

ベクタ画像を対象としたプリミティブ選択モデルに基づく オブジェクト領域抽出

林 貴宏[†] 尾内 理紀夫[†]

本論文では、2D ベクタ画像を対象とするオブジェクト領域抽出法を提案、評価する。ラスタ画像からオブジェクト領域を自動抽出する手法はこれまで提案されてきたが、ベクタ画像を対象とした手法は報告されていない。そのため、既存手法によりベクタ画像からオブジェクト領域を抽出するためには、ベクタ画像をいったんラスタ化する必要があった。しかし、ラスタ化によってベクタ画像が持つ様々な利点は失われる。そこで本研究は、ベクタ画像の利点を活用できる効率的なオブジェクト領域抽出法を提案した。提案手法は、ベクタ画像を構成する個々の図形要素と、これら図形要素が定義しているベクタ画像中の部分領域との関係を解析し、オブジェクト領域を構成している図形要素群を判別する。評価実験として、提案手法と従来のラスタベース手法の1つである ACM (Active Contour Model) の抽出精度と速度性能を比較した。実験結果から、提案手法が ACM よりも高い抽出精度と速度性能を持つことを確認した。また、提案手法の速度性能に改善の余地があることを確認した。

Object Extraction for Vector Images with Primitive Selection Model

TAKAHIRO HAYASHI[†] and RIKIO ONAI[†]

This paper presents a novel object extraction from 2D vector images. Although various object extractions from raster images have been proposed, none of them has a mechanism for handling vector images. Therefore, in order to extract object regions from vector images using previous methods, it is necessary to convert vector images into raster images. However, various merits of vector image are discarded by rasterization. In this study, we have proposed a novel object extraction method using merits of vector images. The proposed method analyzes relations between individual primitives of vector images and the regions defined by these primitives. According to the analysis, a set of primitives defining object region is detected. We have compared accuracy and speed performance between the proposed method and ACM (Active Contour Model), which is a traditional raster-based object extraction method. Experimental results have shown that the proposed method has higher accuracy and speed performance than ACM. In addition, we have confirmed the proposed method has room for improvement in speed performance.

1. はじめに

画像の表現方式にはラスタ方式とベクタ方式がある。ラスタ方式が画素を基本単位として画像を表現するのに対し、ベクタ方式は、図形要素（ポリゴン、矩形、楕円、直線、曲線などの基本的な図形）を基本単位として、それらの重ね合わせにより画像を表現する点が特徴である。ベクタ方式は一般に、形状、配色、濃淡などは単純であるが輪郭の鮮明な画像を描くことができるため、イラスト、ロゴ、図表、グラフ、地図などの表現に適している¹⁾。また、ラスタ方式と異なり拡大縮小による画像の歪みが発生せず、一般にファイル

サイズが小さい。また、従来は描画時間の遅さが問題とされたが、計算機の処理能力向上にともない、この問題はほとんど無視できるようになった。そのため、ベクタ画像の使用量は次第に増加している。特に近年では、WEB のドキュメントに含まれるイラストなどに Flash や SVG (Scalable Vector Graphics) などのベクタ画像が多用されている²⁾。ベクタ画像がデータベース内に蓄積されつつある中、ベクタ画像に対する検索要求は今後高まっていくものと予測できる³⁾⁻⁶⁾。

画像の検索方式の1つに利用者が描いたスケッチやサンプルとして選択した画像を検索キーとする例示検索がある⁷⁾⁻⁹⁾。この方式は、検索キーとなる画像とデータベース内の画像との類似性を内容情報（画像特徴量）に基づき評価する点の特徴である。例示検索はこれまでラスタ画像を対象とした検索手法として開

[†] 電気通信大学

The University of Electro-Communications

発されてきたが、この手法はベクタ画像検索へも応用可能と考えられる³⁾。

例示検索システムでは、システムによる画像間（検索キーとなる画像とデータベース内の画像）の類似性判断を人間の感覚と対応させる必要がある。人間による類似性判断には、画像中のオブジェクト領域の内容情報が、背景領域の内容情報よりも強く影響を与えることが確認されている。たとえば、人間はオブジェクト領域が同一の2枚の画像を見たときに、背景領域が互いに異なっても類似と感じる。そのため、各画像からオブジェクト領域を抽出する処理は、システムによる類似性判断を人間の感覚に対応させ、人間の感覚に合った検索結果を提示するうえで必須の前処理といえる^{10)~13)}。

既存のオブジェクト領域の抽出法^{14)~17)}はラスタ画像を対象としているため、これらの手法をベクタ画像へ適用しようとするとき、ベクタ画像をラスタ化する必要があった。しかし、ラスタ化することはベクタ画像を構成する個々の図形要素が持つ形状や位置などに関する情報を捨てることになる。そこで本研究では、ベクタ画像が持つ個々の図形要素が持つ情報を活用し、イラストなど2Dのベクタ画像からオブジェクト領域を抽出する手法を新たに提案する。また、既存手法（ラスタ画像を対象としたオブジェクト領域抽出法）との比較実験により、提案手法の有効性と限界を評価することを目的とする。

以下、2章で既存のオブジェクト領域抽出法について述べる。3章でベクタ画像中のオブジェクト領域を定義する。4章で図形要素の選択に基づくオブジェクト領域抽出法を提案する。5章で提案手法の有効性と限界を評価するためにに行った実験について述べる。

2. 既存のオブジェクト領域抽出法

エッジ抽出に基づく方法は、ラスタ画像のオブジェクト領域抽出で典型的に使用されてきた方法である。この方法ではエッジ点を抽出して局所的に接続することでボトムアップ的にオブジェクト領域の輪郭を抽出する点が特徴である。医用画像や自然風景画像など様々な分野の画像で使用されているが、領域知識なしに正しい輪郭を得ることは非常に困難である¹⁸⁾。

ACM (Active Contour Model)¹⁴⁾は、画像上で単一閉曲線を動的に変形させながらオブジェクト領域の輪郭に一致させることで輪郭抽出を行う手法である。ACMによる輪郭抽出は、輪郭線に対し定義されるエネルギー E を最小化する問題として定式化されている。一般に、エネルギー E は、閉曲線の形状などに

よって定義される内部エネルギー E_{int} と、閉曲線が通過する各点での局所的な画像特徴などによって定義される外部エネルギー E_{ext} との和で定義されるが、 E_{int} や E_{ext} は特に決まった定義方法は存在しないため、問題領域に応じて適切に設計する必要がある。ACMは、閉曲線の変形を繰り返し、変形のたびにエネルギー計算を必要とするため輪郭を得るまでに時間がかかるという問題がある。また、単一閉曲線を使用するため、オブジェクト領域が複数存在する画像に対して適用できないという問題がある。仮に複数のオブジェクト領域が存在する画像に適用したとしても、オブジェクト領域ごとの輪郭を抽出することはできず、複数のオブジェクト領域とその間の背景領域を合わせて1つのオブジェクト領域として抽出してしまう可能性がある。複数のオブジェクト領域の輪郭を扱えるようにするための改良法が提案されている^{16),17)}が、計算の複雑さがさらに増加するという問題がある。

3. オブジェクト領域の定義

3.1 オブジェクト領域と背景領域の定義

イラストやクリップアートなど2Dのベクタ画像から、計算機によって手続き的にオブジェクト領域を抽出するためには、事前に画像中のオブジェクト領域あるいは背景領域を定義しておく必要がある。しかし、画像中のどの部分がオブジェクト領域あるいは背景領域となるかは人間の主観や感性、過去の経験や知識が作用するため、主観が解明されていない現段階で、オブジェクト領域や背景領域を厳密に定義することは非常に困難と思われる。そこで本論文では、オブジェクト領域と背景領域を客観的に判別するための基準として「幅のある輪郭線によって囲まれる領域およびその輪郭線領域」をオブジェクト領域と定性的に定義することにする。また、「オブジェクト領域以外の領域」を背景領域と定義することにする。

このように定義すると、図1のような、有限幅の輪郭線を持つ画像に対し、オブジェクト領域が抽出可能となる。一方で、図2に示したような、幅のある輪郭線を持たないベクタ画像からはオブジェクト領域を抽出できない。たとえば、図2(a)に示す画像では、直観的には人物やノートパソコンを表している領域がオブジェクト領域になると感じられるが、これらの領域の輪郭はエッジ（幅のない線）により表現されているため、定義に従うとオブジェクト領域と判断することができない。本論文は、ベクタ画像のオブジェクト領域抽出の最初の試みとして、対象を幅のある輪郭線を持つベクタ画像に限定する。

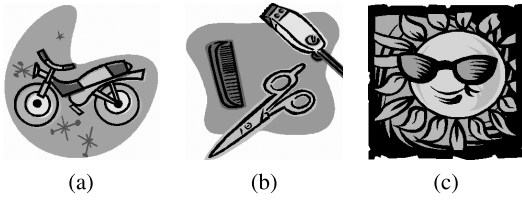


図 1 幅のある輪郭線を持つベクタ画像

Fig. 1 Vector images which have finite-width contours.

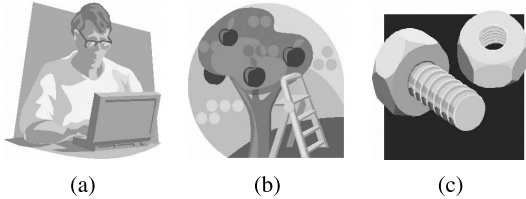


図 2 幅のない輪郭線を持つベクタ画像

Fig. 2 Vector images which have widthless contours.

3.2 輪郭線の判定条件

オブジェクト領域を手続き的に抽出するためには、オブジェクト領域の輪郭線部分を識別する必要がある。本研究では、輪郭線領域を識別するために必要な画像特徴の条件をデータに基づき導出することとした。

そこで、5人の被験者に対し、オブジェクト領域の定義を伝えただけで、1人につき50枚のベクタ画像を順に提示し、オブジェクト領域を手動抽出してもらい、さらに、オブジェクト領域の中の輪郭線領域を手動抽出してもらった。被験者には、画像素材集に含まれる約19万枚のベクタ画像(クリップアート)の中からランダムに選択しながら提示した。ただし、被験者がオブジェクト領域がないと判断した画像はスキップして、次の画像を提示した(オブジェクト領域があると判断した画像が50枚になるまで提示した)。被験者にオブジェクト領域とその輪郭線領域を手動抽出してもらうために使用したソフトウェアのスクリーンショットと抽出過程を図3に示す。画面左フレームにはオリジナルのベクタ画像が表示されている。また、右フレームには、初めはオリジナルのベクタ画像と同じ画像が表示されている。被験者が右フレーム内の画像上の1点をマウスでクリックすると、その点を表現している図形要素が右フレーム内から消去される。この操作を繰り返すことで最終的に、右フレームから背景領域を表す図形要素をすべて消去してもらい、オブジェクト領域のみを残してもらった。次に、オブジェクト領域の中で、輪郭線を表現している線領域をマウスでクリックしてもらった。以上の操作により、輪郭



図 3 オブジェクト領域の手動抽出過程

Fig. 3 A process of manual object extraction.

線を表現している図形要素のみをすべて抽出した。

一般に幅のある輪郭線はオブジェクト領域の境界を際立たせる目的で描かれることから、輪郭線領域はその周囲の領域に比べはっきりと違いの分かる色で描かれている可能性が高い。また、輪郭線領域は線状の領域なので、輪郭線を表現している図形要素は、幅が狭く、かつ、長さは長いことが期待できる。そこで、抽出した輪郭線を表現している各図形要素に対し、以下に示す3種類の画像特徴量、

- 可視領域の色差 Δ_C
- 可視領域の幅 W
- 可視領域を囲む矩形の長さ L

を測定した。ここで、図形要素の可視領域とは、図形要素の領域の中で、上に重なっている図形要素によって隠される部分を取り除いた領域のことである。可視領域を求める理由は3.3節で述べる。また、可視領域を囲む矩形とは、可視領域を内部に包含する最小の矩形である。各特徴量の定義は以下のとおりである。

可視領域の色差 Δ_C は、図形要素 p の可視領域に接するすべての図形要素の集合 Q とすると、

表 1 画像特徴量の測定結果

Table 1 Results of the measurements of the image features.

	Δ_C	W	L
min	75.2	2 (0.004X)	21 (0.044X)
max	160.5	30 (0.065X)	480 (1.000X)

$$\Delta_C = \max_{q \in Q} |C_p - C_q| \quad (1)$$

で定義される．ここで、 $||$ はベクトルの大きさを表し、 C_p および C_q は図形要素 p および図形要素 q の領域を表現している色の $L^*a^*b^*$ 色空間¹⁹⁾ 上での位置ベクトルを表す．

可視領域の幅 W を求めるために、可視領域に対し Hilditch の細線化処理²⁰⁾ を施す．Hilditch の細線化処理は、領域の連結性を保持したまま、線の幅が 1 になるまで領域の周囲から 1 画素幅ずつ内側へと繰り返し画素を削除していく処理である．そのため、領域の幅を W 画素とすると領域の幅が 1 になるまでに必要な削除処理の繰り返し回数 N は $N \simeq W/2$ となることが知られている．この性質を利用し、可視領域の幅 W は、

$$W = 2N \quad (2)$$

で定義される値とする．

可視領域を囲む矩形の長さ L は、可視領域を囲む矩形の幅 L_W 、高さ L_H とすると

$$L = \max\{L_W, L_H\} \quad (3)$$

で定義される．

抽出した輪郭線領域を表現している図形要素に対し測定した 3 種類の画像特徴量の最小値と最大値を表 1 に示す． W と L の値は、ベクタ画像のサイズを縦横ともに 480 ピクセルで規格化したときの値である．また、括弧内の値はベクタ画像のサイズを縦横ともに X としたときの比率を表す．これらのデータに基づき、任意の図形要素が輪郭線を表現していると判断されるために必要な条件を以下のように定義する．

$$70 \leq \Delta_C \leq 165 \quad (4)$$

$$0.003X \leq W \leq 0.070X \quad (5)$$

$$0.040X \leq L \leq 1.000X \quad (6)$$

これら 3 条件をすべて満たす可視領域を持つ図形要素を、輪郭線を表現している図形要素と判断することにする．

3.3 輪郭線判定において可視領域に着目する理由

ベクタ画像において、輪郭線を表現するための図形

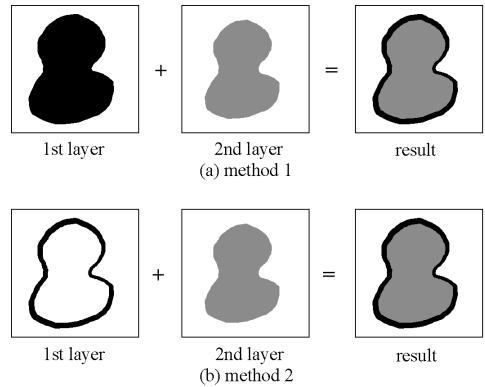


図 4 輪郭線を表現する 2 種類の方法
Fig. 4 Two methods for representing contours.

要素の構成方法は 2 種類ある．1 つは、複数の図形要素の重ね合わせによって輪郭線を表現する方法（以後、方法 1 と呼ぶ）である．もう 1 つは、独立した 1 つの図形要素によって輪郭線を表現する方法（以後、方法 2 と呼ぶ）である．具体例を図 4 に示す．(a) および (b) は方法 1 および方法 2 により輪郭線が表現されている例を示している．(a) では第 1 層（下層）に内部を黒で塗りつぶした図形要素が置かれ、第 2 層（上層）に第 1 層よりも少し小さな図形要素を重ねている．その結果、第 1 層の領域の外側部分が輪郭線として表現されている．一方、(b) では第 1 層にそれ単独で輪郭線を表す図形要素が設定されている．これら 2 つの例において、第 1 層の図形要素に対し可視領域を求めると、可視領域はどちらも (b) の第 1 層と同じ領域となる．すなわち、可視領域を求めることで、2 種類の輪郭線の表現方法の違いが吸収される．輪郭線判定において可視領域に着目する理由は、どちらの方法が使われていようと、統一的な方法で輪郭線かどうか判定可能となるからである．

4. プリミティブ選択モデルに基づくオブジェクト領域抽出法

4.1 対象とするベクタ画像の条件

本論文で扱うベクタ画像は、各図形要素が単色で表現されていることが条件である．一般にベクタ画像では、1 つの図形要素に対しブラシとペンという 2 種類の色属性を指定できることが多い．ブラシは図形要素が表す領域の内部を塗りつぶす色を表し、ペンは領域の輪郭を表す色を表す．これら 2 種類の色属性が指定された図形要素を持つベクタ画像に対しては、以下の前処理を実行することで本論文で扱うベクタ画像の条件を満たすことができる．

L^* の値域 [0, 100], a^* の値域 [-134, 220], b^* の値域 [-140, 122] .

2種類の色属性が指定されている図形要素に対し、この図形要素と同じ形状を持つ図形要素を2つ作成する。このとき一方は、ブラシを透明色、ペンを元の図形要素のペンの色と同じ色に設定する。もう一方は、ブラシを元の図形要素のブラシの色と同じ色に、ペンを透明色に設定する。透明色は実際に表示される時に見えないので、これら2つの図形要素はそれぞれ1つの色属性を持つ図形要素と見なすことができる。さらに、これら2つの図形要素を重ね合わせることで元の図形要素と見た目は等しくなる。したがって、元の図形要素を、新しく作成したこれら2つの図形要素に置き換えることで、本論文で扱うベクタ画像の条件を満たすことができる。

4.2 オブジェクト領域抽出の流れ

ベクタ画像は図5(a)に示すように、複数の図形要素で構成されている。提案手法は、まず、これらの図形要素群から輪郭線領域を表現している図形要素(図5(b)では太枠に囲まれた図形要素)をすべて選択し、これら選択した図形要素のみを統合し、輪郭線領域を抽出する。さらに、抽出した輪郭線領域の外側の境界を抽出する(図5(c))。次に、抽出した境界の内側のみに可視領域が含まれる図形要素(図5(d)では実線の太枠で囲まれた図形要素)をすべて選択し、これらを統合することでオブジェクト領域を得る。同様に、可視領域の一部あるいは全部が境界の外側に存在する図形要素(図5(d)では破線の太枠で囲まれた図形要素)をすべて選択し、これらを統合することで背景領域を得る。

本手法は、図形要素の選択と統合に基づきオブジェクト領域を抽出することから、本論文ではこの方式をプリミティブ選択モデル(Primitive Selection Model)と呼ぶことにする。

4.3 輪郭線領域の抽出

輪郭線を表現している図形要素を判断するために、各図形要素に対し3章で述べた3つの画像特徴量(可視領域の色差 Δ_C 、可視領域の幅 W 、可視領域を囲む矩形の長さ L)を抽出し、これらの画像特徴量がそれぞれ式(4)~(6)を満たすかどうかを判定する。すべての条件を満足する図形要素を統合することで、輪郭線領域を得ることができる。

4.4 オブジェクト領域と背景領域の抽出

オブジェクト領域と背景領域を抽出するために、あらかじめ輪郭線領域から輪郭線領域の外側の境界を抽出しておく。次に、各図形要素に対し、図形要素の可視領域が境界の内側のみに存在するか否か(外側に存在する)を判定する。内側のみに存在すると判定され

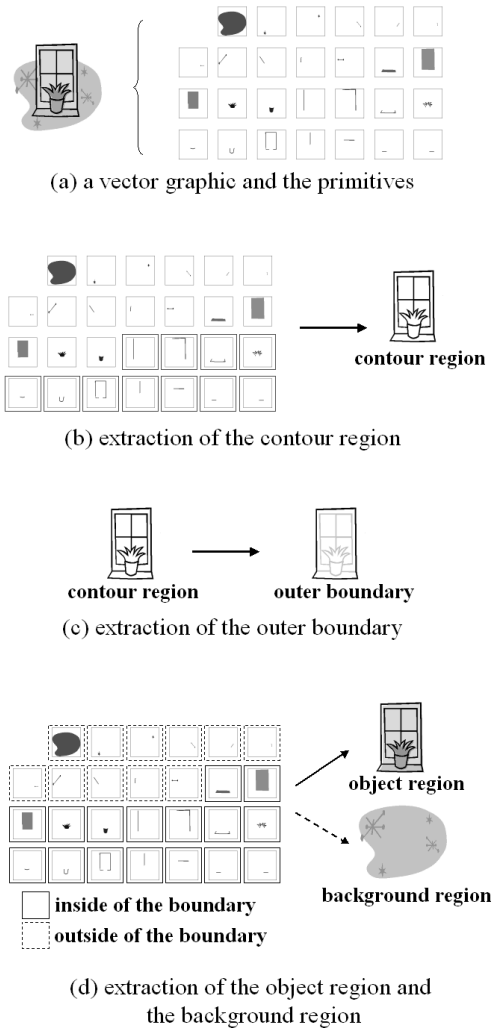


図5 オブジェクト領域の抽出処理の流れ
Fig. 5 Process flow of extraction of object regions.

た図形要素をすべて統合することでオブジェクト領域が抽出できる。同様に、外側に存在すると判定された図形要素を統合することで背景領域が抽出できる。

ただし、この方法の場合以下のような問題点がある。(問題点1)図6(a)に示すように、輪郭線領域が完全には閉じていない場合でも、人間は図6(b)のように閉じた主観的境界²¹⁾を覚覚し、境界の内側と外側を判断する。しかし、計算機は図6(c)のように輪郭線領域の物理的な境界に基づき内側と外側を判断するため、人間の判断と一致しない。(問題点2)図7(a)に示すように、輪郭線内部の塗りつぶしを表現している図形要素(図では primitive 2)が境界からわずかにみ出している場合でも、人間はこの図形要素をオブジェクト領域を表現していると判断する(図7(b))。しかし、計算機は

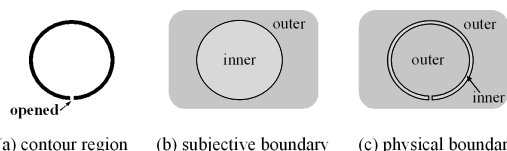


図 6 輪郭線領域が完全には閉じていない場合の主観的境界と物理的境界

Fig. 6 Subjective boundary and physical boundary in case that a contour region is not completely closed.

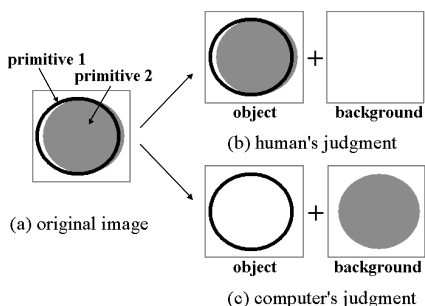


図 7 塗りつぶし領域の一部が境界からわずかにみ出している場合の人間と計算機の判断の違い

Fig. 7 Difference between human's and computer's judgments in case that part of a filled region is excluded from the boundary.

はみ出し部分を境界の外側と判断するため、この図形要素を背景領域を表現していると判断する (図 7(c))。これは人間の判断と一致しない。

この 2 つの問題点を解決するために、オブジェクト領域抽出の前段階である境界を抽出する処理を改良する。具体的には、輪郭線領域に対し、半径 r の塗りつぶし円によるダイレーション²²⁾を実行する。ダイレーションは輪郭線領域内のすべての点上を円の中心が動き回ったときの、この円内の点が描く軌跡を求める処理である。この軌跡は輪郭線領域が外側に r 画素分だけ膨張したような領域となる。たとえば、図 8(a) に示す輪郭線領域に対しダイレーションを求めると、図 8(b) のように輪郭線領域が膨張したような結果が得られる。輪郭線領域を構成している画素の座標集合を A 、半径 r の塗りつぶし円の領域 (円の中心を座標 $(0, 0)$ とする) を構成している画素の座標集合を B としたとき、領域 A に対し、領域 B によるダイレーションを求める演算は $A \oplus B$ と記され、

$$A \oplus B = \{c \in Z^2; c = a + b, a \in A, b \in B\} \quad (7)$$

と定義される (Z は整数の集合を表す)。ダイレーションを実行することで輪郭線は閉じる (図 8(c))。ダイレーションによって膨張した輪郭線領域の物理的境界を、あらためて輪郭線領域の境界と考えると、この境界は人間が判断する主観的境界 (図 6(b)) とほぼ一

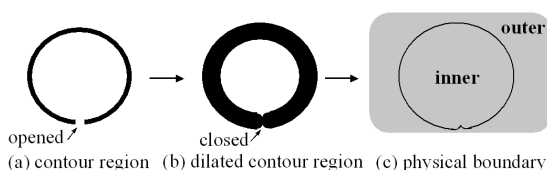


図 8 ダイレーションによる輪郭線領域の膨張

Fig. 8 Expansion of the contour region by dilation.

致する (問題点 1 の解消)。

また、図 7(a) のように塗りつぶし領域の一部が、境界 (ダイレーション実行前の境界) よりもわずかにみ出している場合でも、そのはみ出し幅が r 以下であれば、新しい境界 (ダイレーション実行後の境界) の内側に収まる。その結果、オブジェクト領域を表現していると判定される。これは人間の判断と一致する (問題点 2 の解消)。

適切な r を求めるため、3 章で述べた 5 人の被験者に抽出してもらったオブジェクト領域に対し、内部の図形要素が境界位置から外側の法線方向に何ピクセル分はみ出しているかを測定した。その結果、はみ出し幅の最大値は 23 ピクセルであった。画像サイズを X とするとこの値は約 $0.048X$ である。このことから、ダイレーションで使用する円領域は半径 $r = 0.05X$ 程度が適当と考えられる。

5. 評価実験

5.1 実験目的

評価実験として、人間が抽出したオブジェクト領域と計算機が抽出したオブジェクト領域とを比較し、計算機が抽出したオブジェクト領域の正確さを評価する。計算機はプリミティブ選択モデル (以後、PSM と表記)、ACM の 2 種類の手法を使用する。また、PSM、ACM を使用したときのオブジェクト領域抽出に要した時間を測定する。抽出精度と実行時間の比較結果から、PSM と ACM の有効性と限界を考察する。

5.2 実験方法

まず、40 枚のベクタ画像を使用し、10 人の人間 (被験者) にオブジェクト領域の定義を伝えたくてオブジェクト領域を手動抽出してもらった。使用した 40 枚のベクタ画像は、この 10 人とは別の 2 人の人間が 4,208 枚のベクタ画像を見て、オブジェクト領域と背景領域をともに含むと判断した 398 枚の画像からランダムに選択したものである。実験で使用したベク

これらは Microsoft Office Online (<http://office.microsoft.com/clipart/>) からダウンロード可能なクリップアートである。

タ画像の画像フォーマットは Windows で標準的に使用されている wmf をコンバータにより svg に変換したものである。コンバータには wmf2svg というソフトウェアを使用した。wmf はバイナリデータ、svg は XML 形式のテキストデータという違いがある。svg を使用した理由は、プログラムの作成および保守、画像の解析を容易にするためである。

10 人の被験者にオブジェクト領域を手動抽出してもらったために使用したソフトウェアは 3 章 (図 3) で示したものと同一である。10 人の被験者による抽出結果の中で、最頻の結果を「人間が抽出したオブジェクト領域」と定義した。

次に、40 枚のベクタ画像に対し、PSM および ACM を使用し、オブジェクト領域の自動抽出を行った。ACM のパラメータの設定条件は付録に示したとおりである。

次に、人間が抽出したオブジェクト領域 R_H と、計算機 (PSM または ACM) が抽出したオブジェクト領域 R_C との類似性を評価するため、次式で定義される抽出効率 $S_f(R_H, R_C)$ 、抽出精度 $S_p(R_H, R_C)$ 、抽出再現率 $S_r(R_H, R_C)$

$$S_f(R_H, R_C) = \frac{2S_p(R_H, R_C)S_r(R_H, R_C)}{S_p(R_H, R_C) + S_r(R_H, R_C)} \quad (8)$$

$$S_p(R_H, R_C) = \frac{N(R_H \cap R_C)}{N(R_C)} \quad (9)$$

$$S_r(R_H, R_C) = \frac{N(R_H \cap R_C)}{N(R_H)} \quad (10)$$

を測定した。ここで、 $N(R)$ は領域 R の画素数を表す。また、 $R_H \cap R_C$ は領域 R_H と R_C の共通領域を表す。抽出精度 $S_p(R_H, R_C)$ は、計算機が抽出したオブジェクト領域中、正解領域 (人間が抽出したオブジェクト領域) が占める割合、抽出再現率 $S_r(R_H, R_C)$ は、全正解領域 (人間が抽出したオブジェクト領域) 中、計算機によって抽出できた領域の割合を表す。抽出効率 S_f は、抽出精度と抽出再現率の調和平均により定義される値であり、抽出した領域の正確さを総合的に評価する値である。

また、PSM と ACM のオブジェクト領域抽出に要した時間をそれぞれ測定した。ベクタ画像のサイズは 240×240 で規格化した。また、使用した計算機は Pentium M 1.3 GHz、メモリ 1.5 GB である。

5.3 抽出効率, 抽出精度, 抽出再現率の測定結果と考察

抽出効率, 抽出精度, 抽出再現率の測定結果を表 2

表 2 抽出効率 S_f , 抽出精度 S_p , 抽出再現率 S_r の平均値と標準偏差

Table 2 Means and standard deviations of extraction performance S_f , extraction precision S_p and extraction recall S_r .

	S_f	S_p	S_r
PSM	$\mu = 0.9879$ $\sigma = 0.0068$	$\mu = 0.9857$ $\sigma = 0.0093$	$\mu = 0.9932$ $\sigma = 0.0067$
ACM	$\mu = 0.9161$ $\sigma = 0.0129$	$\mu = 0.8867$ $\sigma = 0.0141$	$\mu = 0.9615$ $\sigma = 0.0164$

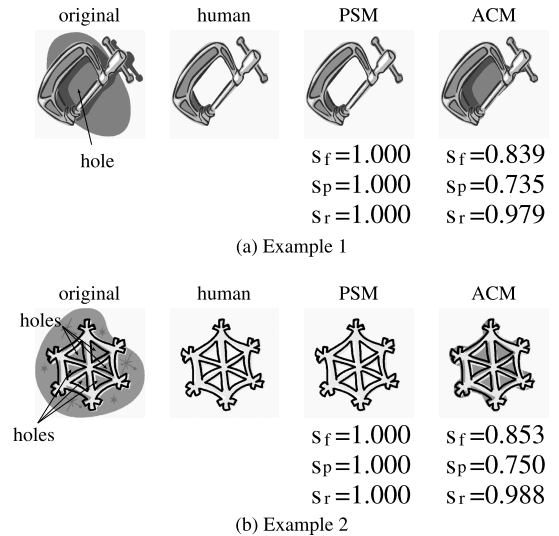


図 9 抽出結果例 1
Fig. 9 Examples of extraction results 1.

に示す。表中の μ と σ はそれぞれ 40 枚のベクタ画像に対する測定値の平均値と標準偏差を表す。この結果から PSM の方が ACM よりも正確にオブジェクト領域を抽出できていることが確認できる。このような結果を得た要因を考察するため、実際に人間および計算機 (PSM と ACM) が抽出したオブジェクト領域を調査した。

まず、PSM と人間の抽出結果が一致 ($S_f = 1$) したもののうち、ACM と人間の抽出結果が異なった ($S_f < 1$) ものを図 9 に示す。図の左側から順にオリジナルのベクタ画像、人間が抽出したオブジェクト領域、PSM により抽出されたオブジェクト領域、ACM により抽出されたオブジェクト領域を表している。また、PSM と ACM の抽出結果の下には抽出効率 S_f 、抽出精度 S_p 、抽出再現率 S_r の各測定値を示した。(a), (b) のベクタ画像はオブジェクト領域にホールが存在し、ホールから背景領域が見えているという特徴がある。これらのベクタ画像のオブジェクト領域の境界に着目したとき、ACM は外側の境界は正しく抽出

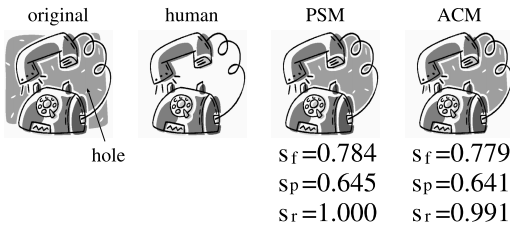


図 10 抽出結果例 2
Fig. 10 Examples of extraction results 2.

できるが、内側の境界（オブジェクト領域とホールとの間の境界）は抽出できない。その結果、ホールから見えている背景領域はオブジェクト領域の一部として誤抽出され、抽出精度が小さくなる。これは、ACM が単一の閉曲線によってオブジェクト領域の境界を表現しようとするために生じる現象である。一方で、PSM ではホールから見えている背景領域をオブジェクト領域から分離できている。これは内側の背景（ホール部分から見えている背景）と外側の背景（オブジェクト領域の外側にある背景）が、同じ 1 つの図形要素で表現されているからである。この背景領域（模様部分は除く）を表現している図形要素は最下層に設定されており、他の図形要素によって領域が上書きされている。結果として、ホール部分と、オブジェクト領域の外側のみが最後まで上書きされずに残って見えている。PSM では、図形要素が背景として選択される条件を、可視領域が（一部でも）オブジェクト領域の境界線よりも外側に存在すること、としているため、この最下層の図形要素は背景として選択される。内側から見えていた背景領域は、実は、外側の背景と最下層でつながっているため、結果として背景として選択される。このことは、図形要素単位でオブジェクト領域抽出を行う PSM の優位性を示す一例と考えられる。

オブジェクト領域にホールが存在するベクタ画像であっても、PSM の抽出精度が小さくなる場合も存在した。図 10 はその一例である。この例では PSM, ACM とともにホールから見えている背景領域をオブジェクト領域の一部として抽出してしまった。図 9 に示したベクタ画像との相異点は、ホールから見えている背景領域とオブジェクト領域の外側の背景領域とで異なる図形要素が使用されている点である。PSM では輪郭線領域の外側の境界に着目し、その内側のみに可視領域が含まれる図形要素をオブジェクト領域を表現する図形要素と判定するため、ホールから見えている背景領域を表現している図形要素はオブジェクト領域を表現すると判定されてしまう。このことは、現時点での PSM の限界を示す一例と考えられる。この問題を回

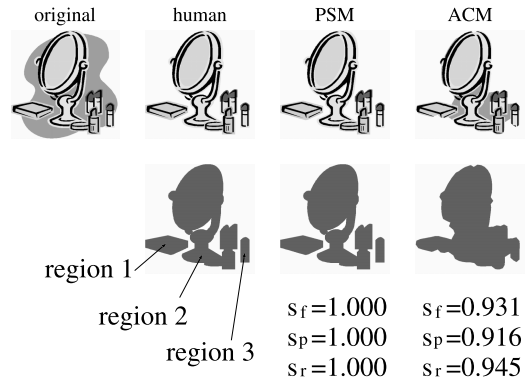


図 11 抽出結果例 3（ベクタ画像が複数の独立したオブジェクト領域を持つ場合）

Fig. 11 Examples of extraction results 3 (in the case that a vector image has some discrete object regions).

避するためには、オブジェクト領域を表現していると判定された図形要素の中から、外側の背景領域と同じ色やテクスチャが使用されている図形要素がないか調べるなど追加処理が必要である。

ベクタ画像中に複数の独立したオブジェクト領域が存在する場合の抽出結果例を図 11 に示す。人間, PSM, ACM によるそれぞれの抽出結果の下に、オブジェクト領域をすべて単一色で表現した画像（マスク画像と呼ぶ）を示した。人間が抽出したオブジェクト領域のマスク画像からこのベクタ画像が 3 つの独立したオブジェクト領域を持つことを確認できる。PSM のマスク画像を見ると、これら 3 つのオブジェクト領域が正しく抽出されていることが確認できる。しかし、ACM のマスク画像を見ると、これら 3 つのオブジェクト領域はまとめて 1 つのオブジェクト領域として抽出されている。これは ACM が単一閉曲線によりオブジェクト領域を表現するからである。ACM の改良法として、複数のオブジェクト領域を取得するための手法が研究されている^{16),17)}。これらの手法を使用することでこの問題はある程度回避可能と考えられる。しかし、計算の複雑さは単一閉曲線を使用する場合よりもさらに増加する。

5.4 実行時間の測定結果と考察

PSM と ACM の実行時間の平均値と標準偏差を表 3 に示す。この結果から平均的には PSM の方が ACM よりも短時間でオブジェクト領域を抽出できることが確認できる。個々のベクタ画像に対する PSM と ACM の実行時間の関係を求めると図 12 のようになった。図の縦軸は ACM による実行時間、横軸は PSM による実行時間を表し、図中の各サンプル点は各ベクタ画像に対応している。また図中の一点鎖線は PSM と

表 3 実行時間 (秒)
Table 3 CPU time (sec).

	PSM	ACM
Mean	11.821	18.975
Standard Deviation	5.687	6.487

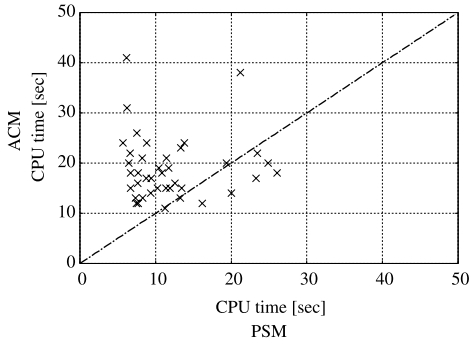


図 12 各ベクタ画像に対する実行時間
Fig. 12 CPU times of each vector image.

ACM の実行時間が等しくなる直線を表している (この直線よりも上側に存在するサンプル点は PSM の方が実行時間が小さくなり、この直線よりも下側に存在するサンプル点は ACM の方が実行時間が小さくなる)。この結果から、必ずしも PSM がつねに高速にオブジェクト領域を抽出できるとは限らないといえる。そこで、ACM と PSM の実行時間を決める要因を調べることにした。

ACM の実行時間を大きくする要因として考えられるのが、閉曲線の変形回数である。そこで、ACM の実行時間と閉曲線の変形回数との関係を調べた。結果は図 13 のようになった。図の縦軸は ACM による実行時間、横軸は閉曲線の変形回数を表す。また、図中の直線は各サンプル点を直線で最小二乗近似したものである (平均二乗誤差 2.1952)。図から実行時間が閉曲線の変形回数とほぼ比例関係にあることが確認できる。閉曲線の変形回数はオブジェクト領域の大きさや、境界の形状に依存する。一般にオブジェクト領域が小さいものや、形状が入り組んでいて複雑なものほど、閉曲線が境界にフィットするまでに時間がかかる。しかし、エネルギーの定義や閉曲線の変形方法を適切に設計すれば、閉曲線の変形回数の上限を有限の値に抑えることができるため、実行時間の上限は固定できると考えられる。

PSM の実行時間を大きくする要因として考えられるのが、ベクタ画像を構成する図形要素の数である。そこで PSM の実行時間と図形要素の数との関係を調べた。結果は図 14 のようになった。図の縦軸は PSM による実行時間、横軸はベクタ画像を構成する図形

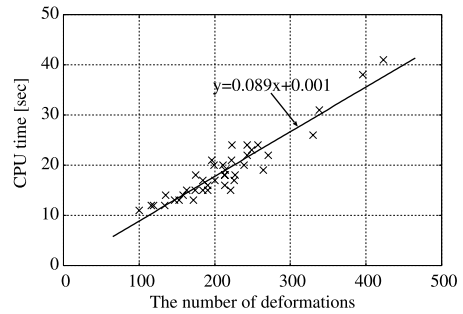


図 13 ACM の実行時間と閉曲線の変形回数との関係
Fig. 13 Relation between CPU times of the ACM and the number of deformations.

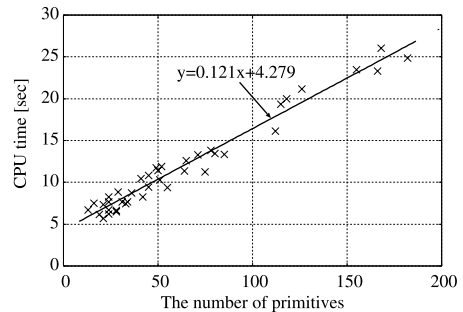


図 14 PSM の実行時間と図形要素の数との関係
Fig. 14 Relation between CPU times of the PSM and the number of primitives.

要素の数を表す。また、図中の直線は、各サンプル点を直線で最小二乗近似したものである (平均二乗誤差 1.0877)。図から実行時間が図形要素の数とほぼ比例関係にあることが確認できる。PSM は図形要素ごとにオブジェクト領域を表現しているかどうかを判定するため、この結果は妥当なものといえるが、図形要素の数が増加するにつれ実行時間が増加するという特徴は、ACM には見られないものである。本実験で使用したベクタ画像は多くとも 200 個程度の図形要素で構成されており、80% (32/40) のサンプルで PSM の実行時間の方が小さくなった。しかし、図形要素の数が多くなるベクタ画像を扱おうとすると、ACM の方が実行時間の点で優位になると考えられる。したがって、PSM の高速化は今後の課題の 1 つとなると思われる。

6. おわりに

本論文では、ベクタ画像のオブジェクト領域を定義し、図形要素の選択と統合に基づくオブジェクト領域抽出法であるプリミティブ選択モデル (PSM) を提案した。また、PSM の有効性と限界を評価するため、既存手法である ACM との比較実験を行い、以下の知見を得た。

- (1) PSMの方がACMよりもオブジェクト領域の抽出効率が高い。
- (2) オブジェクト領域内にホールが存在する場合、ACMはオブジェクト領域を正しく抽出できない。それに対し、PSMは、ホールから見えている背景領域とオブジェクト領域の外側の背景領域が同一の図形要素で表現されている場合、オブジェクト領域を正しく抽出できる。
- (3) PSMは複数の独立したオブジェクト領域を正しく抽出できるが、単一閉曲線を使用するACMでは正しく抽出できない。
- (4) PSMの実行時間は図形要素の数に比例するため、図形要素の数が多いベクタ画像に対しては、実行時間の点でACMの方が優位になる場合がある。

現時点で、オブジェクト領域にホールが存在する場合で、ホールから見えている背景領域とオブジェクト領域の外側の背景領域が異なる図形要素によって表現されている場合、PSMを使用してもオブジェクト領域を正しく抽出できないという問題が残されている。しかし、PSMによって、オブジェクト領域の外側の背景は少なくとも検出可能である。よって、今後は、検出した外側の背景部分から図形要素がグループ化されるパターンを動的に抽出し、抽出した背景のパターンを利用し、ホールの内部の背景を抽出することを検討していきたい。

また、現時点でベクタ画像を構成する図形要素が多い場合の実行時間の改善が問題として残されている。これは、各図形要素をラスタライズして各処理を実行していることが原因と考えられる。図形要素が表現する図形の種類（ポリゴン、楕円、曲線など）によってベクタデータの定義方法が異なるため、すべての図形要素を統一的に扱うことは容易ではないが、速度性能の向上へ向けてベクタ画像をベクタデータのままで扱う技術の開発が今後必要と思われる。

オブジェクト領域抽出は、類似ベクタ画像検索（たとえば、類似ロゴマーク検索など）の前処理として利用可能である。本論文で定義した輪郭線は制約が大きいため、全自動での輪郭線抽出を目指す場合、輪郭線のモデルを拡張し、より制約を小さくしていく必要がある。しかし、人間の視覚認知機能が完全に解明されていない現段階で、より汎用性の高い輪郭線のモデルを作成することは困難と思われる。そこで今後は、輪郭線部分を人間が指定して、指定した輪郭線からモデルを学習し、これと同じ属性の線を計算機に検出させるという半自動の輪郭線抽出のアプローチなどを検討していきたい。

また、背景領域検出のためのアプローチとして、輪郭線の情報を使わず、色やパターンの同一性を考慮し、背景と思われる図形要素をグループ化することで背景領域を判定するというアプローチも考えられる。あらゆるベクタ画像に利用可能な背景モデル（図形要素がグループ化される条件）を事前に定義することは困難であるため、現時点ではこのアプローチは採用していない。しかし、このアプローチは、本論文で扱わなかった幅のない輪郭線（図2など）を持つベクタ画像に対処するために有望なアプローチと考えられ、今後検討していく必要がある。

謝辞 本研究の一部は、文部科学省科学研究費補助金（若手研究B，課題番号16700089）の支援により行われた。

参考文献

- 1) 常盤 繁：マルチメディアデータ入門，コロナ社（2003）。
- 2) Proberts, S., Mong, J., Evans, D. and Brailsford, D.: Vector Graphics: From PostScript and Flash to SVG, *Proc. 2001 ACM Symposium on Document Engineering*, pp.135–143 (2001).
- 3) Fonseca, M.J., Barroso, B., Ribeiro, P. and Jorge, J.A.: Retrieving Vector Graphics Using Sketch, *Proc. Intl. Sympo. Smart Graphics*, LNCS 3031, pp.66–76 (2004).
- 4) Fonseca, M.J., Barroso, B., Ribeiro, P. and Jorge, J.A.: Retrieving ClipArt Images by Content, *Proc. Intl. Conf. Image and Video Retrieval*, LNCS 3115, pp.500–507 (2004).
- 5) Sciascio, E.D., Donini, F.M. and Mongiello, M.: A Knowledge Based System for Content-Based Retrieval of Scalable Vector Graphics Document, *Proc. ACM Sympo. Applied Computing*, pp.1040–1044 (2004).
- 6) Choi, C., Chang, Y., Cho, J. and Kim, S.: Extract Shape from Clipart Image Using Modified Chain Code — Rectangle Representation, *Proc. Intl. Conf. Computational Science and Its Applications*, LNCS 3046, pp.251–260 (2004).
- 7) Flickner, M., Sawhney, H., Niblack, W., Ashley, J., Huang, Q. and Dom, B.: Query by Image and Video Content: The QBIC System, *IEEE Computer*, Vol.28, No.9, pp.23–32 (1995).
- 8) Smith, J.R. and Chang, S.F.: VisualSEEK: A Fully Automated Content-Based Image Query System, *Proc. ACM Multimedia*, pp.87–98 (1996).
- 9) Chalehcale, A., Naghdy, G. and Mertins, A.:

- Sketch-Based Image Matching Using Angular Partitioning, *IEEE Trans. Syst. Man. Cybern.*, Vol.35, No.1, pp.28–41 (2005).
- 10) Herault, L. and Horaud, R.: Figure-Ground Discrimination: A Combinatorial Optimization Approach, *IEEE Trans. Pattern Recognition and Machine Intelligence*, Vol.15, No.9, pp.899–914 (1993).
- 11) Zhao, L. and Davis, L.S.: Iterative Figure-Ground Discrimination, *Proc. Intl. Conf. Pattern Recognition*, Vol.1, pp.67–70 (2004).
- 12) Amir, A. and Lindenbaum, M.: Ground from figure Discrimination, *Computer Vision Understanding*, Vol.1, No.76, pp.7–18 (1999).
- 13) Geiger, D., Kumarau, K. and Parida, L.: Visual Organization for Figure/Ground Separation, *Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition*, pp.155–160 (1996).
- 14) Kass, M., Witken, A. and Terzopoulos, D.: Snakes: Active Contour Models, *Intl. J. Computer Vision*, pp.321–331 (1988).
- 15) Amini, A.A., Weymouth, T.E. and C.Jain, R.: Using Dynamic Programming for Solving Variational Problems in Vision, *IEEE Trans. Pattern Recognition and Machine Intelligence*, Vol.12, No.9, pp.855–867 (1990).
- 16) Sethian, J.A.: *Level Set Methods and Fast Marching Methods*, Cambridge University Press (1999).
- 17) Malladi, R., Sethian, J.A. and Vemuri, B.C.: Shape Modeling with Front Propagation: A Level Set Approach, *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol.17, No.2, pp.158–175 (1995).
- 18) 串間和彦, 赤間浩樹, 紺谷精一, 木本晴夫, 山室雅司: オブジェクトに基づく高速画像検索システム: ExSight, *情報処理学会論文誌*, Vol.40, No.2, pp.732–741 (1999).
- 19) 小林光夫: 均等色空間 (I)—Munsell 色空間から CIELAB へ, *日本色彩学会誌*, Vol.26, No.2, pp.73–83 (2002).
- 20) Hilditch, C.J.: Linear Skeleton from Square Cupboards, *Machine Intelligence*, pp.403–420, Edinburgh Univ. Press (1967).
- 21) Kanizsa, G.: Subjective Contours, *Scientific American*, pp.48–52 (1976).
- 22) Gonzalez, R.C. and Woods, R.E.: Morphological Image Processing, *Digital Image Processing*, pp.519–566 (Chapter 9), Prentice Hall (2002).

付 録

ここでは, 評価実験で使用した ACM のパラメータ

設定条件を示す.

閉曲線のエネルギーが局所最適な値にできるだけラップされないように動的計画法に基づく方法¹⁵⁾を採用した.

また, 閉曲線は, N 個の制御点の集合により近似表現した. 制御点には 1 から N までの識別番号を付加し, これらを順に直線で結んだ多直線を閉曲線と定義した. N を大きくするほど制御点の間隔が狭くなり輪郭線が滑らかになるが, N を大きくしすぎると 2 つの問題が発生する. 1 つは, もし抽出したい輪郭線領域が完全には閉じていない場合 (図 6 (a) のような場合), 制御点が輪郭線の内側へと入り込んでくることである. 制御点が輪郭線の内側へと入り込むと, 閉曲線は物理的境界 (たとえば, 図 6 (c)) へと収束してしまい, 主観的境界 (たとえば, 図 6 (b)) とは大きく異なる境界を抽出してしまう. また, もう 1 つは, 各制御点ごとにエネルギー計算が必要になるため実行時間が増加することである. これらの問題を考慮し評価実験では $N = 100$ とした.

ACM で使用するエネルギー E は, 閉曲線の変形回数 t の関数として,

$$E(t) = E_{\text{int}}(t) + E_{\text{ext}}(t) \quad (11)$$

$$E_{\text{int}}(t) = 0 \quad (12)$$

$$E_{\text{ext}}(t) = \sum_{s=1}^N \{E_{\text{img}}(s, t) + E_{\text{con}}(s, t)\} \quad (13)$$

$$E_{\text{img}}(s, t) = \Delta_c(\mathbf{v}(s, t))^{-1} \quad (14)$$

$$E_{\text{con}}(s, t) = \begin{cases} 1 & \text{(if } \mathbf{v}(s, t) = \mathbf{v}(s, t-1) \\ & \& \Delta_c(\mathbf{v}(s, t-1)) < 70) \\ 1 & \text{(if } \mathbf{v}(s, t) \neq \mathbf{v}(s, t-1) \\ & \& \Delta_c(\mathbf{v}(s, t-1)) \geq 70) \\ 0 & \text{(otherwise)} \end{cases} \quad (15)$$

で定義した. ここで, $\mathbf{v}(s, t)$ は, 閉曲線の変形を t 回行った段階での, s 番目の制御点の位置ベクトルを表す. E_{int} は内部エネルギー, E_{ext} は外部エネルギーを表す. また, E_{img} および E_{con} はそれぞれ画像エネルギーおよび制御エネルギーと呼ばれ, 各制御点に対し定義されるエネルギー項である (後述). また, $\Delta_c(\mathbf{v}(s, t))$ は, 点 $\mathbf{v}(s, t)$ に対し,

$$\Delta_c(\mathbf{v}(s, t)) = \max\{\delta_c(\mathbf{v}(s, t), \mathbf{v}(s, t) + \xi\mathbf{n}(s, t)), \delta_c(\mathbf{v}(s, t), \mathbf{v}(s, t) - \xi\mathbf{n}(s, t))\} \quad (16)$$

により定義される. $\delta_c(\mathbf{v}_i, \mathbf{v}_j)$ は, 画像の位置 \mathbf{v}_i の色と, 位置 \mathbf{v}_j の色の $L^*a^*b^*$ 色空間における色差

(ユークリッド距離)である． $\mathbf{n}(s, t)$ は s 番目の制御点の単位法線ベクトルを表し， ξ は閉曲線を変形させるときの各制御点の移動量を表す．実験では $\xi = 2.0$ を用いた．

E_{int} は恒常的に 0 とした．これはベクタ画像のオブジェクト領域の境界の形状に対する制約を一般的にモデル化することが困難と考えたからである．

また，各制御点に対し定義される画像エネルギー E_{img} は，各制御点における色差の逆数とした．これは輪郭線領域は周囲の領域に比べて色差が大きい(3章)からである．

また，各制御点に対し制御エネルギー E_{con} を定義し，閉曲線の変形に対し制約を加えた．具体的には，輪郭線の変形回数が $t-1$ 回の段階で，制御点が色差 70 未満の画素の位置にいる(この制御点は輪郭線上にないと考えられる)にもかかわらず，次の変形(t 回目の変形)でこの制御点が移動しなかったときにペナルティを与えた($E_{\text{con}} = 1$)．また，閉曲線の変形回数が $t-1$ 回の段階で，制御点が色差 70 以上の画素の位置にいる(この制御点は輪郭線をとらえたと考えられる)にもかかわらず，次の変形(t 回目の変形)でこの制御点が移動してしまったときにペナルティを与えた($E_{\text{con}} = 1$)．

(平成 18 年 6 月 15 日受付)

(平成 18 年 12 月 7 日採録)



林 貴宏(正会員)

1975 年生．1998 年金沢大学工学部電気情報工学科卒業．2000 年同大学大学院自然科学研究科博士前期課程修了．2003 年同研究科博士後期課程修了．博士(工学)．2001 年石川工業高等専門学校電子情報工学科助手を経て，2003 年電気通信大学電気通信学部情報工学科助手，現在に至る．マルチメディアデータベース，情報検索，人工知能の研究に従事．IEEE，電子情報通信学会，日本ソフトウェア科学会，人工知能学会各会員．



尾内理紀夫(正会員)

1950 年生．1973 年東京大学理学部物理学科卒業．1975 年同大学大学院理学系大学院物理学専攻修士課程修了．同年日本電信電話公社(現 NTT)に入社．1982 年から 1985 年に ICOT プロジェクトに参画，1997 年から 1998 年に RWC プロジェクトに参画．2000 年より電気通信大学電気通信学部情報工学科教授．著書に『コンピュータの仕組み』(朝倉書店)，編書に『オブジェクト指向コンピューティング』(近代科学社)，『インタラクティブシステムとソフトウェア』(近代科学社)等．マルチメディア情報処理，情報検索，セマンティックコンピューティング等に興味を持つ．1996 年情報処理学会プログラミングシンポジウム山内奨励賞受賞．工学博士(東京大学)．日本ソフトウェア科学会，人工知能学会，ACM 各会員．