



子-339

博士論文

プリミティブ分解によるユーザの視点に基づいた画像検索

指導教官 浜田 喬 教授

平成 7 年 12 月 20 日 提出

東京大学 工学系研究科電子工学専攻

学籍番号 37113

金 原 史 和

# 目次

<b>1</b>	<b>序 論</b>	<b>1</b>
1.1	マルチメディアデータベース	2
1.2	画像データベースの研究動向	3
1.3	画像の内容検索	4
1.4	本研究の目的と意義	4
1.5	本研究の概要	5
1.6	本論文の構成	5
<b>2</b>	<b>関連研究</b>	<b>7</b>
2.1	はじめに	8
2.2	大局的な類似度に着目した検索	8
2.3	部分的な類似度に着目した研究	9
2.3.1	どの部分か?	9
2.3.2	どのような特徴か?	10
2.3.3	どの部分のどのような特徴?	10
2.4	本研究の方針	11
<b>3</b>	<b>画像検索システム VPoint</b>	<b>13</b>
3.1	はじめに	14
3.2	構想	14
3.2.1	検索インタフェース	14
3.2.2	検索条件の表現	14
3.2.3	プリミティブ分解	15
3.3	構成及び検索の流れ	15
3.4	検索インタフェース	18
3.5	検索処理	20
3.5.1	シグネチャ表現	20
3.5.2	問い合わせシグネチャ	22
3.5.3	マッチング処理	23

3.6	実装	25
4	シルエット画像検索	26
4.1	はじめに	27
4.2	形状分解手法	28
4.2.1	分解アルゴリズムの概要	30
4.2.2	前処理	31
4.2.3	曲率の設定	32
4.2.4	浸食を表現するベクトルの計算	32
4.2.5	分割線の決定	34
4.2.6	併合処理	35
4.3	分解例	36
4.4	特徴記述	38
4.4.1	スケルトン抽出	38
4.4.2	用いる特徴量とシグネチャへの変換	39
4.5	検索例	41
5	カラー画像検索	59
5.1	はじめに	60
5.2	領域分割による画像の構造化	60
5.2.1	領域分割処理	60
5.2.2	プリミティブ抽出アルゴリズムの概要	60
5.2.3	前処理 (Preprocessing)	60
5.2.4	エッジ抽出 (Edge detection)	61
5.2.5	領域分割処理 (Segmentation)	62
5.2.6	併合処理 (Merge)	62
5.2.7	プリミティブの抽出 (Primitive extraction, Shape decomposition)	63
5.2.8	後処理 (Revision)	66
5.2.9	パラメータの設定	66
5.3	国旗を対象とした検索	69
5.3.1	プリミティブ抽出処理	69
5.3.2	特徴記述	71
5.3.3	検索例	72
5.4	自然画を対象とした検索	90
5.4.1	プリミティブ抽出処理	90
5.4.2	特徴記述	90
5.4.3	検索例	91

目次	iii
<b>6 評価</b>	<b>104</b>
6.1 はじめに	105
6.2 定性的評価	105
6.2.1 Vpoint の特徴	105
6.2.2 従来の研究との本研究との比較	106
6.3 定量的評価	107
6.3.1 検索時間	107
6.3.2 検索精度	107
6.3.3 所見	111
<b>7 結論</b>	<b>112</b>

# 目次

3.1	システム構成及び検索の流れ	17
3.2	検索インタフェース	19
3.3	シグネチャの形式	21
3.4	ゆらぎを含む問い合わせシグネチャ	22
3.5	マッチング処理 (1)～プリミティブの対応～	23
3.6	マッチング処理 (2)～2部グラフマッチング～	23
3.7	重みの必要性	24
4.1	どれが類似しているか?	27
4.2	形状分解	27
4.3	凹凸による part の判定	28
4.4	凹部に挟まれた凸部を part とする分解例	29
4.5	形状分解のモデル	29
4.6	分割のフローチャート	30
4.7	前処理	31
4.8	曲率	32
4.9	浸食処理	33
4.10	角の決定	33
4.11	浸食を表現するベクトルの計算結果例	34
4.12	浸食時間の決定	35
4.13	併合処理	36
4.14	分解例 (1)	36
4.15	分解例 (2)	37
4.16	計算量	38
4.17	スケルトン抽出	39
4.18	スケルトン抽出例	40
4.19	シルエット画像検索例 (1a)	45
4.20	シルエット画像検索例 (1b)	46
4.21	シルエット画像検索例 (1c)	47

4.22 シルエット画像検索例 (2)	48
4.23 シルエット画像検索例 (3)	49
4.24 シルエット画像検索例 (4)	50
4.25 シルエット画像検索例 (5a)	51
4.26 シルエット画像検索例 (5b)	52
4.27 シルエット画像検索例 (6)	53
4.28 シルエット画像検索例 (7)	54
4.29 シルエット画像検索例 (8)	55
4.30 シルエット画像検索例 (9)	56
4.31 シルエット画像検索例 (10a)	57
4.32 シルエット画像検索例 (10b)	58
5.1 分割処理のフローチャート	61
5.2 エッジ領域と併合領域	63
5.3 プリミティブの抽出	64
5.4 不十分なプリミティブの判定	65
5.5 領域数と併合コストの関係	68
5.6 雑音による誤った分割	69
5.7 国旗画像のプリミティブ抽出例	70
5.8 国旗画像検索例 (1)	75
5.9 国旗画像検索例 (2)	76
5.10 国旗画像検索例 (3)	77
5.11 国旗画像検索例 (4a)	78
5.12 国旗画像検索例 (4b)	79
5.13 国旗画像検索例 (5)	80
5.14 国旗画像検索例 (6)	81
5.15 国旗画像検索例 (7a)	82
5.16 国旗画像検索例 (7b)	83
5.17 国旗画像検索例 (7c)	84
5.18 国旗画像検索例 (7d)	85
5.19 国旗画像検索例 (7e)	86
5.20 国旗画像検索例 (8)	87
5.21 国旗画像検索例 (9)	88
5.22 国旗画像検索例 (10)	89
5.23 自然画のプリミティブ分割例	90
5.24 自然画像検索例 (1)	93
5.25 自然画像検索例 (2)	94

5.26 自然画像検索例 (3)	95
5.27 自然画像検索例 (4)	96
5.28 自然画像検索例 (5)	97
5.29 自然画像検索例 (6a)	98
5.30 自然画像検索例 (6b)	99
5.31 自然画像検索例 (7)	100
5.32 自然画像検索例 (8)	101
5.33 自然画像検索例 (9)	102
5.34 自然画像検索例 (10)	103

# 第 1 章

## 序 論



## 1.1 マルチメディアデータベース

近年、大容量記憶装置の発達、高解像度ディスプレイの出現などにより、計算機上で画像データを柔軟に扱えるようになったと共に、ネットワーク技術の進歩のおかげで、従来のテキストや音声を中心とした通信から、テレビ会議、ビデオ・オン・デマンドなどの画像、映像を交えた通信が可能になってきた。いわゆるマルチメディアと称されるものである。その一つの応用としてマルチメディアデータベースがあり、これに対する需要も急速に増しているといえる。

ではマルチメディアデータベースとは何であろうか。この言葉の正確な定義は特にないが、簡単にいえば「テキスト以外のメディア、特に音声及び画像や映像をも柔軟に扱うデータベース」といえるだろう。本論文のテーマである画像データベースも、このマルチメディアデータベースの一つである。しかし、記憶方式、データ管理方式、通信方式、検索方式などのマルチメディアデータベースの技術課題のほとんどは、画像・映像の扱いに集中しているといっても過言ではない。その意味からマルチメディアデータベースといった場合、技術的な本質においては画像・映像データベース（以下、画像データベースと統一して表記する）を指すといつてよいであろう。

それでは画像データベースとは一体どういうものでしょうか。しごく簡単に「大量の画像や映像データを納めたシステム」としてしまっても可能であるが、それでは単に計算機上の一つのディレクトリに大量な画像ファイルを蓄積してあるものも該当してしまう。これは単なる画像ファイルシステムであり画像データベースとは呼び難い。単しくもデータベースというからには、大量のデータを効率よく管理しておき、複数の利用者が所望の情報を所望の形態で検索したり、更新、破棄、合成ができるものでなければならない。それでは、なぜ文献データベースのような従来の文字・数値情報を中心としたデータベースと区別して、わざわざ画像とか映像などの語を用いるのであろうか。それは画像や映像情報が以下のような特徴を持つからである [1, 2].

- 扱う情報の物理量が文字数値情報に比べて膨大である。例えば1画素1バイトで500×500ドットの画像を100枚そのまま蓄積すると25MBとなる。映像ならばさらに膨大になる。
- データが二次元または三次元性を有し、内包する情報量が多種多様である。例えば画像内に現れる物体の形状・位置・距離・動きなどの情報が入り込む。
- 一般的にデータが抽象化されておらず、人間の視覚機能を通じて情報化される場合が多いため表現方法が難しい。例えば「青い服を着ている人物」を撮影した画像を表現しようとした場合、統一的な記述を行うことが困難である。

この他にも用途や目的に応じた細かな特徴はあるが、概ねこのような特徴があるために、従来の文字・数値中心のデータベースシステムやデータモデル（関係データモデルなど）では限界があり、新たな技術革新が望まれているわけである。

## 1.2 画像データベースの研究動向

では画像データベースの研究及び技術開発の現状はどうであろうか。本研究に関連する研究例についての詳細は2章で述べるとして、本節では本研究の位置付けを明確にする意味で画像データベース一般の研究動向について簡単に解説する。

画像データベースの研究には大きく二つのアプローチのとり方がある。即ち画像を中心とするか、データベースを中心とするかである。前者は画像処理・画像認識などの技術を開発または応用して、画像からの情報抽出や検索技術に重点を置くものである。一方、後者では抽出された情報の表現方式・データモデル・記憶管理方式などに焦点が当てられる。

1980年代後半くらいまでの初期の研究においては、画像データベースという通常の関係データベースに画像データをポインタとしてリンクしたもの [3] や、キーワードを用いてどれだけ効率良く画像検索を行なうかといった研究 [4, 5, 6] や、地図を対象とした地図データベースや地図情報システム (GIS) に関連した研究が中心 [7, 8, 9, 10] であった。

画像データをリンクさせる方法は实际的であり、現在でも商用化されうるほとんどがこの形態を採用しているか、もしくはハイパーメディアのような方式を用いている。

地図を対象としたものは、二次元のジオメトリックな情報を扱えることができ、例えば「石山駅から半径1km以内にある病院を全て検索せよ」というような検索が可能である。このような検索が効率良く行えるように検索言語やデータ管理方式について様々な研究が精力的になされた。しかし地図は既にある程度抽象化された情報の集合であり、地図データベースは画像データベースの特殊な一形態であるともいえる。

このように、最初はデータベースを中心として研究が始まったが、1980年後半から1990年代に突入すると、ハードウェアの目覚ましい発展も手伝い、画像工学の方面からの研究が盛んになった。画像処理・画像解析・画像認識・画像理解の技術を利用して、画像からデータベースに用いる情報を抽出する試みや、画像間の類似度を測ることで画像メディア特有の検索を可能にするための研究がなされた。これらは画像データベースのうちの画像検索技術に主に焦点が当てられており、画像の内容検索 [11, 12] とも呼ばれる。もちろん、データベース方面からの研究でも、オブジェクト指向モデルやハイパーメディアなどを用いたデータモデル、検索方式及び記憶方式などの研究 [13, 14, 15, 16, 17] が多く行われた。

今現在を含む1990年代中期もこの傾向は変わらないが、画像の検索に関しては扱う対象画像の範囲が多岐にわたり、また提供される検索形態も柔軟性を増してきた。また映像を対象とした研究 [18, 19] や感性や心理空間に基づいた検索 [20] についての研究も行われるようになった。

このように柔軟かつ効率的な画像データベースを構築するには画像工学の技術及びデータベースの技術双方ともに欠かせぬものであるが、それらが抱える課題は依然多い。特に画像の内容に基づいた検索を実現するために越えねばならない障壁は非常に高いというのが現状である。

## 1.3 画像の内容検索

既に述べたように、画像データベースを構築する際に解決すべき最も重要な課題の一つとして、その検索方法が挙げられ、画像工学の側面からの研究が現在盛んである。本研究においても、この画像検索を議論の中心に据えている。現在実用化レベルにある最も一般的な画像検索の方法は、人手により画像データにキーワードを付与し、それを検索に用いる方法である。この付与するキーワードには二次情報と一次情報がある。二次情報は、いわゆる撮影日や撮影者などの書誌情報であり人手による付与が避けられない情報である。これに対し一次情報は画像そのものが持つ情報であり、例えば動物の画像であるとか野球のシーンであるといった画像の内容に基づいた情報である。

しかし画像の内容に関してキーワードを付与する場合には二つの大きな問題点がある。一つは大量の画像データに対する人手によるキーワード付与の労力と客観性の問題、もう一つは言葉では表現し難い画像メディア特有な検索要求への対応の問題である。前者についてはシーン解析や画像認識によるキーワードの自動抽出に関する研究 [21, 22] が行われているが、実用にはまだまだ道のりが遠いというのが実情である。後者に関してはスケッチ検索やナビゲーション検索などの例示画検索の研究が主流であり、技術的には類似検索という言葉でまとめられるが、これも対象を特定したものを除いては満足する検索は実現されていない。将来、柔軟な画像検索を実現するには、上記のキーワードなどの言葉による検索形態と例示画検索のような言葉に依らない検索形態が個々にまたは統合してユーザに提供されるべきであると考えられる。

しかし、このいわゆる画像の内容検索はコンピュータビジョンの技術と関連する所が大きく実用化はまだまだ難しいとされている。画像処理・画像認識などの技術がそれだけ困難なのである。しかし画像検索においては、純粋な画像認識や理解ほどの精度が求められないことが多い。つまり、誤った検索情報が含まれていたとしても所望のデータが検索されればよいのである。またテキストと異なり、人間が画像を認識・識別するのは瞬時であるので、最後の検索はブラウジングによって人間に判別させればよい。これは文献検索などの場合よりも人間にとっては容易な作業である。

つまり、画像の内容検索とは現在伸び悩んでいる画像処理や画像認識の技術を有効に活かせる一つの大きな応用先ともいえる。また言葉では表現しがたい画像メディア特有の検索を可能にするためにも必須のテーマである。

さて、この画像の内容検索こそが本論文で論じるテーマなのである。以下ではこのうちの類似画像検索に焦点をあてて議論する。

## 1.4 本研究の目的と意義

では具体的に言葉で表現し難い検索要求とはどういうものか。例えば、指紋の照合の際に指紋の形状を言葉で説明できるものではない。また医療分野のレントゲン画像も、同じような体形の患者、肺に現れる陰が類似した患者というように、様々な観点からの分類ができれば有用である。さらにデザインの分野や書誌情報が不明だが画像の概略や部分的な特徴をキーとして検索したい

場合なども言葉では表現しがたいものがある。

このような検索要求を満たすには、システムに入力する検索キーの形態が一つの重要な要素となる。過去の研究例を見ると、(a) サンプルの画像そのものを入力するもの、(b) ユーザが描画するスケッチ情報を用いるもの（スケッチ検索）、(c) ユーザーが配置するアイコンを用いるもの（アイコン検索）、(d) 複数の画像や部分的な画像の情報を用いてナビゲーションするもの（ナビゲーション検索）などが挙げられる。(a) の場合は画像と画像の照合に相当し、指紋や商標などの応用には適している。一方 (b) (c) (d) については単なる画像と画像との照合ではなくてユーザの操作性が入り込むので、それだけ検索対象及び検索要求の幅が広がる可能性がある。逆にユーザの記憶や意図を考慮した場合には、このような形態にならざるをえないといえよう。

一般にこのような画像特徴に対するユーザの検索要求は多分に曖昧でありかつ多様である。例えばスケッチ検索において、求める画像を正確に再現することはまず無理であり、どのような特徴に着目してスケッチが行われるかも推測し難い。また、描画された例示画内にはユーザにとって明確な部分もあれば曖昧な部分もある。これはナビゲーションやアイコン検索に関しても同様である。

しかるに、次章で見ると従来多くの研究例においては、ユーザに許される検索要求は限られていたといえ、ユーザが画像のどの部分のどのような特徴に着目して検索しようとしているのかが十分には反映されていない。これは難しい問題ではあるが、柔軟な画像検索を実現するには重要な課題である。特にユーザの漠然とした断片的な記憶を手がかりとした検索には望まれることである。

本研究の主眼はこの点にあり、即ち、ユーザの視点や意図を反映することで柔軟な画像の内容検索を実現する、ことが本研究の目的である。

## 1.5 本研究の概要

以上の背景を基に、本研究では上記のコンセプトを実現できるような画像検索プロトタイプシステム VPoint（仮称）を構築した。VPoint では、プリミティブという低レベルの単位に画像を分解することで画像を構造化し、その画像情報を用いてスケッチ及びナビゲーション形態の検索を提供するものである。本研究の特徴は、ユーザの視点や意図を曖昧度として扱う点、それに伴う効率的なマッチング機構、ユーザの意図や視点を明示的に指示できる検索インタフェース、画像のプリミティブ分解手法、さらにこれらを統合したシステム構築にある。この他、人手による作業を含んでいないこと、何らかの対象特有の知識やモデルを用いていないことも特徴とするところである。現在までに、シルエット画像、国旗画像及び一般の自然画像の三つを対象として実装を行なった。

## 1.6 本論文の構成

以下、本論文の構成について述べる。次の2章においては、上述した課題に沿って従来の研究例を概観するとともに本研究の位置付け及び新規性を検証する際の参考とする。3章ではシステ

ムの構想及び構成，そして検索処理の流れについて述べ，データに依存しない共通部分であるインデックス化・マッチング手法・検索インターフェースの実装について述べる．4章と5章では各々シルエット画像及び国旗画像と自然画像のカラー画像を対象とした実装，特徴抽出手法について述べ，検索例を示す．そして，それらの総括した評価・考察を6章で述べ，7章をもって結びとする．

## 第2章

# 関連研究

## 2.1 はじめに

前章で述べたように、本研究の目的は、画像のどの部分のどのような特徴にユーザが着目しているかというユーザの視点を柔軟に反映した画像の類似検索を実現することにある。以下、この趣旨に沿って関連する従来の研究を概観すると共に、本研究の意義及び新規性を明確にする。

## 2.2 大局的な類似度に着目した検索

従来の類似画像検索の研究では主に大局的な類似度に着目した研究が多かった。まずそれらの研究例を個々に簡単に紹介する。

**指紋検索 [23]** 指紋に現れる識別可能な特徴点を調べることにより照合を行う。同一指紋か否かの識別を目的とするので客観性があり、特にユーザの視点を考慮する必要はない。

**TRADEMARK [24]** 意匠や商標を対象とした検索システムであり完成度が高いシステムである。画像をブロック分割して、ブロックごとの黒画素分布などの特徴量を利用している。サンプルの画像を問い合わせ画像としてもよいし、スケッチ検索も可能である。

**土器検索 [25]** 土器を対象とした研究であり、土器の構成部品をノードとして全体の形状情報をグラフ構造で表現している。個々の部品の特徴としては大きさやフーリエ記述子による特徴を用いており、類似度はグラフマッチングによって測定される。

**工具のシルエット画像検索 [26, 27]** 工具の輪郭に現れる局所的な角度情報を類似度測定に用いている。さらに分類木 feature tree の構成方法を提案しており、検索の効率化を図っている。扱う特徴はかなり局所的であるが類似度の測定は大局的である。

**長方形分解によるシルエット画像検索 [28]** 人工的に作成したシルエット画像データを長方形に分解することで形状を表現し、それらの位置関係や重心の情報を用いて類似度を測定している。

**蝶のシルエット画像検索 [29]** 蝶の輪郭をペリフェラル特徴に基づいて表現し、それを心理空間に写像して人間の直観に沿った分類を行っている。

**COM 表現 [30]** 画像内に現れるオブジェクトの重心に着目した研究であり、Center Of Mass という表現方法を提案している。実際には心臓の断面写真を対象として実験を行っている。

**Photobook [31]** 顔、テキスチャ、工具などの画像を対象としたシステムで、固有ベクトルを利用して画像から代表的な特徴を抽出して検索に用いている。

この他の研究としては、心臓の断面画像を扱ったもの [32]、衛星雲画像の類似検索 [33]、略画作成による検索インタフェースについての議論 [34] などがある。

以上述べてきた研究に共通して言えることは、どれも対象の大局的な類似度のみを扱っているので、ユーザの視点が入り込む余地がないということである。個人識別を目的とした指紋検索のようなアプリケーションにおいては、特にユーザの視点を考慮する必要はないわけだが、多様な情報処理活動を目的とした場合、もっと様々な角度からの検索が行えることが望ましい。

## 2.3 部分的な類似度に着目した研究

### 2.3.1 どの部分か？

前節のような画像検索では特定のアプリケーションを除いては柔軟性に欠けるとして、画像の部分的な特徴をキーとした検索を行う試みもなされている。

**ART MUSEUM[35, 12]** 対象を特定しないスケッチ検索の代表的な研究である。ユーザは検索したい画像の部分情報をスケッチして例示画を作成する。類似度は、エッジ情報を2値化したビットマップ画像間で測られる。またART MUSEUMでは心理空間を利用した感性検索の機能も提供している。この他に類似するスケッチ検索の研究として岡野ら[36]の研究がある。これは画像をグラフ表現にして複数の特徴量を用いて手書き画像とのマッチングを行うものである。

**2D string[37]** 画像内のオブジェクトの位置関係に着目したアイコン検索の代表的な研究である。検索したい画像の部分情報をアイコンを配置することで例示画を作成する。2D stringとはオブジェクト間の位置関係を記述する表現方法であり、この手法では位置関係に関して柔軟な検索が行えるところを特長としている。この研究が発表された後も何人かの研究者によって研究され、2D stringのマッチングの応用・改良[38, 39, 40, 41]、三次元へ拡張した3D string[42]、2D stringを応用した3次元アイコン検索インタフェース[43]などが提案されている。

但し、これらのほとんどは既に画像内からオブジェクトが抽出されているという前提に立っている。この他、アイコン検索に関しては饗場[44]らの研究があるが、これも原画像からの特徴抽出は全て人手によるものである。

**色ヒストグラムを用いた検索[45]** 画像の部分情報をキーとした検索手法で最も扱いやすい特徴量が色ヒストグラム情報である。この研究ではサンプルの画像内の一部分を長方形の領域指定を行い、その色ヒストグラムを含む画像を検索する。対象としてはdrag storeの商品棚の撮影画像を用いている。

**色対を用いた検索[46, 47]** 単に色特徴を用いるのではなく二つの色の組み合わせを考慮して検索を行う。例示画の指示方法は上記の研究[45]と同様である。但し、この研究では映像を対象としており、主に同一オブジェクトが現れるシーン検索を主体としている。

**胸部 X線画像検索[48]** 胸部のX線画像を対象とした研究であり、着目する箇所を指定できる図形検索機能を持つ。



**動きをキーとした検索 [49]** 映像を対象とした動きをキーに用いた検索手法も研究されている。あるオブジェクトの動きをキーとしてシーンを検索する。他には Ioka[19] らの研究がある。

**プラント図面の検索 [50, 51]** プラント図面データ中から類似する部品構成を捜し出す研究である。検索は土器検索の研究同様グラフ構造を用いているが、部分グラフのマッチングを行うところが大きく異なる。

**ポンチ絵検索 [52, 53]** 人工的に作成されたポンチ絵の画像を対象としたナビゲーション検索の研究である。顔の構成部分の特徴を用いて特徴ベクトルを構成し、類似度はベクトル間の距離で測られる。ユーザは 10 枚単位で表示される画像から、趣向に合った画像を複数枚選択することで所望の画像に近付いていく。

**FMR[54]** モンタージュのように複数のカラー画像（主に風景画）から、着目する複数の箇所をユーザが長方形の領域指定を行い、その各々の領域の色情報が含まれる画像を検索する。この FMR の研究者らは、さらに知識ベースも利用した観点画像 [55] という概念の検索手法も提案している。画像特徴は色を中心に用いている。

さて、以上いくつかの研究例を紹介したが、これらはどれも画像の空間的な部分情報を考慮した検索が行える点で柔軟性が増したと言えるが、しかし扱う特徴はほとんど単一であり、ユーザが検索に利用できる検索条件は限定されている。即ちどのような特徴に着目しているかが反映されない。また、ART MUSEUM のスケッチや 2D string のアイコン配置で作成された例示画には、描画または配置されたそのままの状態がシステムに解析されてしまい、描画や配置に伴うユーザの意図が反映されない。例えば、配置されたアイコンの位置について、その絶対位置が重要なのか、それとも相対位置が重要なのかということはシステムには伝わらない。

### 2.3.2 どのような特徴か？

では、着目する特徴を指示できる研究を紹介しよう。

**山を対象とした検索 [56]** 山が写っている画像を対象例とした画像検索の研究で、ユーザは色、形状、大きさ、位置情報の 4 つを検索条件として利用できる。この場合の位置とは山の位置であり、ユーザが画像内の空間的な着目部分を指示できるという意味ではない。

**ハイパーメディア雅 [57]** NEC において開発されたシステムであり、例示画検索の機能も備えている。機能としては色属性と形状属性の二つの検索条件が用意されている。どちらの特徴についても領域分割画像とエッジ画像の縮小画像がインデックスとして利用されている。但しこれも着目する部分を指定することはできない。

### 2.3.3 どの部分のどのような特徴？

このように、どの部分のどのような特徴に着目しているかというユーザの視点を考慮することは、それほど簡単なことではない。しかし、このようなアプローチが全くないわけではない。Peter

の研究 [58] や Yihong の研究 [59] がこれに該当するが、次に挙げる二つがまさにこの課題に取り組んだ研究であり、本研究の競合する研究といえる。

**PinPoint[60]** Jeffrey らは画像データベースがどうあるべきかを論じ、その一般論や構成論を述べた後、顔画像を対象としたシステム [61] を実装した。このシステムでは画像データは全て顔の正面画像であり、その前提の基に顔の輪郭、髪の色、目、鼻、口の位置・形状などを認識して、その特徴量を検索に用いている。ユーザは「もっと唇の小さい人」というような指示を予め定められたキーワードで行える。つまりユーザは画像の空間的な部分情報と複数の特徴を操作して検索を行える。

さらに Jeffrey らは PinPoint というシステムを開発している。PinPoint ではユーザのスケッチ画像やまたはナビゲーションなどで提示された画像をキーとして検索を行うが、その際にキーワード、色、テキストチャ、形状、位置の属性に関して重みをユーザが調整できるメカニズムを提供している。この調整はスクロールバーによって操作され、これによりユーザの検索要求に柔軟に応えることを目指している。PinPoint の実装例の一つとして自然画像が用いられているが、その他に特定の領域のアプリケーションに対しての応用もある。例えば、医療分野での X 線画像を対象として、出血や血管などの局所的な画像特徴に対する重みをスクロールバーで調整できるようなアプリケーションも作られている。この局所的特徴はユーザの手によって記述してもよいし、対象が特定されているので、画像処理技術を用いて自動的に抽出することもいく分容易であろう。もちろん、上記の顔画像検索にも適用されうる。以下は筆者の推察であるが、この PinPoint の最たる特徴は、対象データから独立したフレームワークを持っており、検索インタフェースに重きを置いているところにあると思われる。そのため、対象画像からの特徴抽出は二の次であり、対象に合った手法を採用し、場合によっては人手を介してもよいという方針であると推察される。この考え方は本研究と共通するところでもある。

**QBIC[62]** QBIC は IBM によって現在開発研究プロジェクトとして推進されている画像データベースシステムである。一般の自然画像を対象として、色、形状、テキストチャ、位置情報の特徴を検索条件として扱える。さらに着目領域を多角形領域で指定できるなど全体として完成度の高いシステムに仕上がっている。また検索条件もスクロールバーによって詳細に指示できる検索インタフェース [63] を提供し、さらに複数の特徴を扱うので、その検索効率を上げる方法 [64] も提案されている。画像の特徴は事前にキーとなりえる領域を予め人手により抽出し、そのための支援ツールも用意されている。

## 2.4 本研究の方針

以上、紹介してきたように、ユーザの着目する部分及び特徴を反映した検索に関する研究はそれほど多くなく、本論文をまとめている現在では、PinPoint を含む Jeffrey の研究と QBIC が代表的な研究例であるといえる。このような背景を基にして、本研究では後述する画像検索システ

ムの構築を試みた次第である。即ち、ユーザが着目する空間的な部分情報とその特徴をグラフィカルに明示的に指示でき、検索にその指示された情報が反映され、従来とは異なる幅広い検索を可能とする画像検索の実現を目指すわけである。

PinPoint と QBIC もこれに近い構想で構築されているシステムであるが、これらとの比較は、議論をわかりやすくするために、VPoint の構成や機能及び採用している技術を述べた後、6章の評価で述べることにする。

## 第3章

# 画像検索システム VPoint

## 3.1 はじめに

本章では、提案する画像検索プロトタイプシステム VPoint の構想、構成、検索処理手法について述べる。

## 3.2 構想

### 3.2.1 検索インタフェース

ユーザの視点を反映させるということが重要であるので、最初にシステムとユーザとの接点である検索インタフェースから考える。本研究で扱うのは類似画像検索であるので、とりあえず入力する検索の問い合わせ情報は、キーワードなどのテキスト形式ではなく、従来の研究同様にスケッチ検索またはナビゲーション検索を主体とした例示画の形式を用いる。

まずユーザの視点を検索に反映させるためには、二つの方針が考えられる。一つはシステムがユーザの操作やその履歴を観察して類推するものであり、もう一つはユーザが明示的にシステムに指示するものである。ユーザインタフェースの研究では前者についての研究が盛んに行われているが [65, 66, 67, 68, 69], これはユーザの手間が省ける反面、誤認識の可能性も高くなる。逆に後者は確実ではあるがユーザの操作が煩雑になる恐れがある。この点に関して、我々は以前、作図支援システム DOGS[6] を構築したが、その際の議論では、やはり後者の方針がより現実的であるということから、ユーザの描画の意図を明示的にシステムに指示する方針を採った。DOGSでは概念図を中心とした作図を対象として図形間の関係に着目し、それを明示的に指示することで図形間に制約を入力し、編集作業の簡便さを図っている。

そこで本研究でも同様に、ユーザが着目箇所を柔軟に指示でき、かつ明示的に着目している特徴を指示できるような検索インタフェースを用意する。この際、描画または表示された例示画をビットマップ情報として扱うのではなく、作図ツールのように図形単位で操作・編集できればより柔軟な検索が可能になる。具体的にはスケッチ検索の場合、ユーザは複数の図形（以下オブジェクトと呼ぶ）を描画し、それらの属性及び関係についての検索要求を明示的に指示するわけである。

もちろん複数の検索要求を受容しうる検索インタフェースというのは他の研究でも開発されているが、オブジェクト単位で検索要求を指示できること、オブジェクト間の関係についても指示できる点が従来とは異なる特長といえる。

### 3.2.2 検索条件の表現

次に属性と関係についての検索要求の表現方法について述べる。

まずユーザが過去に目にしたことのある画像を検索する状況を想定する。この場合、一般的にユーザの記憶というのは曖昧であり、画像の概略のみを把握しているか断片的な情報しか明確でないことが多い。例えば、色や形状は明確に覚えているが大きさや位置に関しては覚えていない、などである。このような検索要求は、覚えている覚えていないにかかわらず、色だけに着目した検索を行いたい場合、画像内のある特定部分の形状だけに着目した検索が要求される場合など、様々

な状況で考えられることである。よって、上記の例での曖昧さや明確さというのは、状況によっては重要さや制約の強さとも解釈されうる。

そこで本研究では、これらをゆらぎという概念でまとめて扱い、ユーザの視点や着目点を検索に反映させる。上述の例では、ゆらぎはユーザの曖昧度または明確度に相当し、色と形状に関しては曖昧度は小さい、即ちゆらぎは小さいことになる。ユーザはこのゆらぎの値を指示することで検索の幅を狭めたり広げたりするわけである。しかしユーザが着目する特徴を全て予想して用意することは困難であるので、ここでは予めシステムが用意した特徴に限定する。

2章でみたように、QBIC や PinPoint のように複数の特徴や検索条件を扱う研究例は他にもあり、それらにおいても個々の検索条件の強弱を調整できる。その場合、問い合わせ情報とデータ画像との類似度は個々の特徴間の類似度により決定され、個々の検索条件の強弱は、その個々の類似度の相対的な重みづけとして表現され、加重平均により計算される場合が多い。

しかし VPoint では、後述するシグネチャを利用して範囲検索に似た方法で検索条件の強弱を操作する。つまり検索条件の強弱は検索範囲の幅の広さに相当し、特徴間の相対的な重みづけを課す方法よりユーザの直観に沿うものと考えている。以上の理由から、重みと呼ぶのではなく、ゆらぎと呼称しているわけである。

### 3.2.3 プリミティブ分解

上述のように検索側ではオブジェクト単位での操作を行うので、データとなる原画像にも、それに沿った処理を施さなければならない。本研究では画像を構造化して扱う立場をとる。具体的には画像を何らかの手法によって複数の領域や図形（以下プリミティブと呼ぶ）に分解する。プリミティブに分解できれば、前述した検索インタフェースとの相性もよいし、ナビゲーション検索に際しても編集や操作が簡便に行える。さらに着目箇所の指示も容易であるし、構造符号化などへの拡張性もある。

もちろん画像の分解処理はそれだけで非常に困難な問題である。現在の画像処理やパターン認識の技術では、任意の対象に共通した満足な手法というのはなく、仮に対象を固定して強い制約や仮定を課したとしても十分な結果を得ることは難しい。しかし、高度な画像情報処理及び画像情報検索を行うには、何らかの方法により画像を構造化することが望まれる。従来の研究の多くは、画像の構造化や情報抽出を人手を介して対話的に行う方針をとったり、扱う対象画像のモデルを固定したりしていたが、本研究では敢えて人手の介入を行わず、またモデルや対象に関する特別な知識を用いずに自動的にプリミティブに分解することを試みる。但し扱う画像のクラスや抽出する特徴によってはそれぞれ目的に合った方法を使い分ける。例えば2値画像とカラー画像とでは自ずから処理が異なる。

## 3.3 構成及び検索の流れ

以上述べてきた構想を基に本プロトタイプシステム VPoint の構成及び検索の流れを図 3.1 に示す。

VPoint では、全ての画像データはプリミティブの集合として扱うことで検索インタフェース及び一連の検索処理に関する部分を対象データから独立させている。よって異なる画像データを扱う場合でも、後述するシグネチャによるインデックスに関する仕様記述の変更と、必要に応じたプリミティブの特徴抽出モジュールを変更するだけでそのまま使用できる構成になっている。また、後述するように、マッチング処理はシグネチャファイルを用いて効率化を図るとともに、ゆらぎを用いてユーザの視点を反映できるように構成されている。

ユーザはまず検索インタフェースを用いて線分または多角形などの図形オブジェクトをプリミティブとしてマウスによって描画する。そして必要に応じて各プリミティブの特徴（属性）に関するゆらぎを指示する。またオブジェクト間の特徴（関係）についても同様の操作を行う。この情報はシステム側に入力され、プリミティブの情報とゆらぎの情報を合わせて、特徴とシグネチャの仕様ファイル（Feature functions, Signature configuration）を参照の上、中間構造（グラフ構造）を経てシグネチャ表現に変換される。

同様な処理はオフライン処理としてデータ画像にも施される。まず、対象に応じた手法により画像をプリミティブに分解し、その情報を特徴とシグネチャの仕様ファイルを参照して中間構造、シグネチャへと変換する。マッチングはこのシグネチャ同士で行われる。

プリミティブに分解されたデータ画像は、ユーザへの表示用のデータとしても用いることが可能であり、それをキーとした検索も可能である。つまり、検索結果である画像を検索インタフェース上にプリミティブの集合として表示させ、ユーザはそれに対してスケッチ検索同様の操作をすることによりナビゲーション検索を行うのである。

次節以降、システム共通部の詳細及び実装について順に述べていく。

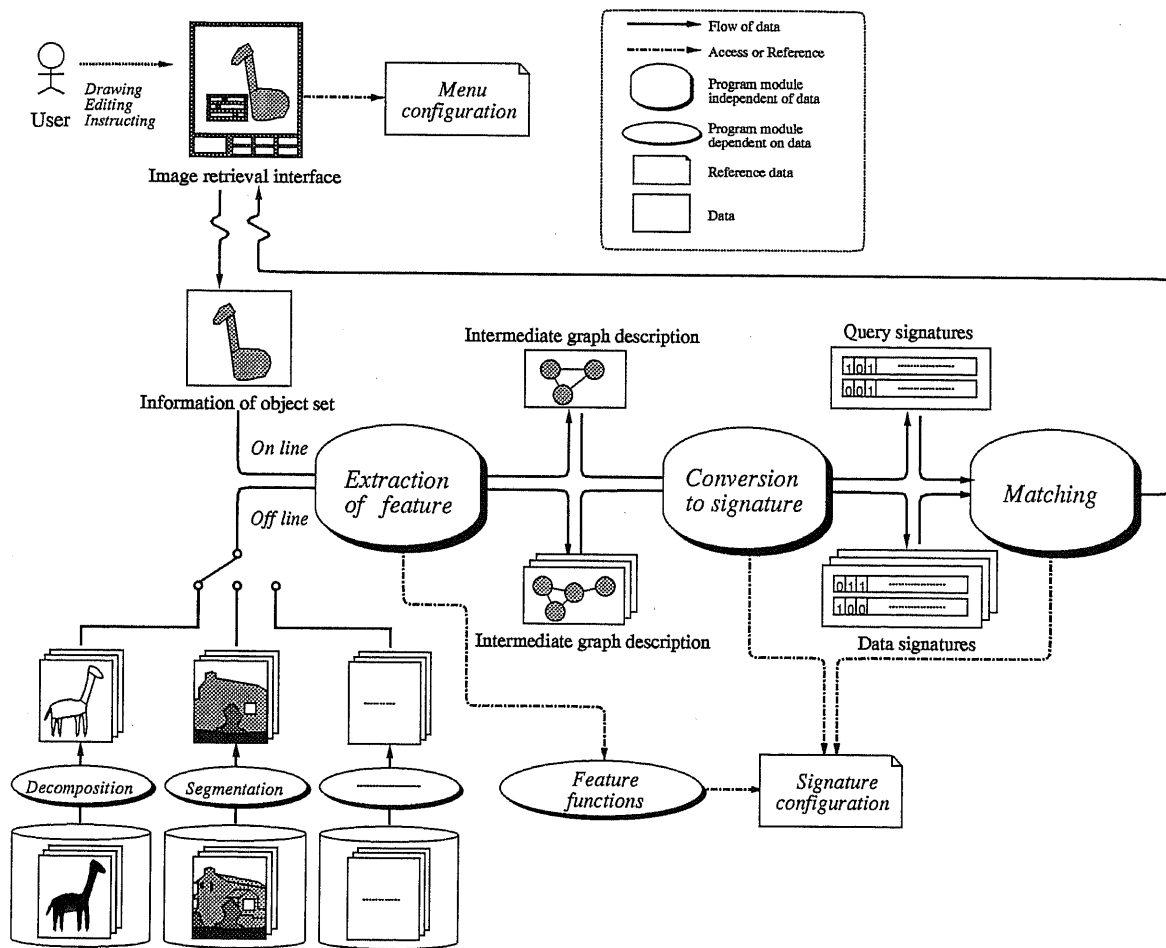


図 3.1 システム構成及び検索の流れ



### 3.4 検索インタフェース

図 3.2 に国旗画像を対象とした VPoint の検索インタフェース及びブラウザを示す。

ユーザは、この検索インタフェースのキャンバス上に多角形図形オブジェクトを描画することで、システムに問い合わせのキーとなる例示画像を与える。そして、各オブジェクトに対して曖昧度や着目度を示すゆらぎを与える。ゆらぎの指示は、各オブジェクトの属性・関係について直接ゆらぎの値をスクロールバーを操作することによって行う。バーにおいて最も左にカーソルを持っていけば、それはゆらぎの値が最小であることを示す。反対に最も右にカーソルが位置する場合はゆらぎの値が最大であることを示す。ここで注意されたいのは、属性または関係の値そのものを指示するのではないということである。例えば、あるプリミティブの大きさの属性のゆらぎの値を 0 にするとは、そのプリミティブの大きさが 0 であることを示すわけではなく、その属性値にかかわらず検索インタフェース上で描かれている状態がユーザにとって明確であるという状態を示す。なお、複数のプリミティブに対する一括指示も個々に対する指示も可能である。

ユーザは、スケッチではなく実際の画像をキーとした検索も行える。スケッチなどで検索した結果はブラウザに表示させることができ、さらにその表示画像を多角形領域の集合として検索インタフェースのキャンバス上にロードすることで、スケッチによる検索同様のオペレーションが可能になる。

検索の判定は後述する類似度をもって行われるが、その判定の基準値即ち類似度の閾値はユーザによって与えられる。

このように VPoint では、ユーザが閾値とゆらぎの値を操作することで検索条件を狭めたり広げたりすることができ、柔軟な検索を提供する。例えば、ある特定の画像をユーザが検索したい状況を想定しよう。ユーザは例示画像を作成し、閾値と属性・関係に関する検索条件を設定することにより問い合わせを行う。もし検索結果の件数が多い場合は、ゆらぎを小さくするか閾値を大きくする。こうして絞り込んだ後、検索結果の画像をブラウジングする。ここで、もし所望の画像がない場合は検索条件を変えてもう一度検索を行う。VPoint の検索の最大のメリットは、多少ユーザの操作が増えるものの、このような試行錯誤による検索が可能であることであり、それだけユーザの意図を反映した検索が可能になる。

さらに、特定の画像の検索だけでなく、様々な情報処理活動に応用できる。例えば、画像や図の分類、特定の対象に限ったデータベース、または画像データベースを構築するためのオーサリングツールのような使い方も可能である。

以下に VPoint の検索インタフェースの機能を簡単に説明しておく。

**描画・編集機能** 折れ線、四角形、多角形、楕円の描画機能を持ち、基本的な作図ツールの描画・編集機能をもつ。

**分解・併合機能** 後述するシルエット画像検索に用いる形状分解手法を適用して、多角形を凸分解する機能を持っている (“Decompose” ボタン)。また、交差または接するオブジェクトを併合して一つのオブジェクトにする機能を持つ (“Merge” ボタン)。これはグルーピング

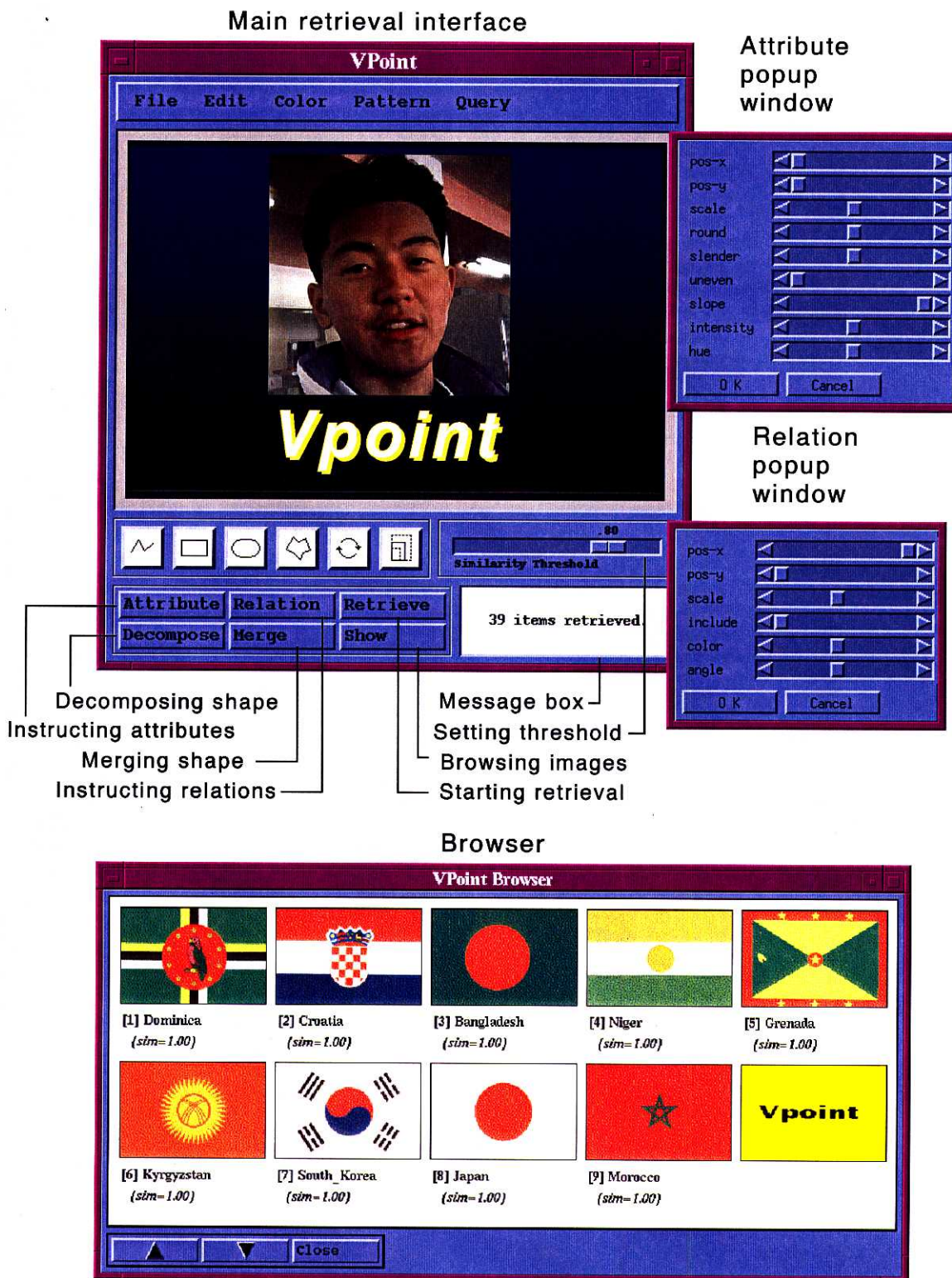


図 3.2 検索インタフェース

操作とは異なり、併合図形の輪郭の情報が残されるので、その情報を検索に利用することができる。

**検索機能** 各オブジェクトの属性とオブジェクト間の関係のゆらぎを指示できる。着目オブジェクトを選択して“Attribute”または“Relation”ボタンを押すとポップアップウィンドウが現れるので、その中のスクロールバーを操作することでゆらぎを指示する。また類似度の閾値も検索インタフェース上に備えられているスクロールバーを操作して設定する。閾値の範囲は0.0から1.0までであり、値が大きいほど類似していることになる。検索は任意のオブジェクトを選択して“Retrieve”ボタンを押すことで実行される。また、検索結果の件数はメッセージボックスに表示され、“Query”メニューの“Log”を選択することで、その検索結果を保存しておくことができる。この保存された情報は、後の検索結果との共通データを抽出するのに用いられる。この操作は“Query”メニューの“And”を選択することで行う。

**ブラウジング機能** 検索結果は“Show”ボタンを押すことで閲覧できる。ブラウザには閲覧用のミニ画像とファイル名及び類似度が表示される。検索結果の画像を基点として次の検索を行いたい場合は、該当する画像をクリックすることでキャンバス上に、その画像がプリミティブの集合として描画されるので、それに対して検索条件を入力すればよい。

## 3.5 検索処理

本節では、具体的な提案する検索処理の機構について述べる。

### 3.5.1 シグネチャ表現

個々のデータ画像を、構造化により得られた各プリミティブを基に検索用の表現に変換する。ここでは連続したビット列であるシグネチャ表現 [71, 72, 73] を用いる。シグネチャはテキストデータベースでよく使用されるものであり、大量のデータをコンパクトな表現で表せること、マッチングがシフト演算及び論理積または論理和を主体とするので、計算機上で扱うのに好都合であることが特長として挙げられる。

各プリミティブはいくつの特徴量（属性値）によって分類され、固定長のプリミティブシグネチャに表現される。各属性値は定められたビット数の範囲内で量子化され、該当する位置にビットが立てられる。各属性は基本的にその値が全て連続値をとるように設定され、画像全体はこのプリミティブシグネチャが連結したもので表現される（属性データシグネチャ）。また、2つのプリミティブ間の関係についてもその関係を表す値（関係値）によって分類され、同様に関係データシグネチャが構成される。ただし関係のプリミティブシグネチャに関しては関係を結ぶ2つのプリミティブの属性のプリミティブシグネチャを先頭に付加する。この際、2つのプリミティブは何らかの基準により、first primitive と second primitive に分けられ、その順序に従って関係のプリミティブシグネチャの前に付加される。

このように対象画像は基本的に一つずつの属性と関係のデータシグネチャにインデックス化さ



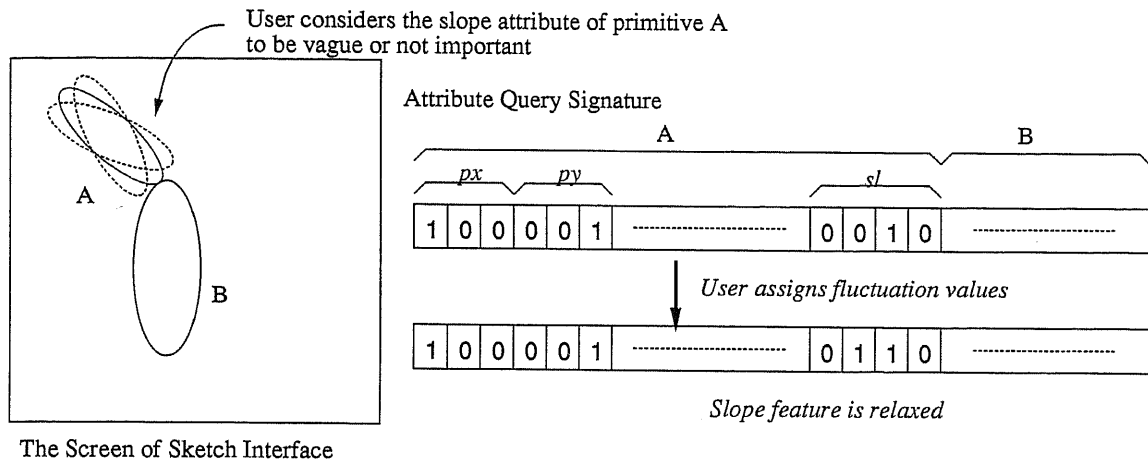


図 3.4 ゆらぎを含む問い合わせシグネチャ

signature method がある。この方法では単純にキーワードのビット列を連結させてドキュメントのシグネチャを作る。上記の例では、D のシグネチャは “1001000000100001” として表される。本手法では、この word signature method を応用した構成をとっているために、superimposed coding に見られる論理和から生じる false drop の恐れはないが、プリミティブの属性や関係の連続値を量子化してビット列に変換しているために、あるカテゴリのプリミティブが一つのビットパターンで表現されてしまう。また属性及び関係共に局所的であるため、実際のデータ上で類似していなくとも、シグネチャ上では類似することがありうる。これらも一種の false drop である。

### 3.5.2 問い合わせシグネチャ

マッチングは個々のデータシグネチャと先述した検索インタフェースからユーザによって入力・指示及び解析された問い合わせシグネチャとの間で行われる。データシグネチャは既に述べたように、各属性と関係の特徴の存在をビット列によって表現しているが、問い合わせシグネチャでは、これに加えてゆらぎの情報が含まれる。ユーザが注目しているまたは明確な特徴はゆらぎの値は小さく、注目してないまたは曖昧な特徴はその値が大きく指示されると述べた。この値は 0.0 から 1.0 までの値をとりうる。このゆらぎの値によって各属性・関係の値は幅を持ち、シグネチャ内の立つビットの数も変化する (図 3.4)。

今、ある属性のとりうる値  $a$  の範囲が  $A_{min} \leq a \leq A_{max}$  として、ゆらぎの値が  $\sigma$  ( $0.0 \leq \sigma \leq 1.0$ ) であるとすると、属性値  $a$  は以下の式により最小値と最大値が与えられる。

$$\begin{aligned}
 a_{min} &= a - (A_{max} - A_{min}) \times \sigma / 2.0 \\
 a_{max} &= a + (A_{max} - A_{min}) \times \sigma / 2.0
 \end{aligned}
 \tag{3.1}$$

即ち、該当するシグネチャは  $a_{min}$  に該当するビットと  $a_{max}$  に該当するビットの間に 1 が立つ

たものになる。但し、ゆらぎの値が1.0の場合は、その特徴に関しては全く制約がないことを示し、全てのビットに1をたてる。

3.5.3 マッチング処理

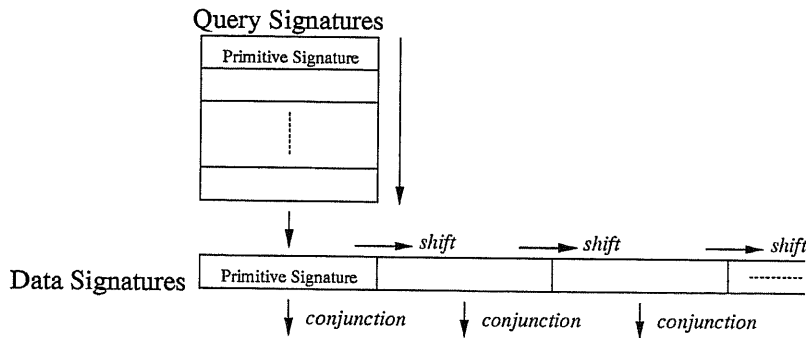


図 3.5 マッチング処理 (1)～プリミティブの対応～

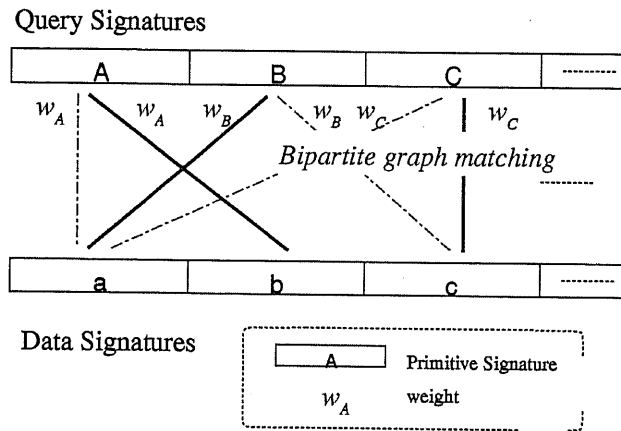


図 3.6 マッチング処理 (2)～2部グラフマッチング～

マッチングは二段階の処理によって行われる。

第一段階は図 3.5 で示されるように、極めて単純なオペレーションで行われる。まず個々の問い合わせシグネチャとデータシグネチャとの間でシフト演算と論理積演算が行われる。もし問い合わせシグネチャ内のプリミティブシグネチャとデータシグネチャ内のプリミティブシグネチャとの論理積の結果が、そのデータシグネチャのプリミティブシグネチャと一致すれば、そのデータのプリミティブは検索要求に合致する可能性があることになる。しかし、このような可能性を持つプリミティブの対応の組み合わせは複数存在する。よってひとまず、このシフトと論理積の演算により検索要求を満足させる可能性のデータシグネチャのプリミティブをすべて見つける。これにより求められた対応関係の様子を示したのが図 3.6 である。

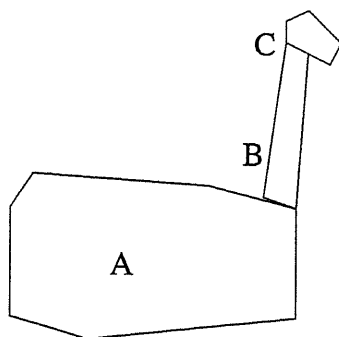


図 3.7 重みの必要性

第二段階の処理は、この可能性ある対応関係の中から最適の対応関係を求めることである。最適というのは対応関係が最も多くなるような組み合わせのことをここでは指しているが、当然、これがユーザの検索要求に対して最適であるとは限らない。しかし、対応関係が最も多くなるようにしておけば、少なくとも検索洩れを少なくできる。例えば図 3.6 の例では、問い合わせ側のプリミティブシグネチャ A, B, C それぞれが、データ側のプリミティブシグネチャ a, b, c のうちのいくつかに対応する可能性があるが、一つの問い合わせのプリミティブにはデータ画像の一つのプリミティブにのみ対応するので、A と a を対応させると B または C に対応するデータ側のプリミティブが存在しなくなる。よって、ここでは図の実線の組み合わせが最適となる。

明らかに、この処理は 2 部グラフのマッチング問題に帰着する。2 部グラフのマッチング問題は既にいくつかの解法が提案されており、最大フロー問題に置き換えるラベリング法 [75]、交互道を辿って増大道を探す Hungary 法 [76]、また同じ Hungary が考えた最小コスト問題への置き換えによる方法 [77] などがある。ここでは増大道を探すハンガリーの手法を用いる。具体的なアルゴリズムは上に示した参考文献を参照されたい。

このように類似度は、問い合わせ側のプリミティブシグネチャと、データ側のプリミティブシグネチャとの対応数によって表現されるわけだが、これでは不十分な点がある。

今までは各プリミティブに対する曖昧度を考慮してきたが、プリミティブ間の重要度は全て同じであった。つまり図 3.7 のような場合、プリミティブ A がマッチングしても C がマッチングしても類似度は同じになる。これは明らかに検索の精度を悪くする一因になる。

しかしユーザの視点に立った場合、A と C のどちらが重要かはこれも場合に依存し、一意には決められない。といって、これまでもユーザに指示させるのは操作の煩雑さを招く。よってここでは単純に大きさに比例した重みを与えることにする。即ち、問い合わせ側の属性に関する各プリミティブの重みを以下の式で定義する。

$$w_k^A = A_k / \sum_{i=1}^N A_i \quad (3.2)$$

$A_i$  はプリミティブ  $i$  の面積であり、 $w_i^A$  はその重みを表す。  $N$  は問い合わせ側のプリミティブ

の数である。

関係については、関係を結ぶ二つのプリミティブの重みの和（または大きさの和）をとり、全ての関係の重みの総和が 1.0 になるように正規化して用いる。次式で表される。

$$w_{k,l}^R = (w_k^A + w_l^A) / \sum_{(P_i, P_j) \in R} (w_i^A + w_j^A) \quad (3.3)$$

$w_{i,j}^R$  はプリミティブ  $i$  と  $j$  が結ぶ関係の重みであり、 $(P_i, P_j) \in R$  は関係が存在するプリミティブの組を示す。

重みをつけた場合、先述の最適な組み合わせを求めることは、2部グラフの重み付き最大マッチング問題を解くことになる。これは最小コスト問題に置き換えることで解ける。しかし、この場合は問い合わせ側の一つのプリミティブシグネチャから出る対応関係を示すエッジの重みは全て同じ値を持つので（図 3.6）増大道を見つける方法で十分である。ちなみに、2部グラフの最大マッチングは、グラフが密の時  $O(V^3)$ 、疎の時には  $O(V(E+V)\log V)$  のコストがかかる。

最終的には類似度は以下の式によって定義される。

$$C_a = \sum_{i \in M} w_i^A \quad (3.4)$$

$$C_r = \sum_{(i,j) \in M} w_{i,j}^R \quad (3.5)$$

$$\text{Similarity} = \frac{C_a N_a + C_r N_r}{N_a + N_r} \quad (3.6)$$

$M$  はマッチングに成功したプリミティブの集合であり、 $N_a$  及び  $N_r$  は問い合わせ属性シグネチャ及び関係シグネチャ中のプリミティブシグネチャの数であり、 $C_a$  と  $C_r$  はそれぞれ属性及び関係シグネチャに関して2部グラフマッチングによって算出された類似度である。

### 3.6 実装

最後に実装について補足しておく。現在、国旗画像を対象とした場合の VPoint はプリミティブ分解のモジュールを含めて、Sun SparcStation 上にモチーフインタフェースを用いて C++ 言語約 30,000 行で実装されている。中間構造は今後の拡張性も考えて、プリミティブの属性をノード、関係をアークとしたグラフ構造になっている。

実際のデータとしては、2値のシルエット画像及び簡易なカラーパターン画像である国旗画像、そしてカラーの自然画像を用いた。いずれも画像内に撮影または描かれている対象に関する知識やモデルなどは用いていない。以降の4章、5章において、それぞれについての要素技術及び実装、検索例について述べる。



## 第4章

# シルエット画像検索

## 4.1 はじめに

さて、前章で述べたシステム VPoint を実際のデータを用いて実装した。まず 2 値のシルエット画像を対象にした。但し、特に対象画像の知識やモデルなどは用いていない。

シルエット画像はその輪郭形状のみが主たる特徴になるわけだが、例えばフーリエ記述子 [78, 79] を用いたマッチングでは形状の大域的な特徴しか扱えず、これでは本システムの趣旨にそぐわない。一つの形状にも様々な構造的な情報が含まれており、視点によって類似度は異なるはずである。例えば図 4.1 では、左の図形が右側の図形のどれに最も類似しているかは着目する部分によって異なる。傾いた長方形に着目するのか、水平方向の長方形が飛び出していることに着目するのか、それは状況に依存する。

そこで、このような形状特徴を多様に扱おうるように形状の構造化を行う。ここでいう構造化とは、形状をいくつかの part に分解することであるとする (図 4.2)。即ち part がプリミティブに相当し、各々の part がシグネチャに変換されるわけである。

本章では、このシルエット画像を対象とした場合の形状の分解手法について新たに提案し、それに基づいた特徴量の抽出、シグネチャの実装について述べ、検索例を示す。

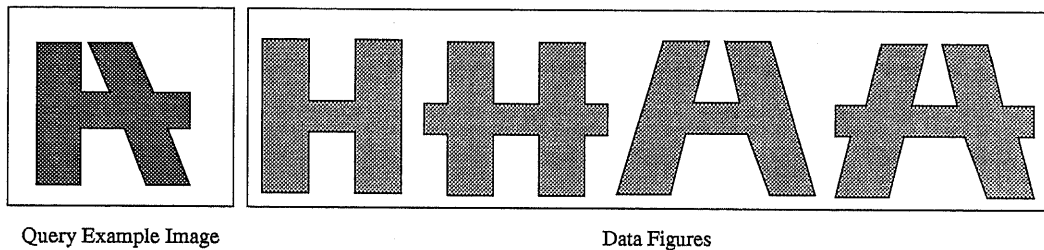


図 4.1 どれが類似しているか?

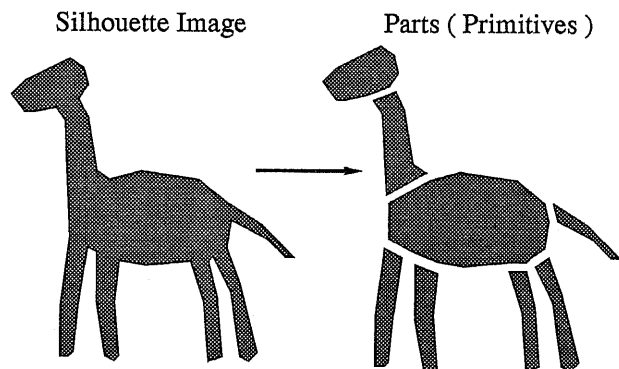


図 4.2 形状分解

## 4.2 形状分解手法

シルエット画像（図形）の形状分解は、古くから研究されている画像処理分野の一つのテーマであり [80, 81], 頂点のクラスタリングによる手法 [82, 83], ボロノイ図を用いた方法 [84], 輪郭線の曲率（凹凸）に着目する方法 [85, 86, 87, 88, 89], モルフォロジー [90] や膨張法を用いて処理する方法 [91, 92, 93] などの手法が提案されてきた。VPoint では、検索インタフェースの機能の一つとしてもこの分割処理を利用したいので、ある程度高速で、かつ、できるだけ人間の直観に合った結果が得られる手法が望まれる。クラスタリングによる手法やボロノイ図を用いた方法は出力結果の可用性に乏しく計算時間にも問題があり、モルフォロジーや膨張法による手法は分割結果である各領域の形状が固定的であり拡張性に欠ける。一方、曲率に着目した方法は、雑音に敏感であるという短所はあるが、上記のような問題はない。ここでは曲率に着目した方針をとる。しかし、従来の手法では完全に凸の領域に分解（凸分解）できないものや、単に曲率の変化だけに着目しただけのものが多く、必ずしも直観に合った結果が得られてはいない。

曲率に着目した場合、一般に分解はその凹部において生じると考えられる。即ち形状の part は凹部と凹部に挟まれた凸部に相当する（図 4.3）。この考え方は Nevins[86] や Rom[88] の手法にも反映されており一見もっともらしく見えるが、この観点に基づいて分解を行うと例えば図 4.4 のような結果が得られる。(a) と (b) は異なる構造に知覚できるにもかかわらず同じ構造に分解されてしまっている。また、(c) についても満足な分解は行われていない。(d) の結果からは、一組の凹凸のシーケンスからなる図形はそれ以上分解されえないことがわかる。

そこで、図 4.5 に示すようなモデルに従って分解を行うことを提案する。この手法の基本的な考え方は、輪郭の凹部に注目して、対象図形はそこから内部に浸食が生じて分割されるというものである。これは単に輪郭の凹凸だけに着目するのではなく、形状の大きさや厚みの全体的な特徴が考慮されるので、より直観に即した結果が期待できる。

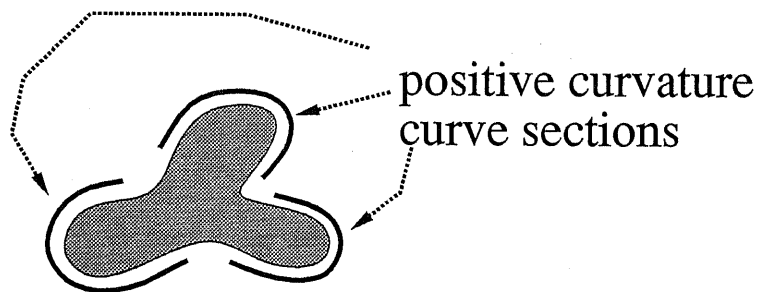


図 4.3 凹凸による part の判定

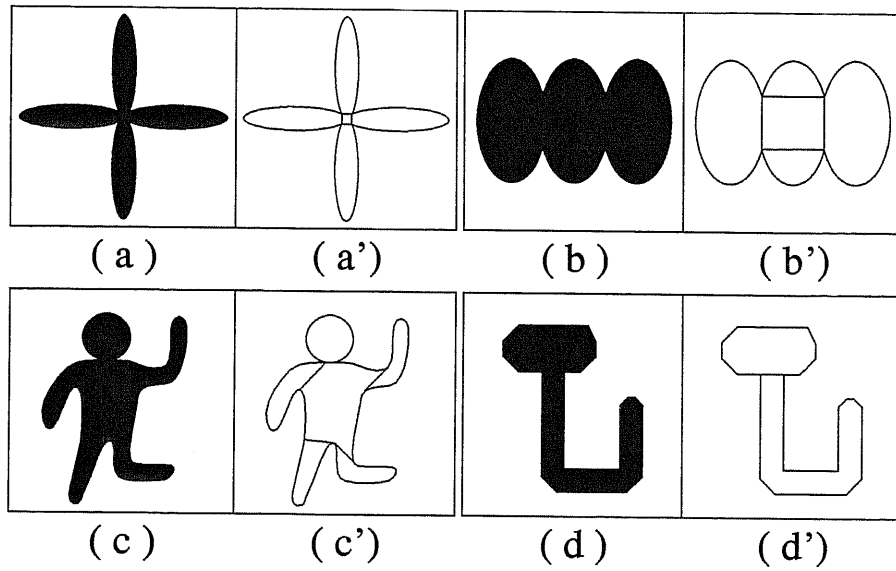


図 4.4 凹部に挟まれた凸部を part とする分解例

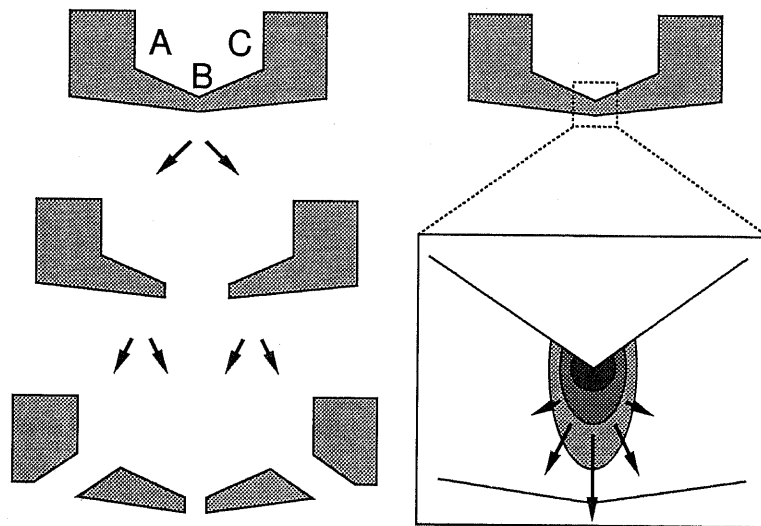


図 4.5 形状分解のモデル

## 4.2.1 分解アルゴリズムの概要

形状分解アルゴリズムの概要を図 4.6 に示す。なお、このアルゴリズムは閉じた 2 次元の交差のない線図形（多角形または領域）を対象としている。よって分割結果も交差のない多角形として出力される。

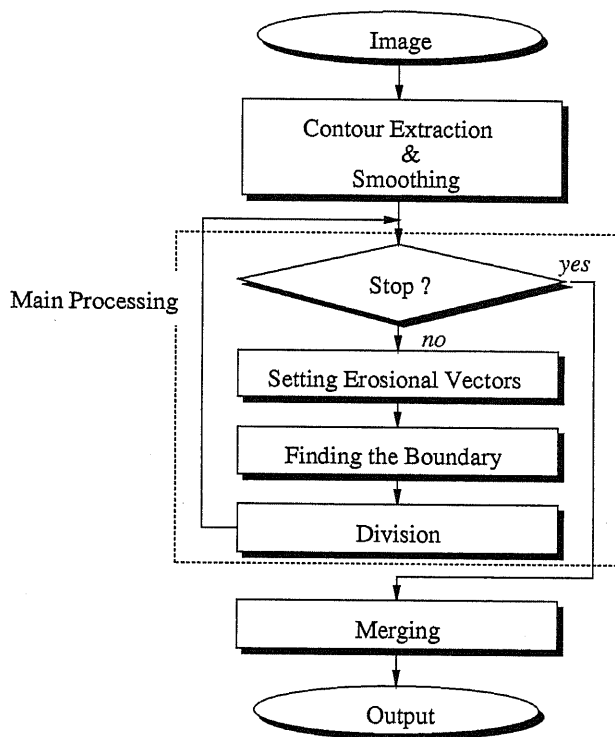


図 4.6 分割のフローチャート

1. 前処理（輪郭線抽出，雑音除去及び多角形近似）。
2. 分割操作が施されていない全ての領域（頂点列）に対して以下の操作を行う。
  - (a) 各頂点の曲率を求める。
  - (b) 各頂点に対して浸食を表現するベクトル  $\vec{v}$  を求める。
  - (c)  $|\vec{v}|$  の極大値を持つ頂点を起点とした分割線を 1 本決定する。
  - (d) 領域を 2 分割する。
3. 分割により新たに生じた領域がある場合は 2 の操作を繰り返す。ない場合は次へ。
4. 過分割領域を併合する。

以下、各手順について説明する。

## 4.2.2 前処理

対象の 2 値のシルエット画像から頂点追跡により輪郭頂点列を抽出し、それを以下のフーリエ記述子表現を用いて高周波成分を遮断することにより平滑化を行う。

$$v(l) = \sum_{k=-K}^K a_k \exp\{2\pi k l j / L\}$$

$$a_k = \frac{1}{L} \int_0^L u(l) \exp\{-2\pi k l j / L\} dl$$

$$u(l) = x(l) + jy(l) \quad (4.1)$$

ここで  $u(l)$  はある輪郭上の固定点から距離  $l$  だけ輪郭に沿った点であり、 $v(l)$  は周波数係数  $K$  による再構成頂点列である。  $L$  は輪郭の周長である。この  $v(l)$  を多角形近似したものを  $V_i = (X_i, Y_i), (i = 1, 2, \dots, N)$  とする。  $V_i$  と  $V_{i+1}$  ( $V_{N+1} = V_1$ ) は輪郭に関して隣接頂点であり、全ての隣接頂点間の距離は等しい。なお、分割処理前のこの最初の頂点列を輪郭線、分割後、新たに生成された頂点列を分割線と呼び分ける。図 4.7 は前処理の流れを示したものである。

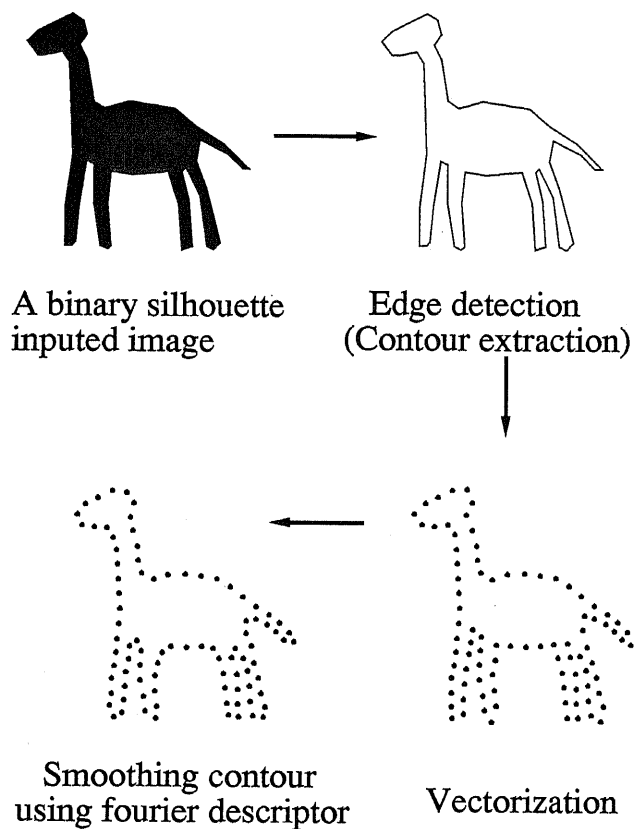


図 4.7 前処理

## 4.2.3 曲率の設定

曲率は  $\overrightarrow{V_{i-1}V_i}$  と  $\overrightarrow{V_iV_{i+1}}$  とのなす角とする（反時計回りを正，図 4.8 参照）．正の曲率（凸）を持つ  $V_i$  に  $C+$ ，0 または負の曲率（凹）を持つ  $V_i$  には  $C-$  のラベル付けをする．

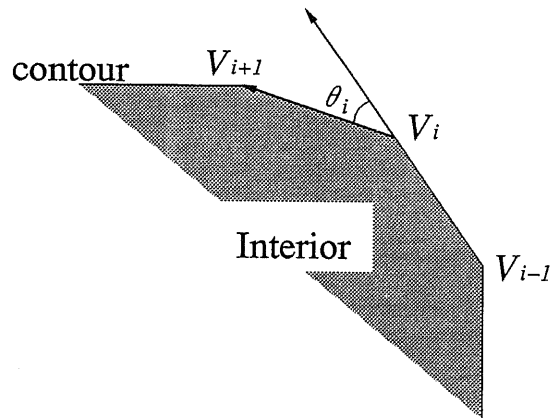


図 4.8 曲率

## 4.2.4 浸食を表現するベクトルの計算

$V_i \in C-$  の各点における輪郭線内部を浸食する力を (4.2) 式で定義されるベクトル  $\vec{v}_i$  として表現する (4.9 参照)．

$$\vec{v}_i = \phi_i^2 d_i \vec{n}_i \quad , \quad \phi_i = \cos^{-1} \frac{\overrightarrow{V_p V_i} \cdot \overrightarrow{V_i V_n}}{|\overrightarrow{V_p V_i}| |\overrightarrow{V_i V_n}|}$$

$$d_i = |\overrightarrow{P V_i}| \quad , \quad \vec{n}_i = \frac{\overrightarrow{P V_i}}{|\overrightarrow{P V_i}|} \quad (4.2)$$

$V_p, V_n$  は， $V_i$  における角を表現する．ここで一般的注意としては，角は一見局所的特徴のように見えるが，実はそうではなく大域的特徴であるということである．よってここでは  $V_i$  における角をその隣接頂点によって測るのではなく，以下のようにして 2 頂点  $V_n, V_p$  を求めて大域的特徴としての角  $\angle V_p V_i V_n$  を定める (図 4.10)． $V_i$  から頂点を調べていき， $V_{i+k}, V_{i-k} (k=1, 2, \dots)$  の少なくとも一方が  $C+$  かつ輪郭線上にあるか， $\cos(\angle V_i V_{i+k} V_{i+k+1}) \geq \epsilon_a$  または  $\cos(\angle V_i V_{i-k} V_{i-k-1}) \geq \epsilon_a$  であるならば， $V_{i+k}, V_{i-k}$  を  $V_n, V_p$  とする．

$\vec{n}_i$  は輪郭内部に向かう単位ベクトルであり，求められた角  $\angle V_p V_i V_n$  を 2 等分する．

頂点  $P$  は  $V_p V_n$  と  $\angle V_p V_i V_n$  の 2 等分線との交点であり， $\phi_i$  と  $d_i$  は，式からわかるように  $V_i$  における大域的な曲率とその角の深さを表す． $\phi_i$  を自乗しているのは，角の深さよりもその曲率の大きさの方が分割への寄与が大きいと推定されるからである．

なお， $V_i \in C+$  の  $\vec{v}_i$  は  $\vec{0}$  である．

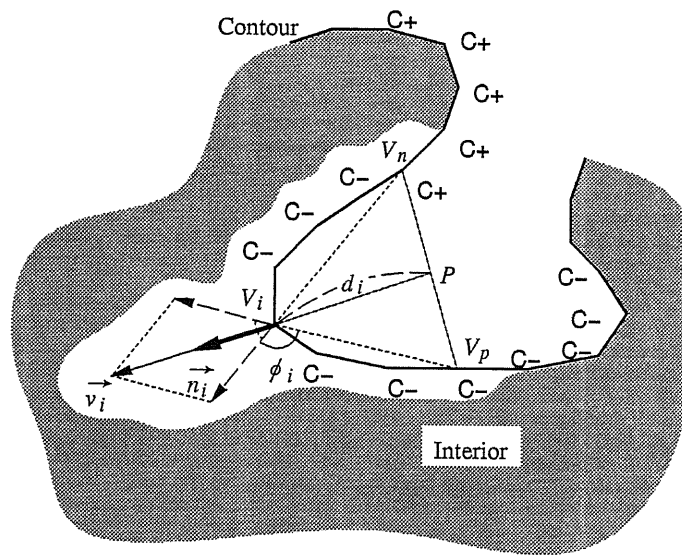


図 4.9 浸食処理

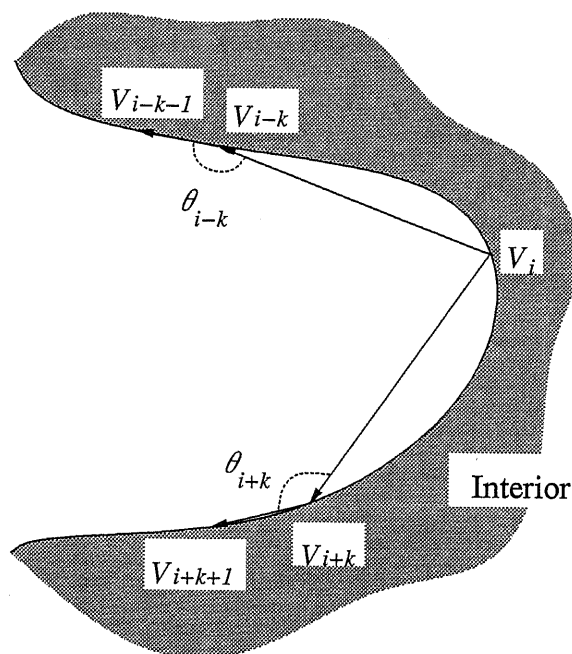


図 4.10 角の決定



## 4.2.5 分割線の決定

以上の処理により求められた  $|\vec{v}_i|$  の結果例を図 4.11 に示す。

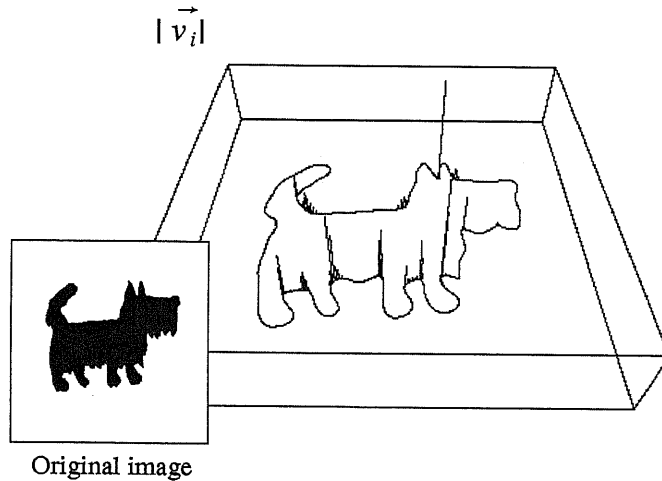


図 4.11 浸食を表現するベクトルの計算結果例

本手法では、全ての凹部から形状内部への浸食が同時に起こり、速く形状を浸食し終える箇所から順に形状を分割していく方針をとっている。この際、角の深い凹部及び曲率の大きい凹部では浸食する速度が速く、また、形状内部で厚みのある部分ほど浸食し終えるのに時間を要すると仮定する。この仮定は、例えば海岸などが削り取られていく自然現象を考慮したものであり、また人間の直観にも沿うことが期待される。

さて、以上の考え方に基づき形状を最初に分割する分割線を求める。 $\vec{v}_i$  は、上記の考え方から凹部における浸食の速度または力を表現するものであるが、全ての  $\vec{v}_i$  についての振舞いを考慮するのは困難なので、その大きさが極大値をとるものを代表として扱う。つまり、 $|v_i| > |v_{i-1}|$ ,  $|v_i| > |v_{i+1}|$  を満たす  $V_i$  に  $Min-$  とラベル付けをし（負の曲率の極小値に相当）、 $V_i \in Min-$  における  $\vec{v}_i$  のみを考慮する。

各  $V_i \in Min-$  について、 $V_i$  から他の点  $V_j$  への浸食し終えるまでの時間  $time$  を (4.3) 式のように求める (図 4.12 参照)。ここで、 $V_i, V_j$  は共に輪郭線上でなければならない。即ち分割線の端点が他の分割線上にあることを許さない。

$$time(V_i, V_j) = \begin{cases} D_{ij}/(|\vec{v}_i| \cos \theta_{ij} + |\vec{v}_j| \cos \theta_{ji}) & \text{for } (|\theta_{ij}| \leq \frac{\pi}{2}, |\theta_{ji}| \leq \frac{\pi}{2}) \\ D_{ij}/(|\vec{v}_i| \cos \theta_{ij}) & \text{for } (|\theta_{ij}| \leq \frac{\pi}{2}, |\theta_{ji}| > \frac{\pi}{2}) \\ \infty & \text{for } (|\theta_{ij}| > \frac{\pi}{2}) \end{cases} \quad (4.3)$$

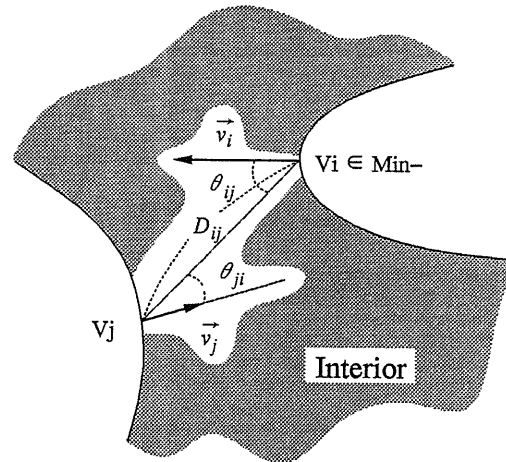


図 4.12 浸食時間の決定

ここで  $D_{ij}$  は  $V_i$  と  $V_j$  の距離であり  $\theta_{ij}$  は  $V_i$  におけるベクトル  $\vec{v}_i$  と  $\overline{V_i V_j}$  のなす角である。分割線は、 $time()$  の最小値を与える  $V_i, V_j$  を結ぶ線分となる。即ち、最も速く浸食し終える箇所から分割を行うわけである。但し、このままでは、どのような場合でも分割線は決定されてしまうので、閾値として  $time()$  の最大値  $t_{max}$  を設けて、 $time(V_i, V_j) > t_{max}$  ならば分割線は設定されず、この領域はこれ以上分割されないとする。

分割線を境界として頂点列を2分割する。線分  $V_i V_j$  の間には各領域のために新たな頂点列を等間隔に補間する。

以上のように、大きな凹部や分割線の長さが短くてすむ箇所から分割が生じるわけである。

#### 4.2.6 併合処理

前節で述べた停止条件だけでは過分割や分割不足が生じ、適切な分割結果は得られない。そこで停止条件である  $t_{max}$  を大き目に設定しておき、一旦過分割を起こしてから併合する方針をとる。併合処理の手順は以下の通りである。まず全ての分割線をその長さに関して降順にソートする。次にソートされた順に各分割線を境界として持つ二つの領域について、併合の可否を検査して、可なら分割線に関して併合する。この処理を併合できなくなるまで繰り返す。

併合の可否は以下のように決定する(図 4.13)。まず対象となる二つの領域を併合した図形の凸包を求め、その凸包と分割線の延長との交点を端点とする線分の長さを  $l$  とする。分割線の長さを  $l_b$  とした時、 $l_b/l$  がある一定閾値  $\epsilon_m$  よりも大きい場合は併合可とする。これは図形の輪郭の凹部は任意の2つの凸多角形が合わさって生じるという事実に基づいている。

この併合処理では大きな領域はほぼ適切に併合されるが、微小領域は併合されない場合がある。多くの場合、これは緩やかに曲率に変化するリボン状の形状に起こりがちである。ここでは、このような微小領域は、ヒューリスティックな知識をもとに隣接領域に併合する。

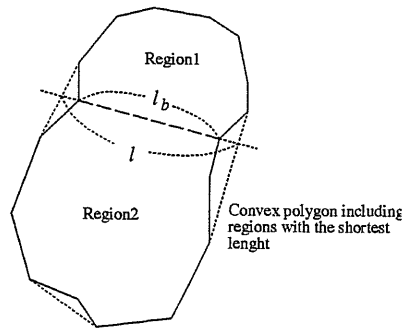


図 4.13 併合処理

### 4.3 分解例

以上のアルゴリズムを用いた分解例を示す。図 4.14 は図 4.4 と同じデータに対して行った結果である。全ての形状は  $K = 40$ ,  $N = 500$ ,  $\epsilon_a = -0.8$ ,  $\epsilon_m = 0.8$ ,  $t_{max} = 100$  の同一の条件の下で分解された結果である。平均処理時間は Sun SparcStation20 上で約 3.3 秒であった。図 4.4 との顕著な差は図 4.14 (b'') (c'') から見てとれる。つまり、この方法が曲率だけでなく全体の形状や厚みを考慮していることがわかる。さらに図 4.4 (d'') からは、この手法が湾曲したりボン状のものも凸に分解してくれることがわかる。

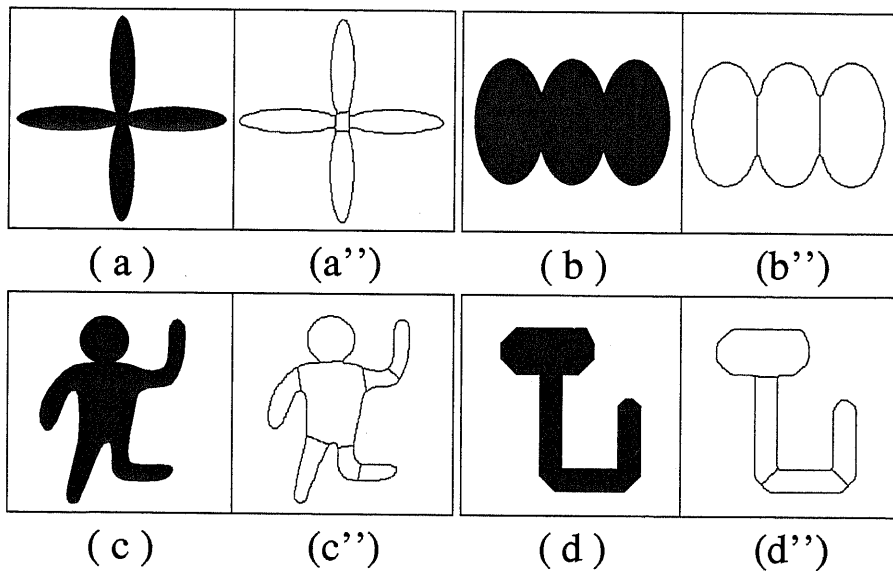


図 4.14 分解例 (1)

次にスキャナから読みとった動物のシルエット画像に対しての分解例を図 4.15 に示す。これらは図 4.14 の例よりも複雑な構造を持ち雑音もかなりののっている。画像のサイズは  $512 \times 512$  であ

り、パラメータの条件は同一である。平均時間は約4.1秒ほどであった。図4.15 (i) や図4.15 (l) のように穴は扱えないが、ほぼ良好な結果が得られている。

さて、最後に計算量の評価をしておく。この手法は  $O(MN)$  の計算量を必要とする。ここで  $M$  は  $Min-$  の数に比例する。 $M$  は形状に依存し、通常  $N$  に比べて小さい値を持つ。この評価の妥当性は図4.16 によって示される。このグラフでは図4.15 の3枚の画像についての結果を示してある。パラメータは  $N$  以外を固定してその処理時間を測った。ほとんど  $O(MN)$  であることがわかる。

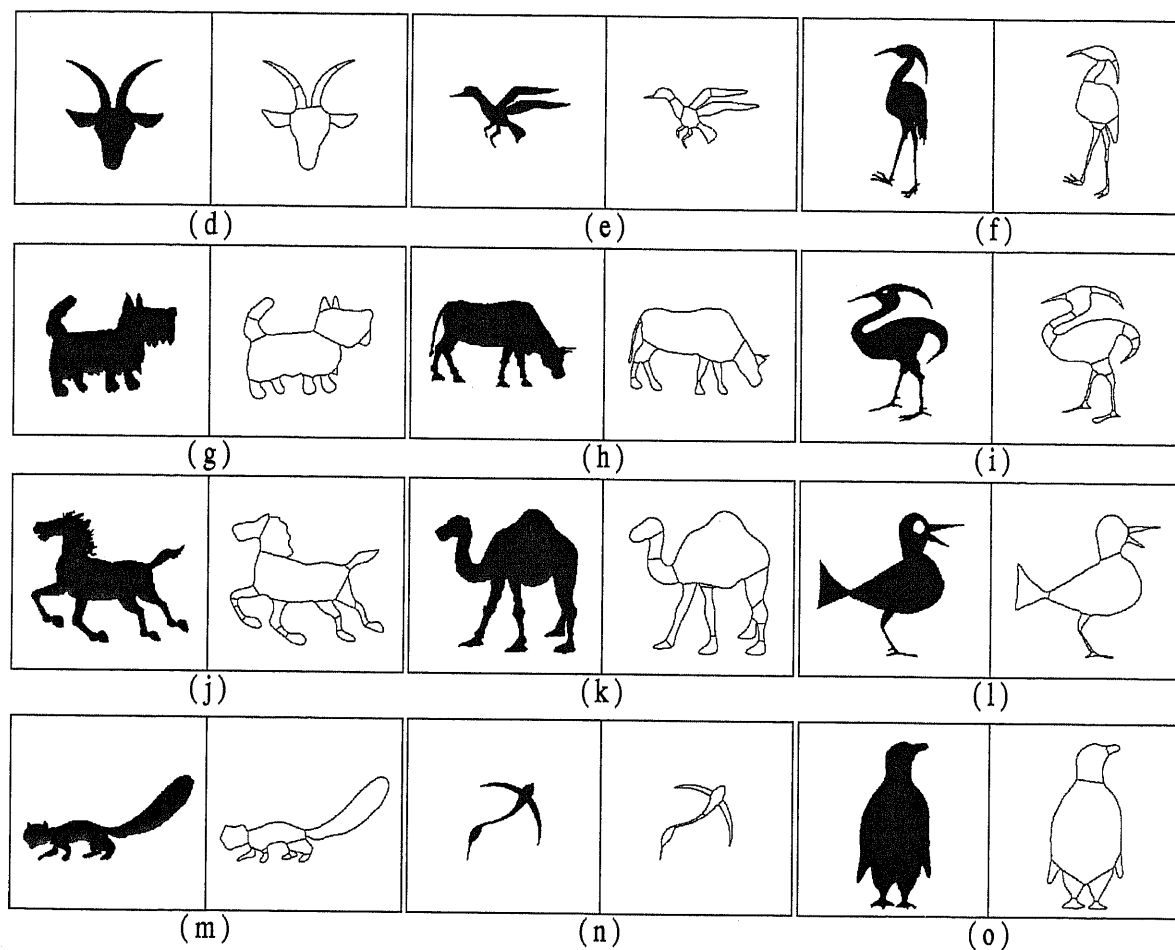


図 4.15 分解例 (2)

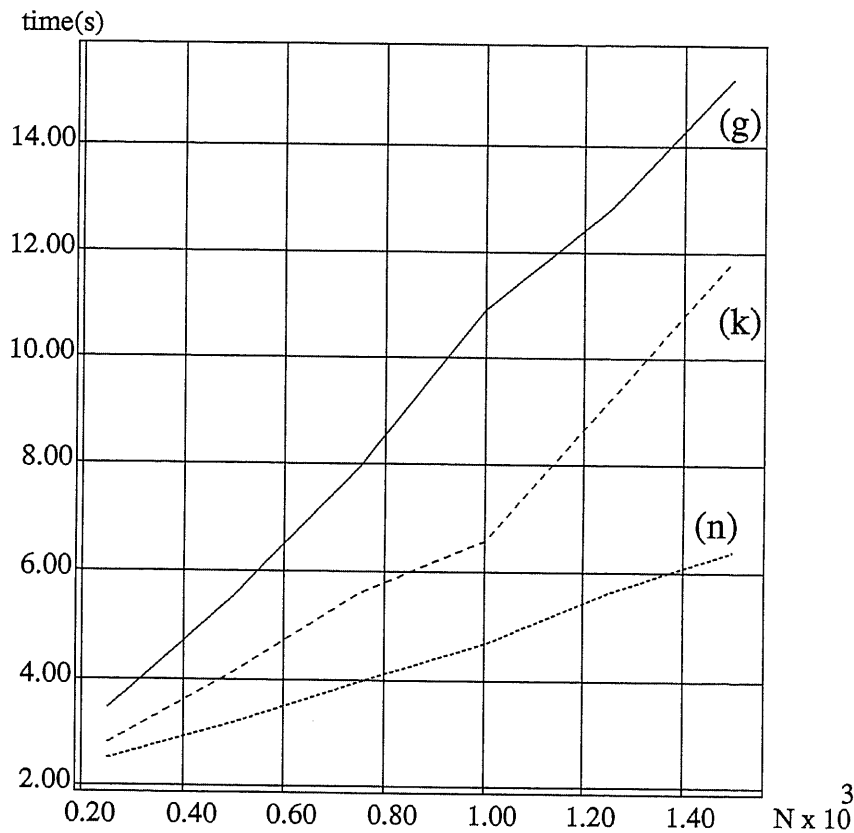


図 4.16 計算量

## 4.4 特徴記述

これまでの処理によりシルエット画像は凸領域の集合に分解された。次に、これらから検索処理に用いる特徴量を抽出する。具体的には、この凸領域と次に述べる準凸領域をプリミティブとして用い、それらの骨格（スケルトン）を主体とした記述を行う。

### 4.4.1 スケルトン抽出

現在までの分解結果である凸領域を純凸領域と呼ぶ。純凸領域のスケルトンは領域の輪郭線上の頂点列の第一主成分（図 4.17 (a)）として求められる。つまりこれはスケルトンというよりは領域の主軸になるわけであり、各領域を楕円に近似表現して記述すると捉えてもよい。但し対象形状を純凸領域で記述するだけでは不十分である。例えば関節で曲がっている動物の脚は二つの領域に分割してもかまわないが、それが動物の尾である場合はむしろ分割するのは好ましくない。この判断には予め対象に関する知識が必要であるが、ここでは特に対象の知識を用いることはせず、特徴となる可能性のある領域は全て扱うことにする。即ち、それが尾であるか脚であるかにかかわらず、分割したものと分割しないもの両方を扱う。

よって次にすべき処理は今までの純凸領域から準凸領域への併合処理である。ここでは主にリボン状の形状領域を準凸領域と定義する。リボン状の領域は分割線の一つまたは二つ持つ領域が連なっていることが多いというヒューリスティックな知識に基づいて併合を行う。まず分割線の一つまたは二つ有する隣接する二つの領域を探し、それらを分割する分割線の両端点を調べる。少なくとも一方の端点における元形状の輪郭線の曲率が一定値以上ならば併合する。その分割線が大きな二つの凹部で生じたものでなければ併合するわけである。

準凸領域のスケルトンは主成分分析では求められないが、分割線が二つ以下の領域は、断面の始点終点が容易に決定できるので、併合時の断面の中点を結んだものとして求めることができる(図 4.17 (b))。

この抽出例を図 4.18 に示す。各々の例の左側は純凸領域のままでスケルトンを抽出した場合で、わかりやすいように楕円近似表現にしてある。右側は準凸領域を含めた場合のスケルトン抽出の例である。

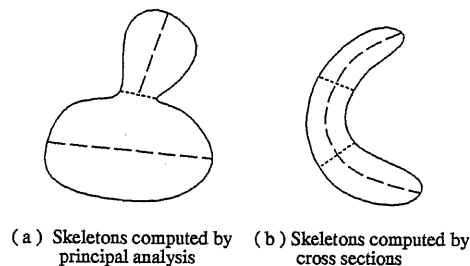


図 4.17 スケルトン抽出

#### 4.4.2 用いる特徴量とシグネチャへの変換

抽出したスケルトンを基にして、各プリミティブ(純・準凸領域)の属性及び関係に関する特徴量を定めて、それを検索用のシグネチャに割り当てる。

用いる属性は以下の7属性である。

##### 属性

1. 水平方向位置：領域の中心の  $x$  座標。
2. 鉛直方向位置：領域の中心の  $y$  座標。
3. 大きさ：領域の面積の平方値。
4. スケルトン長：領域のスケルトンの長さ。
5. 形状：スケルトンの長さに対する領域の平均幅の比。
6. スケルトンの曲率：領域のスケルトンの曲率変化の絶対値の合計。

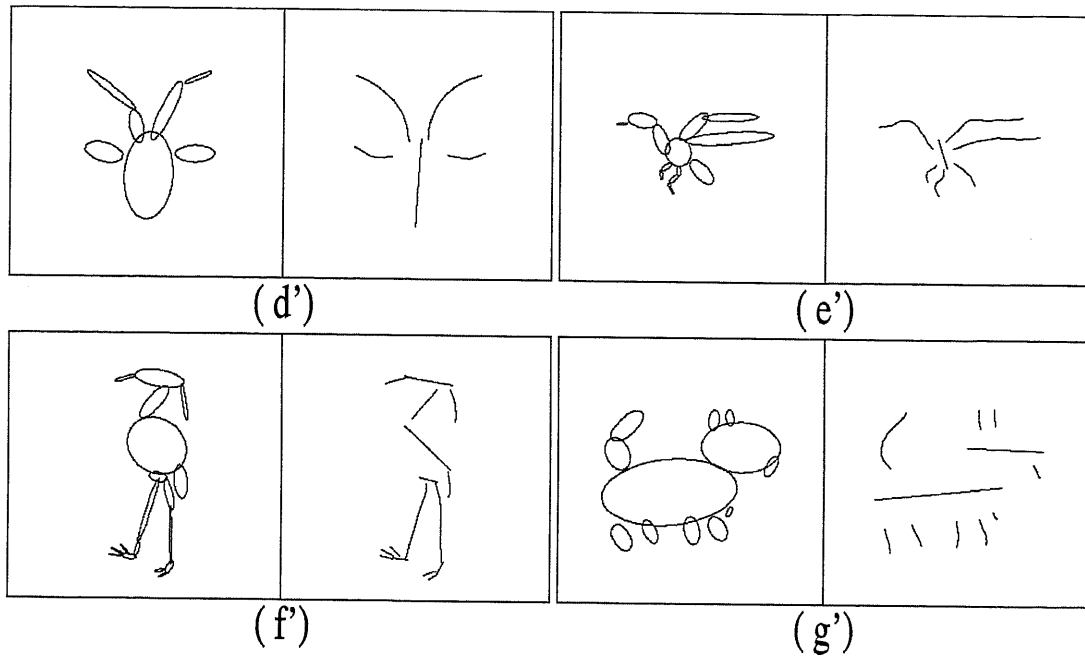


図 4.18 スケルトン抽出例

## 7. 傾き：領域の軸の傾き.

なお全ての属性には一様に 8 bit を割り当てた. 計 7 属性 56 ビットである. 形状属性の値が大きいものは, 円形に近いと考えて傾き属性は設定しない. つまりビット列は全て 0 になる. 逆に問い合わせシグネチャに関しては全て 1 の値が与えられる. 軸というのはプリミティブの輪郭頂点列の第一主成分であり, 純凸領域の場合はスケルトンと軸は同一であるが, 準凸領域の場合は異なることに注意されたい.

## 関係

スケルトンの長いプリミティブを第一プリミティブ (First Primitive), もう一方を第二プリミティブ (Second Primitive) として関係を結ぶ.

1. 相対水平方向位置：第一プリミティブから見た第二プリミティブの x 方向の位置関係.
2. 相対鉛直方向位置：第一プリミティブから見た第二プリミティブの y 方向の位置関係.
3. 相対的な大きさ：二つのプリミティブのスケルトンの長さの比.
4. 交差位置：第一プリミティブから見た二つのスケルトンの交差位置.
5. 角度：二つのプリミティブの軸のなす角.

ここでも各関係 8 bit を割り当てる。関係シグネチャには二つのプリミティブの属性を表すプリミティブシグネチャを付加するので、計 5 関係、152 bit (19 byte) になる。

相対水平方向位置関係は、第一プリミティブの占める x 方向の幅と第二プリミティブの中心の x 座標との位置関係を示す。鉛直方向も同様である。位置関係が水平と鉛直方向に分けられているのは、本システムで扱おうとする関係がスカラー量であるためである。交差位置関係は二つのプリミティブの近接する点の、第一プリミティブのスケルトンに対する位置を表す。交差位置関係と角度関係は軸の傾き属性が設定されていなければならない。第一プリミティブが曖昧なもの、つまりスケルトンの長さが近いプリミティブの組み合わせは、二通りの関係を冗長に記述する。また、ここで全てのプリミティブ間の関係を求めるわけにはいかないため、各プリミティブの面積に比例した量だけプリミティブを膨張させて、交わったプリミティブ同士のみ関係を結ぶ。また包含関係にあるプリミティブ同士は関係を結ばない。

図 4.15 (A) の例では凸領域・準凸領域含めて、プリミティブが 9、関係が 8、属性データシグネチャが 63 byte、関係データシグネチャが 152 byte である。

## 4.5 検索例

以上の分解及び抽出処理のモジュールは、3 章で述べた共通モジュールに結合されている。また、この分解アルゴリズムは検索インタフェース側にも組み込んであり、ユーザが描画した多角形図形に対しても分割を行えるように実装されている。現在扱っているシルエット画像データは 93 枚の  $512 \times 512$  の動物を中心とした 2 値シルエット画像である。これらのデータ画像 1 枚あたりの平均の属性の数は約 17.1、関係は約 76.6 であった。なお、データ画像の分解のパラメータは効率よりも精度を上げるために、先述したパラメータを変えて  $K = 50, N = 1000, \epsilon_a = -0.8, \epsilon_m = 0.8, t_{max} = 200$  の同一の条件の下で行った。

本研究では、このシルエット画像を対象にして実際に VPoint を用いた検索を行ってみた。本節ではそれらの検索例を紹介する。但し、本節では VPoint の特長を示すことを目的とし、これらの検索例に基づいた評価及び考察は、後述する国旗画像検索及び自然画検索と併せて 6 章において述べることにする。

以下に、10 通りの検索例を示す。まず前提として、ユーザは全部の画像を一度見たことがあり(記憶している必要はない)、その上で思いつく検索要求を出すものとする。検索結果はまず検索数が表示され、ある程度絞れたとユーザが判断したら、はじめて実際の検索結果の画像をブラウズするものとする。

**検索例 1** 図 4.19 に検索例を示す。検索要求は「中央付近に胴体があってその下に長い尾がある鼠」である。ユーザは VPoint の描画機能を用いて、楕円と折れ線オブジェクトを描画し、それらの属性と関係のゆらぎを指示する。この場合、描画された状態が全体的に特に曖昧でないため、ゆらぎは全て初期設定の 0.5 のままにした(図 4.19 (a) (b))。閾値は 0.8 とした。検索結果が 4 件であったのでブラウジングしてみると図 4.19 (c) で示される画像が検索された。



次に検索条件を緩めて「とにかく長い尾を持つ動物（鼠）」という検索要求を出した。緩めてよいゆらぎは、位置・大きさ・スケルトン長・傾きの属性とそれに関わる関係である。これらのゆらぎを除々に大きくしていき検索結果の数が増えるまで様子を見る。検索数が5件になった時点でブラウズした結果が図 4.20 である。新たに2番の画像が検索された。この画像は他の4枚に比べると全体的に上方に位置し、胴体と尾の位置関係も上下関係にはない。さらに関係の条件を緩めると、図 4.21 のようにさらに1件検索されたが、これは検索要求と合致しなかったため、検索を終了した。

**検索例 2** 図 4.22 に検索例を示す。検索要求は「胴体と小さな羽根がある鳥（燕）」である。ユーザは多角形オブジェクトを3つ描画し、形状情報のみが重要であることから、向きの属性と位置関係のゆらぎを最大にした。閾値は0.8である。この結果検索された数は30件以上あったので、閾値を0.9に上げたが、なお検索数が多いので、他の属性を少し強め、関係も強めた。しかし、それでも絞り切れないので形状と曲率の条件を強めると4件に絞れた。ブラウズした結果が図 4.22 (c) である。2件の余計な画像が含まれた。これらの画像の分解結果の中に、ちょうどこの検索要求に合致するようなプリミティブの組み合わせが含まれているためである。この他に燕らしきものは図 4.22 (d) に示すように2件あるが、いずれも翼が胴に比べて大きいので検索対象外となった。

**検索例 3** 図 4.23 に検索例を示す。検索要求は「四つ脚で首が長い動物（麒麟）」である。ユーザは多角形オブジェクトを図 4.23 (a) のように描画し、それを“Decompose”ボタンを押すことで形状分解する。その結果が図 4.23 (b) である。次に形状関係の属性を厳しくして、位置、大きさに関する属性と関係全ては初期設定のままにした。検索結果は閾値0.8で1件であった。ブラウズしてみると所望の画像ではなかったので、閾値を0.75を緩めてみたところ6件検索された。ブラウズした結果が図 4.23 (c) である。麒麟は全て含まれているが、この検索要求は大域的な特徴に関する検索であるので6番の画像は正確には検索要求とは合致しない。その意味では5番の画像の方が合致する。ちなみに、位置関係を緩めると2番の画像が最も高い類似度を出した。

**検索例 4** 図 4.24 に検索例を示す。検索意図は「角と耳のある動物の真正面の顔」である。ユーザの操作は検索3と同様である。最初、標準のゆらぎのまま閾値0.8で50件ほど検索されたので、閾値を0.9にして図 4.24 (b) (c) に示される値にまで属性と関係を厳しくしたところ2件まで絞れた。ブラウズの結果は図 4.24 (d) である。2番の画像が検索されたのは、長い角の部分が凸分解されて2つのプリミティブに分解されているためである。なお、この他に動物の顔は図 4.24 (e) の画像があるが、角がないので検索されなかった。

**検索例 5** 図 4.25, 4.26 に検索結果を示す。検索要求は「走っている馬」である。ユーザは一度目にした画像を検索しようとしている。ユーザは今まで同様に描画及び分解を行うが、馬の向きなどが曖昧なので胴体と脚だけに着目した検索を行った。まず図 4.25 (c) のように胴体の属性を少し強め、図 4.25 (d) のように脚の属性のうち、傾きの条件を無視するよう

にし、位置条件も緩めて、形状関連のみ少し強めた。また胴体と脚のオブジェクト間の関係は標準である(図 4.26 (e))。ここで胴体または脚の色が濃く塗られているが、これはわかりやすく図示するために便宜上施したものであり、色自体には何も意味もない。検索結果は図 4.26 (f) (g) に示す。4 件検索できた。所望の画像が 1 番目に検索できている。この他、走っている四つ脚動物としては 3 番や図 4.26 (h) の画像があるが、後者の場合は脚の部分が例示画と類似していない。また、4 番の画像が検索されているが、これは膝の関節の部分で分割されているのと、例示画における脚の傾きの制約がないために、高い類似度を出しているのである。

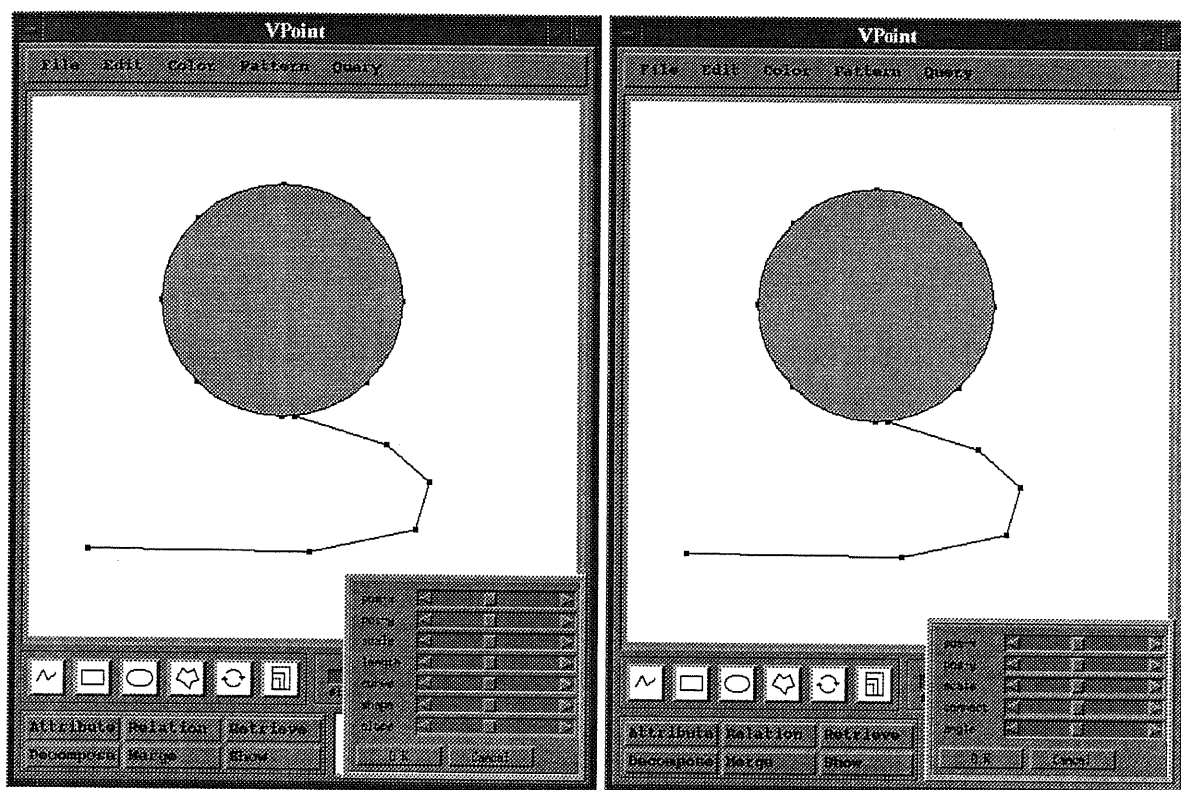
**検索例 6** 図 4.27 に検索例を示す。検索要求は「検索 5 で検索された 4 番の画像と類似する画像(らくだ)」である。つまりナビゲーション検索である。ユーザは、図 4.26 (g) において 4 番の画像をクリックすることで画像をキャンバス上にロードする。この際、ロードされるのは原画像ではなくて、分解処理済みのプリミティブの集合である(図 4.27 (a))。ユーザは、このロードされたオブジェクト群のうち、不要なオブジェクトなどを削除・編集して検索を行う(図 4.27 (b))。属性及び関係共に標準で閾値 0.8 で検索したところ 3 件検索できた。結果は図 4.27 (c) の通りである。こぶ付きのラクダはこの他にはないので、その意味からは検索洩れはない。

**検索例 7** 図 4.28 に検索例を示す。検索 6 の続きである。検索要求はラクダを意識して「四つ脚動物で例示画のような胴体と首を持つ動物(らくだ)」である。今度は問い合わせのオブジェクトから、こぶを除き、位置及び大きさに関する属性の条件を無視するようにして、その他の値を一段階強めた。関係については、首の向きなどがどうでもいいので水平方向の位置関係のゆらぎを緩くした。閾値 0.8 で 7 件検索された(図 4.28 (c))。大きさや向きにかかわらずラクダが検索できている。検索洩れは 8 番目に位置する図 4.29 (d) の画像である(類似度は四捨五入表示してあるので実際の類似度は 0.8 未満である)。この画像は後ろ脚が大きく重なっているために四つ脚の条件に合致せず類似度が低下している。

**検索例 8** 図 4.29 に検索例を示す。検索要求は「にわたりの画像を起点として、ぎざぎざ模様を持つ生き物の画像」である。これもナビゲーション検索である。ゆらぎは標準で閾値 0.9 にして着目部だけを問い合わせの例示画とした。その結果 2 件検索された。検索要求に合致している。この他に同じぎざぎざの形状を持つ画像としては図 4.29 (d) に示す 2 件があるが、これらのぎざぎざ形状は突起が小さいので検索要求と合致しない。

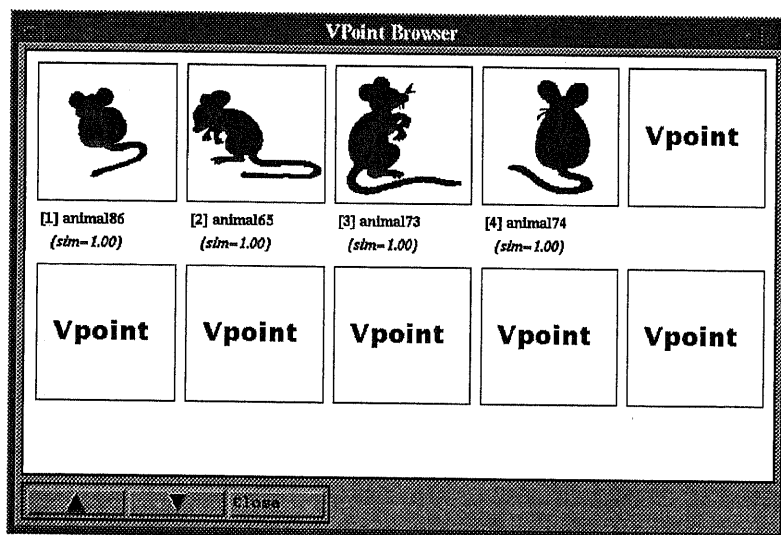
**検索例 9** 図 4.30 に検索例を示す。検索要求はナビゲーション検索で「細長い胴体と短い脚を持つ四つ脚動物」である。ゆらぎは形状と相対的な大きさなどが重要であるので、それらを図 4.30 (a) (b) のように少し強めた。閾値は 0.95 である。この結果 7 件検索された。4, 5, 7 は明らかに余計な検索結果である。また 3 番も四つ脚の動物ではあるが四つ脚の画像ではないので正当な検索結果とは言えない。検索洩れはダックスフンドの画像(図 4.30 (d)) である。原因は図示されているように形状分解が過度に行われたためである。

**検索例 10** 図 4.31 及び図 4.32 に検索例を示す。検索要求は「きじを起点として、細長い二本脚を持つ鳥の画像」である。ゆらぎが標準の場合で検索結果は 5 件であった (図 4.31 (c))。他にもあるかもしれないということで、属性のゆらぎを調整する。位置と傾きは緩めて他は強めた (図 4.32 (c))。閾値 0.8 で 5 件検索されたが、余計なものが入り込み、さらに以前の検索結果の画像が洩れてしまった。そこで閾値を 0.75 に緩めてみると検索結果が 10 件となった (図 4.32 (b) (c))。検索洩れは 1 件で図 4.32 (d) に示される画像であった。原因は脚がわずかに太いために検索されなかった。



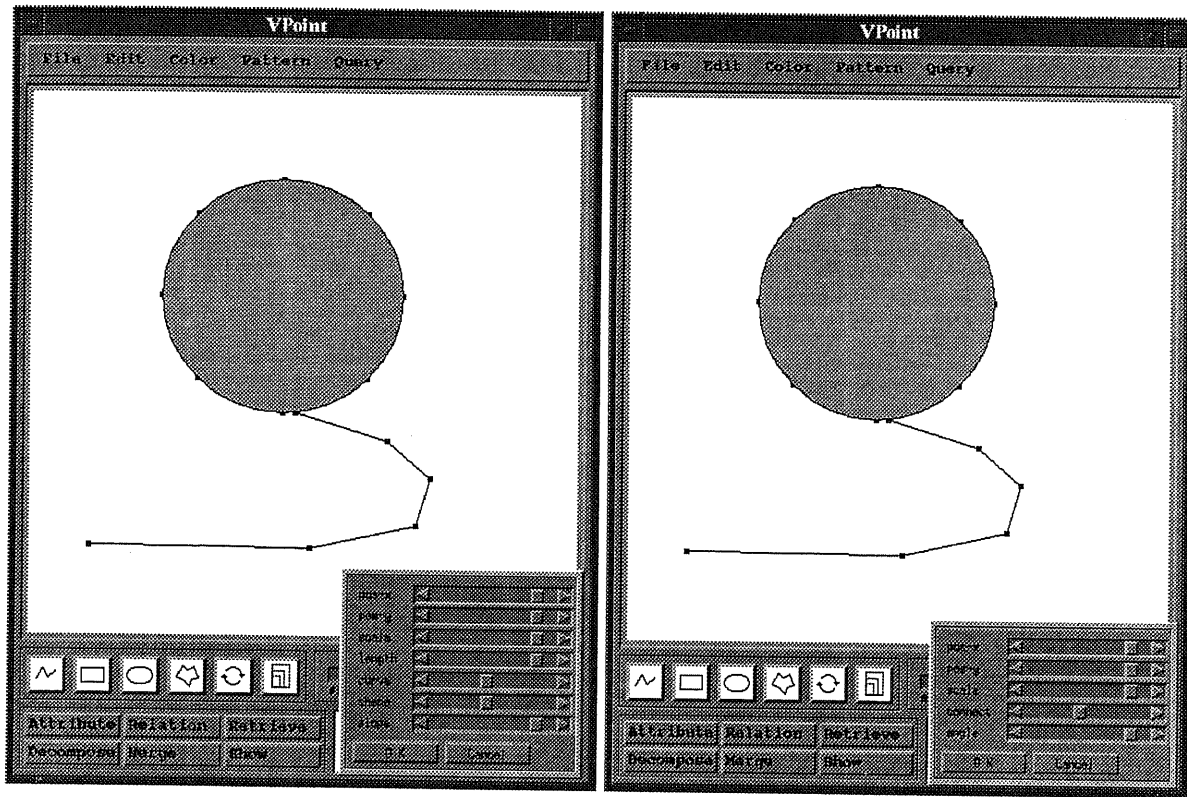
(a)

(b)



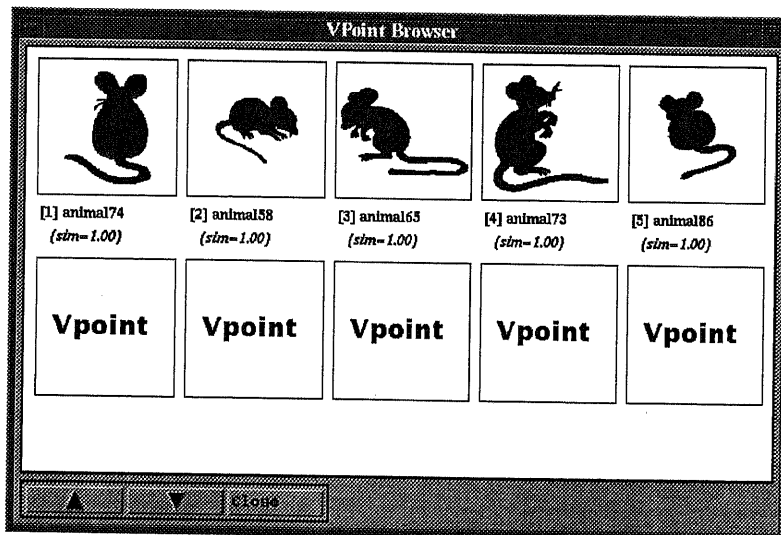
(c)

図 4.19 シルエット画像検索例 (1a)



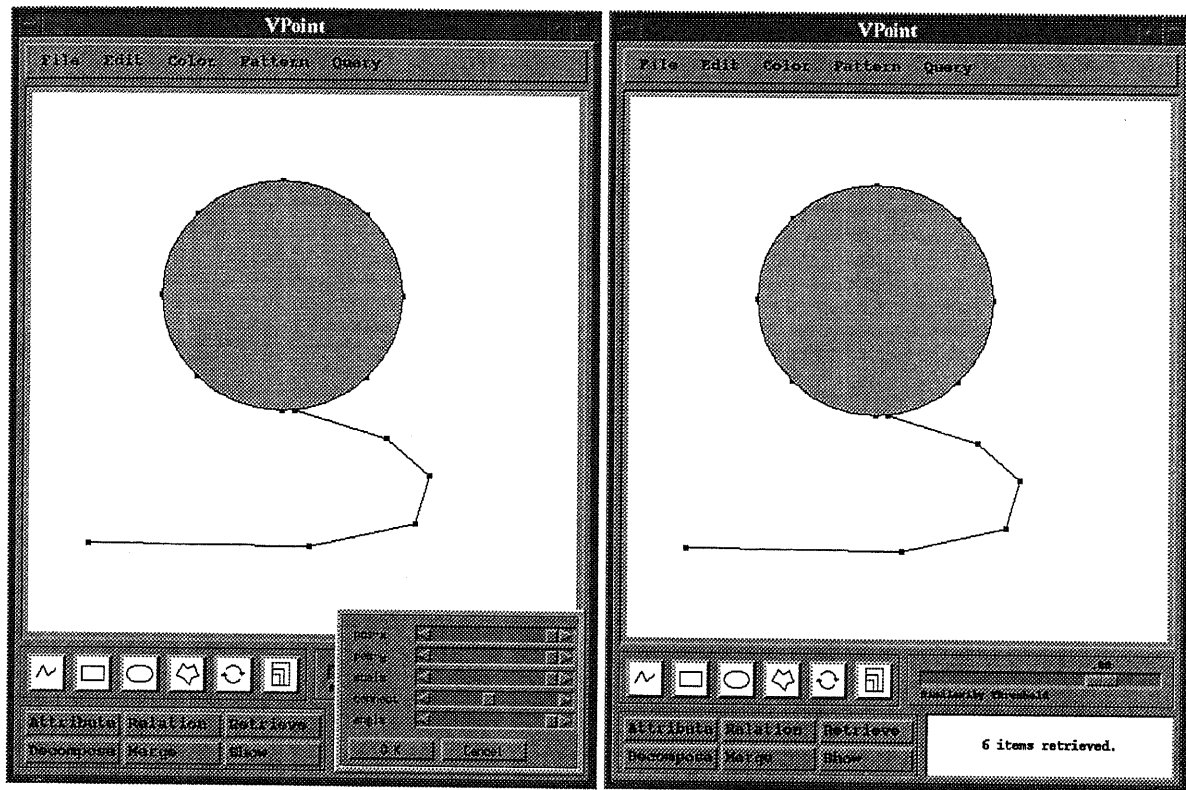
(a)

(b)



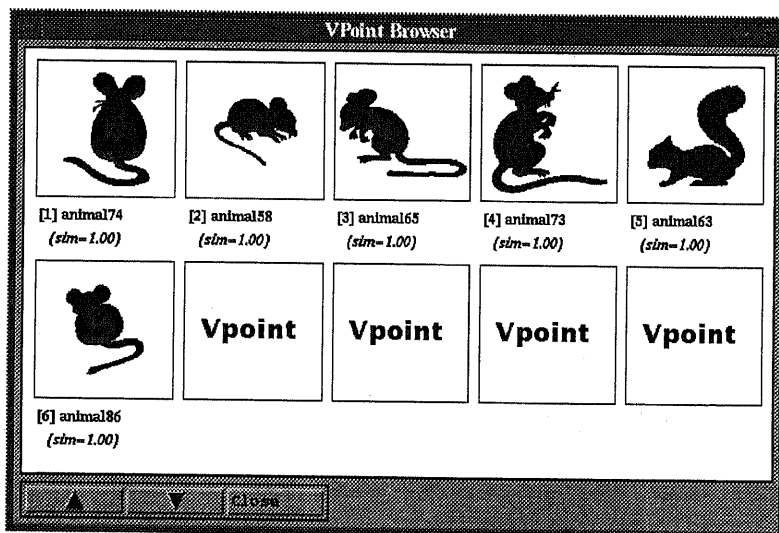
(c)

図 4.20 シルエット画像検索例 (1b)



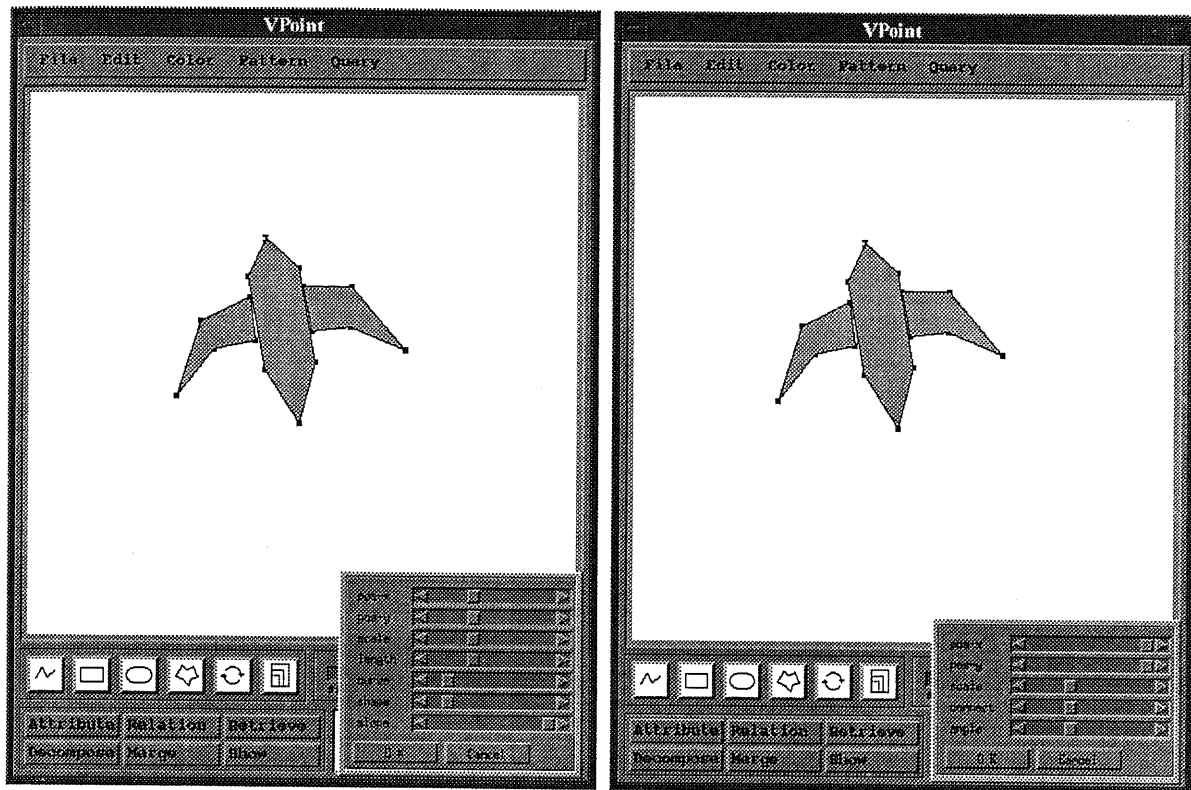
(a)

(b)



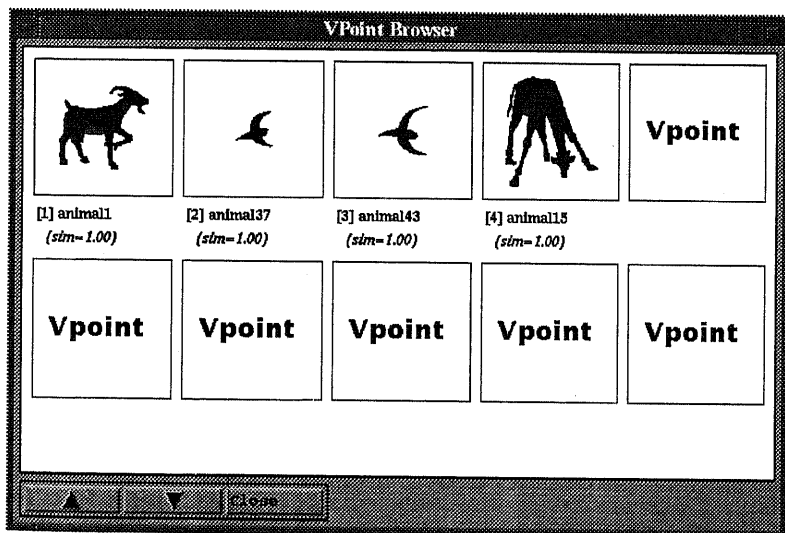
(c)

図 4.21 シルエット画像検索例 (1c)

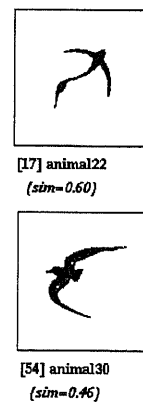


(a)

(b)

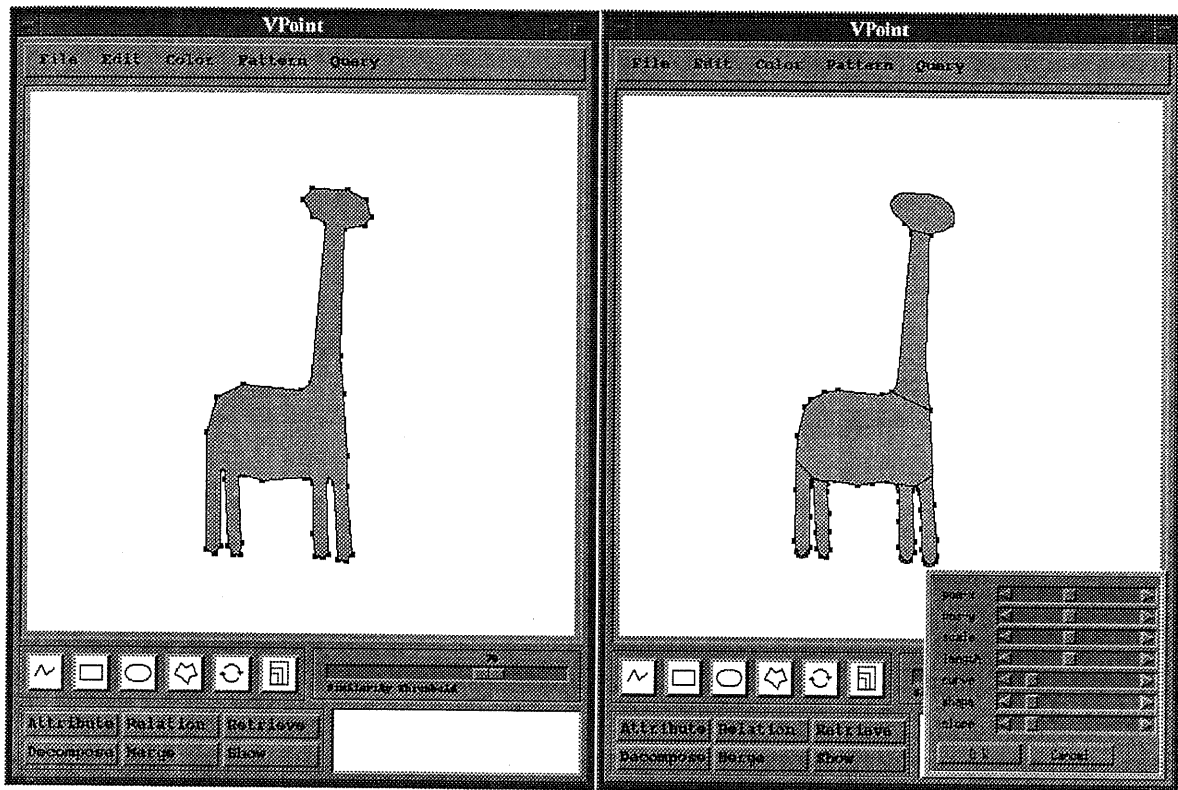


(c)



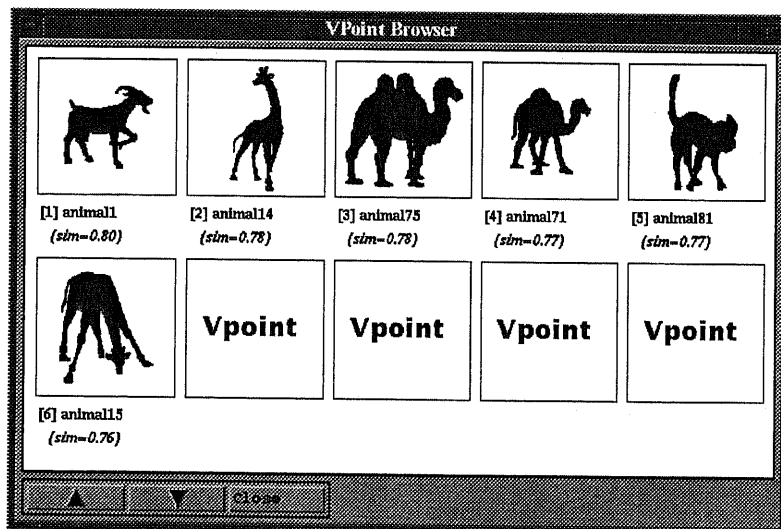
(d)

図 4.22 シルエット画像検索例 (2)



(a)

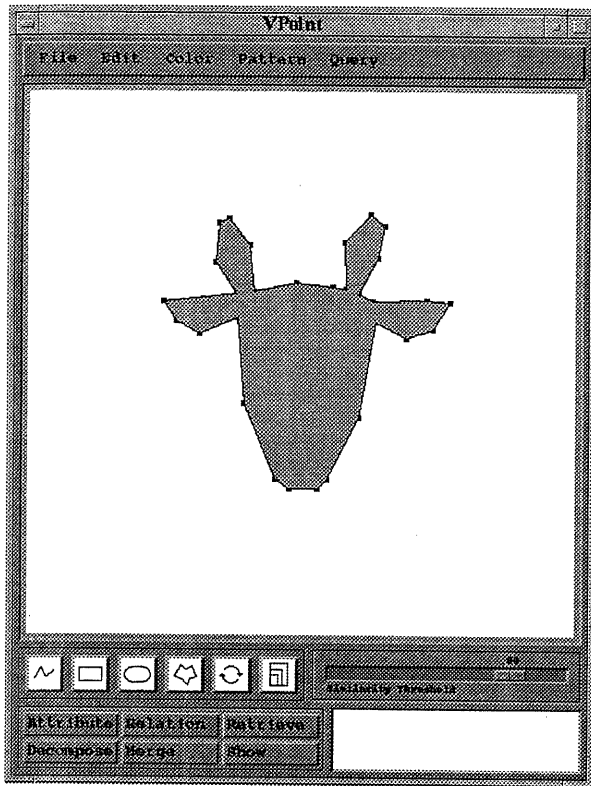
(b)



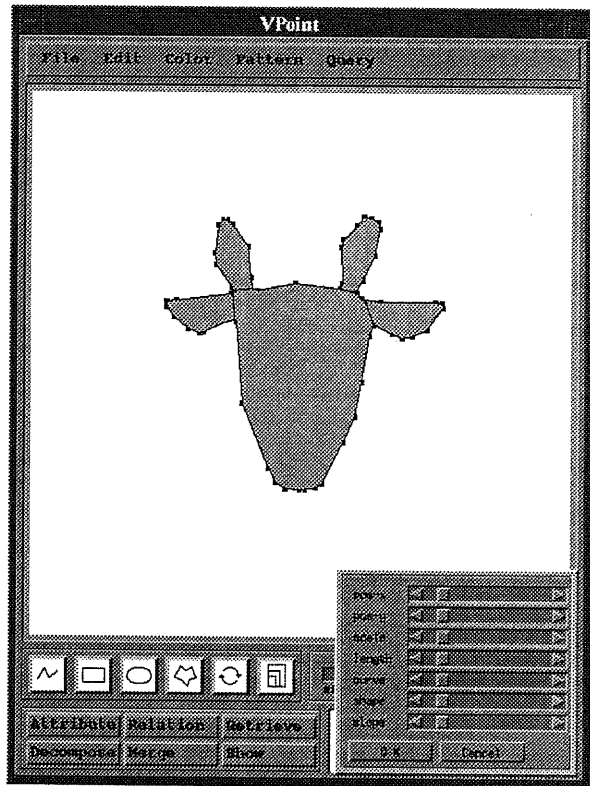
(c)

図 4.23 シルエット画像検索例 (3)

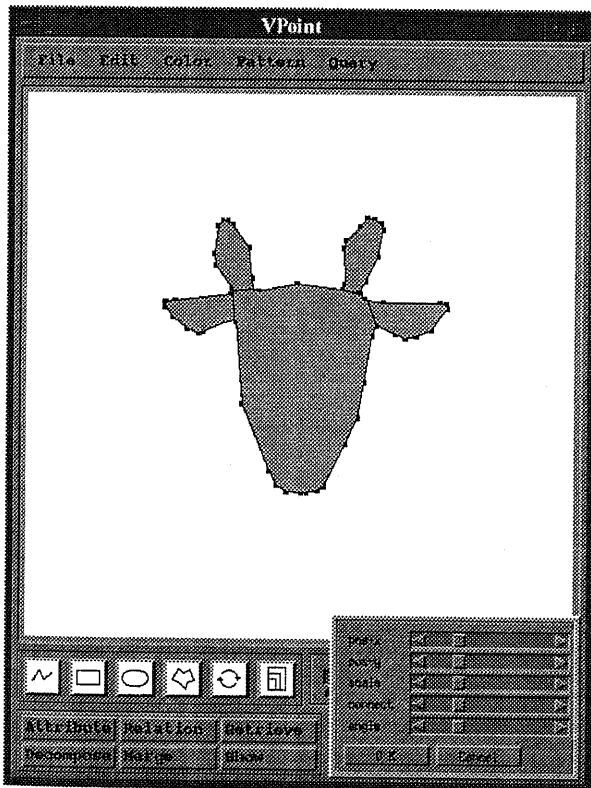




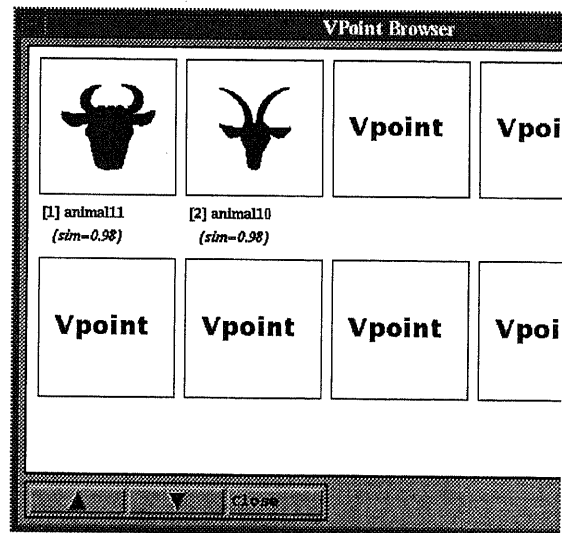
(a)



(b)



(c)

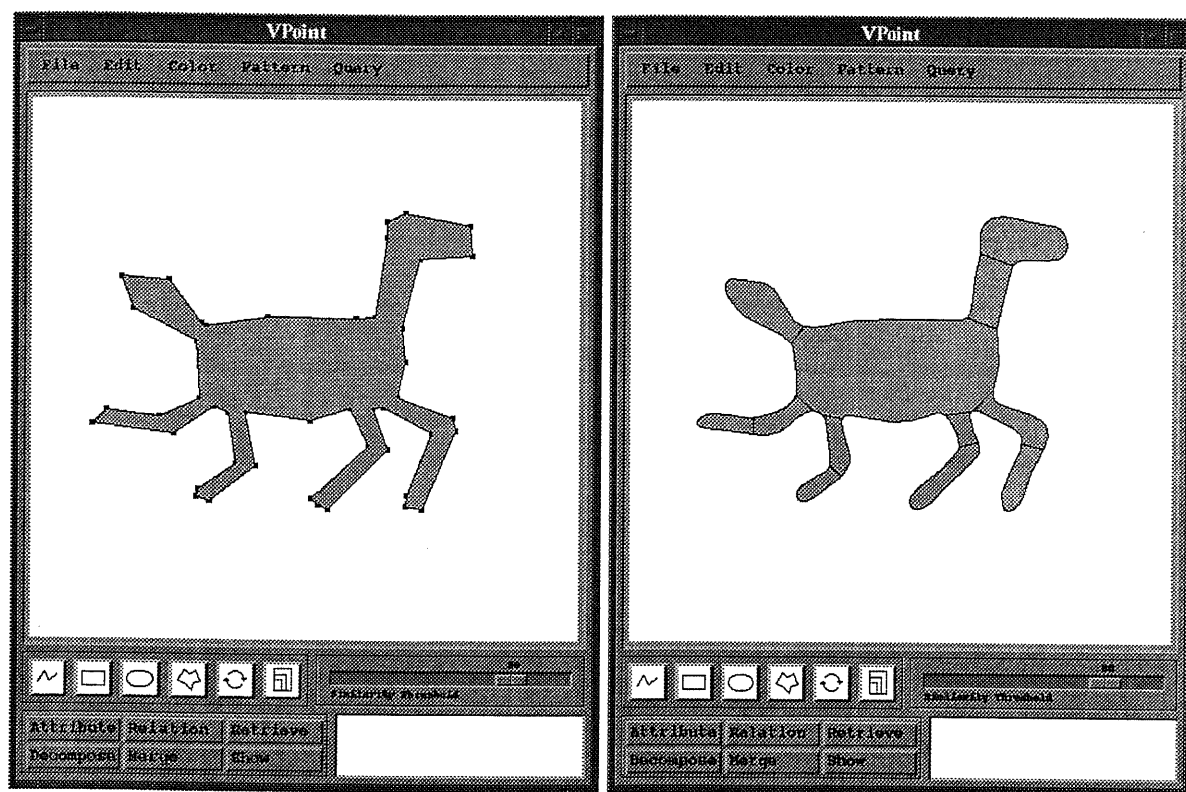


(d)



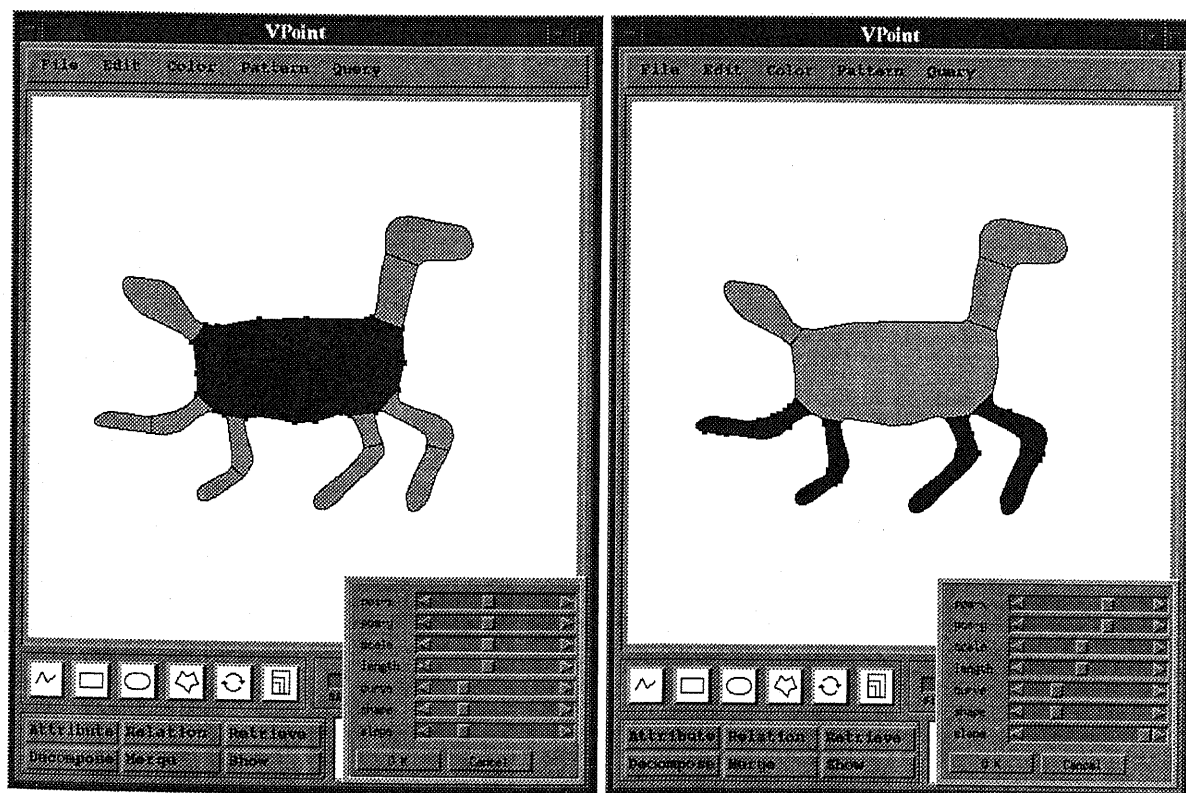
(e)

図 4.24 シルエット画像検索例 (4)



(a)

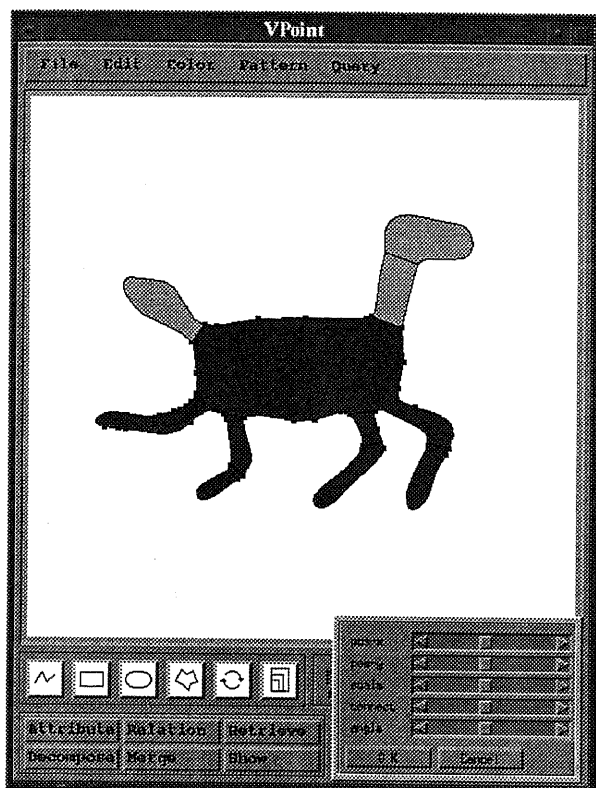
(b)



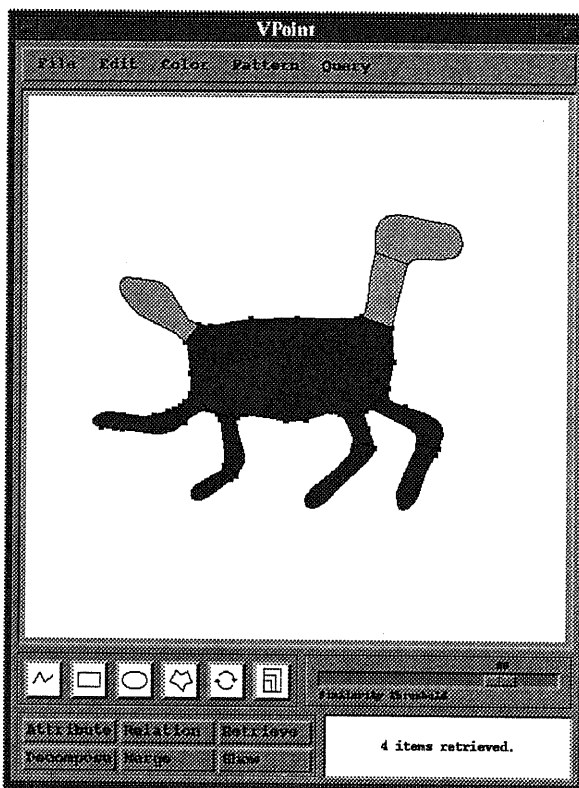
(c)

(d)

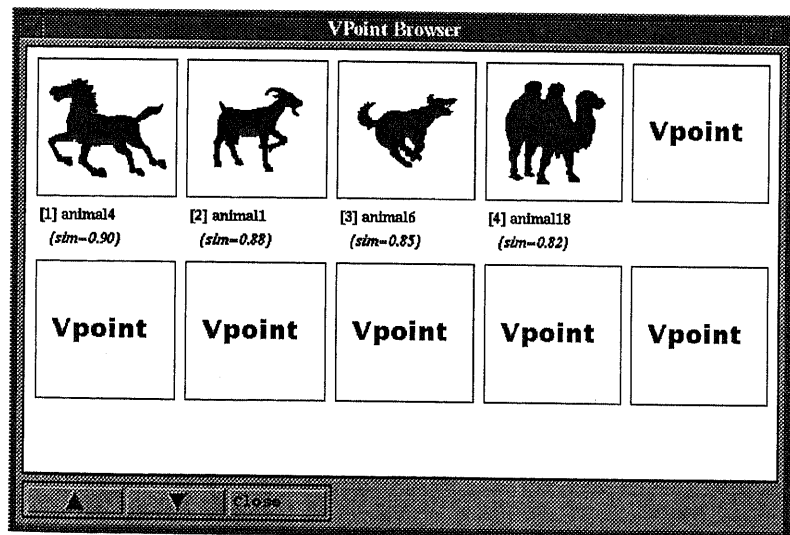
図 4.25 シルエット画像検索例 (5a)



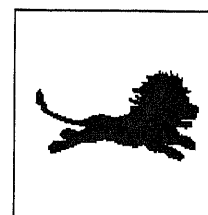
(e)



(f)



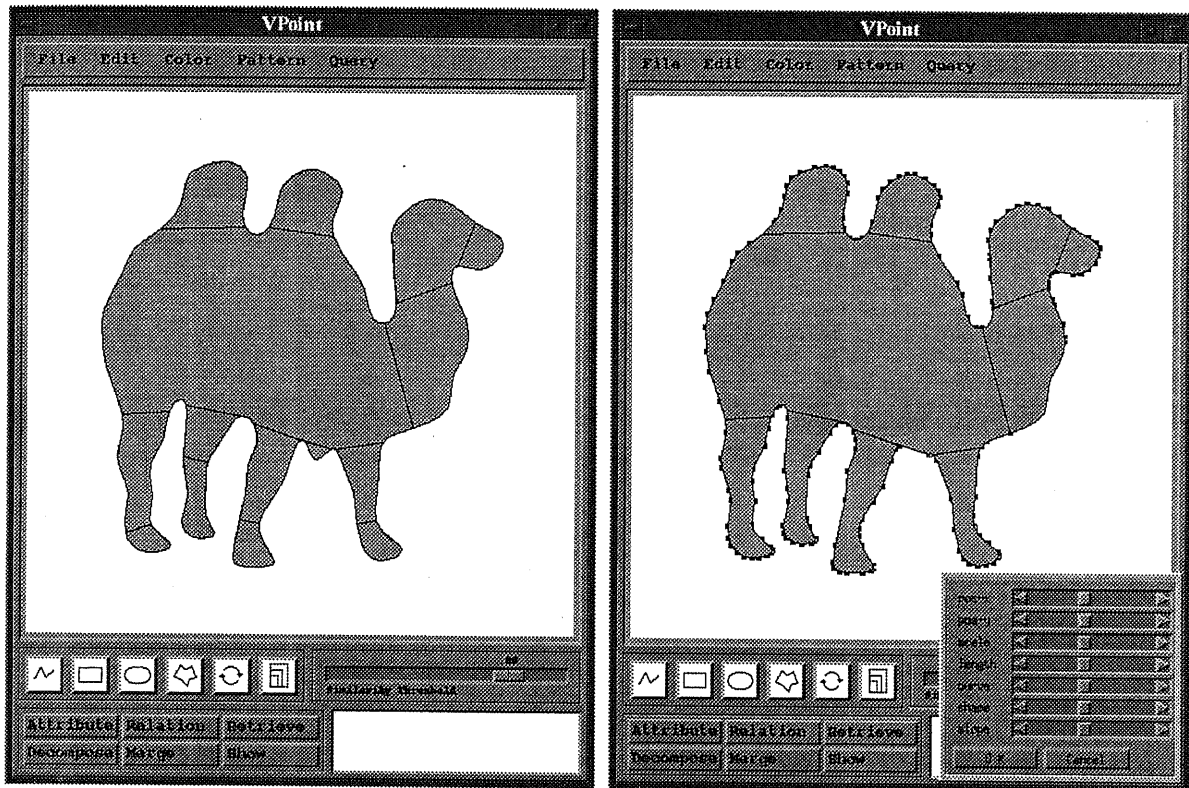
(g)



[45] animal20  
(sim=0.46)

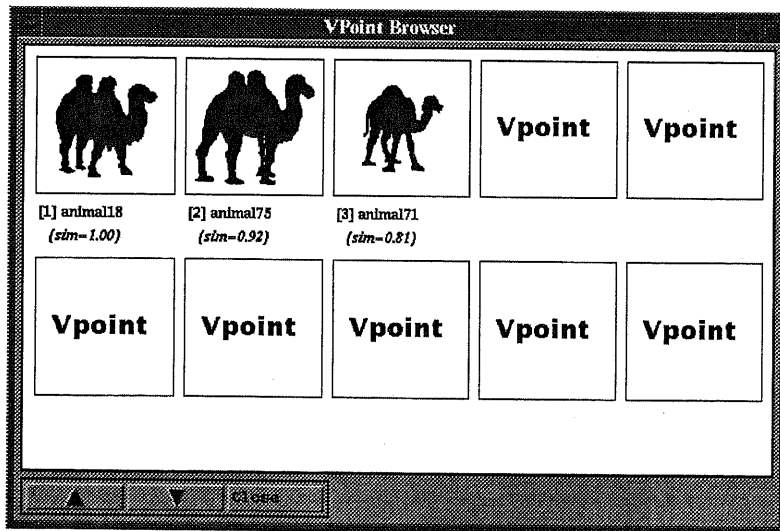
(h)

図 4.26 シルエット画像検索例 (5b)



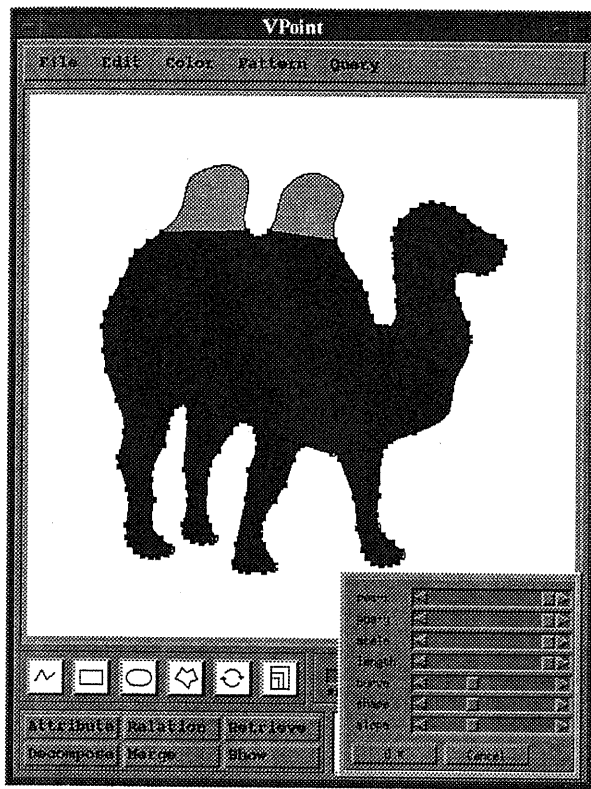
(a)

(b)

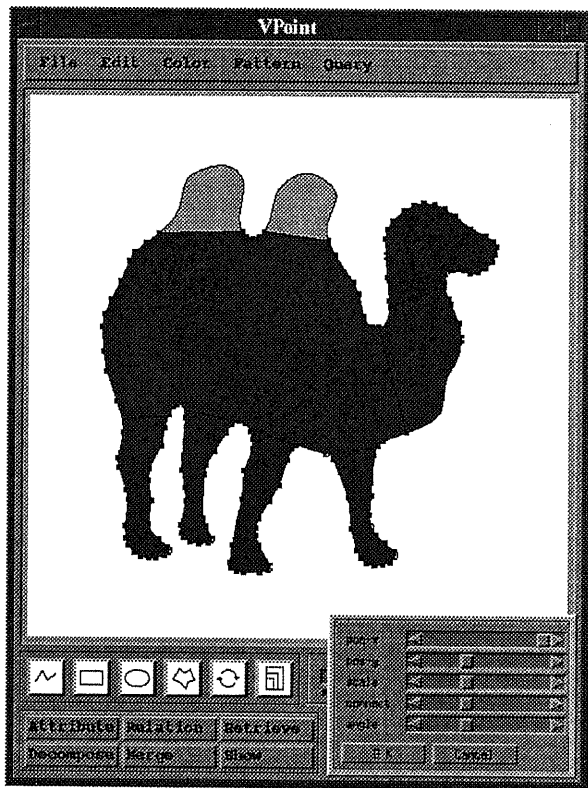


(c)

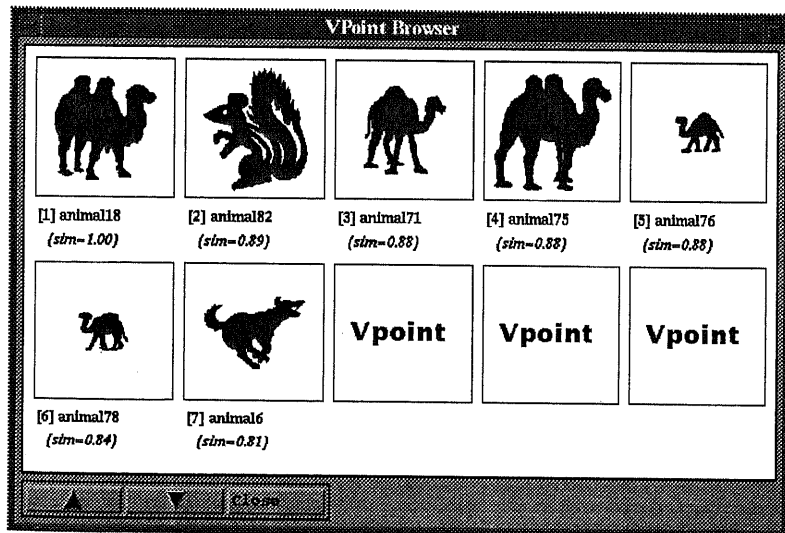
図 4.27 シルエット画像検索例 (6)



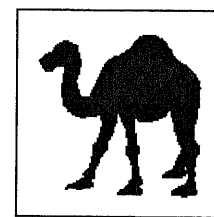
(a)



(b)



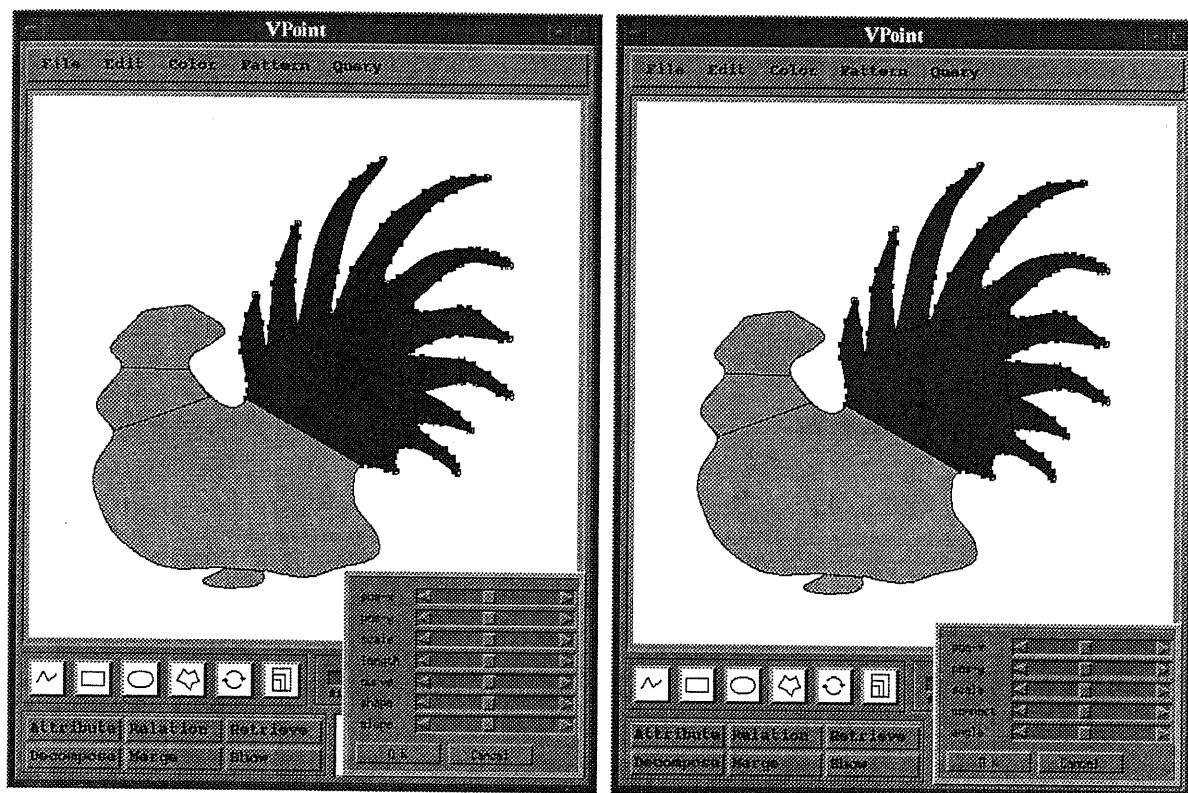
(c)



[8] animal68  
(sim=0.80)

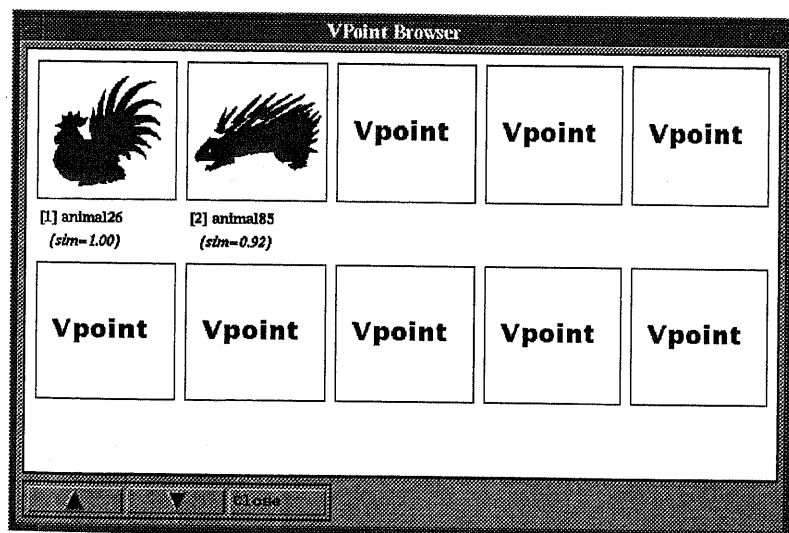
(d)

図 4.28 シルエット画像検索例 (7)

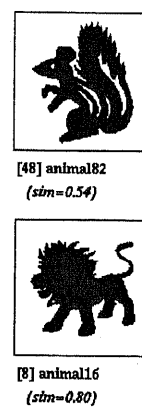


(a)

(b)

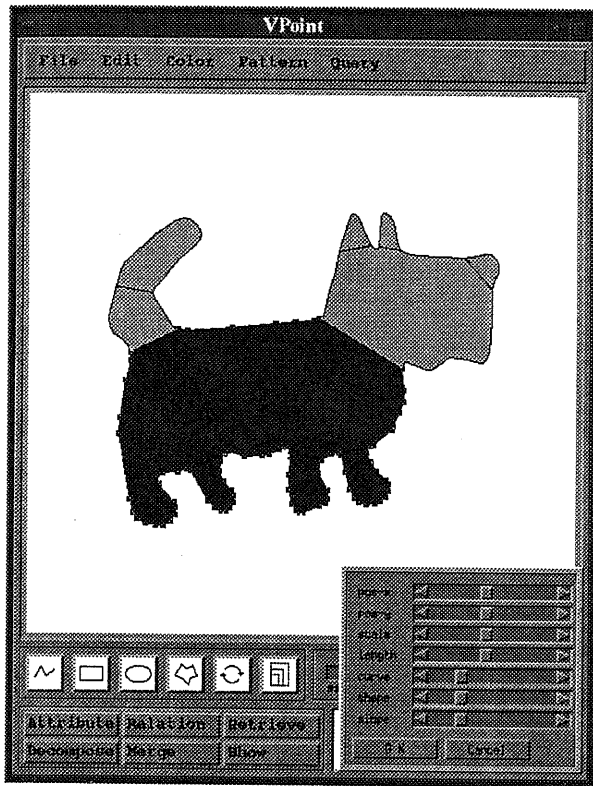


(c)

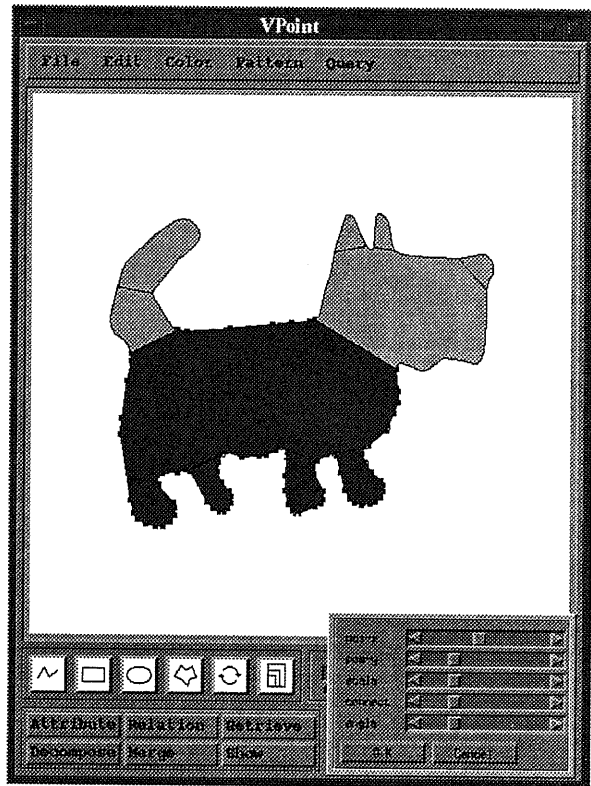


(d)

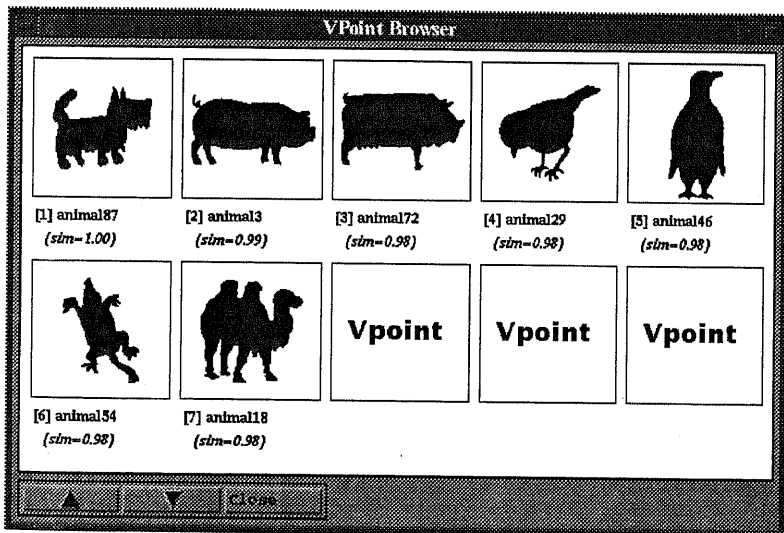
図 4.29 シルエット画像検索例 (8)



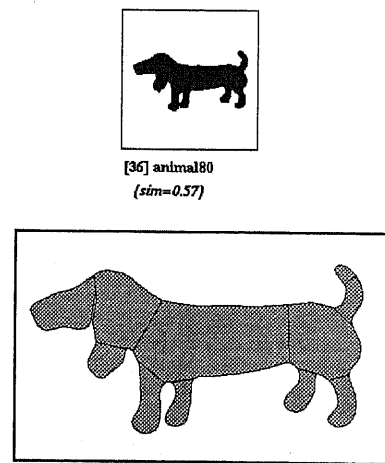
(a)



(b)

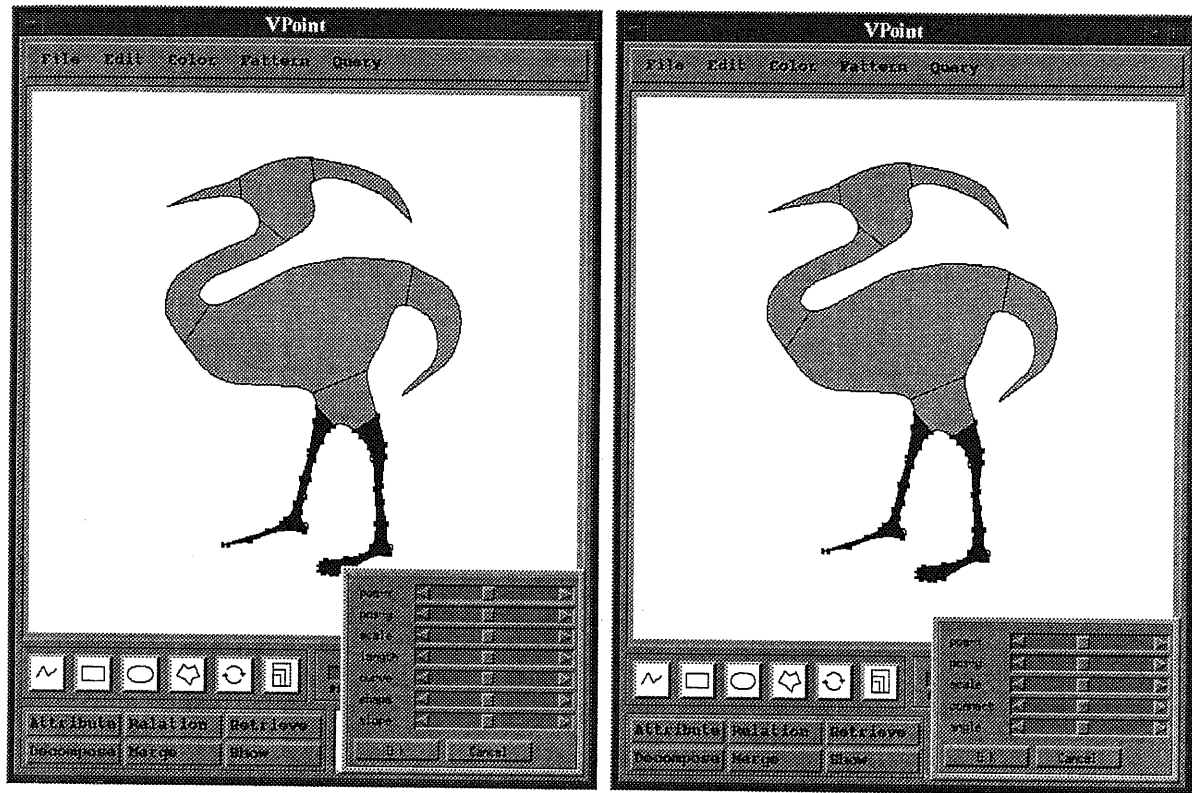


(c)



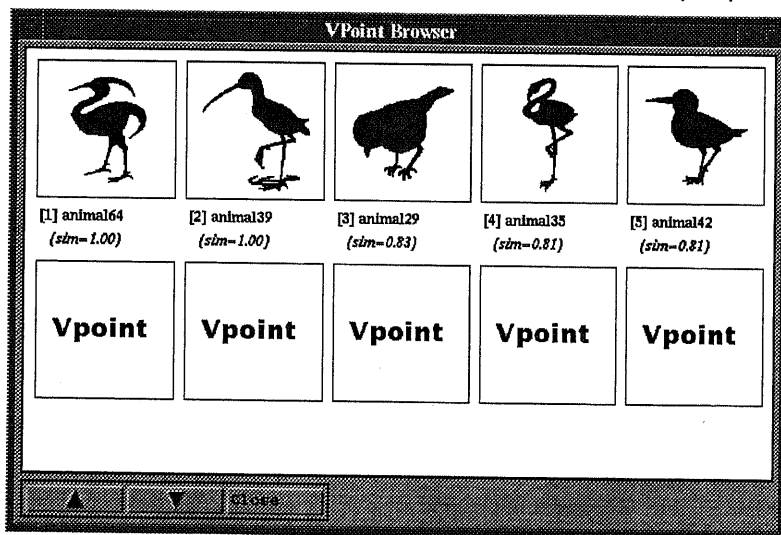
(d)

図 4.30 シルエット画像検索例 (9)



(a)

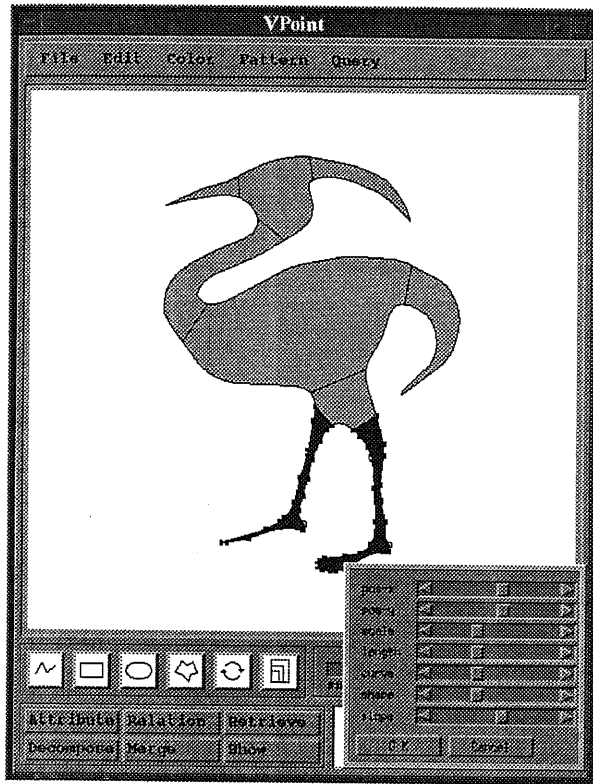
(b)



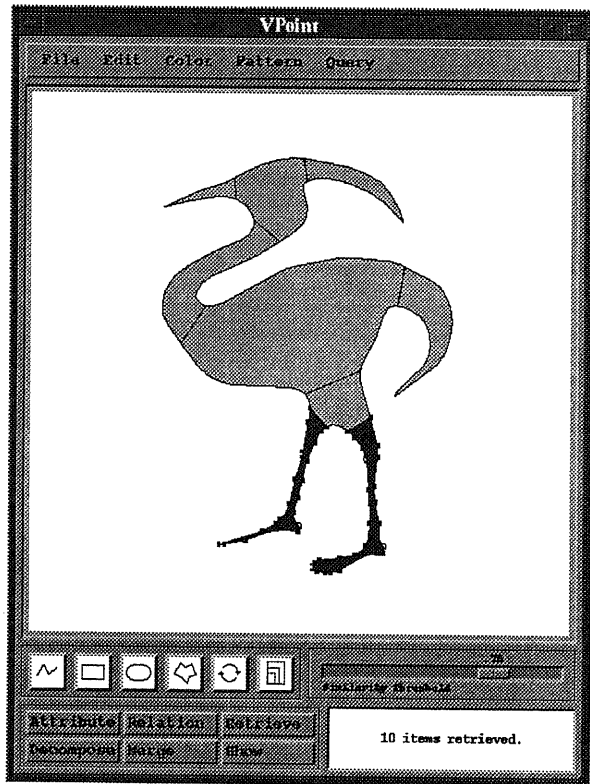
(c)

図 4.31 シルエット画像検索例 (10a)

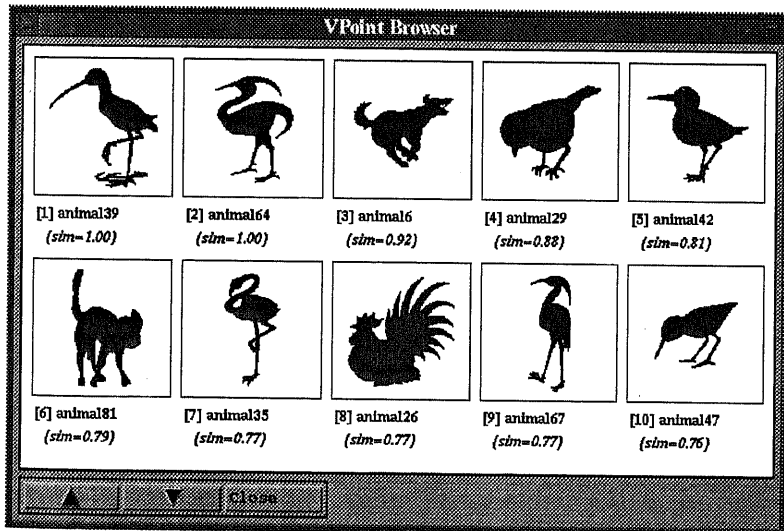




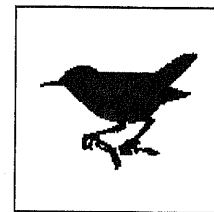
(a)



(b)



(c)



[39] animal25  
(sim=0.56)

(d)

図 4.32 シルエット画像検索例 (10b)

## 第5章

# カラー画像検索

## 5.1 はじめに

前章ではシルエット画像を対象にした検索について紹介したが、本章ではカラー画像を扱った場合を考え、その対象例として簡易なカラーパターン画像である国旗画像と対象不特定の自然画像を取り上げる。この二つの対象は、その扱う特徴量に若干の差はあるが、本研究でのプリミティブの抽出方法はほとんど共通している。よって本章では、まずプリミティブの抽出方法について述べた後、それぞれの対象について利用する特徴量について述べ、さらに検索例を示す。

## 5.2 領域分割による画像の構造化

### 5.2.1 領域分割処理

シルエット画像同様、画像を分解することで構造化し、それらプリミティブの集合として画像を記述する方針をとる。カラー画像の構造化というと、通常は領域分割処理のことが思い浮かばれるが、ここでも画像を領域分割することから始める。但し、主に画像の大域的または部分的概略に興味があるので、大きなしかも有意な領域分割を行い、画像を抽象化することが目的となる。

領域分割は、これも形状分解同様に古典的かつ重要な画像処理のテーマの一つであり [79, 94], 多くの手法が提案されている。古典的なものとしては領域成長法, 分離統合法, 画素結合法, 特徴空間でのクラスタ化手法, 弛緩法などがある [79]。また最近では分割 K 平均アルゴリズム [95], マルコフモデルを用いた手法 [96], 最小フロー問題に置き換えたアプローチ [97], morphology や watershed 法に関する手法 [98] などがあり, 枚挙すればきりが無い。しかしながらどの手法も一長一短であり, 残念ながら現状の技術では不特定の画像を有意な領域に分割できる手法はまずないと断言してよい。

そこで, 有意無意にかかわらず領域情報を抽出し, それら全てをプリミティブとして扱うという方針をとる。よってこれから述べる本手法は, 画像の領域分割についての新たな手法を提案するものではなく, 本研究の画像検索に適した手法という位置付けになる。

以下, その領域 (プリミティブ) 抽出処理について述べる。

### 5.2.2 プリミティブ抽出アルゴリズムの概要

図 5.1 に処理の概略を示す。まず前処理を施した対象画像のエッジ画像を作る。次にこの情報を利用して, 初期領域分割画像 (ラベル画像) の作成及び各領域の併合処理を行う。この併合の過程においてプリミティブ領域を抽出し, シルエット画像に対して用いた形状分解手法を施し, 最後に補正処理を行って出力する。

### 5.2.3 前処理 (Preprocessing)

処理の効率化及び雑音除去のために平滑化を行う。ここでは  $13 \times 13$  のガウスフィルタを R,G,B 各プレーンに畳み込むことで平滑化 [79] する。自然画像に対しては, さらに線形変換によりコントラストの強調 [79] を行う。

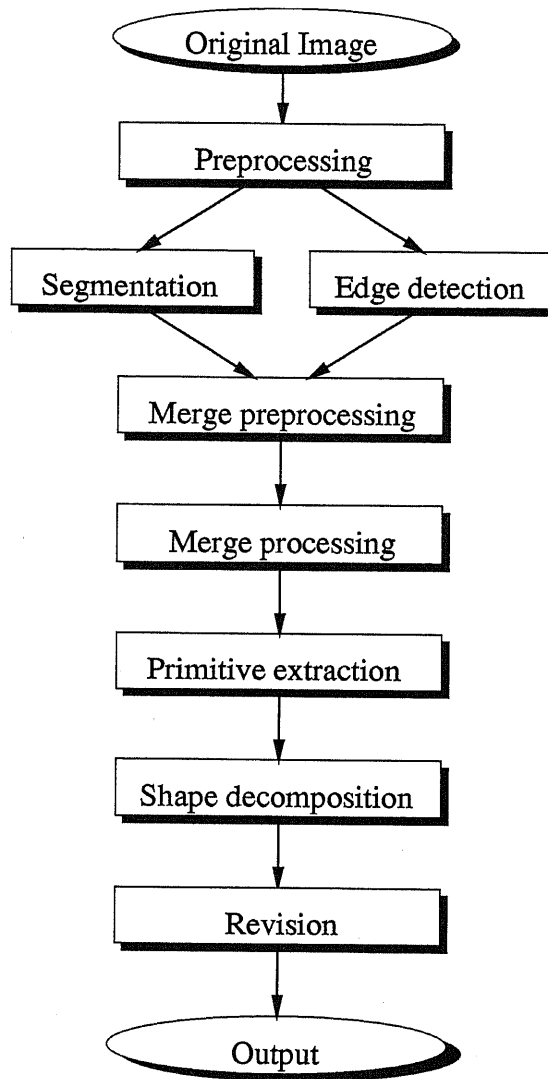


図 5.1 分割処理のフローチャート

#### 5.2.4 エッジ抽出 (Edge detection)

前処理画像を YIQ 変換によりグレースケール画像にし (256 階調), sobel オペレータを用いてエッジを抽出する. 得られたエッジ輝度を  $I(x, y)$ , その最大エッジ輝度を  $I_{max}$ , 最小エッジ輝度を  $I_{min}$  とし, 以下の式のように  $I(x, y)$  を 0.0 から 1.0 の範囲に正規化したものを  $f(x, y)$  とする.

$$f(x, y) = \frac{I(x, y) - I_{min}}{I_{max} - I_{min}} \quad (5.1)$$

### 5.2.5 領域分割処理 (Segmentation)

領域分割処理は Ohlander の再帰処理方法 [79] を用いる。この方法は R, G, B, Y, I, Q, H, S, I の色ヒストグラムを計算し、領域の抽出に最も適していると思われるヒストグラムを探し、そのヒストグラムから顕著なピークを検出する。その後、選ばれたピークに対応する領域を画像から切り出し、それらの画素から成る連結領域を求める。この処理を、求められた連結領域及びそれ以外の領域について再帰的に行う。

この方法では、R+G+B 及び R-B だけを用いるだけで十分であるという報告もなされているので [99]、これに従って2つのヒストグラムのみを扱う。またピークの検出の代わりに大津の2値化手法 [79] を利用して、最もクラス間分散が大きくなるような谷を見つけることでヒストグラムを分割する。また一定の画素数以下の領域に対してはそれ以上の分割処理は行わない。この対象となるのは原画像の画素数に一定比率  $\epsilon_s$  を乗じた値よりも小さな画素数を持つ領域とする。 $\epsilon_s$  の値が大きいと処理時間が短くてすむが満足な分割は得られない。逆に小さいと処理効率が悪くなる。

以上の処理によって得られた結果を初期分割画像とする。

一般に、このような特徴空間を用いる方法は分割領域が画像全体に無意味に広がってしまう欠点があるが、同時に大局的な分割ができるという長所もある。よって、ここでは初期状態としては過度に分割し、その後、併合処理を行うアプローチをとる。

### 5.2.6 併合処理 (Merge)

得られた初期分割画像には  $\epsilon_s$  が 0.01 程度で数百から多い時で数千の領域が生成される。これを以下に示す方法で併合する。

今、得られた分割画像を複数領域集合  $\{R_i\} (i = 0, 1, \dots, N)$  として表される。ここで  $i$  はラベルである。計算機上で扱う際には、ラベル画像  $L$  を用意して領域  $R_i$  に属する画素の値にラベル  $i$  を与える。 $R_i$  の持つ属性は  $A_i, r_i, g_i, b_i$  であり、それぞれ面積及び色 (RGB) の値である。領域の色は該当する原画像の画素値の平均値とする。また任意の異なる二つの領域に含まれる画素同士が4連結関係にある場合、その二つの領域は隣接関係にあると定義する。

全ての隣接関係にある2つの領域  $R_i$  と  $R_j$  について、併合した場合のコスト  $cost(R_i, R_j)$  が最小の組から併合する。階層的クラスタリングを行うわけである。 $cost(R_i, R_j)$  は以下のように定義される。

$$\begin{aligned} cost(R_i, R_j) &= \frac{E_i + E_j - E_{ij}}{B_{ij}} \\ E_i &= (r_i^2 + g_i^2 + b_i^2) A_i \\ E_{ij} &= (r_{ij}^2 + g_{ij}^2 + b_{ij}^2) (A_i + A_j) \end{aligned} \quad (5.2)$$

$E_i$  は領域  $R_i$  のエネルギーを表しており、 $r_{ij}, g_{ij}, b_{ij}$  は領域  $R_i$  と  $R_j$  とを併合した場合の平均色である。即ち、 $r_{ij} = (A_i r_i + A_j r_j) / (A_i + A_j)$  であり、 $g_{ij}, b_{ij}$  も同様である。

$B_{ij}$  は領域  $R_i$  と  $R_j$  との境界値である。ここでいう境界値とは、領域の併合の容易さを示すもので、直観的には領域間をつなぐパイプの太さのようなものを考えればよい。上記のように二つの領域を併合することは、ちょうど二つの異なる容器に入った液体がパイプを通じて一様に混じり合うようなもので、コストは一様に混合する定常状態までの時間とも捉えられる。よって境界値が大きいとはパイプが太いことに相当し、併合に要する時間がそれだけ短くコストが低いことになる。但し、ここで述べている併合の方法が、このような液体の混合をモデル化するものでないことは明らかである。

具体的に  $B_{ij}$  は先に求めたエッジ輝度  $f(x, y)$  から計算される。図 5.2 を参照されたい。まず、 $R_i$  と  $R_j$  の隣接画素に相当するエッジ輝度  $f(k, l)$ ,  $f(k', l')$  の平均を求めて画素間のエッジ輝度とする。そして、最大エッジ輝度 (1.0) からこのエッジ輝度を差し引いた値についての境界全ての総和を  $B_{ij}$  とする。

領域  $R_i$  と  $R_j$  が併合された場合、その併合領域の属性は  $A_{ij} = A_i + A_j$ ,  $r_{ij}$ ,  $g_{ij}$ ,  $b_{ij}$  になる。このように併合した場合の領域の色は 2 つの領域の平均色となる。

この  $cost(R_i, R_j)$  は、小さな領域や画素値が近いものほど早期に併合され、しかも隣接境界エッジ輝度が強いものは併合されにくいという考えのもとに定義されている。

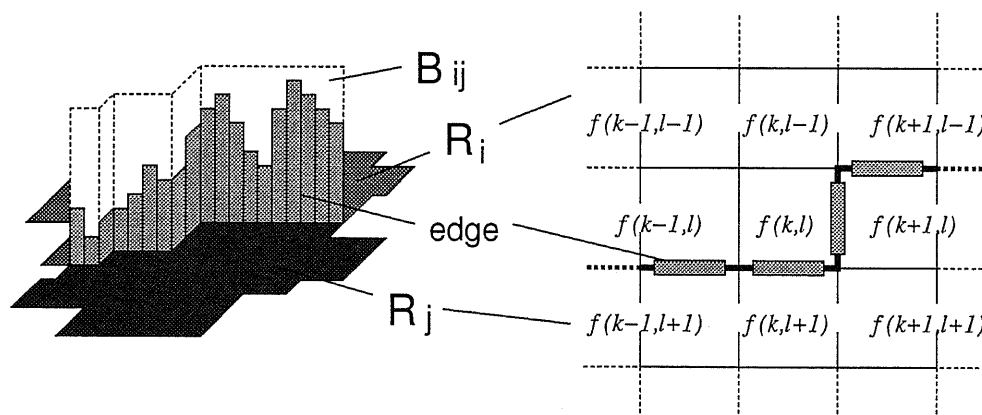


図 5.2 エッジ領域と併合領域

なお、以上の手順では、全ての領域の中から最もコストの低い領域のペアを探さなければならぬので効率が悪い。そこである一定以下の微小領域は、その領域と最小のコストで併合できる領域と強制的に併合させる (図 5.1 中の merge preprocessing に相当)。つまり微小領域に関してはその併合順序に制限をなくすわけである。具体的には、原画像の総画素数一定比率  $\epsilon_r$  を乗じた値よりも小さな画素数を持つ領域を微小領域とする。

### 5.2.7 プリミティブの抽出 (Primitive extraction, Shape decomposition)

以上の処理を領域が一つになるまで行い、併合の履歴をとる。図 5.3 は併合の様子をノードとアークで示したものである。ノードは領域に相当し、実線のアークは併合の過程を示し、全体と

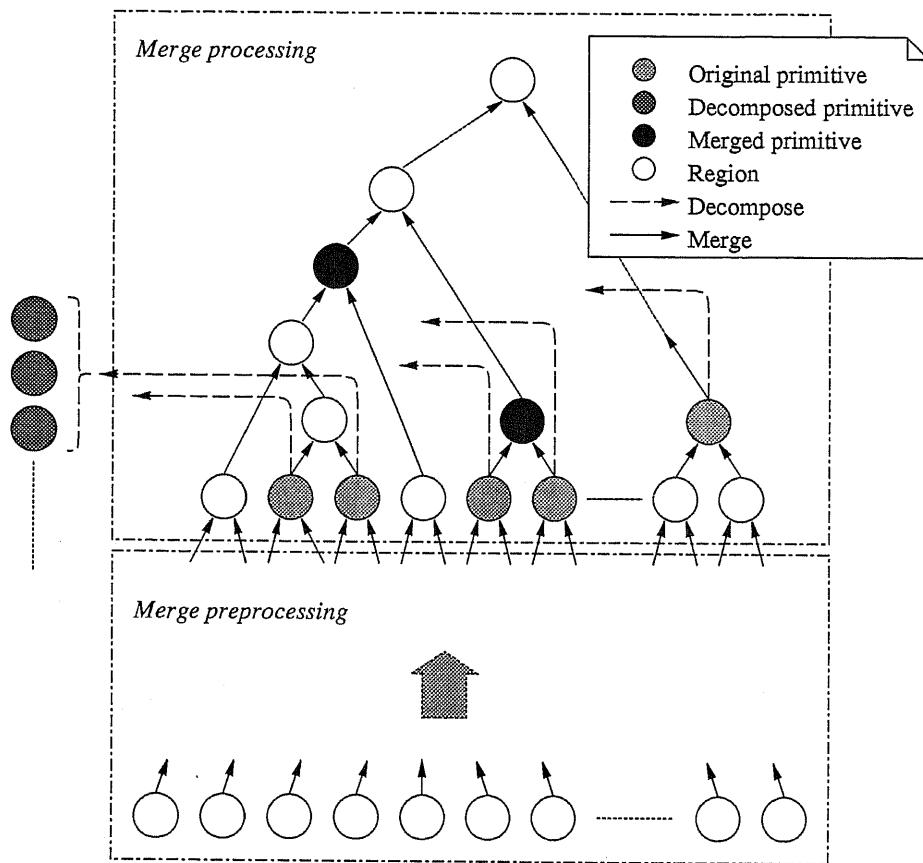


図 5.3 プリミティブの抽出

して木構造になる。

さて、理想的には検索に用いることができる有意な領域をプリミティブとして抽出したいので、全てのノードの中から有意なノードを選びたい。しかし自動的に識別することは非常に困難である。そこで、併合時のコスト  $cost(R_i, R_j)$  が高いものをプリミティブとして抽出することにする。まず、領域が一つになるまでの全コスト  $cost_{all}$  を併合の履歴より計算し (merge preprocessing も含む)、それに対し一定の割合 ( $\epsilon_c$ ) よりも上回るコストを要した領域をプリミティブとして出力する。例えば、 $cost(R_a, R_b) > cost_{all} \times \epsilon_c$  ならば  $R_a, R_b$  は有意である可能性があるととしてプリミティブとして出力する。

今、 $R_a$  と  $R_b$  がプリミティブと判定されたとして、もし  $R_a$  を根とする部分木内に既に出力されたプリミティブがある場合、 $R_a$  を merged primitive と呼ぶことにする。また部分木内にプリミティブがない場合は  $R_a$  を original primitive と呼ぶことにする。

なお、出力されるプリミティブの領域の色は平均色ではなくて代表色を用いる。ここで代表色とは、原画像中の該当領域内で最もヒストグラムの大きい色を指す。平均色では色が濁ってしまい、検索の際に適当でないからである。

このようにプリミティブを抽出していくわけであるが、色とエッジ輝度を基にしたコストを基

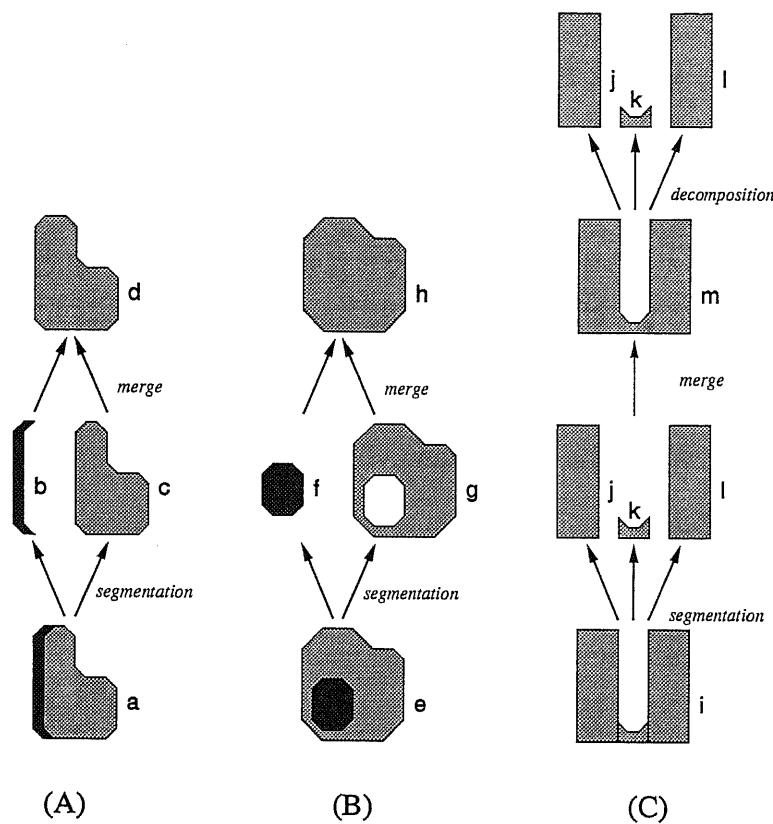


図 5.4 不十分なプリミティブの判定

準にしてプリミティブを決定しているために、形状に関する制約がない。つまり実際には類似した形状にもかかわらずコストが高いために無意味なプリミティブを出力してしまう場合（図 5.4 (A)）と、併合時に形状的に大きな変化があるにもかかわらずコストが低いためにプリミティブとして出力されない場合（図 5.4 (C)）が起きてしまう。

まず前者の例である。図 5.4 (A) では、b, c がプリミティブとして出力されたとして、d も後の併合過程でプリミティブとして出力される恐れがある。明らかに c と d は類似しており 2 つを出力するのは無駄である。そこで、この対策として周長と面積を考慮する。今、領域  $R$  がコストの条件を満たして、merged primitive として出力されうるとする。ここで  $R$  を根とした各葉の方向への全ての経路の中で、最初に見つかる primitive として出力済みの領域を  $R_i (i = 0, 1, 2, \dots)$  とし、その中で最も面積の大きい primitive を  $R_k$  とする。そして以下の式のどちらかの条件を満たした場合、 $R$  は merged primitive として出力されるとする。これは形状の変化が小さければ、たとえコストが高くとも出力しないための条件になる。ここで  $L_X, A_X$  は領域の周長と面積を表す。図 5.4 (A) では、d と c の周長と面積の差に変化がないので、d は出力されない。

$$|L_R - L_{R_k}| / L_{R_k} > \epsilon_p$$



$$|A_R - A_{R_k}|/A_{R_k} > \epsilon_p \quad (5.3)$$

これにより図 5.4 (A) のような場合は対処できるが、なお図 5.4 (B) のような“穴”を持つ領域についても考慮しなければならない。本手法では領域がプリミティブとして出力される際には穴の情報を全て削除するようにしている。しかし領域分割の段階では、穴の情報は残っているので図 5.4 (B) の例では、g から h への面積への変化が大きいので、g も h も全く同じプリミティブとして出力されうる。よって図 5.4 (B) のような重複しているプリミティブを検証して削除する。

次に後者の場合である。図 5.4 (C) では j, k, l が出力されない恐れがある (k はどちらにしろ不要であるが)。また、領域分割の段階で既に i の状態以上に分割されないかもしれない。これに対しては図 5.4 (C) のようにシルエット画像に適用した形状分解手法をここでも用いる。このように形状分解で得られたプリミティブを decomposed primitive と呼ぶことにする (図 5.3)。ここで、全てのプリミティブに対して形状分解を行うのが好ましいが、それではプリミティブの数が増加しすぎるので、基本的に original primitive のみを形状分解するものとする。

### 5.2.8 後処理 (Revision)

以上の処理でプリミティブが抽出できたので、対象や目的に応じて後処理をする。ここでは形状分解によって生じた微小なプリミティブを削除する。

### 5.2.9 パラメータの設定

さて、以上の処理において決定すべきパラメータは、最初の色空間による分割の際の停止条件でもあるパラメータ  $\epsilon_s$ 、微小領域の併合のためのパラメータ  $\epsilon_r$  及びプリミティブ判定のためのコストの閾値を示すパラメータ  $\epsilon_c$  と  $\epsilon_p$  である。多くの画像処理において、いくつかのパラメータはつきものであり、その設定自体も一つの重要な問題である。本手法も例外ではない。ここで最も設定値が重要かつ困難なのは  $\epsilon_c$  である。その他は、多少の値の変動があっても結果に大きな影響をもたらすものではない。以下、このパラメータ値の設定の見通しと影響について述べる。最終的に用いる値は次節以降で改めて述べる。

分割の目的はシグネチャ表現のためのプリミティブの抽出であるので、その数は自然画像でも、せいぜい 100 くらいであると考えられる。また 100 程度もあれば、原画像の概略表現には十分である。特に国旗画像の場合は、細かい模様が多い国旗以外は 20 もあれば十分である。これは複数のサンプル画像に対しての実験に基づく経験的な判断である。

パラメータ  $\epsilon_s$  : さて、仮に  $\epsilon_s = 0.001$  とした場合、原画像の 1/1000 の領域まで分割対象とすることになる。これから得られる初期分割の領域数は、前処理を施した SIDBA の標準画像 home (256×256) で 3,205 であった。また 0.01 とした場合で 1,594、0.1 の場合で 1,117 の初期分割が生じる。しかし、他の条件を同一にした場合の併合結果の最終領域数 (プリミティブの数) には大差は出ない。ちなみに、この場合それぞれ 52, 52, 46 であった。これは他の複数の画像についても同様であった。つまり、ある一定以上の微細な分割は、大勢に影響がないということになる。よって、 $\epsilon_s$  は特に大きな値にしない限りは特に影響はない。

しかし、小さい値にした方がより微小な領域を分割するので、色の区別が不明瞭な対象を細かく分割したい場合は値を小さくとらなければならない。逆に、デザイン画や国旗のように比較的色の区別が明確な対象の場合は大きい値でもよい。目安としては、0.01 以下であれば特に問題はない。特に効率にこだわらなければ 0.001 くらいにしておけば十分である。

パラメータ  $\epsilon_r$  : 次に  $\epsilon_r$  の値であるが、これも  $\epsilon_s$  と同様にある程度小さい値にしておけば結果に及ぼす影響は小さい。ただし、この値が小さいほど精度がよくなることは言うまでもない。値が小さければより適切に併合が行われるので、出力されるプリミティブの形状もより原画像に忠実になる。パターン画像などを対象とした場合には小さめにとるのがよい。大きな値だと、例えば星形のプリミティブが少し丸みを帯びて出力されてしまったりすることがある。目安としては、 $\epsilon_r = 0.001$  くらいであろう。

パラメータ  $\epsilon_c$  : さて、最も設定が難しいのは  $\epsilon_c$  である。図 5.5 に 4 枚の画像についてのコストと領域数の関係を示す。領域が一つになるまでに、各併合処理で算出されるコストを測定したものである。例に用いた画像は SIDBA の標準画像 home と mandrill (共に  $256 \times 256$ ) であり、残りはパターン画像の例として、韓国と米国の国旗 (共に  $450 \times 300$ ) である。この測定値の条件は  $\epsilon_s = 0.01$ ,  $\epsilon_r = 0.0001$  である。その他のパラメータはこの測定に関しては関係がない。また、この二つのパラメータの値を特に大きな値にしない限りは、グラフの概形はこれらのパラメータの変化にほとんど影響を受けない。

まず図を見て分かるように、自然画と国旗のようなパターン画像では顕著な差がある。自然画では全体的に滑らかにコストが上昇していくが、国旗の場合は急激にコストが変化する。米国の国旗の場合には、星形の領域が併合され出すとコストが上昇し、全ての星形の領域が併合されるまで同様のコストがかかる。さらにその後、横縞の領域が併合され出すと、増減を繰り返して高いコストを出す。これらの例より色の区別が明確な画像ほどコストの変化が激しいことがわかる。

明らかに、このような画像に一樣に最適な閾値  $\epsilon_c$  を設けるのは困難なことである。しかし、本手法では最適な値は必要ではない。余計な領域を含んでも構わないので低い値を設定しておけばよい。しかしあまり低いとプリミティブの数が多くなりすぎてしまうので、その辺りを考慮して決める必要がある。目安としては、図 5.5 のようなデータをいくつかとった結果、0.001 から 0.005 くらいの範囲が適当と考えられる。

パラメータ  $\epsilon_p$  :  $\epsilon_p$  の値を大きくすれば merged primitive の判定基準を厳しくすることになる。人間の直観を考慮しなければならないが、元の周長や面積の半分程度の変化が起これば大きい変化であるとして 0.5 あたりを目安とする。

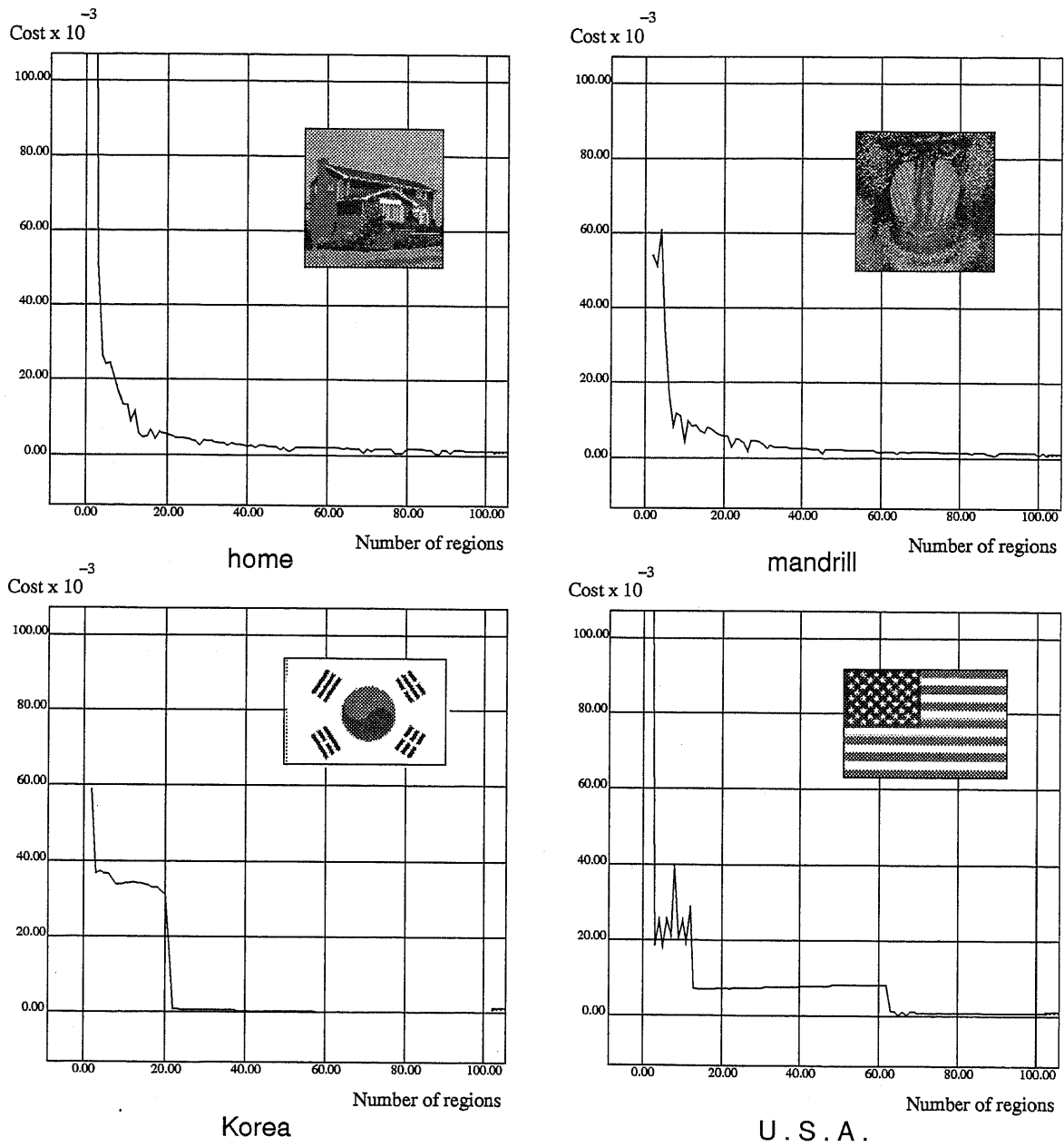


図 5.5 領域数と併合コストの関係

### 5.3 国旗を対象とした検索

既に述べたように、本研究ではカラー画像については二種類の対象について実装を試した。一つは国旗の画像 190 枚を対象としたものである。国旗を対象とした理由は、自然画像などよりも、このようなデザインやパターン画像の方がより本研究の主旨や利点が明確になると考えたからである。よって国旗画像を扱う目的は、あくまで、本研究の主張点を明確にすることにあり、国旗画像のための実際のアプリケーションの構築そのものではない。

#### 5.3.1 プリミティブ抽出処理

プリミティブ抽出の方法は前節で述べた通りである。

国旗画像は自然画像に比べて領域分割がかなり容易になるが、ここで扱う国旗画像はスキャナから入力したので雑音の影響を無視できない。例えば図 5.6 に示されるような影響が出る。このような場合の対策のために前節で説明したように形状分解処理を適用するわけである。

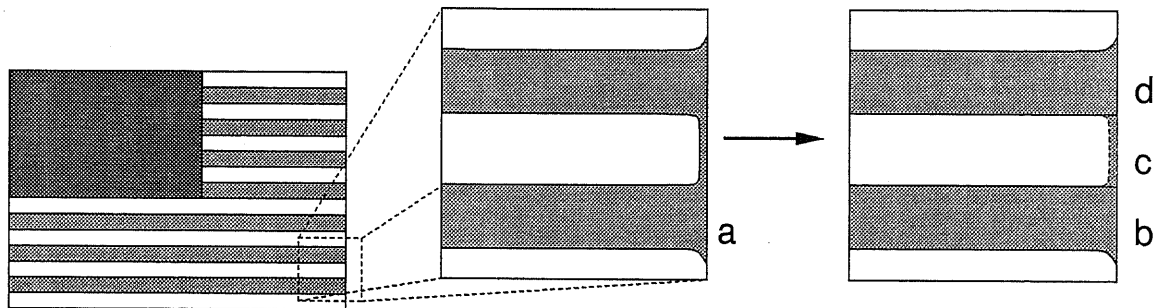


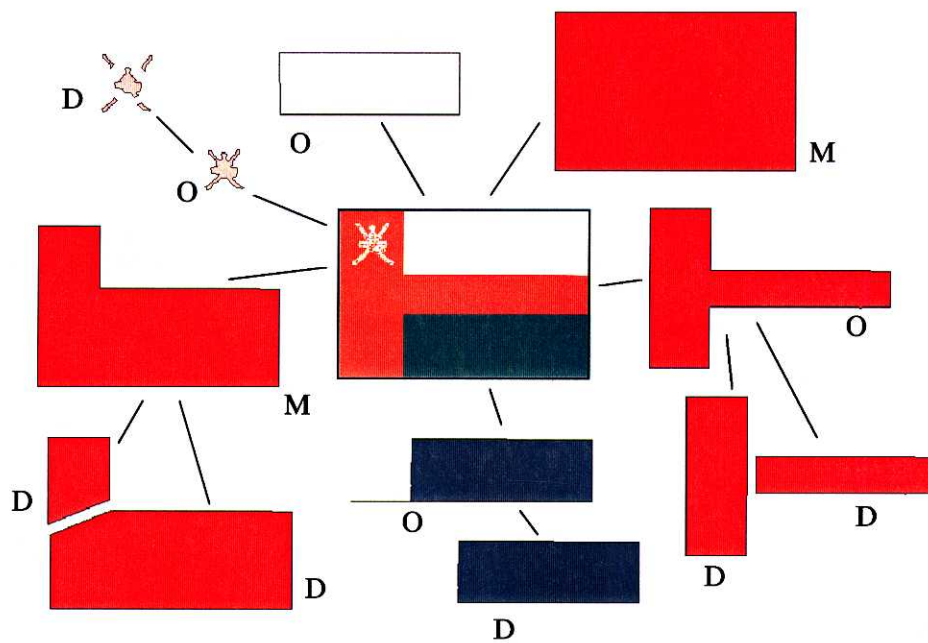
図 5.6 雑音による誤った分割

処理対象の画像はスキャナから取り込んだ 190 枚の  $450 \times 300$  の 24bit の国旗の画像である。パラメータは前節で述べた見通しに従って以下の通りに設定した。

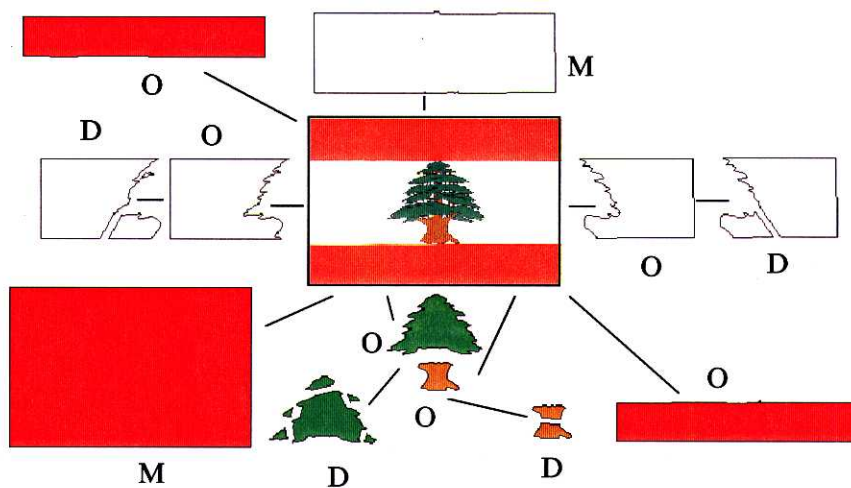
$$\begin{aligned} \epsilon_s & 0.1 \\ \epsilon_r & 0.0001 \\ \epsilon_c & 0.005 \\ \epsilon_p & 0.5 \end{aligned}$$

なお、形状分解におけるパラメータは  $t_{max}$  を 5 に設定した以外は、4 章で述べた値と同じである。このように低い値にした理由は、あまり細かい形状分解は不要と考えるからである。ここでは図 5.4 で示したような場合に対処できる程度でよい。また、形状分解の際の対象形状の輪郭の平滑化は、フーリエ記述子ではなく、Scale Space Filter[100] を用いる。このスケール値（分散値に相当）は 5 に設定した。

図 5.7 に抽出例を示す。図中、O, M, D はそれぞれ original primitive, merged primitive, decomposed primitive を表す。decompose primitive は original primitive からのみ抽出すると述べたが、国旗画像を対象にした場合、merged primitive から形状分解により抽出する。



(a)



(b)

図 5.7 国旗画像のプリミティブ抽出例

### 5.3.2 特徴記述

シルエット画像の場合と同様に、各プリミティブの属性及び関係に関する特徴量を定めて、それを検索用のシグネチャに割り当てる。

#### 属性

用いる属性は以下の 9 属性である。

1. 水平方向位置：プリミティブの中心の  $x$  座標。5 ビットを割り当てる。
2. 鉛直方向位置：プリミティブの中心の  $y$  座標。5 ビットを割り当てる。
3. 大きさ：画像全体の面積に対するプリミティブの面積の比率。5 ビットを割り当てる。
4. 円形度：面積 / (周長  $\times$  周長)  $\times 4\pi$ 。5 ビットを割り当てる。
5. 針状度：まずプリミティブの凸包を求め、その中心と主成分を主成分解析によって求める。次に求めた第一主成分を水平になるように凸包を回転させて、その状態での外接矩形を求める。この外接矩形の縦の長さの横の長さに対する比を針状度と定義する。5 ビットを割り当てる。
6. 凹凸度：プリミティブに対して SSF をかけて平滑化し、さらにコーン交差法により多角形近似する。その上で各頂点の曲率を計算して凹凸のペアの数を計上する。8 ビットを割り当てる。
7. 傾き：プリミティブの輪郭の主成分を求め、その第一主成分の傾きを軸の傾きとする。4 ビットを割り当てる。
8. 明度：領域の RGB の代表色を HSI 変換して、その明度を用いる。5 ビットを割り当てる。
9. 色相：同じく HSI 変換したもののうちの色相を用いる。12 ビットを割り当てる。

以上計 9 属性、54 ビットとなる。位置と大きさ及び向きに関する属性はシルエット画像を対象とした場合と同じである。形状に関しても同様であるが、面積と周長との関係、凸包、曲率などを用いる。色に関しては RGB の値をそのまま用いても人間の直観とそぐわないので、HSI 変換を行ってマンセル表色系を利用する。但し、色相が不定の場合（白、黒に相当）は色相の属性にはビットは立たない。このように個々の特徴はシルエット画像の場合と同様、極めて簡易なものばかりである。

#### 関係

面積の大きいプリミティブを第一プリミティブ (First Primitive)、もう一方を第二プリミティブ (Second Primitive) として関係を結ぶ。

1. 相対水平方向位置：第一プリミティブから見た第二プリミティブの  $x$  方向の位置関係. 3 ビットを割り当てる.
2. 相対鉛直方向位置：第一プリミティブから見た第二プリミティブの  $y$  方向の位置関係. 3 ビットを割り当てる.
3. 相対的な大きさ：二つのプリミティブの面積比. 3 ビットを割り当てる.
4. 包含関係：第二プリミティブが第一プリミティブに包含されているかどうかを示す. 2 ビット (包含, 非包含) を割り当てる.
5. 色関係：色相間の距離. 3 ビットを割り当てる.
6. 角度：二つの軸がなす角度. 3 ビットを割り当てる.

計 6 関係 17 ビットとなる. 相対位置関係, 相対的な大きさ, 角度はシルエット画像の場合と同様である. 色関係は双方のプリミティブの色相が定まっている場合のみ設定される. 距離とはマンセルの色相環における 2 つの色相間の角度のことである.

ここでも全ての領域間の関係を求めるわけにはいかないの, 以下のいずれかに該当するプリミティブの組み合わせに関しては関係を結ばない.

- プリミティブの凸包の面積に比例した量だけ, そのプリミティブの凸包を膨張させて時に, 互いに交わらないプリミティブの組み合わせ.
- 包含関係にあり, かつ色が同じプリミティブの組み合わせ.
- 包含関係にあり, かつ original primitive と decomposed primitive の組み合わせ.

なお, シルエット画像とは異なり, ある二つの領域間の関係は各々が第一プリミティブになるように二通りの記述を行った.

### 5.3.3 検索例

以上の結果を基に構築した VPoint の国旗を対象とした場合の検索例を示す. 190 枚の国旗画像に対して, 出力されたプリミティブ (属性) の数は平均 15.8, 同じく属性の数も 15.8, 関係の数は平均 163.4 であった. 特徴の記述量としては, 属性一つにつき 54 bit 要するので実装上では 7 byte 要する. よって画像一枚当たりの属性の記述量は平均は約 111 byte となる. 同様に関係に関しては, 実装上, 属性二つ分を加えて 17 byte 必要である. よって画像一枚当たりの関係の記述量の平均は約 2780 byte になる. 結局, 合計の記述量は 3 kbyte ほど必要となることになる.

以下に検索例を示す. 検索の流れやユーザの操作はシルエット画像の場合と同様である.

**検索例 1** 検索例を図 5.8 に示す. 検索要求は「赤と緑と白の三色で縦に等分に区切ってある国旗」である. 色の配置に関しては制限はないので, 水平方向の位置関連のゆらぎを最大にして, それ以外は最小にした. 閾値 0.9 で検索数 2 件であり, 検索洩れはなく, 余計な画像も検索されなかった.

**検索例 2** 検索例を図 5.9 に示す。検索要求は「検索例 1 において横に区切った場合の画像」である。今度は鉛直方向の位置関連のゆらぎを最大にした。閾値 0.9 で検索数 3 件であった。等分という条件の上では検索洩れはなかった。

**検索例 3** 検索例を図 5.10 に示す。検索要求は「検索例 2 同様であるが、等分でなくおおまかに区切ってある画像」である。閾値 0.9 で色・傾き属性と包含・角度・色関係のゆらぎを最小にして他は最大にした。この場合 22 件が検索された。多いので凹凸度以外の形状属性のゆらぎを除々に小さくして検索数が 10 件以内になるまで調整した。最終的な結果は図 5.10 の通りである。2 件余計な画像が含まれていた。検索洩れはなかった。

**検索例 4** 検索例を図 5.11 と図 5.12 に示す。検索要求は「大きく右方向に斜めに区切ってある国旗」である。まず、閾値 0.8 で色関連のゆらぎを最大にして、その他は標準の値にした。102 件検索された。絞り込むために他のゆらぎを全て最小にすると 2 件検索された (図 5.11)。他にも該当する国旗があったと思われたので、水平方向の位置関連を標準に戻したが結果は変わらなかった。そこで鉛直方向と大きさに関するゆらぎを標準に戻すと 8 件検索された (図 5.12)。検索洩れは図 5.12 (d) であり、中央に竜のマークがあるために凹凸が生じたために検索できなかった。余計な検索は 8 番の 1 件であった。

**検索例 5** 検索例を図 5.13 に示す。検索要求は「検索例 4 同様だが斜めの区切りの方向は問わない」である。検索例 4 の状態から始めて、斜めの方向はこだわらないので傾きのゆらぎを最大にする。さらに位置関係のゆらぎも全て最大にする。この結果 10 件検索された。余計なものは 1 件で検索洩れ 2 件であった。検索洩れの原因は検索例 4 と同じである。

**検索例 6** 検索例を図 5.14 に示す。検索要求は「左端に三角形の形状を持つ国旗」である。色属性以外のゆらぎはある程度小さくしてみたところ 50 件検索された。そこで、位置と大きさのゆらぎをさらに小さくすると 22 件検索された。このような国旗は多くあったと思われたので、この状態でブラウジングした。7 件余計なものが含まれていた。検索洩れは 3 件であった。これは凸形に関する特徴を扱っていないのが原因である。

**検索例 7** 検索例を図 5.15, 5.16, 5.17 に示す。検索要求は「三つの均等の横縞で真中の横縞内に模様がある国旗」である。図中、オブジェクトが色付けしてあるが、これは選択オブジェクトを明示するためのものであり特に意味はない。中央の模様は円を描画することで表現した。図のように縞のオブジェクトと中の模様のオブジェクト個々に属性のゆらぎを設定し、多少繁雑ではあるが関係も個々に設定した。この結果 12 件検索された。2, 4, 12 番の 3 件が余計なものとして含まれた。検索洩れは図 5.17 (i) 中の 25, 37, 78 番の 3 件である。他のは等分の縞でなく、さらに真中の模様がはみ出ている。

実験的に、図 5.17 に示した国旗全てを検索できるかを試した。22 件検索することを目指すわけである。何回かの試行錯誤を繰り返した結果、図 5.18, 5.19 のようになった。29 件検索して、そのうち 19 件が正解であった。10 件が余計であり、3 件が洩れた (図 5.19 (f))。

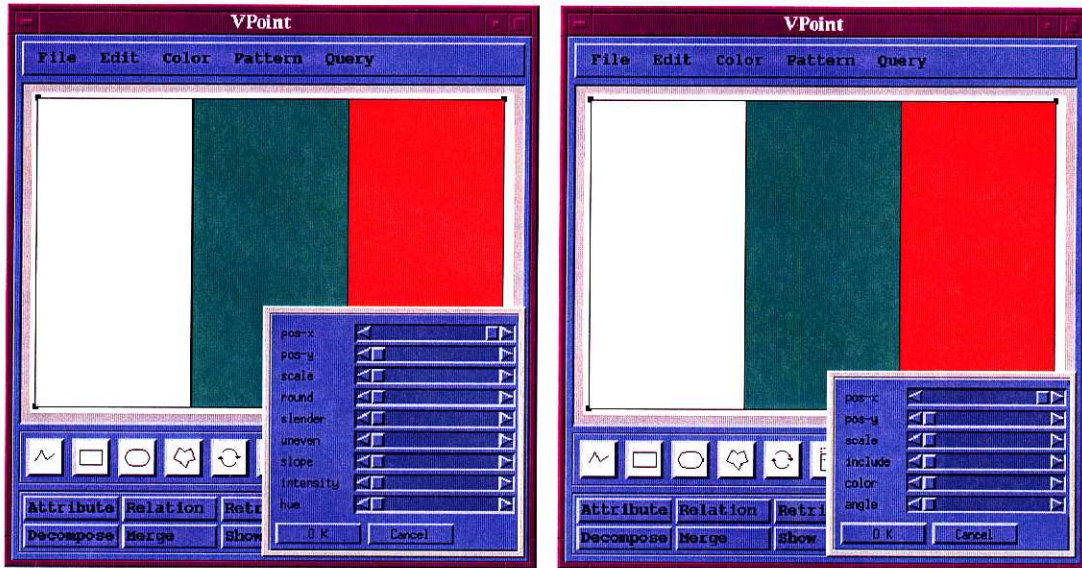


これら3件は、中央の模様が小さくかつ背景と類似した色であったためにプリミティブとして抽出できなかったのである。

**検索例 8** 検索例を図 5.20 に示す。検索要求は「緑地に白の三日月がある国旗」である。緑地の大きさや形状は不明であるが、だいたい長方形であることを想定して緑地の条件を設定して(図 5.20 (a))、月の色と形状のゆらぎを小さくした(図 5.20 (b))。関係に関しては包含と色のゆらぎを最小にして他は最大にした。検索結果は4件であり、余計なものなく検索洩れもなかった。

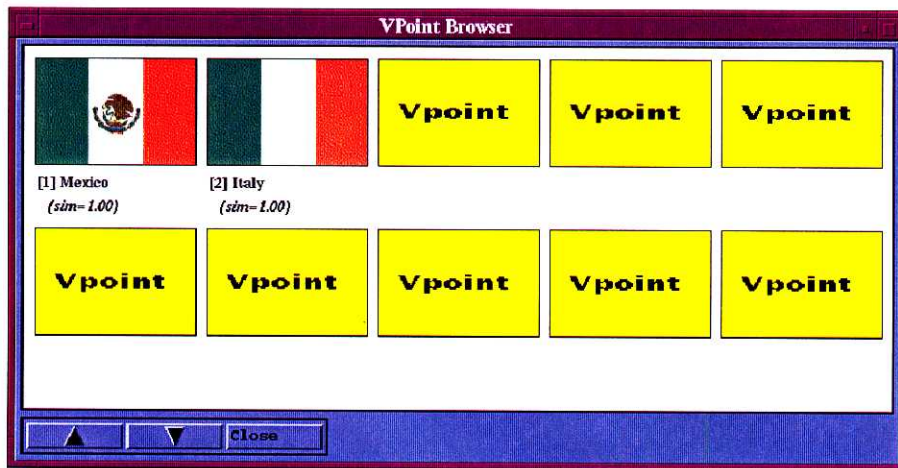
**検索例 9** 検索例を図 5.21 に示す。検索要求は「水色の地に盾のマークがある国旗」である。まず水色の地の属性はすべてゆらぎを最小にし、盾の属性は位置と色のゆらぎを最大にして、あとは標準のままにしておいた。関係のゆらぎは包含のみを最小にし、その他は最大に設定した。この結果25件検索された。絞り込むために盾の属性のゆらぎを図 5.21 (a) のようにした。この結果3件まで絞れたのでブラウズしてみたが該当するものがなかった。そこで図 5.21 (b) のように水色の地の色属性のゆらぎを一段階緩めるとさらに2件追加された。ブラウズしてみると合致する画像が1件あり、他の検索洩れはなかった。

**検索例 10** 検索例を図 5.22 に示す。検索要求は「円の中に星形のマークがある画像」である。一回目の設定の検索で4件検索された。余計なものは1件で検索洩れはなかった。



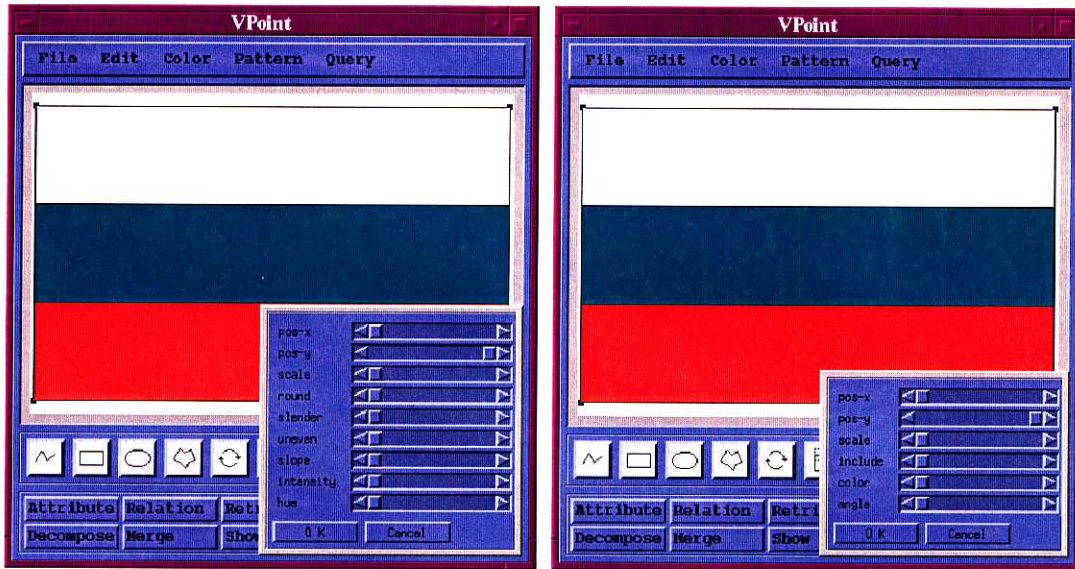
(a)

(b)



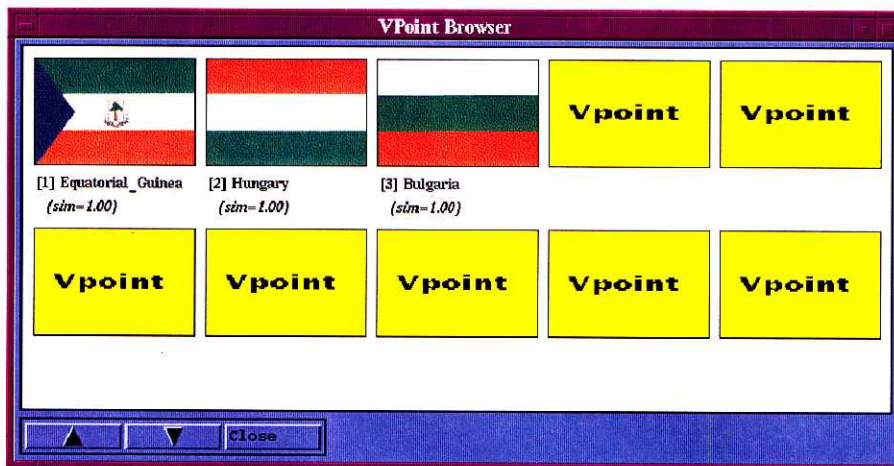
(c)

図 5.8 国旗画像検索例 (1)



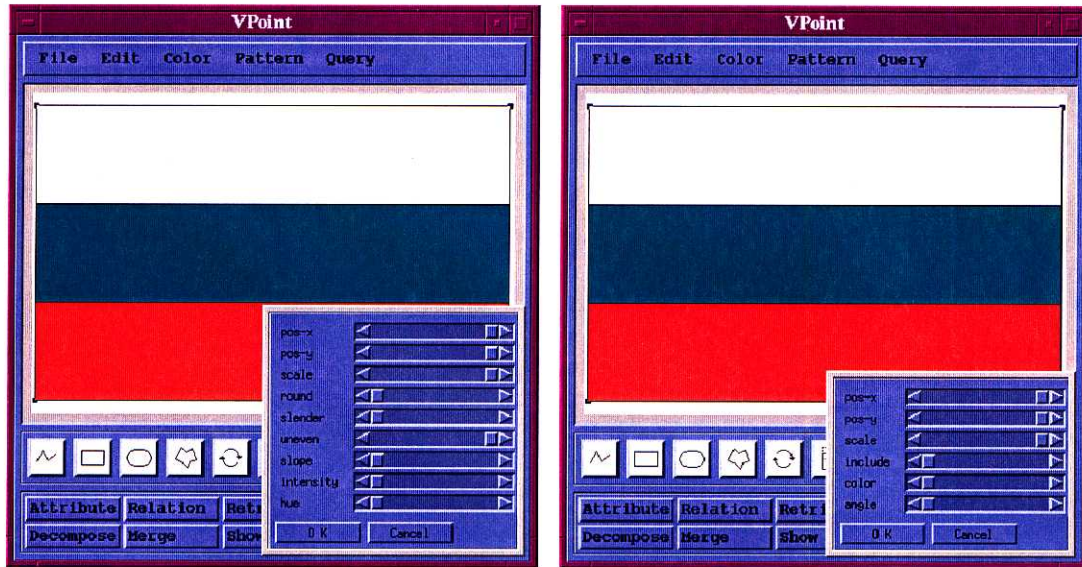
(a)

(b)



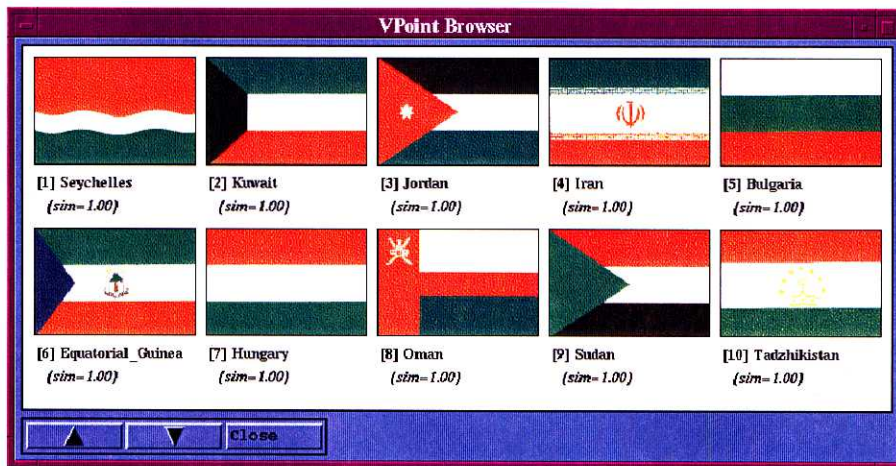
(c)

図 5.9 国旗画像検索例 (2)



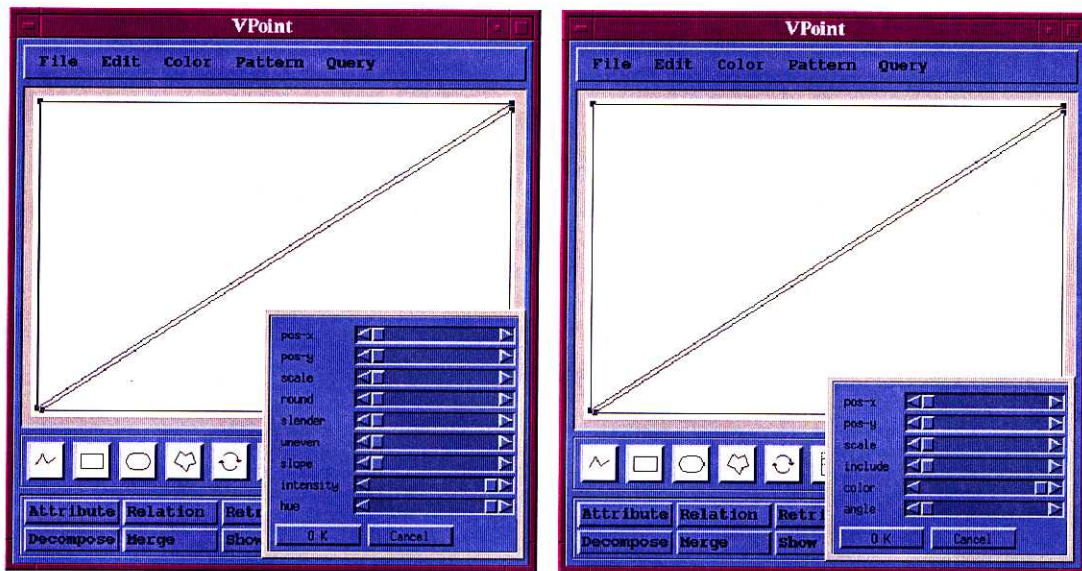
(a)

(b)



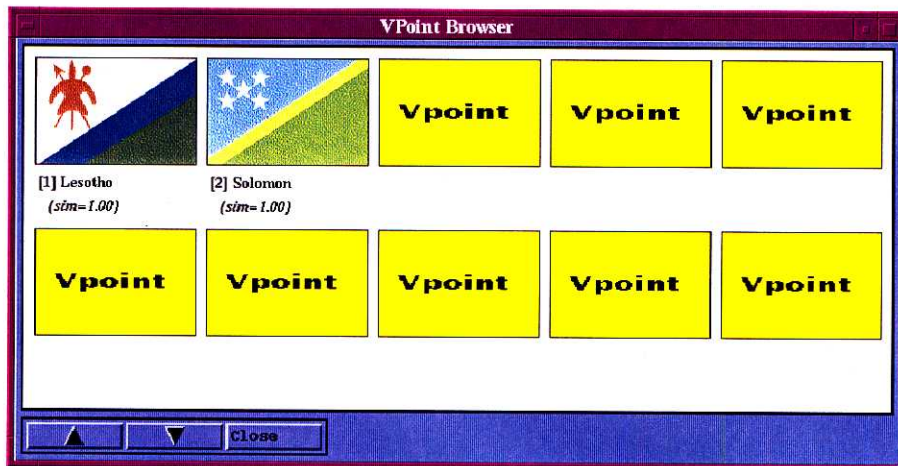
(c)

図 5.10 国旗画像検索例 (3)



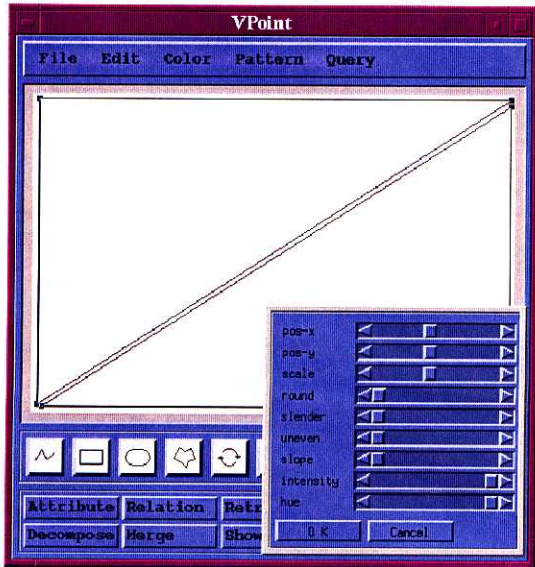
(a)

(b)

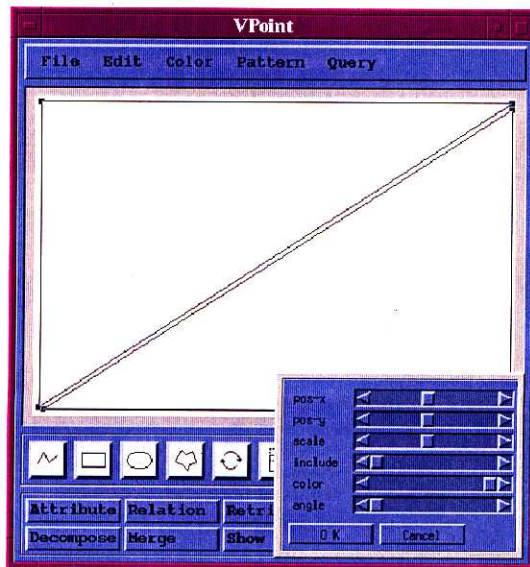


(c)

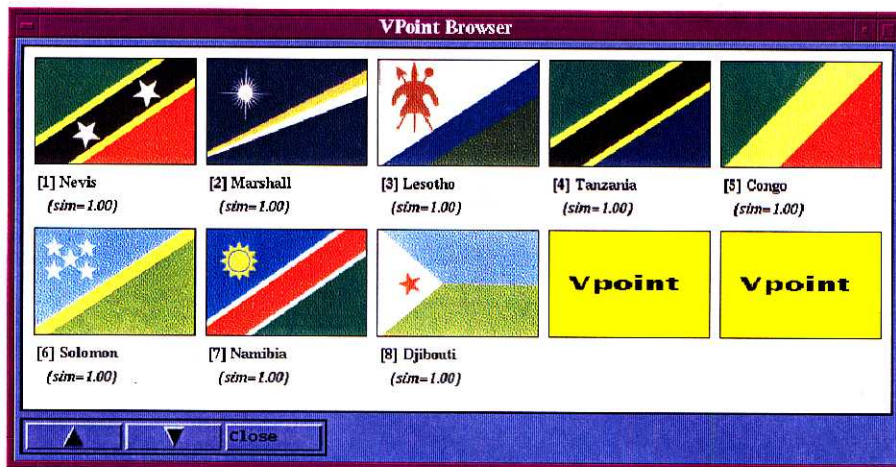
図 5.11 国旗画像検索例 (4a)



(a)



(b)



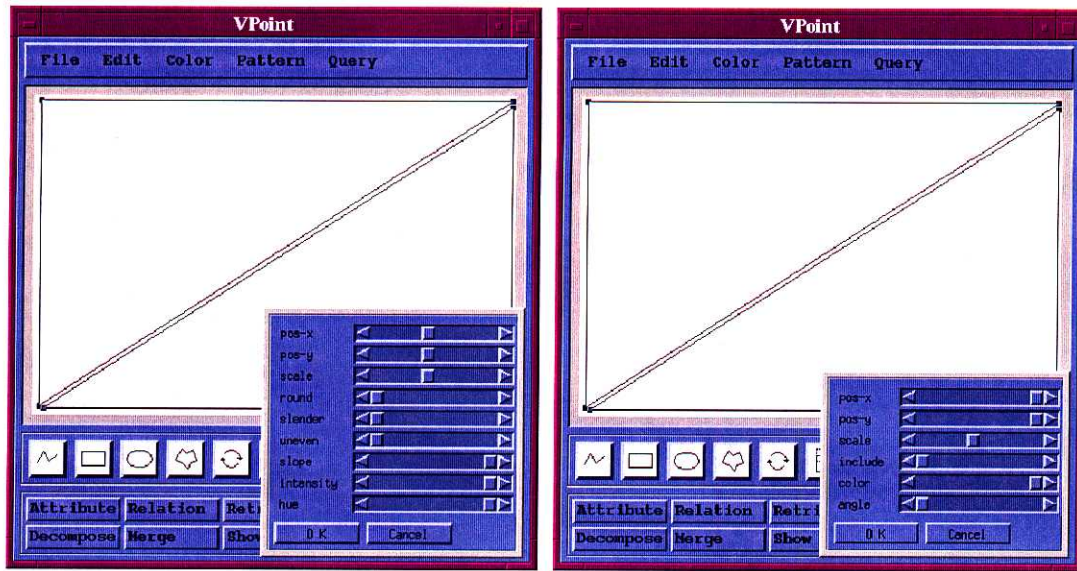
(c)



[29] Bhutan  
(sim=0.00)

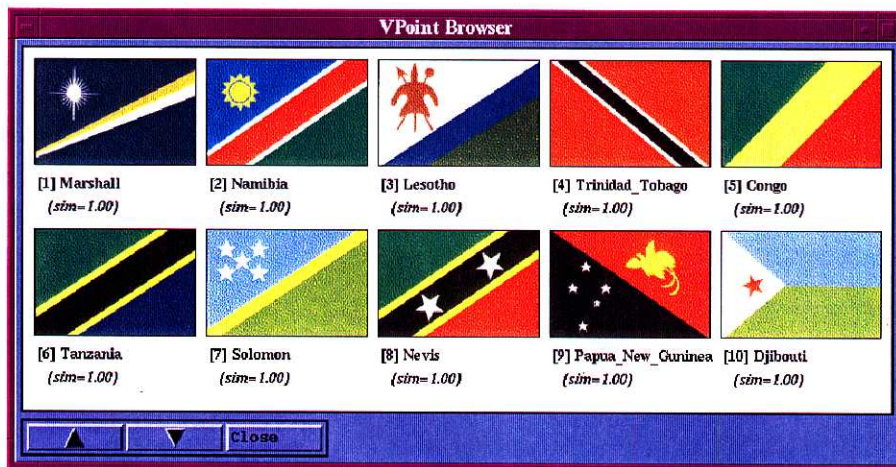
(d)

図 5.12 国旗画像検索例 (4b)

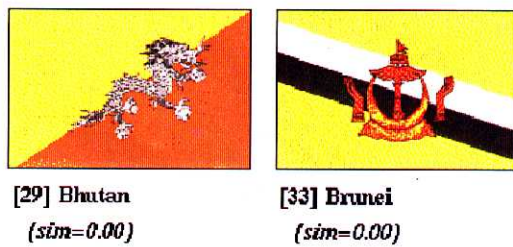


(a)

(b)



(c)



(d)

図 5.13 国旗画像検索例 (5)

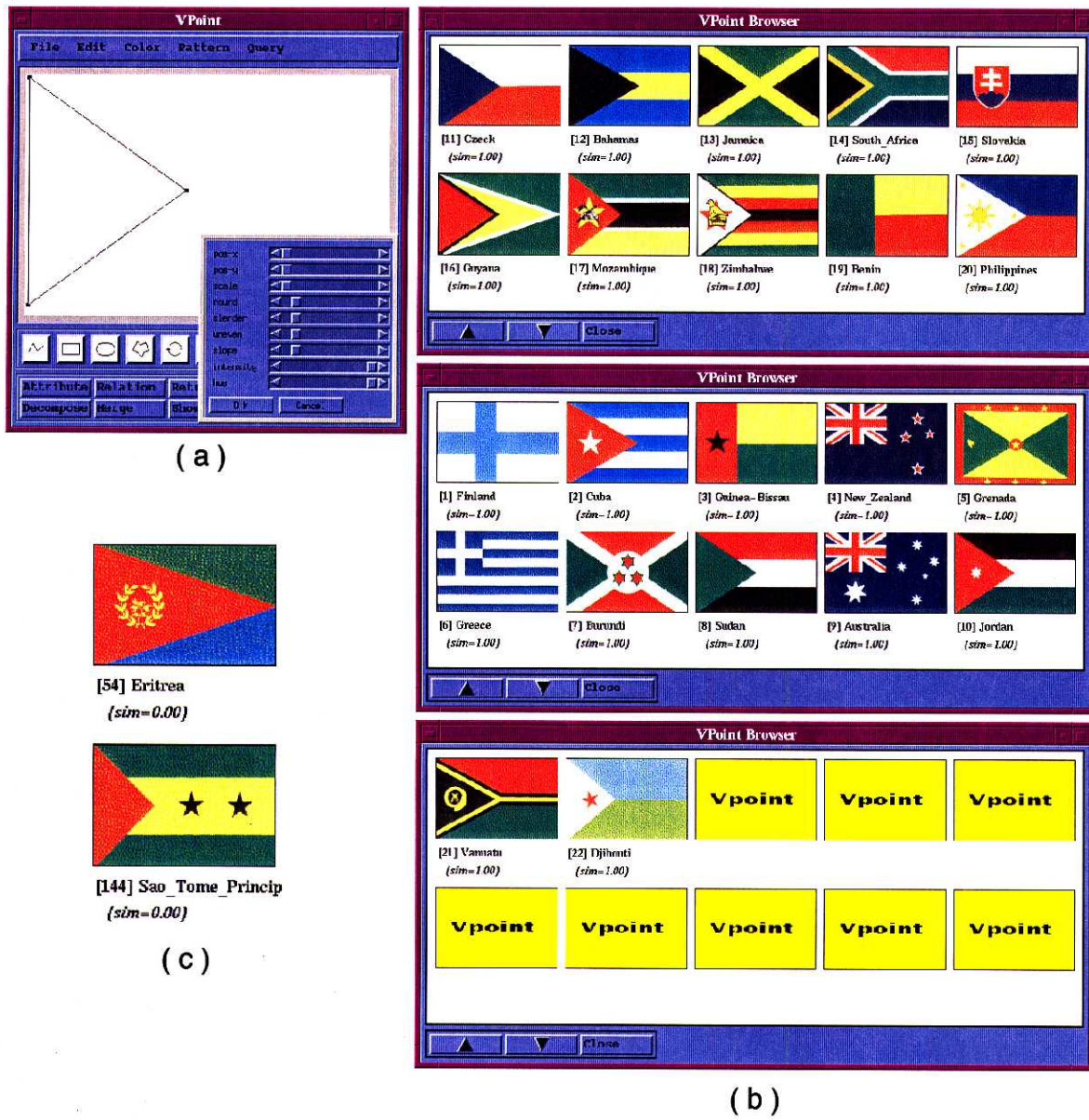
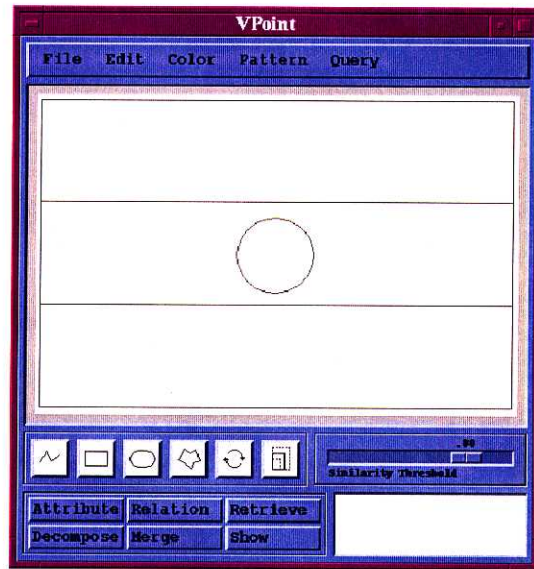
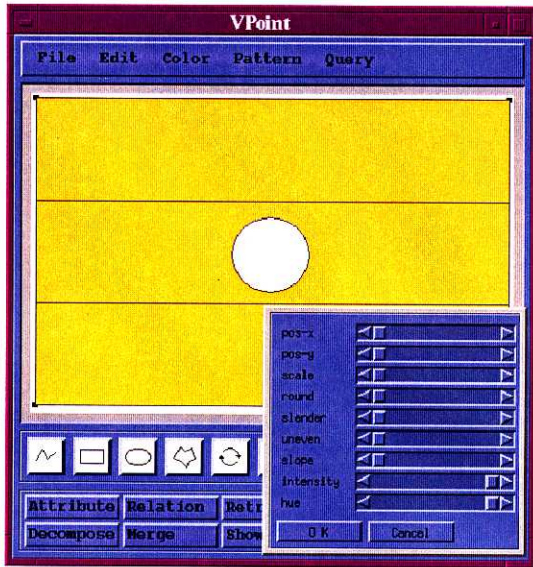


図 5.14 国旗画像検索例 (6)

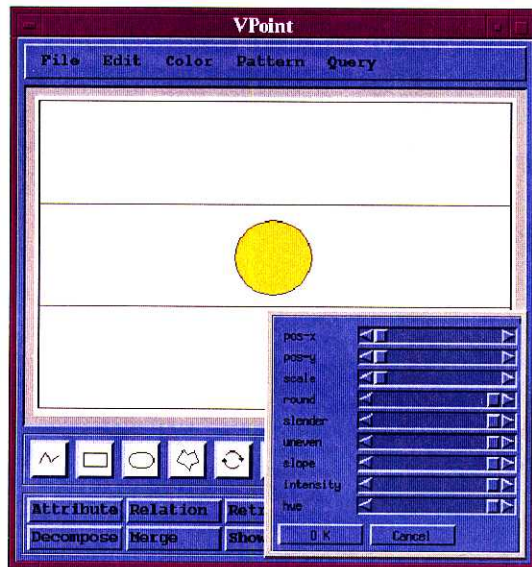




(a)

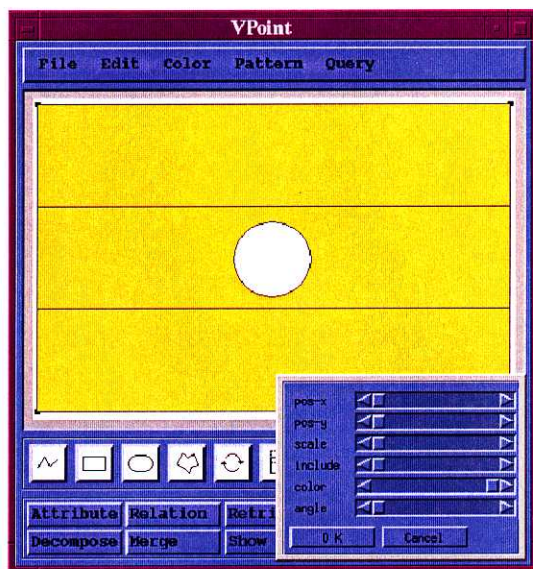


(b)

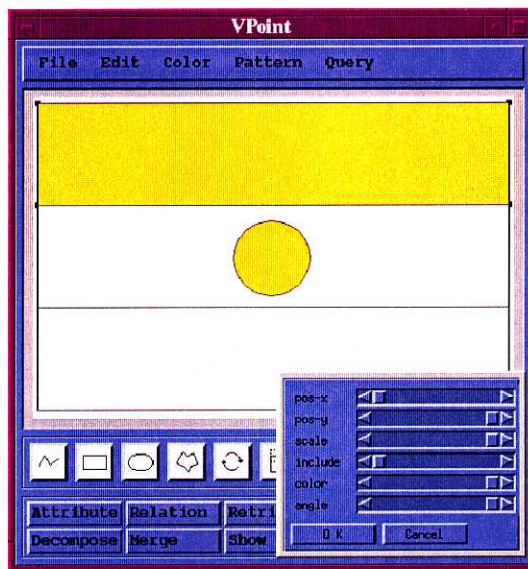


(c)

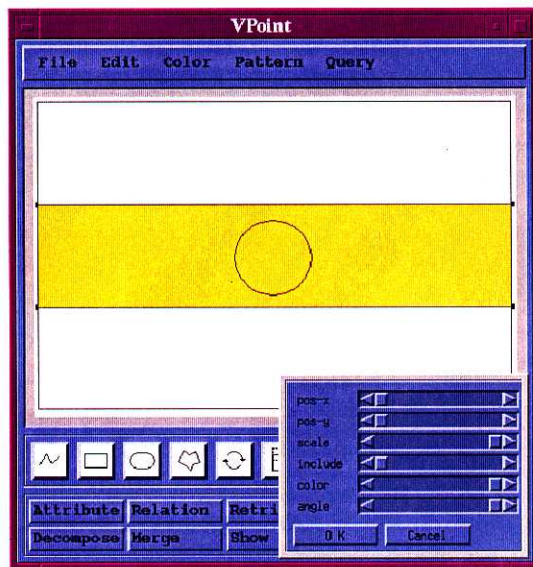
図 5.15 国旗画像検索例 (7a)



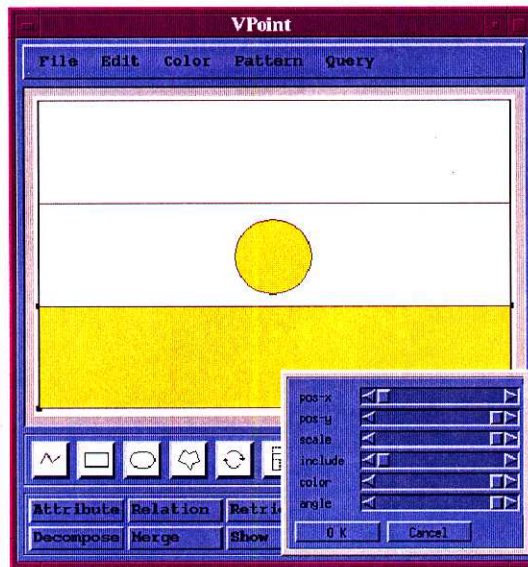
(d)



(e)



(f)



(g)

図 5.16 国旗画像検索例 (7b)

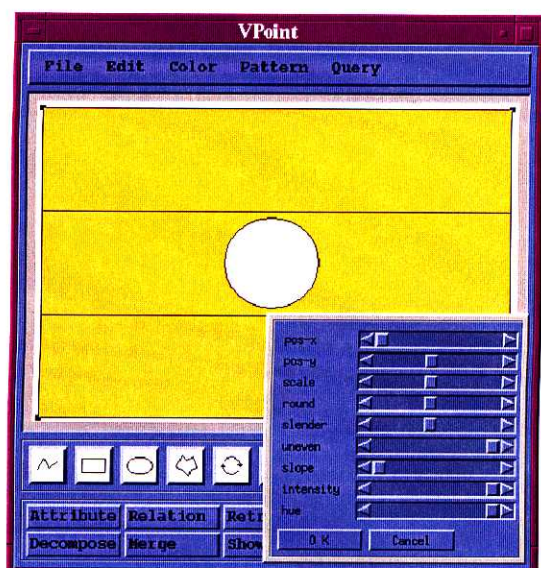


(h)

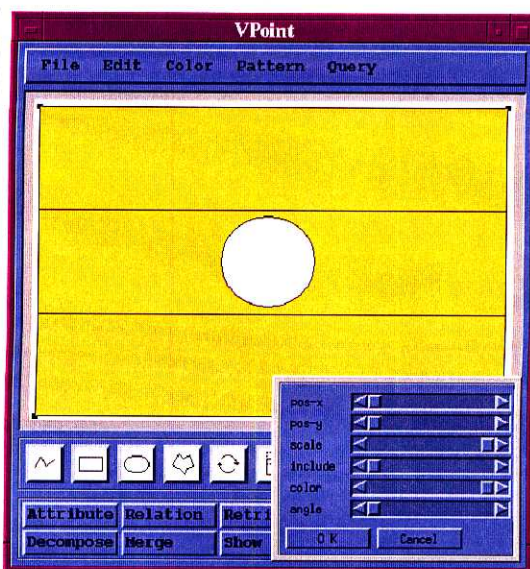


(i)

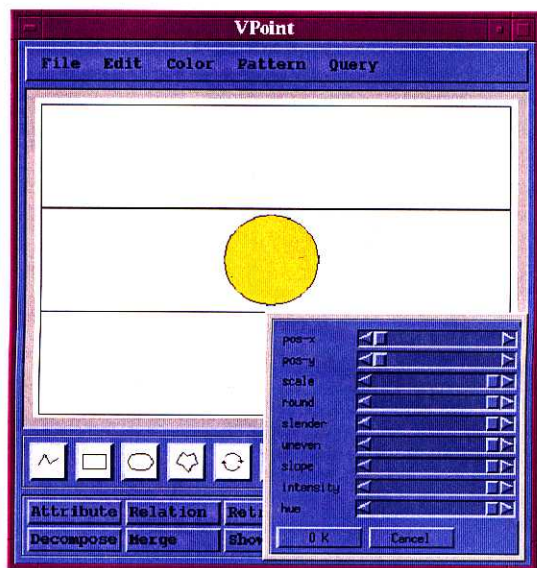
図 5.17 国旗画像検索例 (7c)



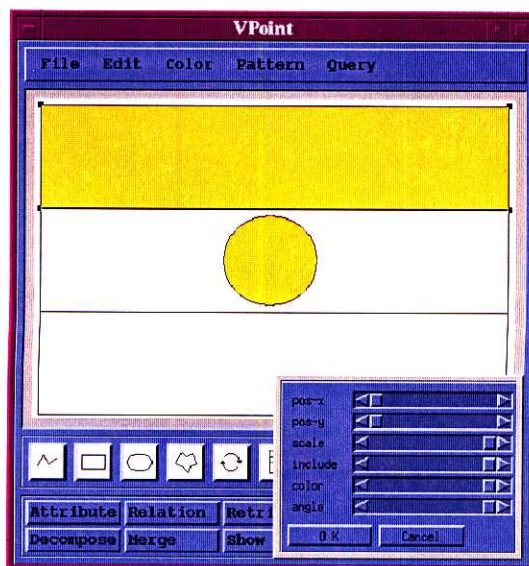
(a)



(b)

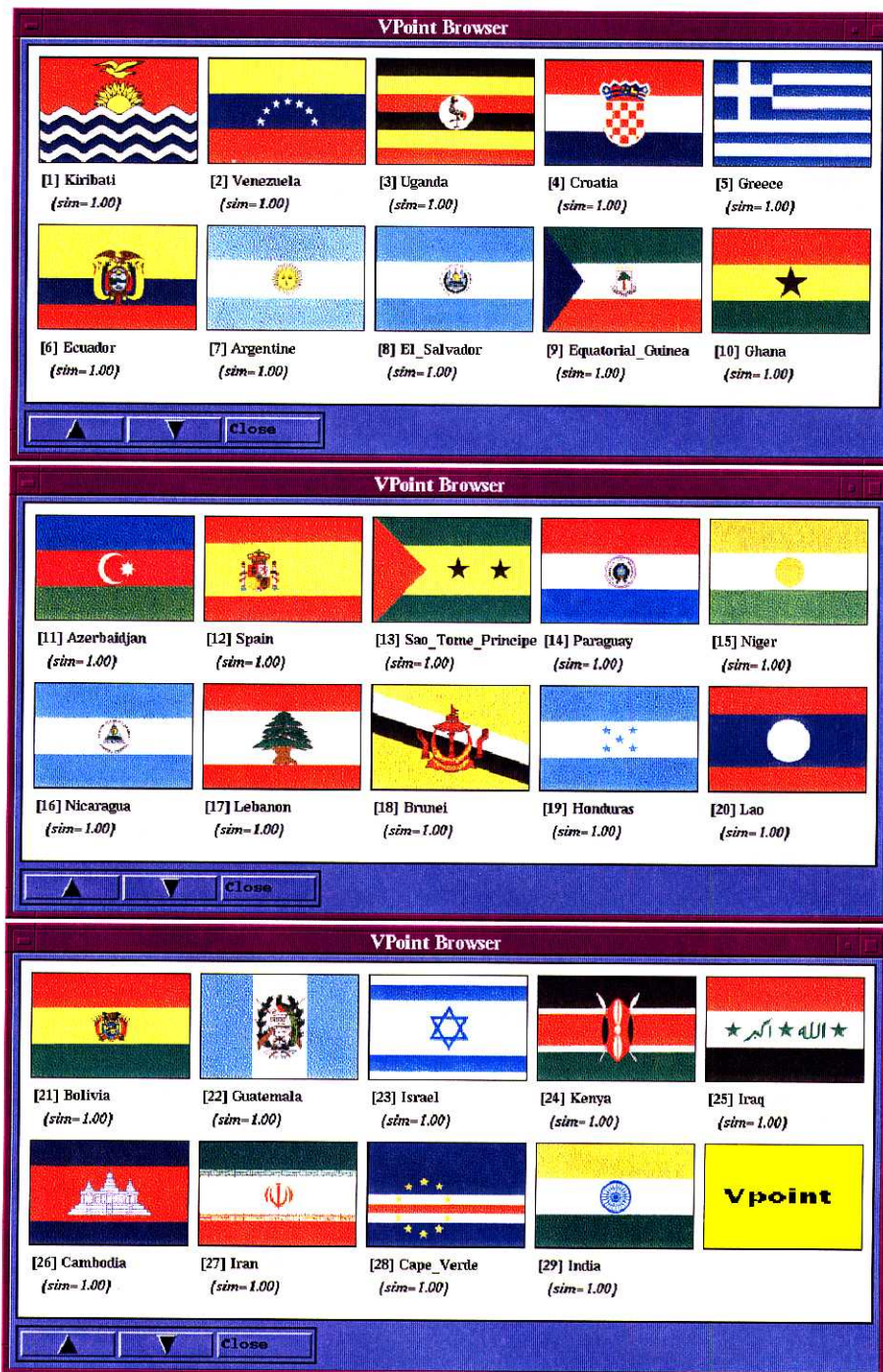


(c)



(d)

図 5.18 国旗画像検索例 (7d)

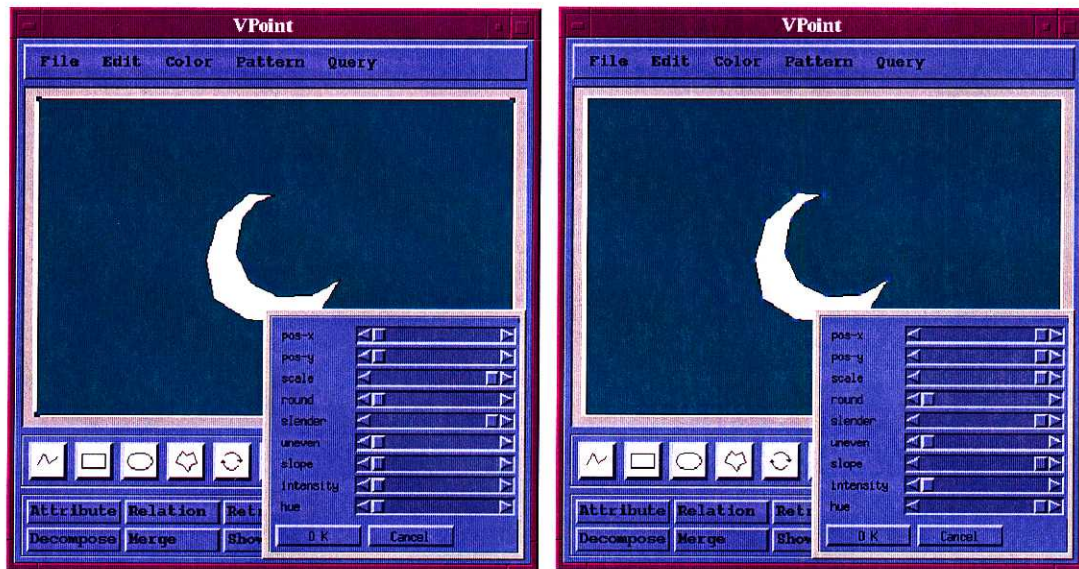


(e)



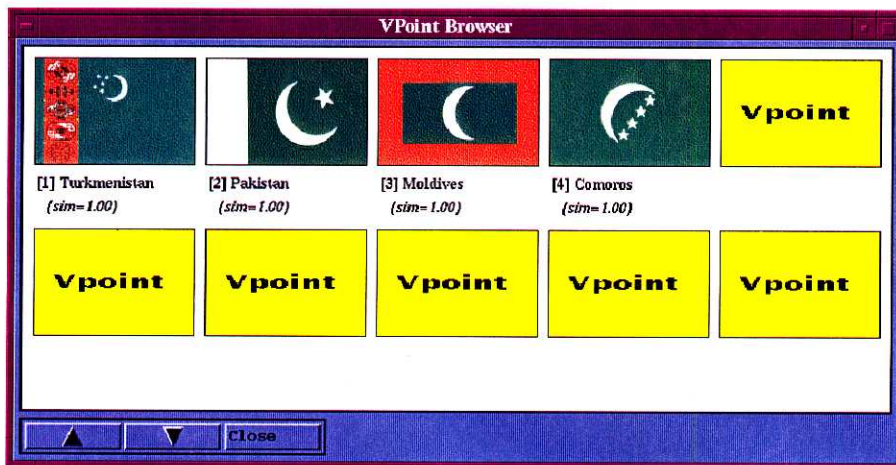
(f)

図 5.19 国旗画像検索例 (7e)



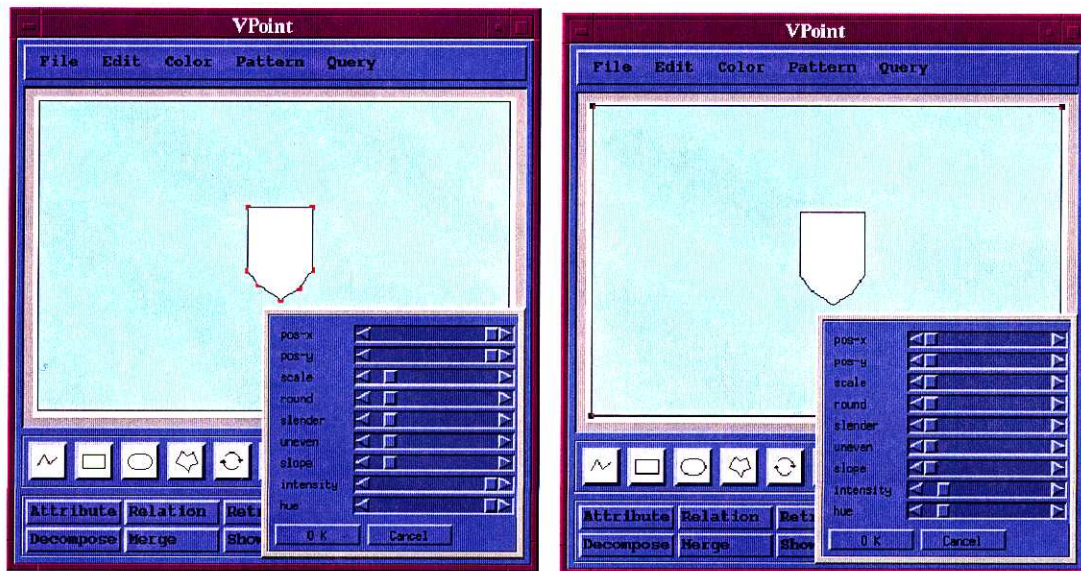
(a)

(b)



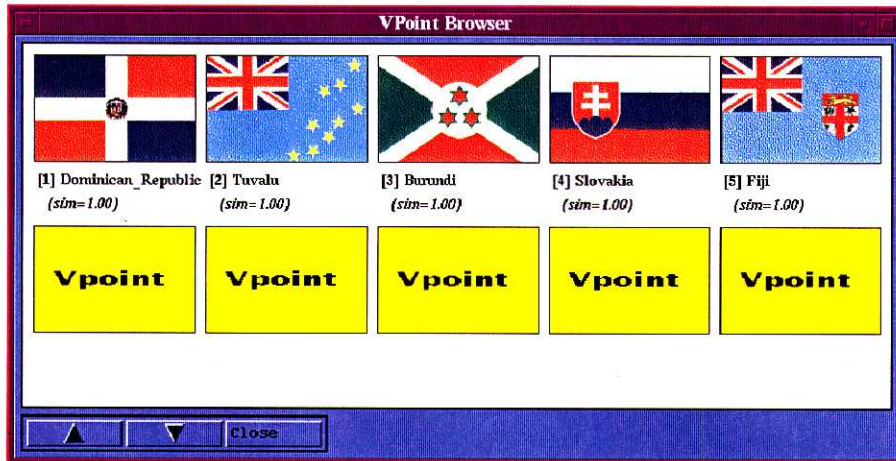
(c)

図 5.20 国旗画像検索例 (8)



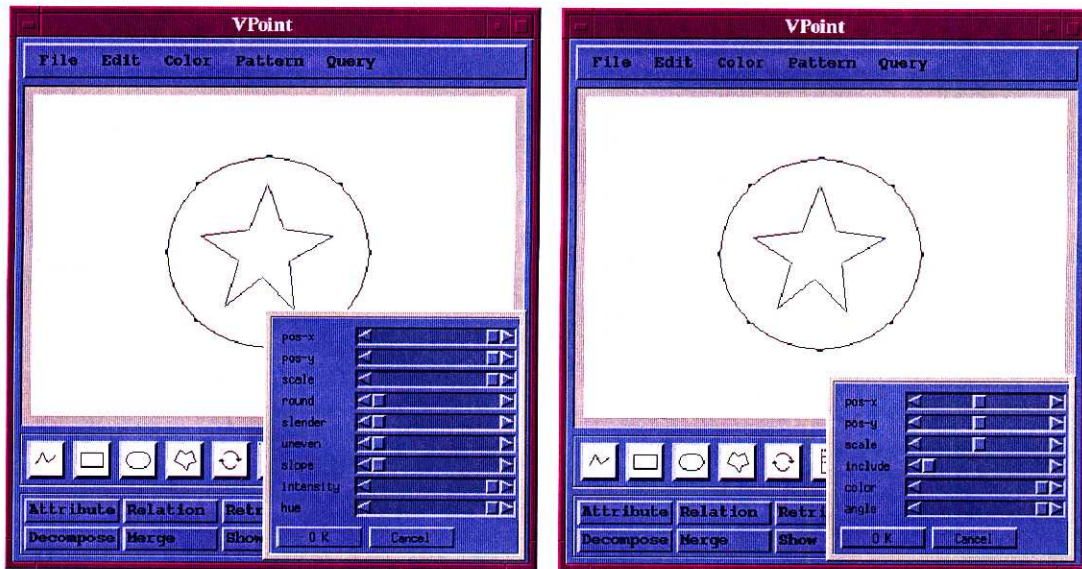
(a)

(b)



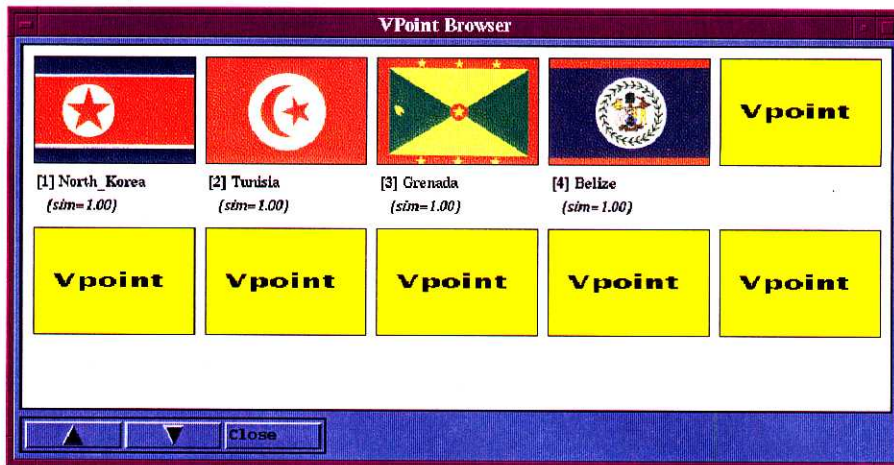
(c)

図 5.21 国旗画像検索例 (9)



(a)

(b)



(c)

図 5.22 国旗画像検索例 (10)



## 5.4 自然画を対象とした検索

### 5.4.1 プリミティブ抽出処理

扱う画像は、 $256 \times 256$  の 24bit の自然画像 300 枚である。プリミティブを抽出する際には、処理時間の短縮のために  $128 \times 128$  に縮小してから処理した。他はほぼ国旗画像同様に扱った。パラメータは以下の通りである。

$$\begin{aligned} \epsilon_s & 0.01 \\ \epsilon_r & 0.001 \\ \epsilon_c & 0.003 \\ \epsilon_p & 0.75 \end{aligned}$$

なお、形状分解におけるパラメータは  $t_{max}$  を 10 , Scale Space Filter のスケール値を 2 にした以外は国旗画像の場合と同様である。

以下に処理例を示す。

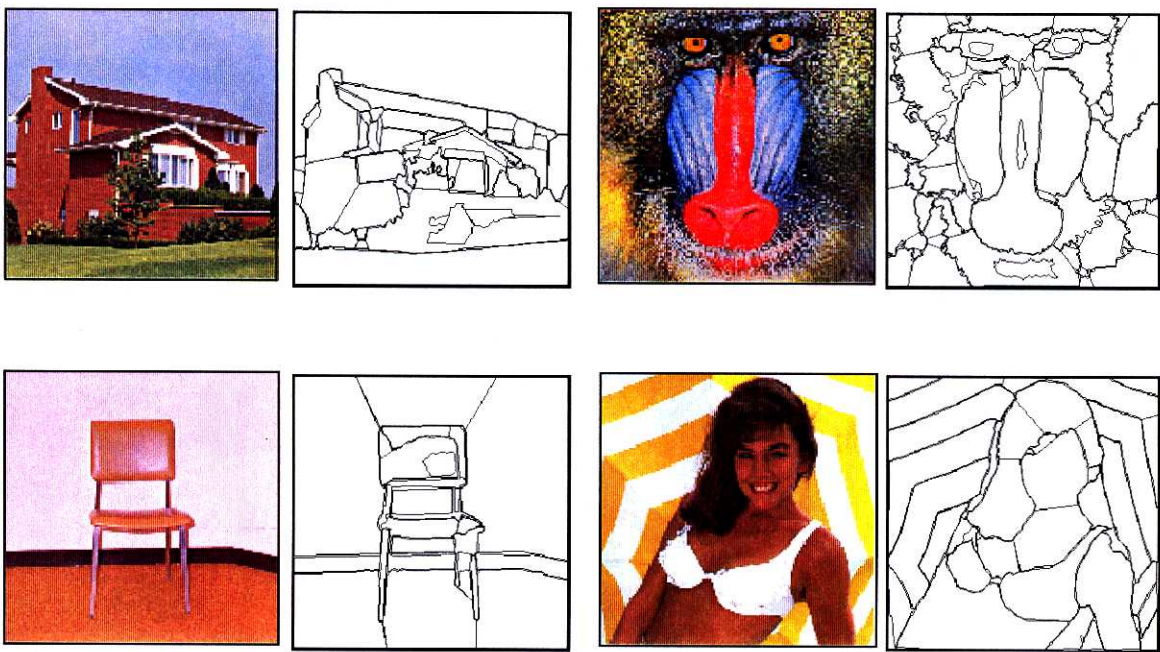


図 5.23 自然画のプリミティブ分割例

### 5.4.2 特徴記述

これも国旗画像の場合とほぼ同様であるが、凹凸度の属性と角度の関係は省いた。よって用いる特徴は 8 属性 46 ビット, 5 関係 14 ビットである。

国旗画像検索では、ある二つのプリミティブ間の関係は各々が第一プリミティブになるように二通りの記述を行ったが、自然画像の場合、冗長なプリミティブが多く、関係の数も膨大になる

ので、二通りの記述は行わないで、シルエット画像の場合の基準に従う。つまり、大きさが近いプリミティブの組み合わせのみ、二通りの関係を冗長に記述する。また、ここでも全ての領域間の関係を求めるわけにはいかないので、以下のいずれかに該当するプリミティブの組み合わせに関しては関係を結ばない。

- プリミティブの凸包の面積に比例した量だけ、そのプリミティブの凸包を膨張させて時に、互いに交わらないプリミティブの組み合わせ。
- merged primitive と decomposed primitive の組み合わせ。
- 包含関係ではなく、互いの領域の一部が重なり合っているプリミティブの組み合わせ。
- original primitive 同士または merged primitive と original primitive の組み合わせ以外で包含関係にある組み合わせ。

### 5.4.3 検索例

以上の処理を基に、構築した VPoint の自然画像を対象とした場合の検索例を示す。300 枚の自然画像に対して、出力されたプリミティブ（属性）の数は平均 52.5，同じく属性の数も 52.5，関係の数は平均 481.8 であった。特徴の記述量としては、属性一つにつき 46 bit 要するので実装上では 6 byte 要する。よって画像一枚当たりの属性の記述量の平均は約 315.0 byte となる。同様に関係に関しては、実装上、属性二つ分を加えて 14 byte 必要である。よって画像一枚当たりの関係の記述量の平均は約 6745.2 byte になる。結局、合計の記述量は 7 kbyte ほど必要となることになる。

以下に検索例を示す。検索の流れやユーザの操作はシルエット画像や国旗画像の場合と同様であるので、ここでは操作に関する説明は簡略化する。

**検索例 1** 検索例を図 5.24 に示す。検索要求は風船を意識して「小さめのカラフルな丸いオブジェクトがいくつかある画像」である。位置関連のゆらぎを最大にして、大きさと色相の属性と、大きさと包含の関係のゆらぎを最小にした。検索数は 2 件であり所望の画像は含まれていた。

**検索例 2** 検索例を図 5.25 に示す。検索要求はヨットの画像の検索であり「上部の背景が青色で船体の色が黄色っぽい三角に近い帆を持つ画像、但し帆の色は不明」である。帆の色のみが不明なので、属性は分けて指示する。関係は標準のままである。この結果、検索数は閾値 0.9 で 5 件であったが、最上位 3 件のうちに所望の画像が含まれていた。

**検索例 3** 検索例を図 5.26 に示す。検索要求は SIDBA 画像 mandrill を検索するのを目的として「赤と水色の鼻を含む画像」である。これは明確なスケッチ検索なので、全てのゆらぎを小さくしたところ、1 件まで絞れ、所望の画像が得られてた。

**検索例 4** 検索例を図 5.27 に示す。これはナビゲーション検索の例であるが、シルエット画像や国旗画像の場合のようにプリミティブの集合としてインタフェース上に表示させてもよいが、自然画の場合、冗長なプリミティブが多過ぎるので、かえって操作が煩雑になる。そこで実画像を表示させて、ユーザが着目する箇所を多角形オブジェクトで指示させることにした。検索要求はクッキーの画像を起点として「丸いクッキーの上にチョコレート色のものがのっている画像」である。図 5.27 (a) の中で白い線の多角形が指示された領域である。計 4 箇所ある。色を重視して位置や大きさ属性のゆらぎは大きくした。この場合あまり絞り切れず、所望の画像は含まれているものの 10 件の検索数となった。

**検索例 5** 検索例を図 5.28 に示す。検索要求は「明るい緑色をきた肩より上の人の画像」である。何回かの試行錯誤の結果 4 件に絞り込んだ。

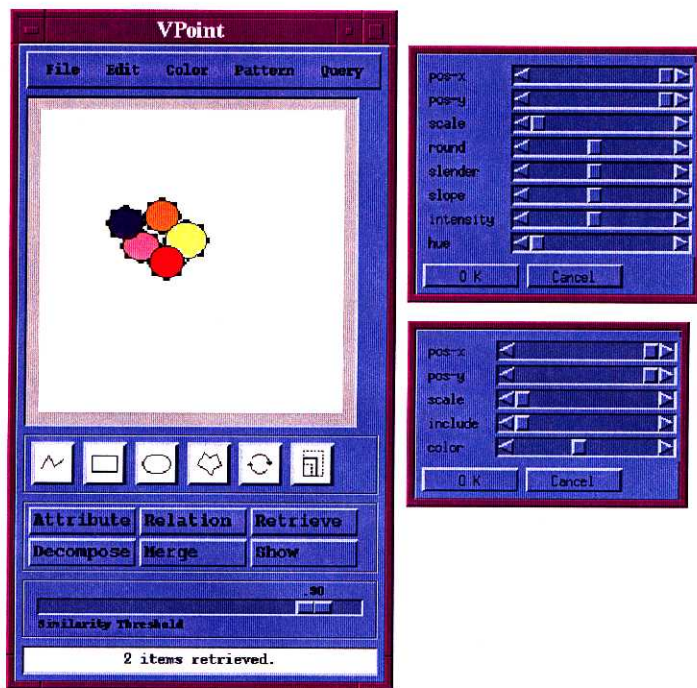
**検索例 6** 検索例を図 5.29, 5.30 に示す。これもナビゲーション検索である。検索要求は「野球の投手と打者の対決シーン」である。一枚の画像を起点として、着目する箇所を長方形領域で指示した。位置、大きさ、色はゆらぎを小さくして、形状関連は大きくまたは標準のままにした。関係は標準のままである。結局 4 件検索された。次に、打者の水平方向の位置に関する制約をなくしたところ、さらに 1 件追加され、左打者のシーンが検索された (図 5.30)。なお検索洩れはなかった。

**検索例 7** 検索例を図 5.31 に示す。検索要求は「時代劇のちょんまげ頭の人物が登場しているシーン」である。ちょんまげ頭を想定してスケッチを行ったが、検索できたのは 1 件のみであった。これは原画像処理に起因するものである。

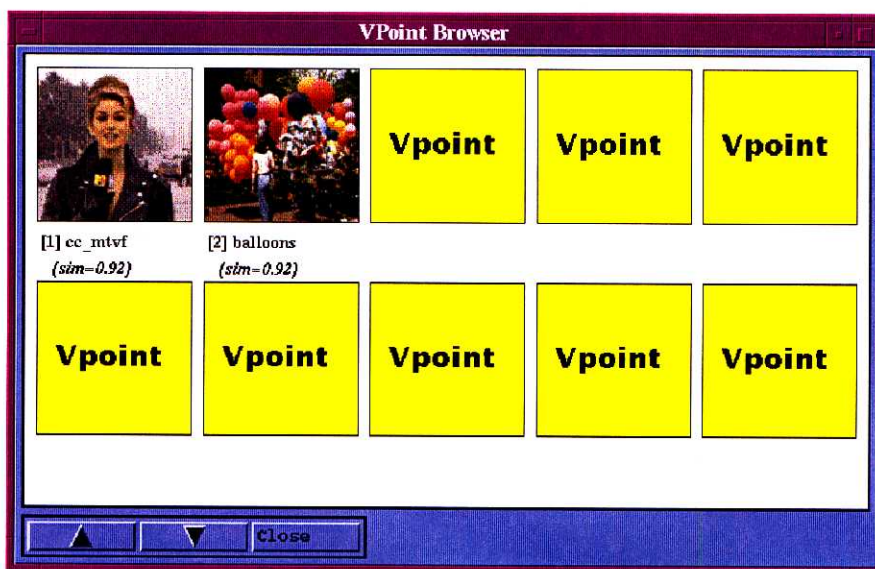
**検索例 8** 検索例を図 5.32 に示す。検索要求は魚の画像を起点として「同じ角度で撮影された魚の画像」である。背景の中央付近に楕円に近い物体が映っていることを想定して検索条件を設定した。色は不明なのでゆらぎを最大にしたが、背景はおそらく暗いということで、輝度に関しては小さくした。最終的に 9 件に絞り込み、2 件が洩れた。

**検索例 9** 検索例を図 5.33 に示す。検索要求は「白のビキニの女性」である。位置や大きさに関しては、あまり条件を厳しくせず、色と関係を厳しくして検索してみた。何回かの試行のうち 4 件に絞り込んだ。検索洩れは 3 件である。この原因は、原画像処理の失敗 (18 番の画像)、肌の色の相違 (69 番の画像)、ビキニのオクルージョン (75 番の画像) などが挙げられる。

**検索例 10** 検索例を図 5.34 に示す。検索要求は、ゴルフのパターシーンを意識して「薄い明るい芝を背景にゴルファーが立っている画像」である。人物は簡単に楕円で表現し、背景と楕円を個々に条件を設定した。検索数は 6 件で 2 件洩れた。

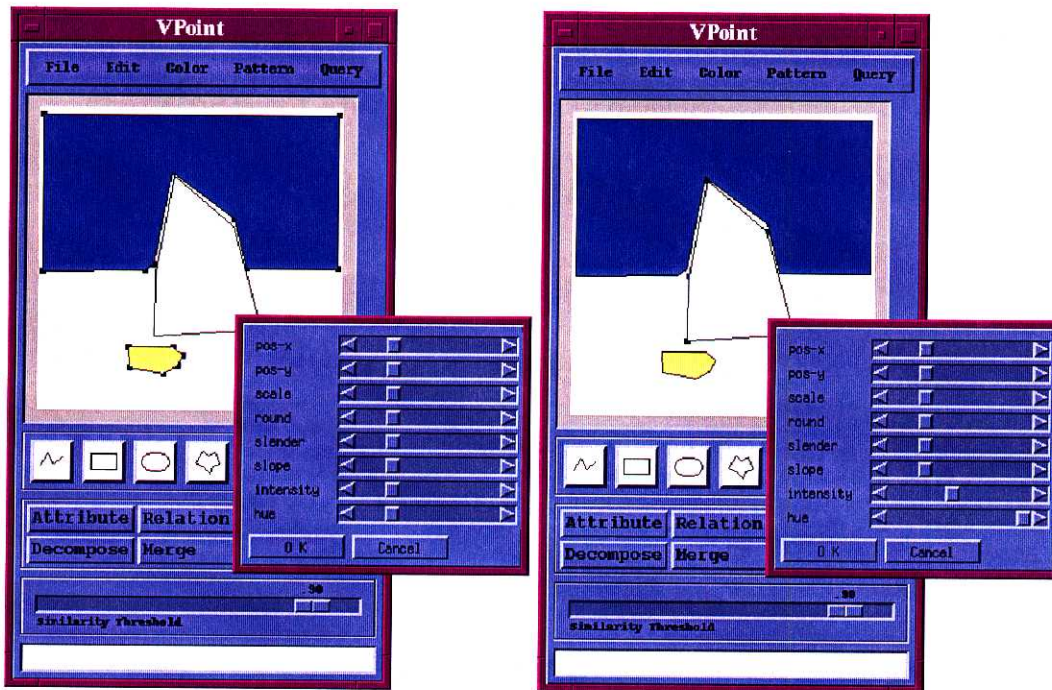


(a)



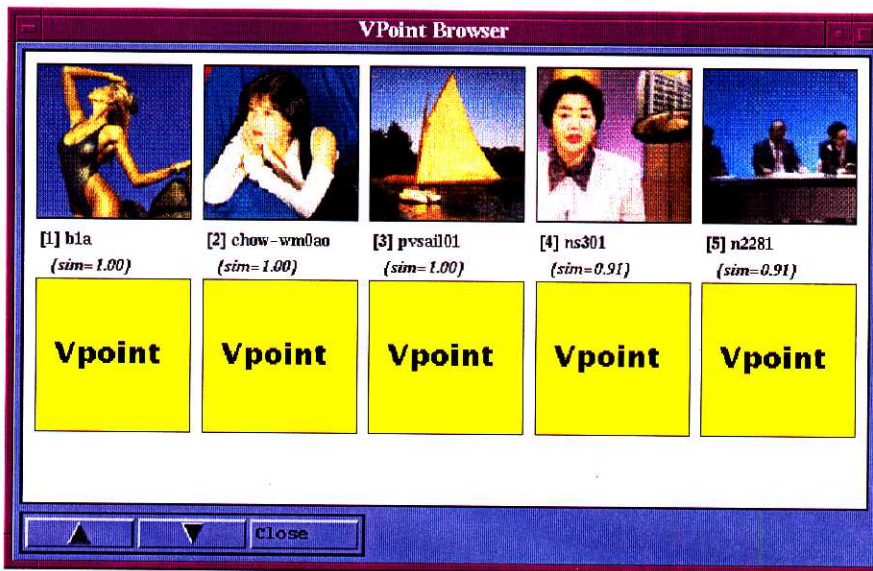
(b)

図 5.24 自然画像検索例 (1)



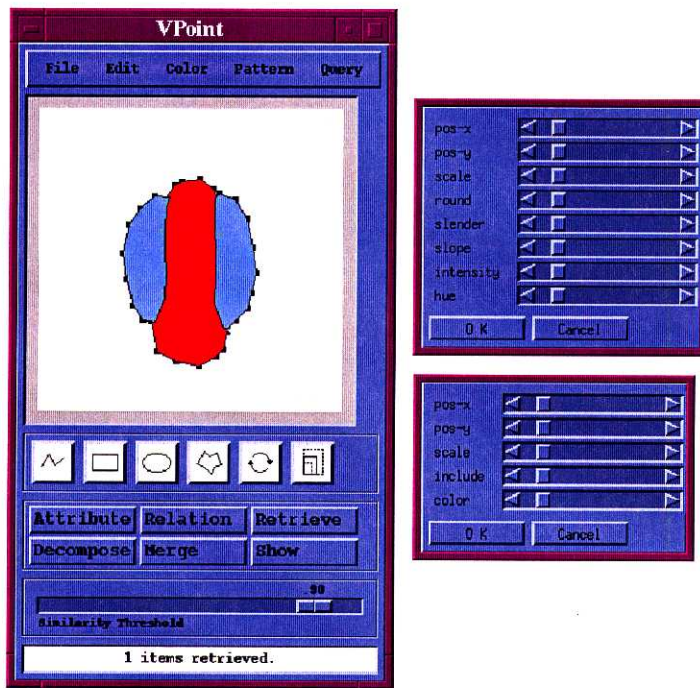
(a)

(b)

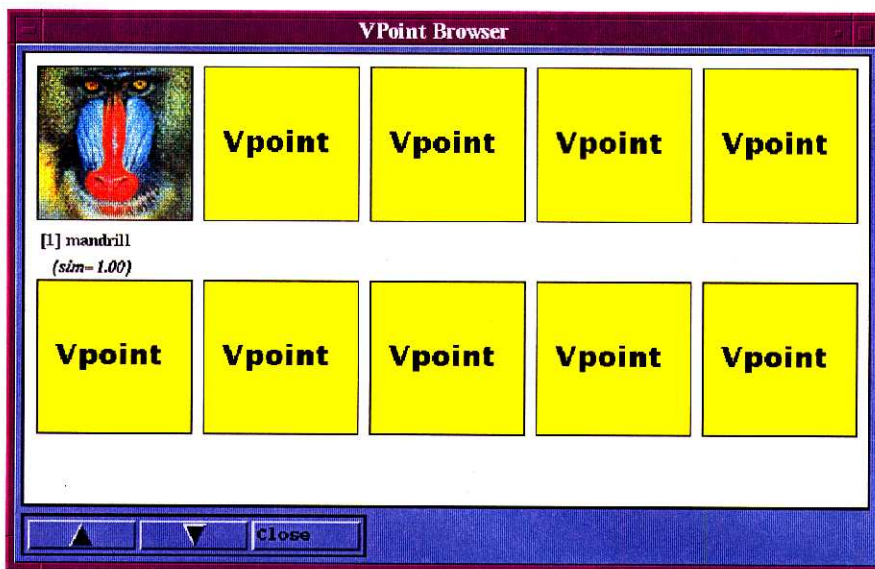


(c)

図 5.25 自然画像検索例 (2)

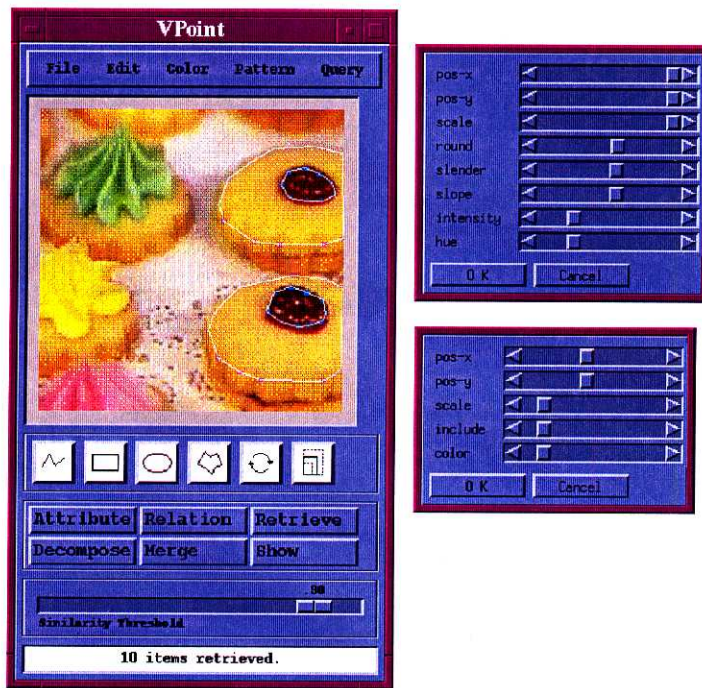


(a)



(b)

図 5.26 自然画像検索例 (3)

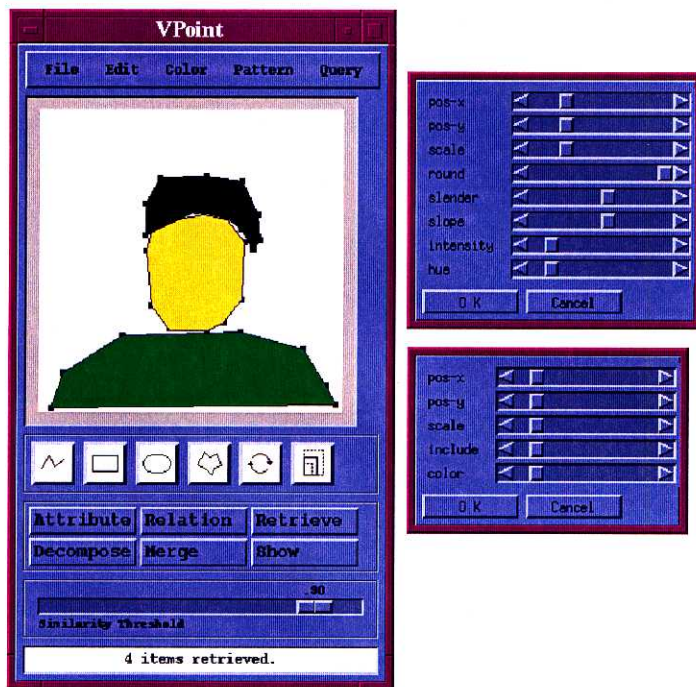


(a)



(c)

図 5.27 自然画像検索例 (4)



(a)



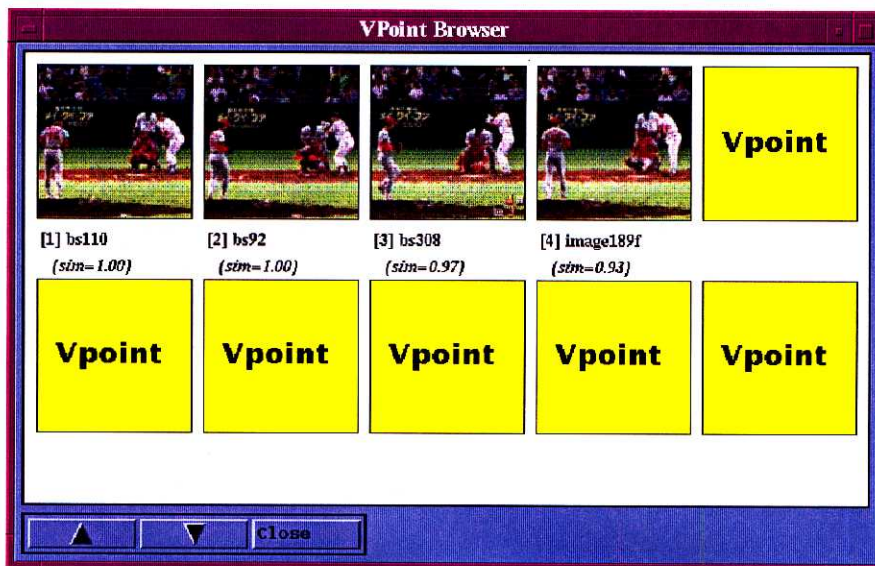
(b)

図 5.28 自然画像検索例 (5)





(a)

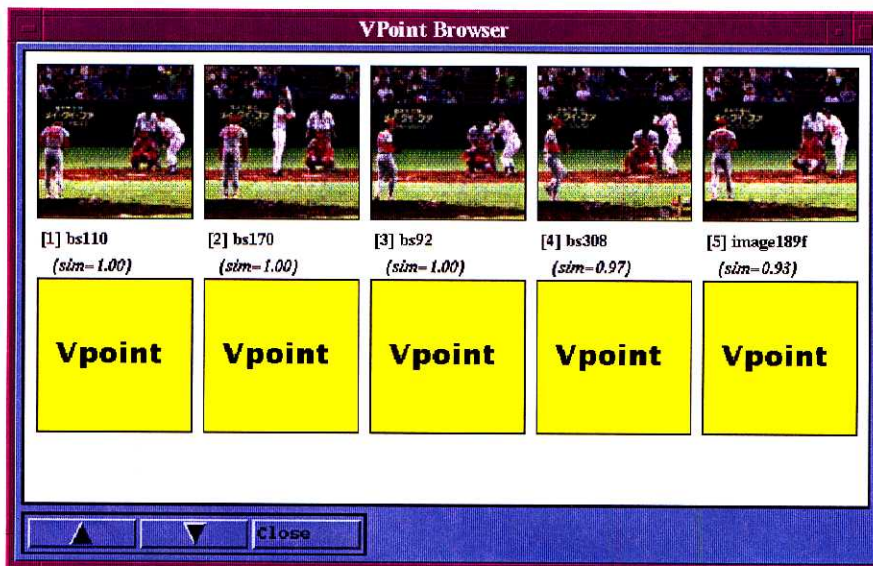


(b)

図 5.29 自然画像検索例 (6a)

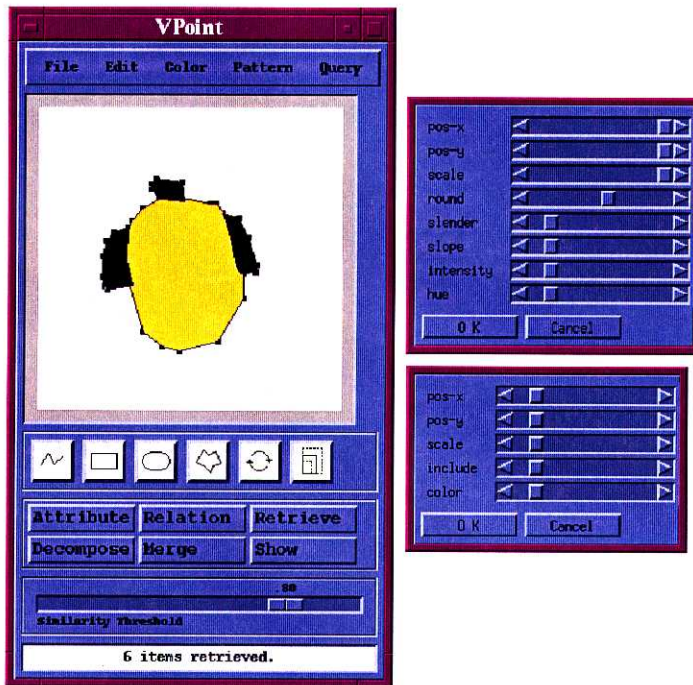


(a)

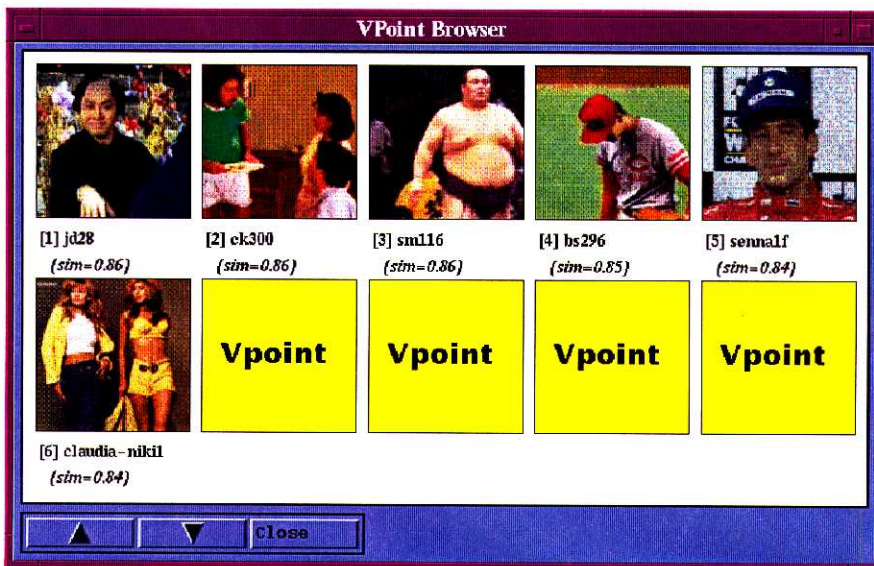


(b)

図 5.30 自然画像検索例 (6b)



(a)



(b)

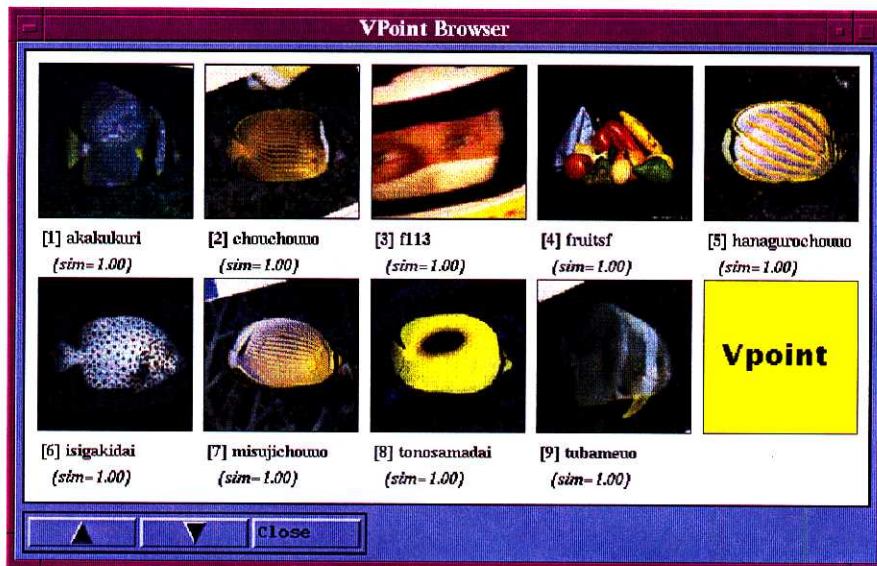


(c)

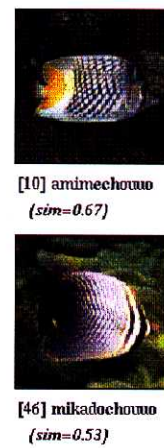
図 5.31 自然画像検索例 (7)



(a)

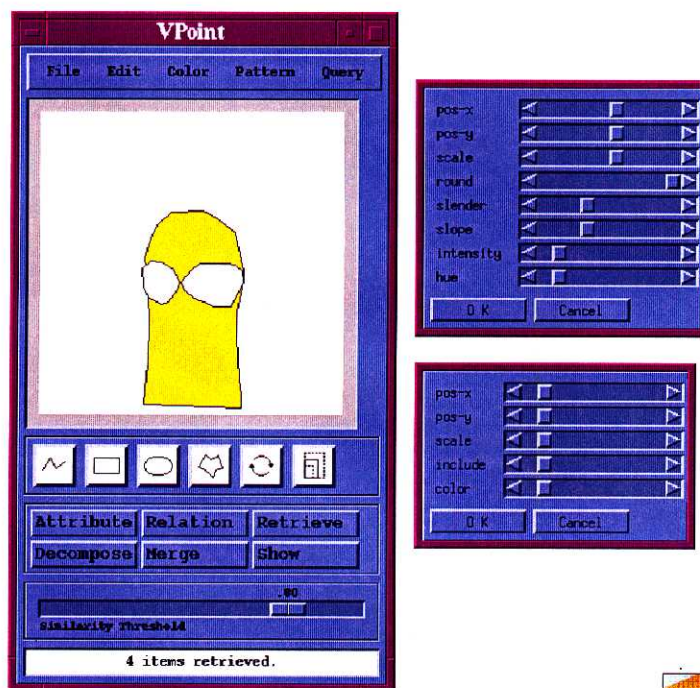


(b)

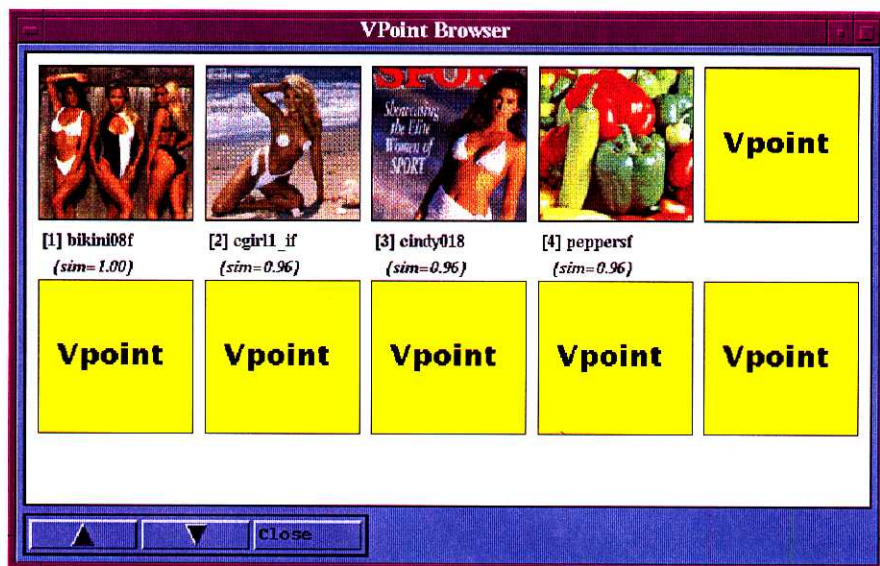


(c)

図 5.32 自然画像検索例 (8)



(a)



(b)



[18] bikini02  
(sim=0.54)



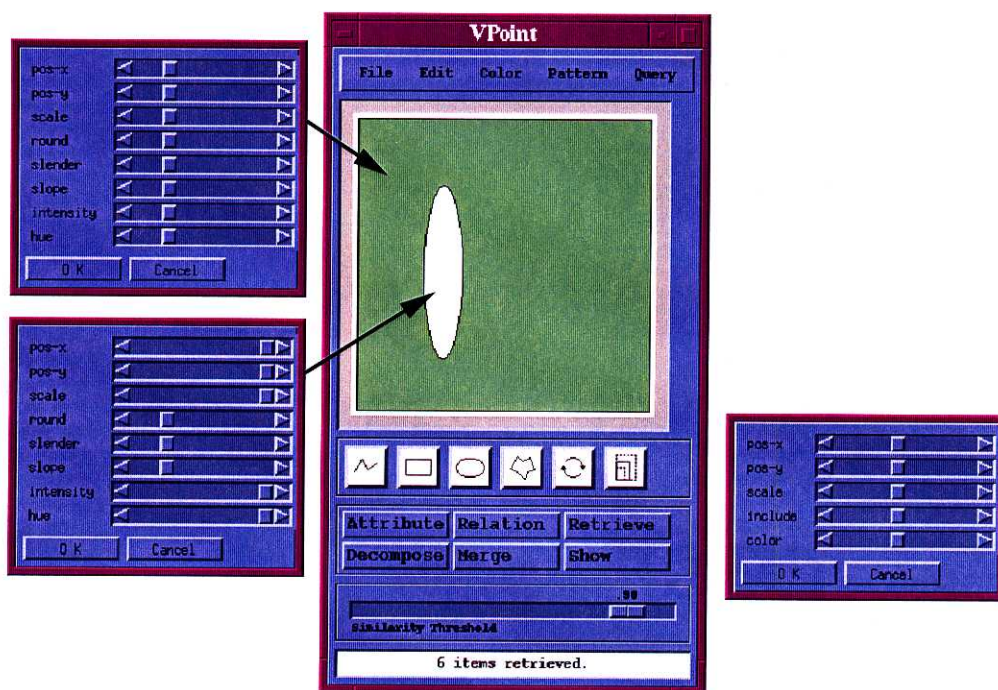
[69] sys08  
(sim=0.50)



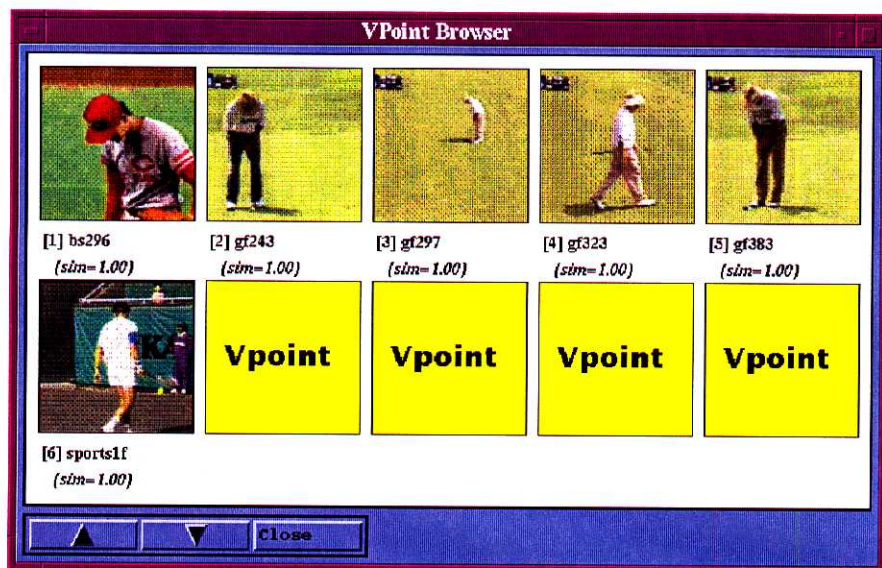
[75] dol3664pa  
(sim=0.50)

(c)

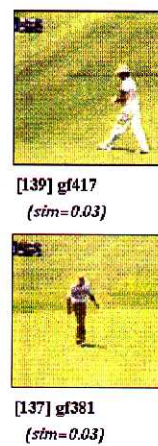
図 5.33 自然画像検索例 (9)



(a)



(b)



(c)

図 5.34 自然画像検索例 (10)

## 第 6 章

# 評 価

## 6.1 はじめに

以上、シルエット画像検索、国旗画像検索、自然画像検索について述べた。本章ではこれらの総括的な評価について述べる。評価の内容としては、VPoint の機能や特性についての定性的な評価、4章と5章で見た検索例を基にした定量的評価を行う。

## 6.2 定性的評価

### 6.2.1 Vpoint の特徴

これまでに述べた検索例により、VPoint の特長は示せたと思われるが、ここでもう一度その特長をまとめる。

- ゆらぎという機構を用いてユーザが試行錯誤的に検索を行うことができ、検索の幅を柔軟に操作できる。特に一度目にしたことがある画像を検索したい場合に有効である。
- プリミティブ分解により大局的な特徴ばかりでなく局所的な特徴も扱える。つまり multi-object and multi-feature であり、個々のプリミティブ（オブジェクト）に対して複数の特徴についての検索条件を扱える。
- 独自のプリミティブ抽出方法を用いている。特に形状分解は有効な手法であり、プリミティブ抽出処理に役立つ。
- プリミティブ間の関係を扱っている。
- 検索をシグネチャによるビット演算を主体としているので効率的である。VPoint のように、一枚の画像につき扱うプリミティブ（オブジェクト）の数が数十に及ぶ場合、通常のマッチング手法では効率が落ちてしまうが、それをシグネチャを用いることで軽減している。

以上、VPoint の特長を列挙したが、もちろん問題点もある。

- 関係については2項関係のみを扱っているので、類似していないにもかかわらずマッチングにおいて高い類似度を出す可能性があり（False drop）、余計な検索結果を多く含む原因となる。
- 従来の研究に多く見られるような特徴間の距離を測定することをしないで、特徴量を量子化してビット演算しているのが効率的だが、その分、検索が粗くなってしまいう傾向がある。そのため、個々の特徴単独では、かなり雑な検索しかできない。
- 全て自動的に原画像からプリミティブを抽出するために、冗長なプリミティブまでも扱ったので、関係の数が大きくなりすぎる傾向がある。また、それでもなお有意なプリミティブを抽出するのは難しい。



2 項関係と量子化の問題は、検索時間の効率とのトレードオフの問題である。関係を全体的に扱おうとすると、グラフ構造を考えなければならず、たちまちグラフマッチングの大きな壁に当たってしまう。また、量子化しないで距離を測るとすると、これもまた計算コストがかかってしまう。よって本手法では、ある程度の余計な検索結果が含まれるのはやむをえないと考えているが、改良するならば、シグネチャのマッチングで大まかに絞り込んだ後、別の手法でさらに絞るという方針が考えられるだろう。

プリミティブの増加や関係の増加の問題も、人手やヒューリスティックを用いてないからであり、トレードオフの問題といえる。本研究では全てシステムに任せて原画像処理を行ったが、実際に際しては、若干のマニュアル的な操作が含まれてもよいであろう。

### 6.2.2 従来の研究との本研究との比較

2 章で述べたように、本研究と比較しうる研究としては PinPoint と QBIC があるが、まず、それ以外の研究またはシステムとの相違点を述べる。VPoint では、既に見たように、ユーザが着目する特徴と空間的な部分情報をゆらぎとしてグラフィカルに明示的に指示でき、検索にその指示された情報が反映され、幅広い検索が行えるところが大きな特長である。もちろん、細かい技術に関しても相違点はあるが、研究の着眼点、システムの構想から大きな相違点がある。

次に PinPoint と QBIC との比較を行う。まず大きな違いとして以下の三つが挙げられる。なお、本論文を執筆している現在では PinPoint に関する詳細な資料は発表されていないので、多少、推察して述べるところがあることを御容赦されたい。

- VPoint では、ほぼ任意のオブジェクト（プリミティブ）単位の検索条件の設定が可能であり、オブジェクト単位のグラフィカルな操作が可能である。一方 PinPoint でもオブジェクト単位の検索条件の設定は可能であるが、それはあくまで対象が固定されている場合である。例えば顔画像を対象とするならば、目や鼻といった顔の構成部分ごとの検索条件は設定できるものの、それは予め認識されていることが前提となっている。よって自然画の例では画像内のオブジェクト単位の検索条件の操作は文献 [60] を見る限り実現されていない。
- VPoint では検索条件に属性だけでなく関係についても扱っている。これは QBIC にも PinPoint にもない特長である。通常、関係というと位置関係を扱う場合が多いが、VPoint では、大きさ・色・角度といった関係も扱った。対象を特定すれば、さらに様々な扱いうる関係があると考えられる。
- VPoint では、対象に対する特別なモデルや知識を用いておらず、さらに、人手を介した処理を一切含んでいない。一方 PinPoint の応用例では対象を特定している場合があり、QBIC は人手により有意な領域を原画像から切り出している。もちろん、人手を介したり対象を特定した方が実用的であり、検索精度も上がるが、国旗画像で見たように、デザイン的な分野の対象などを扱う場合には本手法のようなアプローチが必要とされると考えられる。

この他に、シグネチャを用いた検索機構、分解による形状特徴の記述などの相違点もある。

また、自然画像検索について比較すると、検索の多様性に関しては、VPoint は PinPoint や QBIC に優れた面を持っているといえるが、精度に関しては劣る。これは原画像処理を手で行うか自動で行うかの差であり、人手に頼った方が精度が上がるのは当然である。しかし実用問題としては、人手に頼らざるをえないのも現実である。よって VPoint でも、プリミティブの抽出に人手を加えて、余計なプリミティブを削除していけば、精度は向上するであろうし、マニュアル操作の支援にもなりうる。しかし、シルエット画像検索や国旗画像検索のような応用は VPoint 独自の手法だからこそ実現できるものと考えられる。

## 6.3 定量的評価

### 6.3.1 検索時間

まず検索時間について述べる。検索時間は主に問い合わせの例時画のオブジェクトの数とデータの数に影響される。例示画内のオブジェクトの属性及び関係の特徴を計算するのに若干のコストがかかる。シルエット画像検索の検索例 1 では 1 秒もかからないが、検索例 6 では約 10 秒ほど要した。また国旗画像検索においては検索例 1 において約 1 秒、自然画検索の検索例 1 においても約 1 秒であった。よって問い合わせシグネチャを作成するのに若干の時間が必要であることがわかるし、マッチング自体は効率的であるといえる。但しデータ数が膨大になった場合は、シグネチャを階層的にするなどして、対策を講じる必要があるだろう。

### 6.3.2 検索精度

次に適合率と呼出率を測定することで検索精度の評価を行う。

#### 適合率と呼出率

まず適合率と呼出率の定義を述べておく。ある問い合わせに対して全データ集合のうち適合するデータ集合を  $R_q$  とする。また、検索システムが問い合わせに対して選択抽出した集合を  $E_q$  とし、 $E_q$  のうち問い合わせに適合する集合を  $E'_q$  とする。この時、適合率及び呼出率は

$$\text{適合率} = \frac{E'_q}{E_q}, \text{呼出率} = \frac{E'_q}{R_q} \quad (6.1)$$

となる。

#### 準備

さて、本節ではこの適合率と呼出率について述べるのだが、その前に以下のいくつかの準備及び前提をする。

- 評価に客観性を持たせるために、筆者を含め計四人の人間がいくつかの検索要求を出し、これらを基に評価に用いる検索例を想定した。評価はこれに基づいて行う。

- 本研究において検索の精度が落ちる場合、三つの原因が考えられる。まず原画像から適切にプリミティブが抽出されない場合である。これは主に呼出率の低下の原因となる。次にシグネチャのマッチングが不適切であった場合である。この場合は主に適合率の低下の原因となる。最後はユーザからの検索要求がシステムに正確に伝達されていない場合である。最初の2つは評価の対象に入るが、最後の原因についてはその限りではない。ユーザが想定した検索要求をシステムに指示する場合、システムの操作に不慣れであると、誤った指示や不十分な指示をする可能性があるが、本評価では、ユーザはシステムの操作にある程度熟練していると仮定する。
- VPoint では試行錯誤的な検索を提供しているので、問い合わせとブラウジングを何度か繰り返して検索する。よって検索の終了はユーザの判断に任せる。但し、ブラウジングは基本的に検索数が10件以内まで絞り込んだ時点で行うものとする。もちろん、そこから検索の幅を広げて、さらに追加のデータを検索することは許される。そして、ブラウジングされた中で適合するもののうち、最も類似度が低いものまでを評価の際の検索数とする。例えば、5件検索されて、そのうち類似度が高い方から1番目と3番目が適合したとすると、適合率は66%になる。

なお、4章、5章で示した検索例は、これらの状況に沿って行われたものである。以下、各対象についての評価を述べる。

#### シルエット 画像検索

表 6.1 はシルエット画像検索における想定した検索要求についての評価である。表中の第一列の数字の一部は 4.5 節の検索例に対応している。  $N$ ,  $m$ ,  $n$  はそれぞれ検索数、検索洩れの数、余計な検索数を表す。また呼出率を  $P(\%)$ 、適合率を  $R(\%)$  と表す。

#### 国旗画像検索

表 6.2 は国旗画像検索における想定した検索要求についての評価である。表中の第一列の数字の一部は 5.3.3 節の検索例に対応している。

#### 自然画像検索

表 6.3 は自然画像検索における想定した検索要求についての評価である。表中の第一列の数字の一部は 5.4.3 節の検索例に対応している。

	検索要求	$N$	$n$	$m$	$P$	$R$
1	尾の長い動物 (鼠)	5	0	0	100	100
2	小さい翼の鳥 (燕)	4	0	2	100	100
3	四つ足で首が長い動物 (麒麟)	3	0	2	100	100
4	角と耳のある動物の顔	2	0	0	100	100
5	走っている馬	1	0	0	100	100
6	こぶを二つ持つ右向きのらくだ	3	0	1	100	67
7	らくだ	7	1	2	83	71
8	ぎざぎざ模様	2	0	0	100	100
9	胴長で脚が短い動物	7	1	4	75	43
10	細長い脚を持つ鳥	10	1	3	88	70
11	口ばしを持つ生き物	38	10	22	62	42
12	下を向いている牛	1	1	0	100	100
13	口ばしを開いている鳥	4	0	2	100	50
14	四つ這い状態の動物	4	0	2	100	50
15	蝶々	5	3	3	40	40
平均	-	-	-	-	90	76

表 6.1 シルエット画像検索の評価

	検索要求	$N$	$n$	$m$	$P$	$R$
1	赤白緑で縦に等分に区切った国旗	2	0	0	100	100
2	赤白緑で横に等分に区切った国旗	3	0	1	100	67
3	赤白緑で横におおまかに区切った国旗	10	0	2	100	80
4	右斜めに区切った国旗	8	1	1	88	88
5	斜めに区切った国旗	10	2	1	82	90
6	左端に三角形がある国旗	22	3	7	83	68
7	3つの横縞があり中央にマークがある国旗	12	3	3	75	75
8	緑地に白の三日月がある国旗	4	0	0	100	100
9	水色地に盾のマークがある国旗	5	0	4	100	20
10	円の中に星がある国旗	4	0	1	100	75
11	星マークがある国旗	45	8	5	83	89
12	細かい絵柄の国旗	-	-	-	0	0
13	正三角形のような形状がある国旗	97	0	76	100	22
14	木のシンボルがある国旗	3	0	2	100	33
15	細い横縞模様で色が交互の国旗	8	0	3	100	63
平均	-	-	-	-	87	65

表 6.2 国旗画像検索の評価

	検索要求	$N$	$n$	$m$	$P$	$R$
1	青い空とヨットがある画像	3	0	2	100	33
2	カラフルな丸（風船）がいくつかある画像	2	0	1	100	50
3	SIDBA の猿の画像	1	0	0	100	100
4	チョコのようなものがのっているクッキーの画像	10	0	8	100	20
5	明るい緑の服の肩より上が写っている人の画像	4	0	2	100	50
6	野球の対決シーン	5	0	0	100	100
7	ちょんまげの人物がいる時代劇シーン	3	3	2	25	33
8	魚の画像	9	2	2	78	86
9	白のビキニの女の人	4	3	1	50	75
10	ゴルフのパターシーン	6	2	2	67	67
11	下半分が芝のサッカーシーン	3	1	2	50	33
12	きれいな海がある画像	7	2	6	33	14
13	女の子が何か食べている画像	-	-	-	0	0
14	力士が大きく写っている画像	9	3	3	67	67
15	左打者が正面に立つ野球の画像	2	0	1	100	50
16	スーツの男性が話しているシーン	9	8	7	20	22
17	スポーツシーンで芝を含む画像	75	12	34	77	55
18	下半分が芝の競馬シーン	12	2	10	50	17
19	青服の F1 レーサーの会見シーン	4	0	2	100	50
20	横向きの車の画像	1	2	0	33	100
平均	-	-	-	-	67	51

表 6.3 自然画像検索の評価

### 6.3.3 所見

以上、評価を行ったが、やはり自然画像を対象とした場合は精度が落ちる。これは、明らかに原画像処理の難しさに起因し、さらに、自然画像の検索では、どうしても色情報を頼らざるをえない現状があるからである。形状情報では信頼性が劣ってしまうのだ。

今回の定量評価では、ある特定の一枚の画像を検索するという検索状況も少なからず含まれていた。この場合、所望の画像を見つけるまで操作を繰り返すので、呼出率が高くなるのは当然である。ただ、その際に適合率がどの程度かというのが重要である。このような観点から見ると、シルエット画像や国旗画像では、画像の類似検索としては良好な結果が出ているといえる。自然画像でも、満足いく結果が得られる場合とそうでない場合が混在するが、従来とは異なる柔軟な検索ができる機能を考えれば、良好な結果であるといってもよいであろう。

ここで、試行錯誤の回数は少ない場合は1回ですんだが、多い場合は20回ほどもかかっていることに注意されたい。よって、今回の評価ではある程度、精度がよくなったわけだが、逆に、このような試行錯誤的な検索が可能であるということを示したことになる。

自然画像を対象として、さらに精度を、完成度を高めるには、キーワード検索との融合、人手による補正作業、ヒストグラムやテキスチャを用いた特徴の導入、などの改良を加えなければならぬと考えられる。

## 第7章

### 結 論

本研究では、ユーザの多様な検索要求を柔軟に処理しうる画像検索について述べた。また、その要素技術として、形状分解、領域分割によるプリミティブ抽出方法、シグネチャによる検索機構、それらを基にした例示画検索による画像検索プロトタイプシステム VPoint を提案した。さらに本検索手法により、ユーザが検索要求をグラフィカルかつ明示的にシステムに伝達できることを示し、従来とは異なった柔軟な検索が可能であることを確認し、その有効性を示した。

本システム VPoint の最大の特長は、ユーザによって試行錯誤的な検索が可能であることである。現在、商用の文献検索データベースでは、複数の検索キーワードと和集合や積集合などのコマンドを用いて、検索数をユーザが絞り込むのが一般的である。うまく絞り込めない場合や所望の文献が検索されない場合は、キーワードを変更するなどして検索条件を変えながら試行錯誤を繰り返して検索を行う。これは当然画像検索においても許されることであり、逆にこのような操作ができないのでは不便このうえない。キーとなる画像とデータの画像との類似度が一意に測られ、ユーザの操作の入り込む余地がないのでは困るというものである。VPoint では、この点を重視して構築されたシステムであり、特に、一度目にしたことのある画像を検索する場合や、断片的な情報を基にした画像検索には有効であると考えられる。もちろん、VPoint のコンセプトや手法をそのままオーサリングツールとして用いたり、画像のインデックス化作業に対する支援ツールとして応用してもよいであろう。

本論文では、VPoint の実装対象として、シルエット画像、国旗画像、自然画像の3つを取り扱った。自然画像を扱う場合、最も難しい技術課題は有意なプリミティブの抽出方法である。本研究では、敢えて対象不特定にして対象の知識やモデルといったヒューリスティックな手法を用いることをせず、また人手を介さない方針をとったが、実用を考えた場合、まだまだ不十分である。成功した検索もあったが、失敗した検索も少なくなかった。よって本手法を実用に供することを考えた場合は、人手によるデータの修正作業を行うべきである。しかし、人手によりプリミティブを選択するにしても、その選択基準には客観性がなく、個人差が出る可能性がある。この場合、冗長なプリミティブの記述を許す VPoint の手法が活かされると考えられる。また特定の対象に限った場合は、VPoint の機構はより有効に機能するはずである。

しかし、自然画像よりはむしろシルエット画像や国旗画像などのような、デザイン画像やパターン画像を対象にした場合の方が、本研究の目的に沿うものである。自然画像への検索要求は、やはり色や概念が主体となりがちであり、検索要求の幅の広さからいうと、デザイン画像の方がより興味深い対象といえる。具体的には、美術関係、ホームショッピングなどの応用先があると考えられる。

このように、VPoint そのものはプロトタイプシステムであるので、実用に供するにはまだまだ改良の余地が多く、その道のりも容易なものではないが、本論文において、従来にはない柔軟な画像検索が可能であることを示せたといえる。

最後に、本研究はおそらく筆者のこの論文をもって幕を閉じることになると思われるが、今後、VPoint をはじめ PinPoint, QBIC のような画像検索システムが研究開発されることと思われる。その際に、本研究がそれら新たな研究に対しての何かしらの手掛かりになれば幸いである。



# 謝 辞

最後に、本研究において、研究の進むべき道を示していただくなど多くの御指導を頂きました指導教官の浜田喬教授に心から感謝の意を表明いたします。また、研究を進めるにあたり御相談及び御助言を頂きました佐藤真一助手に心より感謝申し上げます。

さらに、日頃から公私ともに大変お世話になった、大塚キャンパス学術情報センター内濱田研究室、浅野研究室、安達研究室の皆さんに深く感謝いたします。

最後に、陰ながら自分を支えてくれた両親及び姉兄、そして我が妻に、心より感謝の意を捧げる。

## 参考文献

- [1] 岡崎 彰夫. 画像データベース概論. 情報処理学会誌, 33(5):448-456, May 1992.
- [2] 岡崎 彰夫. 画像データベース技術の現状と将来. In *Proceedings of Advanced Database System Symposium'93*, pages 45-54, December 1993.
- [3] 佐藤 真知子, 橋原 秀晴, 井岡 幹博, 黒川 雅人, 洪 政国, 杉田 繁治, 久保 正敏, 山本 泰則. 民族学研究支援のための標本画像検索システム. 情報処理学会論文誌, 29(12):1108-1118, December 1988.
- [4] 浦谷 則好, 柴田 正啓, 野口 英男, 相沢 輝昭. 静止画検索システム FORKS の試作. 情報処理学会論文誌, 28(7):758-767, June 1987.
- [5] 柴田 正啓, 井上 誠喜. 画像データベースの連想検索方式. 電子情報通信学会論文誌, J73-D-II(4):526-534, April 1990.
- [6] 高橋 友一, 島 則之, 岸野 文郎. 位置情報を手がかりとする画像検索. 情報処理学会論文誌, 31(11):1636-1642, November 1990.
- [7] 坂内 正夫, 大沢 裕. 画像データベース. 昭晃堂, 1988.
- [8] Jack A. Orenstein, Frank A. Manola. PROBE Spatial Data Modeling and Query Processing in an Image Database Application. *IEEE Transaction on Software Engineering*, 5(14):611-629, 1988.
- [9] Nick Roussopoulos, Christos Faloutsos, Timos Sellis. An Efficient Pictorial Database System for PSQL. *IEEE Transactions on Software Engineering*, 14(5):639-650, May 1988.
- [10] 大沢 裕, 坂内 正夫. 2種類の補助情報により検索と管理性能の向上を図った多次元データ構造の提案. 電子情報通信学会論文誌, J74-D-I(8):467-475, August 1991.
- [11] 坂内 正夫. 画像検索技術. In 電子情報通信学会誌, volume 71, pages 911-914, September 1988.
- [12] 加藤 俊一, 栗田 多喜夫. 画像の内容検索. 情報処理学会誌, 4(5):466-477, 5月 1992.

- [13] Olivia R. Liu Sheng, Huichin Wang, and Hong-Mei Chen Garcia. IRES - Image Retrieval Expert System. In *SPIE*, volume 1234, pages 832-841, 1990.
- [14] J. N. David Hibler, Clement H. C. Leung, Keith L. Mannoek, Njagi K.Mwara. A System for Content-based Storage and Retrieval in an Image Database. In *SPIE*, volume 1662, pages 80-88, April 1992.
- [15] N. Dal Degan, R. Lancini, P. Migliorati and S. Pozzi. Still images retrieval from a remoto database: The system Imagine. *Signal Processing: Image Communication*, 5:219-234, 1993.
- [16] 増永 良文. マルチメディアデータモデル OMEGA における音データと動画データのリポート. In *Proceeding of Advanced Database System Symposium '93*, pages 163-180, 12月 1993.
- [17] Eitetsu Oomoto, Katsumi Tanaka. OVID: Design and Implementation of a Video-Object Database System. *IEEE Transaction on Knowledge and Data Engineering*, 5(4):629-643, August 1993.
- [18] 有澤 博, 由井 仁, 富井 尚志. 映像データベースシステムの構成の一方式. In *Proceeding of Advanced Database System Symposium '93*, pages 181-190, 12月 1993.
- [19] Mikihiro Ioka, Masato Kurokawa. A Method for Retrieval Sequences of Images on the basis of Motion Analysis. In *SPIE*, volume 1662, pages 35-46, 1992.
- [20] 栗田 多喜夫, 下垣 弘行, 加藤 俊一. 主観的類似度に適応した画像検索. 情報処理学会論文誌, 31(2):227-237, February 1990.
- [21] 坂内 正夫, 佐藤 真一. 画像データベースにおけるモデル形成. 電子情報通信学会論文誌, J74-D-I(8):455-466, August 1991.
- [22] Jun Yamane and Masao Sakauchi. A Construction of a New Image Database System which Realize Fully Automated Image Keyword Extraction. *IEICE*, E76-D(10):1216-1223, October 1993.
- [23] 塩野 充, 真田 英彦. 個人識別技術の最近の研究動向. In 信学技報, volume IE92-49, pages 1-8, September 1992.
- [24] 加藤 俊一, 下垣 弘行, 藤村 是明. 画像対話型商標・意匠データベース TRADEMARK. 電子情報通信学会論文誌, J72-D-II(4):535-544, April 1989.
- [25] 黒川 雅人, 洪 政国. 形状情報を用いた画像の類似検索システム. 情報処理学会論文誌, 32(6):721-730, June 1991.

- [26] William I. Grosky. Index-Based Object Recognition in Pictorial Data Management. *Computer Vision, Graphics, and Image Processing*, 52:416–436, 1990.
- [27] William I. Grosky, Zhaowei jiang. A Hierarchical Approach to Feature Indexing. In *SPIE*, volume 1662, pages 9–20, 1992.
- [28] H. V. Jagadish. A Retrieval Technique for Similar Shapes. In *Proceeding of the 1991 ACM SIGMOD International Conference on Management of Data, SIGMOD Record*, volume 20, pages 138–147, 1991.
- [29] 田辺 勝義, 大谷 淳, 石井 健一郎. 多次元心理空間を用いる類似画像検索法. 電子情報通信学会論文誌, J75-D-II(11):1856–1865, November 1992.
- [30] T.-Y.Hou, A.Hsu, P.Liu, M.-Y.Chiu. A Content-based Indexing Technique using Relative Geometry Features. In *SPIE*, volume 1662, pages 59–68, April 1992.
- [31] A. Pentland, R. W. Picard and S. Sclaroff. Photobook: Tools for Content-Based Manipulation of Image Databases. In *SPIE*, volume 2368, pages 37–50, 1994.
- [32] Hemant D. Tagare, Frans M. Vos, Conrade C. Jaffe and James S. Duncan. Arrangement: A Spatial Relation Between Parts for Evaluating Similarity of Tomographic Section. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 17(9):880–893, 1995.
- [33] 北本 朝展, 高木 幹雄. 最適化原理に基づいた衛星雲画像の類似検索. In 信学技報, volume PRU94-49, pages 15–22, October 1994.
- [34] 西山 晴彦, 饗場 潔, 横山 光男, 松下 温. 略画作成による画像検索システムのインタフェース. In 信学技法, volume SP92-67, pages 47–54, September 1992.
- [35] T. Kato. Database Architecture for Content-based Image Retrieval. In *SPIE*, volume 1662, pages 112–123, 1992.
- [36] 岡野 栄司, 木下 宏揚, 酒井 善則. 手書き画像とのマッチングによる画像検索. In 信学技報, volume IE93-96, pages 53–60, January 1994.
- [37] S. K. Chang, C. W. Yan, D. C. Dimitroff, and Timothy Arndt. An Intelligent Image Database System. *IEEE Transaction on Software Engineering*, 14(5):681–688, May 1988.
- [38] C. C. Chang and S. Y. Lee. Retrieval of Similar Pictures on Pictorial Database. *Pattern Recognition*, 24(7):675–680, July 1991.
- [39] S. K. Chang, C. M. Lee, and C. R. Dow. A 2-D String Matching Algorithm for Conceptual Pictorial Queries. In *SPIE*, volume 1662, pages 47–58, April 1992.

- [40] S. K. Chang and A. Hsu. Image Information Systems: Where Do We Go From Here ? *IEEE Transaction on Knowledge and Data Engineering*, 4(5):431-442, October 1992.
- [41] Suh-Yin Lee, Fang-Jung Hsu. Spatial Reasoning and Similarity Retrieval of Images using 2D C-String Knowledge Representation. *Pattern Recognition*, 25(3):305-318, March 1992.
- [42] G. Costagliola, G. Tortora, and T. Arndt. A Unifying Approach to Iconic Indexing for 2-D and 3-D Scenes. *IEEE Transaction on Knowledge and Data Engineering*, 4(3):205-222, June 1992.
- [43] A. D. Bimbo, M. Campanai, and P. Nesi. A Three-Dimensional Iconic Environment for Image Database Querying. *IEEE Transaction on Software Engineering*, 19(10):997-1011, October 1993.
- [44] 饗場 潔, 永田 尚, 松本 陽一郎, 横山 光男, 松 下 温. 画像検索におけるインタフェースの提案. In 情報処理学会研究会報告, volume HI-37-6, July 1991.
- [45] Michael J. Swain. Interactive Indexing into Image Database. In *SPIE*, volume 1908, pages 95-103, 1993.
- [46] 長坂 晃朗, 田中 譲. カラービデオ映像における自動索引付け法と物体検索. 情報処理学会論文誌, 33(4):543-550, 4月 1992.
- [47] 長坂 晃朗, 上田 博唯, 宮武 孝文. 部分領域の色の組合せに着目した映像中の特定 対象物検索手法の提案. In 信学技法, volume IE92-60, pages 1-8, October 1992.
- [48] 長谷川 純一. スケッチを利用した胸部 X 線像データベースの知的検索. 電子情報通信学会論文誌, J71-D(3), 1988.
- [49] 安部 伸治, 外村 佳伸. 状態の時間変化をキーとする動画像検索法. 電子情報通信学会論文誌, J75-D-II(3):512-519, March 1992.
- [50] K. Wakimoto, M. Shima, S. Tanaka, and A. Maeda. Content-based retrieval applied to drawing image database. In *SPIE*, volume 1908, pages 74-84, 1993.
- [51] 脇本 浩司, 島 光秀, 田中 聡, 前田 章. グラフ表現を利用した図面の類似検索方式. 電子情報通信学会論文誌, J77-D-II(7):1302-1310, July 1994.
- [52] 尾田 政臣. 人間の認知特性を利用した画像検索システム. In 信学技報, volume HC92-45, pages 55-62, September 1992.
- [53] 尾田 政臣. 人間のイメージ形成過程の特性を利用した画像 検索システム. 情報処理学会論文誌, 35(7):1449-1456, 7月 1994.

- [54] 坂本 弘章, 高畑 実, 上森 明. 複数のキー画像の部分的な特徴を使用する静止画像検索方式の提案-フレキシブル・モンタージュ検索-. In 信学技報, volume PRU93-80, pages 31-36, November 1993.
- [55] 高畑 実, 坂本 弘章, 上野 圭一, 上森 明. ユーザの検索意図を反映した画像提供方式. In 信学技報, volume IE93-116, pages 29-36, February 1994.
- [56] 美濃 導彦, 岡崎 洋. 対象物の属性特徴による画像検索法～風景画像中の山を例として～. 情報処理学会論文誌, 32(4):513-522, 4月 1991.
- [57] 平田 恭二, 原 良憲. ハイパーメディアシステム「雅」におけるメディアナビゲーション方式. In *Proceedings of Advanced Database System Symposium '93*, pages 17-26, 12月 1993.
- [58] Peter Eggleston. Constraint Based Feature Indexing and Retrieval for Image Database. In *SPIE*, volume 1819, pages 27-39, 1992.
- [59] G. Yihong, H. Zhang, H. C. Chuan, and M. Sakauchi. An Image Database System with Content Capturing and Fast Image Indexing Abilities. In *1994 International Conference on Multimedia Computing and Systems*, pages 121-130, 1994.
- [60] Jeffrey R. Bach, Charles Fuller, Bradley Horowitz, Ken Liu and Chiao-Fe Shu. Managing Multimedia Data. In *MISH'95*, 1995.
- [61] R. Jeffrey and P. Santanu. A Visual Information Management System for the Interactive Retrieval of Faces. *IEEE Transaction on Knowledge and Data Engineering*, 5(4):619-628, 1993.
- [62] W. Niblack, R. Barber, W. Equitz, M. Flickner, E. Glasman, D. Petkovic, P. Yanker, C. Faloutsos, and C. Taubin. The QBIC Project: Querying Images By Content Using Color, Texture, and Shape. In *SPIE*, volume 1908, pages 173-187, 1993.
- [63] Denis Lee, Ron Barber, Wayne Niblack, Myron Flickner, Jim Hafner and Dragutin Petkovic. Query by Image Content Using Multiple Objects and Multiple Features: User Interface Issues. In *IEEE International Conference on Image Processing*, volume 1, pages 76-80, 1994.
- [64] Denis Lee, Ron Barber, Wayne Niblack, Myron Flickner, Jim Hafner and Dragutin Petkovic. Indexing for Complex Queries on a Query-By-Content Image Database. In *International Conference on Pattern Recognition*, volume 1, pages 142-146, 1994.
- [65] Michael Gleicher. A Graphics Toolkit Based on Differential Constraints. In *UIST'93*, pages 109-120, November 1993.

- [66] D.L.Maulsby, I.H.Witten, K.A.Kittlitz, V.G.Franceschin. Inferring Graphical Procedures: The Complete Metamouse. *Human Computer Interaction*, 7(1):47-89, 1992.
- [67] 松浦 敏雄, 直田 創, 中村 眞. 図形の部品化および接続包含関係の保存機能を持つ作図ツール Key3. 電子情報通信学会論文誌, 73-D-I(11):864-872, 11月 1990.
- [68] D. Kurlander, S. Feiner. A History-Based Macro By Example System. In *UIST'92*, pages 99-106, November 1992.
- [69] D. Kurlander, S. Feiner. Inferring Constraints from Multiple Snapshots. *ACM Transaction on Graphics*, (4):277-304, October 1993.
- [70] 金原 史和, 佐藤 真一, 濱田 喬. 図形間の幾何的および概念的関係を用いた作図支援. 情報処理学会論文誌, 35(5):897-907, May 1994.
- [71] Tavakoli N. and Ray A. A New Signature Approach for Retrieval of Documents from Free-Text Databases. *Information Processing & Management*, 28(2):153-163, 1992.
- [72] P. Tiberio and P. Zezula. Selecting Signature Files For Specific Application. *Information Processing & Management*, 29(4):487-498, 1993.
- [73] 古瀬 一隆, 浅田 一繁, 飯沢 篤志. DBMS へのシグネチャファイルの実装について. In 信学技法, volume DE94-58, pages 23-30, September 1994.
- [74] Gerard Salton. *Automatic Text Processing*. Addison Wesley, 1989.
- [75] 伊理 正夫, 古林 隆. ネットワーク理論. 日科技連, 1976.
- [76] N. Hartsfield and G. Ringel. グラフ理論入門. サイエンス社, 1992.
- [77] Christos H. Papadimitriou and Kenneth Steiglitz. *Combinatorial optimization : algorithms and complexity*. Englewood Cliffs, N.J., 1982.
- [78] 森 俊二, 坂倉 梅子. 画像認識の基礎 [I]/[II]. オーム社, 1986.
- [79] 高木幹男, 下田 陽久, editor. 画像解析ハンドブック. 東京大学出版会, 1991.
- [80] L. G. Shapiro. Recent Progress in Shape Decomposition and Analysis. In *Progress in Pattern Recognition 2*, pages 113-123, 1985.
- [81] 阿部 圭一. 図形の記述と理解. 電子情報通信学会誌, 77(5):507-514, May 1994.
- [82] L. G. Shapiro and R. M. Haralick. Decomposition of two-dimensional shapes by graph-theoretic clustering. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, PAMI-1(1):10-20, January 1979.

- [83] C. Guerra and G. G. Pieroni. A Graph-theoretic Method for Decomposing Two-dimensional Polygonal Shapes into Meaningful Parts. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, PAMI-4(4):405-408, April 1982.
- [84] J. Fairfield. Segmenting Dot Patterns by Voronoi Diagram Concavity. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 5(1):104-110, January 1983.
- [85] Carolyn M. Bjorklund and T. Pavlidis. Global Shape Analysis by k-Syntactic Similarity. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 3(2):144-155, March 1981.
- [86] A. J. Nevins. Region Extraction from Complex Shapes. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 4(5):500-511, May 1982.
- [87] T. Pavlidis. Algorithms for shape analysis and contour and waveforms. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, PAMI-2(4):301-312, July 1980.
- [88] H. Rom and G. Medioni. Hierarchical Decomposition and Axial Shape Description. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 15(10):973-981, October 1993.
- [89] D. M. Wuescher and K. L. Boyer. Robust Contour Decomposition Using a Constant Curvature Criterion. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 13(1):41-51, January 1991.
- [90] 間瀬 茂, 上田 修功. モルフォロジーと画像解析 [I]. 電子情報通信学会誌, 74(2):166-174, 2月 1991.
- [91] 原 潤一, 加藤 博一, 井口 征士. だ円膨張法を用いた動物シルエット画像の記述と認識. 電子情報通信学会論文誌, J74-D-II(3):366-375, March 1991.
- [92] I. Pitas and A. N. Venetsanopoulos. Morphological Shape Decomposition. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 12(1):38-45, January 1990.
- [93] 木本 伊彦, 浅井 基博, 安田 靖彦. だ円体によるシルエット図形の記述と表現. 電子情報通信学会論文誌, 76-D-II(6):1159-1167, 6月 1993.
- [94] 安居院 猛, 長尾 知晴. 画像の処理と認識. 昭晃堂, 1992.
- [95] 笠尾 敦司. 分割 K-平均アルゴリズムを用いた大画像の領域分割. 電子情報通信学会論文誌, J78-D-II(2):221-228, February 1995.
- [96] Dileep Kumar Panjwani and Glenn Healey. Markov Random Field Models for Unsupervised Segmentation of Textured Color Images. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 17(10):939-945, October 1995.



- [97] Z. Wu and R. Leahy. An Optimal Graph Theoretic Approach to Data Clustering: Theory and Its Application to Image Segmentation. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 15(11):1101–1113, November 1993.
- [98] P. Salembier and M. Pardàs. Hierarchical Morphological Segmentation for Image Sequence Coding. *IEEE Transaction on Image Processing*, 3(5):639–651, September 1994.
- [99] 大田 友一, 金出 武雄, 坂井 利之. 領域分割処理によるカラー画像情報の構造化. *情報処理*, 19(12):1130–1136, December 1978.
- [100] N. Ueda and S. Suzuki. Learning Visual Models from Shape Contours Using Multiscale Convex/Concave Structure Matching. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 15(4):337–352, April 1993.

## 発表文献

- [1] 金原 史和, 佐藤 真一, 濱田 喬. 作図支援を例とした視覚的マンマシンインタフェース. In 第45回情報処理学会全国大会予稿集, volume 2T-6, pages 205-206, 10月 1992.
- [2] 金原 史和, 佐藤 真一, 濱田 喬. 図形間の幾何的・意味的關係を用いた作図支援システムに関する研究. Master's thesis, 東京大学工学系研究科電子工学専攻, 1993.
- [3] 金原 史和, 佐藤 真一, 濱田 喬. 図形間の幾何的・意味的關係を用いた作図支援システム. In 第4回機能図形情報システムシンポジウム講演論文集, pages 1-6, 4月 1993.
- [4] 金原 史和, 佐藤 真一, 濱田 喬. 視覚心理に基づいた特徴抽出. In 第47回情報処理学会全国大会予稿集, volume 3L-2, pages 117-118, 3月 1993.
- [5] 金原 史和, 佐藤 真一, 濱田 喬. 属性と關係の明示的指示による例示画検索. In 1994年電子情報通信学会春季全国大会予稿集, volume D-500, 3月 1994.
- [6] 金原 史和, 佐藤 真一, 濱田 喬. 図形間の幾何的および概念的關係を用いた作図支援. 情報処理学会論文誌, 35(5):897-907, May 1994.
- [7] 金原 史和, 佐藤 真一, 濱田 喬. スケッチ画像検索インタフェースについての一提案. In 第49回情報処理学会全国大会予稿集, volume 1U-6, pages 339-340, 9月 1994.
- [8] Fumikazu KANEHARA, Shin'ichi SATOH and Takashi HAMADA. An interactive drawing editor system focused on geometric and conceptual relations. In *Research Bulletin of the National Center for Science Information Systems*, number 7, pages 153-163, March 1995.
- [9] 金原 史和, 佐藤 真一, 濱田 喬. シルエット画像のための形状分解の一手法. In 1995年電子情報通信学会総合大会講演論文集, volume D-601, 3月 1995.
- [10] Fumikazu KANEHARA, Shin'ichi SATOH and Takashi HAMADA. Shape decomposition based on erosion model. In *Proceeding of the Workshop on Physics-Based Modeling in Computer Vision in Cambridge, USA*, pages 128-134, June 1995.
- [11] Fumikazu KANEHARA, Shin'ichi SATOH and Takashi HAMADA. A flexible image retrieval using explicit visual instruction. In *Third International Conference on Document Analysis Recognition in Montreal, Canada*, pages 175-178, August 1995.

- [12] 金原 史和, 佐藤 真一, 濱田 喬. 形状分解によるユーザの視点に基づいたシルエット画像検索.  
情報処理学会論文誌, 36(12):2800-2810, December 1995.