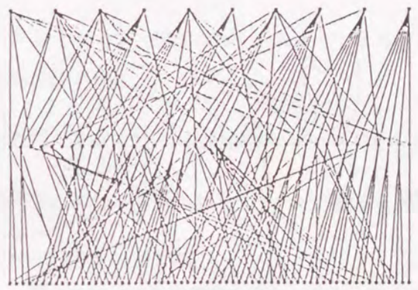
このアプローチでは認識結果は図 5.4(c) に示すように特徴点がどのクラスに属すかを出力することで得られる。図 5.4(c) で、"H" は"HI" を、"D" は"DRAW" を指している。そして、この認識はモデル (グラフ) と未知サンプルとのグラフマッチングにより得られる。

この方法は形状のセグメンテーション、および対象の構造に関する事前知識が不要であるという特長があるものの、やはりグラフによる部分間の関係記述を行なっているため、(1) で掲げた問題点の (iii)(iv) を解決することができない。しかも、特徴点の定義が曖昧であり、最初に抽出される特徴点により学習の性能・効率がかなり左右されると考えられる。

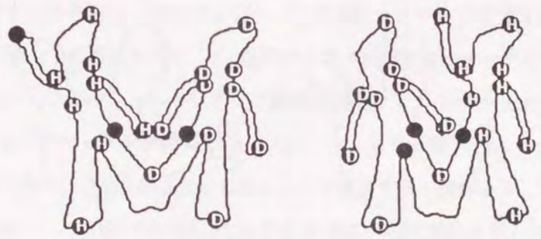
[†]グラフの (準) 同型問題は NP 完全の典型的問題であり多項式時間で解き得るアルゴリズムは存在しない。



(a) 輪郭形状の特徴抽出および特徴間の関係抽出例



(b) 多重レベルグラフによる関係記述の例



(c) 認識の例

図 5.4: Segen のアプローチ (文献 [15] による)

(3) 多重スケールの凹凸構造上での一般化: 新アプローチ

(1),(2) で述べてきたように、従来のアプローチは基本的には形状をプリミティブとプリミティブ間の関係で記述するものである。従って、一般化の際のグラフマッチングの問題点が共有され、十分な成果には至っていない。そこで、こうした問題を解決するために、従来手法の延長ではない新たなアプローチを提案する。提案手法では、対象の構造に関する事前知識を用いずに同一クラスに属する複数の輪郭図形サンプルから図形輪郭の凹凸構造に着目してパターン形状を直接一般化する。本手法は、記号的概念あるいは数値を一般化するのではなくパターン形状そのものを一般化するという点で従来手法と本質的に異なっている。提案するモデル記述 (一般化図形と呼ぶ) は得られたモデルの人間による理解を容易にし、また学習・認識に適したものであるといった特長を有する。

(a) 同一クラスに属す輪郭形状サンプルの例

(b) 一般化されたサンプルの例

図 5.5: 本章で提案するアプローチによる一般化

例えば、図 5.5(a) の 2 図形が同一クラスのパターンとして与えられたとしよう。我々人間はそれらを異なる視野スケールで観察し、2 図形に共通する最適スケールの凹凸構造を抽出し、かつ原形状を反映した図 5.5(b) のような簡単化された図形を各々想像する。本手法は、まさに上記過程をパターン形状の一般化と定義する。そして、この一般化された図形を一般化図形と呼ぶこととする。

本手法は図形の一般化は図形の簡単化であるという概念に基づいている。この概念は直観的に妥当であり、かつ記号世界の一般化において通常用いられている条件削除規則 [9] の自然なアナロジーとなっている。図形の簡単化とは形状の近似を意味するのではなく同一クラスに属するサンプル図形間に共通する最適スケールの凹凸構造の抽出を意味する。最適スケールの凹凸構造の抽出は、一般に容易な問題ではないが、第 4 章で提案した多重スケール凹凸構造マッチングを適用することにより実現可能になる。

5.2.2 一般化の処理過程

今、同一クラスに属す N 個のサンプル図形が与えられたとしよう。このとき、提案する一般化処理は次の 3 つのステップから構成される。

(Step 1) サンプル図形相互間で多重スケール凹凸構造マッチングを行なう。

(Step 2) 各サンプル図形毎に、step.1 で得られたマッチング結果を統合する。

(Step 3) 統合結果から一般化図形を作成する。

図 5.6 は上記過程を図示したものである。図 5.6 で $S_1, S_2, \ldots, S_N$ は同一クラスに属す輪郭形状サンプルを表し、$G_1, G_2, \ldots, G_N$ はそれら原サンプルから得られた各一般化図形を表している。step 1 では第 4 章で提案した多重スケールセグメントマッチング手法をそのまま適用すれば良い。従って、N 個のサンプルの場合、全体で $N(N-1)/2$ 回のマッチングが実行されることになる。

今、あるサンプル S_1 に着目すると、S_1 は同一クラスに属す他のサンプル図形 $S_2, S_3, \ldots, S_N$ に対し、各々、$C_{1,2}, C_{1,3}, \ldots, C_{1,N}$ のマッチング結果を得る。これらのマッチングは各々独立に実行されるため、明らかに、マッチング結果 $C_{1,2}, C_{1,3}, \ldots, C_{1,N}$ は一般に異なっている。従って、N 個のサンプルに共通する凹凸構造を得るためにはこれらを統合する必要がある。$S_2, S_3, \ldots, S_N$ についても同様である。従って、共通の凹凸構造を有する N 個の一般化結果が得られることになる。

一般化のための統合処理は次の 3 つの手続きから成る。

(Step 1) すべてのサンプル図形に対して、セグメント対応結果から半順序集合を作る。

(Step 2) すべてのサンプル図形に対して、得られた半順序集合における極大要素を見い出す。

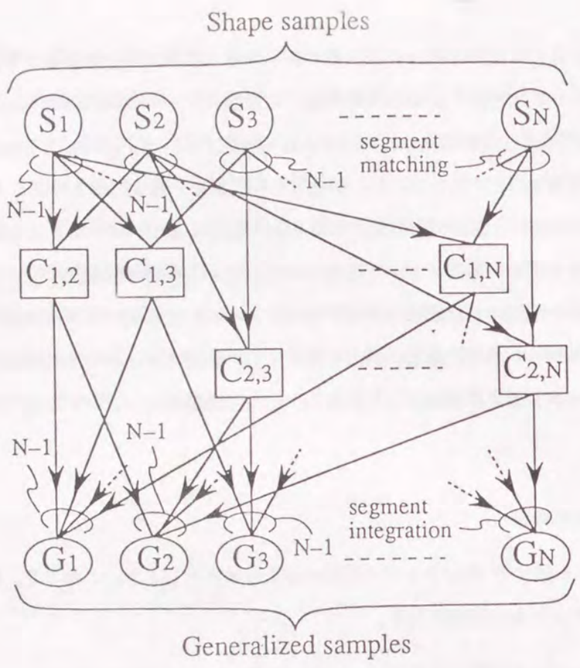


図 5.6: 一般化図形生成過程

(Step 3) 各サンプル図形毎に得られた極大要素間で無矛盾性を調べる。

上記ステップの直感的説明は次のとうりである。step 1, step 2 により、すでに得られたセグメント対応結果から最も簡単な凹凸構造が抽出される。但し、提案する一般化はすでに述べたように凹凸の数を単に最小化するのではなく、すべてのサンプル図形に対して対応する凹凸構造のなかで最も簡単な凹凸構造を抽出する。従って、こうした共通の凹凸構造を得るために、step 3 で対応関係の無矛盾性をチェックしている。

以下の節では上記ステップの詳細について述べる。

5.3 一般化のためのセグメント対応結果の統合

5.3.1 分割セグメント集合族

多重スケールセグメントマッチングから得られた最適対応結果は、第 4 章で述べたように、形式的には最下位スケールを用いて記述することができる。それ故、統合アルゴリズムも最下位スケールセグメントを用いて記述することができる。第 4 章では、図形輪郭を最下位スケールのセグメント系

列で表したが、ここでは、セグメントの順序そのものは重要ではないので、系列ではなく集合として取り扱うことにする。すなわち、$\mathbf{A}_i$ を第 i 番目のサンプル図形 S_i の最下位スケールセグメントを要素とする集合とすると、

$$\mathbf{A}_i = \{a_{i,t} \mid t = 1, 2, \ldots, T_i\} \tag{5.1}$$

と書ける。ここで、T_i は最下位スケールでの S_i のセグメント総数を表す。また、$a_{i,t}$ は N 個のサンプル図形の中の第 i 番面のサンプルの第 t 番目の (最下位スケール) セグメントに相当する。

N 個のサンプル図形の場合、$N(N-1)$ 回のマッチングが実行され、各サンプル図形毎に $N-1$ 個のマッチング結果を得る。いま、$\mathbf{A}_i$ に着目すると、これらのマッチングにより、$\mathbf{A}_i$ は対応の単位である偶数個の部分集合に分割される。そこで、$\mathbf{a}_{i,w}$ を $\mathbf{A}_i$ の第 w 番目の対応単位 (部分集合) とすると、明らかに、$\mathbf{a}_{i,w}$ は奇数個の最下位スケールセグメントから成る。この部分集合の集まり (集合族) を分割セグメント集合族と呼ぶこととする。より厳密な定義は以下のとうりである。

定義 5.1 (分割セグメント集合族 (PSFS: Partitioned Segment Family of Sets)) *2 つのサンプル図形 S_i と $S_j (j \neq i)$ の間で $n_{i,j}$ 個の対応結果が得られたとすると、$\mathbf{A}_i$ と $\mathbf{A}_j$ はともに、$n_{i,j}$ 個の対応の単位 (部分集合) に分割される。このとき、$\mathbf{A}_i$ の $\mathbf{A}_j$ による PSFS は、その分割された部分集合の集まり (集合族) として定義され、$\mathcal{P}_{i,j} = \{\mathbf{a}^j_{i,w} \mid w = 1, 2, \ldots, n_{i,j}\}$ で表わされる。同様に、$\mathbf{A}_j$ の $\mathbf{A}_i$ による PSFS は、$\mathcal{P}_{j,i} = \{\mathbf{a}^i_{j,w} \mid w = 1, 2, \ldots, n_{i,j}\}$ で定義される。*

$\mathbf{a} \in \mathcal{P}_{i,j}$ は最下位スケールにおける図形輪郭上の 1 つあるいはそれ以上の奇数個の連続するセグメント $a_{i,t}, a_{i,t+1}, \ldots, a_{i,t+2n}$ から成る。明らかに、$\mathcal{P}_{i,j}$ は $\mathbf{A}_i$ の非空でかつ互いに素な部分集合で、それらの和集合は $\mathbf{A}_i$ そのものである。すなわち、PSFS, $\mathcal{P}_{i,j}$, は以下の性質を満たさねばならない。

性質 5.1 $\bigcup_{\mathbf{a} \in \mathcal{P}_{i,j}} \mathbf{a} = \mathbf{A}_i$

性質 5.2 すべての $\mathbf{a} \in \mathcal{P}_{i,j}$ に対して、$\mathbf{a} \neq \phi$

性質 5.3 もし、$\mathbf{a}, \mathbf{a}' \in \mathcal{P}_{i,j}$ かつ $\mathbf{a} \neq \mathbf{a}'$ ならば、$\mathbf{a} \cap \mathbf{a}' = \phi$

一般に上記性質を満たす集合 (族) は集合 $\mathbf{A}_i$ の分割と呼ばれる。それ故、$\mathbf{P}_{i,j}$ を分割セグメント集合族と呼ぶこととした。ここで、$\mathcal{P}_{i,j}$ と $\mathcal{P}_{j,i}$ の要素間には 1 対 1 対応 (全射) が存在することに注意する。すなわち、$\mathbf{a}^j_{i,w}$ は $\mathbf{a}^i_{j,w}$ と対応関係をもつ。PSFS の簡単な例を以下に与える。

[例 5.1: 分割セグメント集合族の例] $\mathbf{A}_1 = \{a_{1,1}, a_{1,2}, \ldots, a_{1,8}\}$ および、$\mathbf{A}_2 = \{a_{2,1}, a_{2,2}, \ldots, a_{2,8}\}$ とし、両者の間の最適対応が、$\mathbf{a}^2_{1,1} = \{a_{1,1}\}$, $\mathbf{a}^2_{1,2} = \{a_{1,2}, a_{1,3}, a_{1,4}\}$, $\mathbf{a}^2_{1,3} = \{a_{1,5}\}$, $\mathbf{a}^2_{1,4} =$

$\{a_{1,6}, a_{1,7}, a_{1,8}\}$ および、$\mathbf{a}^1_{2,1} = \{a_{2,3}, a_{2,4}, a_{2,5}\}$, $\mathbf{a}^1_{2,2} = \{a_{2,6}\}$, $\mathbf{a}^1_{2,3} = \{a_{2,7}\}$, $\mathbf{a}^1_{2,4} = \{a_{2,8}, a_{2,1}, a_{2,2}\}$ であったとすると、$\mathcal{P}_{1,2}, \mathcal{P}_{2,1}$ は各々、

$$\begin{aligned}\mathcal{P}_{1,2} &= \{\{a_{1,1}\}, \{a_{1,2}, a_{1,3}, a_{1,4}\}, \{a_{1,5}\}, \{a_{1,6}, a_{1,7}, a_{1,8}\}\} \\ \mathcal{P}_{2,1} &= \{\{a_{2,3}, a_{2,4}, a_{2,5}\}, \{a_{2,6}\}, \{a_{2,7}\}, \{a_{2,8}, a_{2,1}, a_{2,2}\}\}\end{aligned}$$

となる。

5.3.2 極大集合族

N サンプルの場合、各サンプル図形 $S_i, 1 \le i \le N$ に対して、$N-1$ 個の PSFS($\mathcal{P}_{i,j}, 1 \le j \le N, j \ne i$) が得られる。これらの PSFS は一般に異なっている。いま j に関するそれらの和集合:

$$\mathbf{Q}_i = \bigcup_{1 \le j \le N, j \ne i} \mathcal{P}_{i,j} \tag{5.2}$$

を考える。すると $\mathbf{Q}_i$ は以下の性質を満たすことが容易に確かめられる。すなわち、$q, q', q'' \in \mathbf{Q}_i$ に対して、

性質 5.4 *(反射律)* すべての $q \in \mathbf{Q}_i$ に対して、$q \subseteq q$

性質 5.5 *(推移律)* $q \subseteq q'$ かつ $q' \subseteq q''$ ならば $q \subseteq q''$

性質 5.6 *(反対称律)* $q \subseteq q'$ かつ $q' \subseteq q$ ならば $q = q'$

上記 3 つの性質を満たす関係は、一般に半順序と呼ばれる。それ故、集合族 $\mathbf{Q}_i$ は集合の通常の包含関係 ($\subseteq$) に関する半順序集合 (partially ordered set) $(\mathbf{Q}_i, \subseteq)$ となることがわかる。

$\mathbf{Q}_i$ の各要素は連続する凹凸セグメントを要素とする集合であるから、$\mathbf{Q}_i$ における半順序は凹凸構造における一般化の半順序を表している。すなわち、この半順序関係を用いて凹凸構造の一般化を次のように定義することができる。

定義 5.2 (凹凸構造の一般化の半順序) $q, q' \in \mathbf{Q}_i$ に対して、$q \subseteq q'$ が成り立つときに限り q' は q の一般化であるという。このとき、$\subseteq$ は一般化の半順序と呼ばれる。

前述した形状の一般化の概念から、形状の一般化のための統合は半順序集合 $\mathbf{Q}_i$ における半順序 $\subseteq$ に関する極大要素を見い出すことと等価であることが分かる。極大要素とは、最大要素とは異なり、以下で定義される。

定義 5.3 (半順序集合における極大要素) 半順序集合 $\mathbf{Q}_i$ における極大要素とは $q \subseteq q'$ なる $q' \in \mathbf{Q}_i$ が存在しないような要素 $q \in \mathbf{Q}_i$ で定義される。

この極大要素はすべてのペア $q, q' \in \mathbf{Q}_i$ に対して順序関係を調べることにより容易に見い出すことができる。従って、$\mathbf{Q}_i$ の要素数を N_i とすると、極大要素を見い出すのに必要な計算量は $\mathcal{O}(N_i^2)$ となる。

明らかに、$\mathbf{Q}_i$ には複数の極大要素が存在し得る。それ故、これら極大要素の集まり (極大集合族) を定義することができる。

定義 5.4 (極大集合族 (MFS: Maximal Family of Sets)) 半順序集合 $\mathbf{Q}_i$ における極大集合族とは $\mathbf{Q}_i$ における極大要素の集まりとして定義され、$\mathbf{Q}_i^*$ で表される。

言うまでもなく、PSFS の統合結果 $\mathbf{Q}_i^*$ は、一般化された輪郭形状 (一般化図形の凹凸セグメント) を構成しなければならない。換言すれば、$\mathbf{Q}_i^*$ の要素は次の条件を満たさなければならない。

条件 5.1 すべての $q \in \mathbf{Q}_i^*$ に対して、$q \ne \phi$

条件 5.2 $q, q' \in \mathbf{Q}_i^*$ かつ $q \ne q'$ ならば、$q \cap q' = \phi$

これらの条件は、MFS は PSFS と同様、非空で、互いに素な要素 (集合) からなる集合族でなければならないことを表している。もしある MFS が上記条件を満たさなければ、与えられたサンプル図形に共通な凹凸構造がなかったと解釈され、一般化不能となる。以下に条件 5.1,5.2 を満たす MFS の例 (例 5.2) および満たさない例 (例 5.3) を示す。

[例 5.2: 条件 5.1,5.2 を満たす MFS の例] 例 5.1 でさらに S_3 が付加され、それに伴い S_1 に対する新たな PSFS: $\mathcal{P}_{1,3} = \{\{a_{1,1}, a_{1,2}, a_{1,3}, a_{1,4}, a_{1,5}\}, \{a_{1,6}\}, \{a_{1,7}\}, \{a_{1,8}\}\}$ が得られたとすると、例 5.1 の $\mathcal{P}_{1,2}$ とから、$\mathbf{Q}_1 = \mathcal{P}_{1,2} \cup \mathcal{P}_{1,3} = \{\{a_{1,1}\}, \{a_{1,2}, a_{1,3}, a_{1,4}\}, \{a_{1,5}\}, \{a_{1,6}, a_{1,7}, a_{1,8}\}, \{a_{1,1}, a_{1,2}, a_{1,3}, a_{1,4}, a_{1,5}\}, \{a_{1,6}\}, \{a_{1,7}\}, \{a_{1,8}\}\}$ となる。従って、定義により $\mathbf{Q}_i$ の極大集合族は、

$$\mathbf{Q}_1^* = \{\{a_{1,1}, a_{1,2}, a_{1,3}, a_{1,4}, a_{1,5}\}, \{a_{1,6}, a_{1,7}, a_{1,8}\}\} \tag{5.3}$$

となる。$\mathbf{Q}_2, \mathbf{Q}_3$ も同様な手順により得られる。

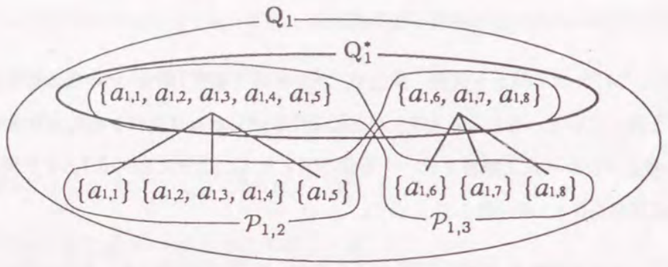
[例 5.3: 条件 5.1,5.2 を満たさない MFS の例] 例 5.2 で S_3 のかわりに別のサンプル $S_{3'}$ が与えられたとする。そして、S_1 に対する PSFS が $\mathcal{P}_{1,3'} = \{\{a_{1,1}, a_{1,2}, a_{1,3}\}, \{a_{1,4}, a_{1,5}, a_{1,6}\}, \{a_{1,7}\}, \{a_{1,8}\}\}$ のように得られたとすると、$\mathbf{Q}_1 = \{\{a_{1,1}\}, \{a_{1,2}, a_{1,3}, a_{1,4}\}, \{a_{1,5}\}, \{a_{1,6}, a_{1,7}, a_{1,8}\}, \{a_{1,1}, a_{1,2}, a_{1,3}\}, \{a_{1,4}, a_{1,5}, a_{1,6}\}, \{a_{1,7}\}, \{a_{1,8}\}\}$ となる。故に、

$$\mathbf{Q}_1^* = \{\{a_{1,1}, a_{1,2}, a_{1,3}\}, \{a_{1,2}, a_{1,3}, a_{1,4}\}, \{a_{1,4}, a_{1,5}, a_{1,6}\}, \{a_{1,6}, a_{1,7}, a_{1,8}\}\} \tag{5.4}$$

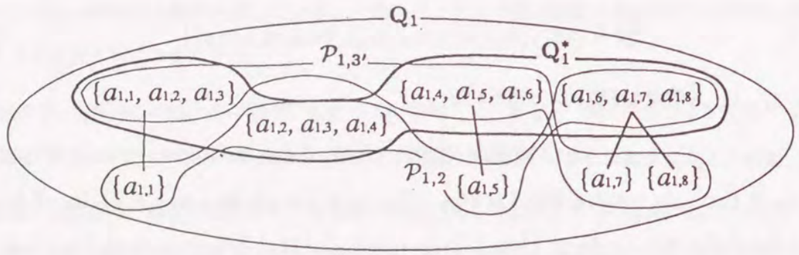
となる。

定義 5.3 より、$\{a_{1,2}a_{1,3}, a_{1,4}\}$ のように、$\mathbf{Q}_1$ のどの要素とも順序関係を持たない要素も極大要素となる。明らかに、例 5.3 における $\mathbf{Q}_1^*$ の要素は互いに素でない。例えば、$a_{1,2}, a_{1,3}$ は最初と 2 番目の要素の両方に含まれている。この場合、これら 3 つのサンプルは共通構造を持たないとして、統合が実行されない。すなわち、$\mathbf{Q}_1^* = \{\{a_{1,1}\}, \{a_{1,2}\}, \ldots, \{a_{1,8}\}\}$ とする。$\mathbf{Q}_2^*, \mathbf{Q}_3^*$ についても同様である。

順序集合を図で表現する手段として Hasse の図式がある。Hasse の図式では、順序関係が線分で表され、かつ $q \subseteq q'$ のときのみ q が q' より高く位置するように描かれる。従って、q と q' の間に線分が結ばれていないときは、q と q' の間には順序関係が存在しないものとみなされ、このときそれらの間の位置 (高さ) 関係は意味がないものと約束する。図 5.7(a),(b) に例 5.2, 5.3 の各々に対する Hasse の図式を示す。



(a) 例 5.2 の Hasse 図式



(b) 例 5.3 の Hasse 図式

図 5.7: 凹凸構造の一般化の半順序を表す Hasse の図式

5.3.3 無矛盾性の検証

上記処理だけでは、サンプル間に共通する凹凸構造の抽出として完全ではない。何故なら、上記統合処理は各サンプル図形毎に独立に行なわれるため、得られた MFS($\mathbf{Q}_i^*, 1 \le i \le N$) 間での対応の一貫性 (無矛盾性) を検証する必要がある。ここでの一貫性とは、厳密には次のように定義される。

定義 5.5 (($\mathbf{Q}_i^*, 1 \le i \le N$ の間での一貫性)) すべての $\mathbf{Q}_i^*$ に対して、以下の *2* つの条件が満たされるとき、かつそのときに限り $\mathbf{Q}_i^*, 1 \le i \le N$ 間での対応が一貫している、あるいは無矛盾であるという。

条件 5.3 いかなる *2* つの *MFS*, $\mathbf{Q}_i^*, \mathbf{Q}_j^* (j \ne i)$ に対しても、必ずその両者の間には *1* 対 *1* 対応 *(*全射*)* が存在する。

条件 5.4 $\mathbf{a}_i \in \mathbf{Q}_i^*, \mathbf{a}_j \in \mathbf{Q}_j^*$ および $\mathbf{a}_k \in \mathbf{Q}_k^*$ に対して、もし $\mathbf{a}_i R \mathbf{a}_j$ かつ $\mathbf{a}_i R \mathbf{a}_k$ ならば、$\mathbf{a}_j R \mathbf{a}_k$

ここに、記号 R は $\mathbf{Q}_i^*$ と $\mathbf{Q}_j^*$ における 2 つの要素間の対応関係を表すものとする。この対応ペアは以下に示す手順により得られる。

(対応ペアを得るためのアルゴリズム)

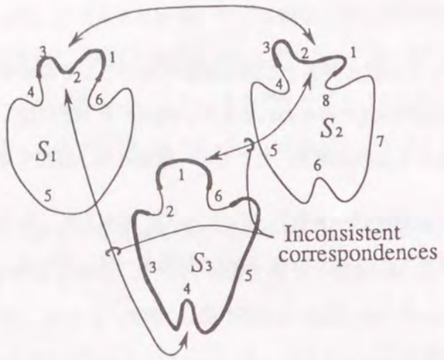
ある $\mathbf{a}_i \in \mathbf{Q}_i^*$ が与えられたとき、この $\mathbf{a}_i$ に対応する $\mathbf{a}_j \in \mathbf{Q}_j^*$ は以下のステップを実行することにより得られる。

(Step 1) $\mathbf{A}_i$ と $\mathbf{A}_j$ の間の対応結果 $C_{i,j}$ において、$\mathbf{a}_i$ を構成する一つあるいはそれ以上の連続するセグメントを見つける。($\mathbf{a}_i$ は $\mathbf{A}_i$ の一つあるいはそれ以上の連続するセグメントからなる集合であったことに注意。)

(Step 2) $C_{i,j}$ を用いて、得られた $\mathbf{A}_i$ 中のセグメントに対応する $\mathbf{A}_j$ 中のセグメントを抽出する。

step 2 で抽出されたセグメント (群) から成る集合が、所望の $\mathbf{a}_j$ である。この対応ペア探索アルゴリズムを用いることにより、条件 5.3, 5.4 を容易にチェックすることができる。

統合結果 ($\mathbf{Q}_i^*, 1 \le i \le N$) が共通構造を抽出しているか否かを判定するためには、得られた MFS の間の一貫性の検証が必要となる。換言すれば、与えられたサンプル図形に共通する凹凸構造というのは、サンプル間におけるセグメント対応が一貫して始めて抽出可能となる。もし、矛盾したセグメント対応を許せば、凹凸構造の一般化は単にサンプル図形における最小数の凹凸の抽出を意味することになり、これは明らかに共通の凹凸構造とは言えない。すなわち、共通の凹凸構造というのは、凹凸セグメントの対応関係を考慮して始めて意味をなすのである。図 5.8 に 3 つのサンプル図形間の対



$\{a_{1,1}, a_{1,2}, a_{1,3}\} R \{a_{2,1}, a_{2,2}, a_{2,3}\}$
$\{a_{1,1}, a_{1,2}, a_{1,3}\} R \{a_{3,3}, a_{3,4}, a_{3,5}\}$
$\{a_{2,1}, a_{2,2}, a_{2,3}\} R \{a_{3,1}\}$

図 5.8: 矛盾したセグメント対応の例

応に一貫性がない場合を示す。矛盾する矛盾がによる統合不可能な場合の例を示す。条件 5.3,5.4 が満たされない場合は、与えられたサンプル図形間に共通する凹凸構造がなかったと見なされ一般化不能となる。

結局、対応結果の統合 (PSFS の一般化) は条件 5.1 ～ 5.4 が満たされたときのみ実現される。一般に、同じクラスに属すサンプル図形は、比較的類似した凹凸構造を持っているので、上記 4 つの条件は、通常は満たされるものと考えられる。例 5.4 は図 5.8 に示した 3 つのサンプル図形に対して、望ましい対応結果からの統合処理結果を示す。

[例 5.4: 無矛盾な統合結果の例] 図 5.8 の 3 つのサンプル図形に対し、以下のような対応結果 (PSFS) が得られたとする:

$\mathcal{P}_{1,2} = \{\{a_{1,1}\}, \{a_{1,2}\}, \{a_{1,3}\}, \{a_{1,4}\}, \{a_{1,5}\}, \{a_{1,6}\}\}$,

$\mathcal{P}_{2,1} = \{\{a_{2,1}\}, \{a_{2,2}\}, \{a_{2,3}\}, \{a_{2,4}\}, \{a_{2,5}, a_{2,6}, a_{2,7}\}, \{a_{2,8}\}\}$

$\mathcal{P}_{1,3} = \{\{a_{1,1}, a_{1,2}, a_{1,3}\}, \{a_{1,4}\}, \{a_{1,5}\}, \{a_{1,6}\}\}$,

$\mathcal{P}_{3,1} = \{\{a_{3,1}\}, \{a_{3,2}\}, \{a_{3,3}, a_{3,4}, a_{3,5}\}, \{a_{3,6}\}\}$,

$\mathcal{P}_{2,3} = \{\{a_{2,1}, a_{2,2}, a_{2,3}\}, \{a_{2,4}\}, \{a_{2,5}\}, \{a_{2,6}\}, \{a_{2,7}\}, \{a_{2,8}\}\}$,

$\mathcal{P}_{3,2} = \{\{a_{3,1}\}, \{a_{3,2}\}, \{a_{3,3}\}, \{a_{3,4}\}, \{a_{3,5}\}, \{a_{3,6}\}\}$.

このとき、統合結果は以下のようになる:

$\mathbf{Q}_1^* = \{\{a_{1,1}, a_{1,2}, a_{1,3}\}, \{a_{1,4}\}, \{a_{1,5}\}, \{a_{1,6}\}\}$,

$\mathbf{Q}_2^* = \{\{a_{2,1}, a_{2,2}, a_{2,3}\}, \{a_{2,4}\}, \{a_{2,5}, a_{2,6}, a_{2,7}\}, \{a_{2,8}\}\}$,

$\mathbf{Q}_3^* = \{\{a_{3,1}\}, \{a_{3,2}\}, \{a_{3,3}, a_{3,4}, a_{3,5}\}, \{a_{3,6}\}\}$.

すなわち、各図形に対して、4 つの最適スケールの凹凸セグメントが 3 つの図形に共通する凹凸構造として抽出されていることが分かる。これらの統合結果は上述の定義のもとで無矛盾であることが確認される。例えば、$\{a_{1,1}, a_{1,2}, a_{1,3}\} R \{a_{2,1}, a_{2,2}, a_{2,3}\}$, $\{a_{2,1}, a_{2,2}, a_{2,3}\} R \{a_{3,1}\}$ かつ $\{a_{1,1}, a_{1,2}, a_{1,3}\} R \{a_{3,1}\}$ が同時に成立する。

5.4 一般化図形の作成

5.4.1 線形補間

前節の統合処理により各サンプル図形に対して他の $N-1$ 個のサンプル図形すべてに共通する最適スケールの凹凸セグメント群が抽出される。これらが一般化図形の凹凸セグメントを構成する。ところで、一般にガウス関数による輪郭の平滑化は輪郭全体を収縮する傾向にあるため、抽出した最適スケールセグメントをそれらの (x, y)– 座標に従って配置すると、図 5.9 に示すように、セグメント端点間に位置ずれ (間隙) が生じる。すなわち抽出された最適スケールセグメントの隣接セグメント同士を単純に接続すると不自然な凹凸が生じる。この収縮は Lowe[89] の収縮補正法を用いてかなり改善することができるが、しかしながら、上記問題の完全な解決にはならない。

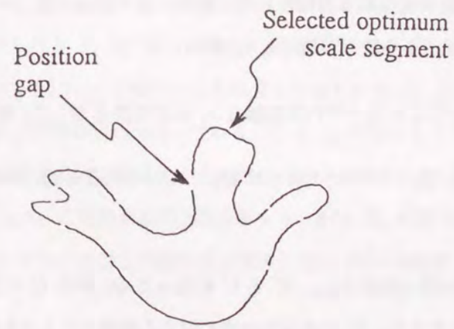


図 5.9: 統合結果に基づく最適スケールセグメントの重畳表示

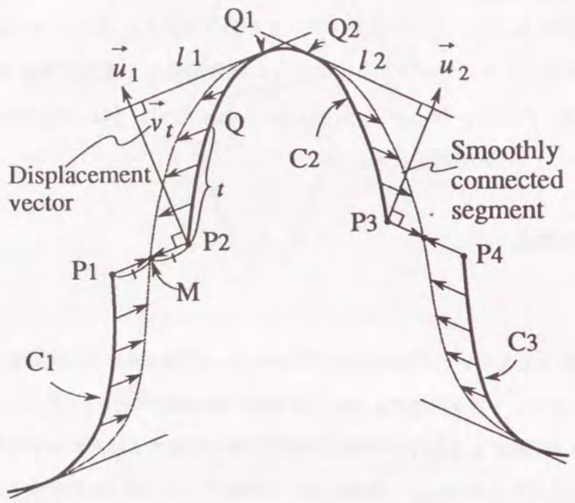


図 5.10: 線形補間法による最適スケールセグメントの接続

そこで以下で述べる線形補間に基づく方法によりこの問題に対処する。図 5.10 はこの線形補間法を説明するための図である。図 5.10 において、C_1, C_2, C_3 は接続すべき 3 つの最適スケールセグメント曲線で、点線で示す曲線がそれらの接続結果 (補正曲線) である。これは、同図に示すように、C_1, C_2, C_3 上の点の適切な量の変位により得られる。簡単のため曲線 C_2 上の変位ベクトルのみを考 える。このとき、変位ベクトルを次の手順により求める。

(Step 1) ベクトル $\overrightarrow{P_1P_2}, \overrightarrow{P_3P_4}$ に各々平行な接線 l_1, l_2 による接点 Q_1, Q_2 を求める。

(Step 2) もし Q_2 が P_2 と Q_1 の間にあるならば Q_1 と Q_2 の両者を Q_1 から Q_2 に至る曲線の中点とする。

(Step 3) C を Q_1 と P_2 の間の曲線とし、$\overrightarrow{v_t}$ を C に沿って Q_1 から C に沿って距離 t だけ離れた点の変位ベクトルとすると、C 上の全ての点に対する変位ベクトルを次式で計算する: ここで t_0 は曲線 C の長さで、M は線分 P_1P_2 の中点である。Q_2 と P_3 間の曲線上の点についても同様に変位ベクトルを計算する。

$$\overrightarrow{v_t} = \frac{t}{t_0}\overrightarrow{P_2M} \tag{5.5}$$

実際、上記 step.1 において、$Q_1(Q_2)$ は、$\overrightarrow{P_1P_2}(\overrightarrow{P_3P_4})$ に垂直なベクトル $\overrightarrow{u_1}(\overrightarrow{u_2})$ と $\overrightarrow{P_2Q}(\overrightarrow{P_3Q})$ との内積として次式で求めることができる:

$$\begin{cases} Q_1 = arg\max_Q\{\overrightarrow{u_1}\cdot\overrightarrow{P_2Q}\} \\ Q_2 = arg\max_Q\{\overrightarrow{u_2}\cdot\overrightarrow{P_3Q}\} \end{cases} \tag{5.6}$$

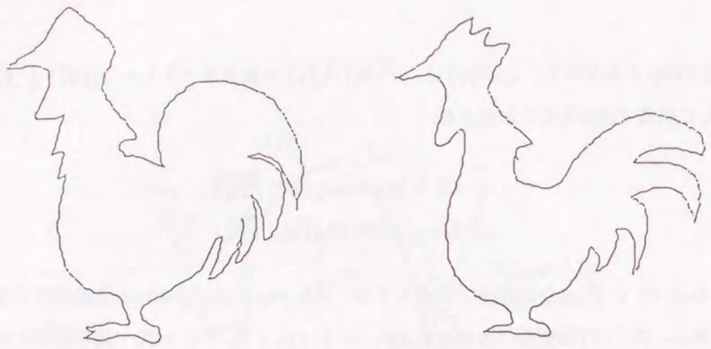
但し、Q は点 P_2 と P_3 の間の曲線上の点とする。また $arg\max_Q\{\}$ は $\{\}$ を最大にするパラメータ Q の値である。式 (5.5) は変位の大きさが $(Q_1, 0)$ と $(P_2, |\overrightarrow{P_2M}\cdot|)$ の間で線形補間されていることを意味する。上記線形補間法は、明らかに新たな変曲点を発生させずに曲線を滑らかに接続できる。

5.4.2 一般化図形作成例

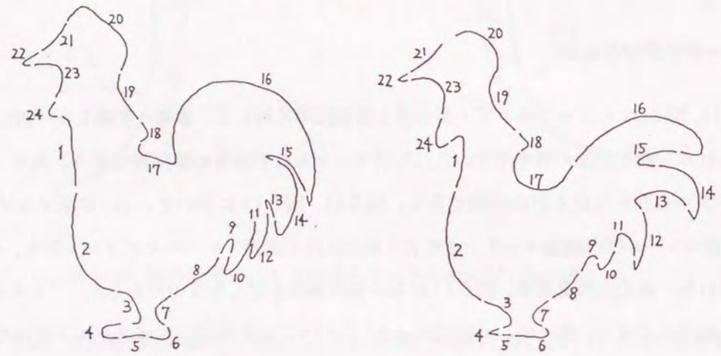
図 5.11, 図 5.12 に各々 2 サンプルから成る輪郭図形に対して、本章で提案した一般化手法を適用して得られた一般化図形の例を示す。これら原サンプルは手書きの線図形であり、高々 1000 点程度のディジタル点列から成る自由曲線である。図 5.11, 図 5.12 において、(a) は原サンプル図形を、(b) は多重スケール凹凸構造マッチングにより抽出された最適スケールセグメント群を、そして、(c) は生成された一般化図形を表す。図 5.11 は鶏の輪郭形状を表したものであるが、"トサカ" は一方にあるが、他方にはない。従って、一般化により"トサカ" を表す凹凸がデフォルメされていることがわかる。また、図 5.12 ではかなりノイズがある輪郭図形であるが、クラスに共通する意味のある凹凸構造が抽出されていることがわかる。

図 5.13 は 3 サンプルの場合に対する一般化図形生成過程を示したものである。図 5.13(b) は、サンプル S_1 と S_2 から得られた S_1 の一般化図形で、図 5.13(c) は、S_1 と S_3 から得られた S_1 の一般化図形である。そして、図 5.13(c) はこれらを統合して得られた S_1 に対する最終的な一般化図形である。図 5.13(b) では、右側の凹凸が残っている (S_1 と S_2 にはともにその凹凸を持っているため) が、統合により S_3 の影響を受けて最終的にその凹凸はデフォルメされている。そして、この一般化図形は明らかに、S_1, S_2, S_3 に共通する凹凸構造をもち、かつ S_1 の原形状を保存した形状となっている。尚、これらの図形は全て手書きの線図形であり、高々 1000 点程度のディジタル点列からなる自由曲線である。

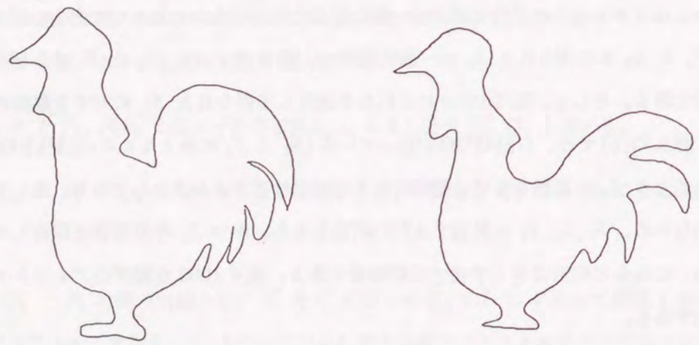
以上の例から明らかなように、得られた一般化図形は各クラスに共通する凹凸構造と、各々の原形状を反映した凹凸形状をもっていることがわかる。すなわち、単に細かな凹凸を除去するのではなく、細かな凹凸でも同一クラスに属す図形間に共通するものであればそれらを残し、逆に、比較的大きな凹凸でもクラス内で共通するものでなければそれらを除去 (平滑化) している。



(a) 学習サンプル

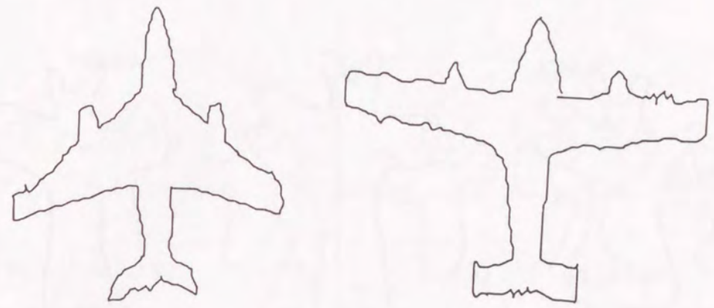


(b) 共通凹凸構造抽出結果

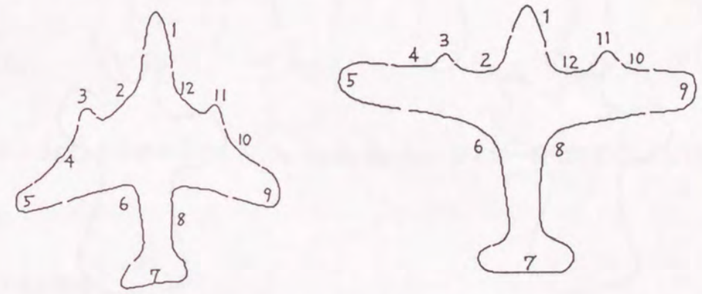


(c) 一般化図形生成結果

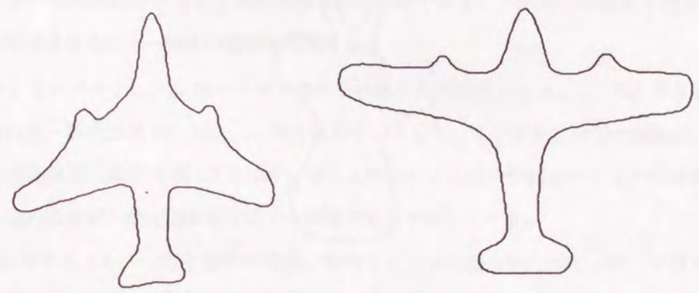
図 5.11: 一般化図形生成例



(a) 学習サンプル



(b) 共通凹凸構造抽出結果



(c) 一般化図形生成結果

図 5.12: 一般化図形生成例

(a) 学習サンプル、左から S_1, S_2, S_3

(b)S_1 と S_2 からの一般化結果

(c)S_1 と S_3 からの一般化結果

(d)S_1, S_2, S_3 からの一般化図形生成結果

図 5.13: 3 サンプルの場合の一般化図形生成過程

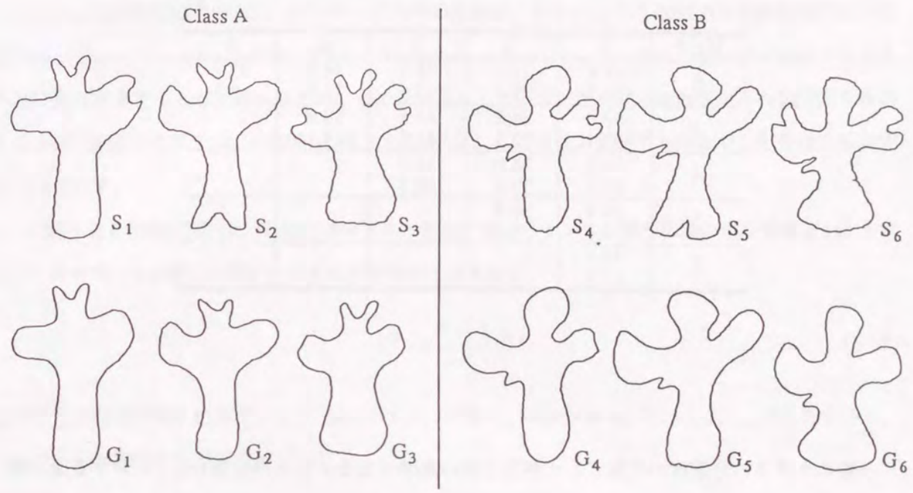


図 5.14: クラス A(B) の学習サンプル $S_1, S_2, S_3(S_3, S_4, S_5)$、および一般化図形 $G_1, G_2, G_3(G_4, G_5, G_6)$

5.5 評価

5.5.1 クラス分離度

本節ではこれまで述べてきた一般化手法の定量的評価を行なう。提案した一般化手法では、5.2 節で述べたようにまず多重スケール凹凸構造マッチングを行なっている。このマッチングでは、2 図形の最適スケールの凹凸セグメント対応が得られるだけでなく、2 図形の相違度も算出される[†]。従って、この相違度を用いて一般化の効果を考察する。

図 5.14 に、2 クラス (A, B)、各々 3 サンプルから成る輪郭図形 $(S_1, S_2, \ldots, S_6)$ および本手法を適用して得られた一般化図形 $(G_1, G_2, \ldots, G_6)$ を示す。そして、サンプル図形間の相違度、および一般化図形間の相違度算出結果を表 5.1 に示す。表 5.1 において上段の数値がサンプル図形相互間での相違度を、下段の数値が一般化図形相互間での相違度を各々示している。

表 5.1 の結果を見ると、一般化図形の場合、学習サンプルの場合に比べて、同一クラス間での相違度がクラス相互間での相違度に対して相対的に値が小さくなっていることが分かる。これは、一般化図形を見れば直観的に妥当であると言える。すなわち、一般化により同一クラスの輪郭形状がより類似したものになったからである。

[†]第 4 章の式 (4.36) 参照。

表 5.1: サンプル図形間 (上段) および一般化図形間 (下段) の相違度

(×10^{-3})

Fig.#	1	2	3	4	5
6	41.1 26.3	21.7 22.5	38.1 47.6	24.5 12.9	35.2 9.3
5	53.8 33.4	24.2 22.7	45.8 48.3	38.9 17.1	
4	31.3 21.1	34.0 32.2	35.6 40.2		
3	16.4 4.8	38.8 9.4			
2	24.4 11.6				

一般にクラスの分離度の尺度として判別分析の基準であるクラス内分散 (σ_W^2) に対する全分散 (σ_T^2) の比:

$$\kappa = \sigma_T^2/\sigma_W^2 \tag{5.7}$$

が用いられる [90]。ここで、クラス内分散とは各クラス毎に分散を求めた後それらを総和したものであり、次式で定義される。

$$\sigma_W^2 = \frac{1}{N}\sum_{i=1}^{C}\sum_{j=1}^{n_i}(x_{i,j}-\bar{x}_i)^2 \tag{5.8}$$

ただし、各変数の意味は以下のとうりである。

C :. クラスの総数

N : 全サンプル数

n_i : 第 i クラスのサンプル数

$x_{i,j}$: 第 i クラスの第 j 番目の要素 (サンプル)

$\bar{x}_i$: 第 i クラスの要素の平均値:

$$\bar{x}_i = \frac{1}{n_i}\sum_{j=1}^{n_i}x_{i,j} \tag{5.9}$$

また、全分散はサンプル全体にわたる分散であり、次式で与えられる。

$$\sigma_T^2 = \frac{1}{N}\sum_{i=1}^{C}\sum_{j=1}^{n_i}(x_{i,j}-\bar{x})^2 \tag{5.10}$$

ここで、$\bar{x}$ はサンプル全体の平均値:

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{C}\sum_{j=1}^{n_i}x_{i,j} \tag{5.11}$$

である。

一方、上記実験結果 (表 5.1) は全サンプル間の相違度、すなわちある仮想的な特徴空間内での距離関係が分かっているだけで特徴空間そのものはわかっていない。それゆえ、表 5.2 の結果から直接上記分散を計算するわけにはいかない。この解決策として、サンプル間の距離値から特徴空間を推測する方法 (多次元尺度法 [91]) の利用が考えられる[†]が、ここではより簡明な方法で上記分散を算出する方法を示す。

一般にある特徴空間内に N 個のサンプルが存在するときサンプル間の距離の自乗平均値 ($\bar{d}^2$ とする) とサンプルの分散との間には次式に示す関係が成り立つ。

$$\sigma^2 = \frac{N-1}{2N}\bar{d}^2 \tag{5.12}$$

[証明] ある特徴空間中に存在する N 個のサンプルを各々、ベクトル $\boldsymbol{x}_i, (i=1,2,\ldots,N)$ とすると、サンプル間の距離の自乗平均値 $\bar{d}^2$ は、

$$\bar{d}^2 = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1,j\neq i}^{N}|\boldsymbol{x}_i-\boldsymbol{x}_j|^2/\binom{N}{2}$$

と書ける。よって、

$$\begin{aligned}
\bar{d}^2 &= \sum_{i=1}^{N}\sum_{j=1,j\neq i}^{N}(|\boldsymbol{x}_i|^2+|\boldsymbol{x}_j|^2-2\boldsymbol{x}_i\cdot\boldsymbol{x}_j)/N(N-1)\\
&= \left(2(N-1)\sum_{i=1}^{N}|\boldsymbol{x}_i|^2-2\sum_{i=1}^{N}\sum_{j=1,j\neq i}^{N}\boldsymbol{x}_i\cdot\boldsymbol{x}_j\right)/N(N-1)\\
&= \left(2N\sum_{i=1}^{N}|\boldsymbol{x}_i|^2-2\left(\sum_{i=1}^{N}|\boldsymbol{x}_i|\right)^2\right)/N(N-1)\\
&= \left(\frac{1}{N}\sum_{i=1}^{N}|\boldsymbol{x}_i|^2-\left(\frac{1}{N}\sum_{i=1}^{N}|\boldsymbol{x}_i|\right)^2\right)/\frac{N-1}{2N}\\
&= \sigma^2/\frac{N-1}{2N}
\end{aligned}$$

故に、

$$\sigma^2 = \frac{N-1}{2N}\bar{d}^2 \quad \text{[証明終り]}$$

従って、式 (5.12) を用いることにより、クラス内分散、全分散を計算することが可能になる。故に、式 (5.7) より分離度 κ が得られる。

[†]特徴空間が得られれば、その空間内でのサンプルの座標値から全分散・クラス内分散が計算できる。

5.5.2 評価結果

表 5.1 の結果を基にクラス分離度 κ を各々算出したところ、原サンプル間同士 (学習前) では $\kappa = 0.8$ であったのに対し、一般化図形間同士 (学習後) では $\kappa = 3.6$ となった。これは、提案手法により、各クラス毎に十分な一般化がなされていることを示している。この結果から、提案した一般化の効果を定量的にも検証することができた。

5.6 結言

本章では、同一クラスに属す複数サンプルからそれらに共通する最適スケールの凹凸構造を抽出するという新たな一般化手法を提案した。本手法はパターン形状を直接一般化するという点で従来手法と本質的に異なっており、以下に示すような従来にない優れた特徴を持つ。

(1) 対象の構造に関する事前知識が不要であるため、任意の 2 次元輪郭図形に適用できる。

(2) パターン形状を一般化した一般化図形を生成できる。

(3) 少数サンプルから学習が可能。

本手法は図形輪郭の多重スケールの凹凸構造といった低次の特徴のみを用いているため汎用的であり、通常の輪郭図形に対しては妥当であると考えられる。本章で提案した一般化手法を実際の輪郭形状に適用した結果、人間の直観にあった一般化結果 (一般化図形) が得られた。さらに、一般化の効果を判別分析の基準であるクラス分離度を導入して定量的に評価した。その結果、一般化によりクラス分離度が増大し、提案手法の有効性を検証することができた。

第 6 章

Visual モデルの生成とその汎用形状認識系への応用

6.1 序言

第 5 章では、同一クラスに属す複数の輪郭形状サンプルが与えられたとき、その各サンプルに対し、全てのサンプルに共通する凹凸構造を有し、かつその原形状をできるだけ保存した一般化図形を生成する手法について述べた。そこで、本章では、これらの一般化図形群からクラスの visual モデルを自動生成するためのアルゴリズムについて述べる。ここでいう、visual モデルとは、クラスの全サンプルに共通する凹凸構造を持つだけでなく、全てのサンプルの平均的な凹凸形状をもつ形状モデルを意味する。すなわち、クラスの全サンプル図形を代表するプロトタイプである。

visual モデルには、部分輪郭の重要性を示す"重要度"が付与されている。この重要度は一般化図形生成時に選択された最適スケールの関数として定義される。上述したように visual モデルはあくまでクラスを代表するプロトタイプ図形であるため、それだけでは、そのクラス内でサンプル図形間にどれくらい類似性があるのかが不明である。ところが、重要度を visual モデルの各部分輪郭に付与することによって、どの部分輪郭がクラスのサンプル図形間で類似していたのか (重要であるか) の尺度を提供することができる。従って、形状の類似性を部分輪郭毎に評価可能となる。

本章ではさらにこの visual モデルを用いた汎用形状認識系について述べる。そして、その具体例として車の profile からの車種認識への適用を試みる。最後に、実験結果を基に visual モデルの形状モデルとしての妥当性・有効性を評価する。

6.2 Visual モデルの生成

6.2.1 基本的な考え方

第5章で述べた一般化図形は、クラスの全サンプル図形に共通する凹凸構造を持つが、凹凸形状に関してはあくまで原サンプルを保存した形状であった。すなわち、原サンプルの各々に対して一つの一般化図形が生成されていた。これらの一般化図形をクラスの形状モデルとすることも考えられるが、モデルとしては冗長である。例えば、図6.1(a)のような、同一クラスに属す3つのサンプルが与えられとしよう。このとき、我々人間はこのクラスのプロトタイプとして図6.1(b)のようなパターン形状を想像する。すなわち、クラスの形状モデルとしては、図6.1(b)のように凹凸構造だけでなく凹凸形状に関してもクラスの全サンプルを代表するようなパターン形状が形状モデルとして適切であると考えられる。そこで、このプロトタイプ図形を visual モデルと呼び、クラスの形状モデルと定義することとする。

(a) 同一クラスの形状サンプル

(b) クラスのプロトタイプ図形
図 6.1: プロトタイプ図形の例

6.2.2 Visual モデル生成アルゴリズム

第5章で述べたように、同一クラスの各サンプル図形から生成された一般化図形は同じ凹凸構造を持つだけでなく、その凹凸セグメント対応も分かっている。従って、要は、その凹凸セグメント上の点列群に対して、全ての一般化図形間にわたる点対応が得られればその対応点の内挿処理により全てのサンプル図形の平均的な形状が求まり、所望の visual モデルが生成できることになる。以下に述べるアルゴリズムは基本的にはこの考え方に基づいている。

visual モデル生成過程では一般化図形を多角形として取り扱う。但し、その多角形の辺の数は十分多いものとし一般化図形の原形状を逸脱しないものとする。すなわち、各一般化図形は辺の数の十分大きな多角形で近似されているものとする。このとき、一般化図形 G_i $(i = 1, 2, \ldots, N)$ は短線分の連結として定義され、その線分の端点の順序リストとして次のように表される。

$$G_i = \{(x_{i,t}, y_{i,t}) | \quad t = 1, 2, \ldots, T_i\}, \qquad i = 1, 2, \ldots, N \tag{6.1}$$

ここに N は一般化図形の総数である。但し、$t = 1$ は一般化図形相互間で対応する第1番目のセグメントを構成する第1番目の点を指すことと約束する。式(6.1)は第 i 番目の一般化図形が T_i 多角形で近似されていることを表している。$(x_{i,t}, y_{i,t})$ は第 i 番目の一般化図形の第 t 番目の点の (x, y)-座標を意味する。さらに、第 i 番目の一般化図形の第 k セグメントを構成する点列数を $u_{i,k}$ で表すこととする(図6.2)。そして、一般化図形における凹凸セグメントの総数を K とすると、明らかに、$T_i = \sum_{k=1}^{K} u_{i,k}$ が成り立つ†。

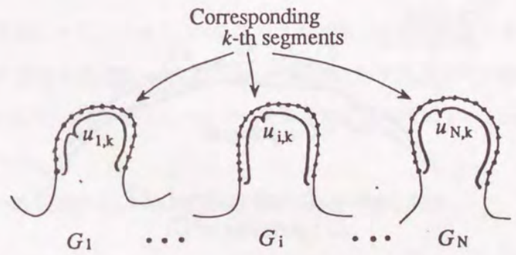


図 6.2: $u_{i,k}$ の説明図

†一般化図形は互いに共通の凹凸構造をもつので、K は全ての一般化図形で同じ値である。

もし N 個の一般化図形間で、これらの点群が 1 対 1 に対応していれば、対応点毎にそれらの座標値を単純に平均する[†] ことにより所望の visual モデルを得ることができる。ところが、残念なことに一般化図形相互間では凹凸の数は同一であるが、その凹凸セグメントを構成する点の数は互いに異なっているのが普通で、従ってそれらの対応も得られていない。そこで、以下に、対応する凹凸セグメント相互間で点対応を得るアルゴリズムを示す。

[一般化図形相互間での点群の対応づけアルゴリズム]

(Step 1) まず、N 個の一般化図形にわたって一般化図形を構成する点列数の平均値 $\bar{T}$ を求める。

$$\bar{T} = \frac{1}{N}\sum_{i=1}^{N} T_i \tag{6.2}$$

(Step 2) 一般化図形 G_i に対し、その T_i- 多角形の辺および頂点上に $\bar{T}$ 個の点を等間隔

$$\delta = \frac{L_i}{\bar{T}} \tag{6.3}$$

になるように再配置する[††]。ここで、L_i は多角形近似された一般化図形 G_i の周囲長である。但し、δ は図 6.3 に示すように G_i の多角形上で測った長さであり、2 点間を結ぶ直線距離でないことに注意する。この処理を全ての一般化図形について行なう。これにより、全ての一般化図形は、$\bar{T}$- 多角形で近似されたことになり、これはすなわち、全ての一般化図形が同じ点列数を有するように正規化されたことを意味する。

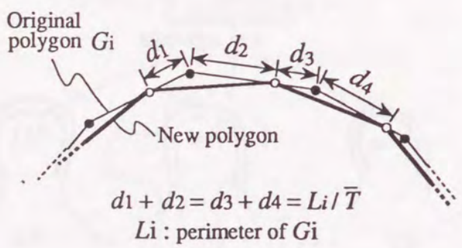


図 6.3: 点列の再配置

[†]図形の内挿により平均的な形状を得ようとするものであり、各点の座標値そのものは全く意味がないことに注意。

[††]δ の値は G_i 毎に異なることに注意。

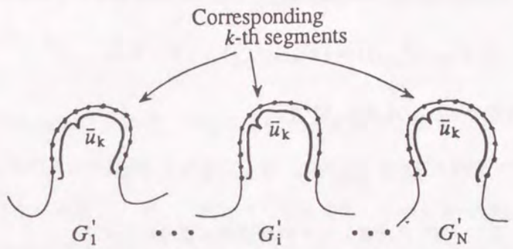


図 6.4: $\bar{u}_k$ の説明図

(Step 3) 一般化図形相互間で対応する第 k 番目のセグメントに対してそのセグメントを構成する点列の平均値 $\bar{u}_k$ を次式により求める。

$$\bar{u}_k = \frac{1}{N}\sum_{i=1}^{N} u_{i,k}, \qquad \text{for} \quad k = 1, 2, \ldots, K \tag{6.4}$$

(Step 4) step 2 で行なったのと同様な方法で、一般化図形の第 k セグメント上で $\bar{u}_k$ 個の点を再配置する。これは、step 1, step 2 で行なった輪郭全体の点列数による正規化を、一般化図形相互間で対応する凹凸セグメントに対して行なったことになる。この処理を全ての一般化図形の各凹凸セグメントに対して実行することにより、最終的に対応する凹凸セグメント内での点対応が得られることになる。

尚、上記ステップにおいて、step 1, step 2 は冗長であると思われるかも知れないが、一般化図形間で点列数が極端に異なる場合に step.3, step.4 を直接実行すると元の一般化図形の形状が変化する (荒い多角形近似となる) ことがある。そこで、それを避けるために予め step.1, step.2 により点列の正規化を行なっている。

上記処理により一般化図形 G_i から点列が再配置された一般化図形 G'_i

$$G'_i = \{(x_{i,t}, y_{i,t}) | \quad t = 1, 2, \ldots, T'\} \qquad i = 1, 2, \ldots, N \tag{6.5}$$

が得られる。但し、T' は G'_i の点列数で

$$T' = \sum_{k=1}^{K} \bar{u}_k \tag{6.6}$$

である。このとき、N 個の一般化図形 G'_i $(i = 1,2,\ldots,N)$ の任意の2つの一般化図形 G'_m, G'_n $(1 \leq m,n \leq N, m \neq n)$ の点群の間には、全単射 $f : G'_m \rightarrow G'_n$ が存在することになり、すなわち

$$f((x'_{m,t}, y'_{m,t})) = (x'_{n,t}, y'_{n,t}) \qquad t = 1,2,\ldots,T' \tag{6.7}$$

が成り立つ。従って、所望の visual モデル M を、

$$M = \{(x_t, y_t)| \quad t = 1,2,\ldots,T'\} \tag{6.8}$$

で表すと、$x_t, y_t, t = 1,2,\ldots,T'$ は次式に示す内挿計算により得られる。

$$x_t = \Big(\sum_{i=1}^{N} w_i x_{i,t}\Big) \Big/ \sum_{i=1}^{N} w_i \qquad \text{および} \qquad y_t = \Big(\sum_{i=1}^{N} w_i y_{i,t}\Big) \Big/ \sum_{i=1}^{N} w_i \tag{6.9}$$

ここで、w_i は第 i 番目の一般化図形の寄与度を表す重みであり、通常の visual モデルでは $w_i = w$(定数) で良いが、w_i を色々と変えることによりクラス内のインスタンスを自動生成することも可能となる。

6.3 重要度

6.3.1 重要度の意味

前節で述べた visual モデルは、ある特徴空間中のサンプル群の重心に相当するパターンを生成したことに対応していると考えられる。従って、この visual モデルだけでクラスのモデルとするのでは、クラスの元のサンプルがどのように分布していたのか、言い替えれば、どういう部分輪郭が支配的であったかという情報が失われてしまう。そこで、この問題に対処するために、visual モデルの部分輪郭に"重要度"なる一種の重みを付与して、部分輪郭毎の重要性を評価できるようにする。

今、簡単のためにあるクラスが2つのサンプル図形から成るとする。そして、第5章で述べた方法によりこの2つのサンプル図形に対して各々一般化図形を生成したとする。このとき、一般化図形の各凹凸セグメントは各原サンプルの最適スケールセグメントから成っている。そして、その2つの一般化図形間で対応する凹凸セグメントはそれぞれ固有の最適スケール値を持つ。今、その一般化図形間で対応するセグメントの最適スケール値のうちの最大値を考える。すると、第4章で述べたように、与えられた2つのサンプル図形の対応する部分輪郭が類似している程、各々スケールの小さな値でマッチングがとれるため、先の最大値が小さくなる。

一方、2つのサンプル図形の対応する部分輪郭が類似している程、その部分輪郭がクラスを特徴づける凹凸構造としてより重要であることを意味する。従って、一般化図形間で対応する最適スケールセグメントのそれらのスケール値の最大値を基に部分輪郭の重要度を定義することができる。

以上の議論は、2サンプルの場合であったが、N サンプルの場合でも同様な議論ができる。次の6.3.2では、一般の N サンプルの場合について、上記重要度の厳密な定義式を与える。

6.3.2 定義式

今、得られた一般化図形相互間で、対応する凹凸セグメントのインデックスを6.2.2と同様に $k = 1,\ldots,K$(K はセグメント総数) で表わし、第 i 番目のサンプル図形に対する一般化図形の第 k セグメントの最適スケール値を、$\sigma^*_{i,k}$ で表わすこととする。このとき一般化図形相互間で対応する第 k セグメントに付与すべき重要度 $(\omega(k))$ を 6.3.1 で述べた考え方に基づき次式で定義する。

$$\omega(k) = \frac{(\log \hat{\sigma}_k)^{-1}}{\sum_{k=1}^{N} (\log \hat{\sigma}_k)^{-1}} \tag{6.10}$$

ここに、$\hat{\sigma}_k$ は、一般化図形相互間で対応する凹凸セグメントの最適スケールの最大値で与えられる:

$$\hat{\sigma}_k = \max_i \{\sigma^*_{i,k}\} \tag{6.11}$$

すなわち、この最大値が小さい程、原形状に近い平滑化図形でマッチングがとれたことになり、従ってその部分セグメントが重要となる。

重要度は輪郭全体で 1.0 となるように正規化している。ここで、スケールの対数をとっているのは、スケールスペース上で一様なサンプリングにするための正規化であり、その妥当性は文献 [92] に示されている。直感的には、ガウス関数が指数関数であるのでそれを対数を用いて相殺していると解釈できる。またスケールの対数の逆数をとっているのは 6.3.1 で述べたようにスケール値が小さい程、重要度が大きくなるようにするためである。明らかに、一般化図形相互間で、対応するセグメントには同一の重要度が付与されることになる。

以上のことから本章で提案する visual モデルを再定義すると、visual モデルはクラスの全てのサンプル図形に共通する凹凸構造とクラスの全てのサンプル図形の平均的な凹凸形状を持ち、かつクラスにおける各凹凸構造の相対的な重要性を示す重要度から成る。

6.4 汎用形状認識への応用

以上述べた方法をさらに拡張することにより、各クラスのサンプル図形群 (学習サンプル) を与えるだけで、未知サンプルがどのクラスに属すかを認識する、いわゆる汎用形状認識系を構築できる。以下にその方法を述べる。

これは、次の 3 つの処理から成る。

[汎用形状認識アルゴリズム]

(Step 1) 各クラスに対して、与えられた学習サンプルから visual モデルを作成する。

(Step 2) 未知サンプルが与えられたとき、未知サンプルと各クラスの visual モデルとの相違度を計算する。この相違度計算は、第 4 章で述べたマッチングにおいて、式 (4.33) の d の代わりに次式で定義する重み付け相違度 $\tilde{d}$ を用いること以外はすべて同じである。

$$\tilde{\psi}\Big(\mathbf{a}(i-2n|i), \mathbf{b}(j-2m|j)\Big) = \Big(\sum_{k=i-2n}^{i} \omega(k)\Big) \cdot \psi\Big(\mathbf{a}(i-2n|i), \mathbf{b}(j-2m|j)\Big) \tag{6.12}$$

上式で添え字 i は、visual モデル、添え字 j は未知サンプルに対応するものとする。すなわち、上式は visual モデルの各凹凸セグメントに付与された重要度で重み付けされた相違度を計算している。

(Step 3) 相違度 $\tilde{\psi}$ が最小となる visual モデルが属すクラスを未知サンプルのクラスとする。

式 (6.12) は、visual モデルと未知サンプルとの相違度計算において、visual モデルの最下位スケールセグメント $(a_{i-2n}, a_{i-2n+1}, \ldots, a_i)$ が未知サンプルの最下位スケールセグメント $(b_{j-2m}, b_{j-2m+1}, \ldots, b_j)$ と対応するときは、visual モデルのその各セグメントに付与された重要度の総和で重み付けされることを表している。尚、visual モデルの 1 つのセグメントが未知サンプルのセグメントと対応する $(n=0)$ ときは、明らかに、そのセグメントに付与された重要度のみが重み付けられる。

以上からわかるように、本モデル獲得手法は、得られたモデルを用いた図形認識に対しても同一の枠組みで取り扱うことが可能である。

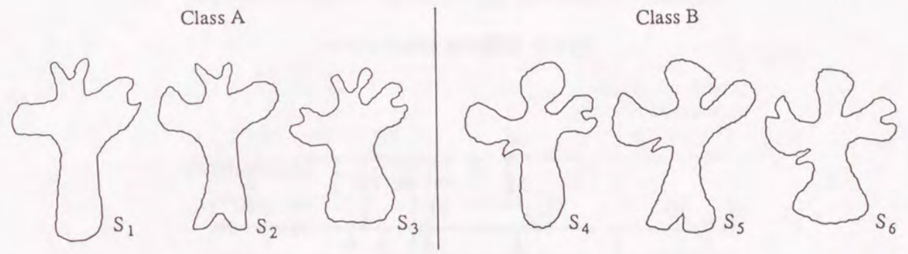
6.5 認識実験とその評価

6.5.1 未知サンプルの認識

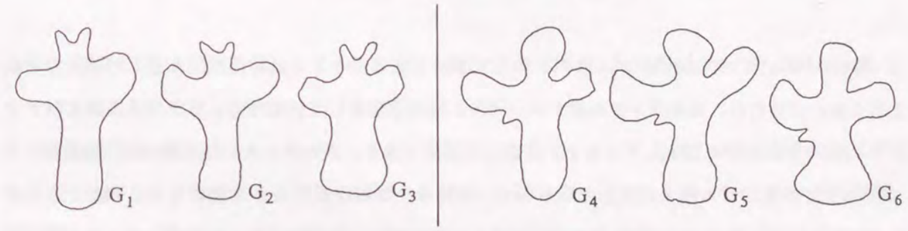
図 6.5 は、第 5 章 (図 5.14) で用いたサンプル図形に対して本章で述べた方法により得られた visual モデルを示したものである。図 6.5(a) は 2 クラスの原サンプル図形 (学習サンプル) であり、図 6.5(b) は各クラス毎に生成された一般化図形である。そして、図 6.5(c) はこれまで述べてきた方法により得られた各クラスの visual モデルである。visual モデルの各凹凸セグメントに付与された数値

は 6.3 節で述べた重要度を表す。得られた visual モデルは各クラスのプロトタイプとして、視覚的に妥当であるといえる。

次に図 6.6 に示す未知サンプルの認識実験を行なった。その結果を図 6.7 に示す。図 6.7 では、比較のために原サンプルと未知サンプルとの相違度と、visual モデルと未知サンプルとの相違度を示してある。



(a) 原サンプル (学習サンプル)



(b) 一般化図形 (各クラス毎に共通の凹凸構造が抽出されている)

(c)visual モデル (数値は重要度を表す)

図 6.5: 2 クラスの visual モデル生成例

図 6.6: 認識対象未知サンプル

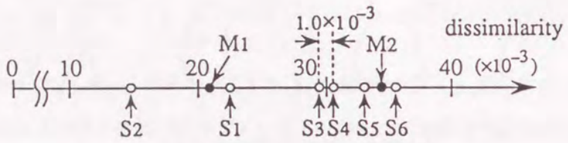


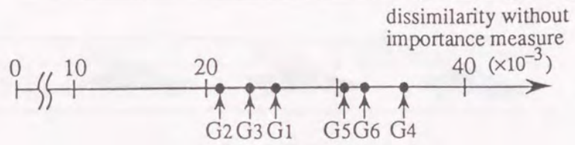
図 6.7: 未知サンプルの認識結果

図 6.7 から、この未知サンプルはクラス A に属すことになり、これは直観的に正しい結果であると言える。何故なら、未知サンプルはグローバルには十字形をしているので、クラス A、およびクラス B の両サンプル群に類似しているが、注意深く観察すると、未知サンプルの上部の凹凸構造がクラス A の全てのサンプル群と類似しているので、未知サンプルはクラス A に属すものと判断できるからである。また、原サンプル群と未知サンプルとの相違度と visual モデルと未知サンプルとの相違度とを比較すると、visual モデルとの相違度が、原サンプル群との相違度の平均的な値になっていることがわかる。これは、visual モデルがクラスのプロトタイプとして妥当であることを示唆するものである。

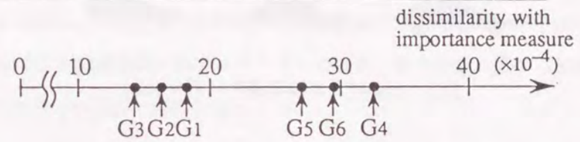
6.5.2 重要度の評価

次に 6.3 節で定義した重要度の評価を行なう。図 6.8 は図 6.5 に示した一般化図形と図 6.6 に示した未知サンプルとの相違度を重要度の有無で比較したものである。すなわち、図 6.8(a) は重要度を付与しなかった一般化図形と未知サンプルとの相違度を、図 6.8(b) は重要度を付与した一般化図形と未知サンプルとの相違度を各々示したものである。ある[†]。従って、図 6.8(a),(b) の相違度は各々式

[†]重要度の効果を比較するために一般化図形に重要度を付与した。但し、この場合、一般化図形相互間で対応する凹凸セグメントには同じ値の重要度が付与されることになる。



(a) 未知サンプルと一般化図形 (重要度付与無し) との相違度



(b) 未知サンプルと一般化図形 (重要度付与有り) との相違度

図 6.8: 重要度の効果

(4.33)、式 (6.12) を用いて算出している。図 6.8(b) の値が図 6.8(a) の値に比べ 1 桁小さいのは、前者では正規化された重み (重要度) 付き相違度計算を実行しているためである。

図 6.8(a),(b) を比較すると分かるように、重要度を付与することにより、より一般化の効果が大きくなっている。この結果を定量的に評価するために、第 5 章で用いたクラス分離度 κ を再度用いる。但し、この場合のクラス分離度は未知サンプルとの相違度から直接、式 (5.5) で計算できる。実際、前者のクラス分離度が $\kappa = 3.7$ であったのに比べ、後者のそれは、$\kappa = 7.3$ と大きくなった。この結果は、重要度を付与することにより、クラス特徴をより反映したモデル記述が実現できることを実証するものである。

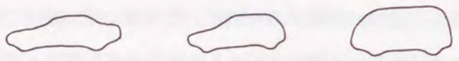
6.5.3 車種認識への適用

汎用形状認識の具体例として、本方式の車種認識への適用について述べる。図 6.9 は、カメラ入力した車の実画像を境界追跡して得た学習用サンプル (3 クラス (C_1: セダン (図 6.9(a))、C_2: ハッチバック (図 6.9(b))、C_3: ワゴン (図 6.9(c)))、各 3 サンプル) および、それらから得られた各クラスの visual モデル (図 6.9(d)) を示す。そして、表 6.1 に図 6.10 の未知サンプルと各 visual モデルとの相違度算出結果 (認識結果) を示す。

(a) セダンの学習サンプル (C_1)

(b) ハッチバックの学習サンプル (C_2)

(c) ワゴンの学習サンプル (C_3)



(d) 各クラスから生成された visual モデル (左から順にセダン、ハッチバック、ワゴンに対応する)

図 6.9: 車種認識のための学習サンプルと得られた visual モデル

図 6.10: 認識対象未知サンプル

表 6.1: 未知サンプルの認識結果

($\times 10^{-4}$)

Class	Sedan	Hatchback	Wagon
Weighted dissimilarity	6.0	44.6	59.6

表 6.1 から未知サンプルはセダンと正しく認識され、得られた visual モデルが認識のモデルとして有効であることがわかる。別の方法でも同様な車種認識が可能であるが、本手法は少数のサンプル図形を与えるだけで学習と認識を統一的アルゴリズムで扱え、かつ対象に関する知識を用いずに広範な図形認識を実現できるという点で意義がある。

6.6 結言

本章では、第 5 章で得られた一般化図形から、さらにクラスのプロトタイプである visual モデルを生成するアルゴリズムについて述べた。この visual モデルはクラスの全てのサンプル図形に共通する凹凸構造と、クラスの全てのサンプル図形の平均的な凹凸形状を持つ。また、クラスの特徴的な凹凸構造を部分輪郭毎に相対的に評価できる重要度を定義した。この重要度は、一般化図形生成時に得られる最適スケール値の関数となっている。この重要度を付与することにより、認識の際のクラス分離度がより大きくなり、形状認識の際に有効であることを実験的に確認した。

本章では、さらに得られた visual モデルの汎用形状認識への応用について述べ、その具体例として車の車種認識実験結果を示した。そして、実験結果により本手法の有効性を示した。

第 7 章

結論

本論文は、画像から得られる 2 次元輪郭形状からの形状モデルの自動獲得に関する研究について筆者がこれまで行なってきた研究成果をまとめたものである。以下にその成果を要約し、最後に今後の課題を述べる。

第 1 章では、画像理解におけるモデル獲得に関する研究の背景について述べるとともにそれらの問題点を明らかにした。次いで、それらの問題点を解決するための基本方針を示し、本論文の構成について述べた。

第 2 章では、輪郭図形のような線要素で構成される線画像の画質改善手法を提案した。まず、低品質線画では濃度ヒストグラムに基づく従来の 2 値化手法では限界があることを明らかに、多値の段階、すなわち 2 値化の前処理として濃度補正を施しておくことが必要であることを明らかにした。そしてその具体策として局所的な濃度形状変化に基づく簡易で実用性の高い濃度補正アルゴリズムを示した。提案手法を、低品質な線画像に適用した結果、従来手法では困難であった線の潰れとかすれをかなり抑制した良好な 2 値画像が得られることを明らかにした。また、提案した濃度補正を施すことにより、2 値化の閾値変動に対して安定な画像に変換されることを確認した。さらに、提案手法はハードウエア化が容易で、設備図面のような大規模な線画像を実用時間で処理でき得ることを示した。

第 3 章ではまず、濃淡画像中から着目対象物体の輪郭を抽出する方法について述べた。そして、最近提案されたエネルギー最小化アプローチ、active contour モデル (snakes) を概観し、このモデルが従来のボトムアップの輪郭抽出手法にはない優れた特徴を有していることを述べるとともに、そのエネルギー最小化解法における従来手法の問題点を明らかにした。そしてその問題を解決すべく、動的計画法に基づく定式化を試み、安定でかつ最適解を保証する新たな DP 方程式を導出した。また、解法における計算の複雑さについても言及した。そして、実際の濃淡画像を用いた解法の比較実

験から、提案解法の優位性を実証した。さらに、提案解法がコンピュータビジョンで有力視されている正則化問題の統一的解法への拡張性について述べた。

第 3 章ではさらに、動画像系列として与えられた濃淡画像群から着目対象輪郭を追跡する手法を提案した。提案手法は基本的には snakes のようなエネルギー最小化アプローチに基づいているが、ここでは弾性的制約を付与した弾性輪郭モデルを新たに定義した。snakes を用いた輪郭追跡は、ロングレンジの動画像、あるいは輪郭内部に複雑なテクスチャや遮蔽エッジが存在する場合には適用困難であったが、提案した弾性輪郭モデルでは良好な追跡が実現できることを明らかにした。提案手法は、背景が動いていても、また追跡対象物体 (の一部) が静止していても追跡が可能であり、さらに照明条件等の環境変動に強い等の特長を有し、画像理解における前処理 (初期処理) として有用なツールになり得ることを明らかにした。

第 4 章では、任意に与えられた 2 つの輪郭図形に対し、それらの妥当な凹凸構造の対応づけを行なう多重スケールセグメントマッチング手法を提案した。まず、変形の大きな図形間のマッチングでは、それに先立って図形を柔軟に記述しておくことの重要性を述べ、その具体策として多重スケール表現法を示した。この多重スケール表現は、最近提案されたスケールスペースの概念を用いているが、スケールスペース表現における計算量・データ量の問題を解決し、しかもスケール変化に伴う凹凸の変化の一貫性を保証するといった利点があることを明らかにした。

次いで、この多重スケール表現に基づく多重スケールセグメントマッチングアルゴリズムを示した。まず、従来手法のマッチング戦略である、coarse-to-fine 戦略の限界を明らかにした。そして、多重スケールの全てセグメントの組合せの中から最適な対応を見い出すという新たなマッチング戦略を提案し、そしてその解を効率良く得るための動的計画法に基づく算法を示した。実際の輪郭図形への適用実験を基に、従来困難であった変形の大きな図形間でも人間の直観にあった妥当な凹凸構造の対応づけが実現できることを示し、汎用的な 2 次元形状マッチング手法として有用であることを明らかにした。

第 5 章では、第 4 章の成果を基に、同一クラスに属す複数の 2 次元輪郭図形群からそれらに共通する最適スケールの凹凸構造の抽出、すなわち、パターン形状の一般化手法を提案した。まず、記号概念に基づく従来の一般化手法を概観し、それらの問題点を明らかにした。すなわち、従来手法では、いずれも形状をプリミティブとプリミティブ間の関係でグラフ表現しているため、一般化の際のグラフマッチングの問題、あるいは形状のセグメンテーションの問題等、従来アプローチの延長では本質的に解決困難な問題があることを指摘した。

そして、パターン形状を直接一般化することの重要性を述べ、その具体策として、パターン形状における多重スケールの凹凸構造の一般化という新たな一般化の枠組を提示した。提案した一般化手

法は、対象の構造に関する事前知識が不要であるため、任意の2次元輪郭図形に適用でき、また少数サンプルから効率良く一般化が可能である等の優れた特徴を有することを明らかにした。さらに、一般化の効果を評価すべく、判別分析の基準であるクラス分離度を導入し、定量的な解析を行なった。その結果、一般化によりクラス分離度の値が数倍大きくなり、提案した一般化手法の有効性を確認した。

第6章では、第5章で得られた一般化図形群からクラスのプロトタイプ図形、すなわち、visualモデルを生成するためのアルゴリズムを示した。このvisualモデルは同一クラスに属す全てのサンプルに共通する凹凸構造を持ち、かつクラスの全てのサンプルの平均的な凹凸形状を持つパターン形状モデルとして定義した。さらに、クラスの特徴的な凹凸構造の指標となる重要度を定義し、visualモデルの各凹凸セグメントに付与した。第5章で用いたクラス分離度により重要度の効果を定量的に評価したところ、未知サンプルの認識の際、クラスの特徴的な凹凸構造がより強調されるので良好な認識が実行できることを確認した。さらに、visualモデルの汎用形状認識への応用について述べ、その具体例として車のプロフィールからの車種認識の実験結果を示し、visualモデルのクラスのモデルとしての妥当性、認識への有効性を確認した。

画像情報からのモデル獲得、さらに抽象化して言うならば、パターン情報からの知識獲得(学習)の研究は入口にさしかかったばかりで成果を体系的に評価・整理するにはまだ熟していないのが現状である。本論文では、輪郭形状の凹凸構造に着目した学習アルゴリズムを提案したが、言うまでもなく、凹凸構造はパターン形状の万能な記述子ではない。しかしながら、本論文で提案したモデル獲得手法は、画像情報から、記号情報に変換することなく、直接、クラスのパターン形状モデルを自動獲得したという点で貴重な成果であると言える。また、本モデル獲得を実現すべく、新たに提案した各種要素技術は、モデル獲得のみならず、形状解析・認識のための有効なツールとなり得るものである。すなわち、本論文で得られた成果はパターン認識、コンピュータビジョン、さらに人工知能の各分野に貢献するものであると確信している。

本研究の次のステップとして、入力パターン集合に応じたパターン構造記述子の自動抽出が挙げられる。これは、すなわち、特徴抽出・選択そのものをも入力パターン集合の統計的構造からシステムに学習させる、より上位の学習の枠組を指す。こうした高度な学習機構を工学的に実現させる際、コンピュータアルゴリズムのような決定論的な形態では、これまでがそうであったように、限界があるように思える。では、一体どのようなアーキテクチャを想定すれば良いのだろうか。パターン情報に基づく学習研究の究極ともいえるこの問いは、同時にパターン認識の本質にも通じる。筆者自身、今後この問題のひとつの解を見つけるべく、新たな検討を進めていく所存である。

謝辞

本論文をまとめるにあたり大阪大学工学部通信工学科 手塚慶一教授には終始懇切丁寧なる御指導と、御教示を賜わりました。ここに、謹んで深謝の意を表します。また、内容について御指導、御助言を賜わりました大阪大学産業科学研究所の 北橋忠宏教授に心から感謝致します。大阪大学工学部通信工学科 森永規彦教授、倉薗貞夫教授には有益な御討論、御助言を賜わりましたことをここに記し、深く感謝致します。さらに、同通信工学科 馬場口 登 講師には本研究を論文としてまとめるに際し、少なからぬ御教示を賜わりました。心から御礼申し上げます。

本研究は、主に、NTTヒューマンインタフェース研究所において行なわれたもので、当初本研究の機会を与えて頂いた松下電気産業株式会社東京情報通信開発センター 塚田啓一 所次長(前NTTヒューマンインタフェース研究所所長)、NTTインテリジェントテクノロジー株式会社 小森和昭 社長(元視覚情報研究部部長)、NTTヒューマンインタフェース研究所 知能ロボット研究部 酒井高志 主幹研究員(元視覚情報研究部視覚認識グループリーダ)に感謝致します。

さらに、NTTヒューマンインタフェース研究所 小林幸雄 音声処理研究部部長(前視覚情報研究部部長)、同研究所 マルチメディア処理研究部(前視覚情報研究部) 末永康仁 博士、小杉 信 博士、石井健一郎 博士、ならび同研究部の皆様には、本研究の途上において種々の御討論および御協力を頂きました。特に、石井健一郎 博士にはパターン認識における学習の理論に関して有益な御議論を頂きました。諸氏に心から感謝致します。

本研究の直接の上司として御教示頂いたNTTヒューマンインタフェース研究所 名倉正計 博士(元画像応用研究室)、同研究所 鈴木 智 博士、間瀬健二 主任研究員に深く感謝致します。特に、鈴木 智 博士には本研究遂行時に有益な御討論を頂くと共に、本研究を論文としてまとめるにあたり激励の御言葉を頂きました。諸氏に心から御礼申し上げます。

参考文献

[1] 松山隆司, 長尾 真: "航空写真の構造解析", 情処学論, **21**, 5, pp. 468-480, 1980.

[2] 松山隆司, ビンセント・ハング: "画像理解システム SIGMA", 情処学論, **26**, 5, pp.877-889, 1985.

[3] Ohta Y.:"Knowledge-based interpretation of outdoor natural color scences", *Research Note in Artificial Intell.* **4**, Pitman, 1985.

[4] 恒川 尚, 下辻成佳:"図面読みとり装置 TOSGRAPH", 信学論 (D), **J68-D**, 4, pp. 466-472, 1985.

[5] Brooks R. A., Greider R. and Binford T. O.:"THe ACRONYM model-based vision system", in *Proc.* **IJCAI79**, pp.105–113, 1979.

[6] Brooks R. A. :"Model based three-dimensional interpretations of two-dimensional images", *IEEE Trans. Pattern Anal. Machine Intell.*, **PAMI-5**, 2, pp. 140–150, 1983.

[7] Winston P. H.: "Learning structural descriptions from examples", The Psychology of Computer Vision, P. H. Winston (Ed.), New York, McGraw Hill, ch. 5, 1975.

[8] Rosenblatt F.: "The perceptron: A probablistic model for information storage and organization in the brain", *Psychological review*, **65**, pp. 386–407, 1958.

[9] R. S. Michalski: "Pattern recognition ad rule-guided inductive inference", *IEEE Trans. Pattern Anal. Machine Intell.*, **PAMI-2**, pp. 349–361(1980).

[10] Mitchell T. M.: "Generalization as search", Artificial Intelligence, **18**, pp. 203–226, 1982.

[11] Brady M and Asada H.: "Smoothed local symmetries and their implementation", *Robotics research*, **3**, 3, pp. 36–61, 1984.

[12] Connell J. H.: "Learning shape descriptions: Generating and generalizing models of visual objects", AI-TR-853, *MIT Artificial Intelligence Laboratory*, Cambridge, MA, 1985.

[13] Connell J. H. and Brady M.: "Generating and generalizing models of visual objects", *Artificial Intelligence*, vol. **31**, pp. 159–183(1987).

[14] Winston P. H.: "Learning and reasoning by analogy", *Comm. ACM*, **ACM-23**, 12, pp. 689–703, 1980.

[15] Segen J.: "Model learning and recognition of nonrigid objects", in *Proc.* **CVPR89**, pp. 597–602, 1989.

[16] Weszka J. S, Nagel R. N., and Rozenfeld A.:"A threshold selection technique",*IEEE Trans. Comput.*, vol.**C-23**, 12,1974.

[17] 森 俊二, 大津展之:"認識問題としての 2 値化と各種方法の検討", 情報処理学会研資, イメージプロセッシング, **15**, 1,1977.

[18] 大津展之:"判別および最小 2 乗規準に基づく自動閾値選定法", 信学論 (D), **J63-D**, 4, pp. 349–356, 1980.

[19] 樋野匡利, 東田剛志, 太田友一, 坂井利之:"尾根点・谷点に基づく線図形の 2 値化方式", 昭和 57 年情報処理学会全大, 4E-1,1982.

[20] 上田修功, 名倉正計, 小杉 信, 森 克己:"図面の 2 値化のための画質改善", テレビ誌, **42**, 8, pp.831–836, 1988.

[21] 上田修功, 名倉正計, 星野肇夫, 森 克己:"手書き図面の画質改善法", 研究実用化報告, **37**, 3, pp.211–216, 1988.

[22] 河田悦生, 上田修功, 小川 博, 小杉 信:"手書き図面の図形認識法", 研究実用化報告, **37**, 3, pp.217–223, 1988.

[23] Okudaira M., Ueda N. and Aoki U.:"Image enhancement of handwritten drawings and their recognition followed by interactive processing", in *Proc.* **SPIE707**, pp.42–50, 1986.

[24] 三ツ矢英司, 末永康仁, 奥平雅士, 安田浩:"マルチプロセッサによる汎用画像処理装置", 信学論 (D), **J68-D**, 4, pp.869-876, 1985.

[25] Prewitt J. M. S. : "Object enhancement and extraction", in *Picture Processing and Pcychopictorics*, Lipkin B. S. and Rosenfeld A. (Eds.) New York:Academic Press, 1970.

[26] Duda R. O. and Hart P. E.: "Pattern Classification and Scene Analysis", New York:John Wiley & Sons, p.271, 1973.

[27] Robinson G. S.: "Edge detection by compass gradient masks", *Computer Graphics and Image Processing*, **6**, pp. 492–501, 1977.

[28] Roberts L. G.: "Machine perception of three-dimensional solids", in *Optical and Electro-optical Informatioon Processing*, J. P. Tippett et al. (Eds.) Cambridge, MA:MIT Press, 1965.

[29] Hueckel M. H.: "A local visual operatior which recognizes edges and lines", *J. ACM*, **20**, 4, pp. 634–647, 1973.

[30] Marr D. and Hildreth E.: "Theory of edge detection", in *Proc. R.Soc.Lond. B207*, pp. 187–213, 1980.

[31] Canny J. F.: "A computational approach to edge detection", *IEEE Trans. Patten. Anal. Machine Intell.*, **PAMI-8**, 6, pp. 679-698, 1986.

[32] Lowe D. G.: "Perceptual Organization and Visual Recognition", Boston:Kluwer Academic, 1985.

[33] Kass M., Witkin A. and Terzopoulos D.: "Snakes: Active contour models", *Int. J. Comput. Vision*, **1**, 4, pp. 321-331, 1988.

[34] 上田修功, 間瀬健二: "動的計画法による active contour エネルギーの最小化", 電子情報通信学会春季全大, D-554, 1991.

[35] 上田修功, 間瀬健二, 末永康仁: "エネルギー最小化に基づく輪郭追跡", 情報処理学会研資, CV73-5, 1989.

[36] 上田修功, 間瀬健二, 末永康仁: "弾性輪郭モデルとエネルギー最小化原理による輪郭追跡手法", 信学論 (D-II), 掲載予定.

[37] 積山, 横矢直和, 坂上勝彦: "Active net: 動的な網のモデル", 情報処理学会研資, CV63-2, 1989.

[38] 坂口嘉之, 美濃導彦, 池田克夫: "SNAKE パラメータの設定についての検討", 電子情報通信学会研資, PRU90-21, 1990.

[39] エルスゴルツ L. E. 著, 瀬川富士 訳: "科学者・技術者のための変分法", ブレイン図書出版, 1972.

[40] Amini A. A., Weymouth T. E. and Jain R. C.: "Using dynamic programming for solving variational problems in vision", *IEEE Trans. Pattern Anal. Machine Intell.*, **PAMI-12**, 9, pp. 855–867, 1990.

[41] Amini A. A., Tehrani S. and Weymouth T.: "Using dynamic programming for minimizing the energy of active contours in the presence of hard constraints", in *Proc. Second Int. Conf. Computer Vision*, **2ndICCV**, Tarpon Springs, FL, Dec. 1988.

[42] Poggio T., Torre V. and Koch C.: "Computational vision and regularization theory", *Nature*, **317**, 6035, 1985.

[43] Bellman R.: "Dynamic Programming", Princeto, NJ:Princeton University Press, 1957.

[44] Ullman S.: "The interpretation of visual motion", Cambrige, MA:MIT Press, 1979.

[45] Horn B. K. P. and Schunck B. G.: "Determining optical flow", *Artificial Intelligence*, Vol. **17**, pp. 185–203, 1981.

[46] Barnard S. T. and Thompson W. B.: "Disparity analysis of images", *IEEE Trans. Pattern Anal. Machine Intell.*, **PAMI-2**,4, pp. 333–340, 1980.

[47] Sethi I. K. and Jain R.: "Finding trajectories of feature points in a monocular image sequence", *IEEE Trans. Pattern Anal. Machine Intell.*, **PAMI-9**,1, pp. 56-73, 1987.

[48] Salari V. and Sethi I. K.: "Correspondence in presence of occlusion", *IEEE Trans. Pattern Anal. Machine Intell.*, **PAMI-12**,1, pp. 87–91, 1990.

[49] Cheng C. L. and Aggarwal J. K.: "A two-stage hybrid approach to the correspondence problem via forward-searching and backward-correcting", in *Proc.*, **ICPR90**, pp. 173–179, Atlantic City, June, 1990.

[50] Dreschler and Nagel H. H.: "Volumetric model and 3D trajectory of a moving car derived from monocular tv-frame sequence of a street scene", in *Proc.*, **IJCAI81**, Vancouver, Aug, 1981.

[51] Yuille A. L., Cohen D. S. and Hallinan P. W.: "Feature extraction from faces using deformable templates", in *Proc.*, **CVPR89**, pp. 104–109, San Diego, June, 1989.

[52] 光本浩士, 岡崎耕三, 梶見直樹, 田村進一, 河合秀夫, 副井 裕: "エネルギー関数とオプティカルフローを用いた口唇輪郭の抽出・補完と追跡", 情処論, **31**, 3, pp. 444–453, 1990.

[53] 山田博三, 山本和彦, 松浦雅彦: "2 次元 DP 整合法による超音波腎臓像の認識", 信学論 (D), **J68-D**, 9, pp. 1649–1656, 1985.

[54] 山田博三: "並列演算による方向パターン弾性整合法の提案 -MAP 整合法", 信学論 (D-II), **J73-DII**, 4, pp. 553–561, 1990.

[55] 喜多泰代: "立位充満正面像からの予測形を利用する 2 重造影像の胃領域輪郭抽出", 信学論 (D-II), **J74-DII**, 7, pp.873–886, 1991.

[56] Rosenfeld A. and Pfaltz J. L.:"Distance functions on digital pictures", *Pattern Recognition*, **1**, pp.33-61(1968).

[57] Witkin A. P.: "Scale space filtering", in *Proc., 8th* **IJCAI**, pp. 1019–1022, karlsruhe, 1983.

[58] Witkin A. P.: "scale space filtering", in *Proc., 8th* **IJCAI**, pp. 1019–1022, karlsruhe, 1983.

[59] Mokhtarian F. and Mackworth A. K.: "Scale-based description and recognition of planar curves and two dimensional shapes", *IEEE Trans. Pattern Anal. Machine Intell.*, **PAMI-8**, 1, pp. 34–43, 1986.

[60] 佐藤, 和田: "構造線による一般化波形の階層的表現", 信学論 (D), **J70-D**, 11, pp. 2154-2159, 1987.

[61] 酒匂, 依田, 江尻: "多重解像度表現を用いた変形波形の照合アルゴリズム", 信学論 (D), **J71-D**, 11, pp. 2311-2318, 1987.

[62] 上田修功, 鈴木智: "多重スケールの凹凸構造を用いた変形図形のマッチングアルゴリズム", 信学論 (D-II), **J73-DII**, 7, pp. 992–1000, 1990.

[63] Ueda N. and Suzuki S.: "A matching algorithm of deformed planar curves using multiscale convex/concave structures", *Systems and computers in Japan*, Scripta Technica, Inc. **22**, 5, pp. 94–104. 1991.

[64] 上田修功, 鈴木智: "多重スケールの凹凸構造マッチング: MC マッチング法", *NTT R& D*, **40**, 3, pp.399–406. 1991.

[65] Zahn C. T. and Roskies R. Z.: "Fourier descriptors for plane closed curves", *IEEE Trans. Computer*, **C-21**, 3, pp. 269–281, 1971.

[66] Rosenfeld A. and Johnston E.: "Angle detection on digital curves", *IEEE Trans. Computer.*, **C-9**, pp. 875–873, 1973.

[67] Freeman H. and Davis L. S.: "A corner finding algorithm for chain coded curves", *IEEE Trans. Computer*, **C-26**, pp. 297–303, 1977.

[68] 飯島泰蔵: "図形の基礎方程式と観測変換", 信学論 (C), **54-C**, 7, pp. 641–648, 1971.

[69] 趙, 飯島: "安定視点木法による画像の特徴抽出構造線による一般化波形の階層的表現", 信学論 (D), **J68-D**, 5, pp. 508–514, 1985.

[70] A. K. Mackworth and F. Mokhtarian: "The renormalized curvature scale space and the evolution properties of planar curves", in *Proc.*, **CVPR88**, pp. 318–326, Boston, June, 1988.

[71] Babaud J., Witkin A. P. and Baudin M.: "Uniqueness of the gaussian kernel for scale-space filtering", *IEEE Trans. Pattern Anal. Machine Intell.*, **PAMI-8**,1, pp. 34–43, 1986.

[72] Yuille A. L. and Poggio A.: "Scaling theorems for zero crossings", *IEEE Trans. Pattern Anal. Machine Intell.*, **PAMI-8**, 1, pp. 15–25, 1986.

[73] Asada H. and Brady M.: "The curvature primal sketch", *IEEE Trans. Pattern Anal. Machine Intell.*, **PAMI-8**, 1, pp. 2–14, 1986.

[74] Baker H. H. and Binford T. O., "Depth from edge and intensity based stereo," in *Proc. 7th IJCAI*, pp. 631–636, 1981.

[75] Drumheller M. and Poggio T., "On parallel stereo," in *Proc. IEEE Conf. Robotics & Automation*, pp. 1439–1447, 1986.

[76] Sakoe H. and Chiba S., "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Trans. Acoust. Speech Signal Process.*, **ASSP-26**, 1, pp. 43-49, 1978.

[77] Ohta Y. and Kanade T., "Stereo by intra-and inter-scanline search using dynamic programming," *IEEE Trans. Pattern Anal. Machine Intell.*, **PAMI-7**, 2, pp. 139–154, 1985.

[78] Maitre H. and Wu Y., "Improving dynamic programming to solve image registration," *Pattern Recognition*, **20**, 4, pp. 443–462, 1987.

[79] Milios E. E, "Shape matching using curvature processes," *Comp. Vision Graphics Image Process.*, **47**, pp. 203–226, 1989.

[80] 上田修功, 鈴木 智: "スケールスペースを用いた輪郭図形記述の一般化", 電子情報通信学会春季全大, D-506, 1989.

[81] 上田修功, 鈴木 智: "スケールスペースを用いた輪郭図形記述の一般化", 信学技報, **PRU85**-95, pp.47–59, 1989.

[82] 上田修功, 鈴木 智: "多重スケールの凹凸構造に基づく変形図形のマッチングアルゴリズムとそのモデル自動獲得への応用", 視聴覚情報研究会, **11**, 1990.

[83] N. Ueda and S. Suzuki: "Automatic shape model acquisition using multiscale segment matching", in *Proc.* **ICPR90**, pp. 897–902, 1990.

[84] 上田修功, 鈴木智: "凹凸構造の一般化に基づく輪郭形状モデルの自動獲得", 信学論 (D-II), **J74-DII**, 2, pp. 220–229, 1991.

[85] Ueda N. and Suzuki S.: "Automatic shape model acquisition based on a generalization of convex/concave structures", *Systems and computers in Japan*, Scripta Technica, Inc. to be appeared.

[86] Blum H.: "Biological shape and visual science(part 1)", *Journal of Theoretical Biology*, **38**, pp. 205–287, 1973.

[87] Blum H. and Nagel R. N.: "Shape description using weighted symmetric axis features", *Pattern Recognition*, **10**, pp. 167–180, 1978.

[88] Dietterich T. G. and Michalski R. S.: "A comparative review of selected methods for learning from examples", In R. S. Michalski, Carbonell J. G. and Mitchell T. M.(EDs.), *Machine Learning*, Los Altos, CA: Morgan Kaufmann, 1983.

[89] Lowe D. G.: "Organization of smooth image curves", in *Proc.* **ICCV88**, pp. 558–567, 1988.

[90] Fukunaga K.: "Introduction to Statistical Pattern Recognition", *Academic Press*, chap.9, 1972.

[91] 高根芳雄: "多次元尺度法", 東京大学出版会, 1978.

[92] Koenderink J. J.: "The structure of images", *Biol. Cybern.*, vol. **50**, pp. 363–370, 1984.

本論文に関する原著論文

[学術論文]

(1) 上田修功, 名倉正計, 小杉 信, 森 克己:"図面の2値化のための画質改善", テレビ誌, 42, 8, pp.831–836, 1988.

(2) 上田修功, 鈴木智: "多重スケールの凹凸構造を用いた変形図形のマッチングアルゴリズム", 信学論, J73-DII, 7, pp. 992–1000, 1990.

(3) 上田修功, 鈴木智: "凹凸構造の一般化に基づく輪郭形状モデルの自動獲得", 信学論, J74-DII, 2, pp. 220–229, 1991.

(4) 上田修功, 間瀬健二, 末永康仁: "弾性輪郭モデルとエネルギー最小化原理による輪郭追跡手法", 信学論, 採録決定済み.

(5) Ueda N. and Suzuki S.: "Learning visual models from shape contours using multiscale convex/concave structure matching", *IEEE Trans. Pattern Anal. Machine Intell.*, to be appeared.

[国際会議]

(6) Okudaira M., Ueda N. and Aoki U.: "Image enhancement of handwritten drawings and their recognition followed by interactive processing", *Proc. SPIE-707*, pp.42–50, 1986.

(7) Ogawa H., Kawada E., and Ueda N.: "Application of image processing equipment with multiprocessors to line-drawing recognition", *Proc. SPIE-845*, pp.97–103, 1987.

(8) Ueda N. and Suzuki S.: "Automatic shape model acquisition using multiscale segment matching", *Proc. ICPR90*, pp. 897–902, 1990.

[機関論文誌等]

(9) 上田修功, 名倉正計, 星野肇夫, 森 克己:"手書き図面の画質改善法", 研究実用化報告, 37, 3, pp.211–216, 1988.

(10) 河田悦生, 上田修功, 小川 博, 小杉 信:"手書き図面の図形認識法", 研究実用化報告, 37, 3, pp.217–223, 1988.

(11) 上田修功, 鈴木智: "多重スケールの凹凸構造マッチング: MCマッチング法",*NTT R& D*, 40, 3, pp.399–406. 1991.

(12) Ueda N. and Suzuki S.: "A matching algorithm of deformed planar curves using multiscale convex/concave structures", *Systems and computers in Japan*, Scripta Technica, Inc. 22, 5, pp. 94–104. 1991.

(13) Ueda N. and Suzuki S.: "Automatic shape model acquisition based on a generalization of convex/concave structure", *Systems and computers in Japan*, Scripta Technica, to be appeared.

[大会・研究会]

(14) 上田修功, 名倉正計, 森 克己: "多値読み取り図面の画質改善手法の検討", 昭和60年度電子通信学会総合全国大会, 1202.

(15) 上田修功, 名倉正計: "手書き図面の画質改善手法", 電子情報通信学会技術報告, IE85-95, pp.31–36, 1985.

(16) 上田修功, 鈴木 智: "スケールスペースを用いた輪郭図形記述の一般化", 平成元年度電子情報通信学会総合全国大会, D-506.

(17) 上田修功, 鈴木 智: "スケールスペースを用いた輪郭図形記述の一般化", 電子情報通信学会技術報告, PRU85-95, pp.47–59, 1989.

(18) 上田修功, 鈴木 智: "2次元形状モデルの自動獲得手法の検討", 平成元年度情報処理学会総合全国大会, 2E-8.

(19) 上田修功, 鈴木 智: "スケールスペースを用いた輪郭図形記述の一般化", 電子情報通信学会技術報告, PRU89-7, pp.47–54, 1989.

(20) 上田修功, 鈴木 智: "輪郭図形の形状モデル自動獲得とその図形認識への応用", 平成2年度情報処理学会総合全国大会, 3M-8.

(21) 上田修功, 鈴木 智: "多重スケールの凹凸構造に基づく変形図形のマッチングアルゴリズムとそのモデル自動獲得への応用", 視聴覚情報研究会, 11, 1990.

(22) 上田修功, 間瀬 健二: "動的計画法による active contour エネルギーの最小化", 平成3年度電子情報通信学会総合全国大会, D-554.

(23) 上田修功, 間瀬健二, 末永康仁: "エネルギー最小化に基づく輪郭追跡", 情報処理学会, CV73-5, 1991.

(24) 上田修功, 間瀬健二, 末永康仁: "弾性制約モデルによる輪郭追跡手法", 平成3年度電子情報通信学会秋期全国大会, D-220.