

再帰的空間分割法と部分空間分類 —自己交差，および重なりのある曲線への拡張—

森本 経 宇^{†,☆} 山口 和 紀^{††}

図形処理の方法として，扱う図形に応じて空間を分割し，その部分空間で図形を処理する適応型空間分割法がある。分割を再帰的に行う場合は再帰的空間分割法という。再帰的空間分割にはさまざまな可能性があり，どのような分割がどのような処理に有効かということは明らかになっていない。前研究では，再帰的空間分割法を近似をせずに曲線に直接適用し，曲線で囲まれる領域の内外判定が可能となる分類と終了条件を取り出した。ただし，そこでは，自己交差や重なりのある曲線を例外として除いていた。本論文では，線や交点などの用語を定義しなおすことにより，例外なく，内外判定アルゴリズムが適用可能であることを示した。実験結果から，この方法により，不要な空間分割が抑えられる一方，それぞれの部分空間が単純になり処理に適していることを示した。

Recursive Subdivision Methods and Subspace Classification —The Extension to Curves with Self-intersections and Overlaps—

KEIU MORIMOTO^{†,☆} and KAZUNORI YAMAGUCHI^{††}

In order to overcome the complexity of operations on surfaces and solids in advanced modeling, often employed is an adaptive, recursive space decomposition method called a recursive subdivision. In our previous paper, we investigated the existing recursive subdivision methods, and identified components of the method: surface approximation, subspace classification, termination criteria. Then, we developed a new classification of subspace with curves which are reasonably smooth and have no self-intersection or overlap point. And we applied it to the problem of identifying regions bounded with curves. In this paper, we first redefined some important terms such as 'intersection point' in the case of self-intersection and overlap, and refined the classification description based on the terms. Then, we applied it to the problem of identifying regions bounded with curves which have self-intersection and overlap points. Experimental results show that the new method brings up a space subdivision superior to the previous results.

1. 序 論

図形処理の方法として，扱う図形に応じて空間を分割し，その部分空間で図形を処理する適応型空間分割法 (adaptive space division) がある。分割を再帰的に行う場合は再帰的空間分割法 (recursive subdivision) という。2次元では，その表現を quadtree (4分木) と呼ぶ。

classical quadtree^{9),10),☆☆}は，曲線を境界とする領域を表現する場合，メモリの使用量が多い，境界の情報がないので法線ベクトルや交線が計算できない，などの問題点がある。これを改善するために，edge quadtree^{6),10)}や PM quadtree^{5),8)}が提案されたが，曲線を直線で近似して扱うために，法線ベクトルや交点などが正確に計算できないという問題がある。3次元では，face octree²⁾が提案されたが，曲面を平面で近似するため，面に不連続性が生じたり，比較的単純な部分空間が分割されたりするという問題がある。

このような従来の quadtree や octree 表現では，対象の近似，部分空間の分類，分割の終了条件の問題が混在している。この問題を整理するために，前論文¹⁴⁾

[†] 筑波大学大学院博士課程工学研究科

Doctoral Degree Program in Engineering, University of Tsukuba

[☆] 現在，シャープ株式会社

Presently with SHARP Corporation

^{††} 東京大学教養学部広域科学専攻広域システム科学系情報・図形科学

The College of Arts and Sciences, University of Tokyo

^{☆☆} 最初は quadtree と呼ばれたが，その後さまざまな変形が提案されたのでここでは classical quadtree と呼ぶ。

表1 classical quadtree のノードのラベルとその定義
Table 1 Definition of labels of classical quadtree nodes.

ノードのラベル	定義
WHITE	ノードに対応するクオドラントは領域にまったく含まれていない。
BLACK	ノードに対応するクオドラント全体が領域に含まれている。
GRAY	ノードに対応するクオドラントの一部が領域に含まれている。

では、近似、分類、終了条件に分けて検討し、近似を行わずに曲線で囲まれた領域を決定する内外判定のための分類、終了条件の構成を示した。そこでは、扱う曲線として、自己交差のあるものと、複数の曲線間で重なり部分のあるものを除いていた。本論文では、曲線の交線や交点などの用語を再定義し、前論文の分類とアルゴリズムを精密化することにより、自己交差がある場合や重なりがある場合でも、正しく扱える構成を示す。

2. 再帰的空間分割法

本章では、現在、提案されている再帰的空間分割法に関して述べる。再帰的分割法は提案された次元に従って、3次元なら octree, 2次元なら quadtree と呼ぶが、それぞれの考え方は必ずしも特定の次元に限定されないで、ここでは2次元に統一して述べる。以下では、曲線で囲まれた対象を領域と呼ぶことにする。

2.1 classical quadtree

classical quadtree¹⁰⁾は2次元空間内の領域を2次元空間の再帰的な4等分割で表現する方式である。xy座標系で classical quadtree を定義する場合は、xy座標系に1つ正方形を定義し、その内部のみを扱うことにする。この正方形を4分木の根に対応させる。正方形を各軸に関して2等分割することにより、全体を4つの正方形に分割する。各正方形には0~3までの番号を付け、分割前の正方形に対応する4分木のノードの4つの子ノードと対応させる。こうしてできた各正方形を分割前の正方形のクオドラント (quadrant) と呼ぶ。クオドラントは空間的概念であり、ノードは木構造の概念であるが、quadtree では両者が対応しているため、しばしば区別せずに使用する。

ノードには、そのノードに対応するクオドラントが、定義されている領域とどのような関係にあるかを表すラベルを付ける。表1に classical quadtree のノードのラベルとその定義を示す。

classical quadtree は BLACK ノードに対応するクオドラントの集まりで領域を近似的に表現するものである。図1に quadtree 表現と正方形の分割の例を

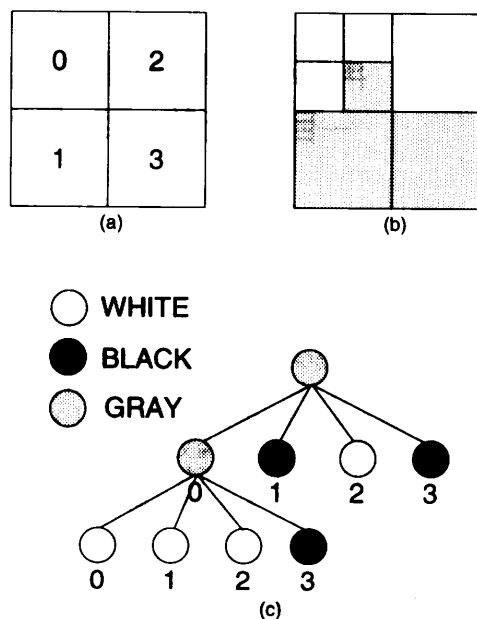


図1 classical quadtree: (a) クオドラントの番号付け; (b) quadtree 表現による領域分割; (c) 領域の木構造による表現
Fig.1 Classical quadtree: (a) quadrant numbering scheme; (b) region decomposition by quadtree; (c) its tree representation.

示す。

分割されるノードを非終端ノード、分割されないノードを終端ノードという。WHITE, BLACK はそれ以上分割しても表現の精度が良くならないので終端ノードとし、GRAY は非終端ノードとする。実用上は、無限に分割することを避けるために木の深さがあらかじめ指定された最大レベルに到達したときは、GRAY ノードでも終端ノードとする。

classical quadtree では、集合演算のアルゴリズムが形状のタイプに依存しないので簡単であるという利点がある。一方、曲線で囲まれた形状や xy 各軸に平行でない直線に囲まれた形状を表現する場合、ノードの数が非常に多くなり、元の形状の再現性が悪い、また、領域の内か外のデータしかないので領域間の交点や交線を求めるのが困難であり、領域の境界となっていない曲線を処理することができないという欠点がある。

2.2 他の quadtree

現在ではさまざまな quadtree が提案されているが、ここでは、本論文と関係が深い edge quadtree, face quadtree, PM quadtree について簡単に述べる。face quadtree は、face octree を2次元で実現したもので、edge quadtree の実装の一種である。

• edge quadtree

edge quadtree^{6),10)}は、領域の境界を直線に近似し、この情報をノードに付加したものである。こ

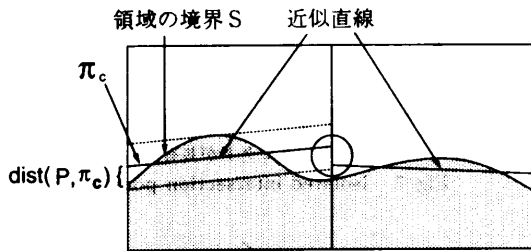


図2 曲線の直線による近似と近似直線がずれる例

Fig. 2 Region boundary approximation scheme and discontinuity of region boundary approximations.

のように、境界の近似情報があるため、classical quadtree と比較して少ない分割で表現できる等の利点があるが、多数の曲線が1点で交わる場合、処理が困難になったり、直線近似のために再現性が低下する、あらかじめ領域の境界となっている線しか扱えない等の問題がある。

edge quadtree の実装の一種として、face quadtree^{12),15)} (提案は3次元²⁾) が提案されている。face quadtree は classical quadtree の終端ノード WHITE, BLACK に加えて境界の情報を持った FACE ノードを導入したものである。FACE ノードは、領域の境界の一部分を含み、ノード中の境界が十分直線に近いものである。領域の境界が十分直線に近いというのは、図2に示すように適当な直線 π_c に対し、そのクオドラントに含まれる領域の境界 S 上のすべての点 P が $\text{dist}(P, \pi_c) \leq \epsilon$ を満たしていることをいう。FACE ノードには近似直線 π_c の情報 (方程式) を格納する。

face quadtree には、近似により図2のようにクオドラントの境界で近似直線がずれるという問題^{2),12),15)}がある。

● PM quadtree

PM quadtree^{5),8),10)} は、edge quadtree と line quadtree^{4),10)} を改良したもので、多角形を近似することなく表すことができる。PM quadtree には、PM₁ quadtree, PM₂ quadtree, PM₃ quadtree があるが、ここでは、PM₁ quadtree について簡単に述べる。PM₁ quadtree では、以下の3つの条件のいずれかを満たすものを終端ノードとする。

- 0 または 1 つの頂点が存在する。
- 頂点が 1 つ存在する場合は、そのノードに存在するすべての線がその頂点を通る。
- 頂点が存在しない場合は、たかだか 1 本の線が存在する。

PM quadtree は、多数の線が1点で交わる場合でも処理可能であるという利点があるが、曲線を境

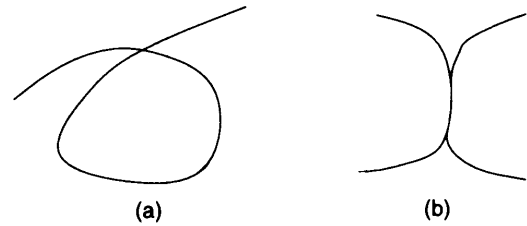


図3 文献13), 14)で除外していた線の状況:(a)自己交差のある線;(b)重なりのある線

Fig. 3 Line configurations excluded in papers 13), 14): (a) self-intersection; (b) overlap.

界とする領域を表現できない、あらかじめ領域の境界となっている線しか扱えない等の問題がある。

3. 部分空間分類

2章に示したように従来の quadtree の問題の中には階層的に分割した空間を用いることによる問題と、対象の曲線を近似することによる問題が混在している。本章では、近似と分割を分離し、分割を分類と終了条件に分けて議論する。部分空間の分類としては、図3のように自己交差のある線 (self-intersection) と、複数の線が重なっている (overlap)^{*1} ような状況も含めて、複数の線が存在する 2次元空間の部分空間分類^{*2}を行う。

人が与える定義上の線を線オブジェクト^{*3}と呼ぶ。線オブジェクトは、定義により識別されるので、同じ位置を占める線を定義した場合でもそれぞれ識別される。線オブジェクトのクオドラント中の部分^{*4}を c 線と呼ぶ。図4の細線と太線を合わせたものが線オブジェクトで、太線が c 線である。c 線は連続な区間を 1つとして識別するので図4の c 線数は2となる。

2次元空間に線オブジェクトが数多く存在する場合、その空間を quadtree により再帰的にクオドラントに分割すれば、クオドラント中の c 線は何種類かの単純な状況に限定できる可能性がある。そのような限定ができれば、領域のフィリング、領域間の集合演算などの処理をそれら単純な状況に対して記述すればよく、

*1 本論文では、2つの線の共通部分が0でない長さを持つ線となる時、線が重なっていると呼ぶ。

*2 ここでは、quadtree 表現を前提としているので、部分空間はクオドラントにより構成される。

*3 本論文で扱う線オブジェクトは、C¹級で連続、有界、かつ、有向なパラメトリック形式の関数で表現されているものとする。また、線オブジェクトの向きは、媒介変数が増加する向きとする。

*4 本論文でクオドラント中と表現した場合は、クオドラントの内部と境界を合わせたものを表す。クオドラント中の線オブジェクトとは、クオドラント中に1次元の広がりを持つ線オブジェクトを表し、1点のみクオドラント中に存在する線オブジェクトはそのクオドラント中に存在しているとは考えない。

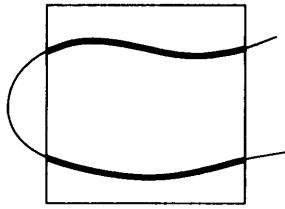


図4 線オブジェクトとc線
Fig.4 Line-object vs. c-line.

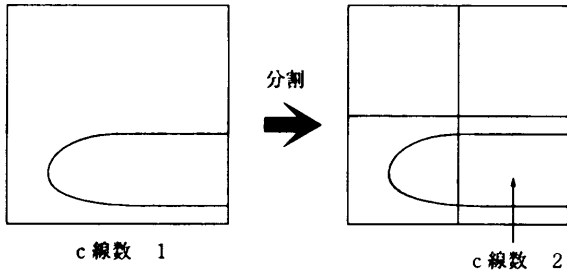


図5 分割によりc線数が大きくなる例
Fig.5 Increase in the number of c-lines by subdivision.

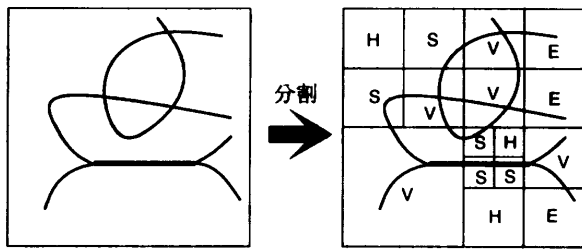


図6 分割により生じる状況
Fig.6 Subspace configuration by recursive subdivision.

単純化される。単純化として、クオドラント中のc線数をできる限り小さくすることを考える。分割により図5のように、c線数が大きくなることもあるが、線オブジェクトがパラメトリックな形式で C^1 級の関数で表現されるなら、あるところまで分割するとc線数は収束し、それ以上分割してもc線数を小さくすることができなくなる。

線オブジェクトは必ず1回は分割したうえで、c線数をそれ以上小さくすることができなくなるまで分割したクオドラントを単純になったと呼び、そこまで分割することを十分分割と呼ぶ。

図6の左図のような線オブジェクトが複数存在する空間を、再帰的にクオドラントに十分分割すると、必ず以下の5つの単純な状況に帰着させることができる。

- (1) クオドラント中にc線が存在しない状況 (Hクオドラント)
- (2) クオドラント中にc線が1つ存在し、かつ、そのc線に対応する線オブジェクトのII交点ではない端点が1つ存在する状況 (Eクオドラント)



図7 2つの線オブジェクトが共有する孤立点
Fig.7 Isolated intersection points.

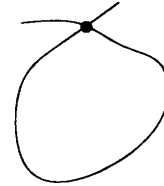


図8 自己交差点
Fig.8 Isolated self-intersection point.

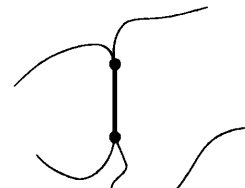


図9 線オブジェクトの重なり端点
Fig.9 End points of overlap line.

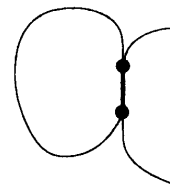


図10 自己交差線の端点
Fig.10 End points of self-intersection line.

- (3) クオドラント中にc線が1つ存在し、かつ、そのc線に対応する線オブジェクトの端点が存在しない状況 (S1クオドラント)
- (4) クオドラント中に2つ以上のc線が存在し、それらがすべて重なっており、II交点が存在しない状況 (S2クオドラント)
- (5) クオドラント中のc線がすべて1つのII交点で交わる状況 (Vクオドラント)

ここで、II交点とは、以下のものをいう。

- 2つの線オブジェクトが共有する孤立点 (図7を参照)
- 1つの線オブジェクトによる自己交差点 (図8を参照)
- 線オブジェクトの重なり端点 (図9を参照)
- 1つの線オブジェクトによる自己交差線の端点 (図10を参照)

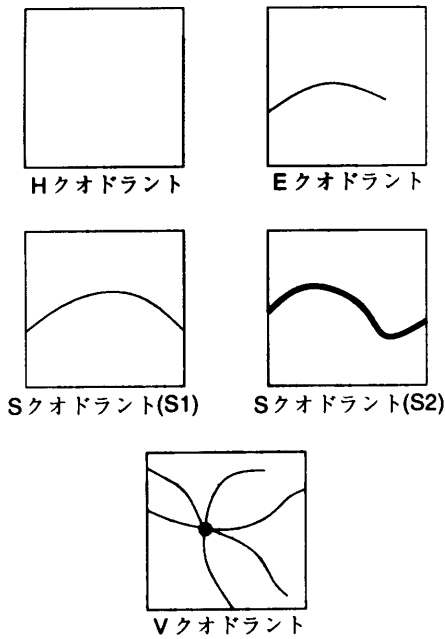


図 11 HESV 分類

Fig. 11 HESV subspace classification.

ll 交点は、その存在する位置で識別される。

以上のものを ll 交点であると定義することにより自己交差、および線オブジェクト間の重なりも含めたクオドラントの分類が可能となる。

S1, S2 クオドラントは状況が類似しているの、それらをまとめて、S クオドラントと呼ぶことにする。H, E, S, V の 4 種類のクオドラントによる部分空間分類を HESV 分類と呼ぶ。その例を図 11 に示す。

H, E, S, V の 4 つのクオドラントを部分空間分類の基礎とするのは、H クオドラントは十分な分割により H クオドラントが生じ、E クオドラントは十分な分割により H, E, S クオドラントが生じ、S クオドラントは十分な分割により、H, S クオドラントが生じ、V クオドラントは十分な分割により H, E, S, V クオドラントが生じるので、それらのクオドラントをそれ以上再帰的に分割しても必ず同種のクオドラントが出てきてしまい、c 線数をそれ以上小さくできないからである。

4. 内外判定

内外判定とは、1 つ以上の線オブジェクトが存在する空間から、以下で定義する内部領域をすべて見つけることである。別の見方をすると、それはまた、一種のリージョンフィリング^{1),3),7),9)}と見ることができる。一般のリージョンフィリングは離散的なグリッドを塗りつぶすものであるが、ここで述べる内外判定は、それを連続空間に拡張したものとも考えられる。

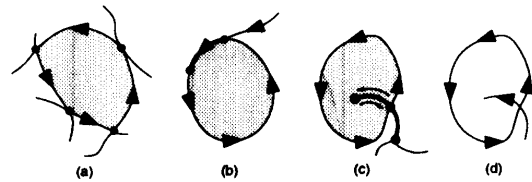


図 12 内部領域、外部領域の例

Fig. 12 Inner-region and outer-region samples.

4.1 内部領域の定義

内部領域を次のように定義する。

線オブジェクトによって反時計回りに囲まれている領域を内部領域とする。ただし、内部領域の境界線以外の部分には、線オブジェクトは存在してはならない。

内部領域でない領域を外部領域と呼ぶ。図 12 の (a), (b), (c) に内部領域の例を、(d) に外部領域の例を示す。

4.2 クオドラントにおける内部領域の表現

以下では、線オブジェクトとクオドラントの境界線の交わりが構成する孤立点、および線オブジェクトとクオドラントの境界線の重なり端点を lq 交点と呼ぶ。lq 交点のうち、線オブジェクトがクオドラントの境界を横断してそのクオドラントに入ってくる点を lq^{c+} 交点、線オブジェクトがクオドラントの境界を横断してそのクオドラントから他のクオドラントに出ていく点を lq^{c-} 交点、線オブジェクトとクオドラントの境界線の交わりで、横断していないものを lq^{nc} 交点と呼ぶ。

クオドラントにおいて、内部領域の部分を反時計回りに囲うように、以下の点をループ状に並べることにより内部領域を表現する。

- lq^{c+} 交点
- lq^{c-} 交点
- lq^{nc} 交点
- クオドラントの頂点
- ll 交点

lq^{c+} 交点より c 線の向きに従って、内部領域である部分を囲うように上記の点を出現順に並べ、ループを構成する。そのループを内部ループと呼ぶ。内部ループは、上記の点と矢印からなり、矢印には次のような属性を付加する。

- c クオドラントの境界線と重なっていない c 線の一部を表している。
- lq クオドラントの境界線と c 線が重なっている部分を表している。
- qb クオドラントの境界線を表している。

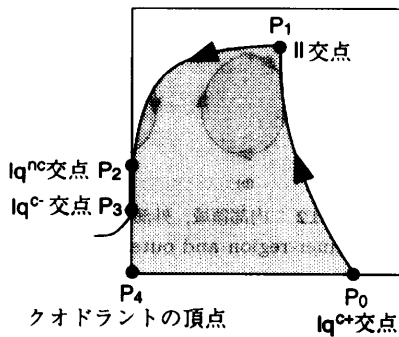


図 13 内部ループ
Fig. 13 Inner-loop.

図 13 における内部ループは $P_0 \rightarrow P_1 \rightarrow P_2 \rightarrow P_3 \rightarrow P_4 \rightarrow P_0$ である。

4.3 内外判定の手法

内外判定は以下の 2 つのプロセスからなる。

● 局所的内外判定

HESV 分類の各クオドラントに対してそのクオドラント内で局所的に内外判定を行い、内部領域になりうる可能性のある部分をすべて内部領域と仮定する。

● 大域的内外判定

局所的内外判定の結果が全体として整合するように、内部領域の仮定が矛盾したところを外部領域と決定する。

4.4 局所的内外判定

局所的内外判定は以下の 2 つのプロセスからなる。

● 内部ループの生成

HESV 分類の各クオドラント各々について、内部ループを生成する。

● クオドラント境界情報の生成

HESV 分類の各クオドラント各々について、内部ループを利用して、クオドラントの境界線にどの部分が内部 (外部) 領域なのか、また、どの部分が c 線と重なっているのかという情報を付ける。その情報のことをクオドラント境界情報と呼ぶ。クオドラントの境界線に付加する情報には以下のものがある。

in 内部領域にあることを表している。

out 外部領域にあることを表している。

cl c 線と重なっていることを表している。

4.4.1 内部ループの生成

HESV 分類の各クオドラントにおける局所的内外判定の手法を以下に示す。

● H クオドラント

H クオドラントは、クオドラント全体が内部領域、

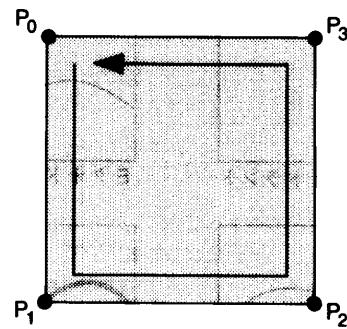


図 14 H クオドラントの内部ループ
Fig. 14 Inner-loop of H quadrant.

または、外部領域である。内部領域になりうる可能性のある部分をすべて内部領域と仮定するという方針に従い、ここでは、H クオドラント全体を内部領域と仮定する。したがって、H クオドラントの 4 つの頂点を $P_0 \sim P_3$ とする (図 14 を参照) と、 $P_0 \rightarrow P_1 \rightarrow P_2 \rightarrow P_3 \rightarrow P_0$ のような内部ループを生成する^{*}。

● E クオドラント

E クオドラントには内部領域が存在することはないので外部領域と決定する。したがって、内部ループは生成しない。

● S クオドラント

c 線とその c 線の進行方向左側にあるクオドラントの境界線とで囲まれた部分を、内部領域と仮定する。実際には、 $[Iq^{c+} \text{ 交点}] \rightarrow [Iq^{c-} \text{ 交点}] \rightarrow [Iq^{c-} \text{ 交点}] \rightarrow [\text{クオドラントの頂点}]$ 、または、 $Iq^{nc} \text{ 交点} \rightarrow [Iq^{c+} \text{ 交点}]$ と順にたどり、矢印に属性を付加しながら、 Iq^{c+} 交点より反時計回りに内部領域を囲む内部ループを生成する。このとき、矢印の両端の点が等しいものは、1 点にまとめる。

生成される内部ループの数は、1 または 2 である。生成される内部ループの数が 2 となるのは、 c 線数が 2 以上 (線オブジェクトが重なっている場合) で、かつ、それら c 線のうち少なくとも 1 つの c 線の向きが異なっている場合である。

図 15 の内部ループは $P_0 \rightarrow P_1 \rightarrow P_2 \rightarrow P_3 \rightarrow P_4 \rightarrow P_0$ となる。

● V クオドラント

V クオドラントの内外判定は、内部ループが 0 個以上いくつでも生成される可能性があるということと内部ループ中に II 交点が存在するという 2

^{*} 反時計回りに内部ループを生成するならば、どの頂点から生成してもよい。

^{**} *は 0 回以上の繰り返しを表す。

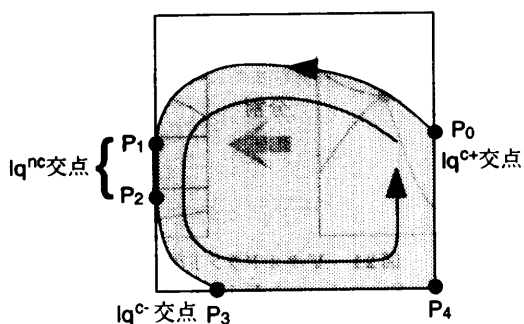


図 15 Sクオドラントの内部ループ
Fig. 15 Inner-loop of S quadrant.

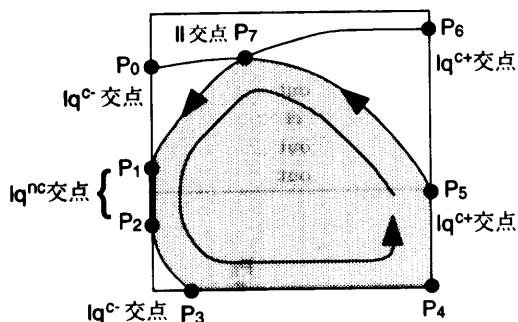


図 16 Vクオドラントの内部ループ
Fig. 16 Inner-loop of V quadrant.

点が変わるだけで、基本的にSクオドラントのアルゴリズムと同様である。実際には、 $[lq^{c+} \text{ 交点}] \rightarrow [lq^{nc} \text{ 交点}] \rightarrow [lq^{c-} \text{ 交点}] \rightarrow [lq^{c+} \text{ 交点}]$ よりクオドラントの境界を時計回りに探索して見つかった lq^{c-} 交点 (lq^{c+} 交点が見つかった場合はそれは内部領域ではない) \rightarrow [クオドラントの頂点, または, lq^{nc} 交点] $\rightarrow [lq^{c+}$ 交点]により, lq^{c+} 交点ごとに反時計回りに内部領域を囲む内部ループを生成する。このとき, 矢印の両端点が等しいものは1点にまとめる。

図16の場合, P_5 と P_6 の2つの lq^{c+} 交点があるが, P_6 を lq^{c+} 交点として内部ループを生成する場合, lq^{c-} 交点を P_6 より時計回りにクオドラントの境界線を探索すると lq^{c+} 交点 P_5 が見つかるので, P_6 を lq^{c+} 交点とする内部ループは存在しない。したがって, 生成される内部ループは1つで, $P_5 \rightarrow P_7 \rightarrow P_1 \rightarrow P_2 \rightarrow P_3 \rightarrow P_4 \rightarrow P_5$ である。

4.4.2 クオドラント境界情報の生成

クオドラント境界情報は, クオドラントの境界線の属性をH, E, S, Vに従って次のように決定していく。デフォルトでは属性outを付けておく。

● Hクオドラント

Hクオドラントは, すべてが内部領域であると仮

表 2 内部ループの矢印の属性とクオドラント境界情報の属性の関係

Table 2 Relationship between attributes of inner-loop arrow and attributes of quadrant-boundary-information.

内部ループの矢印の属性	クオドラント境界情報の属性
c	なし
lq	cl
qb	in

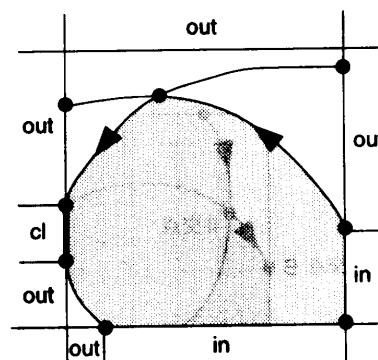


図 17 図 16 のクオドラント境界情報
Fig. 17 Quadrant-boundary-information for Fig. 16.

定しているの, クオドラントの境界線すべてに, 内部領域であるという属性inを付ける。

● Eクオドラント

デフォルトのまま属性outを付ける。(Eクオドラントは, すべてが外部領域であると決定しているので, クオドラントの境界線すべてが, 外部領域である。)

● S, Vクオドラント

S, Vクオドラントに存在するすべての内部ループそれぞれに関して, 内部ループ中の各矢印の属性により, その矢印が示しているクオドラントの境界線の部分に対して, 表2のように属性を付ける。

図16に関するクオドラント境界情報を図17に示す。

4.5 内外判定のための部分空間分類の拡張

局所的内外判定では, クオドラントの境界線とc線との交わりから内部領域の仮定を行う。したがって, クオドラント内部におけるc線の状況は考慮しない。そのために, HESV分類におけるVクオドラントに関しては以下のような場合, 内部領域の仮定が正しく行えない可能性がある。

(1) クオドラント中にII交点ではない線オブジェクトの端点が存在する場合

図18の場合, lq^{c+} 交点Bより時計回りにクオドラントの境界線を探索すると lq^{c-} 交点Aが見つかるので, グレーの部分内部領域と仮

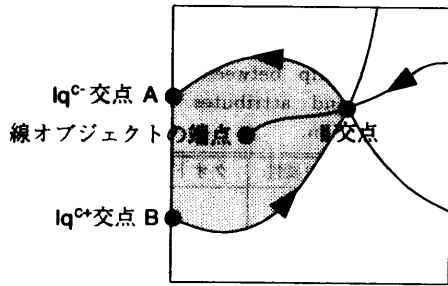


図 18 内部領域を正しく仮定できない例 1

Fig. 18 Incorrect inner-region assumption: case 1.

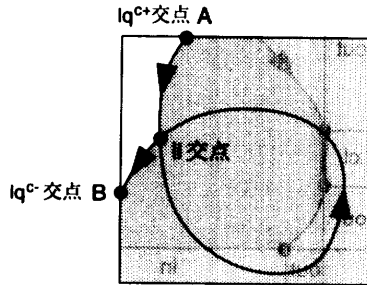


図 19 内部領域を正しく仮定できない例 2

Fig. 19 Incorrect inner-region assumption: case 2.

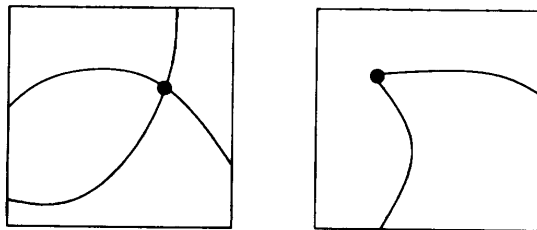


図 20 nV クオドラント

Fig. 20 nV quadrants.

定してしまう。

(2) 1本のc線によるループが存在する場合

図 19 の場合, lq^{c+} 交点 A より時計回りにクオドラントの境界線を探索すると lq^{c-} 交点 B が見つかるので, グレーの部分を実内部領域と仮定してしまう。

上に示した (1) と (2) の状況は, V クオドラントをさらに分割することにより避けられる。クオドラント中に II 交点ではない線オブジェクトの端点が存在しない, かつ, 1本のc線によるループが存在しない V クオドラントを nV クオドラント (図 20 を参照) と呼ぶ。そして, H, E, S, nV によるクオドラントの分類を HESnV 分類と呼ぶ。内外判定はこの HESnV 分類を用いて行う。

図 21 に, V クオドラントを nV クオドラントまで分割した例を示す。

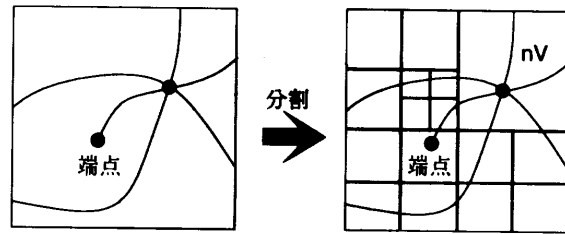


図 21 V クオドラントの分割

Fig. 21 Recursive subdivision of V quadrant.

表 3 クオドラント境界情報の整合表

Table 3 Agreement of quadrant-boundary-information.

クオドラント境界情報の属性の組合せ		
in	in	整合
in	cl	ありえない
in	out	矛盾
cl	cl	整合
cl	out	矛盾
out	out	整合

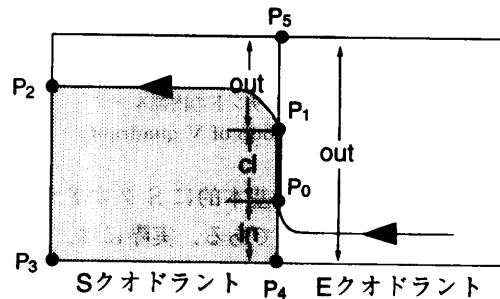


図 22 クオドラント境界情報の比較 1

Fig. 22 Comparison of quadrant-boundary-information: case 1.

4.6 大域的内外判定

大域的内外判定は, HESnV 分類の各クオドラントのクオドラント境界情報が整合するように, quadtree を走査しながら隣接クオドラント間でクオドラント境界情報を比較し, 矛盾した境界を境界を持つ領域を外部領域と決定していく。外部領域と決定するというのは, 矛盾しているクオドラント境界情報に対応する内部ループを消去することにより行う。内部ループが消去された場合, それに従ってクオドラント境界情報を更新する。表 3 にクオドラント境界情報の整合表を示す。

図 22 のような場合, グレーで示された部分が S クオドラントの局所的内外判定で内部領域と仮定され, 内部ループ $IL: P_0 \xrightarrow{lq} P_1 \xrightarrow{c} P_2 \xrightarrow{qb} P_3 \xrightarrow{qb} P_4 \xrightarrow{qb} P_0$ が生成される。

一方, E クオドラントには, 局所的内外判定により, 内部ループは生成されない。

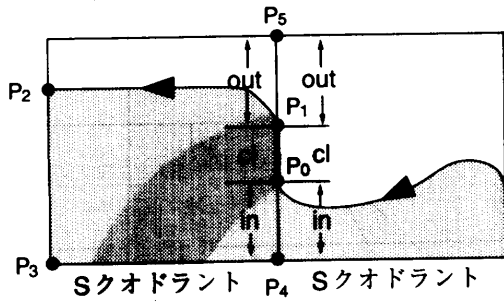


図 23 クオドラント境界情報の比較 2

Fig. 23 Comparison of quadrant-boundary-information: case 2.

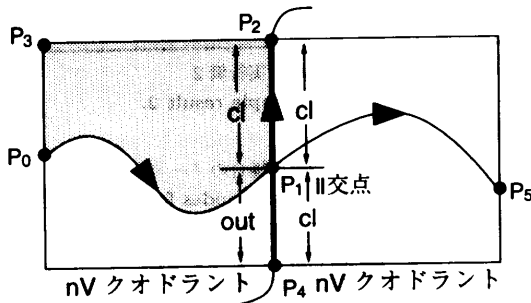


図 24 クオドラント境界情報の比較 3

Fig. 24 Comparison of quadrant-boundary-information: case 3.

大域的内外判定において、S クオドラントと E クオドラントが接している境界線 P_5P_4 に着目すると、 P_5P_1 ではクオドラント境界情報は整合するが、 P_1P_4 では矛盾する。したがって、S クオドラントにおいて、 P_1P_4 の上を通る内部ループ IL が消去され、クオドラント境界情報はすべて out となる。

図 23 の場合、左右の S クオドラント各々が接する境界線 P_5P_4 に関するそれぞれのクオドラント境界情報は、左の S クオドラントでは P_5P_1 が out であり、 P_1P_0 が cl、 P_0P_4 が in である。また、右の S クオドラントも同様であるので、隣接クオドラント間でクオドラント境界情報が整合している。

図 24 の場合、左の nV クオドラントには、局所的内外判定により、内部ループ $IL_{l1} : P_0 \rightarrow P_1 \rightarrow P_2 \rightarrow P_3 \rightarrow P_0$ が生成され、右の nV クオドラントには、局所的内外判定により、内部ループ $IL_{r1} : P_1 \rightarrow P_2 \rightarrow P_1$ と $IL_{r2} : P_4 \rightarrow P_1 \rightarrow P_4$ が生成される。左右の nV クオドラント各々が接する境界線 P_2P_4 に関するそれぞれのクオドラント境界情報は、左の nV クオドラントでは P_2P_1 が cl、 P_1P_4 は out であり、右の nV クオドラントでは P_2P_1 が cl、 P_1P_4 も cl である。したがって、 P_2P_1 ではクオドラント境界情報が整合しているが、 P_1P_4 では矛盾しているため、 P_1P_4 を含む

内部ループ IL_{r2} を右側の nV クオドラントから消去し、 P_1P_4 のクオドラント境界情報を out にする。

このようなクオドラント境界情報の比較は各クオドラントのクオドラント境界情報に矛盾がなくなるまで反復する。反復は、内外の整合性がとれない内部ループが削除されていくだけの一方の処理をするだけなので、振動せず必ず有限回で終了する。また、矛盾するクオドラント境界情報を含む内部ループは必ず削除され、整合するクオドラント境界情報を含む内部ループは削除されないことから結果の一意性も明らかである。内部領域がクオドラントの座標系に依存せず決まることは、

- 線オブジェクトにより囲まれていない部分には、線オブジェクトが少なくとも 1 つ存在するという仮定により、E クオドラントから領域でないことが伝播する。
 - 線オブジェクトにより囲まれている部分の境界以外の部分に線オブジェクトが存在する場合はその線オブジェクトの端点から、分割するとその線オブジェクトの端点から必ず E クオドラントが生じ、E クオドラントから領域でないことがクオドラント境界情報の比較により全体に伝播する。
 - 線オブジェクトの向きが整合しない領域は、分割すると必ず nV クオドラントが生じ、nV クオドラントから内部領域でないということが、クオドラント境界情報の比較により全体に伝播する。
- ということから分かる。

5. 実行例

HESnV 分類を利用した内外判定を行うシステムを部分的に実装し、実験した結果を報告する。実験は、富士通 S-4/20L ワークステーションの X Window 上で行った。プログラミングには C 言語を使用し、線オブジェクトにはパラメトリック形式の階数 4 (次数 3) の B-spline 曲線¹¹⁾を用いた。3 つの実行例を以下に示す。

- 4 本の開曲線と 2 本の閉曲線による内外判定の例
図 25 に実行結果を示す。また、表 4 にノード数を示す。内部領域の中に穴があるが、これは、時計回りに向き付けした閉曲線によるものである。穴の空いた内部領域も処理可能であることが分かる。内部領域を構成しない線オブジェクトが存在しているが、E クオドラントの導入により処理可能となった。
- 自己交差による内外判定の例
図 26 に実行結果を示す。また、表 5 にノード数

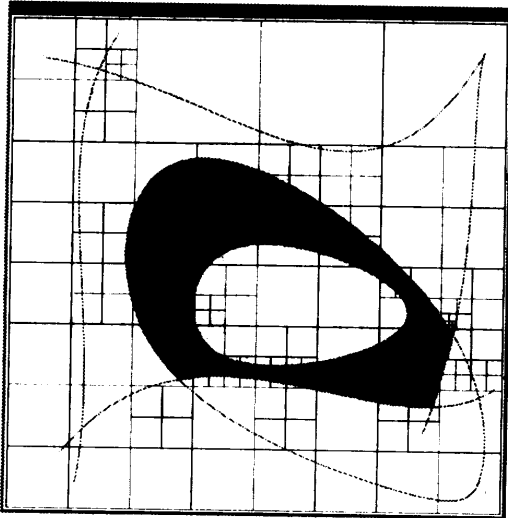


図 25 実行例 1

Fig. 25 Sample result 1.

表 4 図 25 の場合のノード数

Table 4 The number of nodes for the result shown in Fig. 25.

HESnV 分類	ノード数
H	37
E	6
S	98
nV	7
非終端ノード数	49
総ノード数	197

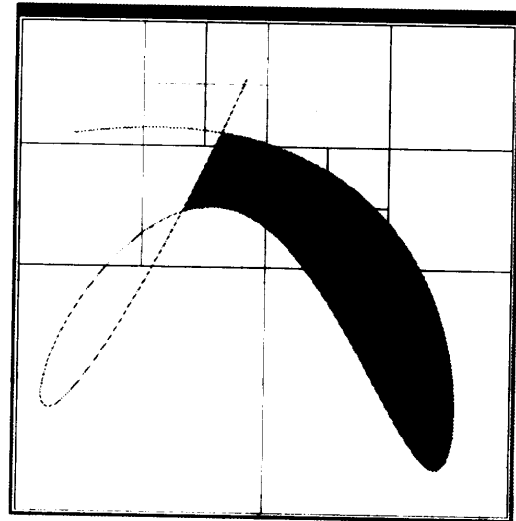


図 26 実行例 2

Fig. 26 Sample result 2.

表 5 図 26 の場合のノード数

Table 5 The number of nodes for the result shown in Fig. 26.

HESnV 分類	ノード数
H	2
E	2
S	10
nV	2
非終端ノード数	5
総ノード数	21

を示す。2回自己交差しているが、 Π 交点の導入により、このような自己交差の場合も内外判定可能となった。

● 細長い領域の内外判定の例

図 27 に実行結果を示す。また、表 6 にノード数を示す。HESnV 分類では、近似を行わないので、細長い内部領域では、c線を分離しようとするため、分割が深くなりノード数が増加する傾向にある。分割が浅い場合でも、分割線（クオドラントの境界線）がc線とc線の間を通過してc線が分離できることがあり、分割の深さとノード数が変化する。

6. 考 察

- 文献 14) の HES(n)V 分類では、自己交差、および線が重なっている場合を除外していたが、 Π 交点の概念を導入することにより、それらの場合も含めて、2次元空間中の線に関する一般的な部分空間分類を行うことができた。また、Eクオドラントの導入により領域の境界となっていない曲線が存在しても処理できるようになった。本論文で

は、quadtree 表現を利用することを考えているが、この分類は他の空間分割法にも適用可能であると思われる。

- 文献 14) の内外判定では、クオドラントの境界線上に Π 交点が存在する場合や、線がクオドラント境界線と重なっている場合、内外判定が正しく行えなかったが、クオドラント境界情報の導入によってそれらの場合も処理可能となった。
- HESnV 分類を用いることにより、デジタル空間上のグリッドのフィリングを連続空間上のフィリングに拡張可能であることを確認した。
- 次数の高い曲線では、交点を計算するコストが高くなったり、交点を計算できない場合がある。HES(n)V 分類では、線間の交点計算を前提としているため、このような曲線へ適用するには、実用的でない場合がある。これに関して、交点計算が困難な場合は計算をせず、近似を行うような分割の終了条件を検討中である。また、その分割の終了条件を基に、内外判定を行うことも検討中である。
- 近似を前提としている quadtree 表現では内外判定

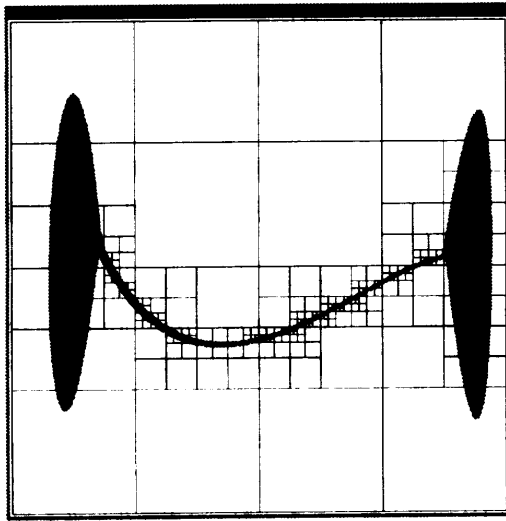


図 27 実行例 3
Fig. 27 Sample result 3.

表 6 図 27 の場合のノード数
Table 6 The number of nodes for the result shown in Fig. 27.

HESnV 分類	ノード数
H	90
E	0
S	198
nV	4
非終端ノード数	97
総ノード数	389

を行う場合、図 28 (a) のように曲線を直線に近似していたためにさらに分割していたが、HES(n)V 分類では図 28 (b) のように、少ない分割で分類可能となる。また、HES(n)V 分類では近似を行わないので、近似直線が隣接クオドラント間でずれるという問題も避けられる。

- 図 29 (a) のように曲線を直線に近似する quadtree 表現では、その近似方式により、近似精度のよくない近似直線が生成される可能性があったが、HES(n)V 分類では図 29 (b) のように近似を行わないので、そのような状況を避けられるようになった。

7. 結 論

本論文では、従来の quadtree の拡張方式において一体となっていた近似、部分空間の分類と分割の終了条件を分離することにより、2次元空間中に存在する線を自己交差や、線どうしが重なっている場合も含めて、従来の quadtree の近似や分割の終了条件により生じた不都合を回避しながら、処理の基本となる部分空間分類を明確にした。また、その分類を拡張した

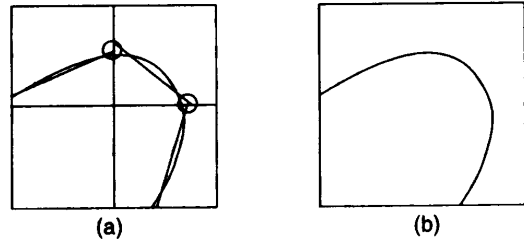


図 28 分割の減少：(a) 直線近似を行う quadtree；
(b) HES(n)V 分類

Fig. 28 Elimination of redundant subdivision:
(a) quadtree with approximation; (b) HES(n)V classification.

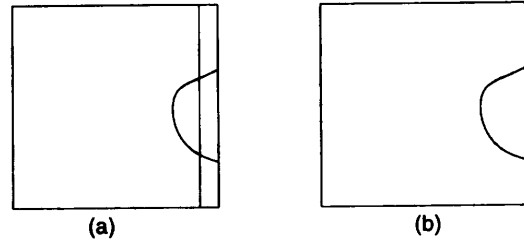


図 29 近似度のよくない直線近似の削除：(a) 直線近似を行う quadtree；(b) HES(n)V 分類

Fig. 29 Elimination of improper approximation line:
(a) quadtree with approximation; (b) HES(n)V classification.

HESnV 分類を内外判定に適用した部分空間分類による内外判定の方法を提案し、実験により、この方式の特徴を具体例により示した。この内外判定法は、従来のデジタル空間におけるフィリングをある条件を満たす連続空間上のフィリングに拡張したもの一種と見られることもできる。

今後の課題は、集合演算などより多くの演算に対し分類を拡張するとともに、ここで提案した内外判定の応用を検討していくことである。

参 考 文 献

- 1) Bauer, M.A., Feeney, S.T. and Gargantini, I.: Parallel 3-D Filling with Octrees, *Journal of Parallel and Distributed Computing*, Vol.22, No.1, pp.121-128 (1994).
- 2) Brunet, P. and Vinacua, A.: Surfaces in Solid Modeling, *Geometric Modeling - Methods and Applications*, Hagen, H. and Roller, D. (Eds.), Springer-Verlag, pp.17-34 (1991).
- 3) Foley, J.D.: *Computer Graphics - Principles and Practice*, pp.446-450, 979-986, Addison Wesley (1990).
- 4) Samet, H. and Webber, R.E.: On Encoding Boundaries with Quadtrees, *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol.6, No.3, pp.365-369 (1984).

- 5) Samet, H. and Webber, R.E.: Storing a Collection of Polygons Using Quadtrees, *ACM Trans. Graphics*, Vol.4, No.3, pp.182-222 (1985).
- 6) Shneier, M.: Two Hierarchical Linear Feature Representations: Edge Pyramids and Edge Quadtrees, *Computer Graphics and Image Processing*, Vol.17, No.3, pp.211-224 (1981).
- 7) Oliver, M.A. and Wiseman, N.E.: Operations on Quadtree Encoded Images, *The Computer Journal*, Vol.26, No.1, pp.83-91 (1983).
- 8) Nelson, R.C. and Samet, H.: A Consistent Hierarchical Representation for Vector Data, *Computer Graphics*, Vol.20, No.4, pp.197-206 (1986).
- 9) Samet, H.: Region Representation: Quadtrees from Boundary Codes, *Comm. ACM*, Vol.23, No.3, pp.163-170 (1980).
- 10) Samet, H.: *The Design and Analysis of Spatial Data Structures*, Addison Wesley (1990).
- 11) 山口富士夫: コンピュータディスプレイによる形状処理工学 [I] [II], 日刊工業新聞社 (1982).
- 12) 森本経宇: face octree 生成アルゴリズムの研究, 筑波大学第3学群情報学類卒業研究論文 (1991).
- 13) 森本経宇: 再帰的空間分割法の研究—曲線と領域への適用—, 修士論文, 筑波大学大学院修士課程理工学研究科 (1993).
- 14) 森本経宇, 山口和紀: 再帰的空間分割法と部分空間分類—曲線で囲まれた領域の内外判定への応用—, 情報処理学会論文誌, Vol.35, No.5,

pp.828-838 (1994).

- 15) 楠 美和: face octree を利用したソリッドモデル表示アルゴリズムの研究, 筑波大学第3学群情報学類卒業研究論文 (1991).

(平成8年3月26日受付)

(平成8年9月12日採録)



森本 経宇 (正会員)

1968年生。1991年筑波大学第3学群情報学類卒業。1993年同大学院修士課程理工学研究科修了。1996年同大学院博士課程工学研究科満期中退。同年シャープ(株)入社。現在、同社技術本部マルチメディア事業化推進本部映像メディア研究所勤務。コンピュータグラフィックスに興味を持つ。電子情報通信学会会員。



山口 和紀 (正会員)

1956年生。1979年東京大学数学科卒業。1981年同大学理学部助手。1985年理学博士(東京大学)。1989年筑波大学電子情報工学系講師。1992年東京大学教養学部助教授。コンピュータグラフィックスとデータベースを主な考察対象とし、モデリング全般に興味を持つ。ACM会員。